V5R2 iDoctor for iSeries Documentation

About This Document
Part I iDoctor for iSeries Overview
.......Chapter 1 General Information
..............1.1 Job Watcher
..............1.2 PEX Analyzer
..............1.3 Heap Analysis Tools for Java
..............1.4 PTDV
.......Chapter 2 Installation
..............2.1 Installing Job Watcher
.....................2.1.1 PTF installation
.....................2.1.2 Manual installation steps for remote service
..............2.2 Installing PEX Analyzer
.....................2.2.1 PTF Prerequisites for PEX Analyzer
.....................2.2.2 Manual installation steps for remote service
..............2.3 Installing Heap Analyzer
..............2.4 Installing PTDV
..............2.5 How to authorize use of iDoctor for iSeries
..............2.6 Uninstalling iDoctor for iSeries
Part II Server-side components
.......Chapter 1 Job Watcher - job-specific mode
..............1.1 Libraries
..............1.2 Output files
.....................1.2.1 QPYRTJWA - Basic job statistics interval deltas
.....................1.2.2 QPYRTJWAAI - Activation groups
.....................1.2.3 QPYRTJWAC1 - Communications/socket statistics
.....................1.2.4 QPYRTJWAC2 - Communications/socket job information
.....................1.2.5 QPYRTJWAD - Wait bucket descriptions
.....................1.2.6 QPYRTJWAJ1 - Java threads waiting
.....................1.2.7 QPYRTJWAJ2 - Java virtual machine statistics
.....................1.2.8 QPYRTJWAL - Basic job statistics snapshots

....................1.2.13 QPYRTJWARI - Run information

....................1.2.14 QPYRTJWASD - SQL statement host variables

....................1.2.15 QPYRTJWASQ - SQL statements

....................1.2.16 QPYRTJWAST - Call stacks

....................1.2.17 QPYRTJWAS3 - SQL cursors and prepared statements summary

....................1.2.18 QPYRTJWAS4 - SQL open cursors

....................1.2.19 QPYRTJWAS5 - SQL prepared statement arrays

....................1.2.20 QPYRTJWAT2 - Enum (wait point) descriptions

....................1.2.21 QPYRTJWA1T - First time thread/task file

....................1.2.22 QPYRTJWDFN - Job watch definitions

....................1.2.23 QPYRTJWAT - Totals for QPYRTJWA

....................1.2.24 QPYRTJWSYS - System information

....................1.2.25 QPYRTJWRDB - Rule definitions

.............1.3 Commands

....................1.3.1 WCHJOB - creates a job watch

....................1.3.2 ENDWCHJOB - ends a job watch

....................1.3.3 ADDPRDACS - apply an access code

....................1.3.4 DLTWCH - delete a job watch

....................1.3.5 JWAFTERIPL - resets the LIC wait buckets

....................1.3.6 ANZJWSTK - displays call stacks

.............1.4 Rules and actions

....................1.4.1 Rule definition file

.......Chapter 2 Job Watcher - system-wide mode

.............2.1 Libraries

.............2.2 Output files

....................2.2.1 QPYRTSWLG1 - Basic job statistics "log file"

....................2.2.2 QPYRTSWLG2 - Basic job statistics "delta file"

....................2.2.3 QPYRTSWLG3 - Basic job statistics "snapshot file"

....................2.2.4 QPYRTSWALX - Run information

....................2.2.5 QPYRTSWDFN - Job watch definition

....................2.2.6 QPYRTSWALQ - SQL statements

....................2.2.7 QPYRTSWALS - Call stacks

....................2.2.8 QPYRTSWIST - Totals for the interval summary file QPYRTSWIS1

....................2.2.9 QPYRTSWIS1 - Interval summary file

....................2.2.10 QPYRTSWNM1 - CHGWCHNAM command cross reference file

....................2.2.11 QPYRTJWSYS - System information

....................2.2.12 QPYRTSWRDB - Rule definitions

.............2.3 Commands

....................2.3.1 WCHSYS - creates a job watch

.....................2.3.2 ENDWCHSYS - ends a job watch

.....................2.3.3 ADDPRDACS - apply an access code

.....................2.3.4 DLTWCH - delete a job watch

.....................2.3.5 JWAFTERIPL - resets the LIC wait buckets

.....................2.3.6 CHGWCHNAM - desensitize customer names

..............2.4 Rules and actions

.......Chapter 3 PEX Analyzer

..............3.1 Performance Analysis with PEX

.....................3.1.1 PEX Background Information

..............3.2 Overview of Collecting PEX Data

..............3.3 PEX Definitions

..............3.4 Libraries

..............3.5 PEX Analyzer collection files

..............3.6 PEX collection files (from QSYS)

..............3.7 Commands - library QYPINT

.....................3.7.1 STRIDOCCOL - starts a PEX collection

............................3.7.1.1 Equivalent ADDPEXDFN commands for STRIDOCCOL problem types

.....................3.7.2 CPYIDOCCOL - copy a PEX collection

.....................3.7.3 CVTIDOCCOL - convert a PEX *MGTCOL object to a PEX collection

.....................3.7.4 DLTIDOCCOL - delete a PEX collection

.....................3.7.5 ENDIDOCCOL - ends a PEX collection

.....................3.7.6 RSMIDOCCOL - resume a suspended PEX collection

.....................3.7.7 SAVIDOCCOL - save a PEX collection

.....................3.7.8 RSTIDOCCOL - restore a PEX collection

.....................3.7.9 ADDPRDACS - apply an access code

..............3.8 Commands - library SMTRACE

..............3.9 PEX Analyzer analysis files

.....................3.9.1 G_ACTGRP3 - Activation group creates/deletes by job-thread

.....................3.9.2 G_ACTGRP3A - Activation group creates/deletes by job

.....................3.9.3 G_ACTGRP3B - Activation group creates/deletes by program

.....................3.9.4 G_ACTGRP3C - Activation group creates/deletes by program within job-thread

.....................3.9.5 G_CPUP5X - Approximate CPU by priority within interval

.....................3.9.6 G_CPU4X - Approximate Job/Task CPU by interval

.......Chapter 4 PTDV

..............4.1 Overview of Collecting PEX Data

..............4.2 Creating PEX definitions

.....................4.2.1 Standard Java definitions

............................4.2.1.1 Java Method Trace with basic set of Java events

............................4.2.1.2 Java Method Trace with no Object Creates

...........................4.2.1.3 Java Trace with only Object Creates

...........................4.2.1.4 Java Trace with Object Locks

....................4.2.2 Standard ILE/OPM Definitions

....................4.2.3 Custom definitions

....................4.2.4 Performance data provided by Java events

....................4.2.5 OS/400 Release Differences

.............4.3 Generating a PEX collection

....................4.3.1 Generating PEX Definitions

....................4.3.2 Analyzing Java applications

....................4.3.3 Analyzing ILE/OPM applications

....................4.3.4 Starting and Ending the Trace

....................4.3.5 Limiting the Size of the Trace

.......Chapter 5 Heap Analyzer

.............5.1 Libraries

.............5.2 Types of Collections

.............5.3 Heap Analyzer Output Files

.............5.4 WCHJVA command

.............5.5 DLTHEAPWCH command

.............5.6 Using iDoctor to investigate Heap Growth

.............5.7 ADDPRDACS command

Part III Client-side components

.......Chapter 1 Common Graphical Interfaces

.............1.0 Overview

....................1.0.1 Component Property Pages

.............1.1 The Main Window

....................1.1.1 Component Views

....................1.1.2 My Connections View

....................1.1.3 Remote Command Status View

.............1.2 Field Selection Window

.............1.3 Libraries

....................1.3.1 Menu Options

....................1.3.2 Copying

....................1.3.3 Saving

....................1.3.4 Clearing

....................1.3.5 Deleting

....................1.3.6 Renaming

....................1.3.7 Properties

...........................1.3.7.1 Overview

...........................1.3.7.2 Save/Restore

............................1.3.7.3 Locks

............................1.3.7.4 Authorities

..............1.5 The Data Viewer

.....................1.5.1 Menu Options

.....................1.5.2 Toolbar

.....................1.5.3 SQL Query View

.....................1.5.4 Open File Window

.....................1.5.5 Table Views

............................1.5.5.1 Report Properties - Record Quick View

............................1.5.5.2 Report Properties - Query

.....................1.5.6 Graph Views

............................1.5.6.1 Graph Properties - Quick View

............................1.5.6.2 Graph Properties - Query

.....................1.5.7 Spool File Views

..............1.6 Query Definitions

.....................1.6.1 Member Selection

.....................1.6.2 Field Selection

.....................1.6.3 Record Selection

.....................1.6.4 Sort By

.....................1.6.5 Group By

.....................1.6.6 Resetting

.....................1.6.7 Saving

.....................1.6.8 Working with Query Definitions

..............1.7 Graph Definitions

.....................1.7.1 General Page

.....................1.7.2 Primary Y-axis

.....................1.7.3 Secondary Y-axis

.....................1.7.4 Saving

.....................1.7.5 Working with Graph Definitions

..............1.8 Find Window

..............1.9 Preferences

.....................1.9.1 Display

.....................1.9.2 Clipboard

.....................1.9.3 File

.....................1.9.4 PEX Analyzer

.....................1.9.5 Job Watcher

..............1.10 Set Font Dialog

.......Chapter 2 Job Watcher

..............2.1 Job Watcher Basics

..............2.2 Job Watcher Component Property Pages

.....................2.2.1 General

.....................2.2.2 iDoctor Client Jobs

.....................2.2.3 Server configuration

..............2.3 Libraries

..............2.4 Job Watches

.....................2.4.1 Menu options

.....................2.4.2 Job-specific Property Pages

............................2.4.2.1 General

............................2.4.2.2 Creation settings

............................2.4.2.3 Jobs

............................2.4.2.4 Tasks

............................2.4.2.5 Totals

............................2.4.2.6 LIC wait bucket mapping

............................2.4.2.7 System

.....................2.4.3 System-wide Property Pages

............................2.4.3.1 General

............................2.4.3.2 Creation settings

............................2.4.3.3 Totals

............................2.4.3.4 LIC wait bucket mapping

............................2.4.3.5 System

.....................2.4.4 Creating - Start Job Watch Wizard - Job-specific mode

............................2.4.4.1 Welcome

............................2.4.4.2 Options

............................2.4.4.3 Data collection options - General page

............................2.4.4.4 Data collection options - Rule page

............................2.4.4.5 Job/Task Option

............................2.4.4.6 Job Selection

............................2.4.4.7 Task Selection

............................2.4.4.8 Summary

.....................2.4.5 Creating - Start Job Watch Wizard - System-wide mode

............................2.4.5.1 Welcome

............................2.4.5.2 Options

............................2.4.5.3 Data collection options - General page

............................2.4.5.4 Data collection options - Rule page

............................2.4.5.5 Data collection options - Top N jobs

............................2.4.5.6 Data collection options - Featured waiting logging

............................2.4.5.7 Data collection options - Call stack and SQL

............................2.4.5.8 Data collection options - Advanced

...........................2.4.5.9 Summary

....................2.4.6 Deleting

....................2.4.7 Stopping

.............2.5 Job-specific Job Watch summary reports

....................2.5.1 Watched threads

....................2.5.2 Wait graphs - summarized

...........................2.5.2.1 Run/wait time signature

...........................2.5.2.2 Run/wait counts signature

...........................2.5.2.3 Objects waited on

...........................2.5.2.4 Holding threads/tasks

....................2.5.3 CPU graphs - summarized

...........................2.5.3.1 CPU/CPUq time signature

....................2.5.4 DASD/IO graphs - summarized

...........................2.5.4.1 DASD pages allocated/deallocated

...........................2.5.4.2 DASD reads and writes

...........................2.5.4.3 DASD waits

...........................2.5.4.4 I/O activity

...........................2.5.4.5 Logical database I/O activity

...........................2.5.4.6 Page faults

....................2.5.5 IFS graphs - summarized

...........................2.5.5.1 IFS lookup cache

...........................2.5.5.2 IFS reads

...........................2.5.5.3 IFS opens

...........................2.5.5.4 IFS creates/deletes

....................2.5.6 Java graphs - summarized

...........................2.5.6.1 Java heap sizes

....................2.5.7 Other graphs - summarized

...........................2.5.7.1 Numeric data type exceptions

...........................2.5.7.2 State transitions

...........................2.5.7.3 Transactions

.............2.6 Job-specific Job Watch detail reports

....................2.6.1 Interval Properties

...........................2.6.1.1 Record Quick View

...........................2.6.1.2 Call stack

...........................2.6.1.3 Object waited on

...........................2.6.1.4 Holding thread/task

...........................2.6.1.5 Wait buckets

...........................2.6.1.6 Java waits

...........................2.6.1.7 Java Virtual Machine

............................2.6.1.8 Physical I/Os

............................2.6.1.9 Logical I/Os

............................2.6.1.10 Transactions

............................2.6.1.11 IFS

............................2.6.1.12 Job state transitions

.....................2.6.2 Wait graphs - detailed

............................2.6.2.1 Run/wait time signature

............................2.6.2.2 Run/wait counts signature

............................2.6.2.3 Objected waited on

............................2.6.2.4 Holding threads/tasks

.....................2.6.3 CPU graphs - detailed

............................2.6.3.1 CPU/CPUq time signature

.....................2.6.4 DASD/IO graphs - detailed

............................2.6.4.1 DASD pages allocated/deallocated

............................2.6.4.2 DASD waits

............................2.6.4.4 Logical database I/O activity

............................2.6.4.3 I/O Activity

............................2.6.4.5 Page faults

.....................2.6.5 IFS graphs - detailed

............................2.6.5.1 IFS lookup cache

............................2.6.5.2 IFS reads

............................2.6.5.3 IFS opens

............................2.6.5.4 IFS creates/deletes

.....................2.6.6 Java graphs - detailed

............................2.6.6.1 Java heap sizes

.....................2.6.7 Other graphs - detailed

............................2.6.7.1 Numeric data type exceptions

............................2.6.7.2 State transitions

............................2.6.7.3 Transactions

..............2.7 System-wide Job Watch summary reports

.....................2.7.1 Wait graphs by interval - summarized

............................2.7.1.1 Seizes and locks time signature

............................2.7.1.2 Contention time signature

............................2.7.1.3 DASD time signature

............................2.7.1.4 Journal time signature

............................2.7.1.5 Pool overcommitment time signature

............................2.7.1.6 Communications time signature

.....................2.7.2 Wait graphs by job - summarized

............................2.7.2.1 Job signatures ranked by <wait type>

.....................2.7.3 CPU graphs - summarized

............................2.7.3.1 CPU utilization

............................2.7.3.2 CPU/CPUq usage by high/low priority

............................2.7.3.3 Job signatures ranked by CPU

............................2.7.3.4 Job signatures ranked by CPU queueing

.....................2.7.4 DASD/IO graphs - summarized

............................2.7.4.1 DASD pages allocated/deallocated

............................2.7.4.2 DASD reads and writes

............................2.7.4.3 I/O activity

............................2.7.4.4 Logical database I/O activity

............................2.7.4.5 Page faults

.....................2.7.5 IFS graphs - summarized

............................2.7.5.1 IFS lookup cache

............................2.7.5.2 IFS reads

............................2.7.5.3 IFS opens

............................2.7.5.4 IFS creates/deletes

.....................2.7.6 Other graphs - summarized

............................2.7.6.1 State transitions

............................2.7.6.2 Transactions

............................2.7.6.3 Threads active

............................2.7.6.4 Threads created/destroyed

............................2.7.6.5 Job watches initiated by rule breakers

..............2.8 System-wide Job Watch detail reports

.....................2.8.1 Interval Properties

............................2.8.1.1 Record Quick View

............................2.8.1.2 Call stack

............................2.8.1.3 Object waited on

............................2.8.1.4 Holding thread/task

............................2.8.1.5 Wait buckets

............................2.8.1.6 Physical I/Os

............................2.8.1.7 Logical I/Os

............................2.8.1.8 Transactions

............................2.8.1.9 IFS

............................2.8.1.10 SQL

............................2.8.1.11 Job state transitions

.....................2.8.2 Wait graphs - detailed

............................2.8.2.1 Run/wait time signature

............................2.8.2.2 Run/wait counts signature

............................2.8.2.3 Objected waited on

............................2.8.2.4 Holding threads/tasks

.....................2.8.3 CPU graphs - detailed

............................2.8.3.1 CPU/CPUq time signature

.....................2.8.4 DASD/IO graphs - detailed

............................2.8.4.1 DASD pages allocated/deallocated

............................2.8.4.2 DASD waits

............................2.8.4.4 Logical database I/O activity

............................2.8.4.3 I/O Activity

............................2.8.4.5 Page faults

.....................2.8.5 IFS graphs - detailed

............................2.8.5.1 IFS lookup cache

............................2.8.5.2 IFS reads

............................2.8.5.3 IFS opens

............................2.8.5.4 IFS creates/deletes

.....................2.8.7 Other graphs - detailed

............................2.8.7.1 State transitions

.......Chapter 4 PEX Analyzer

..............4.1 PEX Analyzer Basics

..............4.2 Libraries

.....................4.2.1 Menu Options

..............4.3 PEX Collections

.....................4.3.1 Menu Options

.....................4.3.2 Copying

.....................4.3.3 Creating - The PEX Collection Wizard

............................4.3.3.1 Welcome

............................4.3.3.2 Problem Type Question Pages

............................4.3.3.3 Options

............................4.3.3.4 Trace Additional Events

............................4.3.3.5 Job/Task Options

............................4.3.3.6 Job Selection

............................4.3.3.7 Add Jobs Window

............................4.3.3.8 Task Selection

............................4.3.3.9 Add Tasks Window

............................4.3.3.10 Program Selection

............................4.3.3.11 Add Programs Window

............................4.3.3.12 Summary

.....................4.3.4 Defining

.....................4.3.5 Deleting

.....................4.3.6 Generating analyses

.....................4.3.7 Property Pages

............................4.3.7.1 General

............................4.3.7.2 Creation settings

............................4.3.7.3 Definition

............................4.3.7.4 Events

............................4.3.7.5 Collection Jobs

............................4.3.7.6 Collection Tasks

............................4.3.7.7 System Page

............................4.3.7.8 Trace CPU Totals

............................4.3.7.9 Profile CPU Totals

............................4.3.7.10 Statistical CPU Totals

............................4.3.7.11 Library Information

............................4.3.7.12 Programs Profiled

............................4.3.7.13 Collection Size

.....................4.3.8 Transferring to another system

..............4.4 PEX Definitions

.....................4.4.1 Creating or Changing - The PEX Definition Wizard

............................4.4.1.1 Welcome

............................4.4.1.2 Type Selection

............................4.4.1.3 Options

............................4.4.1.4 Program Bracketing Events Selection

............................4.4.1.5 Machine Instruction Option

............................4.4.1.6 Event Selection

............................4.4.1.7 Job/Task Options

............................4.4.1.8 Job Selection

............................4.4.1.9 Add Jobs Window

............................4.4.1.10 Task Selection

............................4.4.1.11 Add Tasks Window

............................4.4.1.12 Program Selection

............................4.4.1.13 Add Programs Window

............................4.4.1.14 Summary

.....................4.4.2 Deleting

.....................4.4.3 Properties

.....................4.4.4 Working With

..............4.5 Analyses

.....................4.5.1 Menu Options

.....................4.5.2 Creating - The Analysis Wizard

............................4.5.2.1 Welcome Page

............................4.5.2.2 Select Analysis To Run

............................4.5.2.3 Trace Subset Options

............................4.5.2.4 Profile Presentation Options

............................4.5.2.5 Statistical Subset Options

............................4.5.2.6 Subset Time

............................4.5.2.7 Subset Job or Task

...................................4.5.2.7.1 Browse Jobs Dialog

...................................4.5.2.7.2 Browse Tasks Dialog

............................4.5.2.8 Subset Storage Pool Page

............................4.5.2.9 Subset Disk Unit Page

............................4.5.3.10 Subset Program Page

............................4.5.3.11 Subset Object Page

............................4.5.3.12 Physical Disk IO Analysis Options

............................4.5.3.13 Summary Page

.....................4.5.3 Deleting

.....................4.5.4 Properties

............................4.5.4.1 General

............................4.5.4.2 Intervals

............................4.5.4.3 Subset Information

............................4.5.4.4 Analysis Size

..............4.6 Report Folders

..............4.7 Analysis Reports

..............4.8 Server-side collection output files

.......Chapter 5 PEX Analyzer analysis reports

..............5.1 Database file logical disk IO reports

.....................5.1.1 DB LDIO Detail by File - Last RRN in block and number of records in block

.....................5.1.2 DB File Detail LDIO by File-Library - No Intervals

.....................5.1.3 DB File Detail LDIO by Interval

.....................5.1.4 DB File Detail LDIO by Job-thread - No Intervals

.....................5.1.5 DB File Detail LDIO by Program - No Intervals

.....................5.1.6 DB File Detail LDIO for File-Library With Highest Total LDIO per Interval

.....................5.1.7 DB File Detail LDIO for Job-thread with Highest Total LDIO per Interval

.....................5.1.8 DB File Detail LDIO for Program with Highest Total LDIO per Interval

.....................5.1.9 DB File Detail LDIO by File-library within Interval

.....................5.1.10 DB File Detail LDIO by File-library within Job-thread - No Intervals

.....................5.1.11 DB File Detail LDIO by File-library within Program - No Intervals

.....................5.1.12 DB File Detail LDIO by Job within Interval

.....................5.1.13 DB File Detail LDIO by Job-thread within File-library - No intervals

.....................5.1.14 DB File Detail LDIO by Job-thread within program - No Intervals

.....................5.1.15 DB File Detail LDIO by Program within File-library - No Intervals

.....................5.1.16 DB File Detail LDIO by Program within Interval

.....................5.1.17 DB File Detail LDIO by Program within Job-thread - No Intervals

..............5.2 CPU Profile by job priority reports

.....................5.2.1 Approximate CPU by priority within Interval

.....................5.2.2 Approximate Job/Task CPU by interval

..............5.3 CPU Profile Summary (TPROF) reports

.....................5.3.1 CPU Profile Summary (TPROF)

..............5.4 Events count reports

.....................5.4.1 Event Count by JobThread/Task

.....................5.4.2 Event Count by JobThread/Task Event Type/Subtype

..............5.5 Physical disk IO reports

.....................5.5.1 IO Total

.....................5.5.2 IO Total by Interval

.....................5.5.3 IO Total by Interval/IO-Type

.....................5.5.4 IO Total by Interval/Object

.....................5.5.5 IO Total by Interval/Object/IO-Type

.....................5.5.6 IO Total by Interval/Object/IO-Type/Stack-Index

.....................5.5.7 IO Total by Interval/Pool

.....................5.5.8 IO Total by Interval/Pool/IO-Type

.....................5.5.9 IO Total by Interval/Unit

.....................5.5.10 IO Total by Interval/Unit/IO-Type

.....................5.5.11 IO Total by IO-Type

.....................5.5.12 IO Total by IOP

.....................5.5.13 IO Total by IOP/IO-Type

.....................5.5.14 IO Total by Job-Thread

.....................5.5.15 IO Total by Job-Thread/IO-Type

.....................5.5.16 IO Total by Job-Thread/IO-Type/Stack-Index

.....................5.5.17 IO Total by Job-Thread/Object

.....................5.5.18 IO Total by Job-Thread/Object/IO-Type

.....................5.5.19 IO Total by Job-Thread/Object/IO-Type/Stack-Index

.....................5.5.20 IO Total by Job-Thread/Pool

.....................5.5.21 IO Total by Job-Thread/Pool/IO-Type

.....................5.5.22 IO Total by Job-Thread/Program

.....................5.5.23 IO Total by Job-Thread/Program/IO-Type

.....................5.5.24 IO Total by Job-Thread/Program/Object

.....................5.5.25 IO Total by Job-Thread/Program/Object/IO-Type

.....................5.5.26 IO Total by Job-Thread/Program/ObjectSegment-Desc

.....................5.5.27 IO Total by Job-Thread/Program/ObjectSegment-Desc/IO-Type

.....................5.5.28 IO Total by Job-Thread/Unit

....................5.5.29 IO Total by Job-Thread/Unit/IO-Type

....................5.5.30 IO Total by Object

....................5.5.31 IO Total by Object/IO-Type

....................5.5.32 IO Total by Object/IO-Type/Stack-Index

....................5.5.33 IO Total by Object/Job-Thread

....................5.5.34 IO Total by Object/Job-Thread/IO-Type

....................5.5.35 IO Total by Object/Job-Thread/IO-Type/Stack-Index

....................5.5.36 IO Total by ObjectSegment-Desc

....................5.5.37 IO Total by ObjectSegment-Desc/IO-Type

....................5.5.38 IO Total by ObjectSegment-Desc/IO-Type/Stack-Index

....................5.5.39 IO Total by Pool

....................5.5.40 IO Total by Pool/Interval

....................5.5.41 IO Total by Pool/Interval/IO-Type

....................5.5.42 IO Total by Pool/IO-Type

....................5.5.43 IO Total by Pool/Job-Thread

....................5.5.44 IO Total by Pool/Job-Thread/IO-Type

....................5.5.45 IO Total by Pool/Object

....................5.5.46 IO Total by Pool/Object/IO-Type

....................5.5.47 IO Total by Program

....................5.5.48 IO Total by Program/IO-Type

....................5.5.49 IO Total by Program/Object

....................5.5.50 IO Total by Program/Object/IO-Type

....................5.5.51 IO Total by Program/ObjectSegment-Desc

....................5.5.52 IO Total by Program/ObjectSegment-Desc/IO-Type

....................5.5.53 IO Total by Unit

....................5.5.54 IO Total by Unit/IO-Type

....................5.5.55 IO Total by Unit/Job-Thread

....................5.5.56 IO Total by Unit/Job-Thread/IO-Type

....................5.5.57 IO Total by Unit/Object

....................5.5.58 IO Total by Unit/Object/IO-Type

..............5.6 Size change to objects and segments reports

....................5.6.1 Job/Task Consuming most DASD Space per Interval

....................5.6.2 Net DASD Space Change by Causing Job/Task

....................5.6.3 Net DASD Space Change by Object Name and Symbolic Type

....................5.6.4 Net DASD Space Change by Object Type

....................5.6.5 Net DASD Space Change by Time Intervals

....................5.6.6 Object Name and Symbolic Type Consuming most DASD Space per Interval

....................5.6.7 Object Type Consuming most DASD Space per Interval

..............5.7 Database file full opens/closes reports

.....................5.7.1 Local DB File Full Closes by File-Library within Interval

.....................5.7.2 Local DB File Full Closes by File-Library within Job-Thread - No Intervals

.....................5.7.3 Local DB File Full Closes by File-Library within Program - No Intervals

.....................5.7.4 Local DB File Full Closes by Job within Interval

.....................5.7.5 Local DB File Full Closes by Job-Thread within File-Library - No Intervals

.....................5.7.6 Local DB File Full Closes by Job-Thread within Program - No Intervals

.....................5.7.7 Local DB File Full Closes by Program within File-Library - No Intervals

.....................5.7.8 Local DB File Full Closes by Program within Interval

.....................5.7.9 Local DB File Full Closes Program within Job-Thread - No Intervals

.....................5.7.10 Local DB File Full Opens by File-Library within Interval

.....................5.7.11 Local DB File Full Opens by File-Library within Job-Thread - No Intervals

.....................5.7.12 Local DB File Full Opens by File-Library within Program - No Intervals

.....................5.7.13 Local DB File Full Opens by Job within Interval

.....................5.7.14 Local DB File Full Opens by Job-Thread within File-Library - No Intervals

.....................5.7.15 Local DB File Full Opens by Job-Thread within Program - No Intervals

.....................5.7.16 Local DB File Full Opens by Program within File-Library - No Intervals

.....................5.7.17 Local DB File Full Opens by Program within Interval

.....................5.7.18 Local DB File Full Opens by Program within Job-Thread - No Intervals

..............5.8 Task switch data analysis reports

.....................5.8.1 Collection Overview

.....................5.8.2 Collection Summary

.....................5.8.3 TDE Wait Code Summary

.....................5.8.4 TDE Transaction Summary

.....................5.8.5 TDE Transaction Components

.....................5.8.6 50 Longest Conflicts or Blocks

.....................5.8.7 Physical DASD IO Requests

.....................5.8.8 TDE One Second Intervals

.....................5.8.9 TDE Task Switch *Detail* Records

..............5.9 Data area reports

.....................5.9.1 Data area usage by job

.....................5.9.2 Data area usage by name and type

.....................5.9.3 Data area usage by program

.....................5.9.4 Data area usage by time interval

..............5.10 Data queue activity reports

.....................5.10.1 Data queue activity by causing job/thread

.....................5.10.2 Data queue activity by causing program name

.....................5.10.3 Data queue activity by data queue name and type

.....................5.10.4 Data queue activity by time interval

.......Chapter 6 PTDV

..............6.1 Modes of Operation

..............6.2 Navigating through Frames and Panes

.......Chapter 8 Heap Analyzer

..............8.1 Heap Analyzer Basics

..............8.2 Start Heap Watch Wizard

.....................8.2.1 Welcome

.....................8.2.2 'Object table snapshot' Options page

.....................8.2.3 'Object create profile' Options page

.....................8.2.4 Execution limits page

.....................8.2.5 Summary page

..............8.3 Libraries

..............8.4 Heap Watches

.....................8.4.1 Object table snapshot comparison

Previous Next Expand Collapse Search

Table of Contents | Previous | Next

About This Document

This document provides information about iDoctor for iSeries running on OS/400 V5R2.

Special Note for PEX Analyzer: Any PEX data collected prior to OS/400 V4R5M0 cannot be processed using PEX Analyzer.

Additional useful information is available in the iSeries Performance Explorer Tips and Techniques manual, SG24-4781-00. To view an abstract or to order a copy of the manual, refer to the following link below: Performance Explorer Tips and Techniques Redbook

Table of Contents | Previous | Next

Part I iDoctor for iSeries Overview

This part covers basic information about each iDoctor for iSeries component and installation steps.

Table of Contents Previous Next

Chapter 1 General Information

iDoctor for iSeries is a suite of tools consisting of the following components:

- Job Watcher
- PEX Analyzer
- Heap Analysis Tools for Java^TM (also known as Heap Analyzer)
- PTDV (Performance Trace Data Visualizer)

All of these tools assist the user with performance analysis or performance investigation on the iSeries. A brief explanation of each tool follows:

iDoctor for iSeries - Job Watcher
Job Watcher provides the user with the ability to collect many different types of information about sets of jobs (or all jobs) on a system. The type of data gathered includes wait times, CPU, IO activity, call stacks, SQL statements, communications, and activation group statistics. Many graphs at various scopes are available to help illustrate what jobs are performing badly on a system at a high level down to the low level interval by interval view of what an individual job or thread is doing.

This component is available for a trial evaluation or purchase from this Web site.

iDoctor for iSeries - PEX Analyzer
PEX Analyzer assists the user with the analysis of PEX (Performance Explorer) data. PEX is a component of OS/400. The server-side of PEX Analyzer includes a command STRIDOCCOL which simplifies the process of creating a collection by wrapping the OS/400 commands ADDPEXDFN, STRPEX and ENDPEX into one step.
The client side of PEX Analyzer includes many graphing and query capabilities which allow a user to quickly identify performance bottlenecks.

This component is available for a trial evaluation or purchase via this Web site.

iDoctor for iSeries - Heap Analysis Tools for Java
Heap Analyzer provides two different modes of operation which are useful for investigating Java memory related performance problems.
The first mode provides a dump of the Java Virtual Machine for the selected job. This dump includes object size, and class information for all objects within the JVM at the time the dump was collected.
The second mode provides a profile of the objects being created within the Java Virtual Machine. This profile also includes call stacks.

Heap Analyzer is a free component.

iDoctor for iSeries - PTDV (Performance Trace Data Visualizer)
Performance Trace Data Visualizer (PTDV) for iSeries is a tool for processing, analyzing, and viewing PEX

data with an emphasis on the PEX events used in Java programs.

PTDV is a free component.

Table of Contents | Previous | Next

1.1 Job Watcher

Job Watcher is a new iDoctor for iSeries component that returns real-time information about a selected set of jobs, threads and/or LIC tasks. The data is collected by a server job, stored in database files, and displayed on a client via the iDoctor GUI presentation facilities. Job Watcher is similar in sampling function to the system commands WRKACTJOB and WRKSYSACT in that each "refresh" computes delta information for the ending snapshot interval. Refreshes can be set to occur automatically, as frequently as every 100 milliseconds. The data harvested from the jobs/threads/tasks being watched is done so in a non-intrusive manner (similar to WRKSYSACT).

This "watch" data is summarized in many different types of reports and graphs by the client which can provide a quick picture of what is happening per thread when many different threads are being watched. As the situation merits, the user can select a job, an interval, and then drill down for the details while the watch is in progress or after a watch has ended.

Other features include the options to setup the Job Watch so that when predefined conditions are encountered, a user program can be called. The Job Watcher wizard is used to help setup the rule condtion parameters and actions to occur when the rules are met.

The information harvested includes:

- Standard WRKSYSACT type info: CPU, DASD I/O breakdown, DASD space consumption, etc.
- Some data currently only seen in Collection Services: "real" user name, seize time, breakdown of what types of waits (all waits) that occurred.
- Some data not available anywhere else in real time: details on the current wait (duration, object, conflicting job info, specific LIC block point id), 50 deep invocation stack including LIC stack frames, and more.

Job Watcher is available for trial evaluation or purchase via this Web site. A license for Job Watcher includes:

- Job Watcher software (licensed by system serial number via an acces code)
- Electronic defect support for Job Watcher software for the term of the contract
- No charge updates to Job Watcher software for the term of the contract

Table of Contents Previous Next

1.2 PEX Analyzer

The iDoctor for iSeries PEX Analyzer Service includes a software tool specifically geared towards pinpointing issues affecting system and application performance. The detailed analysis it provides picks up where the PM/400 and Performance Tools products leave off and supplies a drill down capability that provides low-level summary of disk operations, CPU utilization, file opens, MI programs, wait states, DASD space consumption and much more. The client component allows a user to condense and graph iSeries PEX trace, statistical and profile data.

PEX Analyzer Service - "The System Performance X-Ray"

- Includes PEX Analyzer software (licensed by CPU/Serial # via licensed key)
- Electronic defect support for PEX Analyzer software for the term of the contract
- Installation assistance for PEX Analyzer software
- No charge updates to PEX Analyzer software for the term of the contract

Table of Contents Previous Next

1.3 Heap Analysis Tools for Java

Heap Analysis Tools for Java (also known as Heap Analyzer) is used to perform Java application heap analysis and object create profiling (size and identification) over time. Heap Analyzer includes information about:

- JVM heap growth/size
- The objects being created (type of object, count and object size, object heap size)
- The application "Heap Footprint" for memory sizing and performance considerations
- Includes a call stack for every snapshot when running in profile mode so objects created can be correlated to functions in the application.

Heap Analyzer requires an iSeries running OS/400 V5R2 with the required Java Group PTF level. The group PTF level information is listed on the download page.

Table of Contents Previous Next

1.4 PTDV

PTDV is a tool for processing, analyzing, and viewing Performance Explorer Collection data residing in PEX database files. It runs with Java, and uses a different GUI than the other iDoctor components.

PTDV provides table and tree views of Performance Explorer Collection data. It summarizes data on various levels, providing many views and various levels of detail, depending on your path through the data. Tables consist of columns that are sortable and rearranageble; trees provide a "drill down" capability to show parent/child relationship for data where appropriate. Each pane of data can be exported to another format for viewing by other tools, such as spreadsheets. More detail can be displayed as the user determines performance problem areas.

PTDV can process collections containing program and/or java events. Collections containing program events will be processed so that procedure information can be determined, such as inline and cumulative time and cycles. The call flow is reconstructed and displayed in a tree format. If java events are also present, information from those events are summarized and displayed, and the method context information is available. For example, if object create or object lock events are present then a table displaying the class name, number of objects, total sizes, wait times, are all displayed.

Disclaimer for PTDV

This material is provided for your consideration; it has not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of this material. IBM provides no program services for this material.

THIS MATERIAL IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO THE ABOVE EXCLUSION MAY NOT APPLY TO YOU. IN NO EVENT WILL IBM BE LIABLE TO ANY PARTY FOR ANY DIRECT, INDIRECT, SPECIAL OR OTHER CONSEQUENTIAL DAMAGES FOR ANY USE OF THIS MATERIAL INCLUDING, WITHOUT LIMITATION, ANY LOST PROFITS, BUSINESS INTERRUPTION, LOSS OF PROGRAMS OR OTHER DATA ON YOUR INFORMATION HANDLING SYSTEM OR OTHERWISE, EVEN IF WE ARE EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Table of Contents   Previous   Next

Chapter 2 Installation

This chapter includes information about the following:

- iSeries system installation requirements
- PC installation requirements
- PTF prerequisites
- Installing and uninstalling iDoctor for iSeries

Table of Contents Previous Next

2.1 Installing Job Watcher

PC Requirements:

- Windows XP, NT 4.0, 2000
- IBM iSeries Access for Windows (V5R2 or higher)
- 256 MB or higher

Server Requirements:

- OS/400 V5R2
- PTFs MF30028, MF30445, SI08420
- The user profile performing the installation must have *SECOFR user class and special authorities *ALLOBJ and *SECADM.
- The following host servers (identified by the SERVER parameter values on the STRHOSTSVR command) need to be running on the server: *DATABASE, *RMTCMD, *SIGNON, *SRVMAP
- System value QALWOBJRST must be *ALL or (*ALWSYSSTT and *ALWPGMADP)

After installation you will have the following new libraries on your server: QYPBASE and QPYRTJW

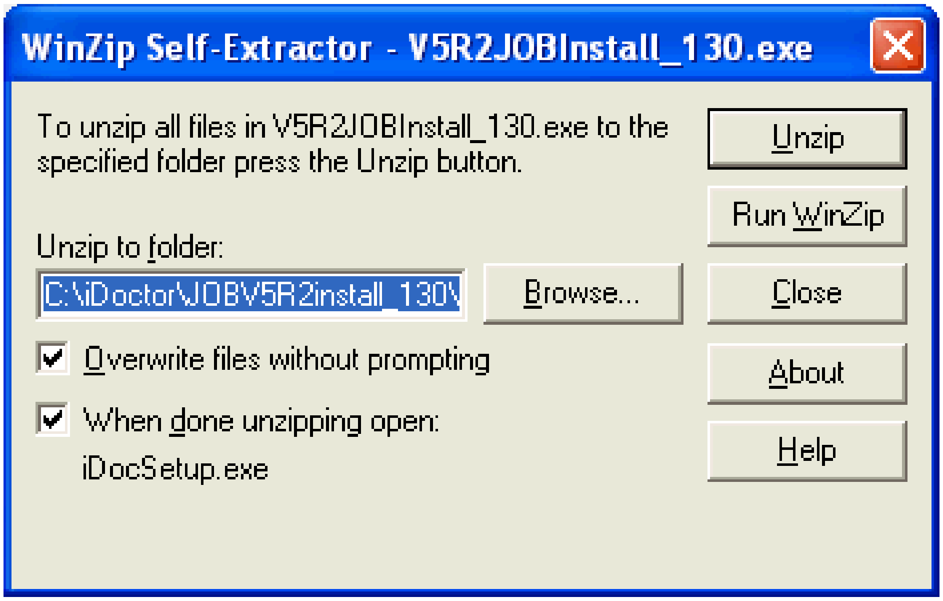
The Install Process:

The GUI install process will install the server objects and programs on your iSeries and the client code on your PC. The PTFs required for Job Watcher must be manually installed via whichever method normally used to get PTFs for your iSeries. The PTFs required are MF30028, MF30445, SI08420. If unsure of which method to use for installing the PTFs, use the Fix Central Web site. Instructions for installing these PTFs can be found in the next section.

After the PTFs are loaded and applied on your iSeries, perform the following steps:

Step 1 Download the install image for Job Watcher from our Web site to your PC.

Step 2 Double-click on the install image (it is a self-extracting .exe.) from within Windows Explorer. You will see the following:



WinZip Self-Extractor - V5R2JOBInstall_130.exe

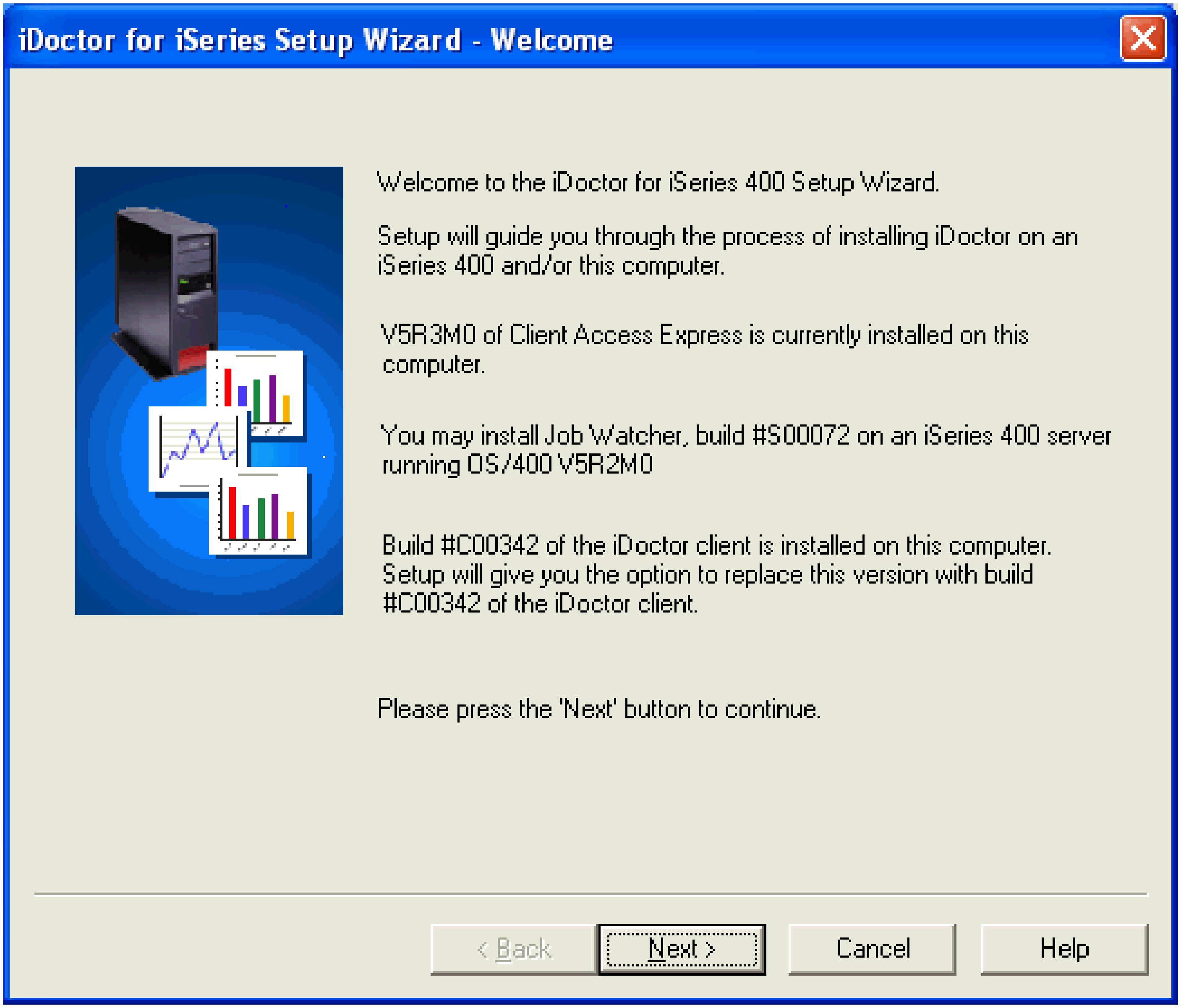
To unzip all files in V5R2JOBInstall_130.exe to the specified folder press the Unzip button.

Unzip to folder: C:\iDoctor\JOBV5R2install_130\

☑ Overwrite files without prompting

☑ When done unzipping open: iDocSetup.exe

Step 3 Change the path where the install image will be extracted to on the PC if desired and click the 'Unzip' button. Wait a moment for the files to be extracted and the setup program to be launched.



iDoctor for iSeries Setup Wizard - Welcome

Welcome to the iDoctor for iSeries 400 Setup Wizard.

Setup will guide you through the process of installing iDoctor on an iSeries 400 and/or this computer.

V5R3M0 of Client Access Express is currently installed on this computer.

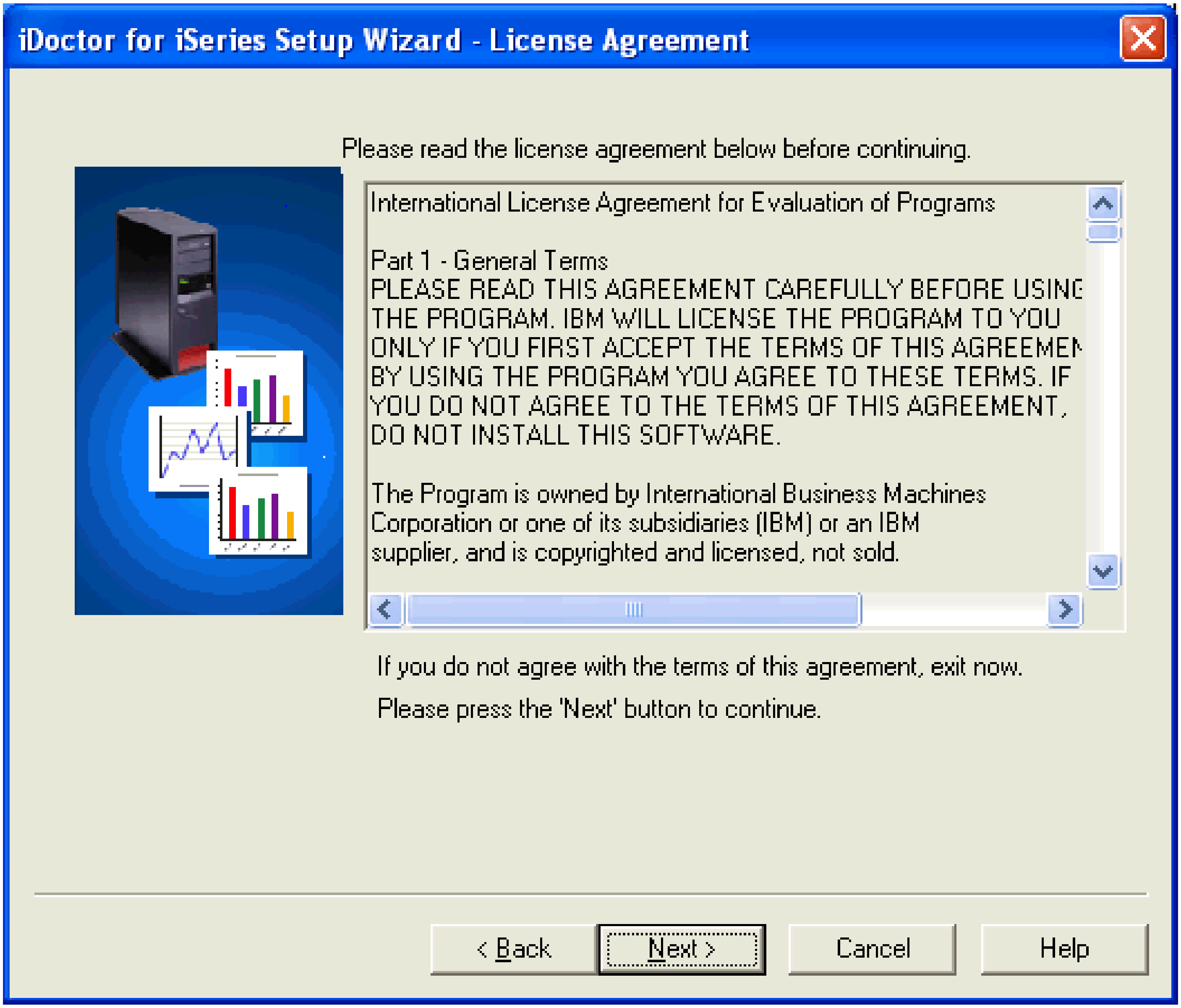
You may install Job Watcher, build #S00072 on an iSeries 400 server running OS/400 V5R2M0

Build #C00342 of the iDoctor client is installed on this computer. Setup will give you the option to replace this version with build #C00342 of the iDoctor client.

Please press the 'Next' button to continue.

This screen identifies the version of iSeries Access installed as well as the version of iDoctor for iSeries client installed (if found). Click 'Next'.

Step 4 On the following screen, click the 'Next' button to indicate acceptance of the license agreement.



iDoctor for iSeries Setup Wizard - License Agreement

Please read the license agreement below before continuing.

International License Agreement for Evaluation of Programs

Part 1 - General Terms

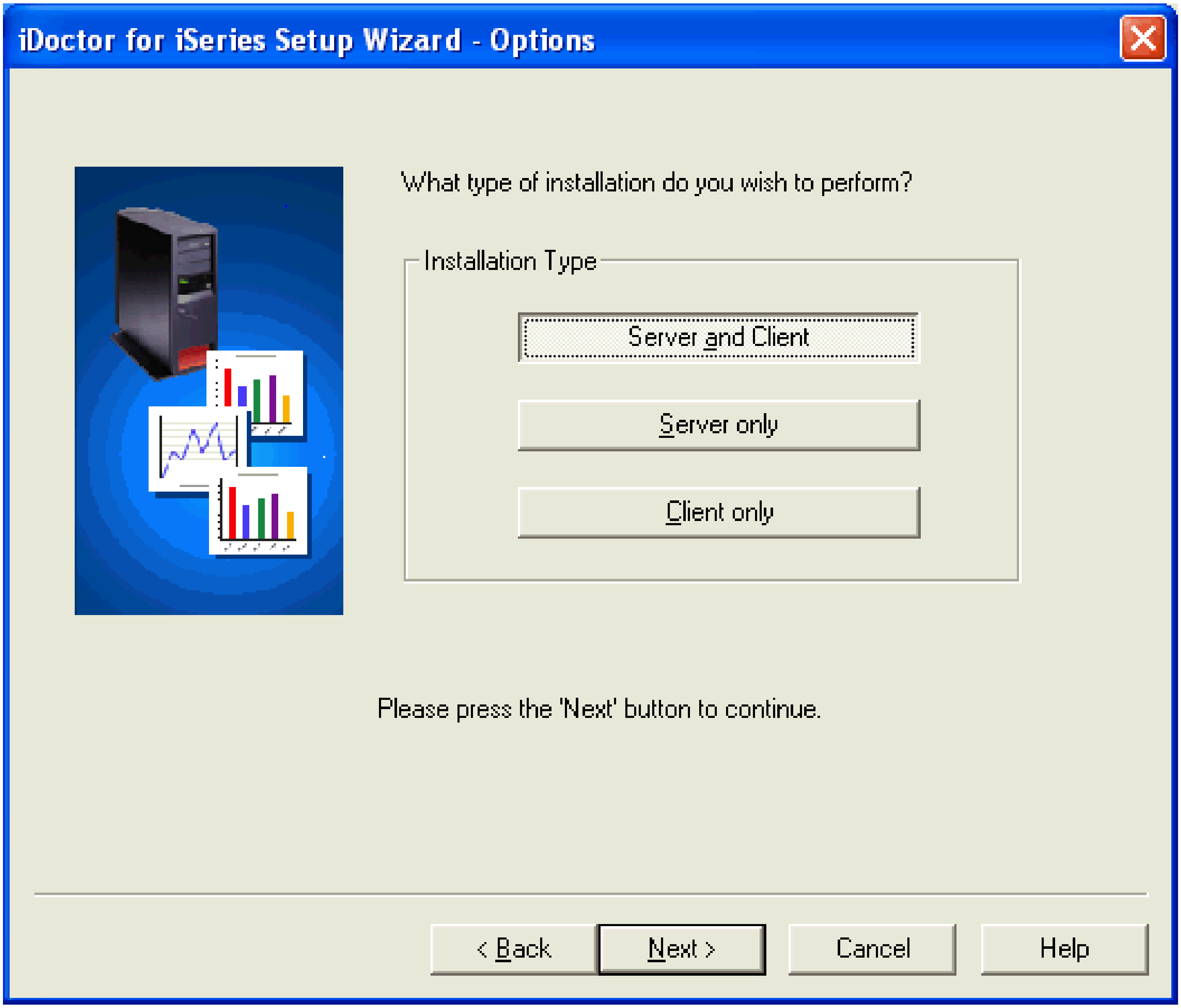
PLEASE READ THIS AGREEMENT CAREFULLY BEFORE USING THE PROGRAM. IBM WILL LICENSE THE PROGRAM TO YOU ONLY IF YOU FIRST ACCEPT THE TERMS OF THIS AGREEMEN BY USING THE PROGRAM YOU AGREE TO THESE TERMS. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, DO NOT INSTALL THIS SOFTWARE.

The Program is owned by International Business Machines Corporation or one of its subsidiaries (IBM) or an IBM supplier, and is copyrighted and licensed, not sold.

If you do not agree with the terms of this agreement, exit now.

Please press the 'Next' button to continue.

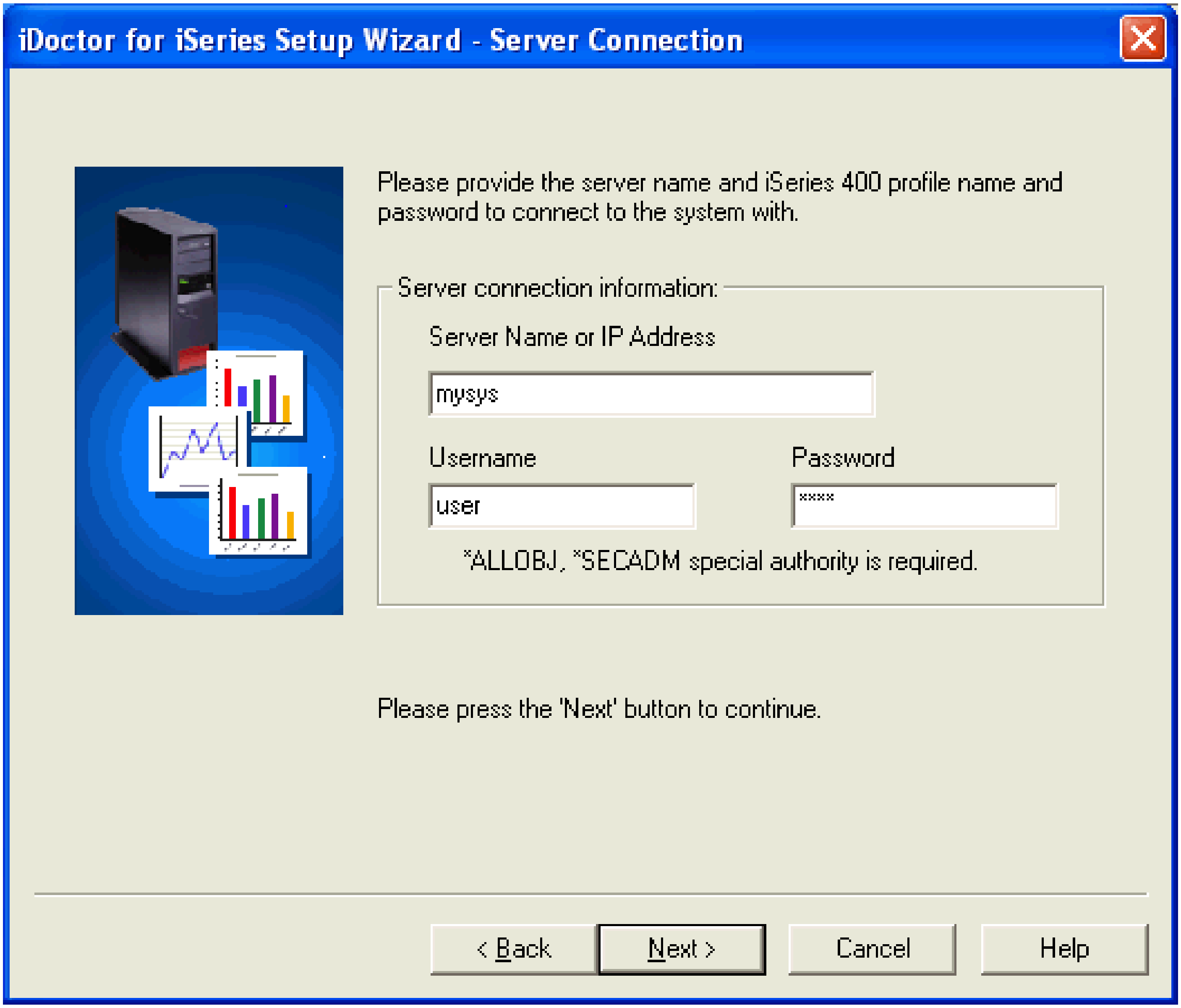
Step 5 Select the type of installation to perform. This screen allows you to choose whether to install the server side of Job Watcher, the client side of iDoctor or both.



What type of installation do you wish to perform?

Please press the 'Next' button to continue.

Step 6 If you are installing the server portion of Job Watcher you will see a screen asking for the connection information to use to access the server. The user profile must have the user class authority of *SECOFR and *ALLOBJ, *SECADM special authorities.



Please provide the server name and iSeries 400 profile name and password to connect to the system with.

Server connection information:

Server Name or IP Address: mysys

Username: user

Password: ****

*ALLOBJ, *SECADM special authority is required.

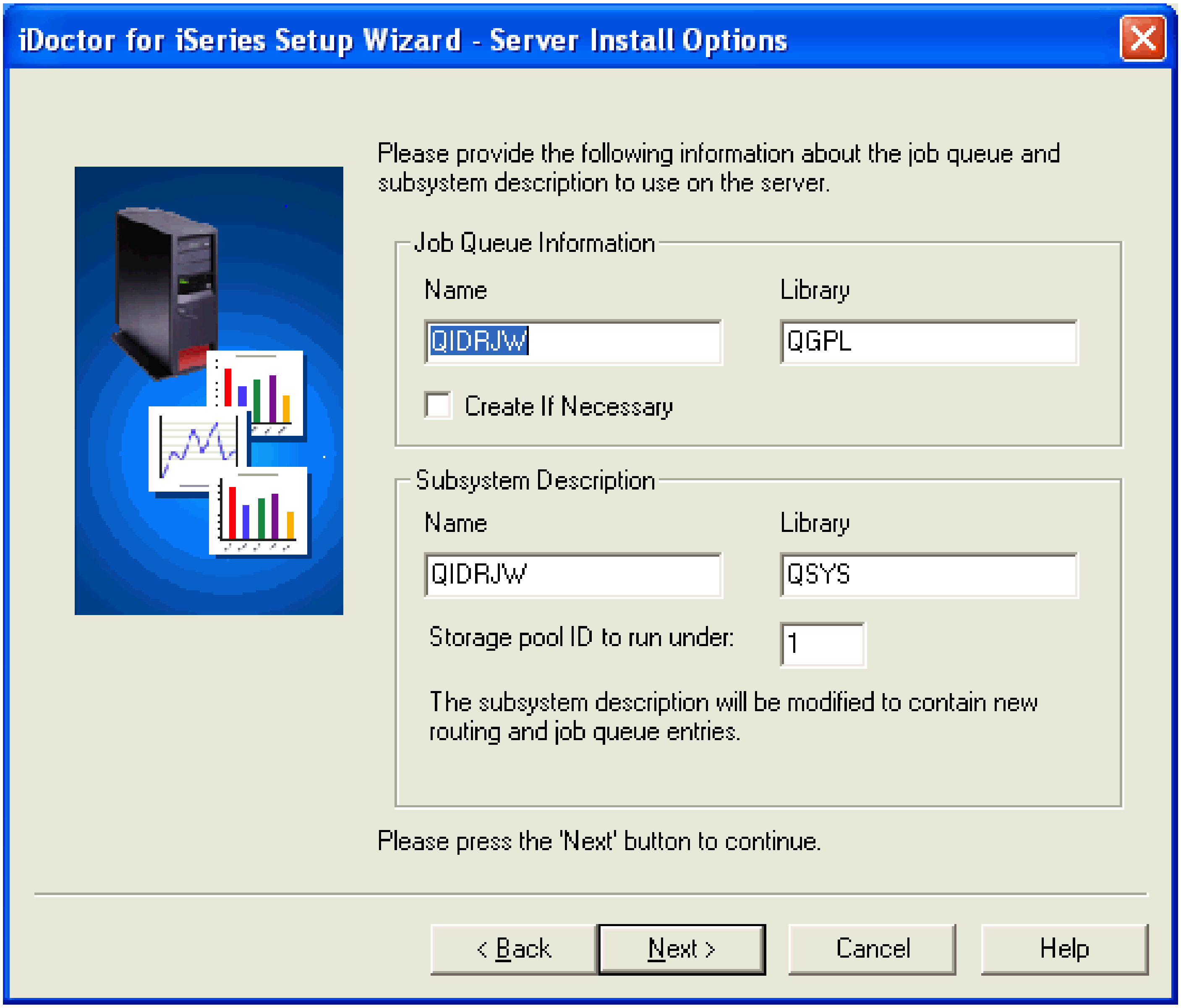
Please press the 'Next' button to continue.

Click the 'Next' button to connect to the server specified and continue on the next page.

Step 7 If you are installing the server portion of Job Watcher you will also be asked to specify a job queue name and a subsystem description the server code may use when running Job Watcher. If the subsystem does not already exist, the install program will ask if it should be created. This subsystem and job queue will be used to run the Job Watcher "watch" jobs named QPYJWJOB for job-specific Job Watches and QPYSWJOB for system-wide Job Watches. Job Watcher does not have a limit to the number of "watches" that can be active at one time.

You must also indicate the storage pool ID that the Job Watcher jobs should run under. This parameter is required in order to add routing entries to the subsystem description.

Press the 'Next' button to continue to the summary page.



iDoctor for iSeries Setup Wizard - Server Install Options

Please provide the following information about the job queue and subsystem description to use on the server.

Job Queue Information

Name: QIDRJW

Library: QGPL

□ Create If Necessary

Subsystem Description

Name: QIDRJW

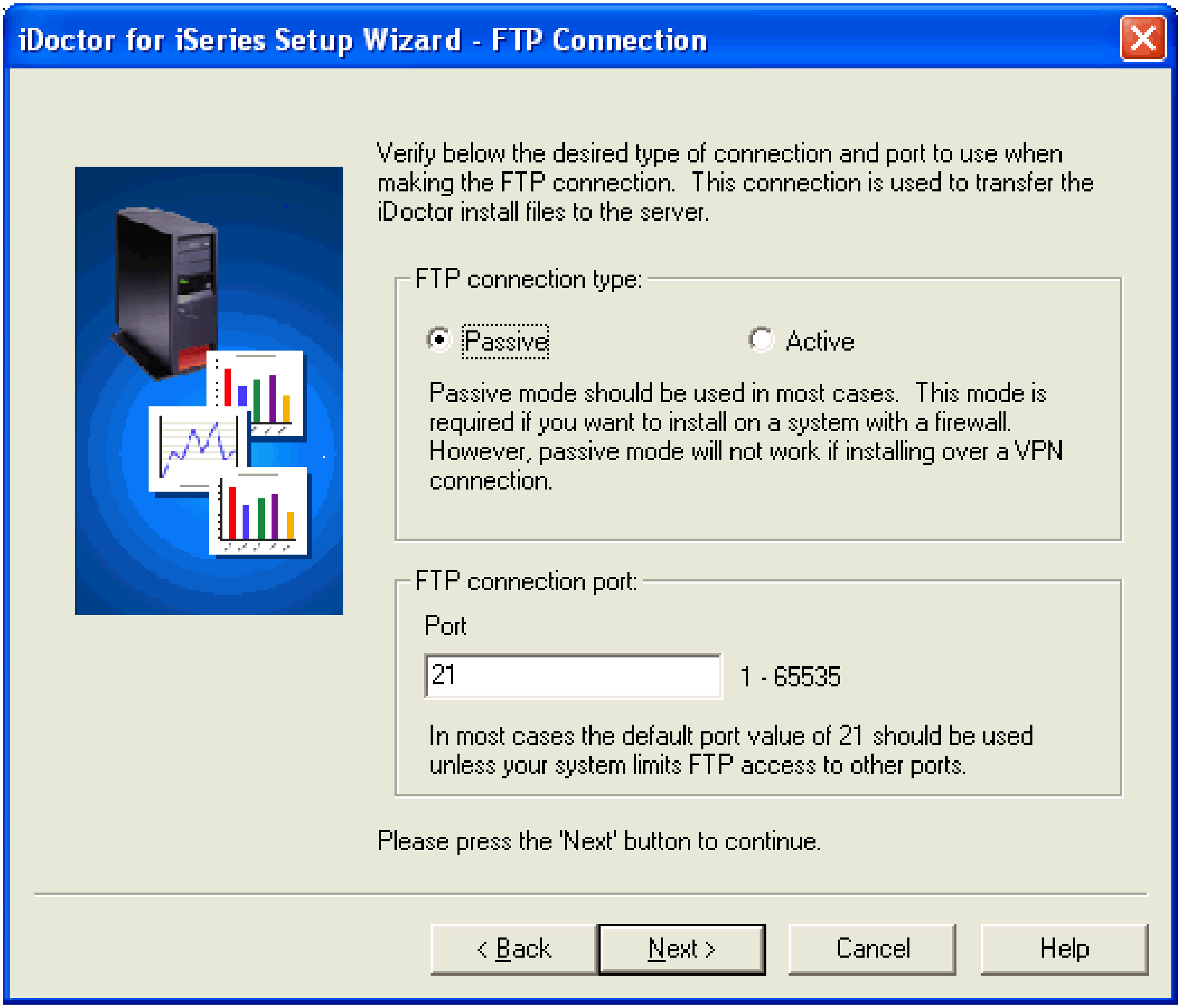
Library: QSYS

Storage pool ID to run under: 1

The subsystem description will be modified to contain new routing and job queue entries.

Please press the 'Next' button to continue.

Step 8 The next page gives the user the option to specify which type of FTP connection should be used when performing the install. Only in unusual circumstances should anything other than the defaults be used on this page. However, if installing over a VPN connection and "Passive" FTP does not work, try using "Active" FTP instead.



iDoctor for iSeries Setup Wizard - FTP Connection

Verify below the desired type of connection and port to use when making the FTP connection. This connection is used to transfer the iDoctor install files to the server.

FTP connection type:

● Passive ○ Active

Passive mode should be used in most cases. This mode is required if you want to install on a system with a firewall. However, passive mode will not work if installing over a VPN connection.

FTP connection port:

Port

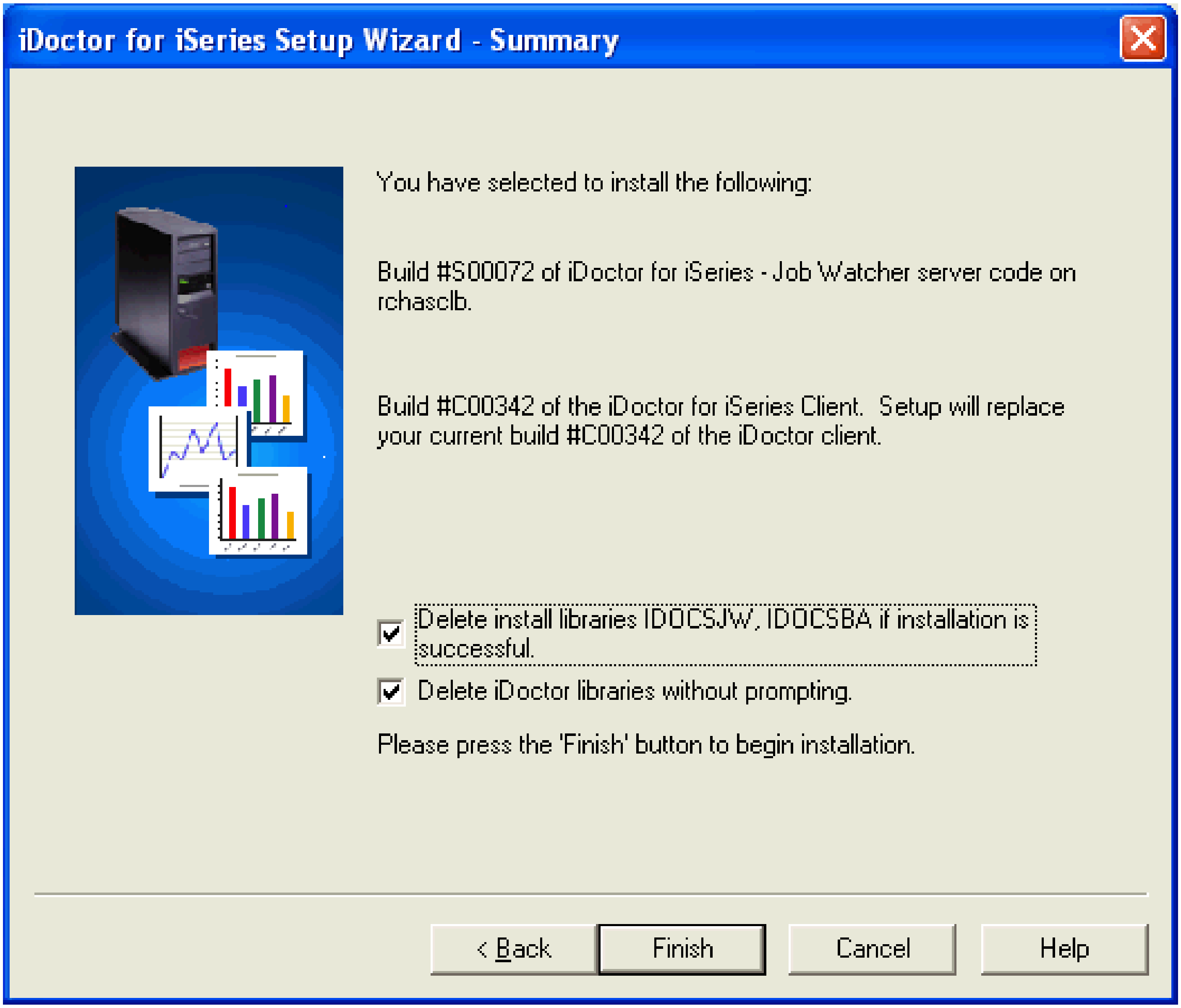
21 1 - 65535

In most cases the default port value of 21 should be used unless your system limits FTP access to other ports.

Please press the 'Next' button to continue.

Step 9 If an access code is required, your will be asked to enter it on the next screen. Enter the access code (optional) and continue to the 'Next' screen.

Step 10 A summary of your selections appears on the final screen. This page also contains options for whether the temporary libraries used during the installation should be deleted. During the install process these libraries will contain the restored contents of the save files sent to the server. Only if you were having problems with the install would the option to keep these libraries around be useful so they could be used for service.



You have selected to install the following:

Build #S00072 of iDoctor for iSeries - Job Watcher server code on rchasclb.

Build #C00342 of the iDoctor for iSeries Client. Setup will replace your current build #C00342 of the iDoctor client.

☑ Delete install libraries IDOCSJW, IDOCSBA if installation is successful.

☑ Delete iDoctor libraries without prompting.

Please press the 'Finish' button to begin installation.

Step 11 Clicking the 'Finish' button will copy all of the files and run the commands necessary to install the server and/or client portion of Job Watcher. The server portion of the installation may take a few minutes.

After the install completes the setup log file will be shown. If any errors occur, send this file to idoctor@us.ibm.com for assistance.

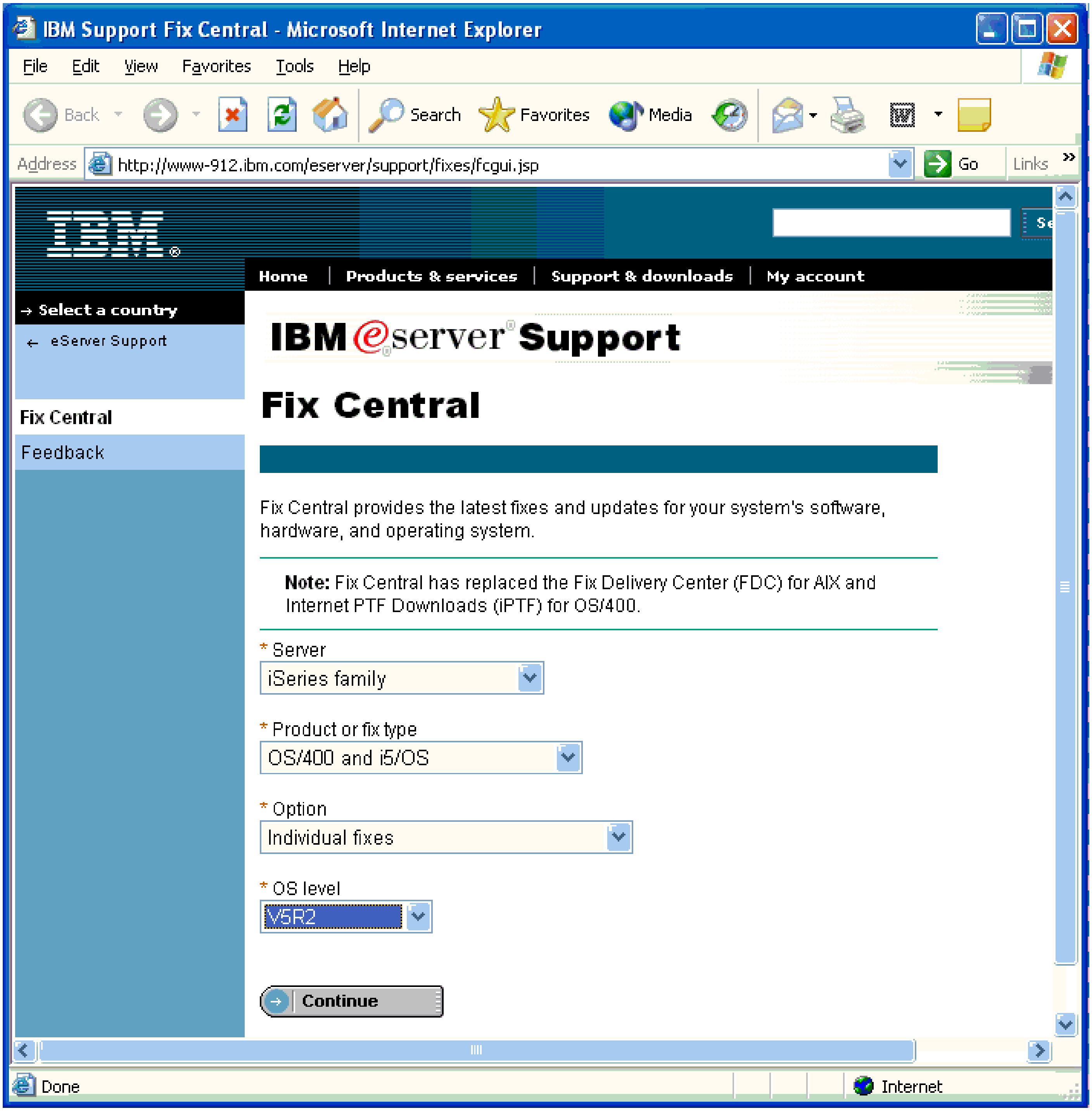
Table of Contents | Previous | Next

2.1.1 PTF installation

The GUI install process will install the server objects and programs on your iSeries and the client code on your PC. However, this does not include the process of installing the required PTFs for Job Watcher. The PTFs required for Job Watcher must be manually installed via whichever method normally used to get PTFs for your iSeries. The PTFs required are MF30028, MF30445, SI08420.

The following information outlines the steps needed to download and install the PTFs onto your iSeries using the Fix Central Web site:

Step 1 Visit the Fix Central Web site and select the choices shown in the screen capture below. Click the 'Continue' button.



IBM Support Fix Central - Microsoft Internet Explorer

Address http://www-912.ibm.com/eserver/support/fixes/fcgui.jsp

IBM eserver Support

Fix Central

Fix Central provides the latest fixes and updates for your system's software, hardware, and operating system.

Note: Fix Central has replaced the Fix Delivery Center (FDC) for AIX and Internet PTF Downloads (iPTF) for OS/400.

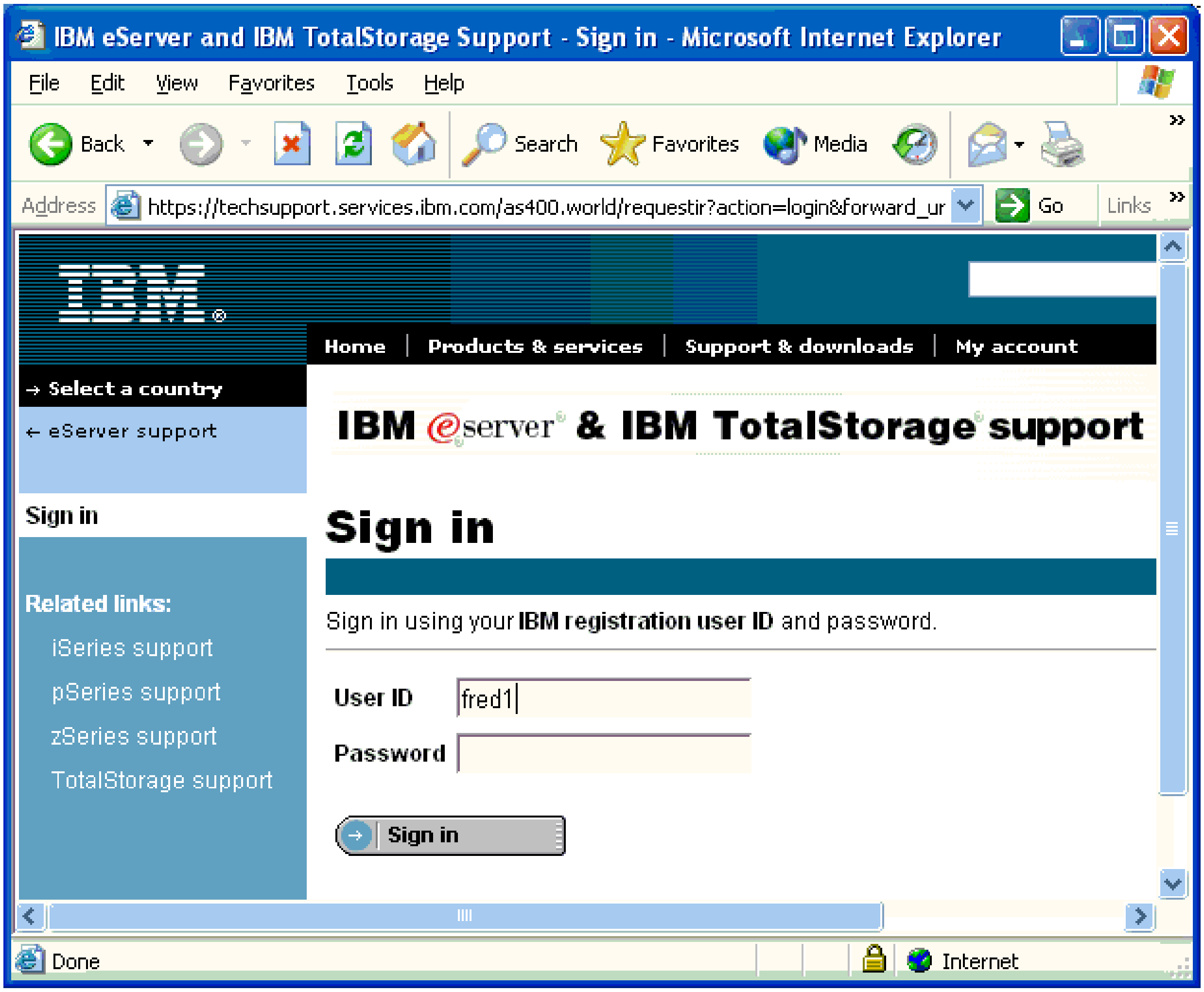
* Server: iSeries family

* Product or fix type: OS/400 and i5/OS

* Option: Individual fixes

* OS level: V5R2

Step 2 Signon using your IBM registration user id and password. If you don't have a user id, click the "Need a user ID?" link in the right navigation bar.

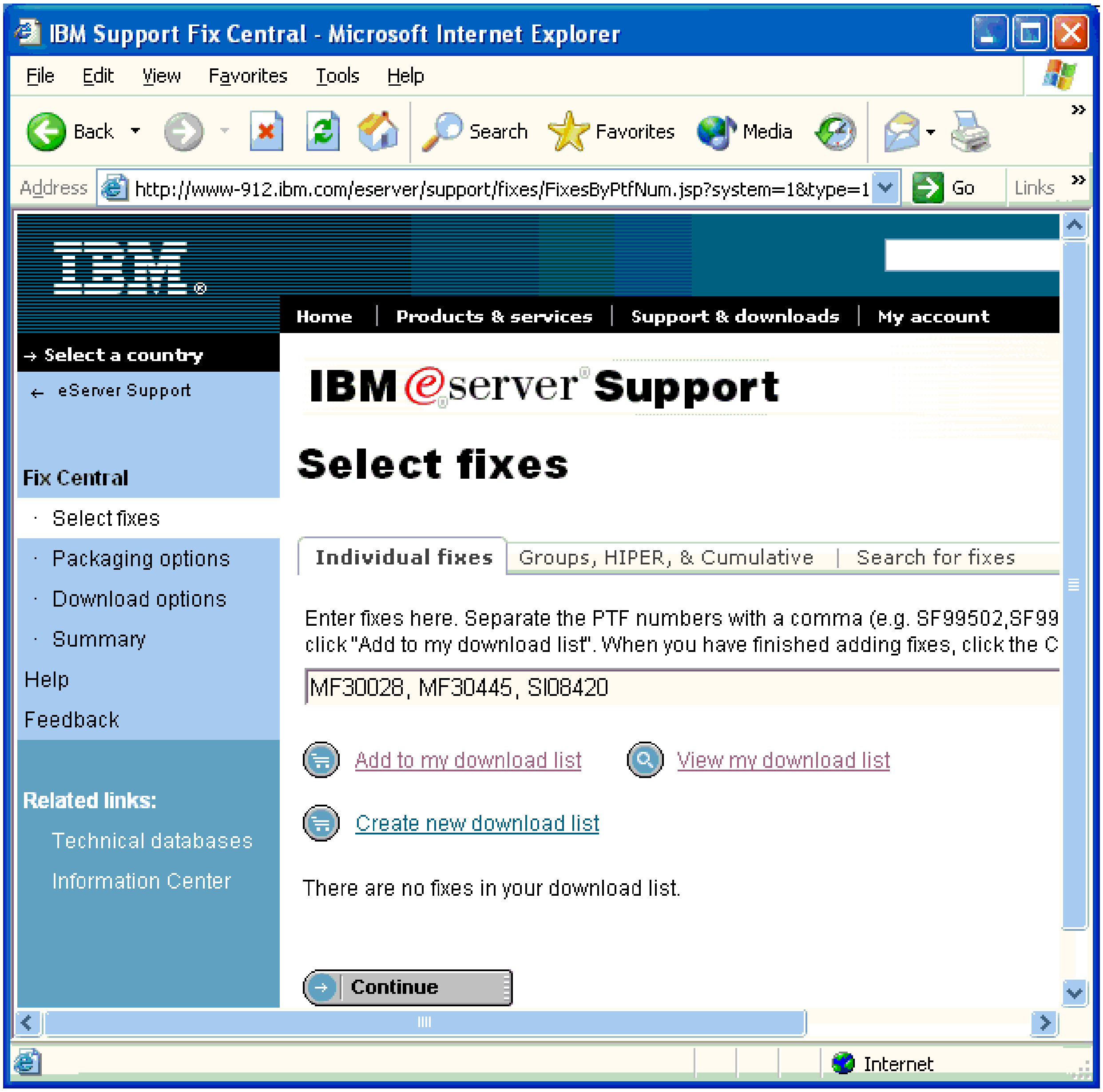

IBM eServer and IBM TotalStorage Support - Sign in - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address https://techsupport.services.ibm.com/as400.world/requestir?action=login&forward_ur
Go Links
IBM
Home | Products & services | Support & downloads | My account
→ Select a country
← eServer support
IBM @server & IBM TotalStorage support
Sign in
Related links:
iSeries support
pSeries support
zSeries support
TotalStorage support
Sign in
Done
Internet

Sign in using your **IBM registration user ID** and password.

User ID: fred1

Password:

Step 3 On the next screen enter the names of the PTFs for V5R2 Job Watcher. The PTF names are: MF30028, MF30445, SI08420



Address: http://www-912.ibm.com/eserver/support/fixes/FixesByPtfNum.jsp?system=1&type=1

Select fixes

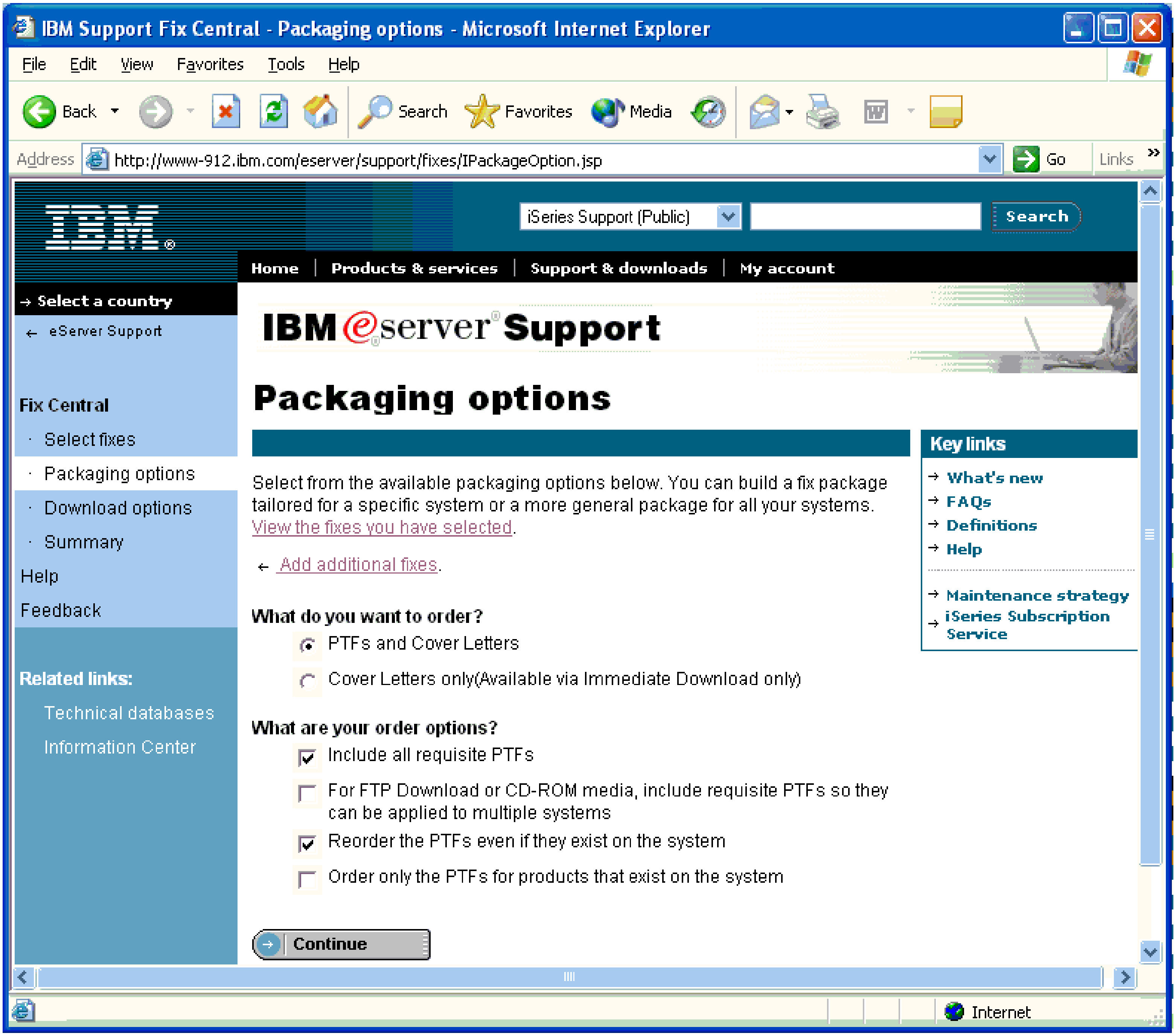
Enter fixes here. Separate the PTF numbers with a comma (e.g. SF99502,SF99
click "Add to my download list". When you have finished adding fixes, click the C

MF30028, MF30445, SI08420

There are no fixes in your download list.

Step 4 Click the 'Continue' button.

Step 5 On the Packaging options page take the defaults and click the 'Continue' button.



IBM Support Fix Central - Packaging options - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://www-912.ibm.com/eserver/support/fixes/IPackageOption.jsp
Go Links
iSeries Support (Public)
Search
Home | Products & services | Support & downloads | My account
Select a country
eServer Support
Fix Central
Select fixes
Packaging options
Download options
Summary
Help
Feedback
Related links:
Technical databases
Information Center
IBM eserver Support
Key links
What's new
FAQs
Definitions
Help
Maintenance strategy
iSeries Subscription Service
Continue
Internet

Packaging options

Select from the available packaging options below. You can build a fix package tailored for a specific system or a more general package for all your systems. View the fixes you have selected.

← Add additional fixes.

What do you want to order?

● PTFs and Cover Letters

○ Cover Letters only(Available via Immediate Download only)

What are your order options?

☑ Include all requisite PTFs

□ For FTP Download or CD-ROM media, include requisite PTFs so they can be applied to multiple systems

☑ Reorder the PTFs even if they exist on the system

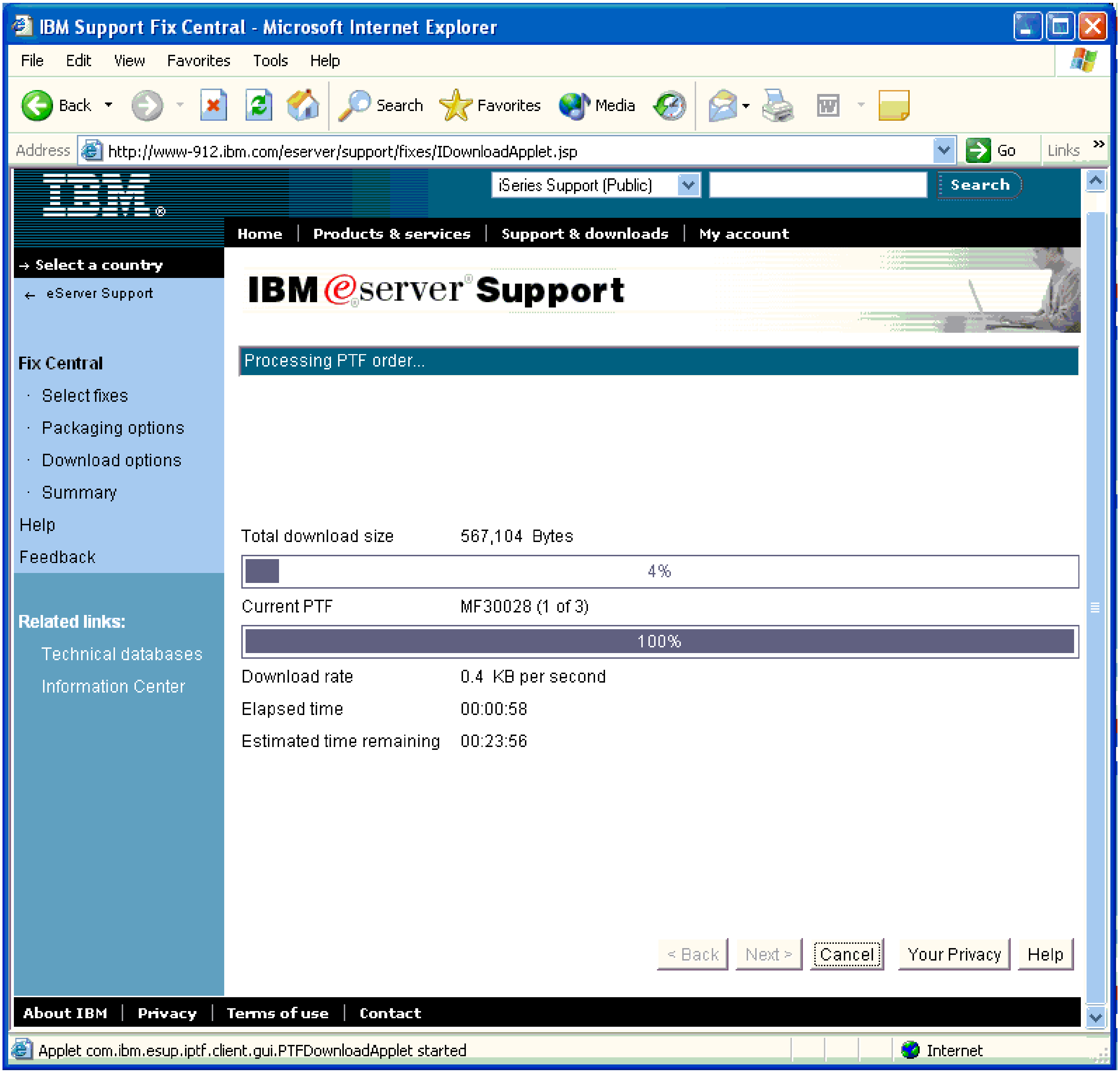
□ Order only the PTFs for products that exist on the system

Step 6 On the Download options page scroll to the last question on the page and answer the question about the location of your iSeries and click the 'Continue' button.

Step 7 On the next page you will be prompted for the system name and user id and password to use to connect to your iSeries so the PTFs can be uploaded to your system. Scroll down and click the 'Next' button on this page after providing this information (if it is not visible).

Step 8 On the next screen, fill in the customer information and click the 'Next' button.

Step 9 While the PTFs are being sent to your system your screen should look like this.



http://www-912.ibm.com/eserver/support/fixes/IDownloadApplet.jsp

Processing PTF order...

Total download size 567,104 Bytes

4%

Current PTF MF30028 (1 of 3)

100%

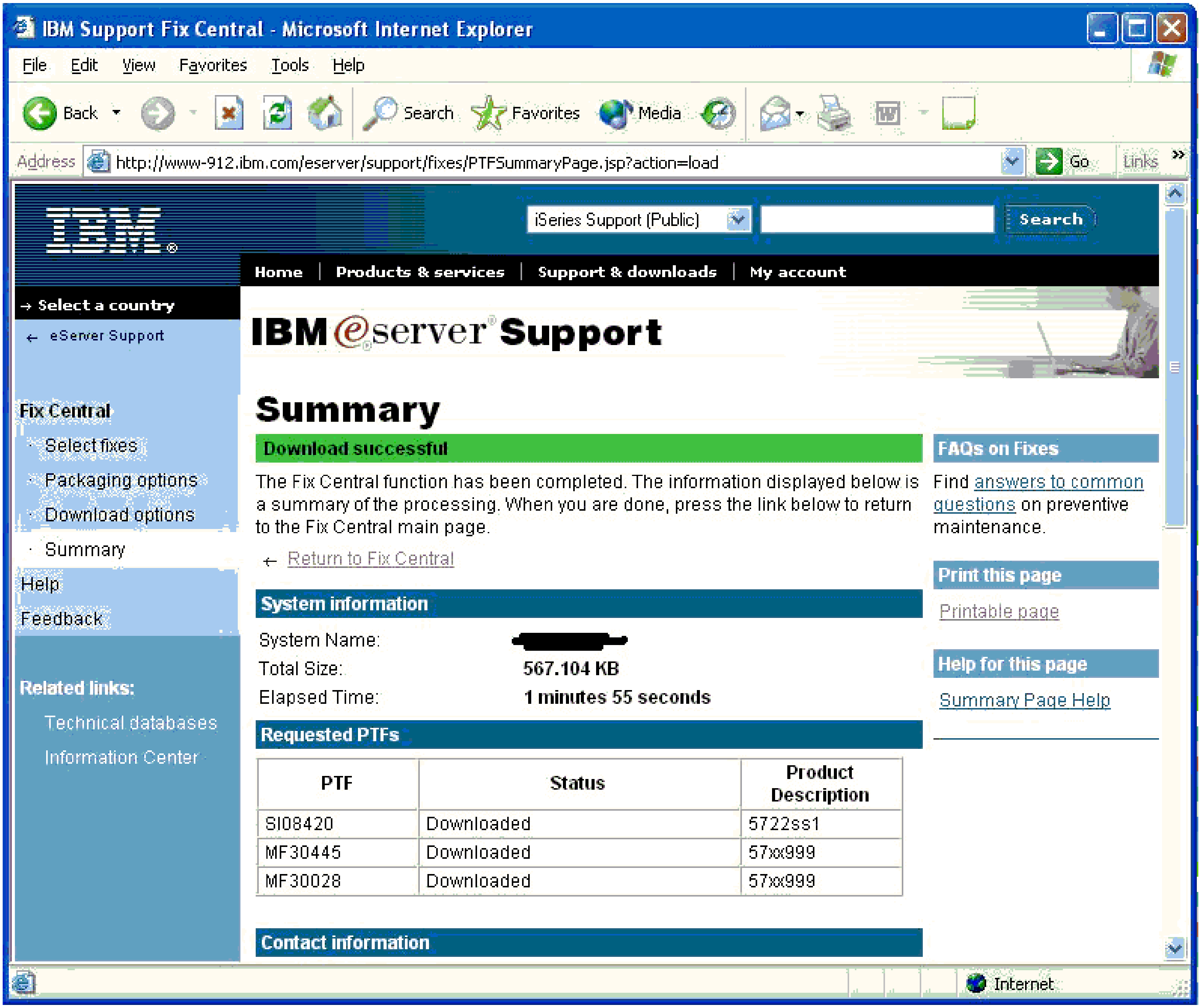
Download rate 0.4 KB per second

Elapsed time 00:00:58

Estimated time remaining 00:23:56

Applet com.ibm.esup.iptf.client.gui.PTFDownloadApplet started

Step 10 If the download is successful your window should look something like this.



Summary

Download successful

The Fix Central function has been completed. The information displayed below is a summary of the processing. When you are done, press the link below to return to the Fix Central main page.

← Return to Fix Central

Find answers to common questions on preventive maintenance.

System information

System Name: [redacted]
Total Size: 567.104 KB
Elapsed Time: 1 minutes 55 seconds

Requested PTFs

PTF	Status	Product Description
SI08420	Downloaded	5722ss1
MF30445	Downloaded	57xx999
MF30028	Downloaded	57xx999

Contact information

Step 11 At this point the PTFs have been sent to your iSeries but are not yet useable. You can check the status of the PTFs that were just sent to your iSeries using the following commands:

```
WRKPTF PTF(MF30445) LICPGM(5722999)
WRKPTF PTF(MF30028) LICPGM(5722999)
WRKPTF PTF(SI08420) LICPGM(5722SS1)
```

If the PTFs are not loaded and applied continue with step 12.

Step 12 To make the PTFs useable by Job Watcher, they must be loaded and applied on your system. Issue the following commands via a green screen session in order to do this:

```
LODPTF LICPGM(5722999) DEV(*SAVF) SELECT(MF30445) SAVF(*LIBL/QMF30445)
APYPTF LICPGM(5722999) RLS(V5R2M0) SELECT(MF30445)

LODPTF LICPGM(5722999) DEV(*SAVF) SELECT(MF30028) SAVF(*LIBL/QMF30028)
APYPTF LICPGM(5722999) RLS(V5R2M0) SELECT(MF30028)

LODPTF LICPGM(5722SS1) DEV(*SAVF) SELECT(SI08420) SAVF(*LIBL/QSI08420)
APYPTF LICPGM(5722SS1) RLS(V5R2M0) SELECT(SI08420)
```

To verify that the PTFs are successfulliy loaded and applied , reissue the commands listed in step 11.

Table of Contents | Previous | Next

2.1.2 Manual installation steps for remote service

Purpose:

This document describes the steps to manually install Job Watcher on an iSeries running OS/400 V5R2. Using the GUI install wizard is preferred in most instances but if this is not possible, these steps are the only other way to get Job Watcher on the desired system. This section only covers the installation of the Job Watcher libraries/objects on the server and does not cover the PTF installation. The required PTFs for Job Watcher at V5R2 must also be loaded and applied.

Self-Installation Prep:

This document assumes that the user has downloaded the Job Watcher install image from the V5R2 downloads page. This self extracting ZIP file is named V5R2JobInstall_130.exe and contains the following files which will be used in this document:

baseV5R2.savf (save file for IDOCSBA temporary install library)
jobV5R2.savf (save file for IDOCSJW temporary install library)

These files should be extracted to a directory on the PC so they may be sent to the iSeries via FTP.

The user profile performing the installation must have *SECADM, *ALLOBJ special authorities. Otherwise all objects may not be restored properly

The system value QALWOBJRST must be set to *ALL or certain objects will not be installed properly. Temporarily change this system value if necessary in order to perform this install. Remember to change it back to what it was when you started after the manual install has completed.

Installation Steps:

Perform the following steps in order to manually install Job Watcher.

1. From a green screen window, run the following commands:
- DLTLIB IDOCINST
- DLTLIB QYPBASE
- DLTLIB QPYRTJW
- DLTLIB IDOCSBA
- DLTLIB IDOCSJW

- CRTLIB IDOCINST

- CRTSAVF FILE(IDOCINST/IDOCINST)

2. Open a DOS FTP session to the iSeries.

3. FTP file baseV5R2.savf from the PC to the iSeries save file IDOCINST in library IDOCINST.

4. From a green screen window, run the following commands:
- DSPSAVF FILE(IDOCINST/IDOCINST)
- verify that the save file contains data
- RSTLIB IDOCSBA *SAVF SAVF(IDOCINST/IDOCINST) MBROPT(*ALL) ALWOBJDIF(*ALL)
- IDOCSBA/INSTIDOCBA
- verify that the message "CPF9898 - SUCCESSFUL iDoctor server Base install." is returned by the command.
- this command will install the QYPBASE library which is needed for the GUI to work
- CLRSAVF FILE(IDOCINST/IDOCINST)

5. FTP file jobV5R2.savf from the PC to the iSeries save file IDOCINST in library IDOCINST.

6. From a green screen window, run the following commands:
- DSPSAVF FILE(IDOCINST/IDOCINST)
- verify that the save file contains data
- RSTLIB IDOCSJW *SAVF SAVF(IDOCINST/IDOCINST) MBROPT(*ALL) ALWOBJDIF(*ALL)
- IDOCSJW/INSTIDOCJW JOBQ(QGPL/QIDOCJW) CRTJOBQ(*NO) SBSD(QSYS/QIDOCJW) POOLID(1)
- the jobq and sbsd parms are the job queue and subsystem that job watches should run under when submitted from the client
- create a job queue and subsystem description if desired or use an existing job queue and subsystem
- verify that the message "CPF9898 - SUCCESSFUL iDoctor server Job Watcher install." is returned by the command.
- this command will install Job Watcher into library QPYRTJW

7. From a green screen window, run the following commands:
- DLTLIB IDOCINST
- DLTLIB IDOCSBA
- DLTLIB IDOCSJW

8. The required PTFs listed on our V5R2 downloads page must be downloaded, loaded and applied on the system before Job Watcher will run correctly.

9. The command QPYRTJW/JWAFTERIPL should be ran once via the green screen before running the WCHJOB or WCHSYS command. Otherwise the wait buckets used in Job Watcher will not be set properly and the data collected will be unreliable.

If you are unable to complete the manual installation successfully e-mail idoctor@us.ibm.com for assistance.

Table of Contents | Previous | Next

2.2 Installing PEX Analyzer

PC Requirements:

- Windows XP, NT 4.0, 2000
- IBM iSeries Access for Windows (V5R2 or higher)
- 256 MB or higher

Server Requirements:

- Performance Tools LPP (PT1)
- OS/400 V5R2
- The user profile performing the installation must have *SECOFR user class and special authorities *ALLOBJ and *SECADM.
- The following host servers (identified by the SERVER parameter values on the STRHOSTSVR command) need to be running on the server: *DATABASE, *DTAQ, *RMTCMD, *SIGNON, *SRVMAP
- System value QALWOBJRST must be *ALL or (*ALWSYSSTT and *ALWPGMADP)

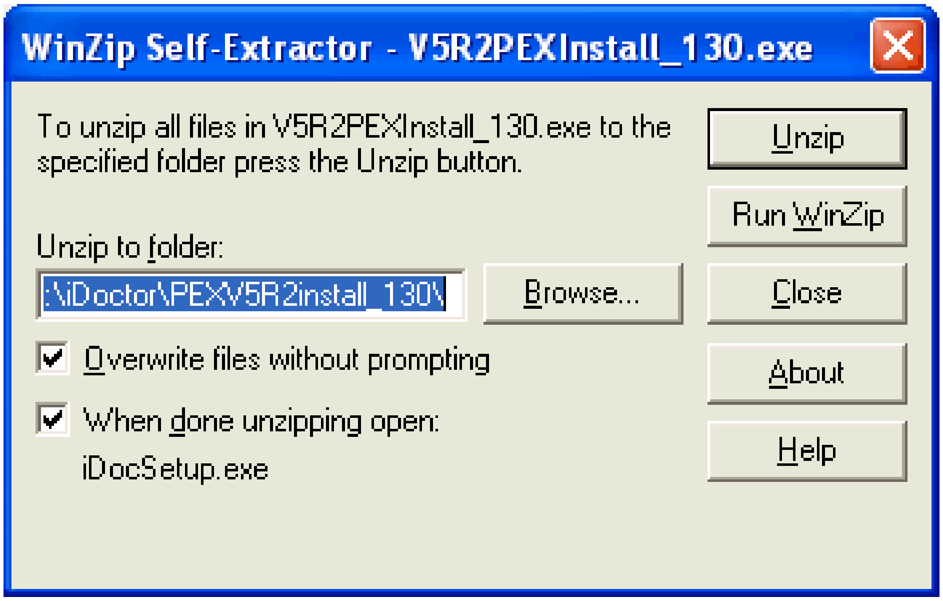
After installation you will have the following new libraries on your server: QYPBASE, QYPINT and SMTRACE.

The Install Process:

The iDoctor for iSeries install image contains a setup program which fully automates the process of installing PEX Analyzer.

Step 1 Download the install image for PEX Analyzer from our Web site to your PC.

Step 2 Double-click on the install image (it is a self-extracting .exe.) from within Windows Explorer. You will see the following:



WinZip Self-Extractor - V5R2PEXInstall_130.exe

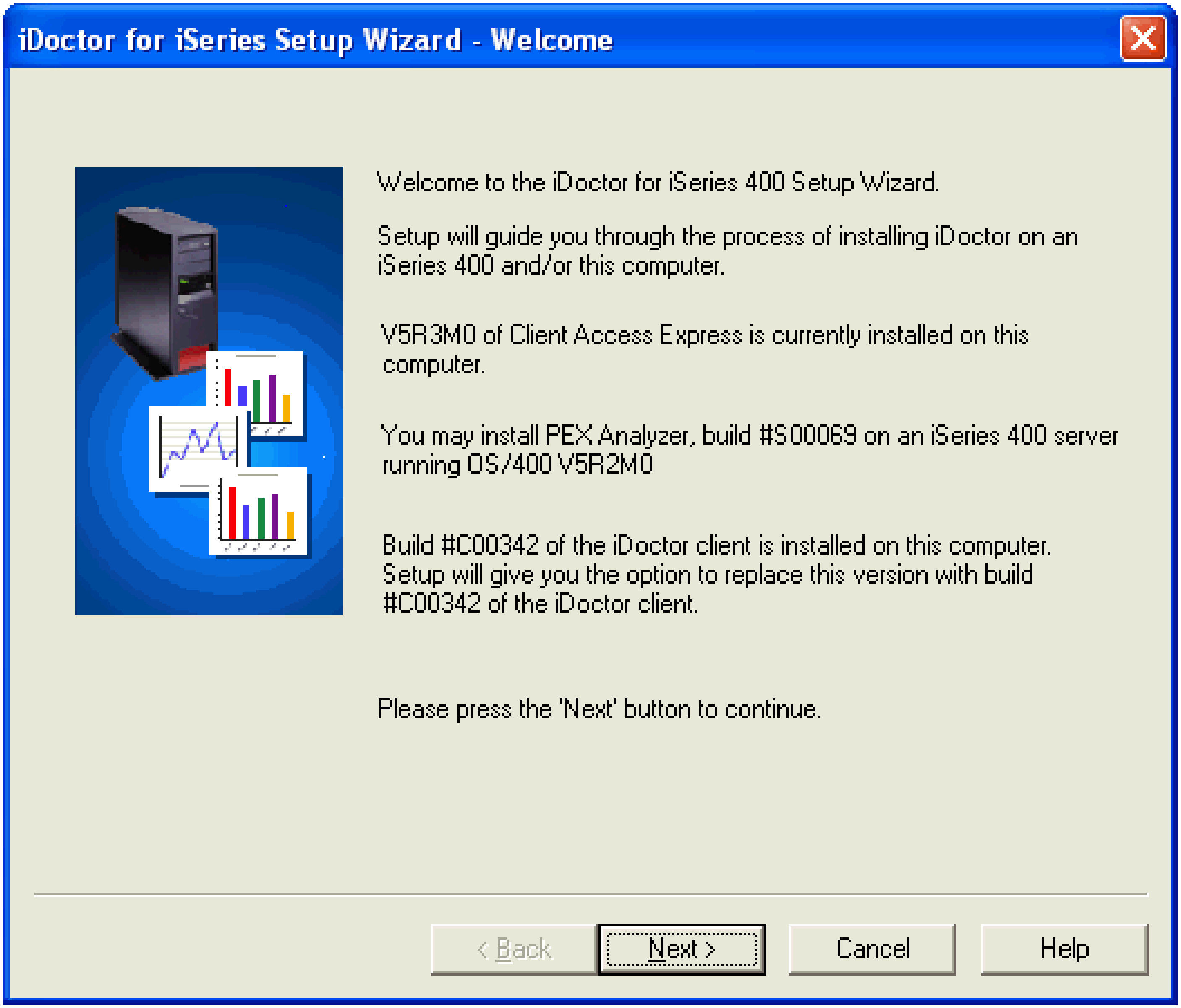
To unzip all files in V5R2PEXInstall_130.exe to the specified folder press the Unzip button.

Unzip to folder: :\iDoctor\PEX\V5R2install_130\

☑ Overwrite files without prompting

☑ When done unzipping open: iDocSetup.exe

Step 3 Change the path where the install image will be extracted to on the PC if desired and click the 'Unzip' button. Wait a moment for the files to be extracted and the setup program to be launched.



iDoctor for iSeries Setup Wizard - Welcome

Welcome to the iDoctor for iSeries 400 Setup Wizard.

Setup will guide you through the process of installing iDoctor on an iSeries 400 and/or this computer.

V5R3M0 of Client Access Express is currently installed on this computer.

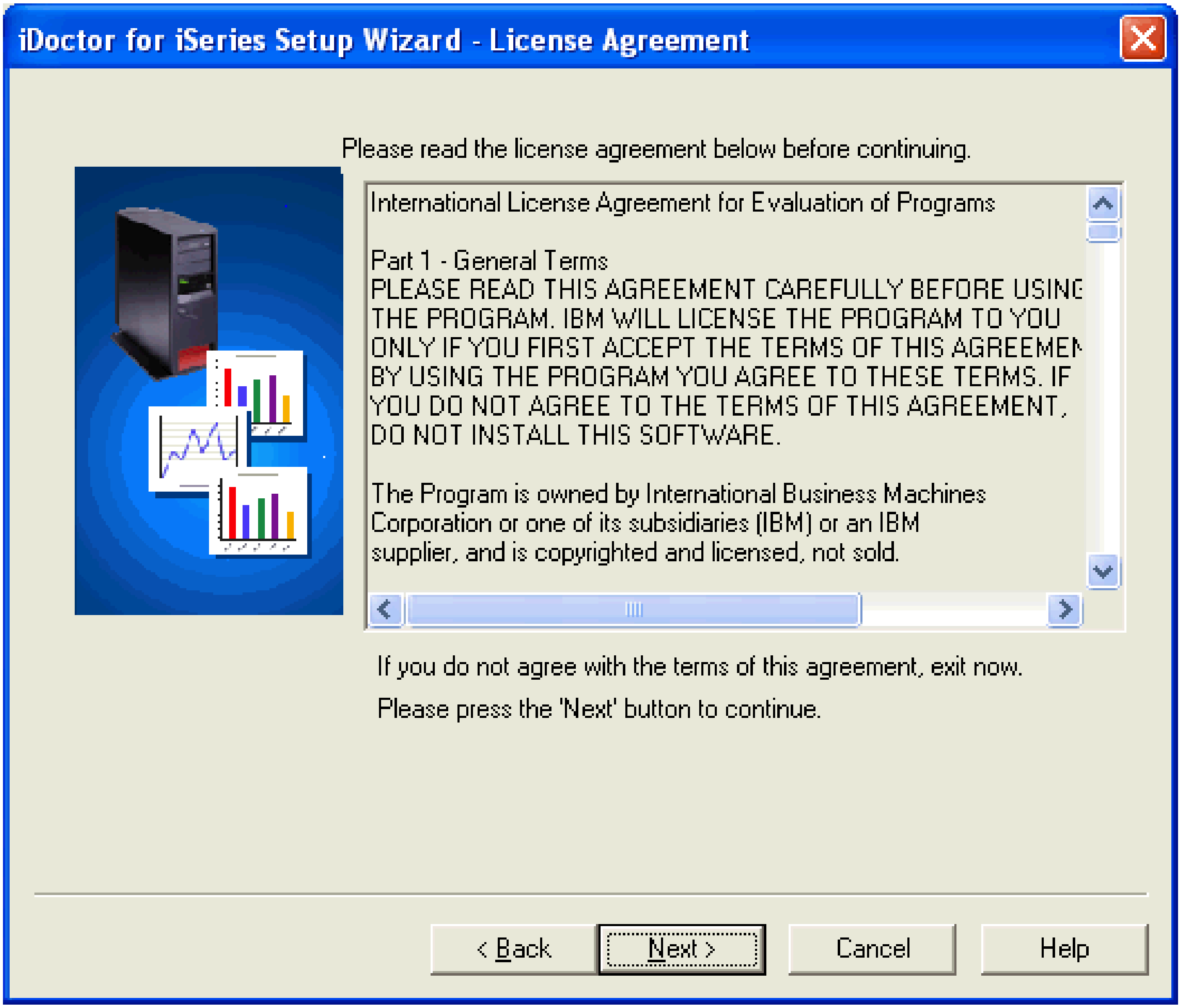
You may install PEX Analyzer, build #S00069 on an iSeries 400 server running OS/400 V5R2M0

Build #C00342 of the iDoctor client is installed on this computer. Setup will give you the option to replace this version with build #C00342 of the iDoctor client.

Please press the 'Next' button to continue.

This screen identifies the version of iSeries Access installed as well as the version of iDoctor for iSeries installed on the client (if any).

Step 4 On this screen, click the 'Next' button to indicate acceptance of the license agreement.



Please read the license agreement below before continuing.

International License Agreement for Evaluation of Programs

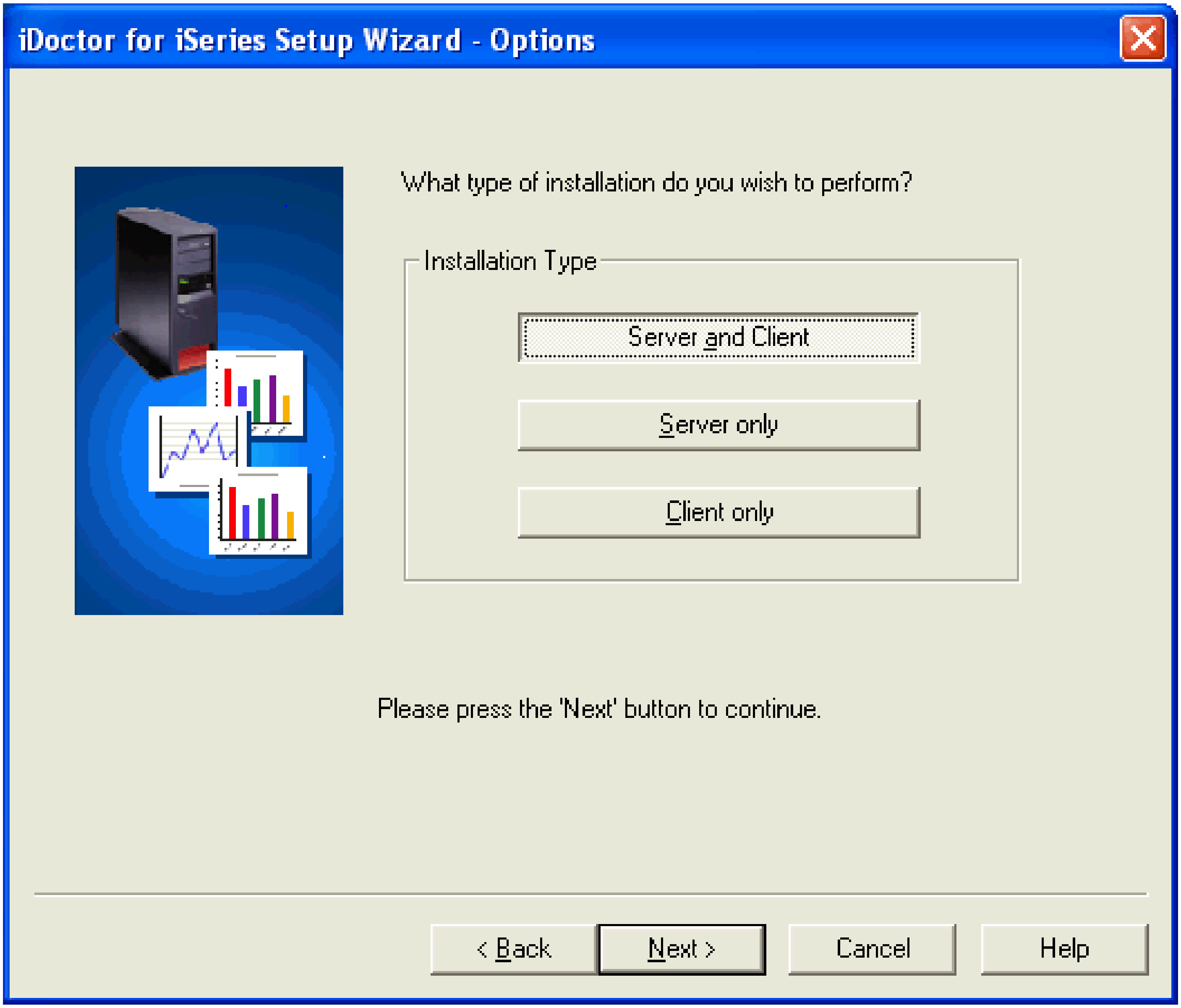
Part 1 - General Terms
PLEASE READ THIS AGREEMENT CAREFULLY BEFORE USING
THE PROGRAM. IBM WILL LICENSE THE PROGRAM TO YOU
ONLY IF YOU FIRST ACCEPT THE TERMS OF THIS AGREEMEN
BY USING THE PROGRAM YOU AGREE TO THESE TERMS. IF
YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT,
DO NOT INSTALL THIS SOFTWARE.

The Program is owned by International Business Machines
Corporation or one of its subsidiaries (IBM) or an IBM
supplier, and is copyrighted and licensed, not sold.

If you do not agree with the terms of this agreement, exit now.

Please press the 'Next' button to continue.

Step 5 Select the type of installation to perform and click 'Next'.



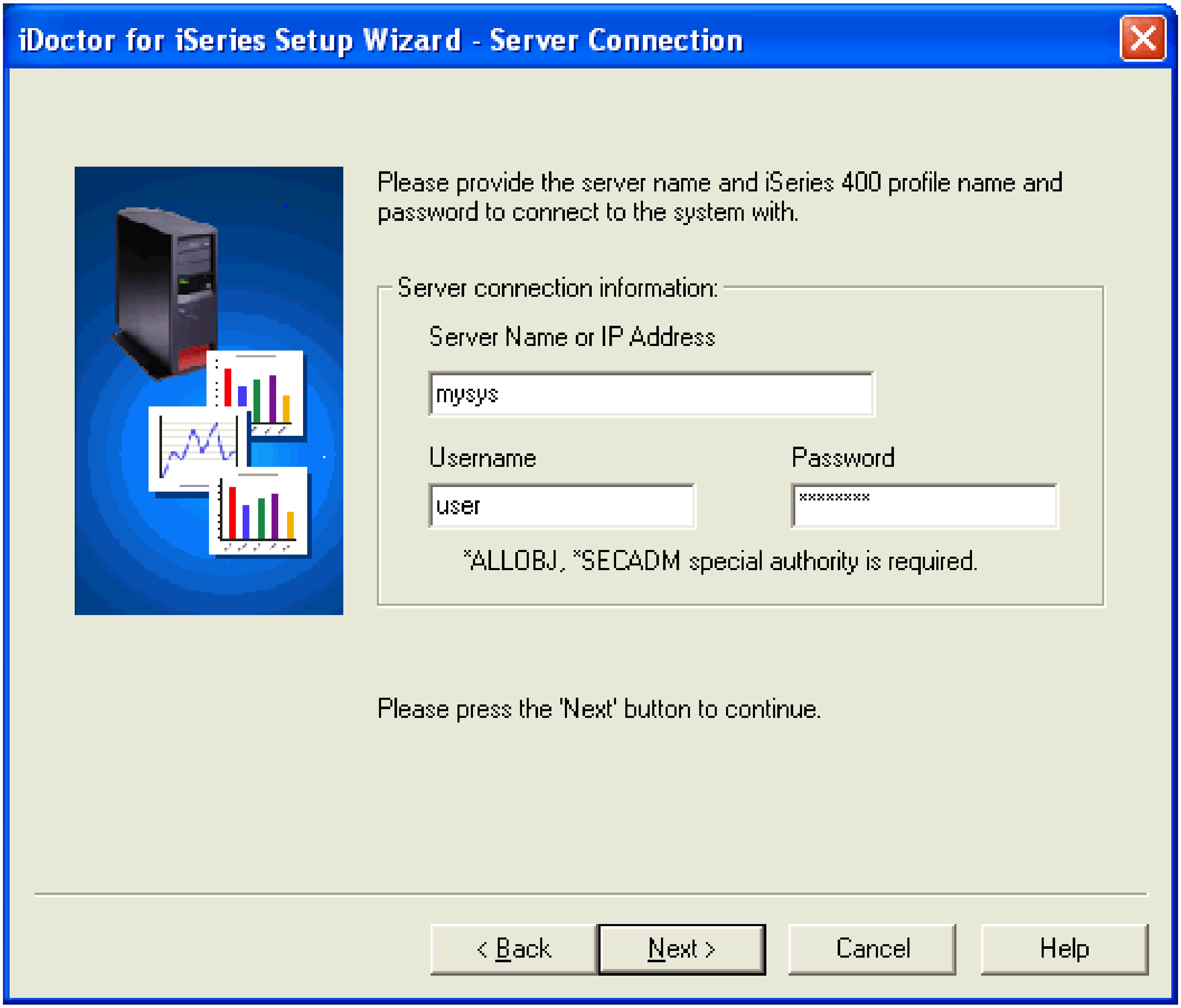
iDoctor for iSeries Setup Wizard - Options

What type of installation do you wish to perform?

Installation Type

Please press the 'Next' button to continue.

Step 6 If you are installing the server portion of PEX Analyzer you will see a screen asking for the connection information to use to access the server. The user profile must have the user class authority of *SECOFR and special authorities *ALLOBJ and *SECADM.



iDoctor for iSeries Setup Wizard - Server Connection

Please provide the server name and iSeries 400 profile name and password to connect to the system with.

Server connection information:

Server Name or IP Address

mysys

Username

user

Password

****

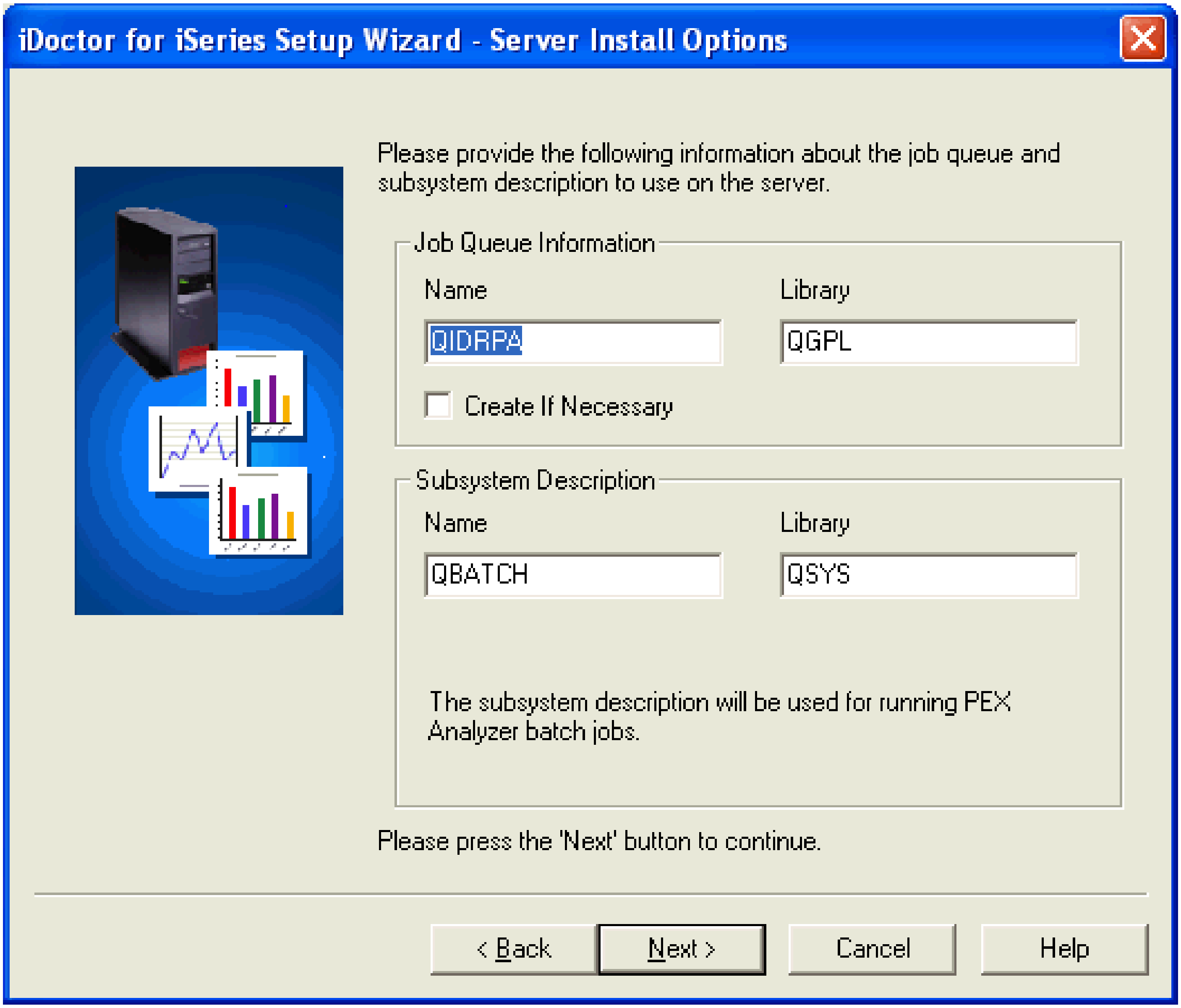
*ALLOBJ, *SECADM special authority is required.

Please press the 'Next' button to continue.

Click the 'Next' button to connect to the server specified and continue to the next screen.

Step 7 If you are installing the server portion of PEX Analyzer you will also be asked to specify a job queue name and a subsystem description the server code may use. This subsystem is used to run the PEX Analyzer analysis jobs. Only one analysis job can be active on the job queue at a time.

Caution for PEX Analyzer installs: There is nothing that prevents this job queue from being modified to allow more than one active job. However, if that is done, PEX Analyzer may fail to function properly and your server may require an IPL. In addition, it may be necessary to delete and restore the libraries containing PEX data.



iDoctor for iSeries Setup Wizard - Server Install Options

Please provide the following information about the job queue and subsystem description to use on the server.

Job Queue Information

Name: QIDRPA

Library: QGPL

□ Create If Necessary

Subsystem Description

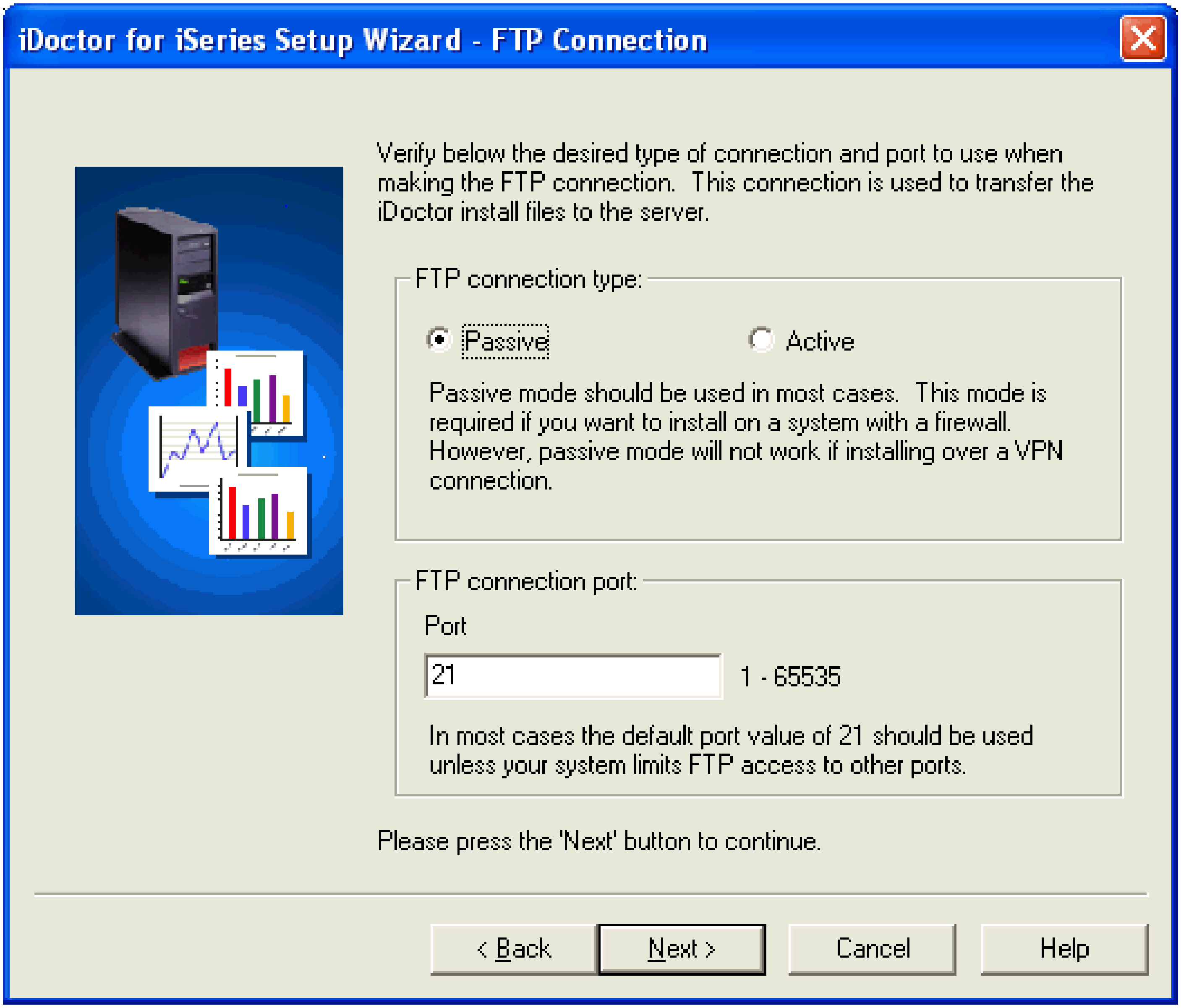
Name: QBATCH

Library: QSYS

The subsystem description will be used for running PEX Analyzer batch jobs.

Please press the 'Next' button to continue.

Step 8 The next page gives the user the option to specify which type of FTP connection should be used when performing the install. Only in unusual circumstances should anything other than the defaults be used on this page. However, if installing over a VPN connection and "Passive" FTP does not work, try using "Active" FTP instead.



iDoctor for iSeries Setup Wizard - FTP Connection

Verify below the desired type of connection and port to use when making the FTP connection. This connection is used to transfer the iDoctor install files to the server.

FTP connection type:

● Passive ○ Active

Passive mode should be used in most cases. This mode is required if you want to install on a system with a firewall. However, passive mode will not work if installing over a VPN connection.

FTP connection port:

Port

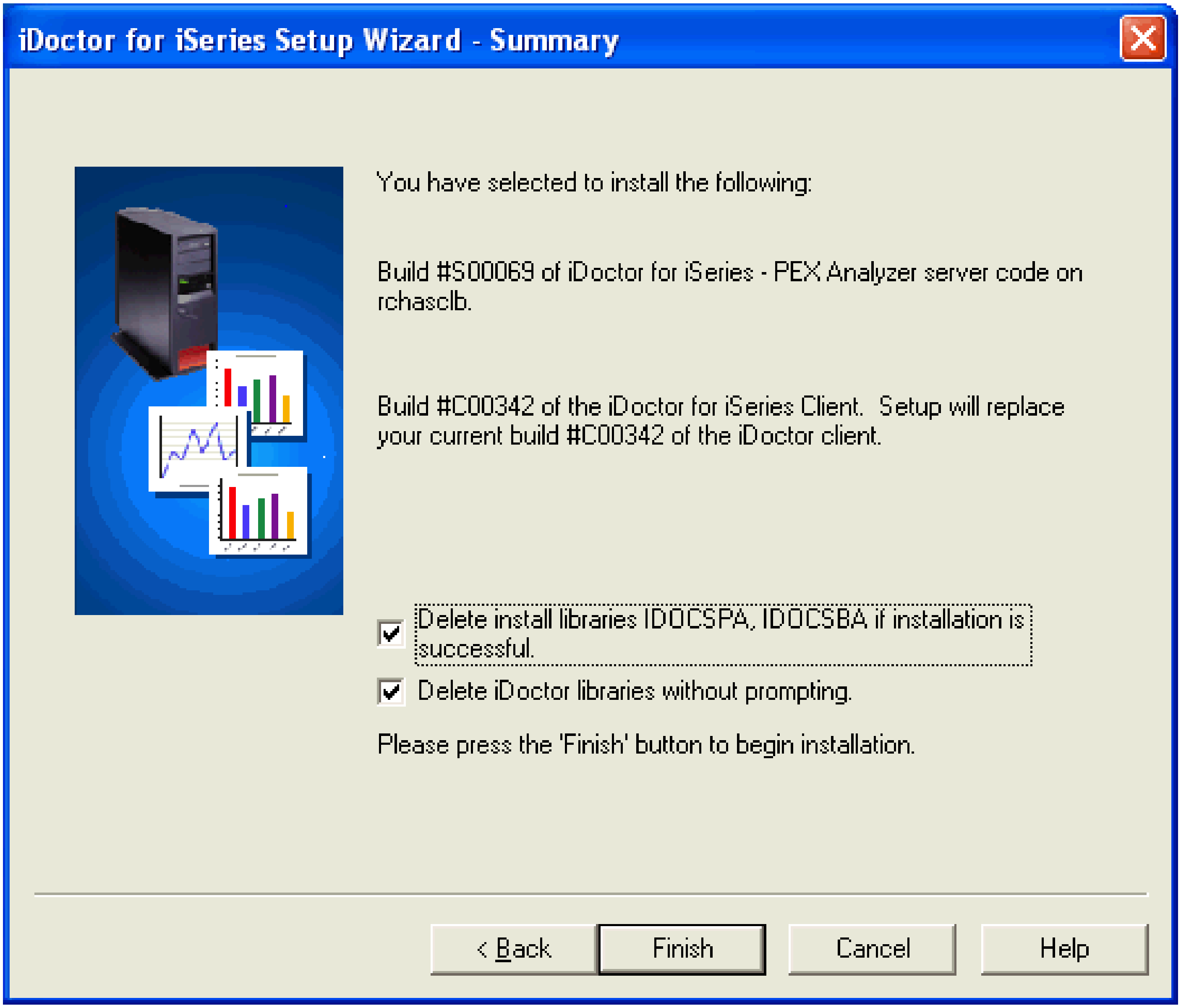
21 1 - 65535

In most cases the default port value of 21 should be used unless your system limits FTP access to other ports.

Please press the 'Next' button to continue.

Step 9 If an access code is required, your will be asked to enter it on the next screen. Enter the access code (optional) and continue to the 'Next' screen.

Step 10 A summary of your selections appears on the final screen. This page also contains options for whether the temporary libraries used during the installation should be deleted. During the install process these libraries will contain the restored contents of the save files sent to the server. Only if you were having problems with the install would the option to keep these libraries around be useful so they could be used for service.



iDoctor for iSeries Setup Wizard - Summary

You have selected to install the following:

Build #S00069 of iDoctor for iSeries - PEX Analyzer server code on rchasclb.

Build #C00342 of the iDoctor for iSeries Client. Setup will replace your current build #C00342 of the iDoctor client.

☑ Delete install libraries IDOCSPA, IDOCSBA if installation is successful.

☑ Delete iDoctor libraries without prompting.

Please press the 'Finish' button to begin installation.

Step 10 Clicking the 'Finish' button will copy all of the files and run the commands necessary to install the server and/or client portion of PEX Analyzer. The server portion of the installation may take a few minutes.

After the install completes the setup log file will be shown. If any errors occur, send this file to idoctor@us.ibm.com for assistance.

Table of Contents | Previous | Next

2.2.1 PTF Prerequisites for PEX Analyzer

Collecting PDC/PEX data on a machine involves running low-level LIC code that runs only when making such a collection. The machine making the collection must be loaded with the latest PTFs for PDC and PEX.

You should periodically use the link below to identify the **Performance Tools for AS/400 Group PTF** for your OS/400 Version/Release level. This Group PTF lists PTFs for performance analysis related tools (including PEX and PDC) in addition to
PTFs for the Performance Tools LPP.

Once you have found the correct Group PTF, download of the Group PTF and subsequent load/apply of all
- PDC PTFs (LIC)
- PEX PTFs (OS/400 or Performance Tools LPP)
within the Group PTF is highly recommended.

All Group PTFs for all releases

Table of Contents | Previous | Next

2.2.2 Manual installation steps for remote service

Purpose:

This document describes the steps to manually install PEX Analyzer on an iSeries running OS/400 V5R2. Using the GUI install wizard is preferred in most instances but if this is not possible, these steps are the only other way to get PEX Analyzer on the desired system.

Self-Installation Prep:

This document assumes that the user has downloaded the PEX Analyzer install image from the V5R2 downloads page. This self extracting ZIP file is named V5R2PEXInstall_130.exe and contains the following files which will be used in this document:

baseV5R2.savf (save file for IDOCSBA temporary install library)
V5R2.savf (save file for IDOCSPA temporary install library)

These files should be extracted to a directory on the PC so they may be sent to the iSeries via FTP.

The user profile performing the installation must have *SECADM, *ALLOBJ special authorities. Otherwise all objects may not be restored properly

The system value QALWOBJRST must be set to *ALL or certain objects will not be installed properly. Temporarily change this system value if necessary in order to perform this install. Remember to change it back to what it was when you started after the manual install has completed.

Installation Steps:

Perform the following steps in order to manually install PEX Analyzer.

1. From a green screen window, run the following commands:
- DLTLIB IDOCINST
- DLTLIB QYPBASE
- DLTLIB QYPINT
- DLTLIB SMTRACE
- DLTLIB IDOCSBA
- DLTLIB IDOCSPA

- CRTLIB IDOCINST
- CRTSAVF FILE(IDOCINST/IDOCINST)

2. Open a DOS FTP session to the iSeries.

3. FTP file baseV5R2.savf from the PC to the iSeries save file IDOCINST in library IDOCINST.

4. From a green screen window, run the following commands:
- DSPSAVF FILE(IDOCINST/IDOCINST)
- verify that the save file contains data
- RSTLIB IDOCSBA *SAVF SAVF(IDOCINST/IDOCINST) MBROPT(*ALL) ALWOBJDIF(*ALL)
- IDOCSBA/INSTIDOCBA
- verify that the message "CPF9898 - SUCCESSFUL iDoctor server Base install." is returned by the command.
- this command will install the QYPBASE library which is needed for the GUI to work
- CLRSAVF FILE(IDOCINST/IDOCINST)

5. FTP file V5R2.savf from the PC to the iSeries save file IDOCINST in library IDOCINST.

6. From a green screen window, run the following commands:
- DSPSAVF FILE(IDOCINST/IDOCINST)
- verify that the save file contains data
- RSTLIB IDOCSPA *SAVF SAVF(IDOCINST/IDOCINST) MBROPT(*ALL) ALWOBJDIF(*ALL)
- IDOCSPA/INSTIDOCPA JOBQ(QGPL/QIDOCPA) CRTJOBQ(*YES) SBSD(QSYS/QIDOCPA)
- the jobq and sbsd parms are the job queue and subsystem that job watches should run under when submitted from the client
- create a job queue and subsystem description if desired or use an existing job queue and subsystem
- verify that the message "CPF9898 - SUCCESSFUL iDoctor server PEX Analyzer install." is returned by the command.
- this command will install PEX Analyzer into libraries SMTRACE and QYPINT

7. From a green screen window, run the following commands:
- DLTLIB IDOCINST
- DLTLIB IDOCSBA
- DLTLIB IDOCSPA

If you are unable to complete the manual installation successfully e-mail idoctor@us.ibm.com for assistance.

Table of Contents | Previous | Next

2.3 Installing Heap Analyzer

PC Requirements:

- Windows XP, NT 4.0, 2000
- IBM iSeries Access for Windows (V5R2 or higher)
- 256 MB or higher

Server Requirements:

- OS/400 V5R2
- PTFs MF30028, MF30445, SI08420 and Java Group PTF 5722JV1 SF99169 level 8 or higher
- The user profile performing the installation must have *SECOFR user class and special authorities *ALLOBJ and *SECADM.
- The following host servers (identified by the SERVER parameter values on the STRHOSTSVR command) need to be running on the server: *DATABASE, *RMTCMD, *SIGNON, *SRVMAP
- System value QALWOBJRST must be *ALL or (*ALWSYSSTT and *ALWPGMADP)

After installation you will have the following new libraries on your server: QYPBASE and QPYRTJW

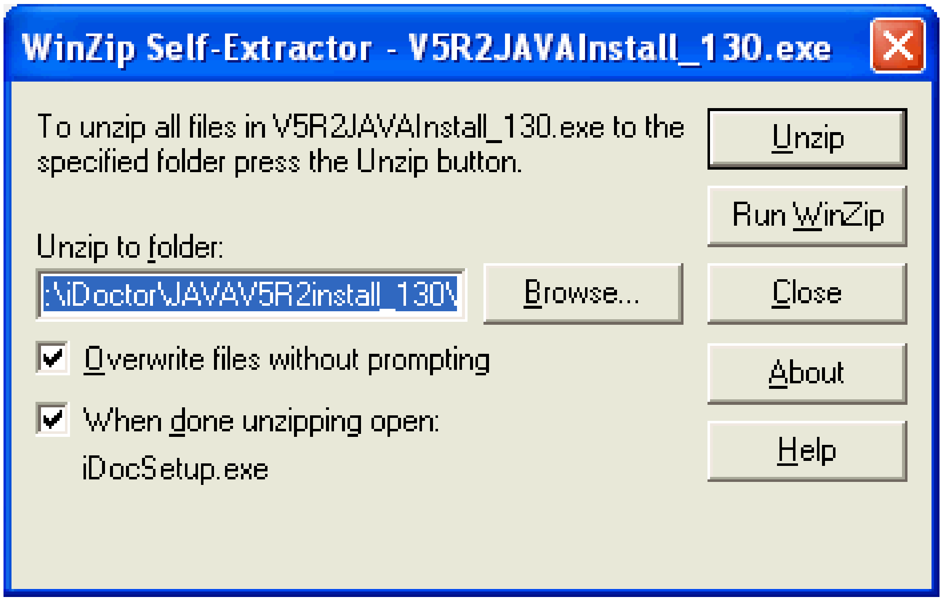
The Install Process:

The GUI install process will install the server objects and programs on your iSeries and the client code on your PC. The PTFs required for Heap Analyzer must be manually installed via whichever method normally used to get PTFs for your iSeries. The PTFs required are MF30028, MF30445, SI08420 and Java Group PTF 5722JV1 SF99169 level 8 or higher. If unsure of which method to use for installing the PTFs, use the Fix Central Web site. Instructions for installing these PTFs can be found in the next section.

After the PTFs are loaded and applied on your iSeries, perform the following steps:

Step 1 Download the install image for Heap Analyzer from our Web site to your PC.

Step 2 Double-click on the install image (it is a self-extracting .exe.) from within Windows Explorer. You will see the following:



WinZip Self-Extractor - V5R2JAVAInstall_130.exe

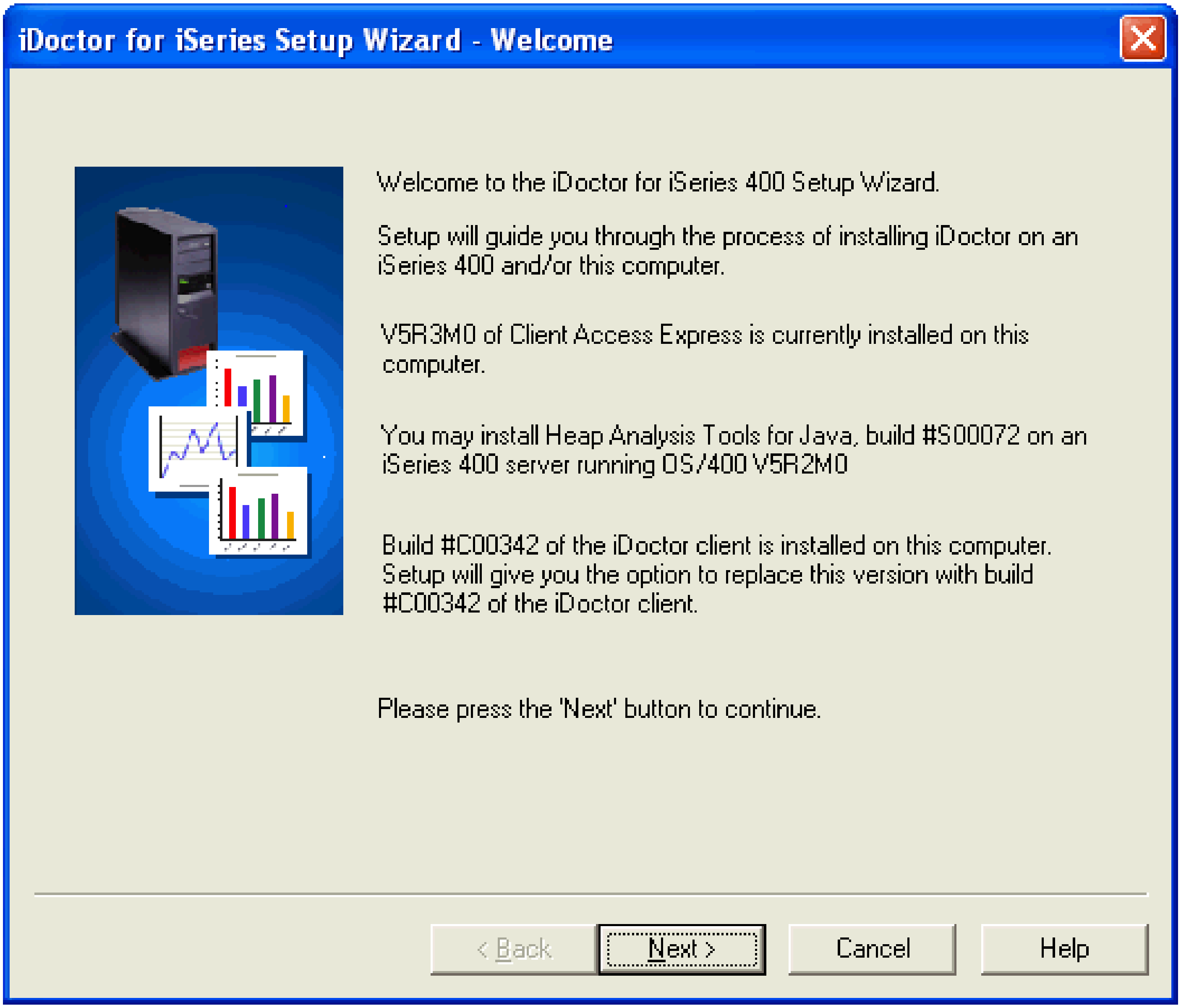
To unzip all files in V5R2JAVAInstall_130.exe to the specified folder press the Unzip button.

Unzip to folder: :\iDoctor\JAVAV5R2install_130\

☑ Overwrite files without prompting

☑ When done unzipping open: iDocSetup.exe

Step 3 Change the path where the install image will be extracted to on the PC if desired and click the 'Unzip' button. Wait a moment for the files to be extracted and the setup program to be launched.



iDoctor for iSeries Setup Wizard - Welcome

Welcome to the iDoctor for iSeries 400 Setup Wizard.

Setup will guide you through the process of installing iDoctor on an iSeries 400 and/or this computer.

V5R3M0 of Client Access Express is currently installed on this computer.

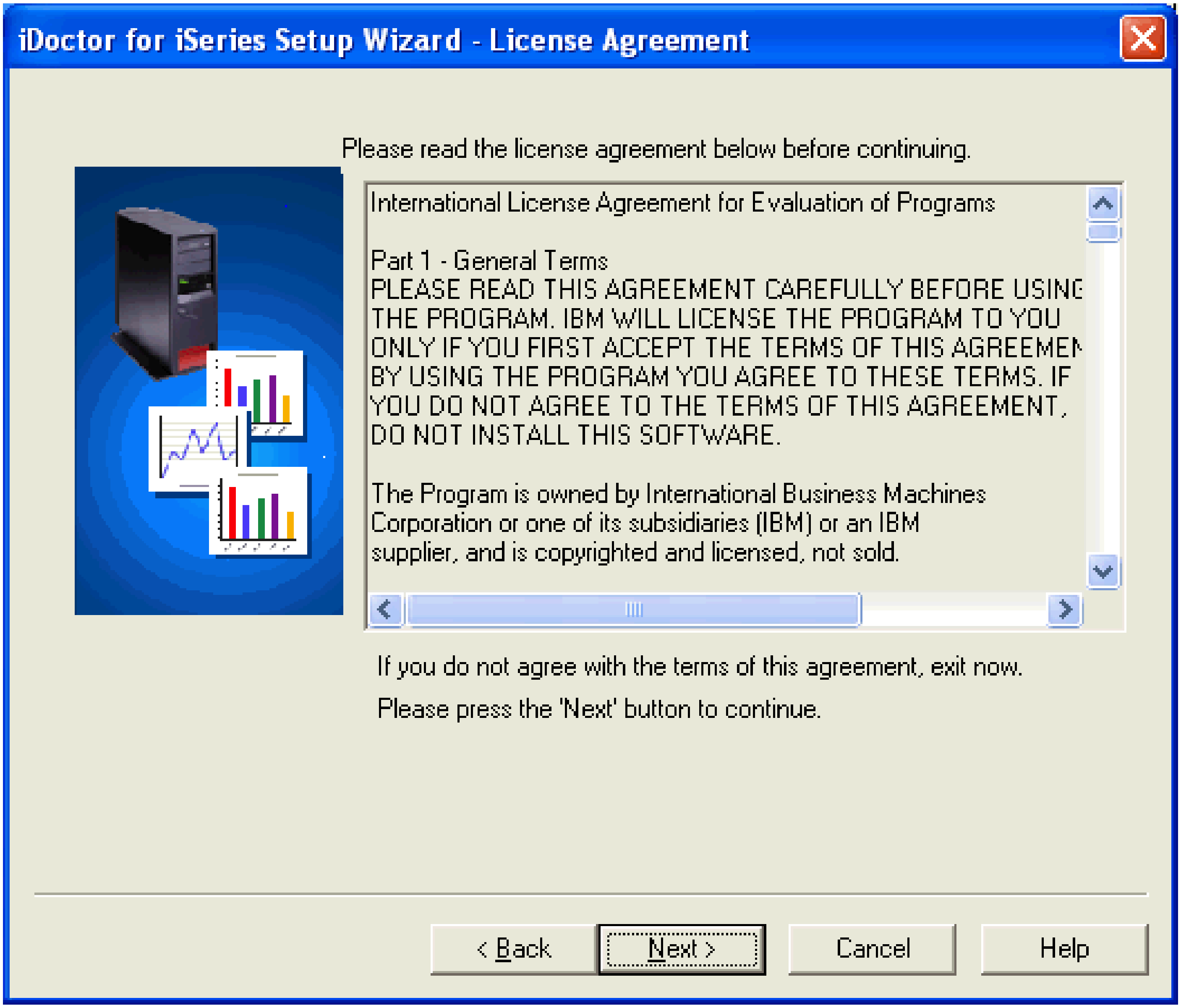
You may install Heap Analysis Tools for Java, build #S00072 on an iSeries 400 server running OS/400 V5R2M0

Build #C00342 of the iDoctor client is installed on this computer. Setup will give you the option to replace this version with build #C00342 of the iDoctor client.

Please press the 'Next' button to continue.

This screen identifies the version of iSeriesAccess installed as well as the version of iDoctor for iSeries client installed (if found). Click 'Next'.

Step 4 On the following screen, click the 'Next' button to indicate acceptance of the license agreement.



Please read the license agreement below before continuing.

International License Agreement for Evaluation of Programs

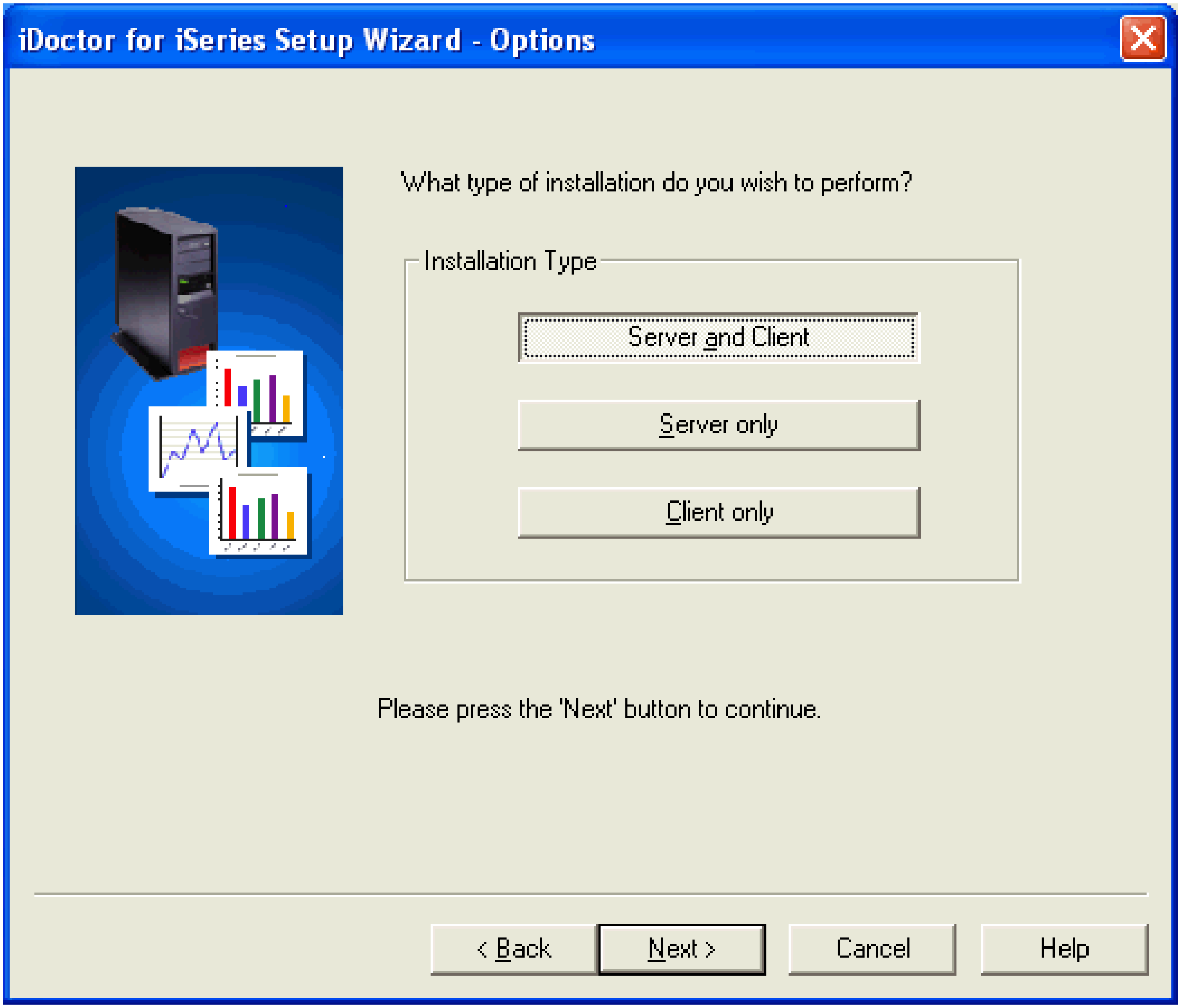
Part 1 - General Terms
PLEASE READ THIS AGREEMENT CAREFULLY BEFORE USING
THE PROGRAM. IBM WILL LICENSE THE PROGRAM TO YOU
ONLY IF YOU FIRST ACCEPT THE TERMS OF THIS AGREEMEN
BY USING THE PROGRAM YOU AGREE TO THESE TERMS. IF
YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT,
DO NOT INSTALL THIS SOFTWARE.

The Program is owned by International Business Machines
Corporation or one of its subsidiaries (IBM) or an IBM
supplier, and is copyrighted and licensed, not sold.

If you do not agree with the terms of this agreement, exit now.

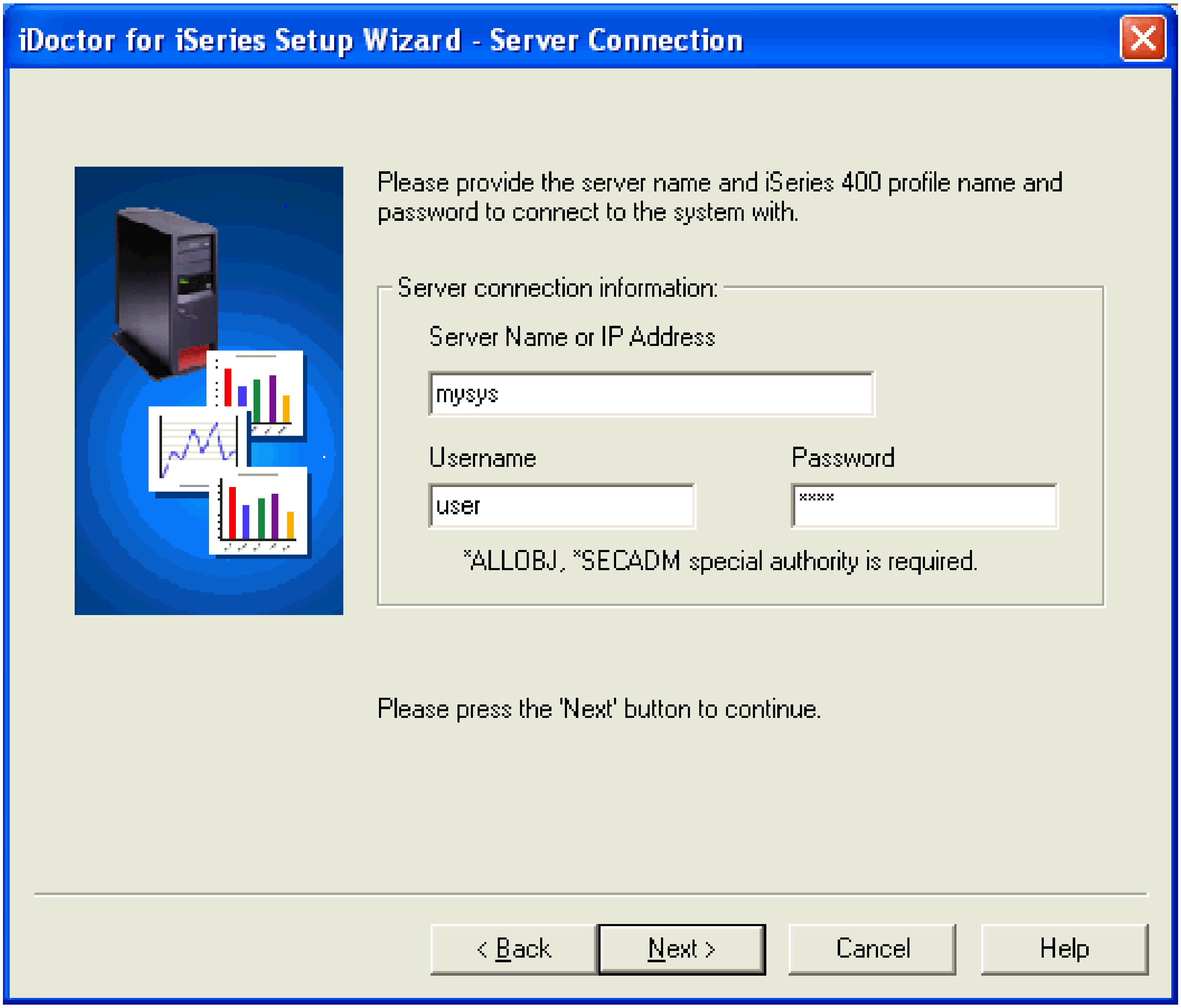
Please press the 'Next' button to continue.

Step 5 Select the type of installation to perform. This screen allows you to choose whether to install the server side of Heap Analyzer, the client side of iDoctor or both.



iDoctor for iSeries Setup Wizard - Options
What type of installation do you wish to perform?
Installation Type
Server and Client
Server only
Client only
Please press the 'Next' button to continue.
< Back
Next >
Cancel
Help

Step 6 If you are installing the server portion of Heap Analyzer you will see a screen asking for the connection information to use to access the server. The user profile must have the user class authority of *SECOFR and *ALLOBJ, *SECADM special authorities.



iDoctor for iSeries Setup Wizard - Server Connection

Please provide the server name and iSeries 400 profile name and password to connect to the system with.

Server connection information:

Server Name or IP Address: mysys

Username: user

Password: ****

*ALLOBJ, *SECADM special authority is required.

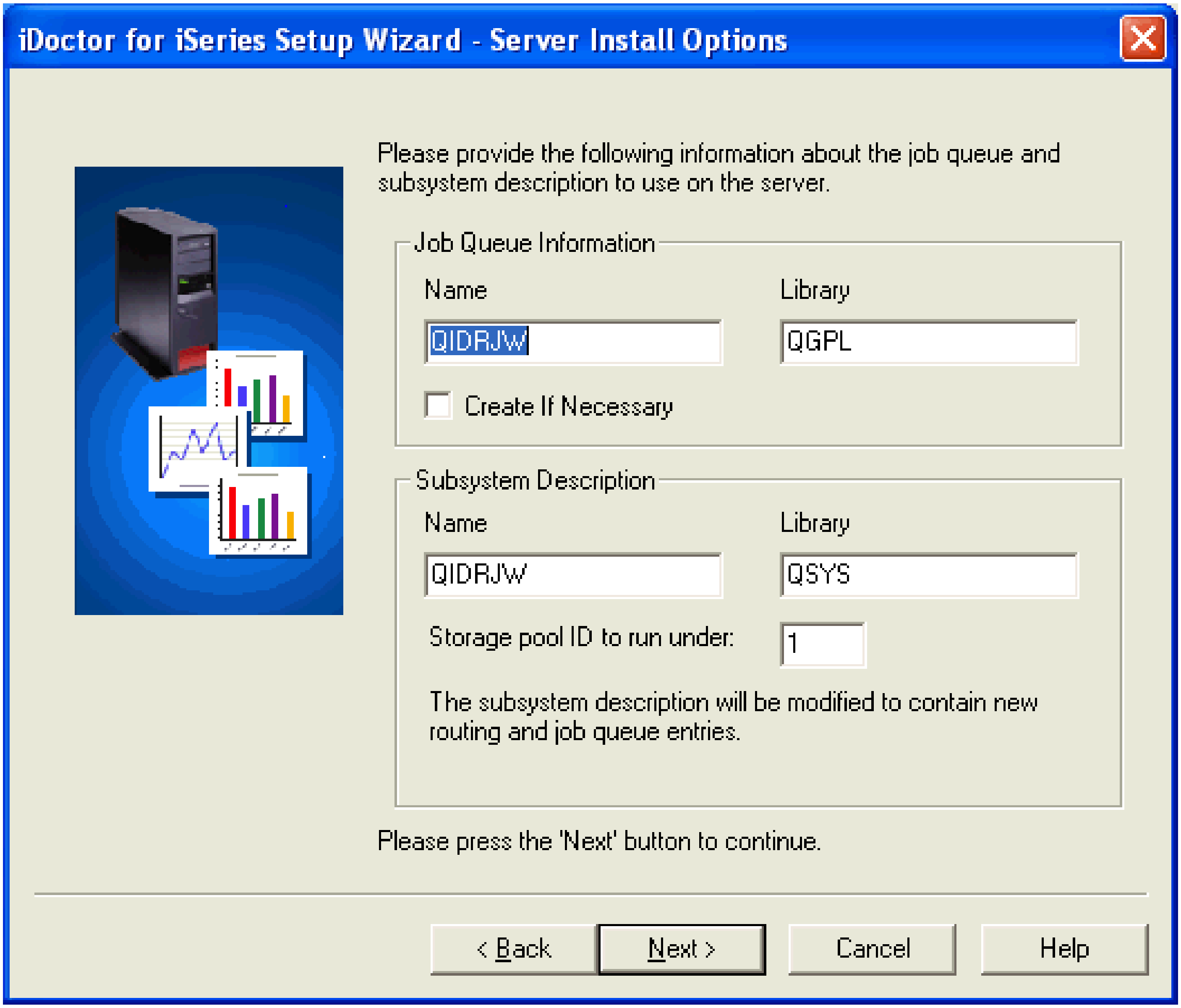
Please press the 'Next' button to continue.

Click the 'Next' button to connect to the server specified and continue on the next page.

Step 7 If you are installing the server portion of Heap Analyzer you will also be asked to specify a job queue name and a subsystem description the server code may use when running Heap Analyzer. If the subsystem does not already exist, the install program will ask if it should be created. This subsystem and job queue will be used to run the Heap Analyzer "watch" jobs producing Java heap information. Heap Analyzer does not have a limit to the number of "watches" that can be active at one time.

You must also indicate the storage pool ID that the Heap Analyzer jobs should run under. This parameter is required in order to add routing entries to the subsystem description.

Press the 'Next' button to continue to the summary page.



iDoctor for iSeries Setup Wizard - Server Install Options

Please provide the following information about the job queue and subsystem description to use on the server.

Job Queue Information

Name: QIDRJW

Library: QGPL

□ Create If Necessary

Subsystem Description

Name: QIDRJW

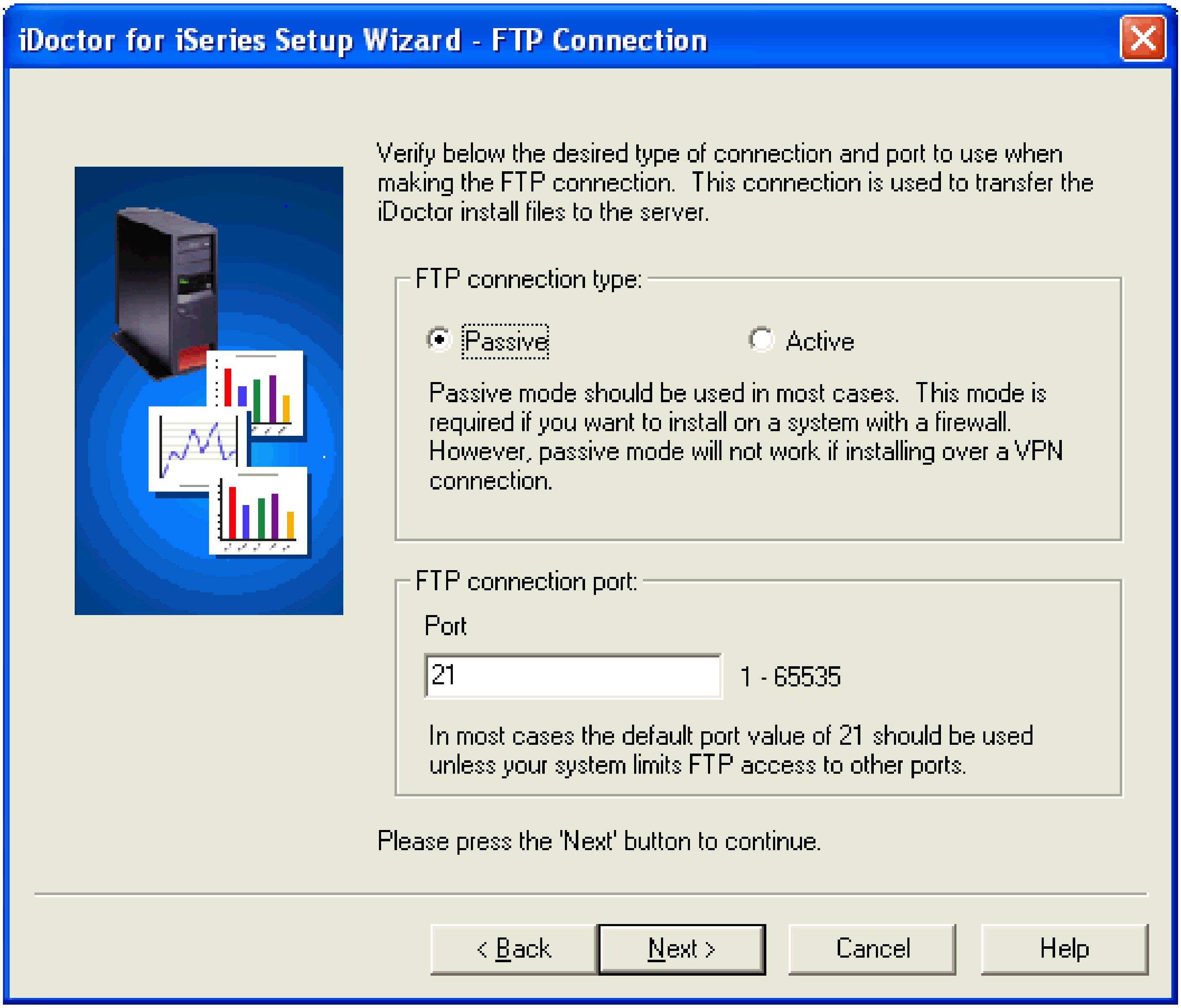
Library: QSYS

Storage pool ID to run under: 1

The subsystem description will be modified to contain new routing and job queue entries.

Please press the 'Next' button to continue.

Step 8 The next page gives the user the option to specify which type of FTP connection should be used when performing the install. Only in unusual circumstances should anything other than the defaults be used on this page. However, if installing over a VPN connection and "Passive" FTP does not work, try using "Active" FTP instead.



iDoctor for iSeries Setup Wizard - FTP Connection

Verify below the desired type of connection and port to use when making the FTP connection. This connection is used to transfer the iDoctor install files to the server.

FTP connection type:

● Passive ○ Active

Passive mode should be used in most cases. This mode is required if you want to install on a system with a firewall. However, passive mode will not work if installing over a VPN connection.

FTP connection port:

Port

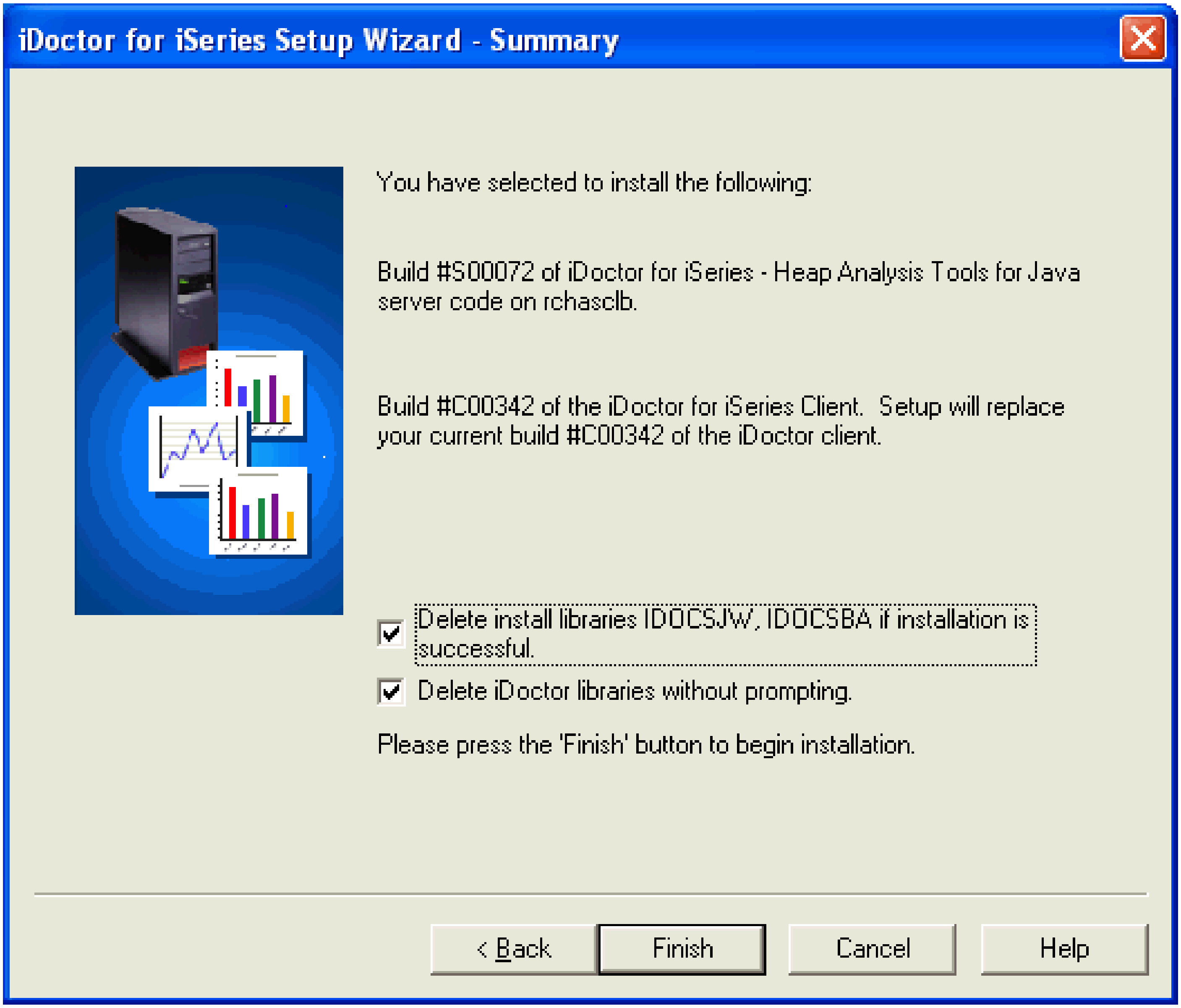
21 1 - 65535

In most cases the default port value of 21 should be used unless your system limits FTP access to other ports.

Please press the 'Next' button to continue.

Step 9 If an access code is required, your will be asked to enter it on the next screen. Enter the access code (optional) and continue to the 'Next' screen.

Step 10 A summary of your selections appears on the final screen. This page also contains options for whether the temporary libraries used during the installation should be deleted. During the install process these libraries will contain the restored contents of the save files sent to the server. Only if you were having problems with the install would the option to keep these libraries around be useful so they could be used for service.



iDoctor for iSeries Setup Wizard - Summary

You have selected to install the following:

Build #S00072 of iDoctor for iSeries - Heap Analysis Tools for Java server code on rchasclb.

Build #C00342 of the iDoctor for iSeries Client. Setup will replace your current build #C00342 of the iDoctor client.

☑ Delete install libraries IDOCSJW, IDOCSBA if installation is successful.

☑ Delete iDoctor libraries without prompting.

Please press the 'Finish' button to begin installation.

Step 11 Clicking the 'Finish' button will copy all of the files and run the commands necessary to install the server and/or client portion of Job Watcher. The server portion of the installation may take a few minutes.

After the install completes the setup log file will be shown. If any errors occur, send this file to idoctor@us.ibm.com for assistance.

Table of Contents | Previous | Next

2.4 Installing PTDV

PTDV has the following requirements:
On the server:

Requirements	**Thin Client**	**Thick Client**	**Three Tier**
OS/400 V4R5 or later	**X**	**X**	**X**
TC1 - TCP/IP Connectivity Utilities	**X**	**X**	**X**
JV1 - Developer Kit for Java		**X**	**X**
JC1 - Toolbox for Java	**X**		**X**
Host servers option of OS/400		**X**	**X**

On the client:
The PTDV client requires the following Java environment to run correctly:

- Java 2 runtime environment (JDK 1.2 or higher) is required. PTDV is not supported for JDK 1.1.x. Java 1.3 or higher is recommended.
- iSeries Toolbox for Java. This must be i5/OS V5R3M0, or a release more recent than that of the server.

Refer to the alphaWorks web site for additional information on downloading and installing PTDV.

Table of Contents | Previous | Next

2.5 How to authorize use of iDoctor for iSeries

Note: This section does not apply to the PTDV or Heap Analyzer components.

There are three ways to authorize use of iDoctor for iSeries on a server.

Option 1 By providing the access code to the setup wizard, when prompted for it.

Option 2 Provide the access code after launching a component from the GUI. You will be prompted for the access code if a valid access code could not be found on your system.

Option 3 By using the green screen command QYPBASE/ADDPRDACS.

Please follow these steps:
a) Open an iSeries interactive session:
b) Type QYPBASE/ADDPRDACS and press F4
c) Type in the access code that you were given by IBM Support.
d) Press enter.

You are now authorized to use iDoctor for iSeries. If you are using the tool with an evaluation code, your access will expire 45 days from the date it was dispatched.

You may request an access code for an evaluation version by sending e-mail to idoctor@us.ibm.com

Table of Contents | Previous | Next

2.6 Uninstalling iDoctor for iSeries

Note: This section does not apply to the PTDV component.

Server side

To uninstall an iDoctor for iSeries component from a server, you need to remove the libraries created during installation. The following table describes the libraries installed on the server, per component.

	PEX Analyzer	**Job Watcher/Heap Analyzer**
Libraries	QYPBASE, QYPINT, SMTRACE	QYPBASE, QPYRTJW

Client side

To remove iDoctor for iSeries from your PC select the uninstall program from the Start Menu: Start -> Programs -> iDoctor for iSeries -> Uninstall iDoctor for iSeries.

Table of Contents | Previous | Next

Part II Server-side components

This part covers the server-side (libraries, database files, commands) of each iDoctor for iSeries component.

Table of Contents | Previous | Next

Chapter 1 Job Watcher - job-specific mode

Job-specific Job Watcher offers the ability to collect information about a set of jobs running on a system. A maximum of 10 jobs may be specified with a total of 500 threads within those jobs being collected during each interval.

The information collected by job-specific Job Watcher includes basic job statistics (cpu, IOs, faulting), wait information (time spent in each type of wait), call stacks, SQL statements, communications detail and activation group information. Unlike system-wide Job Watcher data is collected every interval for every job regardless if the job was idle or not.

This chapter describes the job-specific mode of the Job Watcher component on the server side. The following topics will be covered:

- Libraries QYPBASE, QPYRTJW
- Database files
- Command documentation

Table of Contents Previous Next

1.1 Libraries

When Job Watcher is installed, two libraries are created on the iSeries. These libraries are named QYPBASE and QPYRTJW.

QYPBASE contains programs and commands needed by the iDoctor for iSeries client. It also contains objects that are used by both Job Watcher and PEX Analyzer.

Library QPYRTJW is the Job Watcher library for release V5R2. This library contains programs, commands and database files needed to create and work with Job Watcher data.

Table of Contents | ◀ Previous | ▶ Next

1.2 Output files

Job Watcher provides two different types of collections over the jobs/threads/tasks on a system: job-specific and system-wide. Each type of Job Watch uses several different output files that the client will interpret to produce summarized and detailed reports.

Job-specific job watches are created via the GUI or using the WCHJOB green screen command. This section contains information about each file that makes up a job-specific Job Watch.

Table of Contents | Previous | Next

1.2.1 QPYRTJWA - Basic job statistics interval deltas

Description: This file contains many different types of statistics for a thread/task collected in a job-specific Job Watch.

Optional: No

Job Watch: Yes

Record: One record is created per active thread/task per interval

Field name	Field Description	Format	Comments
INTERVAL	Interval number	Binary(4)	
STATUS	Status	Char(2)	Not used
STARTOD	Interval start date/time	Timestamp	
ENDTOD	Interval end date/time	Timestamp	
STARTTODP	Previous interval start date/time	Timestamp	
ENDTODP	Previous interval end date/time	Timestamp	
CYCLESMS	Cycles per ms	Packed(8)	The central processing unit (CPU) cycles per millisecond rate. This is a system constant based on system model and feature.
DELTAUS	Interval delta usecs	Packed(8)	Elapsed interval time (in microseconds)
TASKCOUNT	Thread/task task count	Packed(11)	

DELTACPU	Delta CPU usecs	Packed(8)	CPU used this interval (in microseconds) **Note:** This field, unlike the CPU time accounted for in the "run/wait buckets" **will include CPU time used by background server tasks** that may be invoked by the current thread/task.
DELTACYC	Delta cycles	Packed(11)	
DELTAINS	Delta instructions	Packed(11)	
BIRTHDAY	Job/task creation date/time	Timestamp	
NAME	Job/task name	Char(32)	Fully qualified job name or task name
ORIGPRI	Original priority	Packed(2)	LIC priority for the thread/task when created
PRIORITY	Current LIC priority	Packed(2)	
EXTENDER	Job name extender	Char(2)	

TYPE	Job or task type flag	Char(1)	T = Task
			P = Job/Thread (Process)

POOL	Pool ID (in decimal)	Packed(2)	ID of the Pool the job is running in
PRICHG	Priority changed?	Char(1)	Y or N if priority changed from previous interval
POOLCHG	Pool changed?	Char(1)	Y or N if pool ID changed from previous interval

TOTWRT	Total DASD writes	Binary(4)	
TOTPRMWRT	Total DASD permanent writes	Binary(4)	
SYNDBRD	Synchronous DB reads	Binary(4)	synchronous database reads
SYNNDBRD	Synchronous non-DB reads	Binary(4)	synchronous non-database reads
SYNDBWRT	Synchronous DB writes	Binary(4)	synchronous database writes
SYNNDBWRT	Synchronous non-DB writes	Binary(4)	synchronous non-database writes
ASYDBRD	Asynchronous DB reads	Binary(4)	asynchronous database reads
ASYNDBRD	Asynchronous non-DB reads	Binary(4)	asynchronous non-database reads
ASYDBWRT	Asynchronous DB writes	Binary(4)	asynchronous database writes
ASYNDBWRT	Asynchronous non-DB writes	Binary(4)	asynchronous non-database writes
IOPENDING	IO pending page faults	Binary(4)	
SMSYNCIO	Waits for asynchronous writes	Binary(4)	
FLTS	Page faults resulting in DASD reads	Binary(4)	
ALLOCATED	Allocated DASD pages	Binary(4)	Number of 4k DASD pages allocated this interval
DEALLOCED	Deallocated DASD pages	Binary(4)	Number of 4k DASD pages deallocated this interval
SEIZE	Seize time (usecs)	Packed(8)	in microseconds
BINOVER	Binary overflows	Binary(4)	

DECOVER	Decimal overflows	Binary(4)	
FLOATOVER	Float overflows	Binary(4)	
THREADID	Thread ID (hex)	Hex(16)	
STMFRD	Stream file reads	Binary(4)	
STMFWRT	Stream file writes	Binary(4)	
MUTEX	Mutex time (usecs)	Packed(8)	in microseconds
ACTWAIT	Active to wait transitions	Binary(4)	
WAITINEL	Wait to ineligible transitions	Binary(4)	
ACTINEL	Active to ineligible transitions	Binary(4)	
DELTAPRCPU	Total job CPU usecs delta	Packed(8)	
ACTTHREADD	Active thread count delta	Binary(4)	
CRTTHREADD	Total threads created delta	Binary(4)	
ACTTHREADC	Active threads count	Binary(4)	
CRTTHREADC	Total threads created	Binary(4)	
SOCKRD	Socket reads	Binary(4)	
SOCKWRT	Socket writes	Binary(4)	
SOCKBRD	Socket bytes read	Packed(11)	
SOCKBWRT	Socket bytes written	Packed(11)	
QCOUNT01-QCOUNT32	Queueing bucket 01-32 count	Binary(4)	Total occurrences of each category of wait
QTIME01-QTIME32	Queueing bucket 01-32 time	Packed(11)	Total time (in microsconds) for each category of wait

CURRSTATE	Current state	Char(4)	RUN - using the CPU CPUQ - waiting to use the CPU WAIT - waiting
ACTB	Active before sample?	Char(1)	Immediately before taking the snapshot for the interval Y or N if the CPU was being used
ACTA	Active after sample?	Char(1)	Immediately after taking the snapshot for the interval Y or N if the CPU was being used
RDYB	Ready before sample?	Char(1)	Immediately before taking the snapshot for the interval Y or N if the job was in a CPU queueing state
RDYA	Ready after sample?	Char(1)	Immediately after taking the snapshot for the interval Y or N if the job was in a CPU queueing state
QBCKTB	Queueing bucket before sample	Binary(4)	
BLOCKBCKT	Current blocking bucket (if blocked)	Binary(4)	
WOBJB	Wait object before sample	Char(3)	
LICWO	LIC 'wait object'	Char(3)	
WOBJAB	Wait object address before sample	Char(16)	
LICWOHNDL	LIC 'wait object' handle	Hex(16)	
CYCB	Cycles used before sample	Packed(11)	
CYCA	Cycles used after sample	Packed(11)	

SWINB	If active - timebase when switched in (before sample)	Packed(11)	
SWINA	If active - timebase when switched in (after)	Packed(11)	
WAITB	If blocked - timebase when blocked (before)	Packed(11)	
WAITA	If blocked - timebase when blocked (after)	Packed(11)	
CPUQB	If CPU queued - timebase when queued (before)	Packed(11)	
CPUQA	If CPU queued - timebase when queued (after)	Packed(11)	
WOSEGTYP	Wait object segment type (hex)	Hex(4)	
WOBASSEG	Wait object base segment address (hex)	Hex(16)	
WOOBJTYP	Wait object object type (hex)	Hex(4)	
WOOBJNAM	Wait object name	Char(30)	The name of the object this thread/task was waiting on when this interval was collected.
WOOBJTYPD	Wait object object type description	Char(35)	
WOSEGTYPD	Wait object segment type description	Char(35)	
HTASKCNT	Holding thread/task task count	Packed(11)	Identifies the thread or task the current thread/task is held by
HTHREADID	Holding thread ID	Hex(16)	
HTYPE	Holding thread/task type	Char(1)	
HTASKTDEID	Holding thread/task TDE ID	Hex(8)	
HTASKNAME	Holding thread/task name	Char(30)	

QENUMB	Queueing enum before sample	Binary(4)	Identifies the current wait type for this thread/task Immediately before taking the snapshot for this interval.
BLOCKENUM	Current blocking enum (if blocked)	Binary(4)	Enums are a specific type of wait within the wait buckets. See the enum description file QPYRTJWAT2.
CURRWTDUR	Total time in current wait (usecs)	Packed(11)	in microseconds
CURRUP	Current user profile	Char(10)	
RECCNFLCT	Oridinal record number if in DB record lock conflict	Packed(11)	
SLUSRSTK	Single level store user stack	Packed(11)	
TSUSRSTK	Terraspace user stack	Packed(11)	Not used
SLSSYSSTK	Single level store system stack	Packed(11)	
TSSYSSTK	Terraspace system stack	Packed(11)	Not used
JOBTYPE	Job type	Char(1)	Identifies the type of job (e.g. B = batch job)
DFTSOCKD	Default socket descriptor	Binary(4)	
LISSOCKD	Listen socket descriptor	Binary(4)	
DFTSOCKTOD	Default socket time of day	Timestamp	
DFTSOCKH	Default socket handle	Hex(16)	
LISSOCKTOD	Listen socket time of day	Timestamp	
LISSOCKH	Listen socket handle	Hex(16)	
PROCHANDLE	Process/job handle	Hex(16)	Uniquely identifies the process or job

ALLOCATEDT	Total allocated DASD pages since thread/task creation	Packed(11)	
DEALLOCEDT	Total deallocated DASD pages since thread/task creation	Packed(11)	

Table of Contents Previous Next

1.2.2 QPYRTJWAAI - Activation groups

Description: This file contains detailed activation group information.

Optional: Yes

Job Watch: Yes

Record: One record is created per process per activation group per program in the activation group.

Field name	Field Description	Format	Comments
INTERVAL	Interval number	Binary(4)	
TASKCOUNT	Thread/task task count	Packed(11)	
NAME	Job/task name	Char(32)	Fully qualified job name or task name
ACTGRP	Name of the activation group	Char(30)	
AGROOTNAME	Activation group root program name	Char(30)	The name of the program that caused the activation group to be created or blank if this record is for the default activation group
AGROOTLIB	Activation group root library name	Char(10)	
AGROOTTYPE	Activation group root program type	Char(8)	*PGM *SRVPGM
AGTYPE	Activation group type	Char(2)	

AGSTATE	Activation group state	Char(2)	01 = User state 02 = System state
AGSTGMODL	Activation group storage model	Char(2)	
AGSHARED	Activation group shared flag	Char(2)	00 = Not shared 01 = Shared
AGMARK	Activation group mark	Packed(11)	
AGDFTHSIZ	Activation group default heap size	Packed(11)	
AGDFTHBLKS	Activation group default heap blocks	Packed(6)	
AGOTHERHS	Activation group other heaps total size	Packed(6)	
PACTNAME	Program activation program name	Char(30)	
PACTPGMTYP	Program activation program type	Char(8)	
PACTLIB	Program activation program library name	Char(10)	
PACTLICTYP	Program activation LIC activation type	Char(2)	
PACTFRAMES	Program activation static frames count	Packed(6)	
PACTFRAMSZ	Program activation total static frame size	Packed(11)	
AGOTID	Activation group other heaps - heap ID	Binary(4)	
AGOTSIZE	Activation group other heaps - total heap size	Binary(4)	
AGOTBLKS	Activation group other heaps - heap total block count	Binary(4)	
PROCHANDLE	Process/Job handle	Hex(16)	Uniquely identifies the process or job

Table of Contents Previous Next

1.2.3 QPYRTJWAC1 - Communications/socket statistics

Description: This file contains detailed socket/TCP information for the job's collected in the Job Watch.

Optional: Yes

Job Watch: Yes

Record: One record is created per process per socket for each interval collected.

Field name	Field Description	Format	Comments
INTERVAL	Interval number	Binary(4)	
KEY	Key into QPYRTJWAC2 file	Binary(4)	Join this field with field KEY in file QPYRTJWAC2 in order to determine the fully qualified job name this socket belongs to.
TASKCOUNT	Job/task task count	Packed(11)	Uniquely identifies the thread/task running on the system.
FD	File descriptor number	Binary(4)	
TOD	Socket creation date/time	Timestamp	
SOCKHAND	Socket handle	Hex(16)	16 byte HEX address identifying the socket
SOCKDESCT	Socket descriptor number	Binary(4)	

SOCKFAM	Socket family	Char(13)	
SOCKTYPE	Socket type	Char(9)	
SOCKSTATE	Socket state	Char(13)	
SOCKLCLADR	Socket local address	Char(46)	
SOCKRMTADR	Socket remote address	Char(46)	
SOCKLCLPOR	Socket local port	Packed(3)	
SOCKRMTPOR	Socket remote port	Packed(3)	
TCPSTATE	TCP state	Char(11)	
TCPTPIST	TCP TPI state	Char(25)	
TCPUPROF	TCP user profile	Char(10)	
TCPSRCADR	TCP source address	Char(46)	
TCPDSTADR	TCP destination address	Char(46)	
TCPSRCPT	TCP source port	Packed(3)	
TCPDSTPT	TCP destination port	Packed(3)	
TCPSRBUF	TCP socket receiver buffer	Binary(4)	
TCPRBUF	TCP receive buffer	Binary(4)	
TCPSSBUF	TCP socket send buffer	Binary(4)	
TCPSBUF	TCP send buffer	Binary(4)	
TCPSQLEN	TCP send queue length	Binary(4)	
TCPSUNA	TCP suna	Packed(6)	
TCPSNEXT	TCP send next	Packed(6)	
TCPRQLN	TCP receive queue length	Packed(6)	

TCPSEQNUM	TCP sequence number	Packed(6)	
TCPACKNUM	TCP ack number	Packed(6)	
TCPMAXBLOG	TCP max backlog	Packed(6)	
TCPCURBLOG	TCP current backlog	Packed(6)	
SOCKULAL	Socket unix local address length	Binary(4)	
SOCKULA	Socket unix local address	Graphic(126)	
SOCKURAL	Socket unix remote address length	Binary(4)	
SOCKURA	Socket unix remote address	Graphic(126)	
SOCKNLIPA	Socket NETBIOS local IP address	Char(15)	
SOCKNLIF	Socket NETBIOS local IF	Binary(4)	
SOCKNLPT	Socket NETBIOS local port	Packed(6)	
SOCKNLNM	Socket NETBIOS local name	Char(16)	
SOCKNRIPA	Socket NETBIODS remote IP address	Char(15)	
SOCKNRIF	Socket NETBIOS remote IF	Binary(4)	
SOCKNRPT	Socket NETBIOS remote port	Packed(6)	
SOCKNRNM	Socket NETBIOS remote name	Char(16)	
SOCKKA	Socket keep alive	Char(1)	
SOCKFC	Socket flow control	Char(1)	
SOCKEOF	Socket end of file	Char(1)	
SOCKLINGO	Socket linger on/off	Char(1)	
SOCKNONBLK	Socket non-blocking	Char(1)	
SOCKLING	Socket linger	Binary(4)	

SOCKERR	Socket error	Char(15)	
SOCKRBUF	Socket receive buffer	Binary(4)	
SOCKRLWAT	Socket receive lowat	Binary(4)	
SOCKSBUF	Socket send buffer	Binary(4)	
SOCKSLEN	Socket data length	Binary(4)	
PROCHANDLE	Process/Job handle	Hex(16)	Uniquely identifies the process or job
DELTAUS	Delta usecs	Packed(11)	Elapsed interval time (in microseconds)
TCPFLAG1	TCP flag 1	Hex(2)	
TCPFLAG2	TCP flag 2	Hex(2)	
TCPFLAG3	TCP flag 3	Hex(2)	
TCPFLAG4	TCP flag 4	Hex(2)	
TCPFLAG5	TCP flag 5	Hex(2)	
TCPFLAG6	TCP flag 6	Hex(2)	
TCPFLAG7	TCP flag 7	Hex(2)	
TCPFLAG1	TCP flag 8	Hex(2)	
TCPBINDID	TCP bindID	Hex(2)	
TCPLNGSYN	TCP linger sync	Hex(2)	
TCPFN2	TCP FN2 sync	Hex(2)	
TCPRXMTSYN	TCP rxmt sync	Hex(2)	
TCPPROBSYN	TCP probe sync	Hex(2)	
TCPDALYSYN	TCP dally sync	Hex(2)	

TCPKEEPSYN	TCP keep-alive sync	Hex(2)	
TCPDUPSYN	TCP DUP sync	Hex(2)	
TCPCWND	TCP congestion window	Packed(6)	
TCPAPPSBYT	Application send bytes to TCP	Packed(6)	
TCPRMTRWIN	TCP remote receive window	Packed(6)	
TCPSENDMAX	TCP send max	Packed(6)	
TCPLCLRWIN	TCP local receive window	Packed(6)	
TCPCURRXMT	TCP current re-transmit	Packed(6)	
TCPRXMTCNT	TCP re-transmit count	Packed(3)	
TCPRXMTTOT	TCP re-transmit total	Packed(3)	
TCPRXMTFST	TCP fast re-transmit	Packed(3)	
TCPLASTACK	TCP last ack	Packed(6)	
TCPDBS	TCP delta bytes sent	Packed(6)	
TCPDBSPS	TCP delta bytes sent per second	Packed(6)	
TCPDBR	TCP delta bytes received	Packed(6)	
TCPDBRPS	TCP delta bytes received per second	Packed(6)	
TCPDBA	TCP delta bytes acknowledged	Packed(6)	
TCPCWNDA	TCP congestion alert	Char(1)	
TCPCWNDCSQ	TCP consequitive congestion decreases	Binary(4)	
TCPDSQL	TCP delta send queue length	Packed(6)	
TCPDSQLA	TCP delta send queue length alert	Char(1)	
TCPAPPSBYD	Delta application send to TCP	Packed(6)	

TCPAPPSBYA	Application send bytes to TCP alert	Char(1)	
TCPRMTRWIA	TCP remote receive window alert	Char(1)	
TCPRQLEND	Delta TCP receive queue length	Packed(6)	
TCPLCLRWID	TCP local receive window alert	Char(1)	
TCPCURBLOD	Delta TCP current backlog	Packed(6)	
SOCKLCLAD4	Socket local IP4 address	Char(15)	
SOCKRMTAD4	Socket remote IP4 address	Char(15)	
TCPSRCAD4	TCP source IP4 address	Char(15)	
TCPDSTAD4	TCP destination IP4 address	Char(15)	

Table of Contents | Previous | Next

1.2.4 QPYRTJWAC2 - Communications/socket job information

Description: This file contains the fully qualified job information for sockets listed in file QPYRTJWAC1.

Optional: Yes

Job Watch: Yes

Record: One record is created per job containing a unique socket per interval.

Field name	Field Description	Format	Comments
INTERVAL	Interval number	Binary(4)	
KEY	Unique identifier for each socket	Binary(4)	This field can be used in a JOIN query with field KEY in file QPYRTJWAC1
NAME	Job or task using socket	Char(32)	Fully qualified job name or task name using the socket associated with the key value for the same interval in file QPYRTJWAC1.

Table of Contents Previous Next

1.2.5 QPYRTJWAD - Wait bucket descriptions

Description: This file contains the wait bucket descriptions as they were set when the job watch was created.

Optional: No

Job Watch: Yes

Record: One record is created per wait bucket.

Field name	Field Description	Format
BUCKETNUM	Bucket number	Packed(3)
BUCKETDESC	Description for this bucket	Char(32)

Wait buckets for release V5R2:

Bucket number	Description	Comments
1	CPU	Using the CPU
2	Total wait	This is the SUM of values in buckets 4 - 32.
3	CPU queueing	The thread/task is ready and waiting to use the CPU
4	Reserved	Not used
5	DASD (other)	Rarely seen in production. The wait points in this bucket deal with DASD configuration and setup
6	Other waits	This category contains wait types that are typically not of concern
7	Mutex/semaphore	Wait points used by C/C++ programs
8	Journal	

9	Seizes	
10	DB record locks	
11	Object locks	
12	DASD space contention	
13	DASD reads	
14	DASD writes	
15	DASD reads or writes	Rarely occur
16	Mainstore pool overcommitment	Indicates dasd reads and page faults are delayed in order to clear new space for operations to complete
17	Java wait	
18	Java lock	Java object and method serialization waits
19	Java other	
20	Comm/socket accept	Normally an idle wait
21	Comm/socket transmit	
22	Comm/socket receive	
23	Comm/socket other	Indicate an application using the SELECT socket API
24	IFS pipe waits	
25	IFS (general)	
26	Keythink/MI response queue	
27	Data queue	
28	Other MI queue	
29	Event wait	
30	Reserved	

31	Severe DASD op-start contention	Occurs when a DASD operation is delayed due to a very high rate of concurrent DASD operations in progress at the moment it is requested
32	Severe contention	Reflect a high rate of concurrent waits/releases occurring against a wide variety of many of the other wait points listed previously

Table of Contents | Previous | Next

1.2.6 QPYRTJWAJ1 - Java threads waiting

Description: This file contains information about Java threads in a wait state for the set of jobs being watched.

Optional: No

Job Watch: Yes

Record: One record is created per process per java thread in a wait for each interval collected.

Field name	Field Description	Format	Comments
INTERVAL	Interval number	Binary(4)	
TASKCOUNT	Job/task task count	Packed(11)	Uniquely identifies the thread/task running on the system.
THDNAMEL	Java thread name length	Binary(4)	The total length of the thread name for this interval. Only the 1st 62 bytes are stored in field THDNAME.
THDNAME	Java thread name	Graphic(124)	1st 62 chars only
ENUM	Current wait's enum	Binary(4)	The current wait point that this thread is experiencing. Enums are identified in file QPYRTJWAT2

LICWO	Current LIC wait object	Char(3)	Low level Licensed Internal Code wait object identifier.
LICWOHNDL	Current LIC wait object handle	Hex(16)	
CURRWTDUR	Total time in current wait (usecs)	Packed(11)	in microseconds
JVAOBJNL	Last waited on java object name length	Binary(4)	The total length of the java object name for this interval. Only the 1st 225 bytes are stored in field JVAOBJN.
JVAOBJN	Last waited on java object name	Char(225)	
PROCHANDLE	Process/Job handle	Hex(16)	Uniquely identifies the process or job

Table of Contents | Previous | Next

1.2.7 QPYRTJWAJ2 - Java virtual machine statistics

Description: This file contains detailed information about the Java virtual machine for each job running a Java application in the collection.

Optional: No

Job Watch: Yes

Record: One record is created per process per JVM found for each interval collected.

Field name	Field Description	Format	Comments
INTERVAL	Interval number	Binary(4)	
TASKCOUNT	Job/task task count	Packed(11)	Uniquely identifies the thread/task running on the system.
PROCHANDLE	Process/Job handle	Hex(16)	Uniquely identifies the process or job
JVMHANDLE	Java virtual machine handle	Hex(16)	
GCHEAPSZ	Garbage collector heap size	Packed(11)	in bytes
GCSWEEPS	Garbage collector sweep count	Binary(4)	
GCSTAGE	Garbage collector current stage	Char(30)	
GCINITSIZE	Garbage collector initial size	Packed(11)	in bytes
GCMAXSIZE	Garbage collector maximum size	Packed(11)	in bytes
SOFTREFS	Garbage collector soft references processed	Packed(11)	

CLEAREDREF	Garbage collector soft references cleared	Packed(11)	
SWEEPS	Garbage collector sweep cycle	Packed(11)	
COL_TYPE	Garbage collector collection type	Char(20)	
WEAKREFS	Garbage collector weak references processed	Packed(11)	
FINALREFS	Garbage collector finalizer references processed	Packed(11)	
PHNTOMREFS	Garbage collector phantom references processed	Packed(11)	
GLOBALWEAK	Garbage collector global weak references processed	Packed(11)	
CLRDSFTREF	Garbage collector cleared soft references processed	Packed(11)	
CLRDWKREF	Garbage collector cleared weak references	Packed(11)	
CLRDFNLREF	Garbage collector cleared finalizer references	Packed (11)	
CLRDPHREF	Garbage collector cleared phantom references	Packed(11)	
CLRDGBLREF	Garbage collector cleared global references	Packed(11)	
CYCLEDURAT	Garbage collector last cycle duration	Packed(11)	in milliseconds
SECGCTHRDS	Garbage collector secondary threads	Packed(6)	
ALCHPFND	Garbage collector allocated heap space found	Packed(11)	in bytes
ALCHPPRV	Garbage collector allocated heap space found previous cycle	Packed(11)	in bytes
ALCHPTCYCL	Garbage collector allocated heap this cycle	Packed(11)	in bytes

ALCHPPCYCL	Garbage collector allocated heap previous cycle	Packed(11)	in bytes
GC_LIVE	Garbage collector active object count	Packed(11)	
GC_DEAD	Garbage collector collected object count	Packed(11)	
GCCURTHRH	Garbage collector current threshold	Packed(11)	
HPSIZESTRT	Garbage collector heap size start	Packed(11)	in bytes
HPSIZEEND	Garbage collector heap size end	Packed(11)	in bytes
HPFREESTRT	Garbage collector heap free start	Packed(11)	in bytes
HPFREEEND	Garbage collector heap free end	Packed(11)	in bytes
COLTIMESTR	Garbage collector collection start date/time	Timestamp	
GCCYCLENBR	Garbage collector cycle number	Packed(6)	
CYCLRUNTIM	Garbage collector cycle run time	Packed(6)	in milliseconds
OLDMARKSET	Garbage collector empty old mark set	Packed(6)	in milliseconds
SYNC1DUR	Garbage collector handshake threads	Packed(6)	in milliseconds
SYNC12DUR	Garbage collector transition 1 -> 2	Packed(6)	in milliseconds
SYNC2ASDUR	Garbage collector transition 2 -> async	Packed(6)	in milliseconds
ASYNCDUR	Garbage collector async handshake duration	Packed(6)	in milliseconds
ASYNFINDUR	Garbage collector asyunc finish trace duration	Packed(6)	in milliseconds
GLOBALDUR	Garbage collector process objects in global registry	Packed(6)	in milliseconds
FINTRCDUR	Garbage collector finish trace duration	Packed(6)	in milliseconds
REFOBJDUR	Garbage collector reference object duration	Packed(6)	in milliseconds

SWEEPDUR	Garbage collector sweep duration	Packed(6)	in milliseconds

Table of Contents | Previous | Next

1.2.8 QPYRTJWAL - Basic job statistics snapshots

Description: Job statistics snapshot file. This file is created by the WCHJOB command with the *GETTDELIST or *GETTASKLIST option. It contains performance statistics for all jobs/threads/tasks on the system at the time the command was executed. This file is not part of a normal job watch. Unlike file QPYRTJWA most of the fields in this file are total counts per thread/task rather than a delta for each interval.

Optional: Yes via green screen use only

Job Watch: No

Record: One record is created per active thread/task on the system.

Field name	Field Description	Format	Comments
NAME	Job/task name	Char(32)	Fully qualified job name or task name
CURRUP	Current user profile	Char(10)	

TYPE	Job or task type flag	Char(1)	T = Task
			P = Job/Thread (Process)

THREADID	Thread ID (hex)	Hex(16)	
INITIAL	Initial thread flag	Char(1)	
TASKCOUNT	Job/task task count	Packed(11)	Uniquely identifies the thread/task running on the system.

TCOUNTH	Thread/task task count (in hex)	Hex(16)	Taskcount value in hex format
TDEID	Task dispatching element identifying the thread/task (in hex)	Hex(8)	
SYNDBRD	Synchronous DB reads	Binary(4)	
SYNNDBRD	Synchronous non-DB reads	Binary(4)	
SYNDBWRT	Synchronous DB writes	Binary(4)	
SYNNDBWRT	Synchronous non-DB writes	Binary(4)	
ASYDBRD	Asynchronous DB reads	Binary(4)	
ASYNDBRD	Asynchronous non-DB reads	Binary(4)	
ASYDBWRT	Asynchronous DB writes	Binary(4)	
ASYNDBWRT	Asynchronous non-DB writes	Binary(4)	
IOPENDING	IO pending page faults	Binary(4)	
SMSYNCIO	Waits for asynchronous writes	Binary(4)	
FLTS	Page faults resulting in DASD reads	Binary(4)	
ALLOCATED	Allocated DASD pages	Binary(4)	
DEALLOCED	Deallocated DASD pages	Binary(4)	
BLOCKENUM	Current blocking enum (if blocked)	Binary(4)	Enums are a specific type of wait within the wait buckets. See the enum description file QPYRTJWAT2.
BLOCKBCKT	Current blocking bucket (if blocked)	Binary(4)	
PRIORITY	Current LIC priority	Binary(2)	

CPU	CPU used (usecs)	Packed(11)	in microseconds
JOBTYPE	Job type	Char(1)	Identifies the type of job (e.g. B = batch job)
JOBSTATUS	Job status	Char(4)	
JOBFUNC	Job function	Char(14)	
JOBSBS	Job subsystem	Char(10)	
CURRWTDUR	Total time in current wait (usecs)	Packed(11)	in microseconds
IPLUSECS	Time since the last IPL when this snapshot was taken	Packed(11)	in microseconds
QCOUNT01-QCOUNT32	Queueing bucket 01-32 count	Binary(4)	Total occurrences of each category of wait
QTIME01-QTIME32	Queueing bucket 01-32 time	Packed(11)	Total time (in microsconds) for each category of wait
HTASKCNT	Holding thread/task task count	Packed(11)	Identifies the thread or task the current thread/task is held by
HTHREADID	Holding thread ID	Hex(16)	
HTASKTDEID	Holding thread/task TDE ID	Hex(8)	
HTASKNAME	Holding thread/task name	Char(30)	
WOSEGTYP	Wait object segment type (hex)	Hex(4)	
WOOBJTYP	Wait object object type (hex)	Hex(4)	

WOOBJNAM	Wait object name	Char(30)	The name of the object this thread/task was waiting on when this snapshot was collected.
RECCNFLCT	Oridinal record number if in DB record lock conflict	Packed(11)	
LICWO	LIC 'wait object'	Char(3)	
LICWOHNDL	LIC 'wait object' handle	Hex(16)	
LDIOWRT	LDIO writes	Binary(4)	
LDIORD	LDIO reads	Binary(4)	
LDIOOTHR	LDIO other non reads/writes	Binary(4)	
CMNWRT	Communication file writes	Binary(4)	
CMNRD	Communication file reads	Binary(4)	
LDIOUPD	LDIO updates	Binary(4)	Not used
LDIODEL	LDIO deletes	Binary(4)	Not used
LDIOFEOD	LDIO feods	Binary(4)	Not used
LDIOCOMIT	LDIO commits	Binary(4)	Not used
LDIOROLLB	LDIO rollbacks	Binary(4)	Not used
LDIOOPEN	LDIO opens	Binary(4)	Not used
LDIOCLOSE	LDIO closes	Binary(4)	Not used
LDIOIXBLD	LDIO index builds	Binary(4)	Not used
LDIOSORT	LDIO sorts	Binary(4)	Not used
LDTAQSND	Data queue sends	Binary(4)	Not used

LDTAQRCV	Data queue receives	Binary(4)	Not used
LUSRSPCIOP	User space index/ops	Binary(4)	Not used
TXAPPIQT	Transaction application input queueing time	Packed(11)	in microseconds
TXRSCUT	Transaction resource usage time	Packed(11)	in microseconds
TXDSPLRT	Transaction I/O response time	Packed(11)	in microseconds
TXINQTRAN	Transaction input queueing transaction time	Binary(4)	in microseconds
TXRSCUTRAN	Transaction resource usage transaction time	Binary(4)	in microseconds
TXDSPLTRAN	Transaction display I/O transaction time	Binary(4)	in microseconds
IFSSYMLRD	IFS symbolic link reads	Binary(4)	
IFSDIRRD	IFS directory reads	Binary(4)	
IFSLUCHIT	IFS lookup cache hits	Binary(4)	
IFSLUCMIS	IFS lookup cache misses	Binary(4)	
IFSOPENS	IFS opens	Binary(4)	
IFSDIRCRT	IFS directory creates	Binary(4)	
IFSNDIRCRT	IFS non-directory creates	Binary(4)	
IFSDIRDLT	IFS directory deletes	Binary(4)	
IFSNDIRDLT	IFS non-directory deletes	Binary(4)	
ORIGPRI	Original priority	Packed(2)	
POOL	Pool ID	Packed(2)	
ACTWAIT	Active to wait transitions	Binary(4)	
WAITINEL	Wait to ineligible transitions	Binary(4)	

ACTINEL	Active to ineligible transitions	Binary(4)	

Table of Contents Previous Next

1.2.9 QPYRTJWALQ - System-wide SQL statements

Description: SQL statements snapshot file. This file is created by the WCHJOB command with the *GETTDELIST or *GETTASKLIST option. It contains SQL statements that were running at the time the snapshot was performed.

Optional: Yes via green screen use only

Job Watch: No

Record: One record is created per SQL statement running at the time the snapshot was taken.

Field name	Field Description	Format	Comments
TASKCOUNT	Job/task task count	Packed(11)	Uniquely identifies the thread/task running on the system.
SRCLIB	SQL package source library	Char(10)	
SRCFILE	SQL package source file	Char(10)	
SRCMBR	SQL package source file member	Char(10)	
SRCDATE	SQL package source last changed date	Char(13)	
PKGLIB	SQL package library/container name	Char(18)	
PKGNAME	SQL package name	Char(18)	
RDBSNAME	Remote DBS name	Char(18)	

HOSTREAL	Actual number of host variables	Binary(4)	Number of host variables associated with this SQL statement
MORE	Another statement associated with this statement	Char(1)	1 = another statement associated with this one
CCSID	CCSID of dynamic statement	Binary(4)	
SQLSTMT	SQL statement	Varchar(300)	

Table of Contents Previous Next

1.2.10 QPYRTJWALS - System-wide call stacks

Description: Call stacks snapshot file. This file is created by the WCHJOB command with the *GETTDELIST or *GETTASKLIST option. It contains call stacks for jobs having the performance conditions indicated by the command.

Optional: Yes via green screen use only

Job Watch: No

Record: One record is created per call stack collected based on the performance statitics specified on the WCHJOB command.

Field name	**Field Description**	**Format**	**Comments**
TASKCOUNT	Job/task task count	Packed(11)	Uniquely identifies the thread/task running on the system.
REASON	Reason code	Binary(4)	Code indicating the reason the call stack was collected. 1 = Holder 2 = Waiter (and there will be a holder) 3 = Blocked (and there is no holder reported) 4 = Faulting 5 = Running
USECSIPL	Time since last IPL	Packed(11)	in microseconds
STACK	Call stack contents	Varchar(300)	

Table of Contents Previous Next

1.2.11 QPYRTJWAMT - Missed threads

Description: A job-specific Job Watch has a built in limit that no more than 500 threads may be watched over the 10 jobs selected in the Job Watch. If over the life of the Job Watch the limit is exceeded for any interval, information about the number of threads not collected will go into this file.

Optional: No

Job Watch: Yes

Record: A record is created only in special circumstances (see description)

Field name	Field Description	Format	Comments
INTERVAL	Interval number	Binary(4)	The interval number that threads were missed (not collected)
JOBNAME	Job name	Char(10)	The name of the job where one or more of its threads were not collected
JOBNUM	Job number	Char(6)	
MISSEDTHD	Missed thread count	Binary(4)	The total number of threads not collected for this job, this interval.

Table of Contents Previous Next

1.2.12 QPYRTJWAPR - Process/job based statistics

Description: The per-process (or job info) file. This file contains a record per job (not thread) being watched. It contains the statistics that cannot be captured at the thread level. The information in this file includes IFS, transactions and LDIO statistics.

Optional: No

Job Watch: Yes

Record: One record is created per active job per interval.

Field name	Field Description	Format	Comments
INTERVAL	Interval number	Binary(4)	
PROCHANDLE	Process/job handle	Hex(16)	Uniquely identifies the process or job
TASKCOUNT	Job/task task count	Packed(11)	Uniquely identifies the thread/task running on the system.
LDIOWRT	LDIO writes	Binary(4)	
LDIORD	LDIO reads	Binary(4)	
LDIOOTHR	LDIO other non reads/writes	Binary(4)	
CMNWRT	Communication file writes	Binary(4)	
CMNRD	Communication file reads	Binary(4)	
LDIOUPD	LDIO updates	Binary(4)	Not used
LDIODEL	LDIO deletes	Binary(4)	Not used

LDIOFEOD	LDIO feods	Binary(4)	Not used
LDIOCOMIT	LDIO commits	Binary(4)	Not used
LDIOROLLB	LDIO rollbacks	Binary(4)	Not used
LDIOOPEN	LDIO opens	Binary(4)	Not used
LDIOCLOSE	LDIO closes	Binary(4)	Not used
LDIOIXBLD	LDIO index builds	Binary(4)	Not used
LDIOSORT	LDIO sorts	Binary(4)	Not used
LDTAQSND	Data queue sends	Binary(4)	Not used
LDTAQRCV	Data queue receives	Binary(4)	Not used
LUSRSPCIOP	User space index/ops	Binary(4)	Not used
TXAPPIQT	Transaction application input queueing time	Packed(11)	in microseconds
TXRSCUT	Transaction resource usage time	Packed(11)	in microseconds
TXDSPLRT	Transaction I/O response time	Packed(11)	in microseconds
TXINQTRAN	Transaction input queueing transaction time	Binary(4)	in microseconds
TXRSCUTRAN	Transaction resource usage transaction time	Binary(4)	in microseconds
TXDSPLTRAN	Transaction display I/O transaction time	Binary(4)	in microseconds
IFSSYMLRD	IFS symbolic link reads	Binary(4)	
IFSDIRRD	IFS directory reads	Binary(4)	
IFSLUCHIT	IFS lookup cache hits	Binary(4)	
IFSLUCMIS	IFS lookup cache misses	Binary(4)	
IFSOPENS	IFS opens	Binary(4)	

IFSDIRCRT	IFS directory creates	Binary(4)	
IFSNDIRCRT	IFS non-directory creates	Binary(4)	
IFSDIRDLT	IFS directory deletes	Binary(4)	
IFSNDIRDLT	IFS non-directory deletes	Binary(4)	

Table of Contents Previous Next

1.2.13 QPYRTJWARI - Run information

Description: This file contains basic information about each collection that has been ran or is running in a library. This is a single member file used by the GUI to the display the list of job-specific job watches in a library.

Optional: No

Job Watch: No

Record: One record is created per collection. If a new collection is created with the same name for the current library, the existing record is overwritten.

Field name	Field Description	Format	Comments
COLLNAME	Collection name	Char(10)	The name of the job watch
USRSPCN	Userspace name	Char(10)	The name of the userspace in the current library used by the GUI. This userspace contains status information for actively running job watches.
JOB	Collecting job name	Char(28)	Fully qualified job name of the job that produced (or is currently running) the collection.

SYSOSVRM	Collecting system OS VRM	Char(3)	Identifies the OS/400 release of the system this Job Watch was created on. This value is 520 on a V5R2 system.
IDOCVRM	Job Watcher VRM	Char(3)	Identifies the version of Job Watcher that was used to create this collection. This value is 130 on a V5R2 system.

Table of Contents | Previous | Next

1.2.14 QPYRTJWASD - SQL statement host variables

Description: This file contains host variables for SQL statements collected in file QPYRTJWASQ.

Optional: Yes

Job Watch: Yes

Record: One record is created per host variable per statement per interval collected.

Field name	Field Description	Format	Comments
DATAKEY	Matches DATAKEY field in file QPYRTJWASQ	Packed(6)	Used to join with file QPYRTJWASQ
DATANUM	Host variable number	Binary(4)	An index of this host variable within the SQL statement it belongs to
TYPE	Host variable type	Binary(4)	Identifies the type of host variable
DATA	Host variable data	Varchar(50)	The value for this host variable. The format of this field is dependent on the host variable type value.

Table of Contents | Previous | Next

1.2.15 QPYRTJWASQ - SQL statements

Description: The SQL statement file. This file includes the SQL statements captured over jobs in the Job Watch. This statement listed each interval is the last SQL statement executed for each job. It does not indicate that the statement is currently executing , or how long it took to execute. These statistics must be captured using a tool like DB monitor (DBMON).

Optional: Yes

Job Watch: Yes

Record: One record is created per SQL statement available per job per interval.

Field name	Field Description	Format	Comments
INTERVAL	Interval number	Binary(4)	
TASKCOUNT	Job/task task count	Packed(11)	Uniquely identifies the thread/task running on the system.
PROCHANDLE	Process/job handle	Hex(16)	Uniquely identifies the process or job
NAME	Job/task name	Char(32)	Fully qualified job name or task name
THREADID	Thread ID (hex)	Hex(16)	
DATAKEY	Matches DATAKEY field in file QPYRTJWASD	Packed(6)	Used to join with file QPYRTJWASD
SRCLIB	SQL package source library	Char(10)	
SRCFILE	SQL package source file	Char(10)	

SRCMBR	SQL package source file member	Char(10)	
SRCDATE	SQL package source last changed date	Char(13)	
PKGLIB	SQL package library/container name	Char(18)	
PKGNAME	SQL package name	Char(18)	
RDBSNAME	Remote DBS name	Char(18)	
HOSTREAL	Actual number of host variables	Binary(4)	Number of host variables associated with this SQL statement
HOSTLOGGED	Number of host variables captured in file QPYRTJWASD	Binary(4)	The number of host variables captured for this SQL statement. A maximum of 100 host variables can be captured.
MORE	Another statement associated with this statement	Char(1)	1 = another statement associated with this one
CCSID	CCSID of dynamic statement	Binary(4)	
SQLSTMT	SQL statement	Varchar(300)	

Table of Contents Previous Next

1.2.16 QPYRTJWAST - Call stacks

Description: The call stack file. This file contains the call stack for each thread being watched each interval. The data in the file is only optionally collected and must be turned on within the Start Job Watch Wizard or using the options on the WCHJOB command.

Optional: Yes

Job Watch: Yes

Record: One record is created per thread per interval collected.

Field name	Field Description	Format	Comments
INTERVAL	Interval number	Binary(4)	
TASKCOUNT	Job/task task count	Packed(11)	Uniquely identifies the thread/task running on the system.
NAME	Job/task name	Char(32)	Fully qualified job name or task name
THREADID	Thread ID (hex)	Hex(16)	
S01OFF - S50OFF	Stack procedure level (1-50) offset	Char(8)	Identifies the location of this procedure in memory. 50 fields are defined in this file for each call stack level.

S01PROC - S50PROC	Stack procedure name	Char(256)	50 fields are defined in this file for each call stack level.
S01MOD - S50MOD	Stack module name	Char(10)	50 fields are defined in this file for each call stack level.
S01PGM - S50PGM	Stack program name (ILE or OMI)	Char(10)	50 fields are defined in this file for each call stack level.

Table of Contents | Previous | Next

1.2.17 QPYRTJWAS3 - SQL cursors and prepared statements summary

Description: SQL open cursor lists and prepared statement array overview. This information is optionally collected and must be turned on using the SQL DETAIL *HIGH setting on the WCHJOB command.

Optional: Yes

Job Watch: Yes

Record: One record is created per job using SQL statements per interval collected.

Field name	Field Description	Format	Comments
INTERVAL	Interval number	Binary(4)	
PROCHANDLE	Process/job handle	Hex(16)	Uniquely identifies the process or job
OCLNUM	Open cursor list count	Binary(4)	
PSEDOCLOD	Number of pseudo closed dummies	Binary(4)	
FULLYOPN	Number of fully opened cursors	Binary(4)	
PSEDOCLO	Number of pseudo closed cursors	Binary(4)	
OLDPSEDOI	Oldest pseudo closed index	Binary(4)	
NEWPSEDOI	Newest pseudo closed index	Binary(4)	
OCLINUSE	Inuse open cursor count	Binary(4)	
PSAALLOC	Number of allocated prepared statement array slots	Binary(4)	

PSAINUSE	Number of prepared statement array slots in use	Binary(4)	
PSAFOUND	Number of prepared statement array entries found	Binary(4)	

Table of Contents Previous Next

1.2.18 QPYRTJWAS4 - SQL open cursors

Description: SQL open cursor list details. This information is optionally collected and must be turned on using the SQL DETAIL *HIGH setting on the WCHJOB command.

Optional: Yes

Job Watch: Yes

Record: One record is created per open cursor per job using SQL statements per interval collected.

Field name	Field Description	Format	Comments
INTERVAL	Interval number	Binary(4)	
PROCHANDLE	Process/job handle	Hex(16)	Uniquely identifies the process or job
LASTRC	Last return code	Binary(4)	
CURNAME	Cursor name	Char(18)	
PREPSTNAME	Prepared statement name	Char(18)	
FIRSTFILE	First file name	Char(10)	
FIRSTLIB	First file library name	Char(10)	
OCFLAGS	Open cursor flags	Char(10)	
FLGPSEOCLO	Pseudo closed flag	Char(1)	
FLGED	Extended dynamic fag	Char(1)	
FLGDRDARMT	DRDA remote flag	Char(1)	
FLGDYNI	Dynamic I-SQL flag	Char(1)	

FLGREOPEND	Reopened pseudo closed flag	Char(1)	
FLGREUSBLE	Reuseable flag	Char(1)	
FLGPRCED	QSQPRCED flag	Char(1)	
STMTNUM	OCL array statement index	Binary(2)	
UNQRFSH	UNQCT_RFSH	Binary(4)	
TRUSTMTLEN	True statement length	Binary(2)	The actual length of the statement listed in field STMTTXT
STMTTXT	First 256 bytes of SQL statement	Char(256)	

Table of Contents Previous Next

1.2.19 QPYRTJWAS5 - SQL prepared statement arrays

Description: SQL open cursor list details. This information is optionally collected and must be turned on using the SQL DETAIL *HIGH setting on the WCHJOB command.

Optional: Yes

Job Watch: Yes

Record: One record is created per open cursor per job using SQL statements per interval collected.

Field name	Field Description	Format	Comments
INTERVAL	Interval number	Binary(4)	
PROCHANDLE	Process/job handle	Hex(16)	Uniquely identifies the process or job
STMTINDX	Statement index in prepared statement array	Binary(4)	
PREPSTNAME	Prepared statement name	Char(18)	
RAWFLAGS	Raw flags	Char(8)	
LITCONV	Literal to parm marker done flag	Char(1)	
SHARED	Shared with another prepared statement array entry flag	Char(1)	
SYSCACHE	In system-wide cache flag	Char(1)	
OVRIDES	Overrides exist flag	Char(1)	
TOLDECERR	Tolerate decimal data errors flag	Char(1)	

NOTUSEDCMP	Not used since compression flag	Char(1)	
USECOUNT	Usage count	Packed(11)	
TRUSTMTLEN	True statement length	Binary(2)	The actual length of the statement listed in field STMTTXT
STMTTXT	First 256 bytes of SQL statement	Char(256)	

Table of Contents Previous Next

1.2.20 QPYRTJWAT2 - Enum (wait point) descriptions

Description: The enum description file. Each enum represents a very specific type of wait. This file contains the enum descriptions and their mappings into each of the 32 different wait buckets. These mappings are shown from when the Job Watch was started and is changeable. When changed it affects the entire system.

Optional: No

Job Watch: Yes

Record: One record is created per enum defined on the system at the moment the job watch was created.

Field name	**Field Description**	**Format**	**Comments**
BUCKETNUM	Bucket number	Packed(3)	
BUCKETDESC	Description for this bucket	Char(32)	
ENUM	Specific wait point ID	Packed(3)	
EYE	Eye catcher	Char(3)	This is the LIC wait object
DESC	Wait point description	Char(40)	

Table of Contents Previous Next

1.2.21 QPYRTJWA1T - First time thread/task file

Description: This file indicates the interval that each job/thread began collecting data (or become active). This is useful if you are watching jobs on a job queue and want a way to quickly see when they became active.

Optional: No

Job Watch: Yes

Record: One record is created per active thread/task per interval

Field name	Field Description	Format	Comments
NAME	Job/task name	Char(32)	Fully qualified job name or task name
JOBTYPE	Job type	Char(1)	Identifies the type of job (e.g. B = batch job)

LICTYPE	Job or task type flag	Char(1)	T = Task
			P = Job/Thread (Process)

THREADID	Thread ID (hex)	Hex(16)	
TASKCOUNT	Thread/task task count	Packed(11)	
STINTERVAL	Starting interval number	Binary(4)	The interval this thread/task was first collected
STTOD	Date/time this thread/task was added to the Job Watch	Timestamp	

PROCHANDLE	Process/job handle	Hex(16)	Uniquely identifies the process or job

Table of Contents Previous Next

1.2.22 QPYRTJWDFN - Job watch definitions

Description: This file stores the command used by the GUI to create this Job Watch. The file is only created if the Job Watch is submitted through the client.

Optional: No

Job Watch: Yes (if created via GUI)

Record: One record is created per collection.

Field name	**Field Description**	**Format**	**Comments**
NAME	Definition name	Char(10)	This is the Job Watch name
OSVRM	OS/400 version/release/mod level	Char(3)	At release V5R2 this value is 520
COMMAND	Command used to create this job watch	Char(1000)	
JOBINFO	Job information	Char(500)	Because the WCHJOB command does not support selecting by fully qualified job names, this string contains the fully qualified job names of any jobs selected to be included for a job-specific Job Watch.

TASKINFO	Task information	Char(500)	Contains the tasknames of any tasks selected to be included for a job-specific Job Watch

Table of Contents | Previous | Next

1.2.23 QPYRTJWAT - Totals for QPYRTJWA

Description: This file is created after a collection has finished running and the user selects to view the property pages for the job watch in the GUI.

Optional: Yes

Job Watch: Yes (but only created once the properties are viewed for a job watch within the GUI)

Record: This file will contain only 1 record after the properties are shown for the first time.

Field name	Field Description	Format	Comments
INTERVALS	Total intervals	Packed(3)	
TOTALTIME	Total elapsed collection time	Packed(16)	in microseconds
STARTTIME	Collection start time	Timestamp	time the collection gathered its first interval of data
ENDTIME	Collection end time	Timestamp	time the collection gathered its last interval of data
AVGINTDUR	Average interval duration	Packed(16)	the average size of each interval over all jobs in the collection
TOTCPU	Total CPU usecs	Packed(16)	
TOTWRT	Total DASD writes	Packed(16)	
TOTPRMWRT	Total DASD permanent writes	Packed(16)	
TOTSYNDBR	Total sync DB reads	Packed(16)	synchronous database reads

TOTSYNNDBR	Total sync NDB reads	Packed(16)	synchronous non-database reads
TOTSYNDBW	Total sync DB writes	Packed(16)	synchronous database writes
TOTSYNNDBW	Total sync NDB writes	Packed(16)	synchronous non-database writes
TOTASYDBR	Total async DB reads	Packed(16)	asynchronous database reads
TOTASYNDBR	Total async NDB reads	Packed(16)	asynchronous non-database reads
TOTASYDBW	Total async DB writes	Packed(16)	asynchronous database writes
TOTASYNDBW	Total async NDB writes	Packed(16)	asynchronous non-database writes
TOTIOPEND	Total IO pending page faults	Packed(16)	
TOTSYMSYNC	Total waits for asynchronous writes	Packed(16)	
TOTFLT	Total page faults resulting in DASD reads	Packed(16)	
TOTALLOCD	Total allocated DASD pages	Packed(16)	
TOTDALLOCD	Total deallocated DASD pages	Packed(16)	
TOTSEIZE	Total seize time (usecs)	Packed(16)	in microseconds
TOTMUTEX	Total mutex time (usecs)	Packed(16)	in microseconds
TOTBINOVER	Total binary overflows	Packed(16)	
TOTDECOVER	Total decimal overflows	Packed(16)	
TOTFLTOVER	Total float overflows	Packed(16)	
TOTSTMFRD	Total stream file reads	Packed(16)	

TOTSTMFWRT	Total stream file writes	Packed(16)	
TOTACTWAIT	Total active to wait	Packed(16)	
TOTWAITINL	Total wait to ineligible	Packed(16)	
TOTACTINL	Total active to ineligible	Packed(16)	
TOTSOCKR	Total socket reads	Packed(16)	
TOTSOCKW	Total socket writes	Packed(16)	
TOTSOCKBR	Total socket bytes read	Packed(16)	
TOTSOCKBW	Total socket bytes written	Packed(16)	

Table of Contents | Previous | Next

1.2.24 QPYRTJWSYS - System information

Description: This file is created at the start of every collection (for both system-wide and job-specific modes) and will contain basic system information and system values for the system that created the collection.

Optional: No

Job Watch: Yes

Record: This file will contain only 1 record

Field name	Field Description	Format	Comments
SYSTEM	System name	Char(8)	
QCCSID	CCSID	Binary(4)	
QCHRID	Character set and code page	Char(20)	
QCNTRYID	Country or region identifier	Char(2)	
QDYNPTYADJ	Dynamic priority adjustment	Char(1)	
QDYNPTYSCD	Dynamic priority scheduler	Char(1)	
QIGC	DBCS installed	Char(1)	
QMODEL	System model	Char(4)	
QPFRADJ	Performance adjustment	Char(1)	
QPRCMLTTSK	Processor multitasking	Char(1)	
QPRCFEAT	Processor feature code	Char(4)	
QQRYDEGREE	Parallel processing degree	Char(10)	
QQRYTIMLMT	Query processing time limit	Char(10)	

QSFWERRLOG	Software error logging	Char(10)	
QSHRMEMCTL	Shared memory control	Char(1)	
QSRLNBR	System serial number	Char(8)	
QTSEPOOL	Time-slice end pool	Char(10)	
QUSEADPAUT	Use adopted authority	Char(10)	
QSYSLIBL	System library list	Char(150)	
QUSRLIBL	User library list	Char(250)	
QPARTID	Parition ID	Binary(2)	
QPARTNUM	Number of partitions	Binary(2)	
QRSFLAG	Restricted state flag	Char(1)	

Table of Contents | Previous | Next

1.2.25 QPYRTJWRDB - Rule definitions

Description: This file contains a copy of the rule definition that was used at the start of the collection (if any). If a rule is not used this file/member will not be created.

Optional: No

Job Watch: Yes

Field name	Field Description	Format	Comments
SRCSEQ		Zoned(6)	
SRCDAT		Zoned(6)	
SRCDTA		Char(80)	See file QPYRTJWRD in library QPYRTJW for examples

📖 Table of Contents ◀ Previous ▶ Next

1.3 Commands

This section covers the green screen commands that are provided with Job Watcher for working with job-specific job watches.
These commands exist in the Job Watcher library QPYRTJW.

Table of Contents Previous Next

1.3.1 WCHJOB - creates a job watch

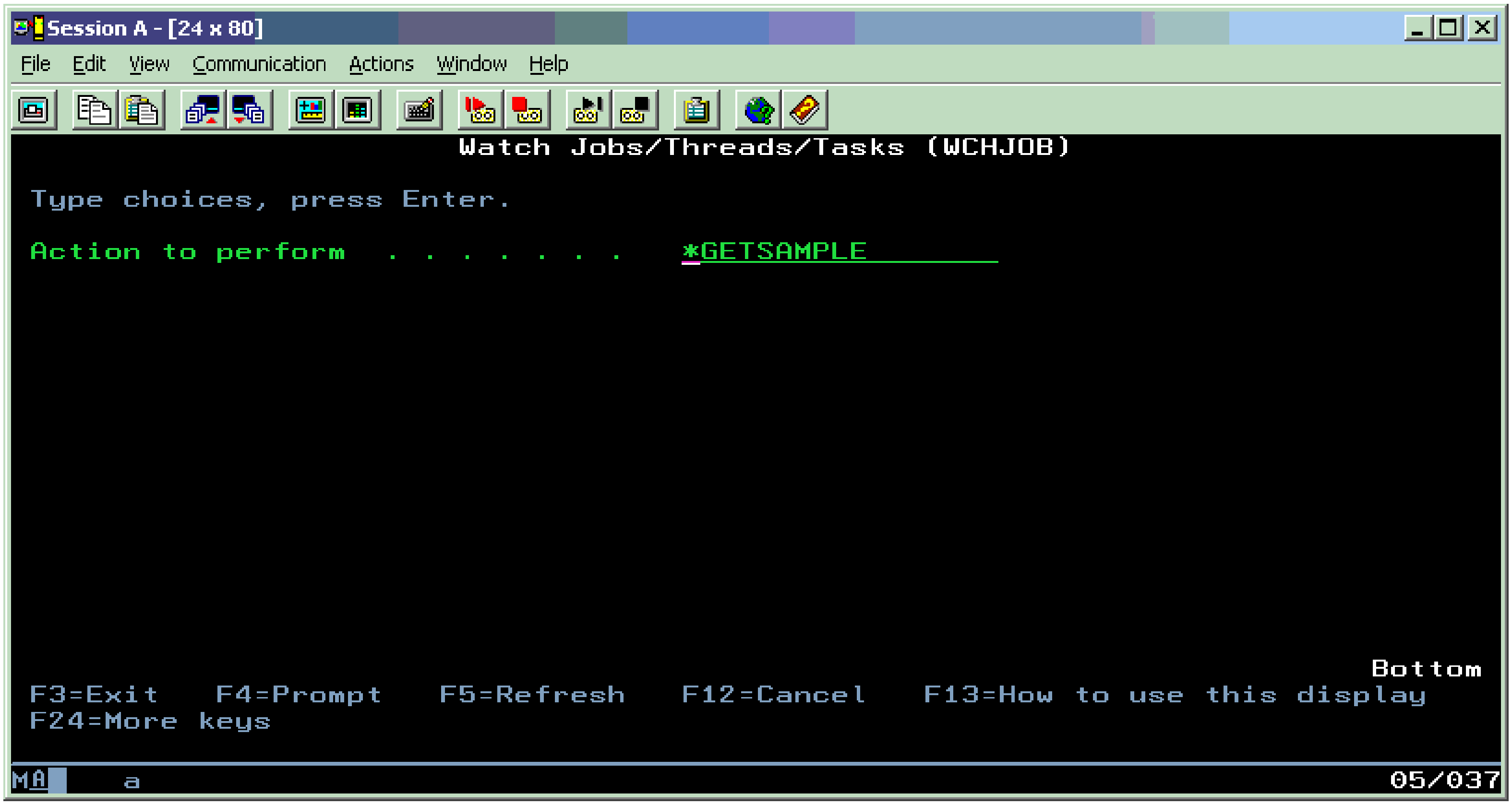
The Job Watcher GUI provides real time views of job/tasks and/or threads on a system. These views are based on data files created by the WCHJOB command in library QPYRTJW (the Job Watcher library). This command produces data for a job-specific Job Watch. The command may be used either via the GUI using the Job Watch Wizard or directly from a green screen command line. To produce a system-wide Job Watch use the WCHSYS command in the QPYRTJW library.

A job-specific Job Watch is a set of data files containing information about up to ten jobs/tasks and up to a maximum of 500 threads within these selected jobs.
The command for starting a job-specific Job Watch is:

QPYRTJW/WCHJOB

This command can perform one of the following actions based on the *Action to perform* (ACTION) parameter:

- Begin collecting data for a new job watch (*FIRSTTIME)
- Write an interval to the job watch files (*GETSAMPLE)
- Stop collecting data for the current job watch (*QUIT)
- Generate a snapshot list of jobs/tasks/threads into an output file. (*GETTDELIST)
- Generate a snapshot list of only LIC tasks into an output file. (*GETTASKLIST)
- Generate a snapshot list of only jobs/threads into an output file. (*GETTHREADLIST)
- Generate the wait code (bucket) description files. (*GETWAITCODES)
- Sets the LIC wait code buckets so they are useable by Job Watcher. This option must be ran once after every IPL before using Job Watcher. (*AFTERIPL)

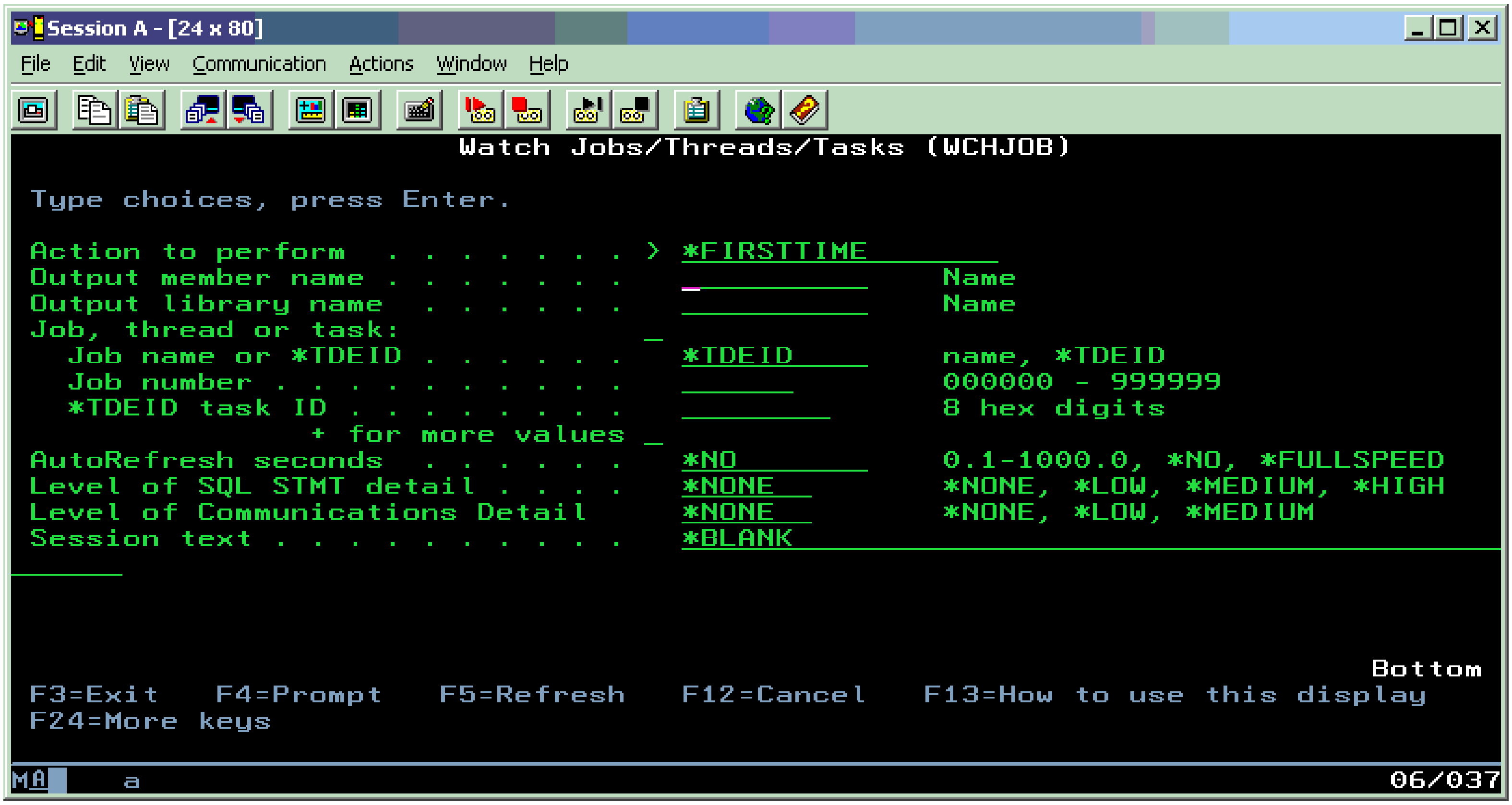


```
                     Watch Jobs/Threads/Tasks (WCHJOB)

 Type choices, press Enter.

 Action to perform . . . . . . .   *GETSAMPLE




                                                                         Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
MA     a                                                               05/037
```

Each of these actions will be discussed in greater detail below:

ACTION(*FIRSTTIME) - Action to perform

The *FIRSTTIME action parameter value will instruct the WCHJOB command to create a new Job Watch. WCHJOB does not have the system wide usage limitation that WRKSYSACT has, there can be one Job Watch session active per iSeries job. By typing *FIRSTTIME in the action to perform value and pressing enter you will see the following:



```
                           Watch Jobs/Threads/Tasks (WCHJOB)

 Type choices, press Enter.

 Action to perform  . . . . . . . > *FIRSTTIME
 Output member name . . . . . . .   _               Name
 Output library name  . . . . . .                   Name
 Job, thread or task:              _
   Job name or *TDEID . . . . . .   *TDEID          name, *TDEID
   Job number . . . . . . . . . .                   000000 - 999999
   *TDEID task ID . . . . . . . .                   8 hex digits
              + for more values _
 AutoRefresh seconds  . . . . . .   *NO             0.1-1000.0, *NO, *FULLSPEED
 Level of SQL STMT detail . . . .   *NONE           *NONE, *LOW, *MEDIUM, *HIGH
 Level of Communications Detail     *NONE           *NONE, *LOW, *MEDIUM
 Session text . . . . . . . . . .   *BLANK



                                                                        Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys

MA     a                                                                06/037
```

MBR - Output member name

This is the name of the job-specific Job Watch. This is the collection member name stored in the QPYRTJW* files.

LIB - Output library name

The name of the Job Watch collection library. The library must already exist.

JOBSTASKS - Job, thread or task

Job name or *TDEID: Specify the name of the job name or leave as *TDEID. If the job name is entered, the six digit job number must be entered in the Job number parameter. If
*TDEID is entered, the 8 hex byte TDE value must be entered in the *TDEID task ID parameter.

NOTE: Up to 10 jobs/tasks and up to a maximum of 500 threads within these selected jobs/tasks can be watched.

Job number: Six digit job number. This is used when a job name is specified in the Job name or *TDEID parameter.

***TDEID task ID**: The 8 hex byte TDE value (TDE ID).

AUTO - Auto refresh seconds

This parameter indicates whether the Job Watch should be automatically refreshed every X number of seconds. There are three possible values for this parameter:

Number:A number from .1 to 1000 - The number of seconds representing the size of each interval of data.

***NO:** Indicates that the Job Watch should NOT be automatically refreshed. Use this if you want to use the ACTION(*GETSAMPLE) parameter each time you want an interval of Job Watch data to be written.

***FULLSPEED:** Refresh as fast as possible. This will tell the WCHJOB command to collect and write an interval of data as fast as it can.

SQLDETAIL - Level of SQL STMT detail

Indicates how much SQL statement information should be collected about any jobs running SQL. The possible values for this parameter are:

***NONE:** No SQL information is collected.

***LOW:** Current or last SQL statement text including at least 2 levels of nesting (e.g. Fetch or Declare). The data is stored in file QPYRTJWASQ.

***MEDIUM:** *LOW plus the values of any host variables. The host variables are stored in file QPYRTJWASD.

***HIGH:** *MEDIUM plus the content of the 'Prepared Statement List' and Open Cursor List' which are primarily used for debugging temporary storage use of QZD* and QSQ* jobs. This data is stored in files QPYRTJWAS3, QPYRTJWAS4 and QPYRTJWAS5.

CMNDETAIL - Level of Communications Detail

Indicates how much communication/socket information should be collected for any jobs that have socket connections. The possible values for this parameter are:

***NONE:** No communications information is collected.

***LOW:** For each job, not each thread, TCP End Point data for each IFS File Open that is attached through a socket to a TCP End Point. This data is stored in file QPYRTJWAC1.

***MEDIUM:** *LOW plus information on each additional job, if any, that is sharing the socket. This additional information is stored in file QPYRTJWAC2.

TEXT- Session text

The up to 50 byte description to give the Job Watch.

AUTOACTGRP - ActGrp data every nth sample

This controls how often, if ever, Activation Group and Program Activation data should be harvested in auto-refresh mode. The reason for having this option is that harvesting this data can slow down the process of looking at a job/thread, and it typically is not as transient as other Job Watch harvested data. Note: in non auto-refresh mode, the option to harvest this data is provided on the command when **ACTION(*GETSAMPLE)** is performed.

The possible values for this parameter are:

Number: A number between 2-9999 that indicates how often the activation group data is collected. For example if the number if 5 the data is collected every 5th interval.

***STARTONLY:** Collected only at the start of the collection.

***NEVER:** Never collect this data.

***ALWAYS:** Collect the data every interval.

AUTOLIMIT - Auto refresh samples limit

This enforces a limit on how many samples are collected into the database, in auto-refresh mode, before the WCHJOB command automatically ends.

RULEMBR - Rule definition file member name

Options allowed are *NONE which indicates there are no rule definition defined or any member name in file QPYRTJWRD in the library specified by the RULELIB parameter. This file defines specific actions to take based on the values found in the data being monitored and collected.

RULELIB - Rule definition file library name

The value specified *NONE if no rule definition file is used or the library name containing a file QPYRTJWRD and a member specified by the RULEMBR parameter to use as the rule definition for the current job watch.

AUTOSTACK - Stack data every nth sample

This controls how often, if ever, the up to 50 level deep call stack is harvested for each thread in the Job Watch. This value can indicate to collect the call stack every nth interval, every interval or not at all. The call stack file consumes the most disk space of any Job Watcher file so there may be situations where it is worth while not to collect the call stack.

The possible values for this parameter are:

Number: A number between 2-999 that indicates how often the call stack data is collected. For example if the number if 5 the call stack is collected every 5th interval.

***NEVER:** Never collect this data.

***ALWAYS:** Collect the data every interval.

ACTION(*GETSAMPLE) - Action to perform

The *GETSAMPLE action parameter value will instruct the WCHJOB command to write the current data collected so far after the last call to *GETSAMPLE to the Job Watch files and begin collecting data for the next interval. You must call the WCHJOB command with ACTION(*FIRSTTIME) with the AUTO parameter set to *NO before calling the WCHJOB command with ACTION(*GETSAMPLE).

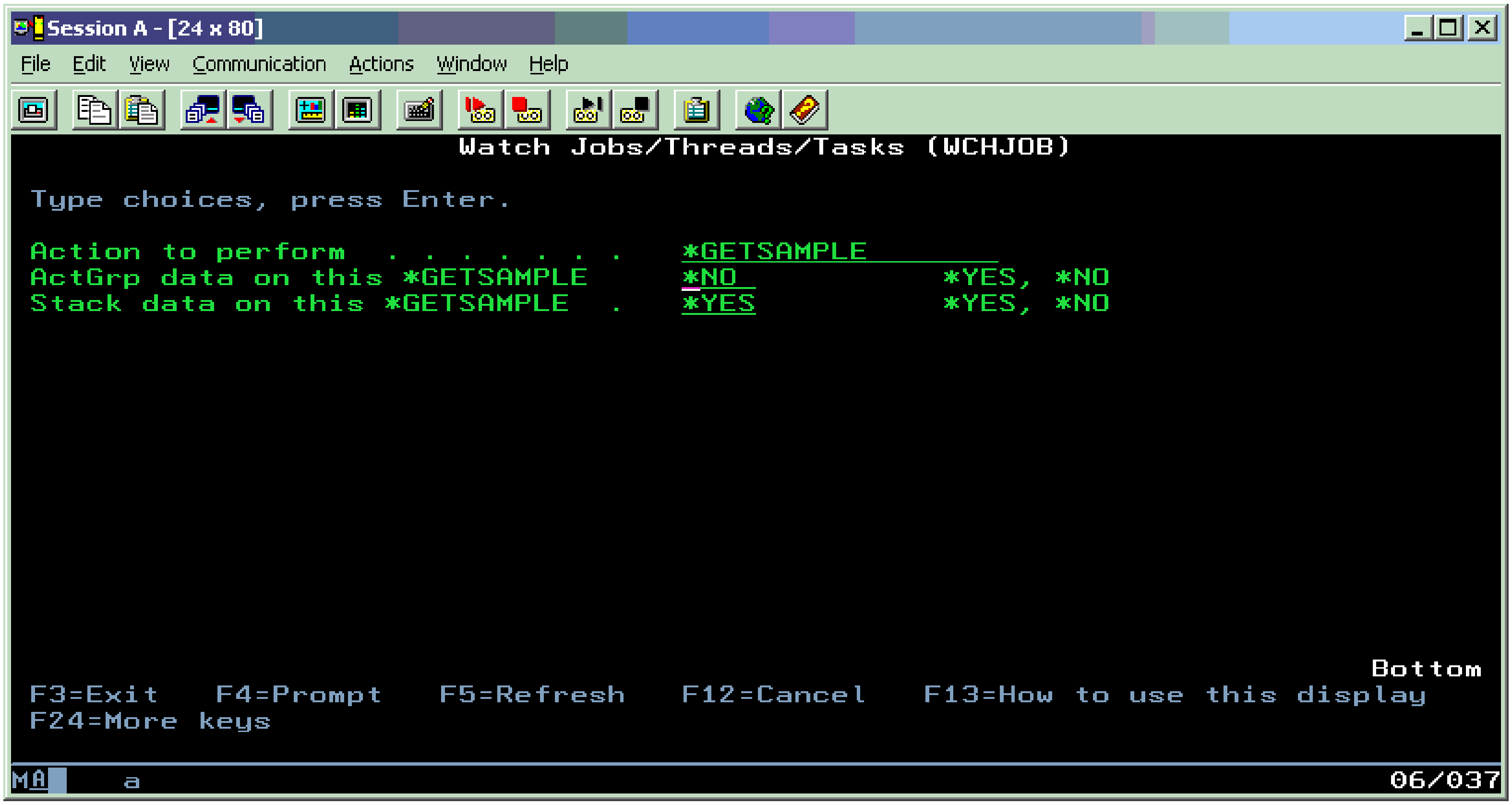
When the action to perform is *GETSAMPLE there are several parameters that can be modified at each sample of data as desired.

ACTGRPDTA()

This parameter indicates whether activation group information should be collected for this Job Watch interval.

STACKDTA()

This parameter indicates whether call stack information should be collected for this Job Watch interval.



```
                      Watch Jobs/Threads/Tasks (WCHJOB)

 Type choices, press Enter.

 Action to perform  . . . . . . .   *GETSAMPLE
 ActGrp data on this *GETSAMPLE     *NO              *YES, *NO
 Stack data on this *GETSAMPLE  .   *YES             *YES, *NO

                                                                        Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
MA     a                                                                06/037
```

ACTION(*QUIT)

The *QUIT action parameter value will immediately stop collecting and writing data to the active Job Watch files. If a Job Watch is not currently running in the current job, then this will do nothing.

ACTION(*GETTDELIST)

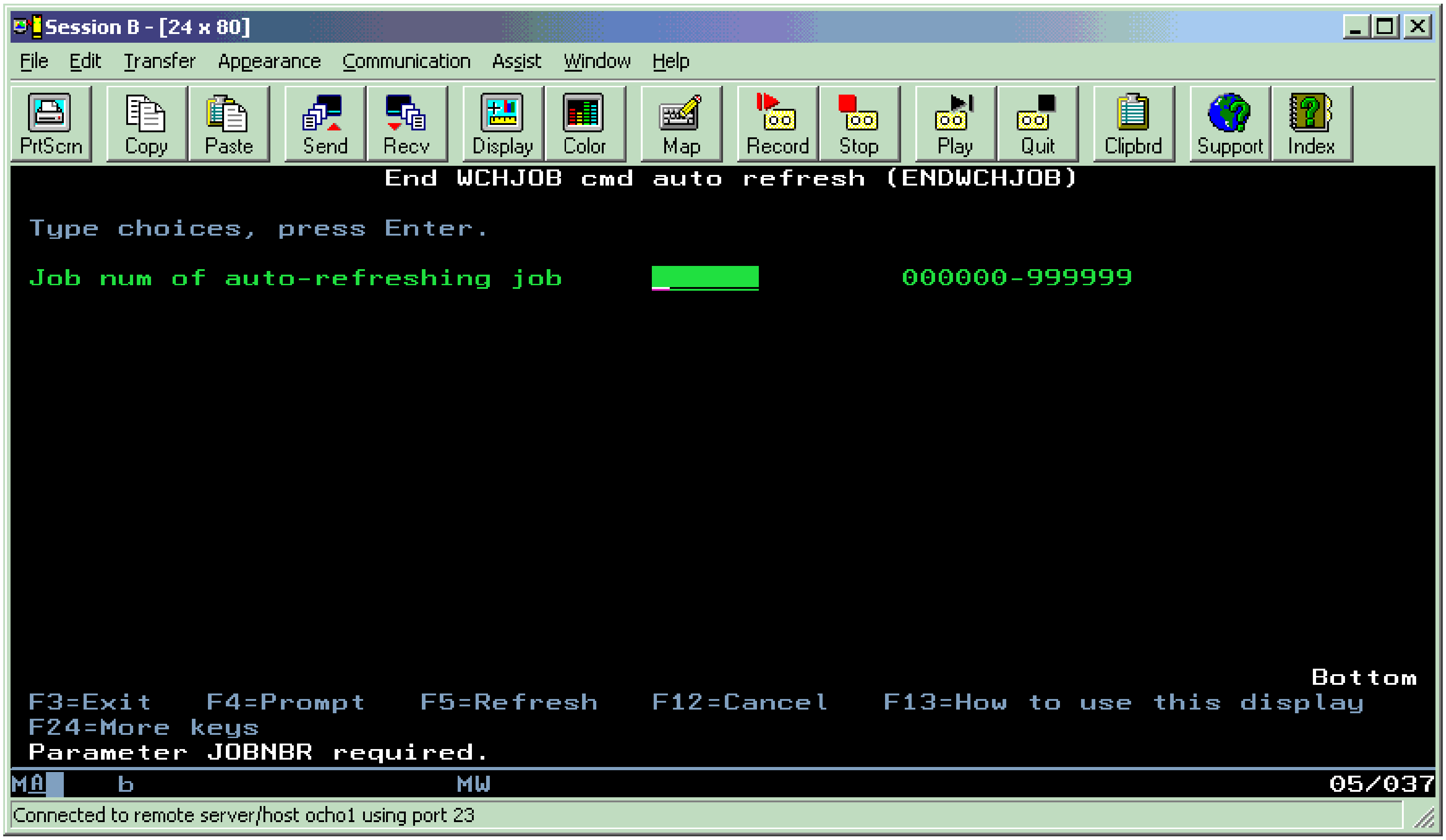
Use this parameter value to create a snapshot list of all the jobs/tasks and threads running on the system. This is not intended for use from the green screen command line. It is used by the GUI within the Start Job Watch Wizard.

Table of Contents Previous Next

1.3.2 ENDWCHJOB - ends a job watch

The ENDWCHJOB command ends a Job-specific Job Watch in progress. The command takes a single parameter which is the job number of the Job running the auto-refreshing job-specific Job Watch. To determine the job number use DSPJOB QPYJWJOB because all auto-refreshing job-specific Job Watches are named QPYJWJOB.

This command is only valid for stopping an auto-refreshing job-specific Job Watch. All job watches started in the GUI are auto-refreshing but there are options on the WCHJOB command to run a Job Watch and collect a sample each time the WCHJOB command is executed. This command does not work to stop that type of Job Watch. To do that use the WCHJOB command with an ACTION value of *QUIT.


Session B - [24 x 80]
File Edit Transfer Appearance Communication Assist Window Help
PrtScrn Copy Paste Send Recv Display Color Map Record Stop Play Quit Clipbrd Support Index

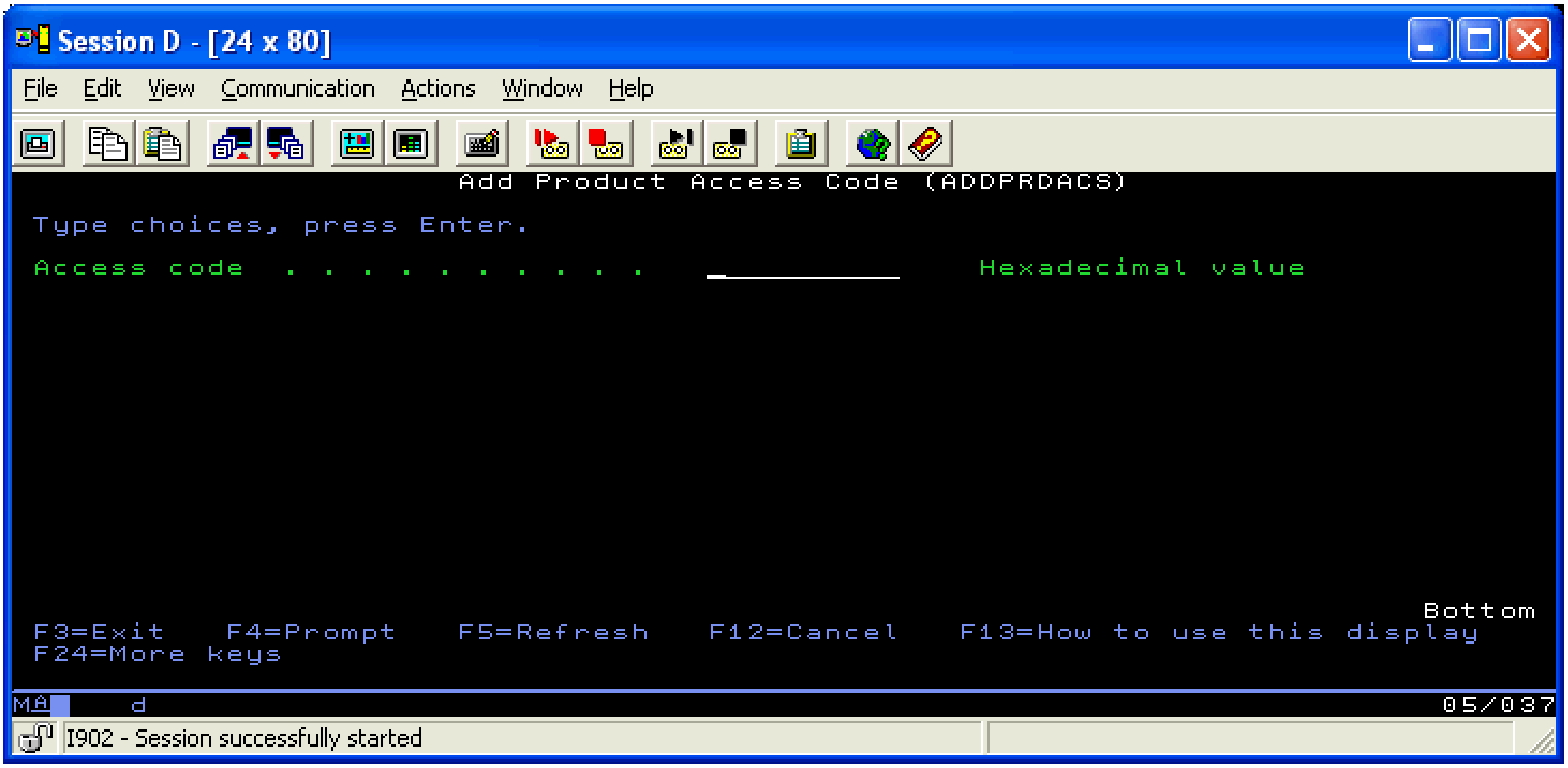
```
                     End WCHJOB cmd auto refresh (ENDWCHJOB)

 Type choices, press Enter.

 Job num of auto-refreshing job                    000000-999999










                                                                          Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
 Parameter JOBNBR required.
MA     b                       MW                                       05/037
```

Connected to remote server/host ocho1 using port 23

Table of Contents Previous Next

1.3.3 ADDPRDACS - apply an access code

This command is used to apply any iDoctor for iSeries component's access code to the current system.
The access codes are provided by IBM through this website for either a trial or licensed version.



```
                          Add Product Access Code (ADDPRDACS)

 Type choices, press Enter.

 Access code . . . . . . . . . .   __________       Hexadecimal value










                                                                          Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
MA      d                                                               05/037
```

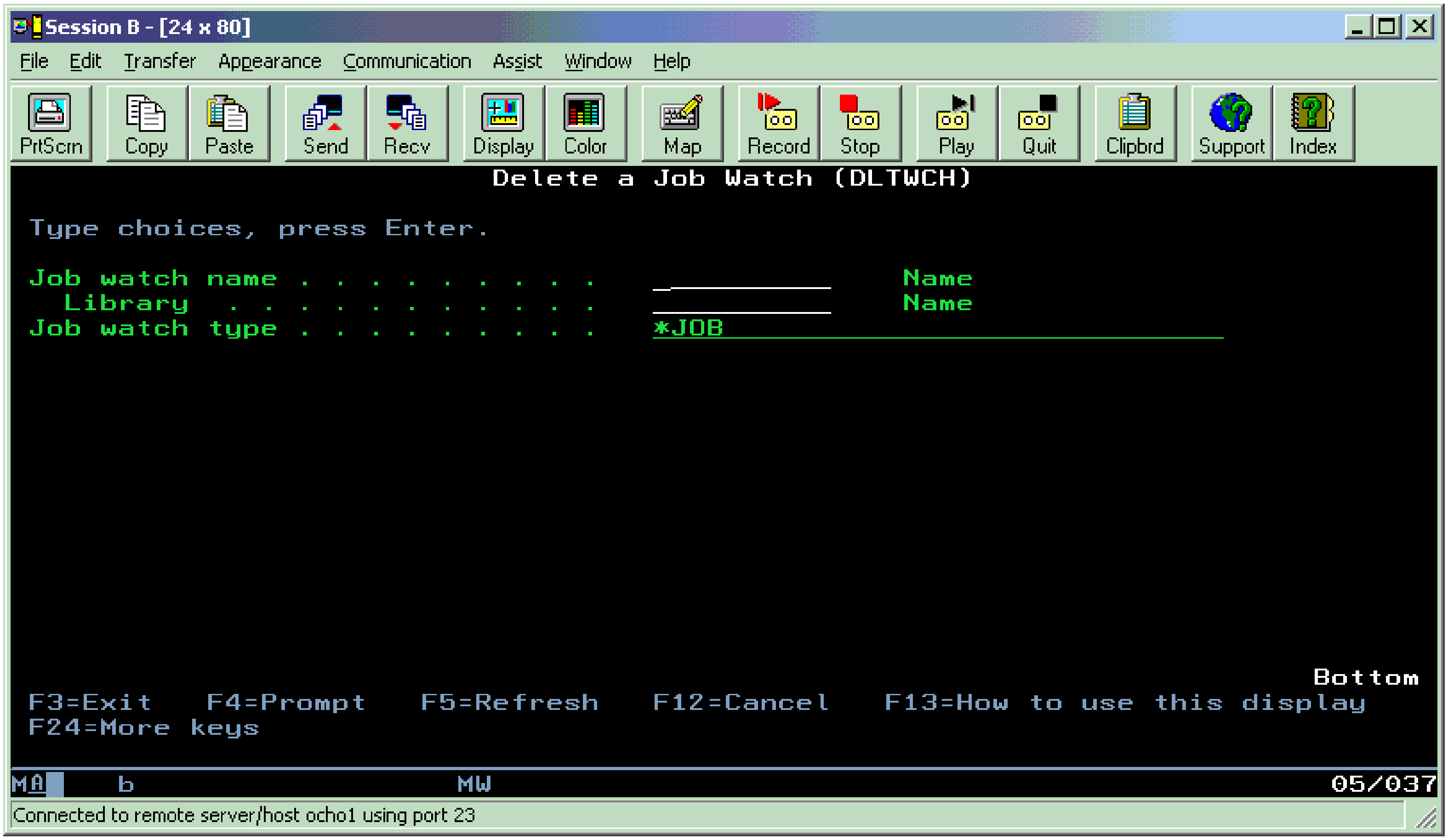
I902 - Session successfully started

Table of Contents | Previous | Next

1.3.4 DLTWCH - delete a job watch

The DLTWCH command removes a Job Watch from a users library. To use this command simply specify the Job Watch name (member name) , library name and the type of Job Watch *JOB or *SW.

Specifying the correct type of Job Watch to delete is critical because each type of Job Watch stores data in a different set of files.



Session B - [24 x 80]
File Edit Transfer Appearance Communication Assist Window Help
PrtScrn Copy Paste Send Recv Display Color Map Record Stop Play Quit Clipbrd Support Index

```
                          Delete a Job Watch (DLTWCH)

 Type choices, press Enter.

 Job watch name . . . . . . . . .   __________          Name
   Library  . . . . . . . . . . .   __________          Name
 Job watch type . . . . . . . . .   *JOB

                                                                        Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
MA     b                       MW                                       05/037
Connected to remote server/host ocho1 using port 23
```

Table of Contents | Previous | Next

1.3.5 JWAFTERIPL - resets the LIC wait buckets

The JWAFTERIPL command resets the LIC wait buckets for use by Job Watcher. This must be done at least once after an IPL has been performed. It is not necessary to run this command manually if you are using the client because the client will execute this command under the covers at startup.

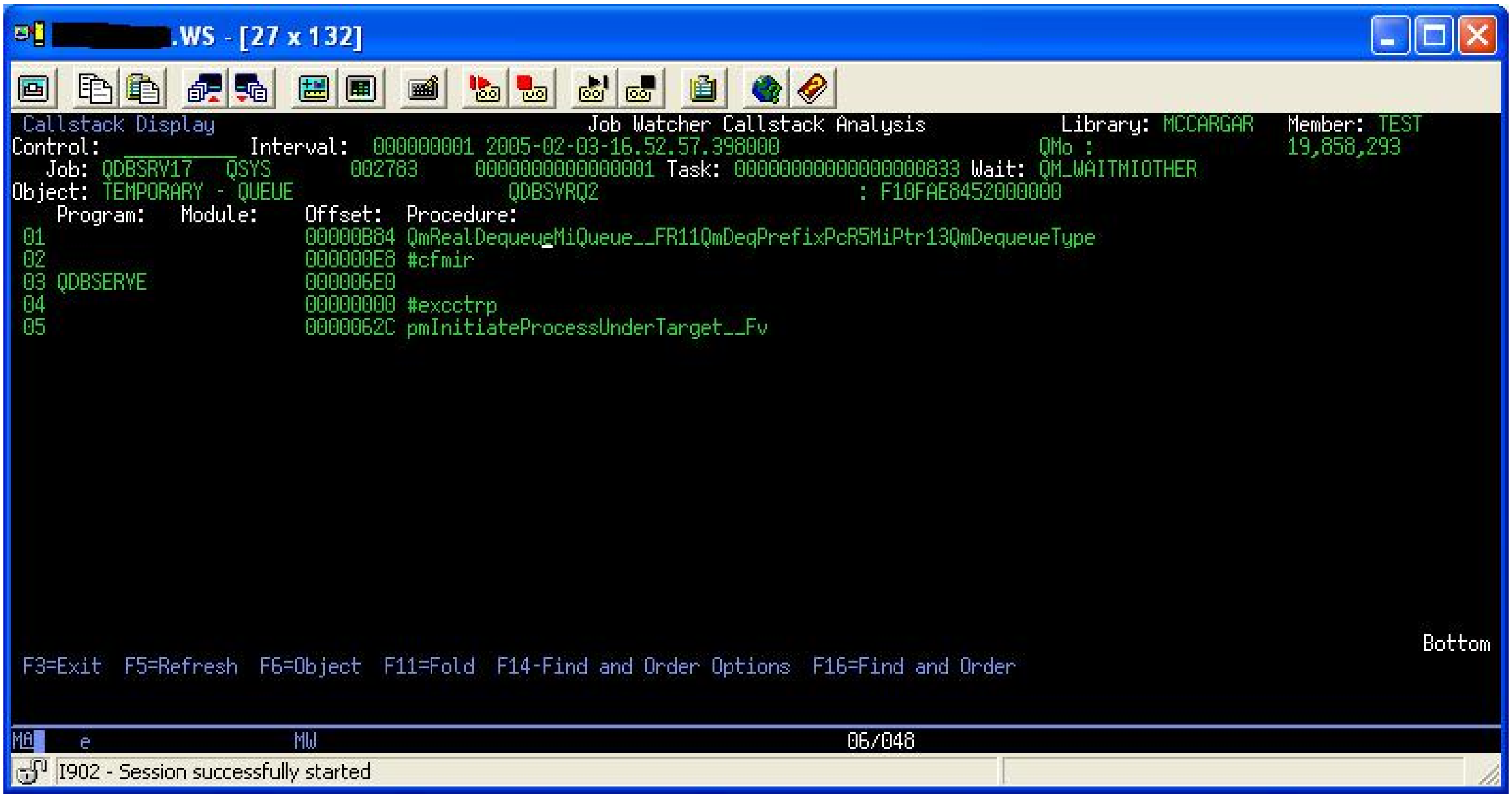
This command has no parameters.

Table of Contents | Previous | Next

1.3.6 ANZJWSTK - displays call stacks

This command is an alternative to the GUI for displaying a Job Watcher call stack and the additional fields used during an analysis (job, wait object, holder). This command only recognizes call stacks created for a job-specific collection and cannot be used with system-wide collections.

When using this command make sure that the screen size of your character based session is 27 x 132. The command will not run if the screen size is 27 x 80. Also the command ADDLIBLE QPYRTJW must be executed before running the command ANZJWSTK.



```
 Callstack Display                                 Job Watcher Callstack Analysis            Library: MCCARGAR   Member: TEST
Control: __________ Interval:  000000001 2005-02-03-16.52.57.398000                        QMo :                  19,858,293
   Job: QDBSRV17   QSYS       002783     0000000000000001 Task: 00000000000000000833 Wait: QM_WAITMIOTHER
Object: TEMPORARY - QUEUE                   QDBSVRQ2                       : F10FAE8452000000
    Program:   Module:    Offset:  Procedure:
 01                       00000B84 QmRealDequeueMiQueue__FR11QmDeqPrefixPcR5MiPtr13QmDequeueType
 02                       000000E8 #cfmir
 03 QDBSERVE              000006E0
 04                       00000000 #excctrp
 05                       0000062C pmInitiateProcessUnderTarget__Fv












                                                                                                                             Bottom
 F3=Exit  F5=Refresh  F6=Object  F11=Fold  F14-Find and Order Options  F16=Find and Order
```

I902 - Session successfully started

Table of Contents | Previous | Next

1.4 Rules and actions

Rules, in the context of a job-specific Job Watch, allow the WCHJOB command to test a user-specified set of conditions and take actions when those conditions are met. Actions taken can be one or both of the following:

- begin collecting Job Watcher data (assuming it was not already being collected)
- call a user exit program

A job-specific Job Watch may be started in a mode where it does not collect data, rather it "lurks" in the background for a long period of time, waiting for a set of conditions to be met in the job threads or system tasks it is "watching".

In order to define the rules for a Job-specific Job Watch, the recommended method is to follow the prompts in the Job Watch Wizard within the client. The wizard will create a rules-definition member in the rules definition file (QPYRTJWRD) in the library the Job Watch is being created in. If you prefer to use the green screen WCHJOB command to collect data you can create your own rules file member by copying file QPYRTJWRD from the Job Watcher library QPYRTJW to the library of your choice and specifying it as a parameter on the WCHJOB command. Detailed information on the format of this file is provided in the next section.

Table of Contents | Previous | Next

1.4.1 Rule definition file

First it needs to be emphasized that running a WCHJOB session in "wait mode" is completely optional and is controlled by the **RULEMBR()** keyword on the WCHJOB command when **ACTION(*FIRSTTIME)** is specified (i.e. starting up a WCHJOB session). The default, **RULEMBR(*NONE)**, specifies that the WCHJOB session should be started **without any rule definition processing**. If a name other than *NONE is provided, this causes WCHJOB to attempt to open and read that named member in a file named QPYRTJWRD ('RD', for rule definition) in the library specified on the **RULELIB()** keyword of the command.

It's important to note that the file QPYRTJWRD is **input** to the WCHJOB command, not output. The file and named member must exist and contain valid trigger control information before the WCHJOB command naming it is entered. There are two steps to create this file.

```
1. CRTDUPOBJ OBJ(QPYRTJWRD) FROMLIB(QPYRTJW) OBJTYPE(*FILE) TOLIB(MYLIB)
DATA(*YES)
2. WRKMBRPDM MYLIB/QPYRTJWRD - to create/edit the member of your choice
```

QPYRTJWRD is a source type physical file and can be edited via the wrkmbrpdm or edtsrc commands. The format and content of a QPYRTJWRD member is obviously restricted to what the WCHJOB command understands... and the interpreting algorithm is fairly simple. Therefore, it is highly recommended that one starts with a copy of the base member, QPYRTJWRD from the file QPYRTJWRD in library QPYRTJW and carefully edit in the necessary changes.

Any "syntax errors" that are encountered while this file is being processed by the WCHJOB command will cause the collecting job to end with a detailed escape message.

What follows are the first 144 lines from the base member (with a line number and a blank preceding each) and will be used for further discussion.

```
 1 QPYRTJWA
 2 *
 3 * first line must have QPYRTJWA in col 1!
 4 * * in column 1 is a comment
 5 *
 6 * This file-mbr is 'activated' or 'used' if the TRIGGERMBR() keyword on the
 7 * WCHJOB command has a value other than '*NONE' during ACTION(*FIRSTTIME).
 8 *
 9 * Once the trigger criteria and thresholds are set during
ACTION(*FIRSTTIME),
10 * they apply for the remainder of that WCHJOB session and cannot be changed
11 * or deactivated.
12 *
13 * A syntax error when reading this file will cause the WCHJOB command to
14 * abnormally terminate.
15 *
```

```
* A problem that occurs if and when a trigger exit program is called will
* cause the WCHJOB command to abnormally terminate.
*
*
* This file is composed of comments and 3 sections of non-comments:
*
* 1 - the first line must have QPYRTJWA starting in column 1
* 2 - control info must be the next non-comments, and is comprised of 8
*     non-comment lines that have required values starting in column 1.
* 3 - the AND OR logic that defines the trigger criteria
*
*
* CONTROL INFO:
*
* the following block of 8 non-comments must be consecutive
*
INITIAL STATE=COLLECT            (COLLECT or TRIGGERWAIT
TRIGGERWAIT DURATION COUNT=00100 (num of intervals
*RIGGERWAIT DURATION TIME=00060   (in seconds
CONSECUTIVE OCCURRENCE TRIGGER COUNT=00001
MULTIPLE TASK/THREAD CONSECUTIVE TRIGGER COUNT TEST=ALL  (or ANY
ACTION=CALL                      (COLLECT, CALL or BOTH
CALL=KURTZ/TRIG                  (must exist at startup time
FIRST CALL PARM='THE FIRST PARM'   (must have single quote marks
COLLECT DURATION COUNT=00100 (num of intervals
*OLLECT DURATION TIME=00060   (in seconds
*
* One and only one of the two TRIGGERWAIT DURATION lines must be uncommented.
*
* One and only one of the two COLLECT DURATION lines must be uncommented
*
* Even if ACTION is set to COLLECT, the CALL= line must exist.
*
* The program in the CALL= line must exist when the WCHJOB command is
* entered and its address is resolved at that time.
*
* The program is passed 7 parameters:
*
* 1 - the char string specified on the FIRST CALL PARM= line (CHAR 50 (max)
* 2 - library name of the job watch session (CHAR 10
* 3 - member name of the job watch session (CHAR 10
* 4 - the job or task name that satisfied the final/only trigger threshold
*     CHAR(32 ... matches the like field in file QPYRTJWA
* 5 - the thread id of the job that satisfied the final/only trigger threshold
*     (PACKED(20,0)
* 6 - the time of day of the triggering (CHAR 28
* 7 - the interval number (PACKED(20,0)
*
```

```
*
* SECTION 3 - TRIGGER CRITERIA
*
* next is the trigger criteria, it is made up of blocks of AND
* conditions connected with ORs.
*
* EQ = equals  NE = not equals GT = greater than  GE = greater than or equals
* LT = less than  LE = less than or equals
*
* for STACKS only CNT (contains) is valid
* for QTIME01 ... 32 PCT (percentage of total time) is an allowable modifier
* for QTIME04 ... 32 AVG (average by QCOUNT04 ... 32) is an allowable modifier
* for QCOUNT04 ... 32 RATE (rate per sec) is an allowable modifier
* for most CHAR fields, EQ or NE is the only allowable test
* for all CHAR fields, single quotes must enclose the compare value
* for all CHAR fields, the comparison is only for the length of the test data
*                      (starting in the left most position)
*
* The pgm that reads this file is not very sophisticated, values must start
* in specific columns. All values, except for quoted strings, are case-
* INsensitive.
*
* String comparisons do NOT take into account the length of the
* internal string. For example, suppose a field named JUNK is a
* CHAR(10) field and contains 'ABCDEFGHIJ':
*
*      JUNK      EQ  'ABCD'  - WOULD match, because the comparison is
*                              performed based on the length of the
*                              compare string provided.
* However:
*
*      JUNK      EQ  'ABCDEFGHIJ ' - probably would NOT match, and could
*                              possibly cause a terminating exception because
*                              the compare would compare 11 bytes starting
*                              with the first type of the field JUNK. The 11th
*                              byte is undefined: it may not exist at all and
*                              most likely would not contain a blank, although
*                              it possibly could.
*
* Numeric values can have at most, one digit to the right of the decimal.
*
* The field named STACK is not an actual field in any QPYRT* output files.
* It is a special case and represents any of the multiple fields holding
* program, module and procedure names:
*
*  STACK      CNT 'QDBGETM' - would match if any of the invocation levels
*                             contained a program, module or procedure
*                             name that began with the string QDBGETM.
*
*
*** *************** ***********************************************************
    STACK            CNT 'QDBGETM'
AND QCOUNT07         GT  0
AND RATE(SYNDBRD)    GT  4.3
OR  STACK            CNT 'QDBGETSQ'
AND QCOUNT07         GT  0
AND PCT(SYNDBRD)     GT  4.3
*
******************************************************************************
*(Most, but not all of the fields in the file QPYRTJWA are supported in this
*trigger file. The list of supported fields follows:
*
*
*              * 'N' = numeric, 'C' = char/string
* *
* * * 'R' = RATE() (per second)
* * *
* * ** 'P' = PCT() (percentage)
* * **
* * *** 'A' = AVG()
* Field * ***
* Name * *** Description
* ********* * *** **********************
* INTERVAL N Interval Number
* STATUS C Status
* DELTAUS N Delta usecs
* TASKCOUNT N Thread/Task Task Count
* DELTACPUU N Delta CPU usecs
* BIRTHDAY C Job/Task birth TOD
* NAME C Job/Task name
* ORIGPRI N Original priority
* BASEPRI N Base priority
```

The very first record of this file must contain the characters QPYRTJWA starting in column 1. After that, comments are any line that contains an asterisk (*) in the first column.

Table of Contents | Previous | Next

Chapter 2 Job Watcher - system-wide mode

System-wide Job Watcher offers the ability to collect information about all jobs/tasks/threads on a system. There is no limit to the number of jobs that can be collected.

The information collected by system-wide Job Watcher includes basic job statistics (cpu, IOs, faulting), wait information (time spent in each type of wait) and optionally call stacks and SQL statements for jobs meeting certain conditions specified by the WCHSYS command. In system-wide mode data is only collected for jobs that used at least some CPU during an interval or are stuck in a bad wait. The predetermined bad waits are wait types such as a seize, or an object lock.

This chapter describes the system-wide mode of the Job Watcher component on the server side. The following topics will be covered:

- Libraries QYPBASE, QPYRTJW
- Database files
- Command documentation

Table of Contents | Previous | Next

2.1 Libraries

When Job Watcher is installed, two libraries are created on the iSeries. These libraries are named QYPBASE and QPYRTJW.

QYPBASE contains programs and commands needed by the iDoctor for iSeries client. It also contains objects that are used by both Job Watcher and PEX Analyzer.

Library QPYRTJW is the Job Watcher library for release V5R2. This library contains programs, commands and database files needed to create and work with Job Watcher data.

Table of Contents | Previous | Next

2.2 Output files

Job Watcher provides two different types of collections over the jobs/threads/tasks on a system: job-specific and system-wide.

The files that make up a system-wide Job Watch are covered within this section.

Table of Contents Previous Next

2.2.1 QPYRTSWLG1 - Basic job statistics "log file"

Description: This file contains a unique set of jobs of interest based on the options provided in the client or on the WCHSYS command. Information about any rules that were met that fired off an action are also included in this file.

Optional: No

Job Watch: Yes

Record: One record is created per thread/task meeting specific performance criteria each interval.

Field name	Field Description	Format	Comments
INTERVAL	Interval number	Binary(4)	
SCORE	Score	Char(2)	A ranking of this job/thread/task within the category specified by the RCODES field
RCODES	Reason code	Char(30)	Identifies the reason the job was collected in this file.

INTPCT	Percent of interval time	Char(10)	Identifies the percentage of time this job spent in CPU, IO or whatever (based on the RCODES field) in relation to all other jobs for this interval
BKTPCT	Percent of bucket time	Char(10)	Identifies the percentage of time this job spent in the bucket identified by the field BLOCKBCKT in relation to all other jobs for this interval
NOTES	Comments	Char(100)	Additional information about the thread/task collected
DELTAUS	Interval delta usecs	Packed(11)	Elapsed interval time (in microseconds)
NAME	Job/task name	Char(32)	Fully qualified job name or task name
CURRUP	Current user profile	Char(10)	

TYPE	Job or task type flag	Char(1)	T = Task
			P = Job/Thread (Process)

THREADID	Thread ID (hex)	Hex(16)	
INITIAL	Initial thread flag	Char(1)	

TASKCOUNT	Job/task task count	Packed(11)	Uniquely identifies the thread/task running on the system.
TCOUNTH	Thread/task task count (in hex)	Hex(16)	Taskcount value in hex format
TDEID	Task dispatching element identifying the thread/task (in hex)	Hex(8)	
PRIORITY	Current LIC priority	Binary(2)	
JOBTYPE	Job type	Char(1)	Identifies the type of job (e.g. B = batch job)
JOBSTATUS	Job status	Char(4)	
JOBFUNC	Job function	Char(14)	
JOBSBS	Job subsystem	Char(10)	
CPU	CPU used	Packed(11)	in microseconds
SYNDBRD	Synchronous DB reads	Binary(4)	synchronous database reads
SYNNDBRD	Synchronous non-DB reads	Binary(4)	synchronous non-database reads
SYNDBWRT	Synchronous DB writes	Binary(4)	synchronous database writes
SYNNDBWRT	Synchronous non-DB writes	Binary(4)	synchronous non-database writes
ASYDBRD	Asynchronous DB reads	Binary(4)	asynchronous database reads
ASYNDBRD	Asynchronous non-DB reads	Binary(4)	asynchronous non-database reads
ASYDBWRT	Asynchronous DB writes	Binary(4)	asynchronous database writes

ASYNDBWRT	Asynchronous non-DB writes	Binary(4)	asynchronous non-database writes
IOPENDING	IO pending page faults	Binary(4)	
SMSYNCIO	Waits for asynchronous writes	Binary(4)	
FLTS	Page faults resulting in DASD reads	Binary(4)	
ALLOCATED	Allocated DASD pages	Binary(4)	
DEALLOCED	Deallocated DASD pages	Binary(4)	
BLOCKENUM	Current blocking enum (if blocked)	Binary(4)	Enums are a specific type of wait within the wait buckets. See the enum description file QPYRTJWAT2.
BLOCKBCKT	Current blocking bucket (if blocked)	Binary(4)	
CURRWTDUR	Total time in current wait (usecs)	Packed(11)	in microseconds
IPLUSECS	Time since the last IPL when this snapshot was taken	Packed(11)	in microseconds
QCOUNT01-QCOUNT32	Queueing bucket 01-32 count	Binary(4)	Total occurrences of each category of wait
QTIME01-QTIME32	Queueing bucket 01-32 time	Packed(11)	Total time (in microsconds) for each category of wait
HTASKCNT	Holding thread/task task count	Packed(11)	Identifies the thread or task the current thread/task is held by
HTHREADID	Holding thread ID	Hex(16)	

HTASKTDEID	Holding thread/task TDE ID	Hex(8)	
HTASKNAME	Holding thread/task name	Char(30)	
WOSEGTYP	Wait object segment type (hex)	Hex(4)	
WOOBJTYP	Wait object object type (hex)	Hex(4)	
WOOBJNAM	Wait object name	Char(30)	The name of the object this thread/task was waiting on when this snapshot was collected.
RECCNFLCT	Oridinal record number if in DB record lock conflict	Packed(11)	
LICWO	LIC 'wait object'	Char(3)	
LICWOHNDL	LIC 'wait object' handle	Hex(16)	
LDIOWRT	LDIO writes	Binary(4)	
LDIORD	LDIO reads	Binary(4)	
LDIOOTHR	LDIO other non reads/writes	Binary(4)	
CMNWRT	Communication file writes	Binary(4)	
CMNRD	Communication file reads	Binary(4)	
LDIOUPD	LDIO updates	Binary(4)	Not used
LDIODEL	LDIO deletes	Binary(4)	Not used
LDIOFEOD	LDIO feods	Binary(4)	Not used
LDIOCOMIT	LDIO commits	Binary(4)	Not used
LDIOROLLB	LDIO rollbacks	Binary(4)	Not used
LDIOOPEN	LDIO opens	Binary(4)	Not used

LDIOCLOSE	LDIO closes	Binary(4)	Not used
LDIOIXBLD	LDIO index builds	Binary(4)	Not used
LDIOSORT	LDIO sorts	Binary(4)	Not used
LDTAQSND	Data queue sends	Binary(4)	Not used
LDTAQRCV	Data queue receives	Binary(4)	Not used
LUSRSPCIOP	User space index/ops	Binary(4)	Not used
TXAPPIQT	Transaction application input queueing time	Packed(11)	in microseconds
TXRSCUT	Transaction resource usage time	Packed(11)	in microseconds
TXDSPLRT	Transaction I/O response time	Packed(11)	in microseconds
TXINQTRAN	Transaction input queueing transaction time	Binary(4)	in microseconds
TXRSCUTRAN	Transaction resource usage transaction time	Binary(4)	in microseconds
TXDSPLTRAN	Transaction display I/O transaction time	Binary(4)	in microseconds
IFSSYMLRD	IFS symbolic link reads	Binary(4)	
IFSDIRRD	IFS directory reads	Binary(4)	
IFSLUCHIT	IFS lookup cache hits	Binary(4)	
IFSLUCMIS	IFS lookup cache misses	Binary(4)	
IFSOPENS	IFS opens	Binary(4)	
IFSDIRCRT	IFS directory creates	Binary(4)	
IFSNDIRCRT	IFS non-directory creates	Binary(4)	
IFSDIRDLT	IFS directory deletes	Binary(4)	
IFSNDIRDLT	IFS non-directory deletes	Binary(4)	

ORIGPRI	Original priority	Packed(2)	
POOL	Pool ID	Packed(2)	
ACTWAIT	Active to wait transitions	Binary(4)	
WAITINEL	Wait to ineligible transitions	Binary(4)	
ACTINEL	Active to ineligible transitions	Binary(4)	
SYNDELTA	Interval total synchronous DASD IO	Packed(11)	
ASYDELTA	Interval total asynchronous DASD IO	Packed(11)	
TOTWRT	Interval total DASD writes	Packed(11)	
BEGSNAP	Snapshot start timestamp	TImestamp	
ALLOCTOT	Total 4K DASD pages allocated	Packed(6)	
DEALLOCTOT	Total 4K DASD pages deallocated	Packed(6)	
LOGSTKCNT	Stack flag	Packed(1)	
LOGSQLCNT	SQL flag	Packed(1)	
IPLUSSNAP	Time to take the snapshot for this TDE	Packed(5)	
PRVBKENUM	Previous current wait enum	Binary(2)	
PRVBKBCKT	Previous current wait bucket	Binary(2)	
PRVCURRWT	Previous current wait duration	Packed(11)	in microseconds
PRVLICWO	Previous LIC wait object	Char(3)	
PRVLICWOH	Previous LIC wait object handle	Hex(16)	

Table of Contents | Previous | Next

2.2.2 QPYRTSWLG2 - Basic job statistics "delta file"

Description: This file contains the delta values (between snapshot x and x+1) for all jobs that were active or in a "bad" type of wait. Jobs that are idle are not written to this file. Almost all of the system-wide graphs and reports in the client use this file.

Optional: Yes

Job Watch: Yes

Record: One record is created per thread/task that is either using CPU or is in a "bad wait" (like a seize conflict).

Field name	Field Description	Format	Comments
INTERVAL	Interval number	Binary(4)	
DELTAUS	Interval delta usecs	Packed(11)	Elapsed interval time (in microseconds)
NAME	Job/task name	Char(32)	Fully qualified job name or task name
CURRUP	Current user profile	Char(10)	

TYPE	Job or task type flag	Char(1)	T = Task
			P = Job/Thread (Process)

THREADID	Thread ID (hex)	Hex(16)	
INITIAL	Initial thread flag	Char(1)	

TASKCOUNT	Job/task task count	Packed(11)	Uniquely identifies the thread/task running on the system.
TCOUNTH	Thread/task task count (in hex)	Hex(16)	Taskcount value in hex format
TDEID	Task dispatching element identifying the thread/task (in hex)	Hex(8)	
PRIORITY	Current LIC priority	Binary(2)	
JOBTYPE	Job type	Char(1)	Identifies the type of job (e.g. B = batch job)
JOBSTATUS	Job status	Char(4)	Job status equivalent to WRKACTJOB status Field is blank if this record is a task (TYPE=T).
JOBFUNC	Job function	Char(14)	Current job function equivalent to WRKACTJOB function Field is blank if this record is a task (TYPE=T).
JOBSBS	Job subsystem	Char(10)	Subsystem the job is running in. Field is blank if this record is a task (TYPE=T).
CPU	CPU used	Packed(11)	in microseconds
SYNDBRD	Synchronous DB reads	Binary(4)	

SYNNDBRD	Synchronous non-DB reads	Binary(4)	
SYNDBWRT	Synchronous DB writes	Binary(4)	
SYNNDBWRT	Synchronous non-DB writes	Binary(4)	
ASYDBRD	Asynchronous DB reads	Binary(4)	
ASYNDBRD	Asynchronous non-DB reads	Binary(4)	
ASYDBWRT	Asynchronous DB writes	Binary(4)	
ASYNDBWRT	Asynchronous non-DB writes	Binary(4)	
IOPENDING	IO pending page faults	Binary(4)	
SMSYNCIO	Waits for asynchronous writes	Binary(4)	
FLTS	Page faults resulting in DASD reads	Binary(4)	
ALLOCATED	Allocated DASD pages	Binary(4)	
DEALLOCED	Deallocated DASD pages	Binary(4)	
BLOCKENUM	Current blocking enum (if blocked)	Binary(4)	Enums are a specific type of wait within the wait buckets. See the enum description file QPYRTJWAT2.
BLOCKBCKT	Current blocking bucket (if blocked)	Binary(4)	
CURRWTDUR	Total time in current wait (usecs)	Packed(11)	in microseconds
IPLUSECS	Time since the last IPL when this snapshot was taken	Packed(11)	in microseconds
QCOUNT01-QCOUNT32	Queueing bucket 01-32 count	Binary(4)	Total occurrences of each category of wait

QTIME01-QTIME32	Queueing bucket 01-32 time	Packed(11)	Total time (in microsconds) for each category of wait
HTASKCNT	Holding thread/task task count	Packed(11)	Identifies the thread or task the current thread/task is held by
HTHREADID	Holding thread ID	Hex(16)	
HTASKTDEID	Holding thread/task TDE ID	Hex(8)	
HTASKNAME	Holding thread/task name	Char(30)	
WOSEGTYP	Wait object segment type (hex)	Hex(4)	
WOOBJTYP	Wait object object type (hex)	Hex(4)	
WOOBJNAM	Wait object name	Char(30)	The name of the object this thread/task was waiting on when this snapshot was collected.
RECCNFLCT	Oridinal record number if in DB record lock conflict	Packed(11)	
LICWO	LIC 'wait object'	Char(3)	
LICWOHNDL	LIC 'wait object' handle	Hex(16)	
LDIOWRT	LDIO writes	Binary(4)	
LDIORD	LDIO reads	Binary(4)	
LDIOOTHR	LDIO other non reads/writes	Binary(4)	
CMNWRT	Communication file writes	Binary(4)	
CMNRD	Communication file reads	Binary(4)	

LDIOUPD	LDIO updates	Binary(4)	Not used
LDIODEL	LDIO deletes	Binary(4)	Not used
LDIOFEOD	LDIO feods	Binary(4)	Not used
LDIOCOMIT	LDIO commits	Binary(4)	Not used
LDIOROLLB	LDIO rollbacks	Binary(4)	Not used
LDIOOPEN	LDIO opens	Binary(4)	Not used
LDIOCLOSE	LDIO closes	Binary(4)	Not used
LDIOIXBLD	LDIO index builds	Binary(4)	Not used
LDIOSORT	LDIO sorts	Binary(4)	Not used
LDTAQSND	Data queue sends	Binary(4)	Not used
LDTAQRCV	Data queue receives	Binary(4)	Not used
LUSRSPCIOP	User space index/ops	Binary(4)	Not used
TXAPPIQT	Transaction application input queueing time	Packed(11)	in microseconds
TXRSCUT	Transaction resource usage time	Packed(11)	in microseconds
TXDSPLRT	Transaction I/O response time	Packed(11)	in microseconds
TXINQTRAN	Transaction input queueing transaction time	Binary(4)	in microseconds
TXRSCUTRAN	Transaction resource usage transaction time	Binary(4)	in microseconds
TXDSPLTRAN	Transaction display I/O transaction time	Binary(4)	in microseconds
IFSSYMLRD	IFS symbolic link reads	Binary(4)	
IFSDIRRD	IFS directory reads	Binary(4)	
IFSLUCHIT	IFS lookup cache hits	Binary(4)	

IFSLUCMIS	IFS lookup cache misses	Binary(4)	
IFSOPENS	IFS opens	Binary(4)	
IFSDIRCRT	IFS directory creates	Binary(4)	
IFSNDIRCRT	IFS non-directory creates	Binary(4)	
IFSDIRDLT	IFS directory deletes	Binary(4)	
IFSNDIRDLT	IFS non-directory deletes	Binary(4)	
ORIGPRI	Original priority	Packed(2)	
POOL	Pool ID	Packed(2)	
ACTWAIT	Active to wait transitions	Binary(4)	
WAITINEL	Wait to ineligible transitions	Binary(4)	
ACTINEL	Active to ineligible transitions	Binary(4)	
SYNDELTA	Interval total synchronous DASD IO	Packed(11)	
ASYDELTA	Interval total asynchronous DASD IO	Packed(11)	
TOTWRT	Interval total DASD writes	Packed(11)	
BEGSNAP	Snapshot start timestamp	TImestamp	
ALLOCTOT	Total 4K DASD pages allocated	Packed(6)	
DEALLOCTOT	Total 4K DASD pages deallocated	Packed(6)	
LOGSTKCNT	Stack flag	Packed(1)	
LOGSQLCNT	SQL flag	Packed(1)	
IPLUSSNAP	Time to take the snapshot for this TDE	Packed(5)	
PRVBKENUM	Previous current wait enum	Binary(2)	
PRVBKBCKT	Previous current wait bucket	Binary(2)	

PRVCURRWT	Previous current wait duration	Packed(11)	in microseconds
PRVLICWO	Previous LIC wait object	Char(3)	
PRVLICWOH	Previous LIC wait object handle	Hex(16)	

Table of Contents | Previous | Next

2.2.3 QPYRTSWLG3 - Basic job statistics "snapshot file"

Description: This file contains snapshot records for every interval for every job on the system. This file is rarely generated because it consumes a lot of disk space (much of it wasted due to records for idle threads). This file is optionally collected and intended for service use only.

Optional: Yes

Job Watch: Yes

Record: One record is created per thread/task per interval.

Field name	Field Description	Format	Comments
INTERVAL	Interval number	Binary(4)	
NAME	Job/task name	Char(32)	Fully qualified job name or task name
CURRUP	Current user profile	Char(10)	

TYPE	Job or task type flag	Char(1)	T = Task
			P = Job/Thread (Process)

THREADID	Thread ID (hex)	Hex(16)	
INITIAL	Initial thread flag	Char(1)	
TASKCOUNT	Job/task task count	Packed(11)	Uniquely identifies the thread/task running on the system.

TCOUNTH	Thread/task task count (in hex)	Hex(16)	Taskcount value in hex format
TDEID	Task dispatching element identifying the thread/task (in hex)	Hex(8)	
SYNDBRD	Synchronous DB reads	Binary(4)	
SYNNDBRD	Synchronous non-DB reads	Binary(4)	
SYNDBWRT	Synchronous DB writes	Binary(4)	
SYNNDBWRT	Synchronous non-DB writes	Binary(4)	
ASYDBRD	Asynchronous DB reads	Binary(4)	
ASYNDBRD	Asynchronous non-DB reads	Binary(4)	
ASYDBWRT	Asynchronous DB writes	Binary(4)	
ASYNDBWRT	Asynchronous non-DB writes	Binary(4)	
IOPENDING	IO pending page faults	Binary(4)	
SMSYNCIO	Waits for asynchronous writes	Binary(4)	
FLTS	Page faults resulting in DASD reads	Binary(4)	
ALLOCATED	Allocated DASD pages	Binary(4)	
DEALLOCED	Deallocated DASD pages	Binary(4)	
BLOCKENUM	Current blocking enum (if blocked)	Binary(4)	Enums are a specific type of wait within the wait buckets. See the enum description file QPYRTJWAT2.
BLOCKBCKT	Current blocking bucket (if blocked)	Binary(4)	
PRIORITY	Current LIC priority	Binary(2)	

CPU	CPU used	Packed(11)	in microseconds
JOBTYPE	Job type	Char(1)	Identifies the type of job (e.g. B = batch job)
JOBSTATUS	Job status	Char(4)	
JOBFUNC	Job function	Char(14)	
JOBSBS	Job subsystem	Char(10)	
CURRWTDUR	Total time in current wait (usecs)	Packed(11)	in microseconds
IPLUSECS	Time since the last IPL when this snapshot was taken	Packed(11)	in microseconds
QCOUNT01-QCOUNT32	Queueing bucket 01-32 count	Binary(4)	Total occurrences of each category of wait
QTIME01-QTIME32	Queueing bucket 01-32 time	Packed(11)	Total time (in microsconds) for each category of wait
HTASKCNT	Holding thread/task task count	Packed(11)	Identifies the thread or task the current thread/task is held by
HTHREADID	Holding thread ID	Hex(16)	
HTASKTDEID	Holding thread/task TDE ID	Hex(8)	
HTASKNAME	Holding thread/task name	Char(30)	
WOSEGTYP	Wait object segment type (hex)	Hex(4)	
WOOBJTYP	Wait object object type (hex)	Hex(4)	

WOOBJNAM	Wait object name	Char(30)	The name of the object this thread/task was waiting on when this snapshot was collected.
RECCNFLCT	Oridinal record number if in DB record lock conflict	Packed(11)	
LICWO	LIC 'wait object'	Char(3)	
LICWOHNDL	LIC 'wait object' handle	Hex(16)	
LDIOWRT	LDIO writes	Binary(4)	
LDIORD	LDIO reads	Binary(4)	
LDIOOTHR	LDIO other non reads/writes	Binary(4)	
CMNWRT	Communication file writes	Binary(4)	
CMNRD	Communication file reads	Binary(4)	
LDIOUPD	LDIO updates	Binary(4)	Not used
LDIODEL	LDIO deletes	Binary(4)	Not used
LDIOFEOD	LDIO feods	Binary(4)	Not used
LDIOCOMIT	LDIO commits	Binary(4)	Not used
LDIOROLLB	LDIO rollbacks	Binary(4)	Not used
LDIOOPEN	LDIO opens	Binary(4)	Not used
LDIOCLOSE	LDIO closes	Binary(4)	Not used
LDIOIXBLD	LDIO index builds	Binary(4)	Not used
LDIOSORT	LDIO sorts	Binary(4)	Not used
LDTAQSND	Data queue sends	Binary(4)	Not used

LDTAQRCV	Data queue receives	Binary(4)	Not used
LUSRSPCIOP	User space index/ops	Binary(4)	Not used
TXAPPIQT	Transaction application input queueing time	Packed(11)	in microseconds
TXRSCUT	Transaction resource usage time	Packed(11)	in microseconds
TXDSPLRT	Transaction I/O response time	Packed(11)	in microseconds
TXINQTRAN	Transaction input queueing transaction time	Binary(4)	in microseconds
TXRSCUTRAN	Transaction resource usage transaction time	Binary(4)	in microseconds
TXDSPLTRAN	Transaction display I/O transaction time	Binary(4)	in microseconds
IFSSYMLRD	IFS symbolic link reads	Binary(4)	
IFSDIRRD	IFS directory reads	Binary(4)	
IFSLUCHIT	IFS lookup cache hits	Binary(4)	
IFSLUCMIS	IFS lookup cache misses	Binary(4)	
IFSOPENS	IFS opens	Binary(4)	
IFSDIRCRT	IFS directory creates	Binary(4)	
IFSNDIRCRT	IFS non-directory creates	Binary(4)	
IFSDIRDLT	IFS directory deletes	Binary(4)	
IFSNDIRDLT	IFS non-directory deletes	Binary(4)	
ORIGPRI	Original priority	Packed(2)	
POOL	Pool ID	Packed(2)	
ACTWAIT	Active to wait transitions	Binary(4)	
WAITINEL	Wait to ineligible transitions	Binary(4)	

ACTINEL	Active to ineligible transitions	Binary(4)	

Table of Contents | Previous | Next

2.2.4 QPYRTSWALX - Run information

Description: The system-wide Job Watch control file. It provides a record per system-wide Job Watch created (or being created) in a library. This is a single member file.

Optional: Yes

Job Watch: No

Record: One record is created per collection. If a new collection is created with the same name for the current library, the existing record is overwritten.

Field name	**Field Description**	**Format**	**Comments**
INTERVAL	Interval number	Binary(4)	
BEGSNAP	Snapshot start date/time	Timestamp	
ENDSNAP	Snapshot end date/time	Timestamp	
WCHINTNO	Number of intervals to collect	Packed(3)	
WCHINTDUR	Interval duration (secs)	Packed(3)	in seconds
WCHINTDESC	Watch description	Char(50)	Description given to this job watch
SYSOSVRM	Collecting system OS VRM	Char(3)	Identifies the OS/400 release of the system this Job Watch was created on. This value is 520 on a V5R2 system.

IDOCVRM	Job Watcher VRM	Char(3)	Identifies the version of Job Watcher that was used to create this collection. This value is 130 on a V5R2 system.
WCHENDSTAT	Watch run/end status	Char(10)	
WCHJOBNAME	Watch job name	Char(10)	The name of the job running the collection
WCHUSRNAME	Watch job user	Char(10)	The user name of the job running the collection
WCHJOBNBR	Watch job number	Char(6)	The job number of the job running the collection
WCHCRTDTTM	Watch creation date/time	Char(15)	
WCHMBRNAME	Job watch name (member name)	Char(10)	The name of the job watch
WCHLIBNAME	Job watch library name	Char(10)	
WCHTYPE	Watch type	Char(8)	
WCHFILEOPT	Watch file option	Char(6)	Indicates which of the LG files should be collected
WCHTOP01	Top 01 list option	Packed(2)	
WCHTOP02	Top 02 list option	Packed(2)	
WCHTOP03	Top 03 list option	Packed(2)	
WCHTOP04	Top 04 list option	Packed(2)	

WCHTOP05	Top 05 list option	Packed(2)	
WCHCURENUM	Enum to watch	Char(3)	Identifies the specific wait point identifer to look for and log in the QPYRTSWLG2 file. This value could be a numeric enum identifier or 3 character lic wait object.
WCHRULE1	Watch rule prompt #1	Char(61)	
WCHRULE2	Watch rule prompt #2	Char(61)	
WCHRULE3	Watch rule prompt #3	Char(61)	
WCHRULE4	Watch rule prompt #4	Char(61)	
WCHRULEMBR	Watch rule file member	Char(10)	The member name to use as the rule definition from file QPYRTSWRD. This value is not required and could be blank.
WCHRULELIB	Watch rule file library	Char(10)	The library name containing a rule definition file QPYRTSWRD or blank
WCHRULECMD	Watch rule #1 command	Char(200)	Command string to execute if the rule condition is met
WCHSTKBLK	Collect selected call stacks flag	Char(4)	

WCHSTKBLK1	Stack waiter/holder limit	Packed(3)	Indicates the minimum time a job must be waiting or holding another job before the call stack is collected (in microseconds)
WCHSTKBLK2	Stack other bad wait limit	Packed(3)	Indicates the minimum time a job must be in a bad wait before the call stack is collected (in microseconds)
WCHSTKFLT	Include faulting call stacks	Char(4)	Indicates if call stacks for jobs in a page fault state should be collected each interval
WCHSQL	Collect active SQL statements	Char(4)	Indicates if SQL statements should be collected if they are active at the time the snapshot is taken each interval.
WCHACT	Ignore actions to iDoctor jobs	Char(4)	Indicates whether rule actions should apply to jobs named QPYJWJOB or QPYSWJOB or IDOCCOL. These job names are used when creating PEX collection or job watches.

WCHSRV	Service remap enum/wait bucket	Char(43)	Indicates whether an enum should be remapped to a new bucket before the collection is started.
WCHLIMIT	Collection limit	Packed(3)	This value indicates the maximum size the collection could reach (in megabytes) before the job watch is automatically ended.
WCHSWITCH1	Reserved watch switch 1	Char(4)	
WCHTOTLG1	Count of LG1 file records	Packed(5)	
WCHTOTLG2	Count of LG2 file records	Packed(5)	
WCHTOTALS	Count of stack file records	Packed(5)	
WCHTOTALQ	Count of SQL file records	Packed(5)	
WCHTOTMB	Collection size estimate (in MB)	Packed(8)	in megabytes
WCHTOTALA	Latest active job count	Packed(5)	Number of jobs using CPU or in a bad wait during the last interval collected
WCHTOTALI	Latest idle job count	Packed(5)	Number of jobs not using CPU during the last interval collected
WCHRESRVD1	Reserved field #1	Char(50)	

Table of Contents | Previous | Next

2.2.5 QPYRTSWDFN - Job watch definition

Description: This file stores the command used by the GUI to create this Job Watch. The file is only created if the Job Watch is submitted through the client.

Optional: No

Job Watch: Yes (if created via GUI)

Record: One record is created per collection.

Field name	**Field Description**	**Format**	**Comments**
NAME	Definition name	Char(10)	This is the Job Watch name
OSVRM	OS/400 version/release/mod level	Char(3)	At release V5R2 this value is 520
COMMAND	Command used to create this job watch	Char(1000)	

Table of Contents Previous Next

2.2.6 QPYRTSWALQ - SQL statements

Description: This file contains SQL statements that were captured by the system-wide job watch.

Optional: Yes

Job Watch: Yes

Record: One record is created per SQL statement per job

Field name	Field Description	Format	Comments
INTERVAL	Interval number	Packed(3)	
TASKCOUNT	Job/task task count	Packed(11)	Uniquely identifies the thread/task running on the system.
SRCLIB	SQL package source library	Char(10)	
SRCFILE	SQL package source file	Char(10)	
SRCMBR	SQL package source file member	Char(10)	
SRCDATE	SQL package source last changed date	Char(13)	
PKGLIB	SQL package library/container name	Char(18)	
PKGNAME	SQL package name	Char(18)	
RDBSNAME	Remote DBS name	Char(18)	
HOSTREAL	Actual number of host variables	Binary(4)	Number of host variables associated with this SQL statement

MORE	Another statement associated with this statement	Char(1)	1 = another statement associated with this one
CCSID	CCSID of dynamic statement	Binary(4)	
SQLSTMT	SQL statement	Varchar(300)	

Table of Contents Previous Next

2.2.7 QPYRTSWALS - Call stacks

Description: This file contains call stacks for jobs and holding jobs that matched the performance statistics specified when creating the collection.

Optional: Yes

Job Watch: Yes

Record: One record is created per call stack collected

Field name	Field Description	Format	Comments
INTERVAL	Interval number	Packed(3)	
TASKCOUNT	Job/task task count	Packed(11)	Uniquely identifies the thread/task running on the system.
REASON	Reason code	Binary(2)	Code indicating the reason the call stack was collected. 1 = Holder 2 = Waiter (and there will be a holder) 3 = Blocked (and there is no holder reported) 4 = Faulting 5 = Running
USECSIPL	Time since last IPL	Packed(11)	in microseconds
HTDECOUNT	Holder task count	Packed(11)	the taskcount of the holder

STACK	Call stack contents	Varchar(300)	

Table of Contents | Previous | Next

2.2.8 QPYRTSWIST - Totals for the interval summary file QPYRTSWIS1

Description: This file is created after a collection has finished running and the user selects to view the property pages for the job watch in the GUI.

Optional: Yes

Job Watch: Yes (but only created once the properties are viewed for a job watch within the GUI)

Record: This file will contain only 1 record after the properties are shown for the first time.

Field name	Field Description	Format	Comments
INTERVALS	Total intervals	Packed(3)	
TOTALTIME	Total elapsed collection time	Packed(16)	in microseconds
STARTTIME	Collection start time	Timestamp	time the collection gathered its first interval of data
ENDTIME	Collection end time	Timestamp	time the collection gathered its last interval of data
AVGINTDUR	Average snapshot usecs	Packed(8)	the average time it took to create a snapshot of job information for an interval across the entire system
AVGINTDUR	Average interval duration	Packed(7)	the average size of each interval
AVGACT	Average active tasks/threads	Packed(7)	

AVGIDL	Average idle tasks/threads	Packed(7)	
TOTBLKS	Total blocked at snapshot	Packed(7)	the total of the number of threads in a bad wait at the start of an interval
TOTBLKT	Total blocked thru interval	Packed(7)	the total of number of threads in a bad wait throughout an interval
TOTCPU	Total CPU usecs	Packed(16)	
TOTCPU1	Total CPU01 usecs	Packed(16)	
TOTIO	Total IO requests	Packed(14)	
TOTSYN	Total synchronous requests	Packed(14)	
TOTASY	Total asynchronous requests	Packed(14)	
TOTRD	Total reads	Packed(14)	
TOTWRT	Total writes	Packed(14)	
TOTIOP	Total IOP requests	Packed(14)	
TOTWIO	Total waits for async writes	Packed(14)	
TOTFLT	Total page faults	Packed(14)	
TOTJWINIT	Total job watch initiations	Packed(3)	
TOTJBINIT	Total job initiations	Packed(3)	
TOTJBTERM	Total job terminations	Packed(3)	
TOTSTKCNT	Total job stack counts	Packed(3)	The total number of call stacks collected
TOTSQLCNT	Total job SQL counts	Packed(3)	The total number of SQL statements collected
TOTRULEH	Total satisfied rule hits	Packed(3)	

TOTUSCWH	Total user select wait hits	Packed(3)	
TOTUSCWU	Total user select wait usecs	Packed(8)	in microseconds

2.2.9 QPYRTSWIS1 - Interval summary file

Description: This file provides a summary of the jobs in the system-wide collection for every interval.

Optional: No

Job Watch: Yes

Record: One record is created per interval.

Field name	Field Description	Format	Comments
INTERVAL	Interval number	Packed(3)	
BEGSNAP	Snapshot start date/time	Timestamp	
ENDSNAP	Snapshot end date/time	Timestamp	
SNAPDELTA	Snapshot elapsed time (usecs)	Packed(8)	in microseconds
AVGINTDUR	Average interval duration	Packed(7)	
INTTOTACT	Number of active tasks/threads	Packed(7)	number of jobs/tasks/threads that used > 0 CPU this interval or were in a bad wait
INTTOTIDL	Number of idle tasks/threads	Packed(7)	number of jobs/tasks/threads that used 0 CPU this interval and were not in a bad wait

INTTOTBLKS	Number blocked at snapshot start	Packed(7)	number of jobs/tasks/threads in a bad wait when the 1st interval snapshot was taken
INTTOTBLKT	Number blocked thru interval	Packed(7)	number of jobs/tasks/threads in a bad wait throughout the entire interval
INTTOTCPU	Interval CPU usecs	Packed(7)	
INTTOTCPU1	Interval CPU01 usecs	Packed(7)	
INTTOTIO	Interval total IO requests	Packed(7)	
INTTOTSYN	Interval total synchronous requests	Packed(7)	
INTTOTASY	Interval total asynchronous requests	Packed(7)	
INTTOTRD	Interval total read requests	Packed(7)	
INTTOTWRT	Interval total write requests	Packed(7)	
INTTOTIOP	Interval total IOP requests	Packed(7)	
INTTOTWIO	Interval total waits for asynchronous IO requests	Packed(7)	
INTTOTFLT	Interval total page faults	Packed(7)	
INTJWINIT	Interval JW initiations	Packed(3)	number of job-specific job watches started this interval based on statisfied rule conditions
INTJBINIT	Interval job initiations	Packed(3)	number of jobs/threads/tasks created

INTJBTERM	Interval job terminations	Packed(3)	number of jobs/threads/tasks destroyed
INTSTKCNT	Interval job stack counts	Packed(3)	number of call stacks collected
INTSQLCNT	Interval job SQL counts	Packed(3)	number of SQL statements collected
INTDUMPREQ	Dump request member name	Char(10)	service use only
INTDUMPR#	Number of dump requests	Packed(3)	service use only
INTDUMPSEC	Estimated record secs	Packed(3)	service use only
INTRULEH	Interval satisified rule hits	Packed(3)	number of jobs that satsified the rule conditions specified by this job watch
USRSCW	User selected current wait	Char(4)	
INTUSCWH	Interval USCW hits	Packed(3)	
INTUSCWU	Interval USCW usecs	Packed(8)	
SNPUSECS	Usecs since IPL this snapshot was started	Packed(11)	
INTCPC	Current processor capacity	Packed(3)	
INTNUMPROC	Number of processors	Packed(3)	
INTPROCSHR	Processor sharing attribute	Char(1)	
INTSSTSPCU	QWCRSSTS percent cpu used	Packed(3)	
INTSSTSHMS	QWCRSSTS elapsed HHMMSS	Char(6)	
INTSNAPPCU	Snapshot percent cpu used	Packed(3)	

INTJBIANT	Interval job initiations and terminations	Packed(3)	number of jobs/threads/tasks created - destroyed

Table of Contents Previous Next

2.2.10 QPYRTSWNM1 - CHGWCHNAM command cross reference file

Description: This file is created by the command CHGWCHNAM. This command will modify customer sensitive fields like system names and user profiles in the job watcher data files and replace them with generic names. This file is a mapping of the real names with the fake names. It should not be sent along with the rest of the collection data.

Optional: Yes

Job Watch: No

Record: One record is created per unique name converted by the CHGWCHNAM command within the collection.

Field name	Field Description	Format	Comments
REFNUM	Reference number	Zoned(5)	
ORGNAM	Original name	Char(10)	

Table of Contents | Previous | Next

2.2.11 QPYRTJWSYS - System information

Description: This file is created at the start of every collection (for both system-wide and job-specific modes) and will contain basic system information and system values for the system that created the collection.

Optional: No

Job Watch: Yes

Record: This file will contain only 1 record

Field name	Field Description	Format	Comments
SYSTEM	System name	Char(8)	
QCCSID	CCSID	Binary(4)	
QCHRID	Character set and code page	Char(20)	
QCNTRYID	Country or region identifier	Char(2)	
QDYNPTYADJ	Dynamic priority adjustment	Char(1)	
QDYNPTYSCD	Dynamic priority scheduler	Char(1)	
QIGC	DBCS installed	Char(1)	
QMODEL	System model	Char(4)	
QPFRADJ	Performance adjustment	Char(1)	
QPRCMLTTSK	Processor multitasking	Char(1)	
QPRCFEAT	Processor feature code	Char(4)	
QQRYDEGREE	Parallel processing degree	Char(10)	
QQRYTIMLMT	Query processing time limit	Char(10)	

QSFWERRLOG	Software error logging	Char(10)	
QSHRMEMCTL	Shared memory control	Char(1)	
QSRLNBR	System serial number	Char(8)	
QTSEPOOL	Time-slice end pool	Char(10)	
QUSEADPAUT	Use adopted authority	Char(10)	
QSYSLIBL	System library list	Char(150)	
QUSRLIBL	User library list	Char(250)	
QPARTID	Parition ID	Binary(2)	
QPARTNUM	Number of partitions	Binary(2)	
QRSFLAG	Restricted state flag	Char(1)	

Table of Contents | Previous | Next

2.2.12 QPYRTSWRDB - Rule definitions

Description: This file contains a copy of the rule definition that was used at the start of the collection (if any). If a rule is not used this file/member will not be created.

Optional: No

Job Watch: Yes

Field name	Field Description	Format	Comments
SRCSEQ		Zoned(6)	
SRCDAT		Zoned(6)	
SRCDTA		Char(80)	See file QPYRTSWRD in library QPYRTJW for an example

Table of Contents Previous Next

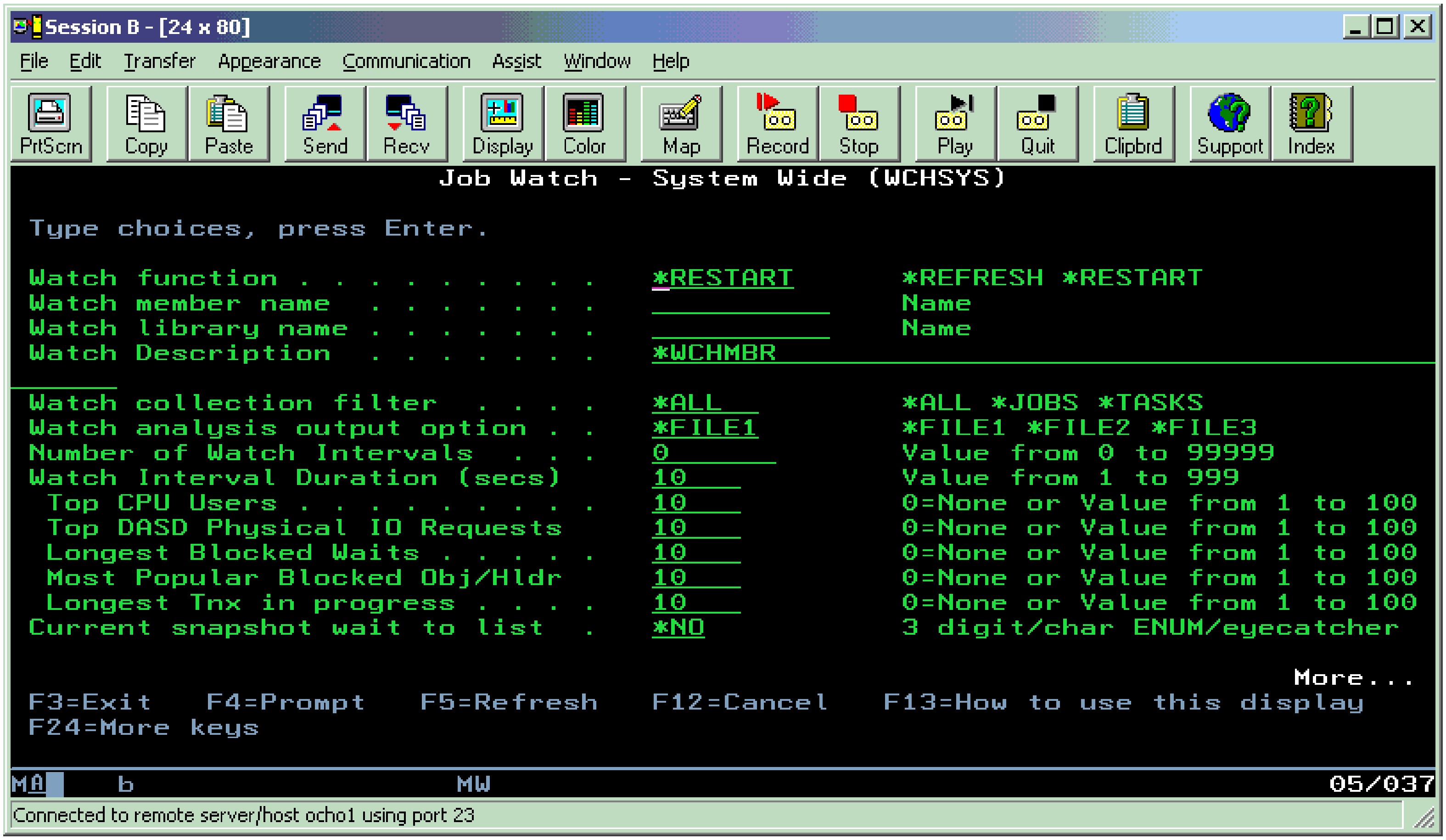
2.3 Commands

This section covers the green screen commands that are provided with Job Watcher for working with system-wide job watches.
These commands exist in the Job Watcher library QPYRTJW.

Table of Contents Previous Next

2.3.1 WCHSYS - creates a job watch

The WCHSYS command is used to define and start a system-wide Job Watch.



```
                        Job Watch - System Wide (WCHSYS)

 Type choices, press Enter.

 Watch function . . . . . . . . . .   *RESTART       *REFRESH *RESTART
 Watch member name  . . . . . . . .   __________     Name
 Watch library name . . . . . . . .   __________     Name
 Watch Description  . . . . . . . .   *WCHMBR

 Watch collection filter  . . . . .   *ALL           *ALL *JOBS *TASKS
 Watch analysis output option . . .   *FILE1         *FILE1 *FILE2 *FILE3
 Number of Watch Intervals  . . . .   0              Value from 0 to 99999
 Watch Interval Duration (secs)       10             Value from 1 to 999
  Top CPU Users . . . . . . . . . .   10             0=None or Value from 1 to 100
  Top DASD Physical IO Requests       10             0=None or Value from 1 to 100
  Longest Blocked Waits . . . . . .   10             0=None or Value from 1 to 100
  Most Popular Blocked Obj/Hldr       10             0=None or Value from 1 to 100
  Longest Tnx in progress . . . . .   10             0=None or Value from 1 to 100
 Current snapshot wait to list  . .   *NO            3 digit/char ENUM/eyecatcher

                                                                        More...
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys

MA     b                      MW                                        05/037
Connected to remote server/host ocho1 using port 23
```

Job Watcher is a non-intrusive real time sampling of jobs/threads/tasks which collects performance information. This command is used to collect a system-wide view of files on a V5R2 system. All data collected is stored in database files named QPYRTSW*.

NOTE: This function runs immediately "under" the WCHSYS command, so it is recommended that the command be submitted to batch.

WCHTYP - Type of request
Always use the default value of *RESTART.

WCHMBR - Job watch member name
The name of the Job Watch System Wide collection. This will also be name of the member in the QPYRT* files located in the Job Watch library specified by the WCHLIB parameter.

WCHLIB - Job watch library name
The name of the library to create the Job Watch System Wide collection in. The library must exist.

WCHDESC
Description of the collection.

WCHCOL - Job watch collection filter
Provides the option to filter out all jobs or all tasks, or you many collect information about all jobs and tasks on the system.

WCHOUT - Job watch analysis output option

FILE1
Includes only exception or top N jobs/tasks each interval.

FILE2
Includes all active or 'badly blocked' jobs/tasks for each

interval. This option also collects the *FILE1 option.

FILE3
Includes all jobs/tasks each interval (NOT recommended - service use only). This option also collects the *FILE1 and *FILE2 options.

WCHINTNUM - Number of job watch intervals
The total number of intervals to collect data for.

WCHINTDUR - Job watch interval duration (sec)
The time between interval samples in seconds.

WCHLST1 - Top CPU users
Selects how many jobs/tasks that are the top CPU users should be added to the log file (QPYRTSWLG1) every interval. Value = 0-100.

WCHLST2 - Top DASD Physical IO requests
Selects how many jobs/tasks that had the most DASD physical IOs should be added to the log file (QPYRTSWLG1) every interval. Value = 0-100.

WCHLST3 - Longest Blocked Waits
Selects how many jobs/tasks that had the longest bad waits should be added to the log file (QPYRTSWLG1) every interval. Value = 0-100.

WCHLST4 - Most Popular Blocked Objects/Holders
Selects how many objects/holders that were most commonly blocked should be added to the log file (QPYRTSWLG1) every interval. Value = 0-100.

WCHLST5 - Longest Transactions in progress
Selects how many jobs/tasks having long transactions should be added to the log file (QPYRTSWLG1) every interval. Value = 0-100.

WCHENUM - Current snapshot wait to list
Selects a job with a current wait based on a 3 digit ENUM, or a 2 digit wait bucket preceded by the character "B", or a 3 charater wait code eye-catcher value like RMo and records the instance in the log file.

WCHRULE1 - Job watch rule statement parm #1
A rule can be defined using the EXACT positioning and filling in the values layed out by the default rule format without having to use the rules file.

For more information on this format, see the rules file QPYRTSWRD in library QPYRTJW.

WCHRULE2 - Job watch rule statement parm #2
A rule can be defined using the EXACT positioning and filling in the values layed out by the default rule format without having to use the rules file.

WCHRULE3 - Job watch rule statement parm #**3**
A rule can be defined using the EXACT positioning and filling in the values layed out by the default rule format without having to use the rules file.

WCHRULE4 - Job watch rule statement parm #4
A rule can be defined using the EXACT positioning and filling in the values layed out by the default rule format without having to use the rules file.

WCHRULEMBR - Job watch rule definition file member name
The name of the rules file QPYRTSWRD member if one has been created.

WCHRULELIB - Job watch rule definition file library name
Library in which the rules file QPYRTSWRD resides to use for this collection.

WCHRULECMD - Job watch rule #1 action command
Specify a command that a rule action would execute if the 1st rule

specified by the WCHRULE1-4 parameters was satisfied.

WCHSTKBLK - Job watch selective call stack?
Provides the option to include call stacks for selective conditions every interval.

WCHSTK1 - Stack waiter/holder usecs
Harvest the call stack for any job/task held by another job/task longer than N usecs. This value is ignored if parameter WCHSTKBLK is *NO.

WCHSTK2 - Stack selected waits usecs
Harvest the call stack fo any job/task in a "bad wait" longer than N usecs. This value is ignored if parameter WCHSTKBLK is *NO.

WCHSTKFLT - Watch page faulting stack?
Harvest the call stack for any job/task in a "page fault" or "fault pending" state every interval. This value is ignored if parameter WCHSTKBLK is *NO.

WCHSQL - Watch active SQL statements?
Harvest "in progress" SQL statements every interval.

WCHACT - Ignore actions to iDoctor jobs
Ignore any actions based on rules that are satisfied by an iDoctor collection job (QPYSWJOB, QPYJWJOB, IDOCCOL). These are the job names used by the GUI when creating PEX collections in PEX Analyzer and Job Watches in Job Watcher. The default is *YES.

WCHSRV - SERVICE USE ONLY
Provides the ability to remap a specific ENUM to a wait bucket and change a wait bucket description.

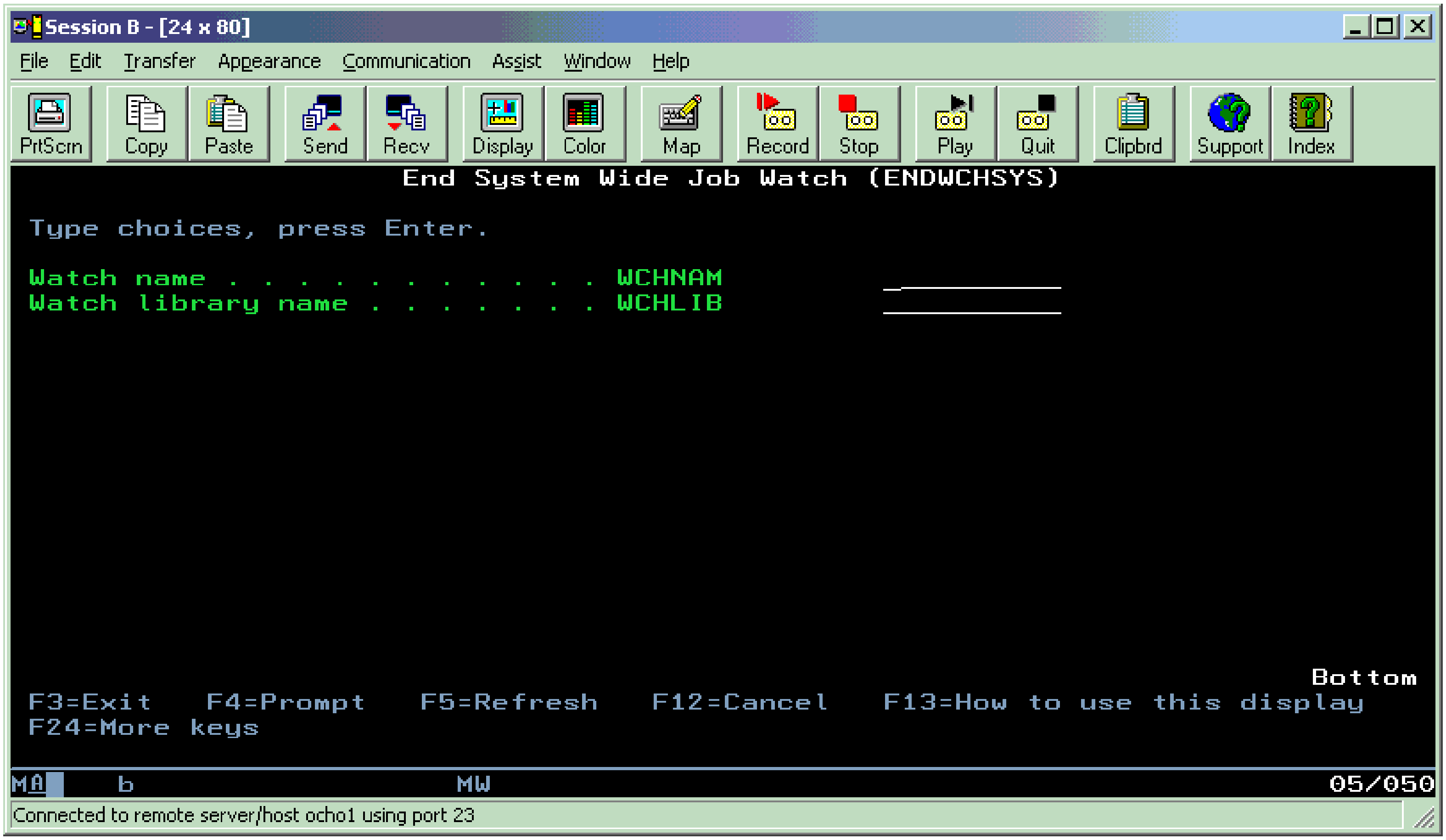
WCHLIMIT - Maximum collection size (in megabytes)
This value indicates the maximum number of megabytes of Job Watcher data that should be collected. If the total size of all job watcher files exceeds the value specified by this parameter, the Job Watch will end prematurely.

RESERVED1 - SERVICE USE ONLY

Table of Contents | Previous | Next

2.3.2 ENDWCHSYS - ends a job watch

The ENDWCHSYS command stops an active system-wide Job Watch. System-wide job watches run under the job name QPYSWJOB. Simply provide the name of the system-wide Job Watch and the library it is being created in to end it.

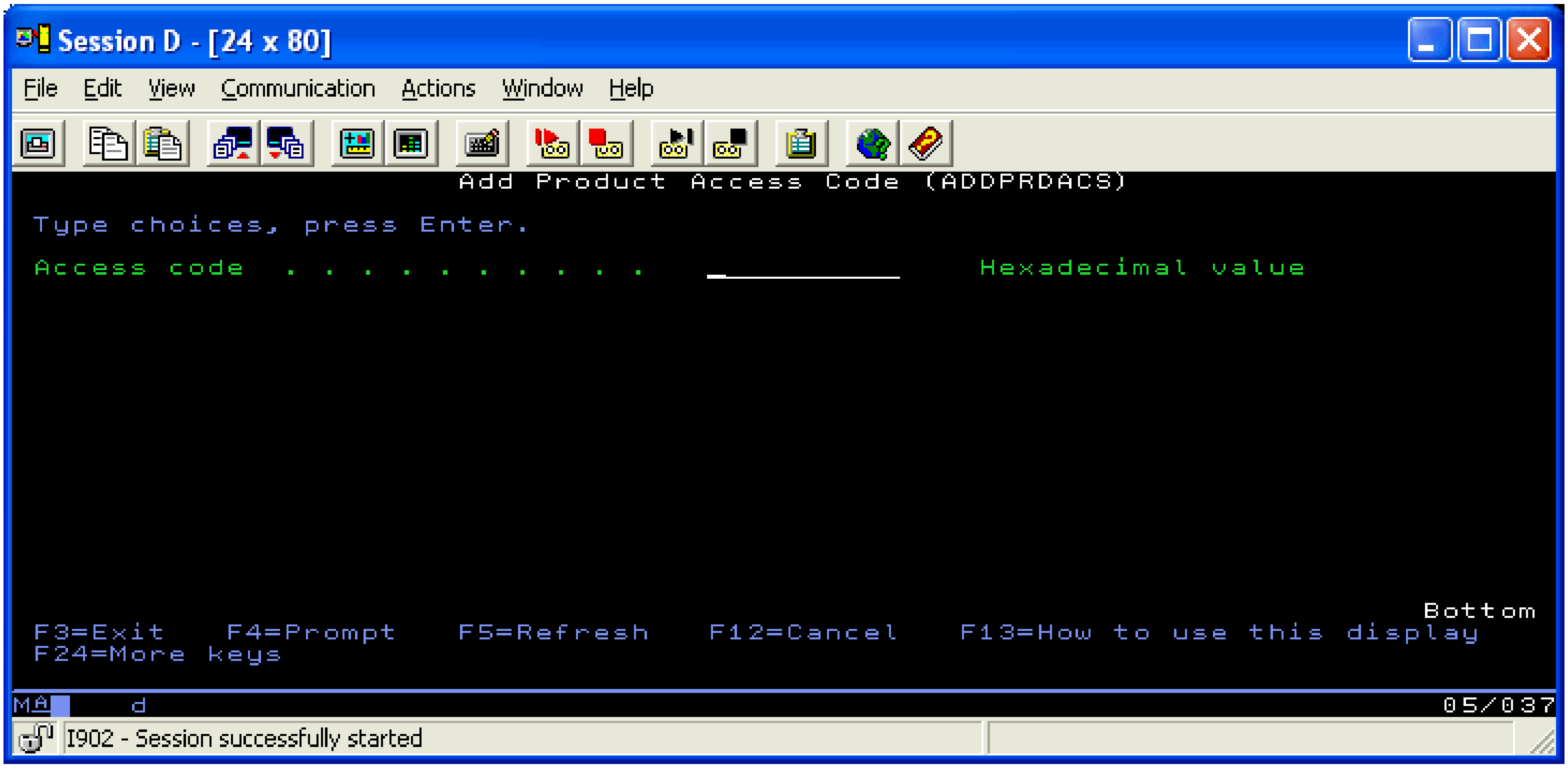

Session B - [24 x 80]
File Edit Transfer Appearance Communication Assist Window Help
PrtScrn Copy Paste Send Recv Display Color Map Record Stop Play Quit Clipbrd Support Index

```
                         End System Wide Job Watch (ENDWCHSYS)

 Type choices, press Enter.

 Watch name . . . . . . . . . . . . WCHNAM         __________
 Watch library name . . . . . . . . WCHLIB         __________

                                                                        Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
MA     b                    MW                                           05/050
```

Connected to remote server/host ocho1 using port 23

2.3.3 ADDPRDACS - apply an access code

This command is used to apply any iDoctor for iSeries component's access code to the current system. The access codes are provided by IBM through this website for either a trial or licensed version.



```
                         Add Product Access Code (ADDPRDACS)

 Type choices, press Enter.

 Access code . . . . . . . . . .   _________       Hexadecimal value


                                                                          Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
MA      d                                                                 05/037
```

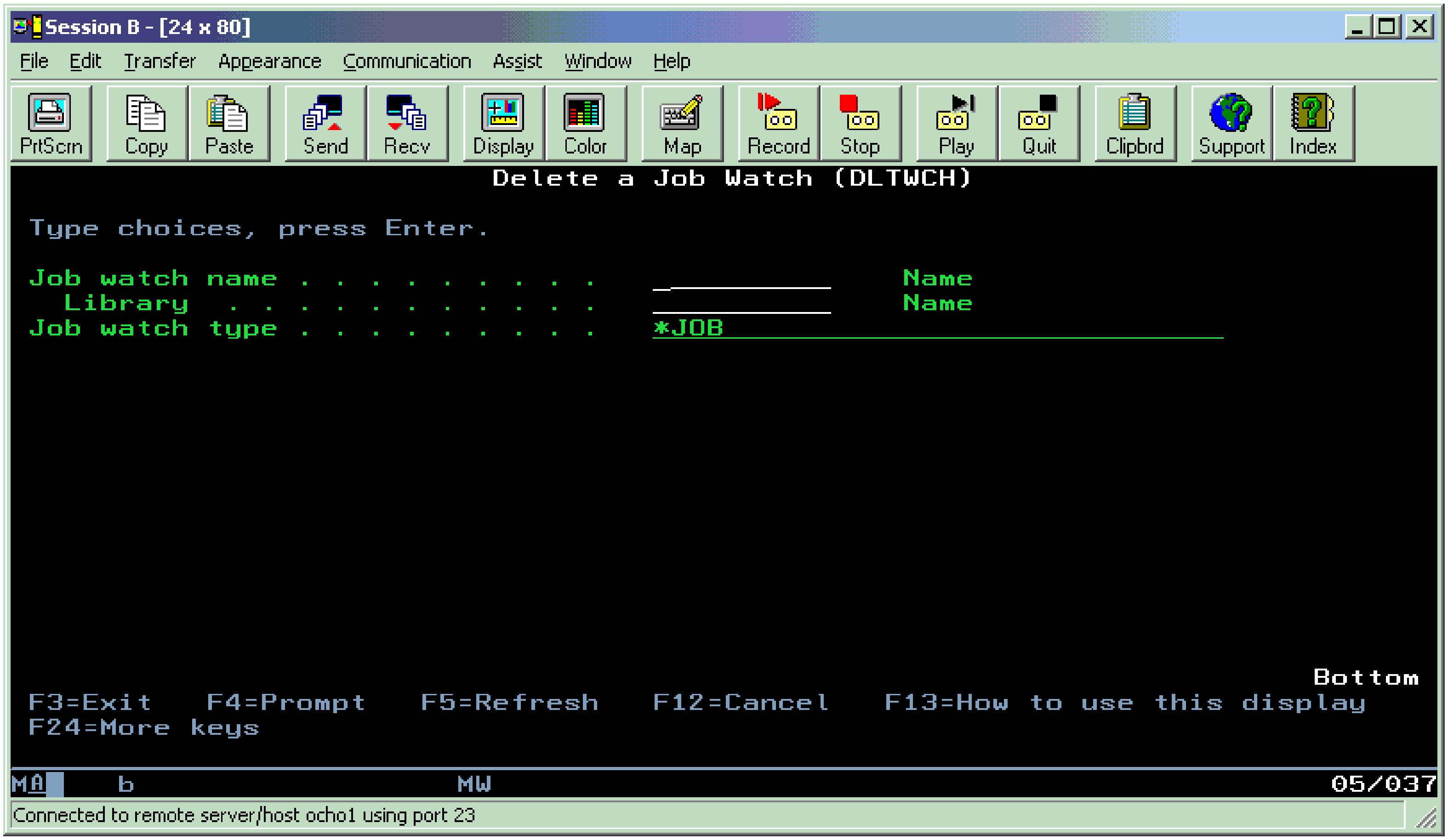
I902 - Session successfully started

Table of Contents Previous Next

2.3.4 DLTWCH - delete a job watch

The DLTWCH command removes a Job Watch from a users library. To use this command simply specify the Job Watch name (member name) , library name and the type of Job Watch *JOB or *SW.

Specifying the correct type of Job Watch to delete is critical because each type of Job Watch stores data in a different set of files.


Session B - [24 x 80]
File Edit Transfer Appearance Communication Assist Window Help
PrtScrn Copy Paste Send Recv Display Color Map Record Stop Play Quit Clipbrd Support Index

```
                        Delete a Job Watch (DLTWCH)

 Type choices, press Enter.

 Job watch name . . . . . . . . .   __________      Name
   Library  . . . . . . . . . . .   __________      Name
 Job watch type . . . . . . . . .   *JOB______________________________

                                                                        Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
MA     b                    MW                                         05/037
```

Connected to remote server/host ocho1 using port 23

Table of Contents Previous Next

2.3.5 JWAFTERIPL - resets the LIC wait buckets

The JWAFTERIPL command resets the LIC wait buckets for use by Job Watcher. This must be done at least once after an IPL has been performed. It is not necessary to run this command manually if you are using the client because the client will execute this command under the covers at startup.

This command has no parameters.

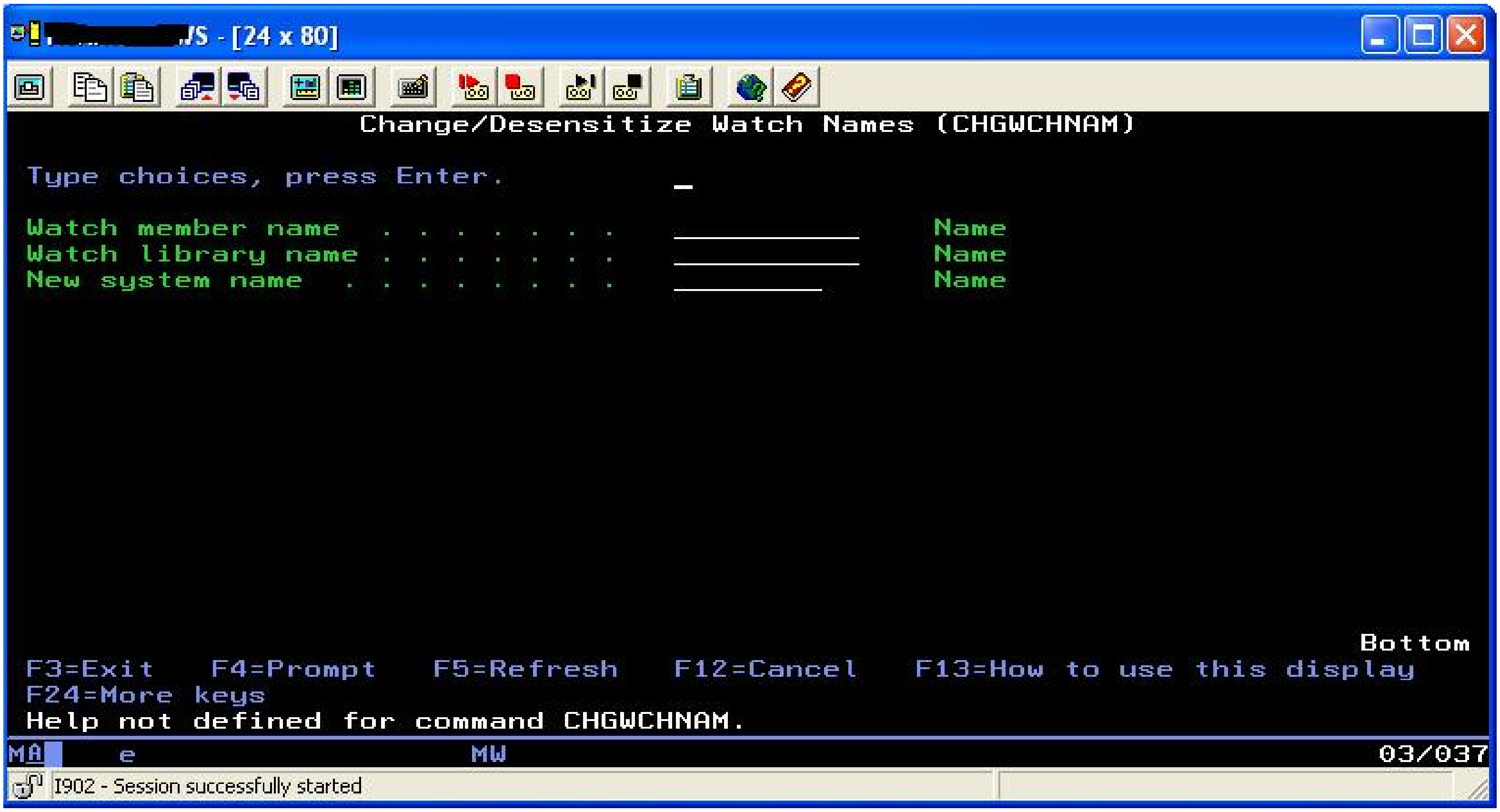
Table of Contents | Previous | Next

2.3.6 CHGWCHNAM - desensitize customer names

This command is used to replace any customer sensitive names (or IBM sensitive names) in the Job Watcher data for a system-wide collection. It does not work over a job-specific collection. The command will update any user profile names (other than standard QUSER profile names) in the Job Watcher data.

The new system name parameter will be used to replace any instances of the current system name in the Job Watcher data with the value provided. Program names are not modified.

File QPYRTSWNM1 is created by this command and keeps track of which fake names were used for the real names. This file should be not sent along with the rest of the collection.



```
                     Change/Desensitize Watch Names (CHGWCHNAM)

 Type choices, press Enter.

 Watch member name . . . . . . . .   __________     Name
 Watch library name  . . . . . . .   __________     Name
 New system name . . . . . . . . .   ________       Name

                                                                         Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
 Help not defined for command CHGWCHNAM.
MA     e                    MW                                           03/037
```

Table of Contents Previous Next

2.4 Rules and actions

Rules, in the context of a system-wide Job Watch, allow the WCHSYS command to test a user-specified set of conditions and take actions when those conditions are met. Actions taken can be one or more of the following:

- log the occurence to the QPYRTSWLG1 file
- start a Job-specific Job Watch over the job(s) that met the rules
- start a Job-specific Job Watch over the holding job(s) of the job(s) that met the rules
- execute a command (not supported in the GUI)

In order to define the rules for a System-wide Job Watch, the recommended method is to follow the prompts in the Job Watch Wizard within the client. Information on doing this is covered in the chapter on using the Job Watcher Wizard in the client documentation.

For green screen users, there are parameters on the WCHSYS command that may be used to define a rule for a system-wide Job Watch. These parameters give the user the option to use a rule definition file or not. If the rule file is not used up to 4 conditions/actions may be specified on the command.

A rule definition file is also supported using the WCHSYS command. Create a member in file QPYRTSWRD in the library of your choice and specify this file on the WCHSYS command in order to use a rule definition from the green screen. An example of the format needed for this file is located in file QPYRTSWRD in library QPYRTJW.

Table of Contents | Previous | Next

Chapter 3 PEX Analyzer

This chapter describes how to collect Performance Explorer (PEX) data for analysis with iDoctor for iSeries PEX Analyzer. This chapter includes:

- A brief overview of PEX
- Libraries, QYPINT, SMTRACE and QYPBASE
- The method and command syntax needed to collect PEX data
- Descriptions of green-screen commands shipped with PEX Analyzer

Table of Contents | Previous | Next

3.1 Performance Analysis with PEX

Performance Explorer (PEX) is a set of performance collection and reporting functions and commands that became available on RISC versions of OS/400, 5716-SS1 and the iSeries Performance Tools/400, licensed program product (LPP) 5716-PT1. PEX is the above-the-Machine-Interface portion and is the term most often used in this and other related documentation. Another term that might be seen is **Performance Data Collector** (PDC). PDC is the below-the-Machine-Interface (Licensed Internal Code) layer that supports PEX functions.

The PEX functions, which are part of a total iSeries performance management methodology, should **not** be the first set of performance tools used when analyzing a performance problem. PEX performance analysis is a detailed, **bottom up** approach. An overall picture of system or application performance should be known (using the GO PERFORM functions of the Performance Tools/400 licensed program) before using PEX for detailed analysis.

The PEX **collection** functions and related commands are included as part of the base OS/400 support.

The base PEX **reporting** function and its associated command are part of the Performance Tools/400 LPP, manager feature 5101.
This separation of PEX definition and collection capabilities (under OS/400) and reporting capabilities (under the Performance Tools/400 licensed program) is similar to using OS/400 Performance Monitor support to collect performance data and the Performance Tools/400 licensed program to generate reports from the data collected using Performance Monitor. The data collected using Performance Monitor and the data collected using PEX are placed in multiple OS/400 database files.

The iDoctor for iSeries component **PEX Analyzer** provides graphical visualization of PEX collection data. Previously, the Print PEX Report (PRTPEXRPT) command (part of the Performance Tools/400 LPP) was the typical method used to view reports over PEX data. PEX Analyzer is a value-add to PRTPEXRPT by providing more flexible interfaces for viewing potentially large and complex data.

PEX supports the following independent methods, or modes, of data collection:

STATS: Gathers CPU and paging information that is summarized on the fly about MI program invocation and complex MI instruction invocation. For those familiar with CISC-based iSeries performance tools, STATS mode is the replacement for TPST (Time and Paging Statistics). PEX STATS is focused on identifying application and IBM programs or modules consuming excessive CPU utilization or performing a high number of disk I/O operations. STATS is **program-oriented**.

PROFILE: Gathers CPU usage profile information over a select set of programs. For those familiar with CISC-based AS/400 performance tools, PROFILE mode is the replacement for SAM (Sampled Address Monitor). PEX PROFILE support is focused on identifying high-level language (HLL) program hot spots (high CPU consumption) based on source program statement numbers. Typically, PEX STATS is used to identify programs that should be investigated further as potential performance bottlenecks. PEX PROFILE can then be used on the programs identified for further analysis.

TRACE: Gathers detailed, information that is not summarized about a wide variety of **events**.

The iDoctor for iSeries client component PEX Analyzer provides graphical data management and visualization for a subset of PEX **TRACE**, **PROFILE** and **STATS** data.

Types of Data That Can Be Collected by PEX and Analyzed by PEX Analyzer

PEX (and the underlying PDC) provide approximately one hundred different types of TRACE events. While all of these can provide interesting performance-related and nonperformance-related perspectives, only a subset of these TRACE events currently are **available** within PEX Analyzer.

The following PEX TRACE events can be analyzed within PEX Analyzer:

- Activation group events
- Logical DASD requests
- Physical DASD operations
- DASD **space** change requests
- Local (non-DDM) data base file full OPENs and CLOSEs
- Local (non-DDM) data base file logical I/Os
- Local and remote (DDM) change and retrieve data area
- Local and remote (DDM) send and receive data queue
- MI program CALLs/RETURNs
- Task switch events
- Websphere events

Note: The MI CALL/RETURN events are optionally used to assign responsibility of the occurrences of the other event types to specific system or application programs.

Why Use PEX Analyzer to Analyze a System or Application

At a high level, the following may result in the need to analyze a system or application:

- Some other command or tool indicates a possible problem. PEX Analyzer can be used to provide more information.
- As part of a general system clean-up to investigate possible inefficient operations.

The following reviews the event types and describes briefly how each could be used.

The first, **Logical DASD requests** and **Physical DASD Operations,** work in tandem to provide detailed information about DASD operations. The information gathered in each event includes the following:

- A tag that indicates the job or system task requesting the operation
- The time of day the operation started (with microsecond granularity)
- The virtual or real address involved
- The name of the object (or LIC segment) involved
- Type of operation (sync-read, async-read, sync-write, and so on)
- DASD unit involved
- DASD sector numbers involved

- LIC and/or MI programs requesting the operation
- Number of DASD sectors read or written
- Main store pool involved
- Elapsed time the operation took to complete (microsecond granularity) (Optional)

This information can be gathered about **every** DASD operation that occurs during a PDC/PEX TRACE. It is a very detailed source of DASD operation information. This information could provide additional information on questions or problems. For example:

- Some DASD units are much busier than others (what objects or jobs are paging on them?)
- After adding a new application, all of the DASD units are **too busy**. What exactly are they doing?
- Sometimes the Work with Disk Status (WRKDSKSTS) command and performance monitor data disagree on **percent busy**. What is the real DASD response time? What are the average and maximum times?
- Would a Save/Restore of a file help in its distribution across DASD units?

DASD space change request events gather the following information about **every** DASD space change request during a trace:

A tag that indicates the job or system task requesting the change

Time of day the change was requested (with microsecond granularity)

- The virtual or real address involved
- The name of the object (or LIC segment) involved
- Type of operation (create, extend, truncate or destroy)
- LIC and/or MI programs requesting the change
- Size of the space being consumed or freed on DASD

Like the DASD operations above and all the other types of TRACE events, DASD space change events provide a tremendous amount of detailed information. These events are used primarily to investigate how DASD free space is being consumed. It is important to note that this can determine only **changes** to DASD space allocations. It does **not** statically determine where objects are placed or the size of objects. It does, however, determine how new objects are being created and how old objects are extended.

Local (non-DDM) data base file full OPENs and CLOSEs gathers the following information about every local data base full file open:

- A tag that indicates the job requesting the OPEN or CLOSE
- Time of day (microsecond granularity)
- File, library, and member name involved

Users who work on iSeries application analysis should not need usage examples for this, and should be anxious to use it.

Local (non-DDM) data base file logical I/Os (LDIOs) is probably the most valuable application-analyzing event type. It gathers the following:

A tag that indicates the job requesting the LDIO

- Time of day (microsecond granularity)
- File, library, and member name involved
- Detailed operation type information (get-by-get-nodata, get-by-key-data, put-single, put-multiple, and so on
- Record-not-found and end-of-file indication
- First relative record number involved
- Number of records retrieved

Note: This information is gathered about **every** non-DDM LDIO that occurs during the trace.

Additional Details Available with the Database OPEN/CLOSE and LDIO Trace Events

As mentioned above, PEX Analyzer can use a combination of trace events to derive additional details about certain events. If a TRACE includes MI CALLs/RETURNs and the two database events, PEX Analyzer reports the following:

- Elapsed time each OPEN, CLOSE, or LDIO took (microsecond granularity)
- CPU time each OPEN, CLOSE, or LDIO took (microsecond)
- Number and type of physical DASD operations each OPEN, CLOSE, and LDIO caused
- Causing program name (with an attempt to determine application versus system program name)

Local and remote (DDM) change and retrieve data area is similar to the database events. It collects information on:

- Tag that indicates the job requesting the data area operation
- Time of day (microsecond granularity)
- Data area and library names
- Detailed operation type information (change local, change DDM, retrieve local, and so on)
- Data area type (char, decimal, logical)
- Up to the first 20 bytes of the changed/retrieved data is also collected (although this is not reported on by PEX Analyzer)

Like the database events, the data area events can be combined with MI CALLs/RETURNs to derive **additional details**; for example:

- Elapsed time each data area operation took (microsecond granularity)
- CPU time each data area operation took (microsecond)
- Number and type of physical DASD operations for each operation
- Causing program name (with an attempt to determine application versus system program name)

Local and remote (DDM) send and receive data queue is also similar to the database events. It collects the following information:

- Tag that indicates the job requesting the data queue operation
- Time of day (microsecond granularity)
- Data queue and library names
- Detailed operation type information (send local, send DDM, receive keyed local, receive keyed DDM, and so on)
- Key length, message length, message, and key data are also collected, but not reported on by PEX Analyzer

Like the database events, the data queue events can be combined with MI CALLs/RETURNs to derive the following **additional details:**

- Elapsed time each data area operation took (microsecond granularity)
- CPU time each data area operation took (microsecond)
- Number and type of physical DASD operations for each operation
- Causing program name (with an attempt to determine application versus system program name)

Note: PEX Analyzer analyzes PEX data that has been previously collected on the current or some other iSeries system. The following is an obvious statement, but needs to be made: **PEX Analyzer can analyze only data that has been collected**. For example, if the PEX collection did **not** include database file activity, PEX Analyzer cannot produce reports on database file activity. It is important to understand that it is at the time of PEX data **collection** that the decision is made concerning which types of data will be collected. The decision about what data is collected controls the type of analyses available.

PEX Analyzer analyses can be used to do additional job and program performance analysis. In cases where other performance tools usage cannot identify an application, set of jobs, or programs, PEX Analyzer can be used system-wide to assist in identifying applications or programs that need further analysis.

Table of Contents | Previous | Next

3.1.1 PEX Background Information

Performance analysis based on PEX Trace is designed to answer questions at a level lower than the iSeries Performance Monitor reports can provide.

Note: The following examples refer to a particular type of PEX TRACE event: DASD operations. This is just one of many PEX TRACE types that can be analyzed with iDoctor for iSeries tools.

For example, the Performance Monitor can report that DASD unit U is performing X operations per second with an average I/O response time of T. PEX trace of DASD operations gathers detailed data about **every** DASD operation. Specifically, it can report the following:

- What objects were being paged into and out of main storage/DASD
- What object **types** were being paged into and out of main storage/DASD
- How many DASD sectors were read/written on each operation
- What job or system task was performing the DASD operations
- How long individual DASD operations took to complete
- What DASD units and main storage pools were involved with the DASD operations

A typical scenario involving the Performance Monitor and PEX trace includes the following:

- Performance Monitor sample data shows that between 7:05 and 7:30 PM (on most days) DASD units 45, 46, and 54 are much busier than all other units
- A PEX trace collection is made between 7:10 and 7:20 PM on 1 or 2 typical days. The PEX definition and resulting collection include the following types of trace events:
  -- Logical DASD/Main storage operation starts
  -- Physical DASD operation starts
  -- Possibly all MI program CALLs/RETURNs
- The PEX collection is analyzed on the collecting machine or Save/Restored to a different AS/400 system at the same version and release.
- The iDoctor for iSeries tool is used to analyze and graph such items as:
  -- DASD operations by disk unit and causing jobs/tasks
  -- DASD operations by paging object names
  -- DASD operations by paging object types
  -- DASD operations by DASD unit

In summary, a graphical presentation is created showing what jobs/tasks and objects are making units 45, 46, and 54 busier than usual.

Table of Contents | Previous | Next

3.2 Overview of Collecting PEX Data

The basic steps for creating a PEX Collection are the following:

1. Creating or choosing an existing **PEX definition**(explained in Section 3.4, PEX Definitions).
2. Starting a collection using the definition.
3. Ending the collection and directing the data to be discarded or stored in a library.

If an iSeries system has the iDoctor for iSeries PEX Analyzer component installed, the user has the following choices when making a PEX collection:

- The **manual** method of using the individual QSYS commands including the Add PEX Definition (**ADDPEXDFN**), Start PEX, (**STRPEX**), and End PEX (**ENDPEX**) commands.
- The **automatic** method by using the single command named STRIDOCCOL in library QYPINT (see Section 3.5, Collecting and Transporting Data).

If you are not interested in the details of what a PEX definition is, and want to know enough to make an appropriate PEX collection (for the current problem), skip to Section 3.5.

Note: The PEX Analyzer client provides interfaces for creating and/or changing PEX definitions. The client also has an interface for creating a collection, viewing it as runs, and then analyzing its results. See the client side documentation for PEX Analyzer for more information.

Table of Contents | Previous | Next

3.3 PEX Definitions

It is not necessary to understand how to specifically create a PEX definition before reading this section. Section 3.3 describes a command (STRIDOCCOL) used to create and use PEX definitions for use by iDoctor for iSeries PEX Analyzer.

A **PEX definition** is a member in a specific system database file, and it controls most of the aspects of making a PEX collection except the following:

- When to begin making the collection
- When to end the collection
- The library where collection data is stored

A PEX definition also controls:

- The type of collection to create: Trace, Profile or Statistical
- Which subset of the hundreds of event types are to be activated
- Granularity of CPU sampling
- Maximum amount of data to be collected
- Subset of jobs or system tasks (or all jobs and all tasks) that are to be traced

To build a PEX definition, use the Add PEX Definition (**ADDPEXDFN**) command in library QSYS.

This document does not go into detail of the very complex **ADDPEXDFN** command. However, the types of controls and data that are contained in a PEX definition will be described.

The **ADDPEXDFN** command and resulting PEX definition contain keywords and data that control the following:

- Definition name
- What subset (or all) of jobs and/or system tasks are to participate in the tracing
- Maximum amount of DASD space that should be consumed to hold the trace event data
- A CPU profiling interval (in milliseconds)
- The exact set of specific trace events that are to be active (more detail below)

There are dozens of PDC/PEX event types. A subset of them are analyzed by iDoctor for iSeries PEX Analyzer. Some of the event types known by iDoctor for iSeries are:

- MI program CALLs and RETURNs
- CPU profiling tics

- Logical DASD I/O start requests
- Physical DASD operation starts and completions
- DASD space change requests
- Database file full OPENs and CLOSEs
- Logical database file I/Os
- Data area change and retrieves
- Data queue sends and receives
- Task switch trace
- IFS events

Note: The PEX Analyzer client provides interfaces for creating and/or changing PEX definitions. The client also has an interface for creating a collection, viewing it as runs, and then analyzing its results. See the client side documentation for PEX Analyzer for more information.

Table of Contents | Previous | Next

3.4 Libraries

When PEX Analyzer is installed, three libraries are created on the iSeries. These libraries are named QYPBASE, QYPINT and SMTRACE. QYPBASE contains programs and commands needed by the iDoctor for iSeries client. It also contains objects that are used by all other iDoctor for iSeries components (except PTDV).

Library QYPINT contains the collector command STRIDOCCOL and the other commands and programs needed for managing collections (deleting, copying, etc). Library SMTRACE contains programs and files used when running analyses.

Table of Contents | Previous | Next

3.5 PEX Analyzer collection files

This section describes the structure of the output files produced by PEX Analyzer that are created along with a PEX collection.

Table of Contents | Previous | Next

3.6 PEX collection files (from QSYS)

This section describes the PEX files for release V5R2. These files are used by PEX (STRPEX, ENDPEX commands) and are shipped with OS/400 and not shipped with PEX Analyzer.

Table of Contents Previous Next

3.7 Commands - library QYPINT

This section covers the commands provided with PEX Analyzer in library QYPINT

Table of Contents Previous Next

3.7.1 STRIDOCCOL - starts a PEX collection

Reminder:

- PEX collections can be made on one iSeries and saved or restored to another for analysis by iDoctor for iSeries. However, both systems **must** be at the same version/release (i.e. V4R5, V5R1, or V5R2)

The OS/400 commands used to make a PEX collection are:

Add PEX Definition (**ADDPEXDFN**): Creates a PEX definition
Start PEX (**STRPEX**): Start a PEX collection
End PEX (**ENDPEX**): End a PEX collection

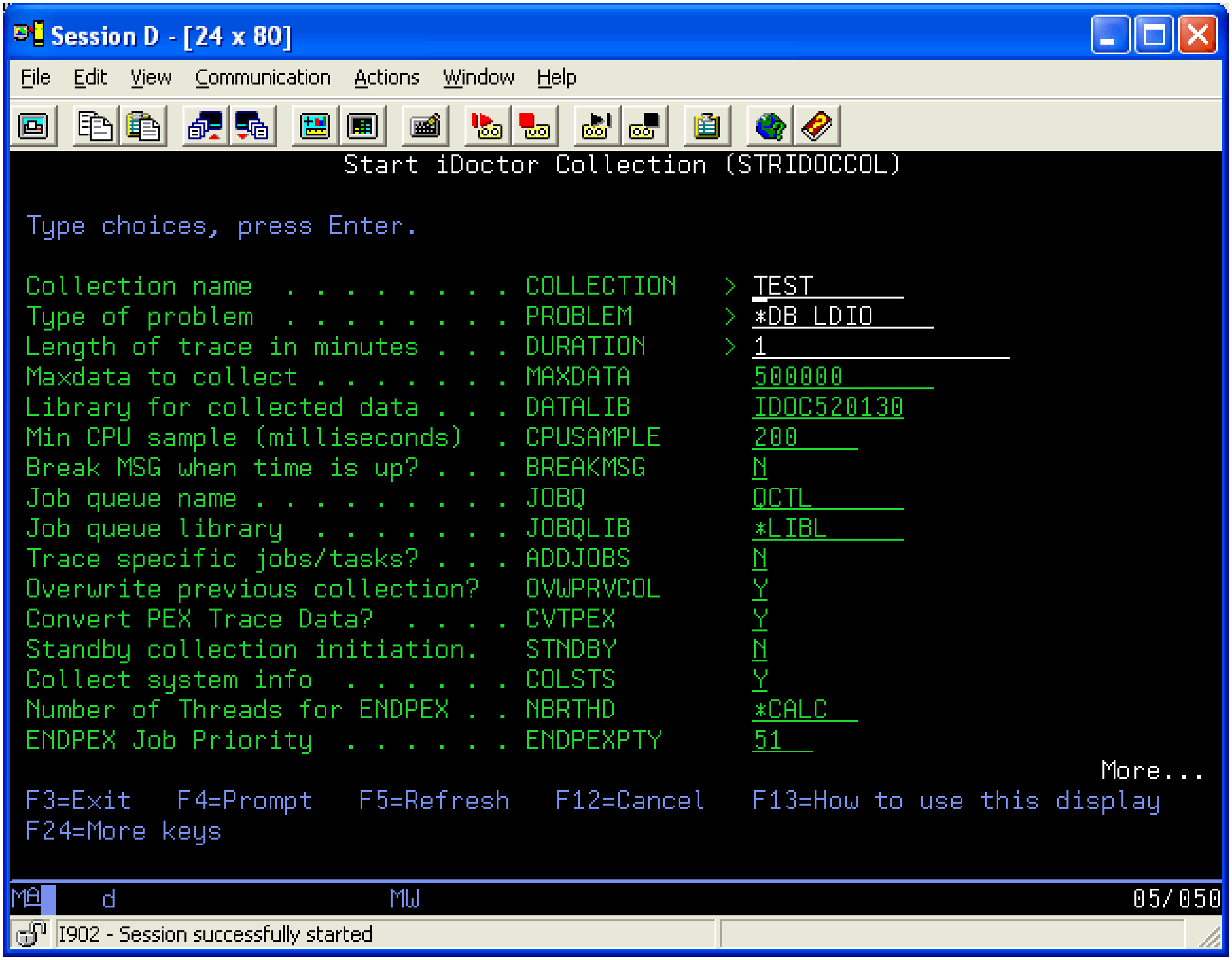
The above commands can be used; however, PEX Analyzer includes an ease-of-use feature that incorporates all of the above commands into a single command.
The command recommended for making a PEX collection is:

QYPINT/STRIDOCCOL

This command does the following:

- Creates the necessary PEX definition (deleting and replacing one by the same name, if necessary)
- Starts the PEX collection
- Ends the PEX collection after a predetermined amount of time and directs the data to a predetermined library
- Periodically issues the Work with System Status (**WRKSYSSTS**) and Work with Disk Status (**WRKDSKSTS**) commands and copies output of each to files named SMTRSTS and SMTRDTS in the target library

A prompt of the Start iDoc Collection (**STRIDOCCOL**) command is shown below:



```
                     Start iDoctor Collection (STRIDOCCOL)

 Type choices, press Enter.

 Collection name  . . . . . . . . COLLECTION    > TEST
 Type of problem  . . . . . . . . PROBLEM       > *DB_LDIO
 Length of trace in minutes . . . DURATION      > 1
 Maxdata to collect . . . . . . . MAXDATA         500000
 Library for collected data . . . DATALIB         IDOC520130
 Min CPU sample (milliseconds)  . CPUSAMPLE       200
 Break MSG when time is up? . . . BREAKMSG        N
 Job queue name . . . . . . . . . JOBQ            QCTL
 Job queue library  . . . . . . . JOBQLIB         *LIBL
 Trace specific jobs/tasks? . . . ADDJOBS         N
 Overwrite previous collection?   OVWPRVCOL       Y
 Convert PEX Trace Data?  . . . . CVTPEX          Y
 Standby collection initiation.   STNDBY          N
 Collect system info  . . . . . . COLSTS          Y
 Number of Threads for ENDPEX . . NBRTHD          *CALC
 ENDPEX Job Priority  . . . . . . ENDPEXPTY       51
                                                                       More...
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

STRIDOCCOL Command

It is recommended to **use the default values** in most cases.

The value specified for the PROBLEM keyword is used to build a PEX definition that collects the MINIMUM types of events necessary to enable report generation of that type. The less events collected, the longer the collection can be run.

When the PROBLEM parameter chosen results in a TRACE collection (as opposed to a STATS or PROFILE collection), it is possible to add to the events collected by use of the ADDEVTS keyword.

STRIDOCCOL Parameters

COLLECTION - name of the collection
The value provided here will be used to uniquely identify the PEX collection within the specified library.

PROBLEM - type of problem for which data is being collected
The value entered here causes a PEX definition to be created with predetermined values. The possible responses are:

***DB_OPEN**

Used to determine jobs or programs that are performing excessive file opens. This does not include IFS files.

Suggested maximum value for the DURATION parameter: 30 minutes

***DB_LDIO**
Used to determine the rates at which jobs or programs are performing logical data base I/O's. This does not include IFS files.

Suggested maximum value for the DURATION parameter: 10 minutes

***DTAQ_IO**
Allows you to determine the rates of logical I/O's against your Data Queues.

Suggested maximum value for the DURATION parameter: 30 minutes

***DTAARA_IO**
Allows you to determine the rate of logical I/O's against your Data Areas.

Suggested maximum value for the DURATION parameter: 30 minutes

***PDIOCOUNT**
Allows you to determine the rates of physical I/O's jobs for Jobs, Objects, Disk Units.

Suggested maximum value for the DURATION parameter: 15 minutes

***PDIOTIME**
Allows you to determine the rates of physical I/O's for Jobs, Objects, Disk Units, and the time values associated with these I/O's.

Suggested maximum value for the DURATION parameter: 10 minutes

***TASKSWITCH**
Allows you to determine how a job spends it's time within the system during the collection interval.

Suggested maximum value for the DURATION parameter: 5 minutes

***OBJ_NETSIZE**
Allows you to determine where storage is being used within the system during the collection interval.

Suggested maximum value for the DURATION parameter: 4 hours

***PROFILE**
User specifies from 1 to 16 programs and/or service programs for CPU profiling PEX PROFILE data collection. This collection type helps identify CPU 'hot spots' within the selected programs/service programs. The user specified pane size (default of 4) is used to logically divide each program/service program up into 'panes' for granularity of analysis. When CPU is used by a program/service program in the collection data is captured by pane within program/service program.

***STATSFLATY**
This PEX STATISTICAL collection captures CPU and IO statistics by program/procedure/method. Procedure/method statistics are only provided for programs/service programs with an ENDPFRCOL value other than *NONE or *PEP. One collection-wide count of both CPU and IO in-line and cumulative values is stored for every program/procedure/method that was called (program/procedure/method statistics for all jobs/tasks in the collection are merged).

***STATSHIER**
This PEX STATISTICAL collection captures call flow, CPU and IO statistics by program/procedure/method by calling program/procedure/method. Procedure/method statistics are only provided for programs/service programs with an ENDPFRCOL value other than *NONE or *PEP. One count of both CPU and IO in-line and cumulative

values is stored by calling program/procedure method for every program/procedure/method that was called by the job. We recommend that you try to keep the number of jobs in a collection of this type to less than 10.

***TPROF**
This PEX TRACE collection type helps identify system-wide CPU 'hot spots' within application programs/service programs, OS/400 programs/procedures/methods and SLIC procedures. Unlike *PROFILE, the user does not need to specify program names. The ENDPFRCOL setting is not used to determine whether a program/procedure/method will be recorded.

***STATSFLATN**
This PEX STATISTICAL collection captures CPU and IO statistics by program/procedure/method. Procedure/method statistics are only provided for programs/service programs with an ENDPFRCOL value other than *NONE or *PEP. One count of both CPU and IO in-line and cumulative values is stored for every program/procedure/method that was called by job (program/procedure/method statistics for jobs/tasks in the collection are stored separately for each individual job-thread or task). We recommend that you try to keep the number of jobs in a collection of this type to less than 10.

***TAPE_IO**
Collects information on commands issued to the tape device for use in diagnosing tape performance issues. This function can be run for an extended period of time. You may require the additional information provided by the *DASDSTR and *DASDEND options to diagnose the problem.

***ILEACTGRP**
Collects information on the creation of activation groups and the destruction of activation groups. This can aid you in determining if there are issues within your jobs normal execution affected by the creation or destruction, such as what is the performance cost associated with their creation/destruction, etc. This can be useful in squeezing out that last bit of application performance. In addition, for those problem types which collect trace type data, you will be presented with an additional parameter ADDEVTS(14), that allows you to extend the data collections capabilities.

***HEAPSTG**
Collects information about the creation and destruction of heap allocations on your system. This option can collect a lot of events in a very short period of time, particularly on systems running newer workloads.

***WEBSPHERE**
Collects information about WebApp/Servlet's, Container Transactions, EJB/EJB methods, Database Connection Pool, EJB Object Pool, and ORB Thread Pool. This option could collect a lot of events in a very short period of time. It is advisable to only run this trace for short periods of time, 5-10 minutes.

***USR_TRNSACT**
Collects information on CPU usage, response time, I/O activity, and Seize/Lock activity for those applications which are using the Transaction API's. The length of time you can collect data for will vary with the numbers of jobs running the application which is using the Transactions API's. Most likely you will be able to collect data for 20-30 minute intervals.

***IFSEVTS**
Collects information whenever an IFS file is accessed, and when an IFS file is opened or closed. The length of time you can collect data for will vary with the numbers of jobs over which data is being collected which are actually using the Itegrated File System.

***DOMINOTRC**
Collects information about Domino server activity. The data which is collected is based on the value supplied to DEBUG_OS400_PEX in the NOTES.INI file. The length of time you can collect data for will vary with the number of Domino server jobs included in the trace collection and the kind of data which is being collected.

These terms were used in the previous information:

LDIO (Logical Data Base I/O): Actual calls to OS/400 database run-time routines made by any MI program (application or system). Example: GET-BY-KEY-DATA, GET-BY-GET-NODATA, PUT-SINGLE, PUT-MULTIPLE, and so on. This term is also expanded within PEX Analyzer to include data area and data queue operations. For example, a program calls to QSNDDTAQ or the issuing of the CHGDTAARA commands are termed data queue LDIOs and data area LDIOs.

PDIO (Physical DASD I/O): Physical (real) DASD reads and writes.

DURATION - Length of trace in minutes
The supplied value, in minutes, is used to control the amount of time spent collecting events. This is the time from STRPEX to ENDPEX. The job this command submits, IDOCCOL, will run for a much longer time than the value supplied on the DURATION parameter because this job is doing the ENDPEX function, which can be long running.

This is probably the most difficult keyword to explain and is the most difficult to give a recommended value for. The size of a collection and how long it takes to process are directly proportional to the number of events in a collection. In turn, this number is controlled by:

- Type of events that are active
- How long the collection is active
- How **busy** the machine is

General DURATION Recommendations

The following are based on tracing large (8+ processors), busy, machines.

Note: The term **busy** is used to describe how much work a system is doing. Contrast a system with 2000 interactive users at lunch time versus a night time batch run with 4 active batch jobs. It is more likely that the batch run would be busier than the lunch time interactive run.

This information is also based on the assumption that **all** jobs and system tasks are participating in the trace. If a single job or task, or even a small subset, is traced in place of all jobs and tasks, the recommended DURATION can be increased.

Of all the PROBLEM types, *OBJ_NETSIZE can run the longest. This is intentional. It could take a few hours to detect a problem in DASD space usage. **A run of 4 hours for *OBJ_NETSIZE is reasonable.** If the DASD space problem is detected in less time, run the trace for the time required.

On the other extreme, the *TASKSWITCH can produce trace events at a particularly high rate. **A run of 2 minutes for *TASKSWITCH is reasonable.**

For the remainder of the PROBLEM types, a duration of 5 to 10 minutes is reasonable. A longer time may be required for types that collect less data (for example, *PDIO_COUNTS with ADDEVTS(*NONE) . A shorter time may be sufficient for types that collect more data (*DB_LDIO with ADDEVTS(*DASDSTR *DASDEND)).

The addition of ADDEVTS can dramatically increase the number of trace events logged per second.

DURATION Considerations

Similar to **STRPFRMON ... TRACE(*ALL)**, the submitted job running the **STRIDOCCOL** functions uses little resources (CPU, DASD I/O) during the actual trace collection. However, when the DURATION expires, (or the **ENDIDOCCOL** command is used, see below) and the trace data is being surfaced into DB2/400 PEX database files, the job can and will use considerable resources. In addition, the elapsed time the submitted job runs can be determined.

MAXDATA -Maximum amount of data to collect, in DASD kilobytes
The default amount of space in K byes to store PEX TRACE event data is 500000K. This allocates DASD space to record the trace data and should be sufficient for as long as a TRACE collection should be run. Assume roughly 80 bytes per trace event; therefore, 500000K gives approximately 500000 x 1024 / 80 = 6 400 000 events. When MAXDATA is reached, the PDC/PEX trace goes into **suspended** mode, and trace event collection halts. This does not hurt anything but can dilute the content of some of the data. In particular, reporting of object names and types can suffer. How? During trace, collection of events that contain object references only, the address of the object is captured when the trace record is cut. When the trace is stopped and the data is surfaced to PEX, a translation of address to name/lib/type occurs. **Therefore, the longer the time between the trace event is cut and the trace is formally ended, the more likely some objects will be destroyed in the meantime, therefore, eliminating the possibility of reporting about these by name.** When MAXDATA is reached, the trace is suspended rather than ended. This address to name/lib/type translation does not occur until the trace is formally ended. **Therefore, a big suspension time due to MAXDATA being reached may reduce the number of objects that are reported on by name.**

MAXDATA works hand-in-hand with DURATION.

Generally, use the default value of 500000.

More Detail on the MAXDATA Keyword

There is an important aspect to the MAXDATA keyword. This keyword controls the maximum number of DASD bytes that are used to hold trace event data **during trace collection**. Trace events are collected and stored by LIC programs into internal, LIC storage segments. These segments are limited by MAXDATA. However, during the ending of a trace the trace event data exists in the internal LIC storage **and** in PEX DB2/400 database file members. It is not until all the trace data is surfaced into PEX files that LIC frees the internal storage. The database format of the data takes slightly more bytes/event than the LIC format. Therefore, **count on a peak DASD storage demand of approximately double the MAXDATA value (assuming that MAXDATA is reached).**

DATALIB - Library for collected data
If a library name other than IDOCDATA is specified, the library is created if it does not already exist.

CPUSAMPLE - CPU sample time, in milliseconds
The value, in milliseconds, is used to cause a process to wake up at this inteval value, and indicate what program is executing and the percentage of CPU used between this wakeup and the previous wakeup within this job. By setting this to a low value, 1-10ms, you can profile the entire system, if all jobs are within the collection, and determine what programs are using the CPU.

This event data is collected regardless of the ENBPFRCOL setting for a particular pgm.

BREAKMSG - whether to issue a break message when the time is up
This keyword allows a user to decide after the DURATION if the data should be:

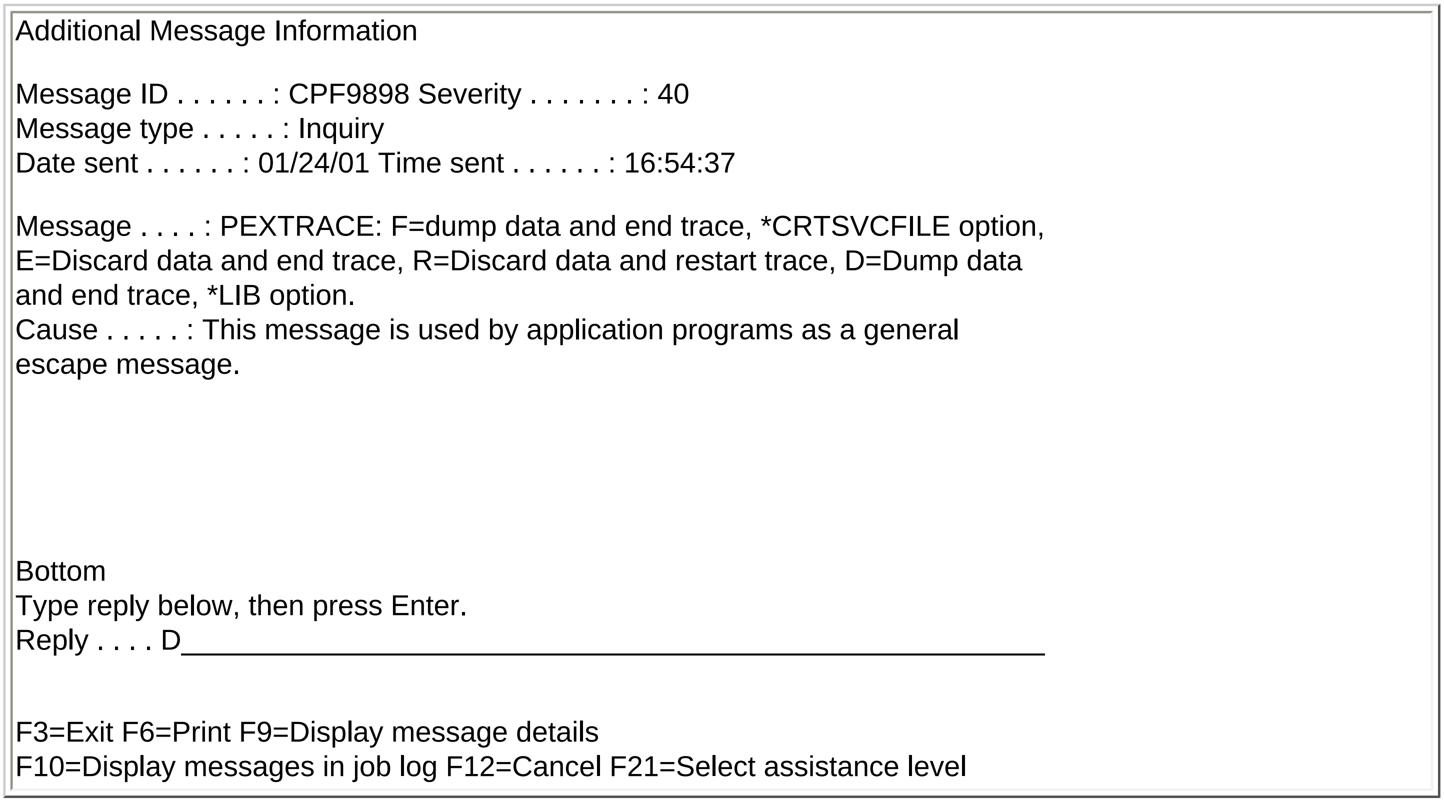
- Saved and tracing stopped
- Discarded and tracing stopped
- Discarded and tracing restarted

This keyword is useful when a problem occurs intermittently (busy DASD units, lost DASD space, and so on). If it does not, the choice can be made to discard the trace data and optionally start the trace again.

Normally, N (No) is specified for this keyword, thus permitting the PEX collection to complete without operator intervention. To be prompted when the trace DURATION has been reached, specify Y (Yes). If Y is specified, an inquiry message is sent to QSYSOPR when the trace DURATION is reached. The operator must respond with one of the following options:

F = End the trace and dump the data into a single file (**Note:** This is **not** the recommended format for the data if it is to be analyzed with PEX Analyzer. Rather, use D below.)

E = End the trace and discard the collected data.
R = Discard the data and start the collection again.
D = End the trace and dump the collected data into several database files; ready for transport and/or analysis.



```
Additional Message Information

Message ID . . . . . . : CPF9898 Severity . . . . . . . : 40
Message type . . . . . : Inquiry
Date sent . . . . . . : 01/24/01 Time sent . . . . . . : 16:54:37

Message . . . . : PEXTRACE: F=dump data and end trace, *CRTSVCFILE option,
E=Discard data and end trace, R=Discard data and restart trace, D=Dump data
and end trace, *LIB option.
Cause . . . . . : This message is used by application programs as a general
escape message.




Bottom
Type reply below, then press Enter.
Reply . . . . D__________________________________________________


F3=Exit F6=Print F9=Display message details
F10=Display messages in job log F12=Cancel F21=Select assistance level
```

JOBQ and **JOBQLIB** - job queue to be used for the batch job
These keywords control what job queue is used to for running the collection in batch.

ADDJOBS - Trace specific jobs/tasks
The default of **N** (no) ensures that you are tracing all jobs and all system tasks on the system.
Specify **Y** to type an individual job name/user/number to be collected; this causes a lower collection overhead than ADDJOBS(N).

Specifying **Y** causes a selective interactive prompt (of the QYPINT/ADDPEXDFN command) to be issued, which permits the user to direct the tracing to a subset of jobs and/or system tasks.

OVWPRVCOL - Overwrite previous collection
This option will allow the user to overwrite a previous PEX Analyzer collection with the same name in the target data library. The default of **N** (no) ensures that any existing collection data will not be destroyed unless **Y** (yes) is specified.

CVTPEX - Convert PEX Trace Data
With the default value of 'Y', the data at ENDPEX time is written to the necessary set of QAYPE* files which allows for execution of the PRTPEXRPT command and the ability to generate PEX Analyzer reports. If 'N' is supplied as the value, the data is written to a single file as a management collection object, which will later require that a CVTIDOCCOL command be executed against this object if any data analysis is to be performed.

STNDBY - Standby mode
With the default value of 'N', the start of data collection begins as soon as the start pex message is received. With a value of 'Y' the collection object is built and the collection is in suspended status waiting infinitely on a data queue. The command RSMIDOCCOL is used to cause a collection which has been initiated with this option, to wake up and begin collecting data.

COLSYS - Collect system information
With the default value of 'Y', WRKSYSSTS and WRKDSKSTS information is collected and saved into two files, SMTRSTS

and SMTRDTS in the collection library. The member names in each file will be the same as the collection name. This information can give you insight into what the system was doing during the collection of the PEX data. Change this value to 'N' if you do not want to collect this information. There are some circumstances where the collection start up time will be significantly reduced by changing this to 'N'.

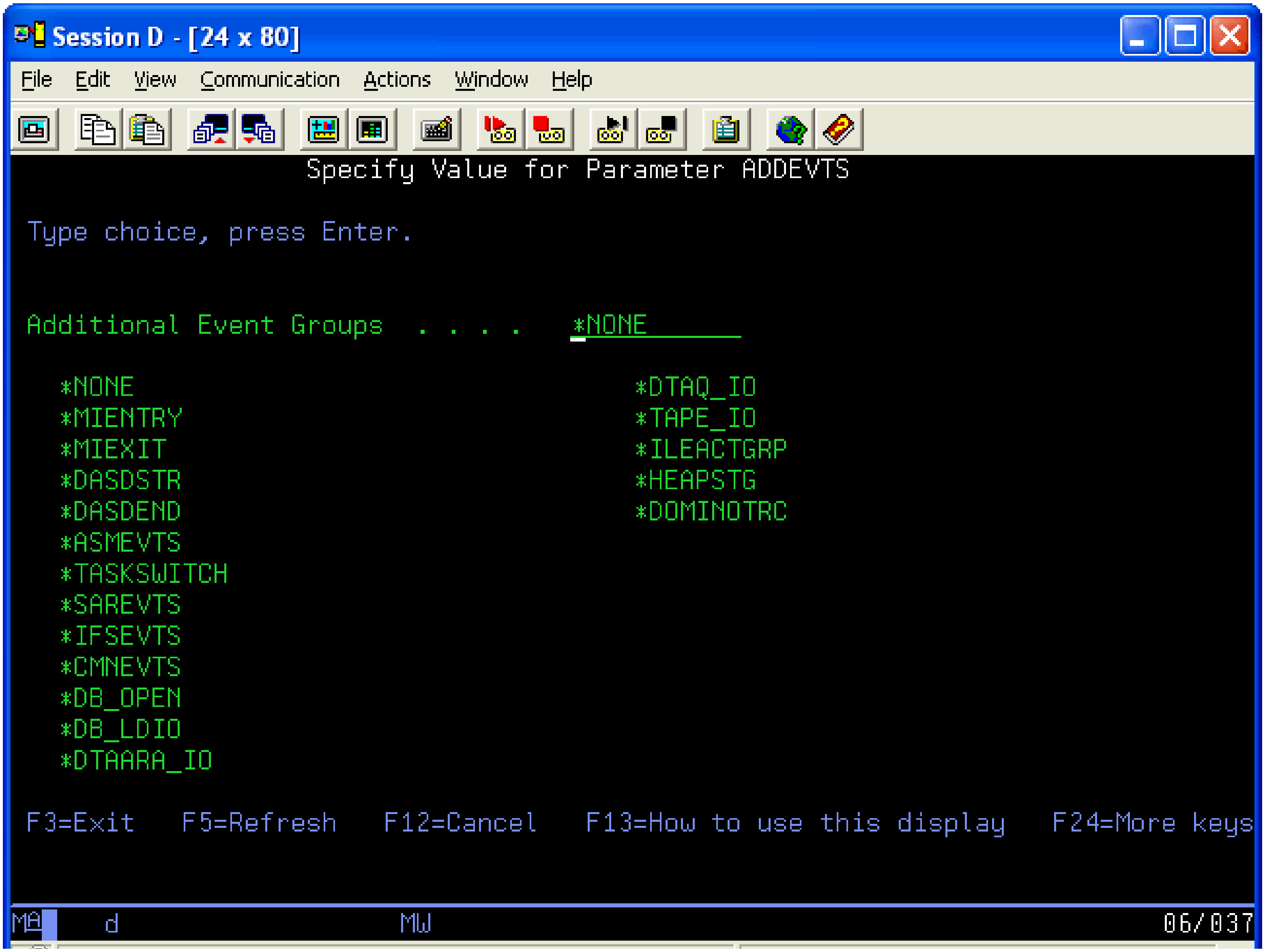
NBRTHD - Number of threads for ENDPEX
With the default value of '*CALC', ENDPEX will determine the an appropriate number of threads to use when the data is being dumped into the collection files. You can supply a value of between 1 and 256 depending on the impact that you want on the system when the data is being dumped.

ENDPEXPTY - ENDPEX job priority
This value controls the run priority of the job/threads created for use by the ENDPEX processing of the collected data. The default value is 51, change as you feel is appropriate.

PEXFTR - PEX Filter
Supply the name of a previously created PEX filter to enable more precise collecting of data.

TEXT - a description for this collection
Use this keyword to provide a more meaningful description of your collection.

ADDEVTS - Add Additional Events
Allow for the addition of extra events to beyond those normally collected for a specific problem type. Any event group or combination of event groups can be added to any problem type that results in a *TRACE type definition. This means that you cannot add these events to problem type that creates a *STATS or *PROFILE definition.



```
                    Specify Value for Parameter ADDEVTS

 Type choice, press Enter.

 Additional Event Groups . . . .   *NONE

   *NONE                                 *DTAQ_IO
   *MIENTRY                              *TAPE_IO
   *MIEXIT                               *ILEACTGRP
   *DASDSTR                              *HEAPSTG
   *DASDEND                              *DOMINOTRC
   *ASMEVTS
   *TASKSWITCH
   *SAREVTS
   *IFSEVTS
   *CMNEVTS
   *DB_OPEN
   *DB_LDIO
   *DTAARA_IO

 F3=Exit   F5=Refresh   F12=Cancel   F13=How to use this display   F24=More keys

MA     d                        MW                                       06/037
```

I902 - Session successfully started

Prematurely Ending an STRIDOCCOL Collection

When the **STRIDOCCOL** command is used to gather a PEX collection, the collection is normally active for the DURATION number of minutes. However, frequently it is necessary to end the collection early. For that purpose, the **ENDIDOCCOL** command is provided (also in library QYPINT).

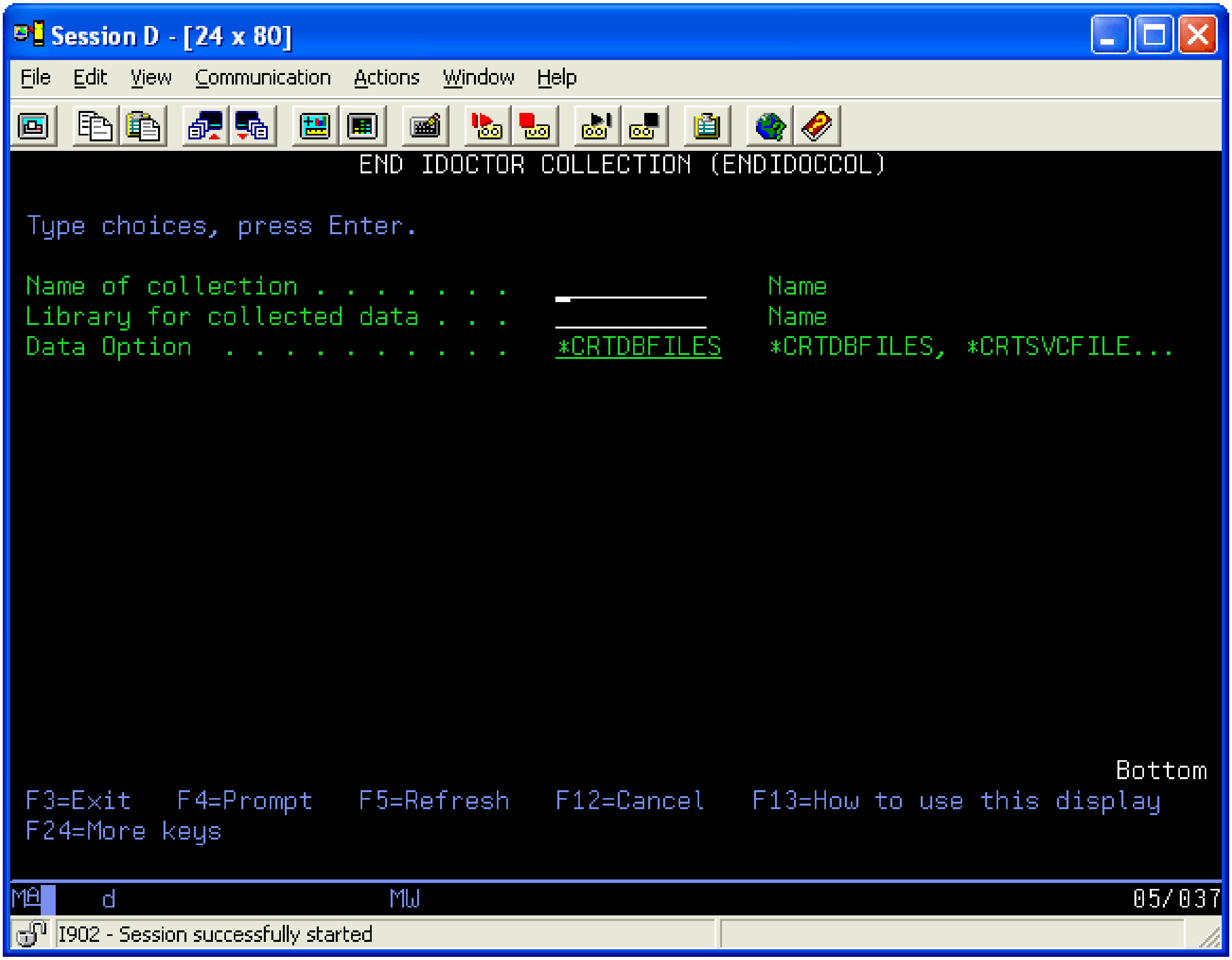
This command has two parameters, DATALIB, which must be the same library as the library that you supplied on the **STRIDOCCOL** command, and DATAOPT which controls what actions are to be taken on this collection.

DATAOPT keyword - data option

- ***CRTDBFILES** - Stop the collection immediately and save the PEX data into DB2/400 in a format ready for analysis.
- ***CRTSVCFILE** - Stop the collection immediately and save the PEX data into a management collection object for sending to IBM for analysis
- ***DELETE** - Stop PEX collection immediately without saving the data because the problem you are investigating did not occur.
- ***RESTART** - Stop PEX collection immediately, discard the data and immediately start another collection.

The **ENDIDOCCOL** command works only when the collection job (the submitted batch job named IDOCCOL or the interactive job running **STRIDOCCOL**) is still collecting data. If ENDPEX has already started the command will not work.

The command is shown below:


Session D - [24 x 80]
File Edit View Communication Actions Window Help

```
                      END IDOCTOR COLLECTION (ENDIDOCCOL)

 Type choices, press Enter.

 Name of collection . . . . . . .   __________    Name
 Library for collected data . . .   __________    Name
 Data Option  . . . . . . . . . .   *CRTDBFILES   *CRTDBFILES, *CRTSVCFILE...

                                                                         Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

MA   d   MW   05/037
I902 - Session successfully started

Transporting PEX Collection Data

Assuming the above recommendations are followed on the **STRIDOCCOL** command, the collection data will be stored into approximately 52 different DB2/400 database files starting with the prefix QAYPE*. Some of these files hold much more data than others. However, when saving or restoring this data from machine to machine, it is required that **all** the QAYPE* files be saved or restored. The easiest way to do this is to use the FTPIDOCCOL command to transfer a collection to another system. Another option would is to use the SAVLIB command or the SAVIDOCCOL command to save the collection to a save file and transfer the file to the other system manually. If SAVIDOCCOL is used then RSTIDOCCOL must be used on the target system to properly restore the collection from the save file.

Version/Release Considerations

Unfortunately, PEX data is version and release sensitive. You can only perform an analysis using the PEX Analyzer GUI on collections matching the VRM of the OS/400 VRM. However, you may view already created analyses for prior level collections using PEX Analyzer (look at already created V4R5 analyses on a V5R1 system for example).

Table of Contents | Previous | Next

3.7.1.1 Equivalent ADDPEXDFN commands for STRIDOCCOL problem types

This section contains the ADDPEXDFN command strings that may be used instead of the STRIDOCCOL command in order to collect the same PEX events as provided by the Problem Type(PROBLEM) parameter of the STRIDOCCOL command.

Note that for all of these commands including *MIENTRY/*MIEXIT events is often desirable in order to collect program information. Doing this will greatly increase the number of events collected within the collection however, and should generally not be done over all jobs on the system. If *MIENTRY/*MIEXIT events are not desired then do not specify PGMEVT(*MIENTRY *MIEXIT) on the ADDPEXDFN command. The JOB and TASK parameters should also be changed as appropriate to collect information for the desired jobs or system tasks. *MIENTRY/*MIEXIT events are not collected by default by the STRIDOCCOL command but they are included in the command strings listed here because usually customers will want to collect this information.

***DB_OPEN**
Used to determine jobs or programs that are performing excessive file opens. This does not include IFS files.
Suggested maximum collection duration: 30 minutes

```
ADDPEXDFN DFN(DBOPEN) TYPE(*TRACE) INTERVAL(200) MAXSTG(500000)
TRCFULL(*STOPTRC) TRCTYPE(*SLTEVT) JOB(*) TASK(*NONE ) SLTEVT(*YES)
MCHINST(*NONE) BASEVT(*PMCO ) PGMEVT(*MIENTRY *MIEXIT ) OSEVT(*DBOPEN )
```

***DB_LDIO**
Used to determine the rates at which jobs or programs are performing logical data base I/O's. This does not include IFS files.
Suggested maximum collection duration: 10 minutes

```
ADDPEXDFN DFN(LDIO) TYPE(*TRACE) INTERVAL(200) MAXSTG(500000)
TRCFULL(*STOPTRC) TRCTYPE(*SLTEVT) JOB(*) TASK(*NONE ) SLTEVT(*YES)
MCHINST(*NONE) BASEVT(*PMCO ) PGMEVT(*MIENTRY *MIEXIT ) OSEVT(*DBIO )
```

***DTAQ_IO**
Allows you to determine the rates of logical I/O's against your Data Queues.
Suggested maximum collection duration: 30 minutes

```
ADDPEXDFN DFN(DTAQ) TYPE(*TRACE) INTERVAL(200) MAXSTG(500000)
TRCFULL(*STOPTRC) TRCTYPE(*SLTEVT) JOB(*) TASK(*NONE ) SLTEVT(*YES)
MCHINST(*NONE) BASEVT(*PMCO ) PGMEVT(*MIENTRY *MIEXIT ) OSEVT(*DTAQ )
```

***DTAARA_IO**
Allows you to determine the rate of logical I/O's against your Data Areas.
Suggested maximum collection duration: 30 minutes

```
ADDPEXDFN DFN(DTAARA) TYPE(*TRACE) INTERVAL(200) MAXSTG(500000)
TRCFULL(*STOPTRC) TRCTYPE(*SLTEVT) JOB(*) TASK(*NONE ) SLTEVT(*YES)
MCHINST(*NONE) BASEVT(*PMCO ) PGMEVT(*MIENTRY *MIEXIT ) OSEVT(*DTAARA )
```

***PDIOCOUNT**
Allows you to determine the rates of physical I/O's jobs for Jobs, Objects, Disk Units.
Suggested maximum collection duration: 15 minutes

```
ADDPEXDFN DFN(PDIOCOUNT) TYPE(*TRACE) INTERVAL(200) MAXSTG(500000)
TRCFULL(*STOPTRC) TRCTYPE(*SLTEVT) JOB(*) TASK(*NONE ) SLTEVT(*YES)
MCHINST(*NONE) BASEVT(*PMCO ) PGMEVT(*MIENTRY *MIEXIT ) DSKEVT(*WRTSTR
*READSTR )
```

***PDIOTIME**
Allows you to determine the rates of physical I/O's for Jobs, Objects, Disk Units, and the time values associated with these I/O's.
Suggested maximum collection duration: 10 minutes

```
ADDPEXDFN DFN(PDIO) TYPE(*TRACE) INTERVAL(200) MAXSTG(500000)
TRCFULL(*STOPTRC) TRCTYPE(*SLTEVT) JOB(*) TASK(*NONE ) SLTEVT(*YES)
MCHINST(*NONE) BASEVT(*PMCO ) PGMEVT(*MIENTRY *MIEXIT )
DSKEVT(*READSTR *READEND *WRTSTR *WRTEND )
```

***TASKSWITCH**
Allows you to determine how a job spends it's time within the system during the collection interval.
Suggested maximum collection duration: 5 minutes

```
ADDPEXDFN DFN(TASKSWITCH) TYPE(*TRACE) INTERVAL(200) MAXSTG(500000)
TRCFULL(*STOPTRC) TRCTYPE(*SLTEVT) JOB(*) TASK(*NONE ) SLTEVT(*YES)
MCHINST(*NONE) BASEVT(*TASKSWTIN *TASKSWTOUT *PMCO *TASKSWTOUTQ
*TASKSWTOUTINT *TASKAVAIL *CPUSWTIN *CPUSWTOUTP ) PGMEVT(*MIENTRY
*MIEXIT )
```

***OBJ_NETSIZE**
Allows you to determine where storage is being used within the system during the collection interval.
Suggested maximum collection duration: 4 hours

```
ADDPEXDFN DFN(NETSIZE) TYPE(*TRACE) INTERVAL(200) MAXSTG(500000)
TRCFULL(*STOPTRC) TRCTYPE(*SLTEVT) JOB(*) TASK(*NONE ) SLTEVT(*YES)
```

```
MCHINST(*NONE) BASEVT(*PMCO ) PGMEVT(*MIENTRY *MIEXIT )
STGEVT(*TRUNCSEG *EXDSEG *CRTSEG *DLTSEG )
```

***PROFILE**

User specifies from 1 to 16 programs and/or service programs for CPU profiling PEX PROFILE data collection. This collection type helps identify CPU 'hot spots' within the selected programs/service programs. The user specified pane size (default of 4) is used to logically divide each program/service program up into 'panes' for granularity of analysis. When CPU is used by a program/service program in the collection data is captured by pane within program/service program.

```
ADDPEXDFN DFN(PROFILE) TYPE(*PROFILE) INTERVAL(1) JOB(*) TASK(*NONE )
PGM((LIB/PGM *ALL '*ALL' *PGM 4))
```

***STATSFLATY**

This PEX STATISTICAL collection captures CPU and IO statistics by program/procedure/method. Procedure/method statistics are only provided for programs/service programs with an ENDPFRCOL value other than *NONE or *PEP. One collection-wide count of both CPU and IO in-line and cumulative values is stored for every program/procedure/method that was called (program/procedure/method statistics for all jobs/tasks in the collection are merged).

```
ADDPEXDFN DFN(FLATY) TYPE(*STATS) DTAORG(*FLAT) MRGJOB(*YES)
PGMBKTEVT(*MISTREND *JVA *MIENTRYEXIT ) JOB(*) TASK(*NONE ) SLTEVT(*YES)
```

***STATSHIER**

This PEX STATISTICAL collection captures call flow, CPU and IO statistics by program/procedure/method by calling program/procedure/method. Procedure/method statistics are only provided for programs/service programs with an ENDPFRCOL value other than *NONE or *PEP. One count of both CPU and IO in-line and cumulative values is stored by calling program/procedure method for every program/procedure/method that was called by the job. We recommend that you try to keep the number of jobs in a collection of this type to less than 10.

```
ADDPEXDFN DFN(HIER) TYPE(*STATS) DTAORG(*HIER) PGMBKTEVT(*MISTREND
*JVA *MIENTRYEXIT ) JOB(*) TASK(*NONE ) SLTEVT(*YES)
```

***TPROF**

This PEX TRACE collection type helps identify system-wide CPU 'hot spots' within application programs/service programs, OS/400 programs/procedures/methods and SLIC procedures. Unlike *PROFILE, the user does not need to specify program names. The ENDPFRCOL setting is not used to determine whether a program/procedure/method will be recorded.

```
ADDPEXDFN DFN(TPROF) TYPE(*TRACE) INTERVAL(1) MAXSTG(500000)
TRCFULL(*STOPTRC) TRCTYPE(*SLTEVT) JOB(*) TASK(*NONE ) SLTEVT(*YES)
MCHINST(*NONE) BASEVT(*PMCO )
```

***STATSFLATN**
This PEX STATISTICAL collection captures CPU and IO statistics by program/procedure/method. Procedure/method statistics are only provided for programs/service programs with an ENDPFRCOL value other than *NONE or *PEP. One count of both CPU and IO in-line and cumulative values is stored for every program/procedure/method that was called by job (program/procedure/method statistics for jobs/tasks in the collection are stored separately for each individual job-thread or task). We recommend that you try to keep the number of jobs in a collection of this type to less than 10.

```
ADDPEXDFN DFN(FLATN) TYPE(*STATS) DTAORG(*FLAT) MRGJOB(*NO)
PGMBKTEVT(*MISTREND *JVA *MIENTRYEXIT ) JOB(*) TASK(*NONE ) SLTEVT(*YES)
```

***TAPE_IO**
Collects information on commands issued to the tape device for use in diagnosing tape performance issues. This function can be run for an extended period of time. You may require the additional information provided by the *DASDSTR and *DASDEND options to diagnose the problem.

```
ADDPEXDFN DFN(TAPE) TYPE(*TRACE) INTERVAL(200) MAXSTG(500000)
TRCFULL(*STOPTRC) TRCTYPE(*SLTEVT) JOB(*) TASK(*NONE ) SLTEVT(*YES)
MCHINST(*NONE) BASEVT(*PMCO ) DSKSVREVT(*USER )
```

***ILEACTGRP**
Collects information on the creation of activation groups and the destruction of activation groups. This can aid you in determining if there are issues within your jobs normal execution affected by the creation or destruction, such as what is the performance cost associated with their creation/destruction, etc. This can be useful in squeezing out that last bit of application performance. In addition, for those problem types which collect trace type data, you will be presented with an additional parameter ADDEVTS(14), that allows you to extend the data collections capabilities.

```
ADDPEXDFN DFN(ILEACTGRP) TYPE(*TRACE) INTERVAL(200) MAXSTG(500000)
TRCFULL(*STOPTRC) TRCTYPE(*SLTEVT) JOB(*) TASK(*NONE ) SLTEVT(*YES)
MCHINST(*DEACTPG ) BASEVT(*PMCO *ACTGRPDLT *ACTGRPCRT
*ACTGRPACTPGM ) PGMEVT(*MISTR *MIEND )
```

***HEAPSTG**
Collects information about the creation and destruction of heap allocations on your system. This option can collect a lot of events in a very short period of time, particularly on systems running newer workloads.

```
ADDPEXDFN DFN(HEAP) TYPE(*TRACE) INTERVAL(200) MAXSTG(500000)
TRCFULL(*STOPTRC) TRCTYPE(*SLTEVT) JOB(*) TASK(*NONE ) SLTEVT(*YES)
MCHINST(*NONE) BASEVT(*PMCO ) STGEVT(*HDLHEAP *RESHEAP *USRHEAP
*LCLHEAP *SYSHEAP *ACTGRPHEAP )
```

***IFSEVTS**
Collects information whenever an IFS file is accessed, and when an IFS file is opened or closed. The length

of time you can collect data for will vary with the numbers of jobs over which data is being collected which are actually using the Itegrated File System.

```
ADDPEXDFN DFN(IFS) TYPE(*TRACE) INTERVAL(200) MAXSTG(500000)
TRCFULL(*STOPTRC) TRCTYPE(*SLTEVT) JOB(*) TASK(*NONE ) SLTEVT(*YES)
MCHINST(*NONE) BASEVT(*PMCO ) PGMEVT(*MIENTRY *MIEXIT ) OSEVT(*IFSOPEN
*IFSIO )
```

***WEBSPHERE**

Collects information about WebApp/Servlet's, Container Transactions, EJB/EJB methods, Database Connection Pool, EJB Object Pool, and ORB Thread Pool. This option could collect a lot of events in a very short period of time. It is advisable to only run this trace for short periods of time, 5-10 minutes.

```
ADDPEXDFN DFN(WEBSPHERE) TYPE(*TRACE) INTERVAL(200) MAXSTG(500000)
TRCFULL(*STOPTRC) TRCTYPE(*SLTEVT) JOB(*) TASK(*NONE ) SLTEVT(*YES)
MCHINST(*NONE) BASEVT(*PMCO ) APPEVT(*WAS )
```

Table of Contents | Previous | Next

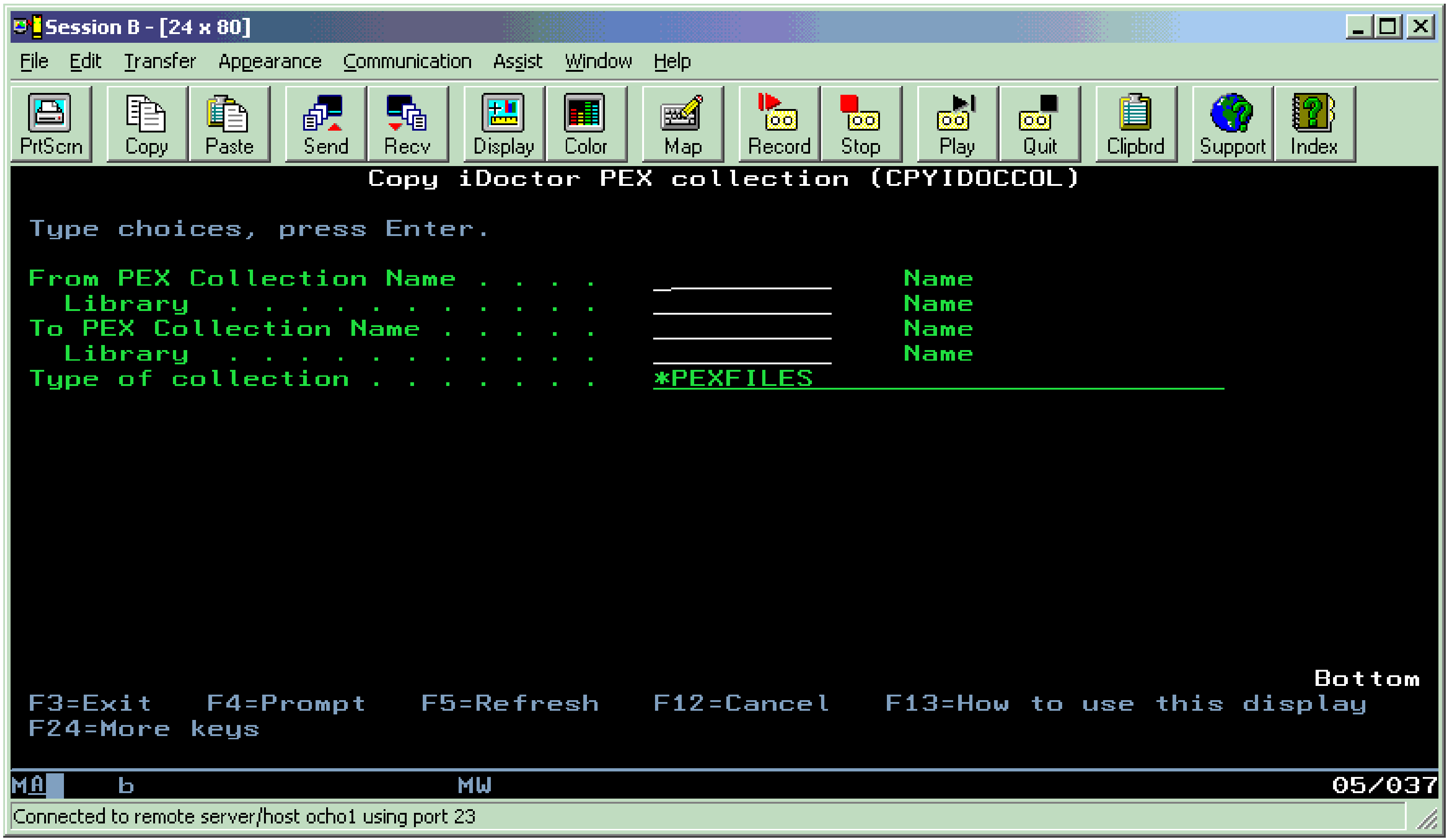
3.7.2 CPYIDOCCOL - copy a PEX collection

This command is used to copy a PEX collection that may or may not have been created with the STRIDOCCOL command, from one library to another.

The STRIDOCCOL command will create a PEX collection but may optionally include several SMTR* files in the user's library. If these files exist, they will be copied by the CPYIDOCCOL command along with the rest of the collection data. This command also supports the copying of single object PEX collections which are stored as *MGTCOL objects.

Any analyses created for a collection are NOT copied by this command.

This commands exists in library QYPINT.


Session B - [24 x 80]
File Edit Transfer Appearance Communication Assist Window Help
PrtScrn Copy Paste Send Recv Display Color Map Record Stop Play Quit Clipbrd Support Index

```
                    Copy iDoctor PEX collection (CPYIDOCCOL)

 Type choices, press Enter.

 From PEX Collection Name . . . .   _          Name
   Library  . . . . . . . . . . .              Name
 To PEX Collection Name . . . . .              Name
   Library  . . . . . . . . . . .              Name
 Type of collection . . . . . . .   *PEXFILES

                                                                        Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

MA b MW 05/037
Connected to remote server/host ocho1 using port 23

\

CPYIDOCCOL Parameters

FROMCOL
Specify the name of the collection you want to copy.

FROMLIB
Specify the name of the library containing the collection you want to copy.

TOCOL
Specify the name to give the new collection that will be copied.

TOLIB
Specify the name of the library to copy the collection to.

TYPE - type of collection
This value determines the type of PEX collection that will be copied.

***PEXFILES**
This is the normal type of PEX collection that exists in the QAYPE* files in the user library.

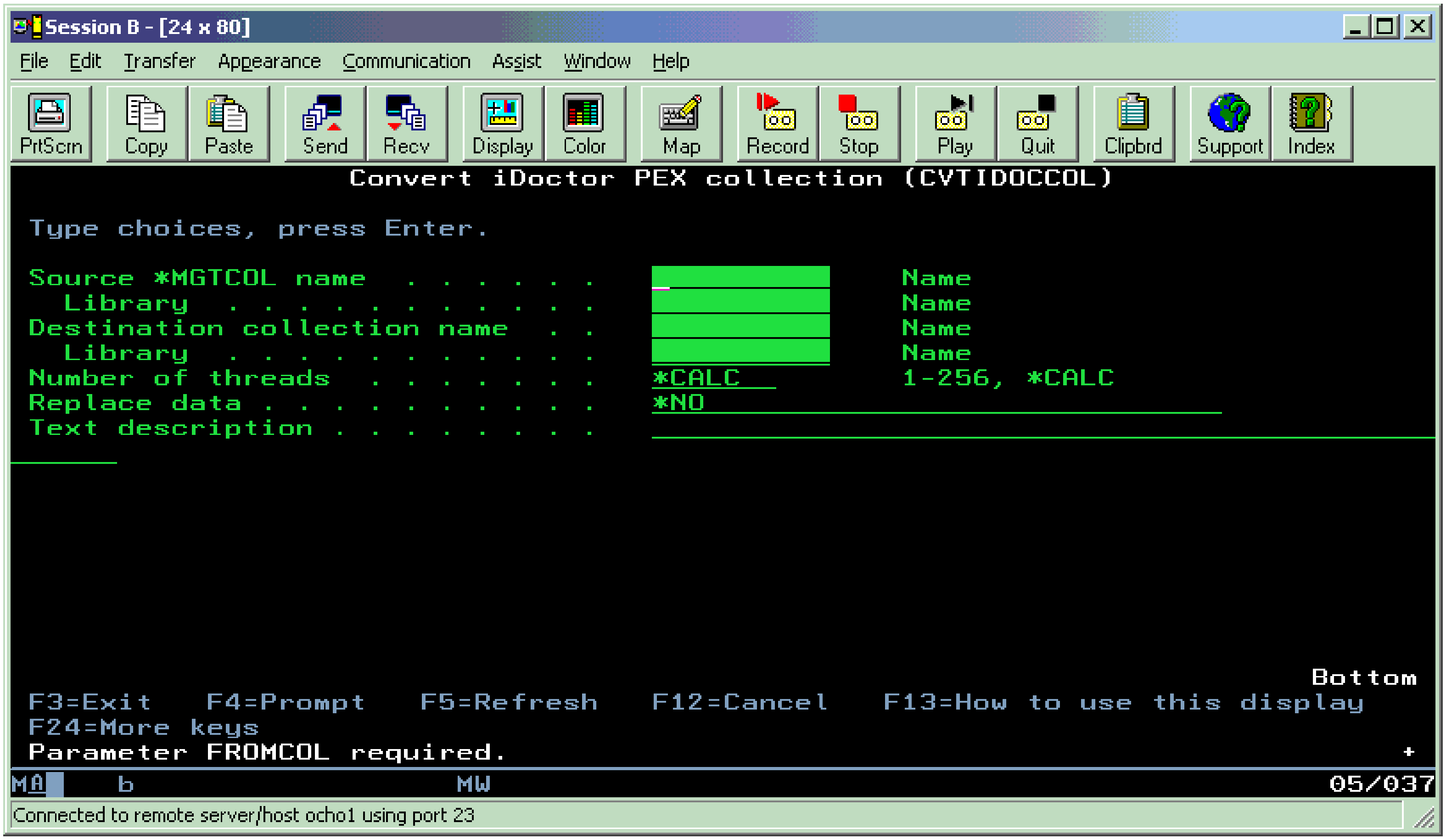
***SINGLEOBJ**
A single object collection is a *MGTCOL object that contains all of the data of a PEX collection but is in an unuseable format until it is split out into the QAYPE* files. Use the command CVTIDOCCOL to split the *MGTCOL object into useable QAYPE* files.

3.7.3 CVTIDOCCOL - convert a PEX *MGTCOL object to a PEX collection

This command is used to convert a PEX collection from a *MGTCOL object format to the useable QAYPE* collection file format.

During the creation process the ENDPEX command is called in order to produce the PEX output files. The 'number of threads', 'replace data' and 'description' parameters on the CVTIDOCCOL command are passed to the ENDPEX command during the creation process. By using this command you can ensure that the information stored in the collection information file (if you used STRIDOCCOL or the client to create the collection) is modified appropriately so it displays correctly in the client.

This commands exists in library QYPINT.


Session B - [24 x 80]
File Edit Transfer Appearance Communication Assist Window Help
PrtScrn Copy Paste Send Recv Display Color Map Record Stop Play Quit Clipbrd Support Index

```
                     Convert iDoctor PEX collection (CVTIDOCCOL)

 Type choices, press Enter.

 Source *MGTCOL name . . . . . .                  Name
   Library . . . . . . . . . . .                  Name
 Destination collection name  . .                 Name
   Library . . . . . . . . . . .                  Name
 Number of threads  . . . . . . .   *CALC         1-256, *CALC
 Replace data . . . . . . . . . .   *NO
 Text description . . . . . . . .




                                                                       Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
 Parameter FROMCOL required.                                                 +
MA    b                    MW                                          05/037
```

Connected to remote server/host ocho1 using port 23

CVTIDOCCOL Parameters

FROMCOL
Specify the name of the *MGTCOL object that will be converted to a useable PEX collection.

FROMLIB
Specify the name of the library containing the collection that will be converted to a useable PEX collection.

TOCOL
Specify the name to give the new collection that will be created.

TOLIB
Specify the name of the library to contain the new collection.

NBRTHD - Number of threads
Specifies the number of concurrent threads that the ENDPEX command uses to process the data in the session being ended. Specifying a number

greater than 1 allows the ENDPEX command to take advantage of available CPU cycles, especially on a multi-processor system. While this may speed up the command processing, it may also degrade the performance of other jobs on the system. You can minimize this impact by changing the priority of the job that runs the ENDPEX command to a higher number. You should also verify that the disk subsystem can handle the additional threads.

***CALC**
The system calculates a reasonable number of threads to do the command processing which does not use excessive CPU or disk resources.

***MAX**
The system calculates a maximum number of threads to do the command processing. An attempt will be made to maximize utilization on all resources to minimize processing time. This may cause severe degradation for all other jobs on the system.

number-of-threads
Specify the number of threads for the ENDPEX command to use to process the collected data.

RPLDTA - Replace data
Specifies whether to replace the data in an existing collection with the new PEX collection data. The possible values are:

***NO**
If a collection already exists with the same name, an error message is sent to the user. This prevents the user from inadvertently writing over existing data.

***YES**
If a collection already exists with the same name, the old data is lost and is replaced by the new data.

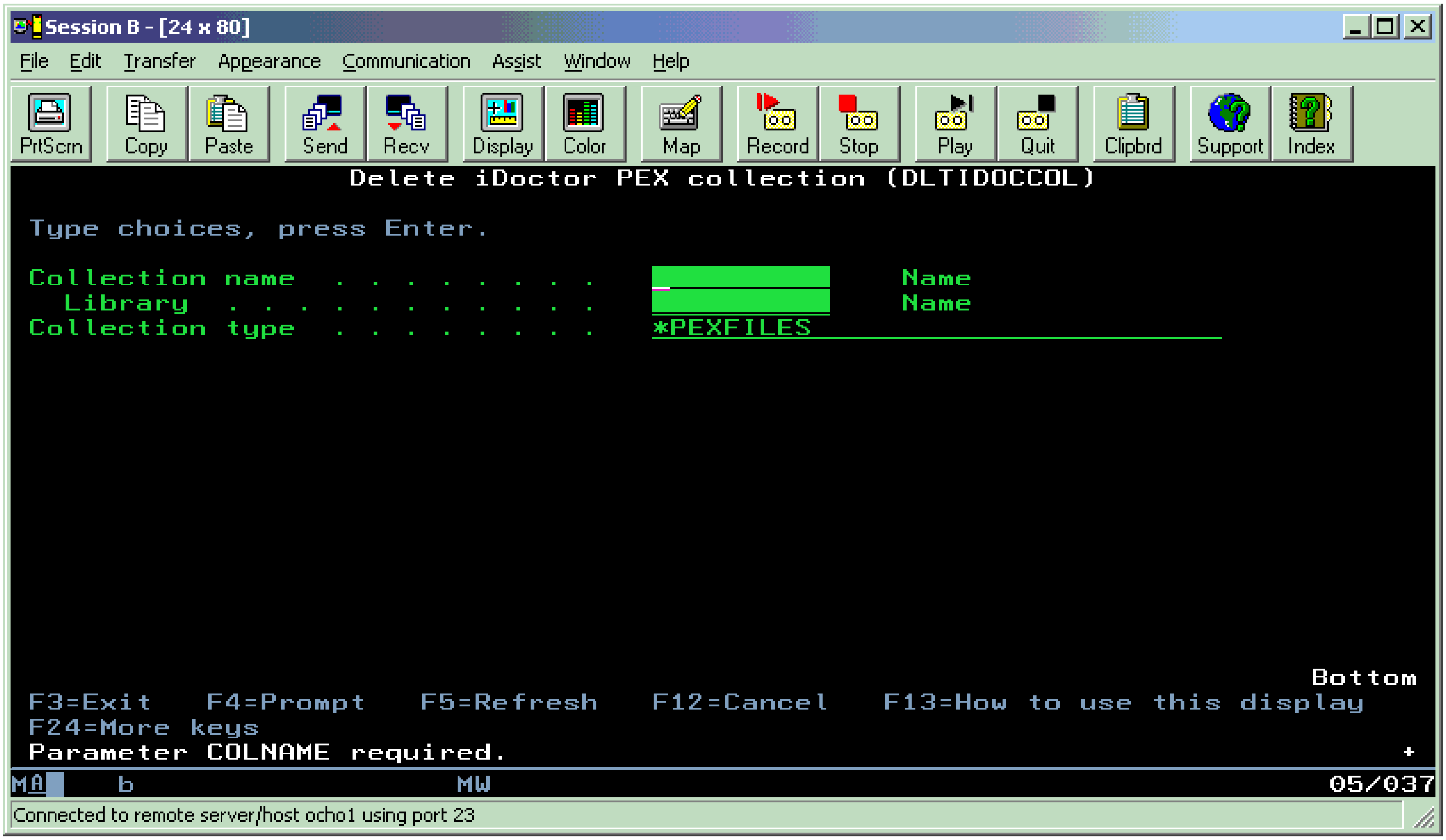
TEXT
Specifies the text that briefly describes the type of data collected. Specify no more than 50 characters of text, enclosed in apostrophes.

Table of Contents Previous Next

3.7.4 DLTIDOCCOL - delete a PEX collection

This command deletes a PEX collection (and all analysis members associated with it) from a library. You may delete either type of collection using this command (PEX files or a single object collection). By using this command you can ensure that the information stored in the collection information file (if you used STRIDOCCOL or the client to create the collection) is deleted as well as the collection itself.

This commands exists in library QYPINT.


Session B - [24 x 80]
File Edit Transfer Appearance Communication Assist Window Help
PrtScrn Copy Paste Send Recv Display Color Map Record Stop Play Quit Clipbrd Support Index

```
                   Delete iDoctor PEX collection (DLTIDOCCOL)

 Type choices, press Enter.

 Collection name . . . . . . . . .                Name
   Library . . . . . . . . . . . .                Name
 Collection type . . . . . . . . .   *PEXFILES




                                                                        Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
 Parameter COLNAME required.                                                 +
MA    b                  MW                                              05/037
```

Connected to remote server/host ocho1 using port 23

DLTIDOCCOL Parameters

COLNAME
This is the name of the collection that will be deleted from the user library.

LIB
Indicates the library name which contains the PEX collection to be deleted.

TYPE
This value determines the type of PEX collection that will be deleted.

> ***PEXFILES**
> This is the normal type of PEX collection that exists in the QAYPE* files in the user library.
>
> ***SINGLEOBJ**
> A single object collection is a *MGTCOL object that contains all of the data of a PEX collection but is in an unuseable format until it is split out into the QAYPE* files. Use the command CVTIDOCCOL to split the *MGTCOL object into useable QAYPE* files.

Table of Contents Previous Next

3.7.5 ENDIDOCCOL - ends a PEX collection

This command is used to end a PEX collection that was started with the STRIDOCCOL command, prior to the expiration of the provided time value.

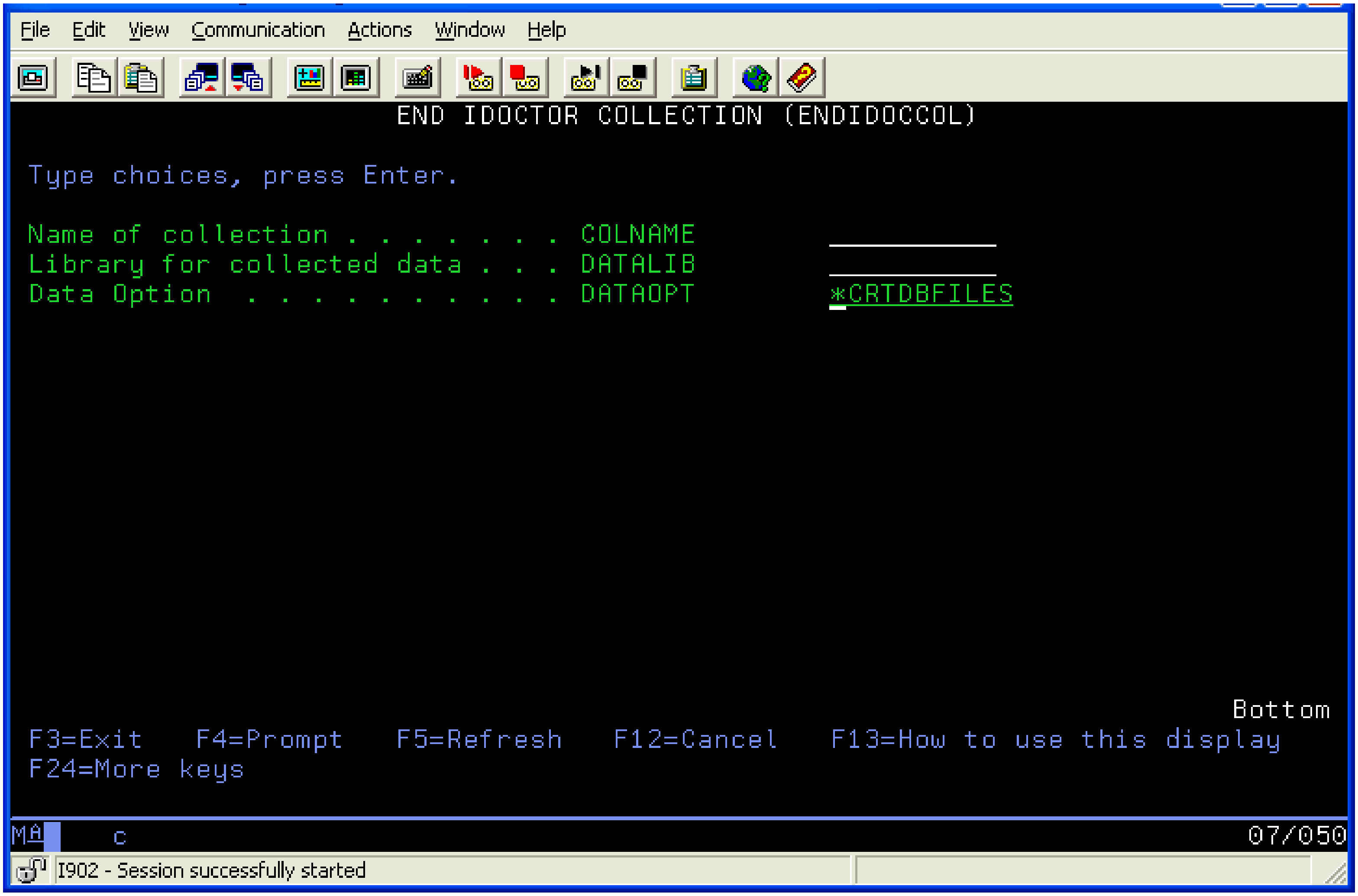
This command has no effect once the time value provided on the STRIDOCCOL command has expired.

This command can also be used to override the option on the STRIDOCCOL command for how the data is stored.

The options are either:

- Create DB files - Ends the PEX collection prematurely and creates the PEX files.
- Create service file (single object) - Ends the PEX collection prematurely and produces a single object for transport to another system.
- Delete - Ends the PEX collection and destroys the data gathered.
- Restart - Ends the PEX collection and restarts it with the same parameters used previously.
- Stop - Stops the collection and changes its status to "suspended".

This commands exists in library QYPINT.



```
                        END IDOCTOR COLLECTION (ENDIDOCCOL)

 Type choices, press Enter.

 Name of collection . . . . . . . . COLNAME       __________
 Library for collected data . . . DATALIB       __________
 Data Option  . . . . . . . . . . DATAOPT       *CRTDBFILES

                                                                            Bottom
 F3=Exit    F4=Prompt    F5=Refresh    F12=Cancel    F13=How to use this display
 F24=More keys
MA     c                                                                    07/050
```

I902 - Session successfully started

ENDIDOCCOL Parameters

COLNAME
Specify the name of the collection you want to end.

DATALIB - Library for collected data
The library you specify here must be the one which was supplied on the STRIDOCCOL command. Depending on the value supplied to the DATAOPT parameter on this command, any data that is kept will be put into this library.

DATAOPT - Data option
The value entered here determines the action taken for this collection.

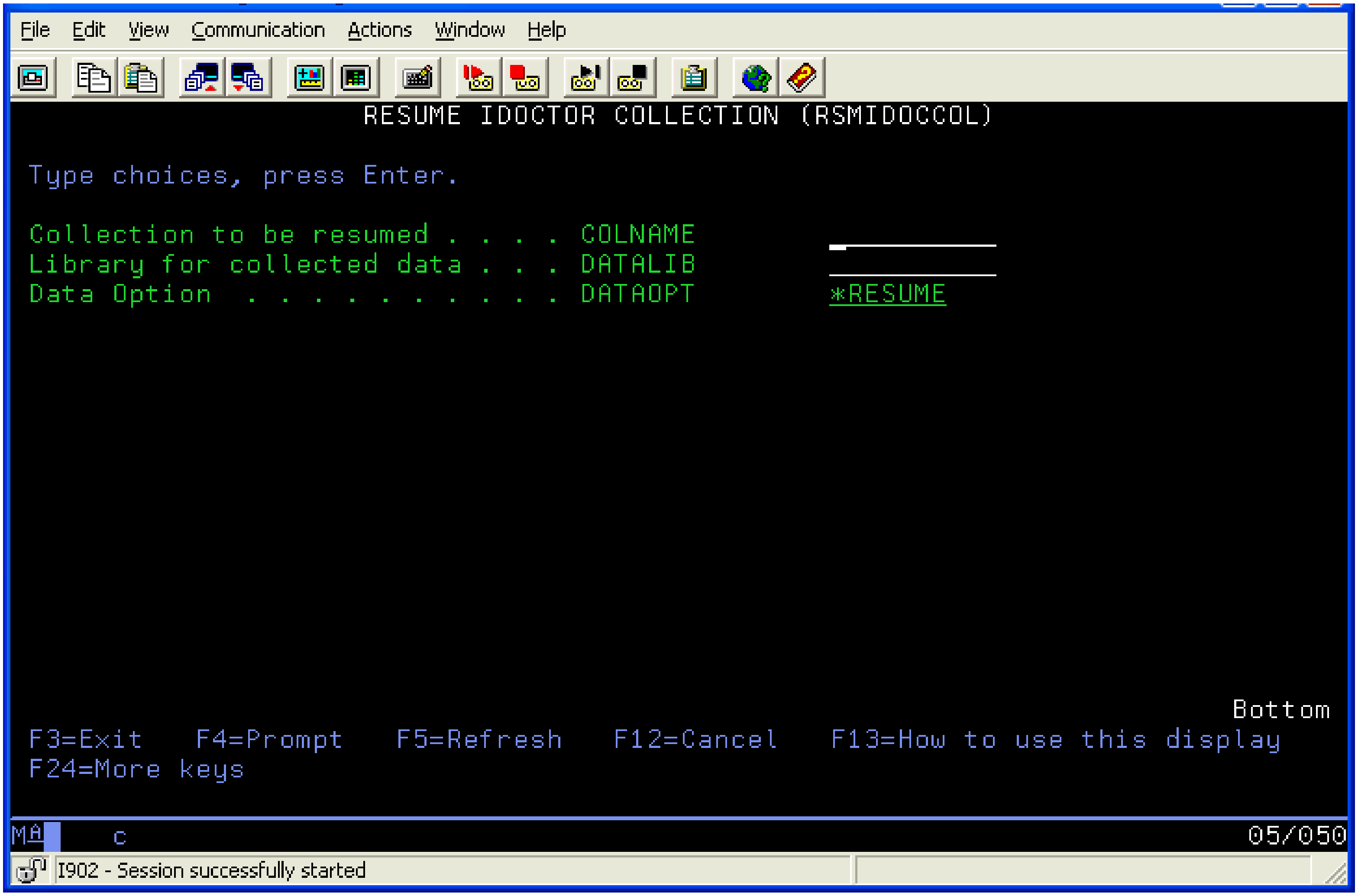
> ***CRTDBFILES**
> Data is collected and put into the QAYPE* files, ready for data analysis commands.
>
> ***CRTSVCFILE**
> Data is collected and put into a single file which is easily transported to another system at the same release where data analysis can be performed.
>
> ***DELETE**
> The collection of data is stopped and any data already collected is discarded.
>
> ***RESTART**
> The collection is stopped, any data that was collected is discarded, then the collection is restarted with the same values originally provided on the STRIDOCCOL command.
>
> ***STOP**
> The collection is stopped. It will remain in a stopped status until another action is requested from the ENDIDOCCOL command.

Table of Contents Previous Next

3.7.6 RSMIDOCCOL - resume a suspended PEX collection

This command is used to resume a PEX collection that was started with the STRIDOCCOL (or the GUI) in "standby" mode.

This commands exists in library QYPINT.



```
                       RESUME IDOCTOR COLLECTION (RSMIDOCCOL)

Type choices, press Enter.

Collection to be resumed . . . . COLNAME       __________
Library for collected data . . . DATALIB       __________
Data Option  . . . . . . . . . . DATAOPT       *RESUME

                                                                        Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
MA     c                                                               05/050
I902 - Session successfully started
```

RSMIDOCCOL Parameters

COLNAME
Specify the name of the collection you want to resume.

DATALIB - Library for collected data
The library you specify here must be the one which was supplied on the STRIDOCCOL command. Depending on the value supplied to the DATAOPT parameter on this command, any data that is kept will be put into this library.

DATAOPT - Data option
The value entered here determines the action taken for this collection.

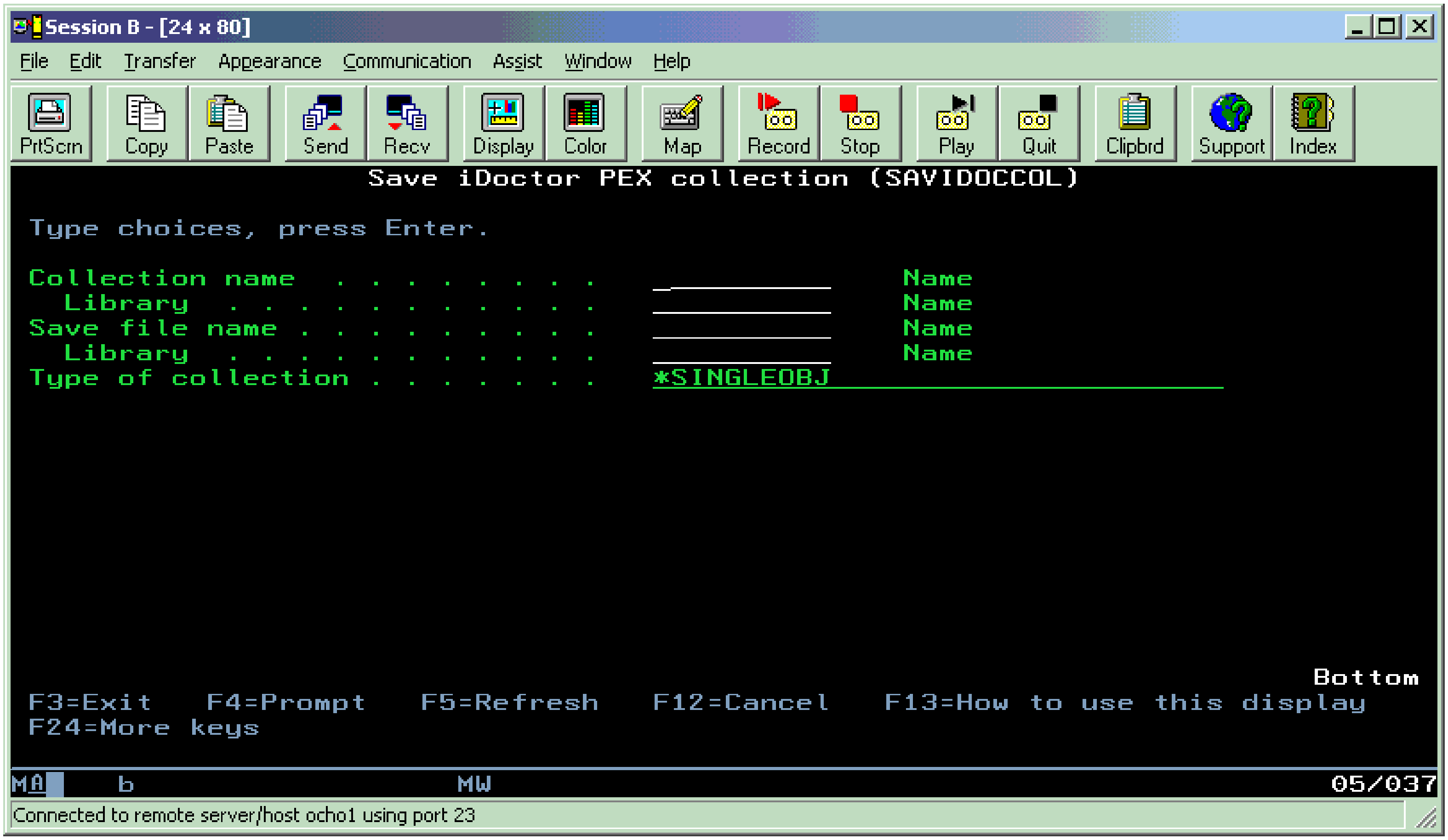
> ***RESUME**
> Collection of PEX data to begin.

Table of Contents | Previous | Next

3.7.7 SAVIDOCCOL - save a PEX collection

This command saves a collection of either type (single object or PEX files) to a save file. When dealing with a PEX files type collection this command can be very helpful in isolating out the collection member you want to save for transport to another system. Use the RSTIDOCCOL command to restore the collection on the target system.

This commands exists in library QYPINT.


Session B - [24 x 80]
File Edit Transfer Appearance Communication Assist Window Help
PrtScrn Copy Paste Send Recv Display Color Map Record Stop Play Quit Clipbrd Support Index

```
                   Save iDoctor PEX collection (SAVIDOCCOL)

 Type choices, press Enter.

 Collection name  . . . . . . . .   _               Name
   Library  . . . . . . . . . . .                   Name
 Save file name . . . . . . . . .                   Name
   Library  . . . . . . . . . . .                   Name
 Type of collection . . . . . . .   *SINGLEOBJ




                                                                        Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

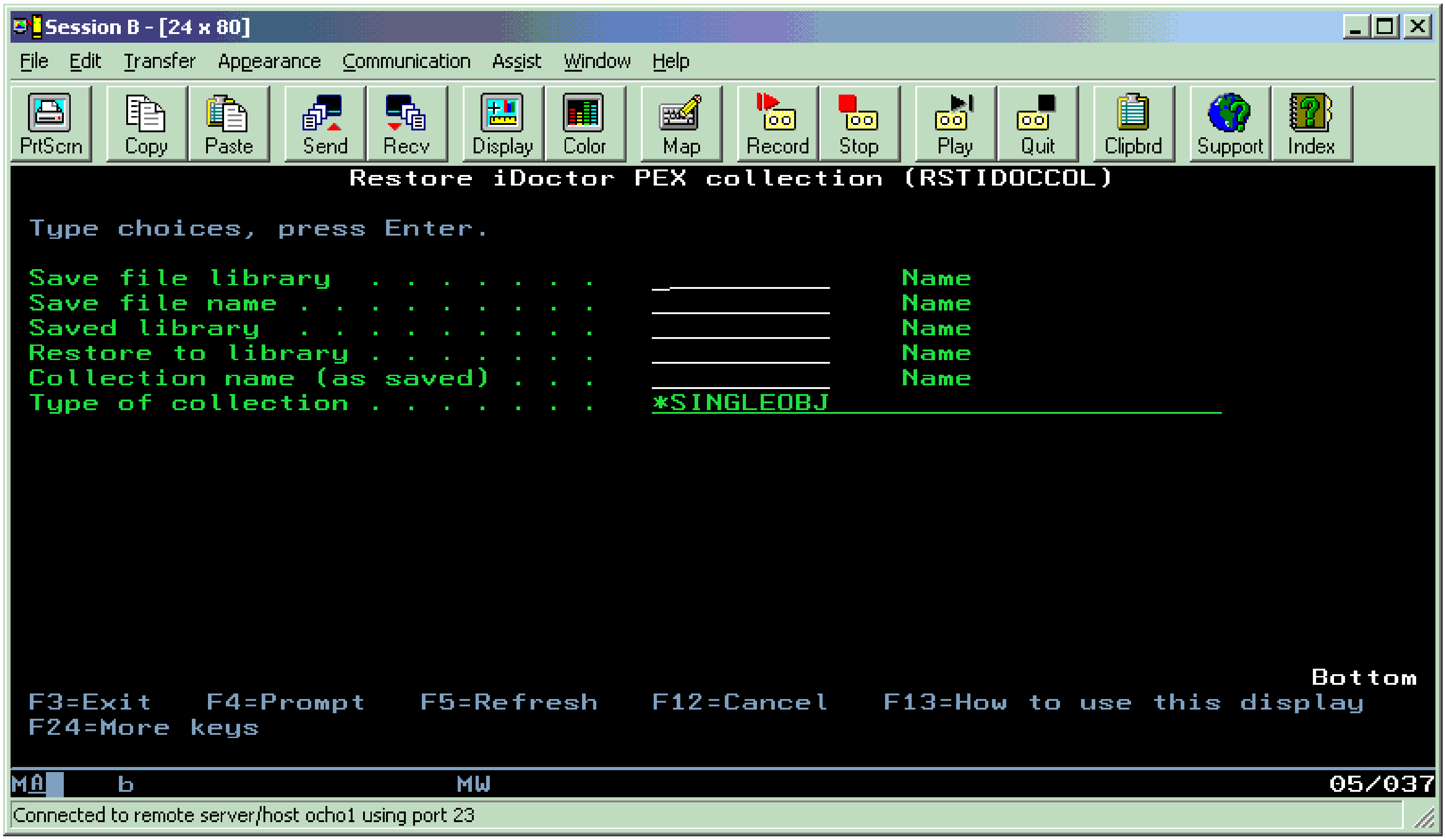
MA   b   MW   05/037
Connected to remote server/host ocho1 using port 23

Table of Contents Previous Next

3.7.8 RSTIDOCCOL - restore a PEX collection

This command restores a collection of either type (single object or PEX files) from a save file created using the SAVIDOCCOL command.

This commands exists in library QYPINT.



Session B - [24 x 80]
File Edit Transfer Appearance Communication Assist Window Help
PrtScrn Copy Paste Send Recv Display Color Map Record Stop Play Quit Clipbrd Support Index

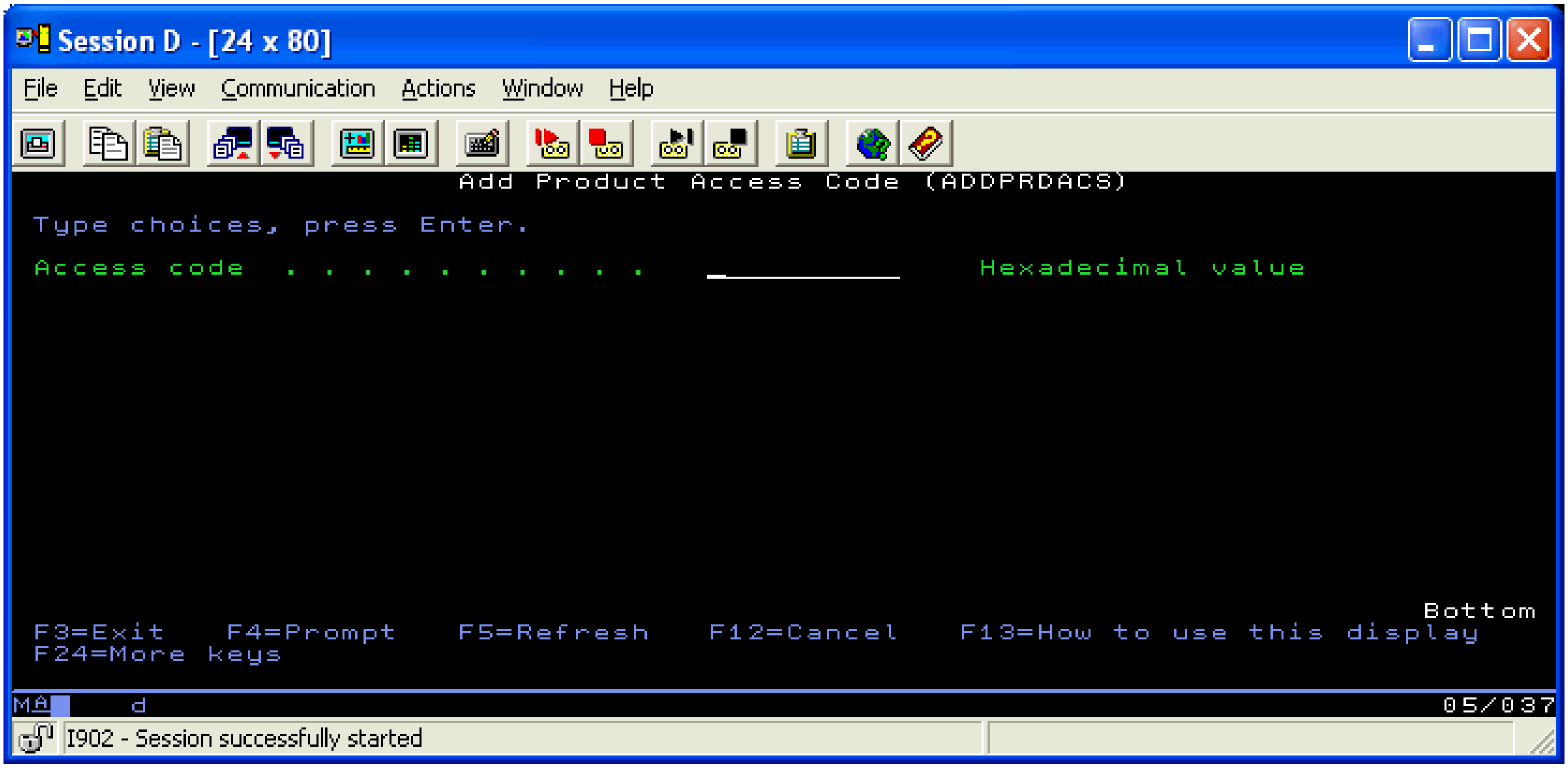
```
                 Restore iDoctor PEX collection (RSTIDOCCOL)

 Type choices, press Enter.

 Save file library  . . . . . . .   __________     Name
 Save file name . . . . . . . . .   __________     Name
 Saved library  . . . . . . . . .   __________     Name
 Restore to library . . . . . . .   __________     Name
 Collection name (as saved) . . .   __________     Name
 Type of collection . . . . . . .   *SINGLEOBJ

                                                                        Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
MA     b                 MW                                            05/037
```

Connected to remote server/host ocho1 using port 23

Table of Contents | Previous | Next

3.7.9 ADDPRDACS - apply an access code

This command is used to apply any iDoctor for iSeries component's access code to the current system. The access codes are provided by IBM through this website for either a trial or licensed version.



```
                        Add Product Access Code (ADDPRDACS)

 Type choices, press Enter.

 Access code . . . . . . . . . .   __________      Hexadecimal value












                                                                          Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
MA     d                                                                05/037
```

I902 - Session successfully started

Table of Contents | Previous | Next

3.8 Commands - library SMTRACE

This section covers the commands provided with PEX Analyzer in library SMTRACE.

Table of Contents | Previous | Next

3.9 PEX Analyzer analysis files

This section describes the structure of the output files produced by PEX Analyzer that are created as part of the analyses.

These files are all begin with G_.

Table of Contents | Previous | Next

3.9.1 G_ACTGRP3 - Activation group creates/deletes by job-thread

Description: This file provides statistics for the activation groups created/deleted by job-thread.

Record: This file will contain 1 record per job/thread in the collection that created/deleted activation groups.

Field name	**Field Description**	**Format**	**Comments**
QRYTNAME	Task/Job query name	Char(47)	name of the job/thread or task
JOBTHRTOT	Job-thread ACTGRP Crt/Dlt Total	Zoned(7)	
PCTTOT	Percent Total ACTGRP Crt/Dlt	Zoned(5)	
AGTOT	Total ACTGRP Crt/Dlt	Zoned(7)	

Table of Contents | Previous | Next

3.9.2 G_ACTGRP3A - Activation group creates/deletes by job

Description: This file provides statistics for the activation groups created/deleted by job.

Record: This file will contain 1 record per job in the collection that created/deleted activation groups.

Field name	**Field Description**	**Format**	**Comments**
JOB	Job	Char(26)	fully qualified job name
JOBTOT	Job ACTGRP Crt/Dlt Total	Zoned(7)	
PCTTOT	Percent Total ACTGRP Crt/Dlt	Zoned(5)	
AGTOT	Total ACTGRP Crt/Dlt	Zoned(7)	

Table of Contents | Previous | Next

3.9.3 G_ACTGRP3B - Activation group creates/deletes by program

Description: This file provides statistics for the activation groups created/deleted by program.

Record: This file will contain 1 record per program in the collection that created/deleted activation groups.

Field name	Field Description	Format	Comments
PGM	Program	Char(10)	
PGMTOT	Program ACTGRP Crt/Dlt Total	Zoned(7)	
PCTTOT	Percent Total ACTGRP Crt/Dlt	Zoned(5)	
AGTOT	Total ACTGRP Crt/Dlt	Zoned(7)	

Table of Contents Previous Next

3.9.4 G_ACTGRP3C - Activation group creates/deletes by program within job-thread

Description: This file provides statistics for the activation groups created/deleted by program within job/thread.

Record: This file will contain 1 record per program within job/thread in the collection that created/deleted activation groups.

Field name	Field Description	Format	Comments
QRYTNAME	Task/Job query name	Char(47)	name of the job/thread or task
PGM	Program	Char(10)	
JTPGMTOT	Job-thread ACTGRP Crt/Dlt Total	Zoned(7)	the total activation groups created/deleted by program within job/thread
PCTTOT	Percent Total ACTGRP Crt/Dlt	Zoned(5)	
AGTOT	Total ACTGRP Crt/Dlt	Zoned(7)	

Table of Contents Previous Next

3.9.5 G_CPUP5X - Approximate CPU by priority within interval

Description: This file provides the approximate cpu usage by priority within an interval.

Record: This file will contain 1 record per distinct priority occuring per interval within the collection.

Field name	**Field Description**	**Format**	**Comments**
INTERVAL	Interval Number	Zoned(8)	
QTSPRIX	Initial Priority	Char(10)	
SYSPCT	Priority Percent of Total CPU this Interval	Zoned(12)	
INTPCT	Priority Percent of Used CPU this Interval	Zoned(9)	
INTPRICNT	Number of Distinct Priorities this Interval	Zoned(7)	

Table of Contents | Previous | Next

3.9.6 G_CPU4X - Approximate Job/Task CPU by interval

Description: This file provides information about the CPU usage of jobs for each interval within the PEX collection.

Record: This file will contain 1 record per job using CPU per interval within the collection.

Field name	**Field Description**	**Format**	**Comments**
INTERVAL	Interval Number	Zoned(8)	
QRYTNAME	Job/Task Name	Char(47)	
QTSPRIX	Job/Task Initial Priority	Char(10)	
SYSPCT	Job/Task Percent of Total CPU this Interval	Zoned(9)	
INTPCT	Job/Task Percent of Used CPU this Interval	Zoned(9)	
INTJOBCNT	Number of Jobs/Tasks Using CPU this Interval	Zoned(7)	

Chapter 4 PTDV

This chapter describes how to collect Performance Explorer (PEX) data for analysis with iDoctor for iSeries PTDV. This chapter includes:

- The method and command syntax needed to collect PEX TRACE data for PTDV

Table of Contents | Previous | Next

4.1 Overview of Collecting PEX Data

The basic steps for creating a PEX Collection are the following:

1. Creating or choosing an existing **PEX definition**(explained in Section 4.2, PEX Definitions).
2. Starting a collection using the definition.
3. Ending the collection and directing the data to be discarded or stored in a library.

Note: The PEX Analyzer client provides interfaces for creating and/or changing PEX definitions. The client also has an interface for creating a collection, viewing it as runs, and then analyzing its results. See the client side documentation for PEX Analyzer for more information.

Table of Contents | Previous | Next

4.2 Creating PEX definitions

A variety of PEX definitions can be used with PTDV. The most commonly used definitions (those listed in Standard Java Definitions or Standard ILE Definitions) can be created automatically once you've installed the tools library PTDV according to section "Installing the Tools Library" from the Web page where this tool was installed. To add the standard definitions mentioned in this section, invoke the commmand:

```
PTDV/ADDPTDVDFN
```

Alternatively, they can be created manually by using the command given in the description of each definition. You can also create your own Custom Definitions so that your trace contains only the data you need.

Note: You can use copy-paste to easily enter any of the command examples from this section into an iSeries command entry screen.

Table of Contents | Previous | Next

4.2.1 Standard Java definitions

The following definitions are commonly used with PTDV.

4.2.1.1 Java Method Trace with basic set of Java events

4.2.1.2 Java Method Trace with no Object Creates

4.2.1.3 Java Trace with only Object Creates

4.2.1.4 Java Trace with Object Locks

Table of Contents | Previous | Next

4.2.1.1 Java Method Trace with basic set of Java events

For Java applications, the most commonly used PEX definition include Java method entry and exit events, along with Java object create/delete events, Java object lock/unlock events, and WebSphere events. In V5R2, more events are available. The exact definition command will depend on the release, and the type of events you want to include.

For releases prior to V5R2:

```
ADDPEXDFN DFN(PTDVJAVA) TYPE(*TRACE) JOB(*ALL) TASK(*ALL) MAXSTG(1000000)
TRCTYPE(*SLTEVT) SLTEVT(*YES) BASEVT(*PMCO) PGMEVT(*JVAENTRY *JVAEXIT)
JVAEVT(*OBJCRT *THDSTTCHG) OSEVT(*MIEV13) TEXT('PTDV basic Java events')
```

This collection includes:

- Java method entry and exit events
- Object create and delete events
- Object lock and unlock events
- WebSphere events

For V5R2 and later releases:

```
ADDPEXDFN DFN(PTDVJAVA) TYPE(*TRACE) JOB(*ALL) TASK(*ALL) MAXSTG(1000000)
TRCTYPE(*SLTEVT) SLTEVT(*YES) BASEVT(*PMCO) PGMEVT(*JVAENTRY *JVAEXIT)
JVAEVT(*OBJCRT *LCKSTR *UNLCK *THDCRT *THDDLT) APPEVT(*WAS) TEXT('PTDV basic Java
events')
```

This collection includes:

- Java method entry and exit events
- Object create and delete events
- Object lock and unlock events
- Thread create and delete events
- WebSphere events

In V5R2, additional events can also be added:

```
ADDPEXDFN DFN(PTDVALLJVA) TYPE(*TRACE) JOB(*ALL) TASK(*ALL) MAXSTG(1000000)
TRCTYPE(*SLTEVT) SLTEVT(*YES) BASEVT(*PMCO) PGMEVT(*JVAENTRY *JVAEXIT)
JVAEVT(*OBJCRT *LCKSTR *UNLCK *THDCRT *THDDLT *CLSLOAD *GBGCOLSWEEP)
APPEVT(*WAS) TEXT('PTDV all supported Java events')
```

This collection includes:

- Java method entry and exit events
- Object create and delete events
- Object lock and unlock events
- Thread create and delete events
- Class load events
- Garbage collector sweep events
- WebSphere events

Table of Contents Previous Next

4.2.1.2 Java Method Trace with no Object Creates

In some cases, a trace like PTDVJAVA is needed, but object create events are not needed. Since the number of object create events can be very large, it is sometimes useful to use a definition with the basic set of PTDV events excluding object create events.

For releases prior to V5R2:

```
ADDPEXDFN DFN(PTDVNOCRT) TYPE(*TRACE) JOB(*ALL) TASK(*ALL) MAXSTG(1000000)
TRCTYPE(*SLTEVT) SLTEVT(*YES) BASEVT(*PMCO) PGMEVT(*JVAENTRY *JVAEXIT)
JVAEVT(*THDSTTCHG) OSEVT(*MIEV13)
TEXT('PTDV Java method entry/exit, Object Lock, and WAS events')
```

This definition includes:

- Java method entry and exit events
- Object lock and unlock events
- WebSphere events

For V5R2 and following releases:

```
ADDPEXDFN DFN(PTDVNOCRT) TYPE(*TRACE) JOB(*ALL) TASK(*ALL) MAXSTG(1000000)
TRCTYPE(*SLTEVT) SLTEVT(*YES) BASEVT(*PMCO) PGMEVT(*JVAENTRY *JVAEXIT)
JVAEVT(*LCKSTR *UNLCK) APPEVT(*WAS) TEXT('PTDV Java method entry/exit, Object Lock, WAS
events')
```

Table of Contents Previous Next

4.2.1.3 Java Trace with only Object Creates

Note that **no** Java method entry and exit events appear in this definition, so no performance information for the methods in the application will appear when PTDV processes a collection created with this definition.

For all releases:

```
ADDPEXDFN DFN(PTDVCRT) TYPE(*TRACE) JOB(*ALL) TASK(*ALL) MAXSTG(1000000)
TRCTYPE(*SLTEVT) SLTEVT(*YES) BASEVT(*PMCO) JVAEVT(*OBJCRT)
TEXT('PTDV Java Object Create events')
```

This collection includes:

- Object create and delete events

Table of Contents | Previous | Next

4.2.1.4 Java Trace with Object Locks

Note that **no** Java method entry and exit events appear in this definition, so no performance information for the methods in the application will appear when PTDV processes a collection created with this definition.

For releases prior to V5R2:

ADDPEXDFN DFN(PTDVLOCK) TYPE(*TRACE) JOB(*ALL) TASK(*ALL) MAXSTG(1000000)
TRCTYPE(*SLTEVT) SLTEVT(*YES) BASEVT(*PMCO) JVAEVT(*THDSTTCHG)
TEXT('PTDV Java Object Lock events')

This definition includes:

- Object lock and unlock events

For V5R2 and following releases:

ADDPEXDFN DFN(PTDVLOCK) TYPE(*TRACE) JOB(*ALL) TASK(*ALL) MAXSTG(1000000) TRCTYPE(*SLTEVT) SLTEVT(*YES) BASEVT(*PMCO) JVAEVT(*LCKSTR *UNLCK) TEXT('PTDV Java Object Lock events')

This definition includes:

- Object lock and unlock events

In V5R2, thread notify events can also be added:

ADDPEXDFN DFN(PTDVLCKNFY) TYPE(*TRACE) JOB(*ALL) TASK(*ALL) MAXSTG(1000000)
TRCTYPE(*SLTEVT) SLTEVT(*YES) BASEVT(*PMCO)
JVAEVT(*LCKSTR *UNLCK *THDNFY *THDNFYALL *THDWAIT)
TEXT('PTDV Java Object Lock and Notify events')

This definition includes:

- Object lock and unlock events
- Thread notify and wait events

Table of Contents Previous Next

4.2.2 Standard ILE/OPM Definitions

For ILE or OPM applications, PTDV can provide procedure summary and call trace information. Refer to Generating ILE data for more information on generating procedure and program hooks in non-Java code.

The following definition will generate events for ILE and OPM code which have been compiled to contain standard performance hooks:

```
ADDPEXDFN DFN(PTDVMI) TYPE(*TRACE) JOB(*ALL) TASK(*ALL) MAXSTG(1000000)
TRCTYPE(*SLTEVT) SLTEVT(*YES) BASEVT(*PMCO)
PGMEVT(*MIENTRY *MIEXIT) TEXT('PTDV MI Entry/Exit events')
```

ILE code can also contain "Trace Job" hooks, and a PEX collection can contain program events of this type if the code has been compiled appropriately (see Generating ILE data). Events of this type will be generated with this definition:

```
ADDPEXDFN DFN(PTDVMI) TYPE(*TRACE) JOB(*ALL) TASK(*ALL) MAXSTG(1000000)
TRCTYPE(*SLTEVT) SLTEVT(*YES) BASEVT(*PMCO)
PGMEVT(*PRCENTRY *PRCEXIT) TEXT('PTDV MI Trace Job Entry/Exit events')
```

To include MI complex information in either of these definitions, add *MISTR and *MIEND to the PGMEVT parameter.

Table of Contents | Previous | Next

4.2.3 Custom definitions

You can also create your own custom PEX definitions for use with PTDV. You will use a command such as:

```
ADDPEXDFN DFN(MyPTDVDef) TYPE(*TRACE) JOB(*ALL) TASK(*ALL) MAXSTG(1000000)
TRCTYPE(*SLTEVT) SLTEVT(*YES) BASEVT(*PMCO) PGMEVT(program entry/exit events)
JVAEVT(Java events) OSEVT(OS events) APPEVT(APP events) TEXT('My PTDV PEX Definition')
```

You can create a definition containing the specific event types that your analysis requires. PTDV can provide procedure/method summary and call trace information if the appropriate program events are in the definition, and those procedures or methods contain performance hooks. The following table describes the set of program events supported by PTDV. For information on how to generate performance hooks in your application code, refer to Generating Java Data or Generating ILE data.

All entries in this table will result in entries in the Procedure Summary Tables, Procedure Call Tables, and the Call Trace Tree.

Event Type	**Event Subtype**	**Performance hooks needed**	**Generates entry and exit events for**
Program Event	*MIENTRY and *MIEXIT	yes	ILE procedures and ILE or OPM programs
Program Event	*MISTR and *MIEND	no	MI complex instructions
Program Event	*JVAENTRY and *JVAEXIT	yes	Java methods (JIT and DE)
Program Event	*PRCENTRY and *PRCEXIT	yes	ILE programs and procedures
OS Event	*DBSVRREQ	no	DB server requests

Most Java events are supported by PTDV. A table containing the supported Java events are listed below. When these events are included in a definition containing Java method entry and exit events, event data is analyzed as well as method context information. For example, in a trace containing Java method entry and exit events and object create events, the methods which create objects is known as well as the class name and size of the object. In a trace containing object lock events along with Java method entry and exit events, the locking/unlocking methods can be determined as well as the class name for the object being locked.

Be aware, however, that Java method entry and exit events, as well as some of the events listed below, have the potential for producing an extremely large collection in a short period of time. Refer to Limiting the size of your collection for more information on keeping the collection to a reasonable size.

In some cases, it may be preferable to create a definition that doesn't contain Java method entry and exit events.

For more details on how each individual event is displayed within PTDV, see the following section.

Event Type	Event Subtype	Release	Description
Java Event	*OBJCRT	all	Java Object Create and Delete events
Java Event	*THDSTTCHG	V4R5, V5R1	Java Object Lock and Unlock events
Java Event	*LCKSTR and *UNLCK	V5R2+	Java Object Lock and Unlock events
Java Event	*THDCRT and *THDDLT	V5R2+	Java Thread Create and Delete events
Java Event	*CLSLOAD	V5R2+	Java Class Load event
Java Event	*THDNFY, *THDNFYALL, and *THDWAIT	V5R2+	Java Thread Notify and Wait events
Java Event	*GBGCOLSWEEP	V5R1+	Java Garbage Collection Sweep events
OS Event	*MIEV7	V4R5	Transaction events (for WebSphere Application Server)
OS Event	*USRTNS	V5R1+	Transaction events (for WebSphere Application Server)
OS Event	*MIEV13	V4R5, V5R1	WebSphere events
APP Event	*WAS	V5R2	WebSphere events

Table of Contents | Previous | Next

4.2.4 Performance data provided by Java events

The following list includes the supported Java events, and the data that is viewed about that event through PTDV

- ***OBJCRT**
  The object create event generates an event at each object create during the run of your application, and at each object delete during garbage collection. Information about the object's class name, allocated size, and array information is provided in PTDV. This data is summarized in the Object Summary Table on the Main Frame, Job Frame, Thread Frame, providing totals for the corresponding collection level. This information is also summarized for each Procedure that the object creates occur in, and can be viewed from a Procedure Summary Table and a Procedure Call Table. In addition, for each row in the Call Trace Pane, a count appears showing the number of objects that have been created by the selected procedure call.

- ***THDSTTCHG (prior to V5R2) or *LCKSTR and *UNLCK (V5R2 or after)**
  The lock events generate a record whenever a thread tries to acquire an object that is held by another thread, and generates another record when releasing the object. As in the object create event, this information is summarized on each of the Object Summary Tables, and if Java method events are present, statistics will be collected for object events that occur within each method, and that data is summarized on the Call Trace Pane, the Procedure Call Table, and the Procedure Summary Table. Note that for this latest version of PTDV, there is a count for both locks and lock events. The lock count is the number of combined lock events that have occurred on a single object at one point in time, i.e., if a thread acquires an object and then later releases it, that is a single lock. The lock event count is the total number of individual lock events that have occurred for a given object, and therefore would include the event signalling the lock was acquired, and the event indicating the lock was released.
- ***THDNFY, *THDFYALL, *THDWAIT (V5R2 or after)**
  These events correspond to the use of the notify(), notifyAll(), and wait() methods in java.lang.Object. These events are summarized as in the lock case. On the Object Info Frame, there is a Thread Notify Pane containing information about the set of these events that have occurred on a single object. An entry for these events also appears in the Call Trace Pane at the point where the event occurs.

- ***CLSLOAD (V5R2 or after)**
  A class load event is generated whenever the loader loads a class in the application. Each event contains information about the name of the class loader and the DE optimization level if it has been compiled for Direct Execution. This information will be displayed on the Class Summary Table on the Main Frame.

- ***GBGCOLSWEEP (V5R1 or after)**

This event provides collection information about each GC sweep cycle that occurred during the collection, such as live objects, size collected, size of the heap, etc. It is summarized on the Garbage Collection Sweep Summary Table on the Job Frame.

- ***THDCRT and *THDDLT (V5R2 or after)**
  These events provide timestamp information for the thread creates and deletes that have occurred during the collection. It will also provide the task count identifier for the creator thread. This information is provided on the Thread Pane of the Job Frame.

Table of Contents | Previous | Next

4.2.5 OS/400 Release Differences

The current version of PTDV includes support for PEX collections generated on V5R2, but no longer supports collections created on V4R4.

- **Object Lock events**
  In V4R5 and V5R1, the *THDSTTCHG event is used to generate java lock events. In V5R2, the *THDSTTCHG event is no longer used. The *LCKSTR (for lock) and *UNLCK (for unlock) events must be used to obtain the same information that is available in previous releases with the *THDSTTCHG event.

- **Object Create events**
  In V4R5 and V5R1, the primitive array names are not available on object delete events. As a result, the delete counts for primitive arrays are not correct when processing collections generated on these releases. In V5R2, these names are available and delete counts are accurate.

- **Thread Notify, Thread Notify All, and Thread Wait**
  For collections created for V5R2 or later, PTDV provides support for the Thread Notify, Thread Notify All, and Thread Wait events. These events appear in the Call Trace Frame for the thread where the events occur; and a table summarizing these events is generated on the Object Group Frame and Object Info Frame for those classes that generate this type of event. Each Job Frame and Thread Frame will have an Event Summary Table containing these events if they occurred in the corresponding job or thread.

- **Thread Create and Thread Delete**
  These events are processed in collectios created for V5R2 or later. When these events are present, timestamp information for thread creates and deletes are added to the Thread table, and the task count identifier for the creating thread is available if Thread Create events are included in the collection.

- **Class Load**
  PTDV supports Class Load events for collections created on V5R2 or later. When these events are present, the class name, class loader name, and DE optimization level information is available and is shown on the Object Summary Table for the initial frame.

- **Primitive array names**
  On releases prior to V5R2, if the object associated with any type of Java event was a primitive array (i.e., int[]), the name was not available to PTDV so appeared in the object tables as "unknown". In V5R2, these names are available and show up correctly in the tables.

- **Instruction counters**

  On releases prior to V5R2, when method/procedure entry and exit events are present in the collection, the instruction counts for each method/procedure are available in the PEX data and are displayed on all panes containing method information. This information is no longer available in V5R2, and if displayed on a pane containing method or procedure information, any columns referring to instruction counts will be 0.

Table of Contents | Previous | Next

4.3 Generating a PEX collection

The next sections cover useful information on generating PEX collections for use by PTDV.

4.3.1 Generating PEX Definitions

4.3.2 Analyzing Java applications

4.3.3 Analyzing ILE/OPM applications

4.3.4 Starting and Ending the Trace

4.3.5 Limiting the Size of the Trace

Table of Contents | Previous | Next

4.3.1 Generating PEX Definitions

PTDV processes and displays performance information found in PEX collection data. To collect this type of data for your application, you will need to create a PEX definition containing events that are supported and can be viewed by PTDV. If you have installed the PTDV library as described in the Installation documentation, you can create a standard set of PEX definitions for use with PTDV by calling the program:

CALL PTDV/ADDPTDVDFN

For more information about PEX definitions and PEX events (including information on creating custom definitions) for use with PTDV, please see the section on Creating PEX definitions. The above program will create the definitions described in that section.

Table of Contents | Previous | Next

4.3.2 Analyzing Java applications

PTDV can see Java method information for methods running with the interpreter, the Just In Time (JIT) compiler, or Direct Execution (DE). See the iSeries Java documentation for information on selecting which mode to use when running a Java program.

In order to see Java method information in the Performance Trace Data Visualizer, entry/exit hooks must be enabled in the Java classes. There are two basic methods of doing this:

Run your program with the JIT (or interpreter), and enabled method entry/exit hooks. One way to force your program to run with the JIT is to set the property java.compiler=jitc. To enable performance hooks, you can set os400.enbpfrcol=1. For more information on how to set these properties, see the FAQ.

-or-

Enable method entry/exit hooks in transformed (aka DE'd) Java code. In order to do this, you will need to use CRTJVAPGM with the ENBPFRCOL(*ENTRYEXIT) option to create Java program objects with hooks. You will need to do this for each jar, zip, or class file you want hooks for. The Performance Data Visualizer can view method information for DEd code at any optimization level, but level 40 is preferred, since this is what your application should normally be running at. Using CRTJVAPGM on large files or on slow machines may result in a long wait. If you are running on a machine with a greater batch than interactive capacity, be sure to submit the CRTJVMPGM to batch.

If you are doing performance data collection using Direct Execution java code, then the classes in the JDK will not have hooks enabled and will therefore not show up in your trace.

Note that only classes loaded via the system or bootstrap class loaders can run under Direct Execution mode. Classes loaded with a user class loader (such as servlets running with WebSphere) will always run with the JIT compiler (unless you explicitly place these classes in the system classpath).

If you are interested in seeing what object types are being created and/or which object types are being locked, you can use a PEX definition containing the *OBJCRT and *THDSTTCHG Java events. Object information can be collected with or without enabling entry/exit hooks. For more information on setting up PEX definitions, see the Creating PEX definitions section.

Warning: Do not attempt to run the CRTJVAPGM command against the java.zip, sun.zip, classes.zip, or rt.jar files supplied by IBM on your iSeries. If you do, they will become damaged and you will need to re-install the JV1 product to recover. You **can** run CRTJVAPGM against the classes used by the native JDBC driver (part of JV1), the Java Toolbox classes installed with JC1, and the open-source version of the Java Toolbox (JTOpen). Each release of IBM's WebSphere Application Server contains several jar files that need special options if they are to be re-created with hooks using CRTJVAPGM. Therefore, you should not use CRTJVAPGM on jar files shipped with WebSphere. In all of these cases, you can force use of the JIT compiler and enable hooks using the first method listed above to see performance information on these classes.

Table of Contents Previous Next

4.3.3 Analyzing ILE/OPM applications

If you want to trace code written in ILE or OPM languages, including programs called from Java using JNI (Java Native Interface), you must use a PEX defintion which includes the appropriate program events, and your application code must have been compiled to contain performance hooks. See Standard ILE/OPM Definitions for information on creating the correct definition.

In OPM code, standard performance hooks are included by default in every program entry point.

In ILE code, standard performance hooks can be included in your ILE modules and programs in one of two ways:

- On initial create of a module, set the parameter ENBPFRCOL to *ENTRYEXIT.
- For a program that already exists, use the CHGPGM or CHGSRVPGM command, specifying *ENTRYEXIT for the ENBPFRCOL parameter.

In ILE code, there is another type of hook, the trace job hook, which is used by the iSeries TRCJOB command and can also be generated in a PEX collection. These hooks can be generated by:

- Compiling your ILE code at optimization level 30 or below.
- Using the LICOPT CallTracingAtHightOpt when compiling your code at optimization level 40.

MI complex instructions can also be included in your PEX collection, and no hooking is necessary by the user to enable these.

Table of Contents | Previous | Next

4.3.4 Starting and Ending the Trace

Data collection begins once the STRPEX command is invoked. The type of data collected depends on the PEX definition specified on this command; the definition must already exist.

An example command:

STRPEX SSNID(***COLL_NAME***) OPTION(*NEW) DFN(PTDVJAVA)

In V5R2, setting the DFN parameter to the value '*SELECT' will open a list of the available PEX definitions for you to choose from.

Data collection stops once the ENDPEX command is invoked. For example:

ENDPEX SSNID(***COLL_NAME***)

Starting in release V5R1, specifying '*SELECT' on the SSNID parameter will allow you to view the set of collections that are currently active, along with the current event counts. Hitting F5 will refresh the data, allowing you to watch how your collection is growing. You can then end your active collection at an appropriate point.

Table of Contents | Previous | Next

4.3.5 Limiting the Size of the Trace

It is very easy to generate a huge number of PEX trace events in a very short period of time, especially when collecting data for a multithreaded program. In Thin Client mode, PTDV can process collections containing 5,000,000 events or more; however processing time can take quite a while in Thin Client mode. For Thick Client mode, a collection of less than 300,000 events is recommended.

There are several methods of limiting the amount of data in a trace. These include:

- Use a PEX definition that includes only the events you need. For example, traces which include method entry/exit events can be very large, so don't include these events in your PEX definition if you are only interested in object creates/locks.
- Likewise, if you are analyzing a Java locking problem, you may need method entry/exit events as well as Java lock events, but you can turn off the object create events. For more information on creating a PEX definition with only the necessary events, see the section on Creating PEX definitions.
- If multiple jobs are running on the system, change the PEX definition to collect events only for the job you are interested in. To do this, you can use the CHGPEXDFN command, and specify the job name in the JOB parameter.
- If you do need method entry/exit events in your trace, try to add entry/exit hooks only to the classes that you need to see the events for. For example, if you are using the CRTJVAPGM ENBPFRCOL(*ENTRYEXIT) method of generating hooks, only generate them for the jar files or classes that you are interested in. If the part of your code you are interested in normally runs in the JIT, then you can enable hooks for the JIT, but don't force all of the other code to run in the JIT. (In other words, specify the os400.enbpfrcol=1 property, but not java.compiler=jitc.) This can be particularly helpful when analyzing code which runs under WebSphere, because you can enable hooks for your application code (which normally runs with the JIT) but not for the WebSphere code (which normally runs DE).
- Run only a small number of transactions or invocations during the trace. For example, if you are analyzing a Java servlet, try just loading the servlet a couple times manually from your web browser, rather than tracing under a full load.
- Note: This method will not be effective for analyzing Java lock problems, since the locking behavior of your application may change significantly under heavier loads. But it can work very well for pathlength or object creation analysis.
- If none of the other methods will work, sometimes the best thing to do is just to start and end PEX as quickly as possible. One method of doing this is to start your program and let it ramp up to steady state. Then type the STRPEX and ENDPEX statements into two separate windows so that you can do the STRPEX, quickly switch windows, and just press enter to do the ENDPEX.

Table of Contents | Previous | Next

Chapter 5 Heap Analyzer

This chapter describes the Heap Analyzer component on the server side. Heap Analyzer provides tools for doing heap analysis, advanced debug, and application review.

The following topics will be covered:

- Libraries QYPBASE, QPYRTJW
- The types of collections available
- Database files
- Command documentation

Table of Contents | Previous | Next

5.1 Libraries

When Heap Analyzer is installed, two libraries are created on the iSeries. These libraries are named QYPBASE and QPYRTJW.

QYPBASE contains programs and commands needed by the iDoctor for iSeries client. It also contains objects that are used by both Heap Analyzer, Job Watcher and PEX Analyzer.

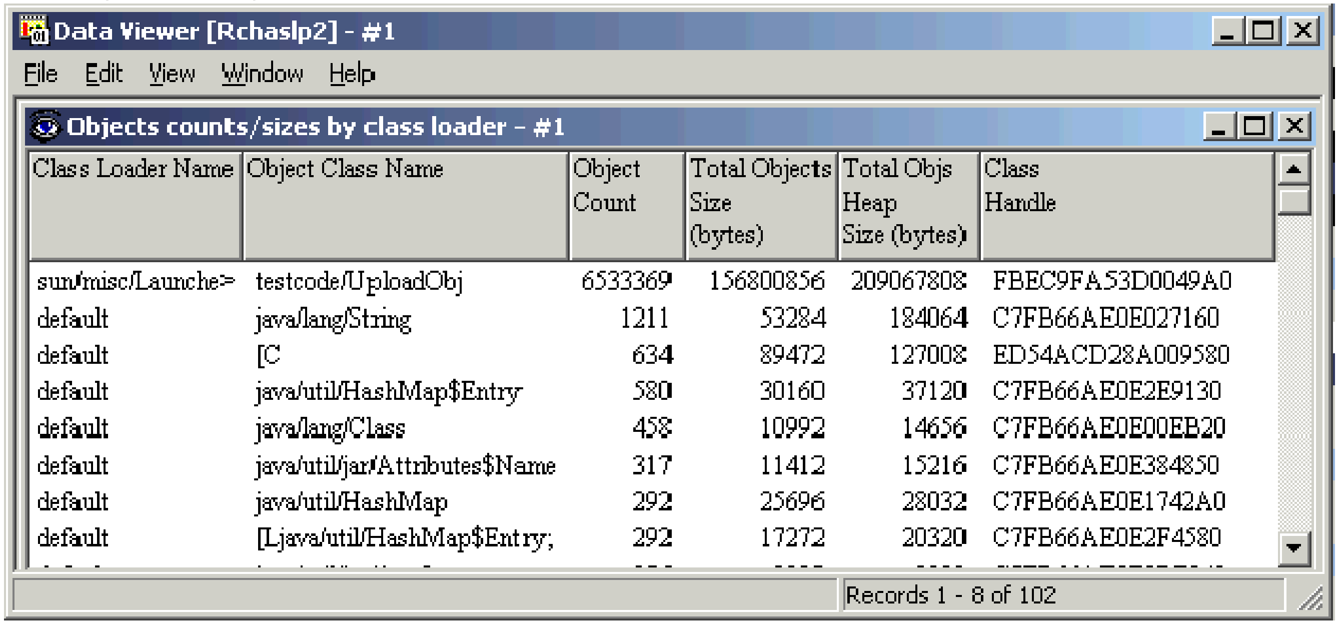
Library QPYRTJW is the Job Watcher/Heap Analyzer library for release V5R2. This library contains programs, commands and database files needed to create and work with Heap Analyzer data.

5.2 Types of Collections

Heap Analyzer provides two different modes for collecting information about a heap: Object Table Snapshot and the Object Create Profiler. Collections are created using the WCHJVA command or by using the client interface over a single job's JVM.

Object Table Snapshot - Similar to DMPJVM, however it is a non-intrusive snapshot which provides detailed information on the object counts, object size, and actual heap size used. DMPJVM requires the job to be held in order to collect heap object table statistics and it will not work on large Websphere heaps. Heap Analyzer runs in the background and will run on enormous heaps without any affect on the JVM operation while it's running.

An example of the output:


Data Viewer [Rchaslp2] - #1
File
Edit
View
Window
Help
Objects counts/sizes by class loader - #1

Class Loader Name	Object Class Name	Object Count	Total Objects Size (bytes)	Total Objs Heap Size (bytes)	Class Handle
sun/misc/Launche>	testcode/UploadObj	6533369	156800856	209067808	FBEC9FA53D0049A0
default	java/lang/String	1211	53284	184064	C7FB66AE0E027160
default	[C	634	89472	127008	ED54ACD28A009580
default	java/util/HashMap$Entry	580	30160	37120	C7FB66AE0E2E9130
default	java/lang/Class	458	10992	14656	C7FB66AE0E00EB20
default	java/util/jar/Attributes$Name	317	11412	15216	C7FB66AE0E384850
default	java/util/HashMap	292	25696	28032	C7FB66AE0E1742A0
default	[Ljava/util/HashMap$Entry;	292	17272	20320	C7FB66AE0E2F4580

Records 1 - 8 of 102

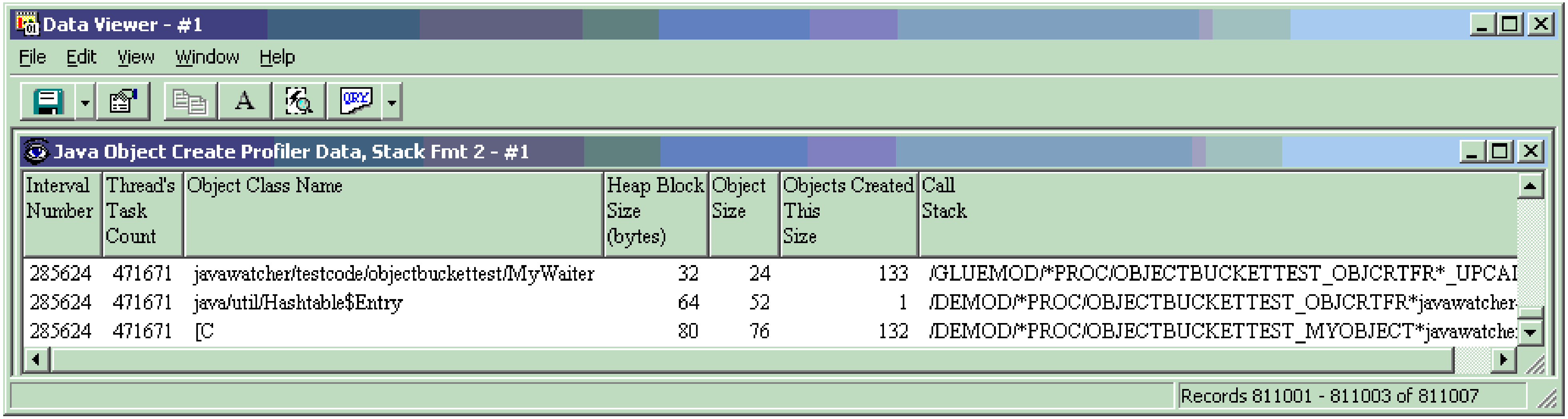
Object Create Profiler - Used to "profile" object creates, **in real time.** To really understand how this works you have to understand a little about the architecture of how the JVM handles object creates. Every object has two sizes associated with it, the actual object size and the heap size that object required. For instance in OS/400 V5R2 there are 23 bucket sizes, every object will fall into one of these buckets.

When you turn on the profiler for a JVM it causes every object created for each bucket and thread to be logged. The information logged includes:

1. A counter for the bucket in this thread
2. The object class name
3. The block size
4. The object size
5. The object, class, and loader address
6. Stack information to identify where the object was created from.

The code that does the harvesting of this data can not run as fast as 100's of threads can create 1000's of objects, so we get the information as fast as we can. What we do preserve is the count for each bucket. So at each interval that we harvest the data we will know how many objects of this bucket size were created and we will have the details for this specific one. With the information we've reviewed so far this has proven to be a very reliable way of profiling an application for who is creating objects, what objects they are and what size they are, even with a limited sample.

An example of the output:



Java Object Create Profiler Data, Stack Fmt 2 - #1

Interval Number	Thread's Task Count	Object Class Name	Heap Block Size (bytes)	Object Size	Objects Created This Size	Call Stack
285624	471671	javawatcher/testcode/objectbuckettest/MyWaiter	32	24	133	/GLUEMOD/*PROC/OBJECTBUCKETTEST_OBJCRTFR*_UPCAI
285624	471671	java/util/Hashtable$Entry	64	52	1	/DEMOD/*PROC/OBJECTBUCKETTEST_OBJCRTFR*javawatcher
285624	471671	[C	80	76	132	/DEMOD/*PROC/OBJECTBUCKETTEST_MYOBJECT*javawatche:

Records 811001 - 811003 of 811007

Table of Contents Previous Next

5.3 Heap Analyzer Output Files

Heap Analyzer provides two different types of analysis over a Java Virtual Machine geared towards helping a user analyze problems like a "runaway heap". They are known as the 'object table snapshot' collection and the 'object create profile' collection. Each type of collection uses several different output files that the client queries to produce meaningful reports and graphs. A collection is created via the WCHJVA command in library QPYRTJW.

The files below exist in the QPYRTJW library, and get duplicated when running a collection in a library for the first time. Each of these files will contain a member name matching each collection created in the given library unless otherwise noted.

The files that contain the data for a object table snapshot collection are:

File name	Description
QPYRTJVMH0	The control file that keeps track of the collections that are running or completed (for this type). One record will exist per successful execution of the WCHJVA command ACTION(*OBJTBLDMP). If the user specifies to create a watch of the same name as an existing one, the record will be updated.
QPYRTJVMH1	Class loader names. Contains one member per Object Table Snapshot collection.
QPYRTJVMH2	Summarized object information (object class name, sizes, loader ID, etc). Contains one member per Object Table Snapshot collection.

The files that contain the data for a object create profile collection are:

File name	Description
QPYRTJVMF	The control file that keeps track of the collections that are running or completed (for this type). One record will exist per execution of the WCHJVA command ACTION(*OBJCRTPROFILE). If the user specifies to create a watch of the same name as an existing one, the record will be updated.
QPYRTJVMF0	Interval/samples file. One record per interval/sample. Contains one member per Object Create Profile collection.
QPYRTJVMF1	Thread information file. One record per interval/sample per thread found. Contains one member per Object Create Profile collection.
QPYRTJVMF2	Flight recorder profile data. (also see QPYRTJVMF3 file... only one of these gets populated per run). One record per interval/sample per thread per each of 13 object bucket sizes for threads/buckets that had one or more objects created since the last sample. This file also has the invocation stack information provided within 25 separate fields.

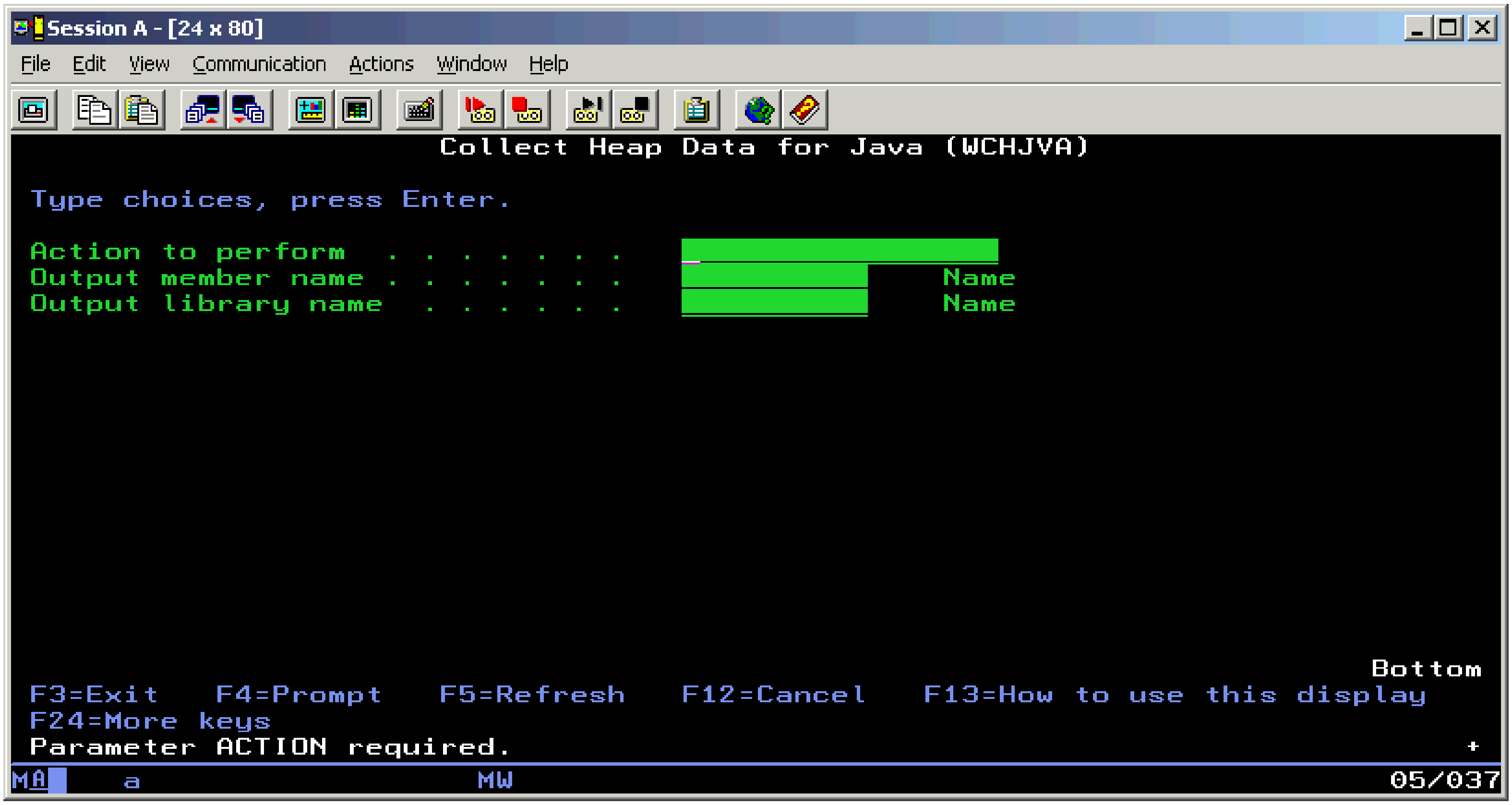
QPYRTJVMF3	Flight recorder profile data. (also see QPYRTJVMF2 file... only one of these gets populated per run). One record per interval/sample per thread per each of 13 object bucket sizes for threads/buckets that had one or more objects created since the last sample. This file also has the invocation stack information provided within a single field.

Table of Contents | Previous | Next

5.4 WCHJVA command

Heap Analyzer is shipped in library QPYRTJW. You need to add this library to your library list prior to running from the green screen.

To create a collection, prompt the WCHJVA command for the following display:



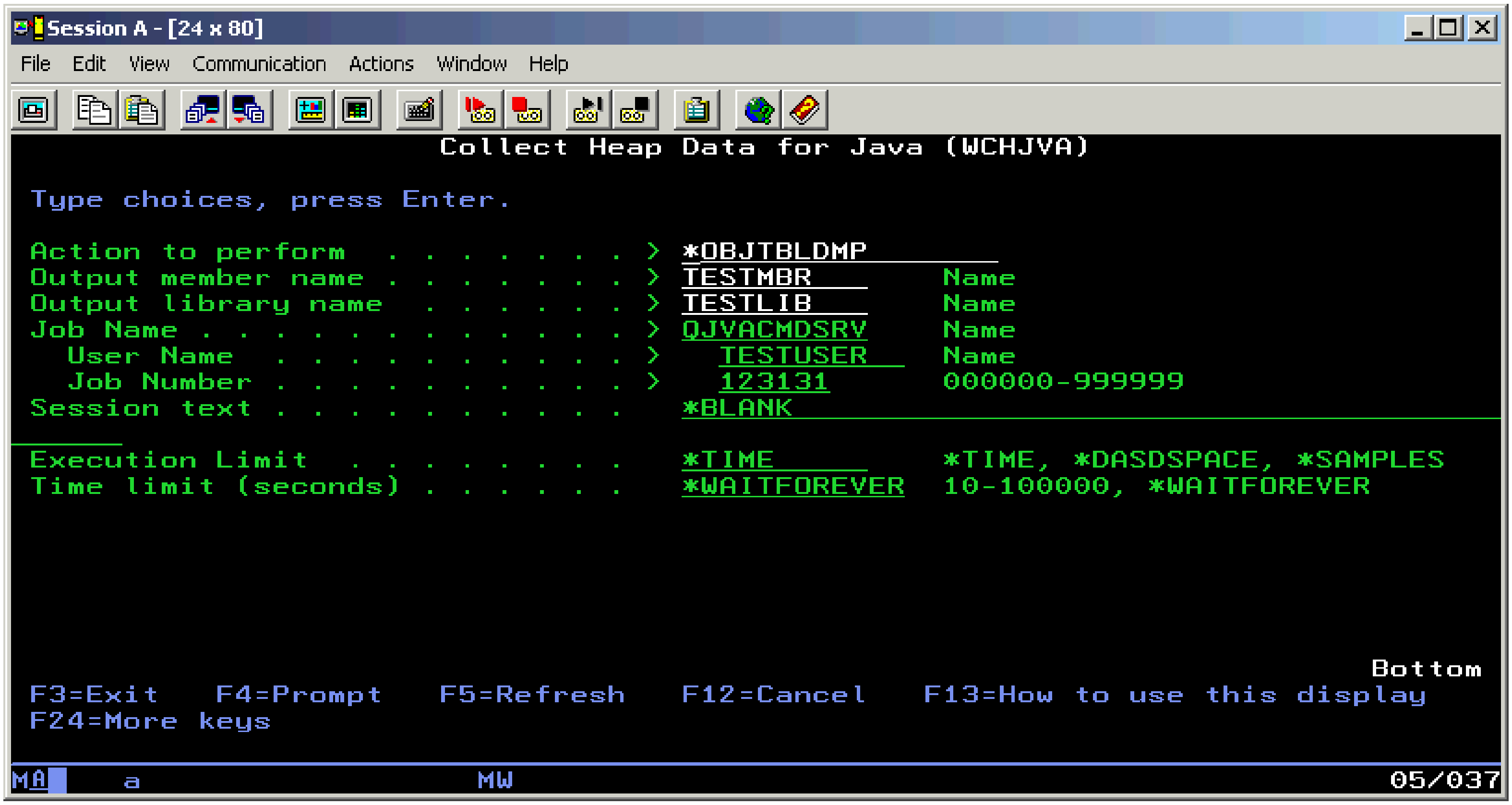
```
                   Collect Heap Data for Java (WCHJVA)

 Type choices, press Enter.

 Action to perform . . . . . . .
 Output member name . . . . . . .              Name
 Output library name  . . . . . .              Name

                                                                      Bottom
 F3=Exit    F4=Prompt    F5=Refresh    F12=Cancel    F13=How to use this display
 F24=More keys
 Parameter ACTION required.                                                +
MA    a                    MW                                           05/037
```

There are three options that you can select for the **Action to perform** parameter.

- *OBJTBLDMP - Used to get a snapshot of the object table in the heap.
- *OBJCRTPROFILE - Used to collect object create statistics.
- *OBJROOTFINDER - Used to investigate an object for roots **(in development and not yet working)**

The **Output member name** and **Output library name** define the name of the library and member name to use when creating the collection. When running from green screen the library you specify must exist or the command will fail.

The available options when the **Action to perform** parameter is *OBJTBLDMP are:

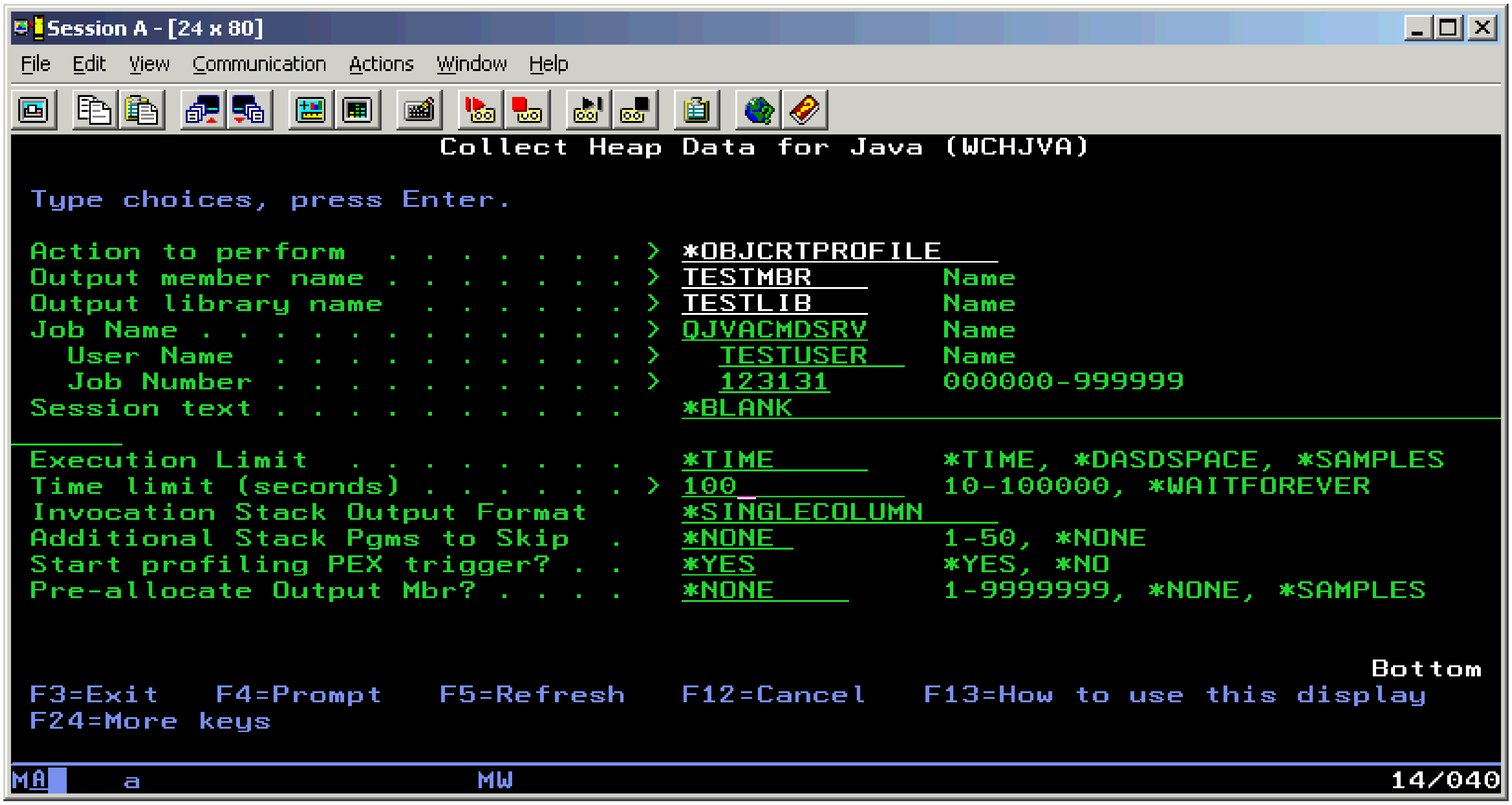


```
                        Collect Heap Data for Java (WCHJVA)

 Type choices, press Enter.

 Action to perform . . . . . . . > *OBJTBLDMP
 Output member name . . . . . . . > TESTMBR       Name
 Output library name  . . . . . . > TESTLIB       Name
 Job Name . . . . . . . . . . . . > QJVACMDSRV    Name
   User Name  . . . . . . . . . . >   TESTUSER    Name
   Job Number . . . . . . . . . . >   123131      000000-999999
 Session text . . . . . . . . . .   *BLANK

 Execution Limit  . . . . . . . .   *TIME         *TIME, *DASDSPACE, *SAMPLES
 Time limit (seconds) . . . . . .   *WAITFOREVER  10-100000, *WAITFOREVER







                                                                         Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys

MA      a                MW                                              05/037
```

The **Job Name** is the job that contains the JVM that you need to collect information about.

The **Execution Limit** should be set to *TIME and then set the Time Limit as appropriate for your situation. When possible it's recommended that you use *waitforever (no timeout). If you specify a time limit and you hit it then you have to be aware that you are working with incomplete data, and use it accordingly. The volume of data collected is relatively low and we are not collecting multiple samples so *DASDSPACE and *SAMPLES cannot be used and in fact are not available selections when using the GUI.

The available options when the **Action to perform** parameter is *OBJCRTPROFILE are:



```
                        Collect Heap Data for Java (WCHJVA)

 Type choices, press Enter.

 Action to perform . . . . . . . > *OBJCRTPROFILE
 Output member name . . . . . . . > TESTMBR       Name
 Output library name  . . . . . . > TESTLIB       Name
 Job Name . . . . . . . . . . . . > QJVACMDSRV    Name
   User Name  . . . . . . . . . . >   TESTUSER    Name
   Job Number . . . . . . . . . . >   123131      000000-999999
 Session text . . . . . . . . . .   *BLANK

 Execution Limit  . . . . . . . .   *TIME         *TIME, *DASDSPACE, *SAMPLES
 Time limit (seconds) . . . . . . > 100           10-100000, *WAITFOREVER
 Invocation Stack Output Format     *SINGLECOLUMN
 Additional Stack Pgms to Skip  .   *NONE         1-50, *NONE
 Start profiling PEX trigger? . .   *YES          *YES, *NO
 Pre-allocate Output Mbr? . . . .   *NONE         1-9999999, *NONE, *SAMPLES


                                                                         Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys

MA      a                MW                                              14/040
```

The **Job Name** is the job that contains the JVM that you need to collect information about.

The **Execution Limit** can be set for TIME, DASDSPACE, or SAMPLES and should be set appropriately based on any resource concerns. DO NOT use *TIME without a limit set as you would eventually run out of space if the JVM remained active.

The **Invocation Stack Output Format** should be set to *SINGLECOLUMN for the format. This will put all the stack information into a single "STACK" field in QPYRTJVMF3 file. The *SEPARATECOLUMNS format is not often used and will likely be removed from future support.

The **Additional Stack Programs to Skip** is a parameter that allows you to skip x number of frames. Our stack support gives us a 5 frame view of the request to create the object that was logged, this skip parameter allows this view to float on the call stack, the default view is to skip the first (oldest) 4 frames and then log 5 through 9. If you supply a skip count that puts us off the end of the stack, rather than fail, you will get the default output. The stack information is presented in a single field (*SINGLECOLUMN) with the most recent, or top of the stack, on the left side of the data.

The **Start profiling PEX trigger** parameter should be set to *yes, with this setting the PEX support required for the logging to occur will automatically be started (and ended) when you submit the command. The only reason to specify *no would be if the PEX definition is already active for some reason. The PEX definition has to have the Java *Service Event enabled for the object creates to be logged.

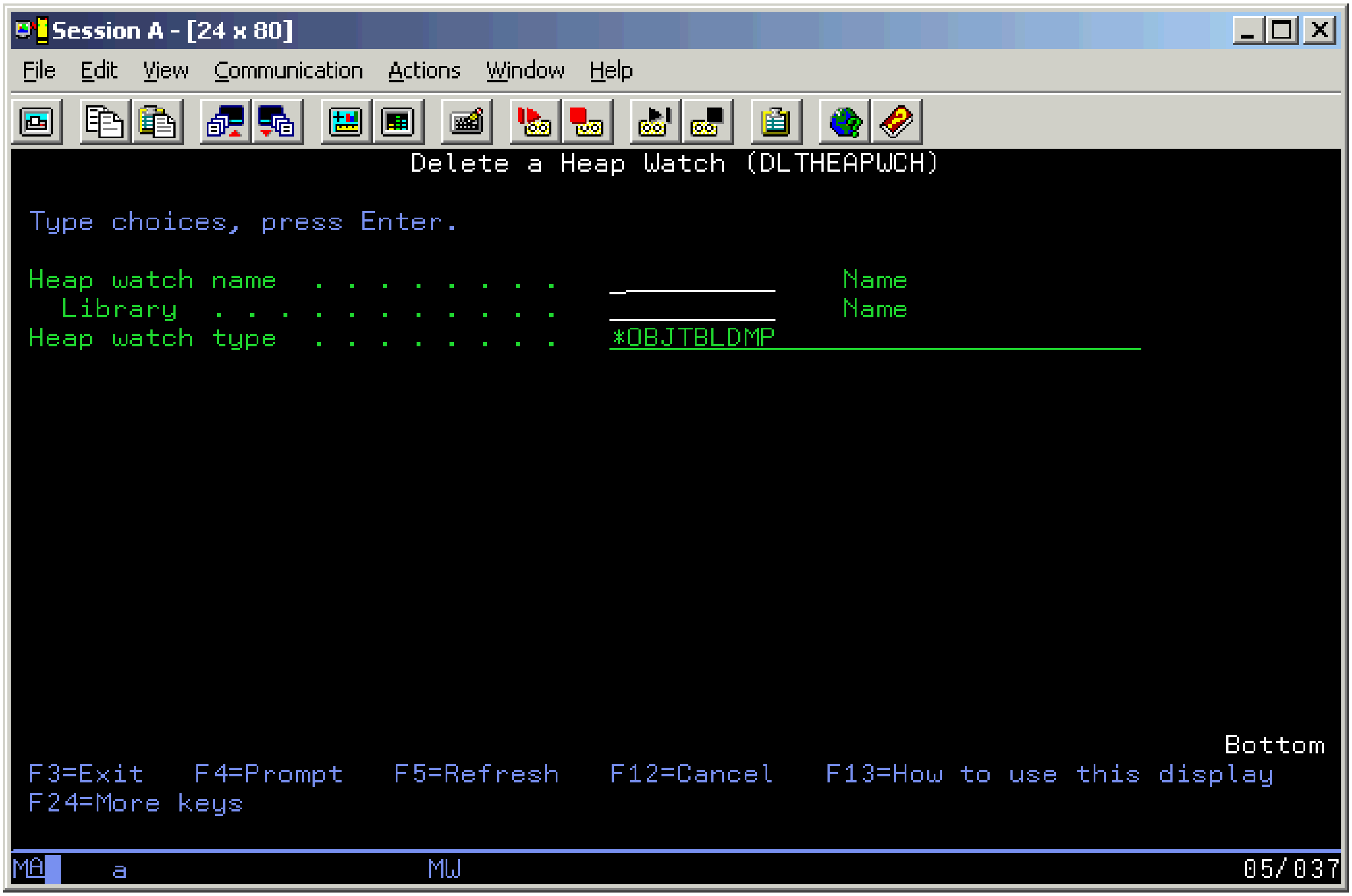
The **Pre-allocate Output Mbr** parameter can be used in some cases to get more consistent times between intervals or collections. In some cases you can notice delays or gaps between some of the intervals, this is due to the overhead of allocating space for the output member, with this setting you can take care of this work up front and avoid these delays at collection time.

Table of Contents Previous Next

5.5 DLTHEAPWCH command

The DLTHEAPWCH command removes a collection (a.k.a. Heap Watch) from a users library. To use this command simply specifiy the collection name (member name) , library name and the type of collection *OBJTBLDMP (shown as type 'object table snapshot' in the GUI) or *OBJCRTPROFILE (shown as type 'object create profile' in the GUI).

Specifying the correct type of collection to delete is critical because each type of collection stores data in a different set of files.



```
                         Delete a Heap Watch (DLTHEAPWCH)

 Type choices, press Enter.

 Heap watch name . . . . . . . . .   __________     Name
   Library . . . . . . . . . . . .   __________     Name
 Heap watch type . . . . . . . . .   *OBJTBLDMP________________________

                                                                        Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys

MA     a                    MW                                          05/037
```

Table of Contents Previous Next

5.6 Using iDoctor to investigate Heap Growth

Table of Contents | Previous | Next

5.7 ADDPRDACS command

This command applies an access code to your system. To use simply pass the access code your were provided to this command. This command can be found in either the QYPINT, QYPBASE or QPYRTJW libraries depending on the component(s) installed on the system.

Table of Contents | Previous | Next

Part III Client-side components

This part covers the client-side (graphical screen interfaces) for each iDoctor for iSeries component.

Table of Contents Previous Next

Chapter 1 Common Graphical Interfaces

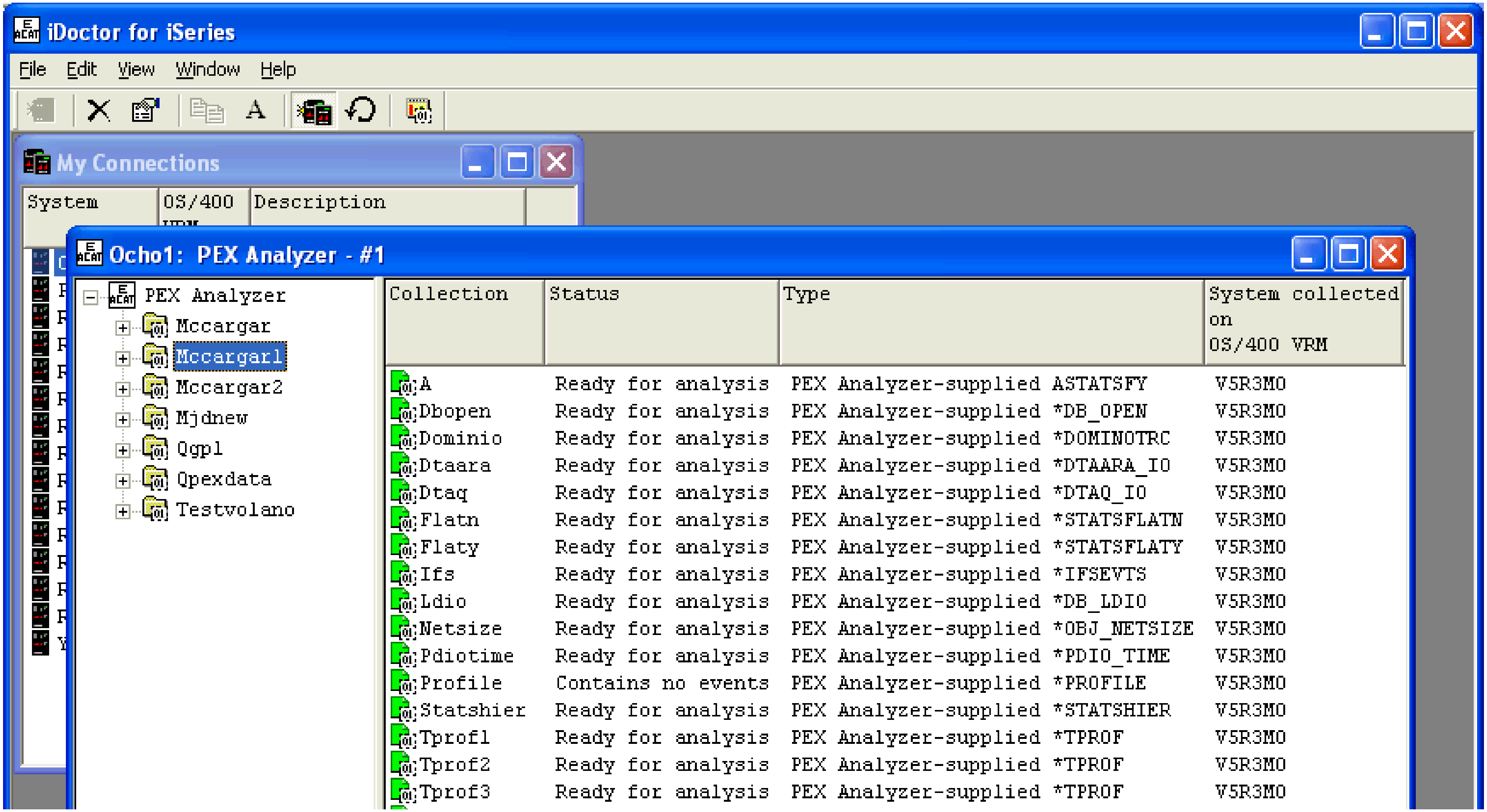
This chapter provides documentation for the iDoctor for iSeries interfaces that are found in all components (except PTDV).

1.0 Overview

iDoctor for iSeries may be launched via the desktop icon or using the Start Menu. The shortcut folder on the Start Menu is Start -> Programs -> iDoctor for iSeries.

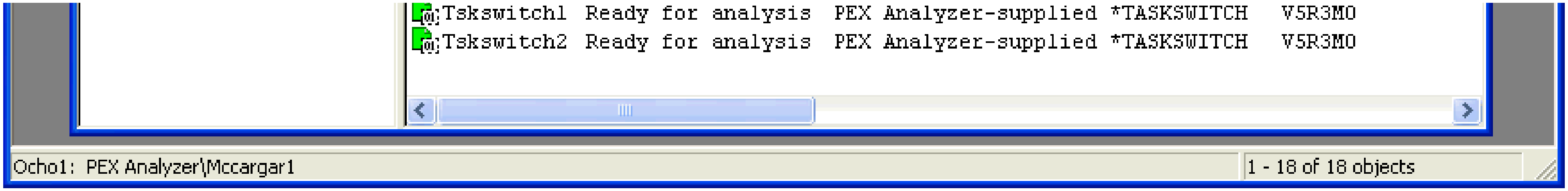
The Main Window

The heart of the iDoctor for iSeries application is the interface shown below. All components are displayed within this GUI each within a seperate "tree/list" view. Each view can be resized, tiled or manipulated at runtime. Multiple component views (a.k.a tree/list) views may be in use within the Main Window as needed.



- PEX Analyzer
  - Mccargar
  - Mccargar1
  - Mccargar2
  - Mjdnew
  - Qgpl
  - Qpexdata
  - Testvolano

Collection	Status	Type	System collected on OS/400 VRM
A	Ready for analysis	PEX Analyzer-supplied ASTATSFY	V5R3M0
Dbopen	Ready for analysis	PEX Analyzer-supplied *DB_OPEN	V5R3M0
Dominio	Ready for analysis	PEX Analyzer-supplied *DOMINOTRC	V5R3M0
Dtaara	Ready for analysis	PEX Analyzer-supplied *DTAARA_IO	V5R3M0
Dtaq	Ready for analysis	PEX Analyzer-supplied *DTAQ_IO	V5R3M0
Flatn	Ready for analysis	PEX Analyzer-supplied *STATSFLATN	V5R3M0
Flaty	Ready for analysis	PEX Analyzer-supplied *STATSFLATY	V5R3M0
Ifs	Ready for analysis	PEX Analyzer-supplied *IFSEVTS	V5R3M0
Ldio	Ready for analysis	PEX Analyzer-supplied *DB_LDIO	V5R3M0
Netsize	Ready for analysis	PEX Analyzer-supplied *OBJ_NETSIZE	V5R3M0
Pdiotime	Ready for analysis	PEX Analyzer-supplied *PDIO_TIME	V5R3M0
Profile	Contains no events	PEX Analyzer-supplied *PROFILE	V5R3M0
Statshier	Ready for analysis	PEX Analyzer-supplied *STATSHIER	V5R3M0
Tprof1	Ready for analysis	PEX Analyzer-supplied *TPROF	V5R3M0
Tprof2	Ready for analysis	PEX Analyzer-supplied *TPROF	V5R3M0
Tprof3	Ready for analysis	PEX Analyzer-supplied *TPROF	V5R3M0



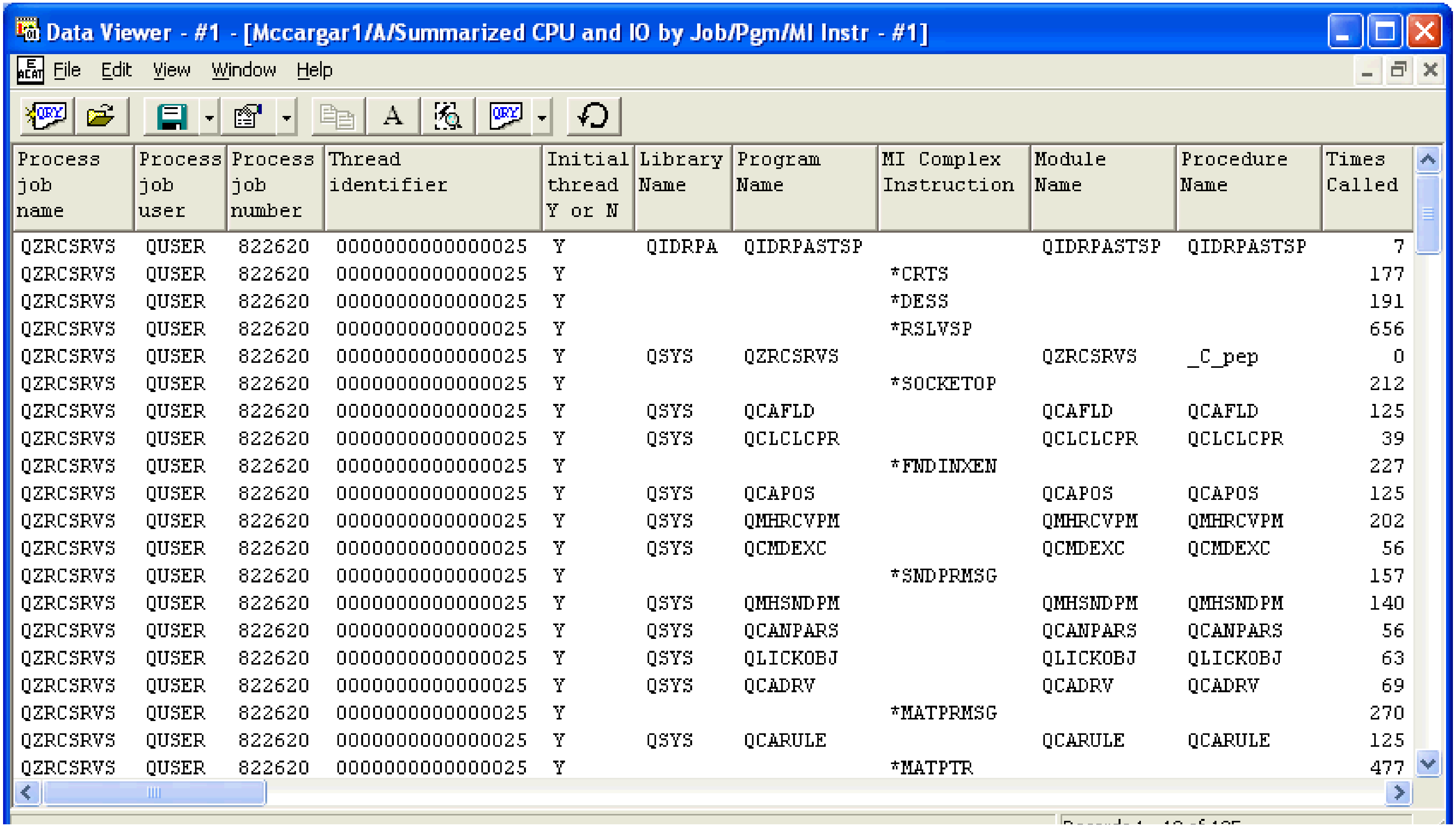
Tskswitch1 Ready for analysis PEX Analyzer-supplied *TASKSWITCH V5R3M0
Tskswitch2 Ready for analysis PEX Analyzer-supplied *TASKSWITCH V5R3M0

Ocho1: PEX Analyzer\Mccargar1 | 1 - 18 of 18 objects

[The Main Window]

Whenever you want to look at a table or graph, another window called the Data Viewer will be opened for that purpose. You can have several Data Viewers open at one time over a single main window. Only one Main Window can be open per running instance of the iDoctor application.

An example of a Data Viewer window is shown below.



Data Viewer - #1 - [Mccargar1/A/Summarized CPU and IO by Job/Pgm/MI Instr - #1]
File
Edit
View
Window
Help

Process job name	Process job user	Process job number	Thread identifier	Initial thread Y or N	Library Name	Program Name	MI Complex Instruction	Module Name	Procedure Name	Times Called
QZRCSRVS	QUSER	822620	0000000000000025	Y	QIDRPA	QIDRPASTSP		QIDRPASTSP	QIDRPASTSP	7
QZRCSRVS	QUSER	822620	0000000000000025	Y			*CRTS			177
QZRCSRVS	QUSER	822620	0000000000000025	Y			*DESS			191
QZRCSRVS	QUSER	822620	0000000000000025	Y			*RSLVSP			656
QZRCSRVS	QUSER	822620	0000000000000025	Y	QSYS	QZRCSRVS		QZRCSRVS	_C_pep	0
QZRCSRVS	QUSER	822620	0000000000000025	Y			*SOCKETOP			212
QZRCSRVS	QUSER	822620	0000000000000025	Y	QSYS	QCAFLD		QCAFLD	QCAFLD	125
QZRCSRVS	QUSER	822620	0000000000000025	Y	QSYS	QCLCLCPR		QCLCLCPR	QCLCLCPR	39
QZRCSRVS	QUSER	822620	0000000000000025	Y			*FNDINXEN			227
QZRCSRVS	QUSER	822620	0000000000000025	Y	QSYS	QCAPOS		QCAPOS	QCAPOS	125
QZRCSRVS	QUSER	822620	0000000000000025	Y	QSYS	QMHRCVPM		QMHRCVPM	QMHRCVPM	202
QZRCSRVS	QUSER	822620	0000000000000025	Y	QSYS	QCMDEXC		QCMDEXC	QCMDEXC	56
QZRCSRVS	QUSER	822620	0000000000000025	Y			*SNDPRMSG			157
QZRCSRVS	QUSER	822620	0000000000000025	Y	QSYS	QMHSNDPM		QMHSNDPM	QMHSNDPM	140
QZRCSRVS	QUSER	822620	0000000000000025	Y	QSYS	QCANPARS		QCANPARS	QCANPARS	56
QZRCSRVS	QUSER	822620	0000000000000025	Y	QSYS	QLICKOBJ		QLICKOBJ	QLICKOBJ	63
QZRCSRVS	QUSER	822620	0000000000000025	Y	QSYS	QCADRV		QCADRV	QCADRV	69
QZRCSRVS	QUSER	822620	0000000000000025	Y			*MATPRMSG			270
QZRCSRVS	QUSER	822620	0000000000000025	Y	QSYS	QCARULE		QCARULE	QCARULE	125
QZRCSRVS	QUSER	822620	0000000000000025	Y			*MATPTR			477

Records 1 - 19 of 135

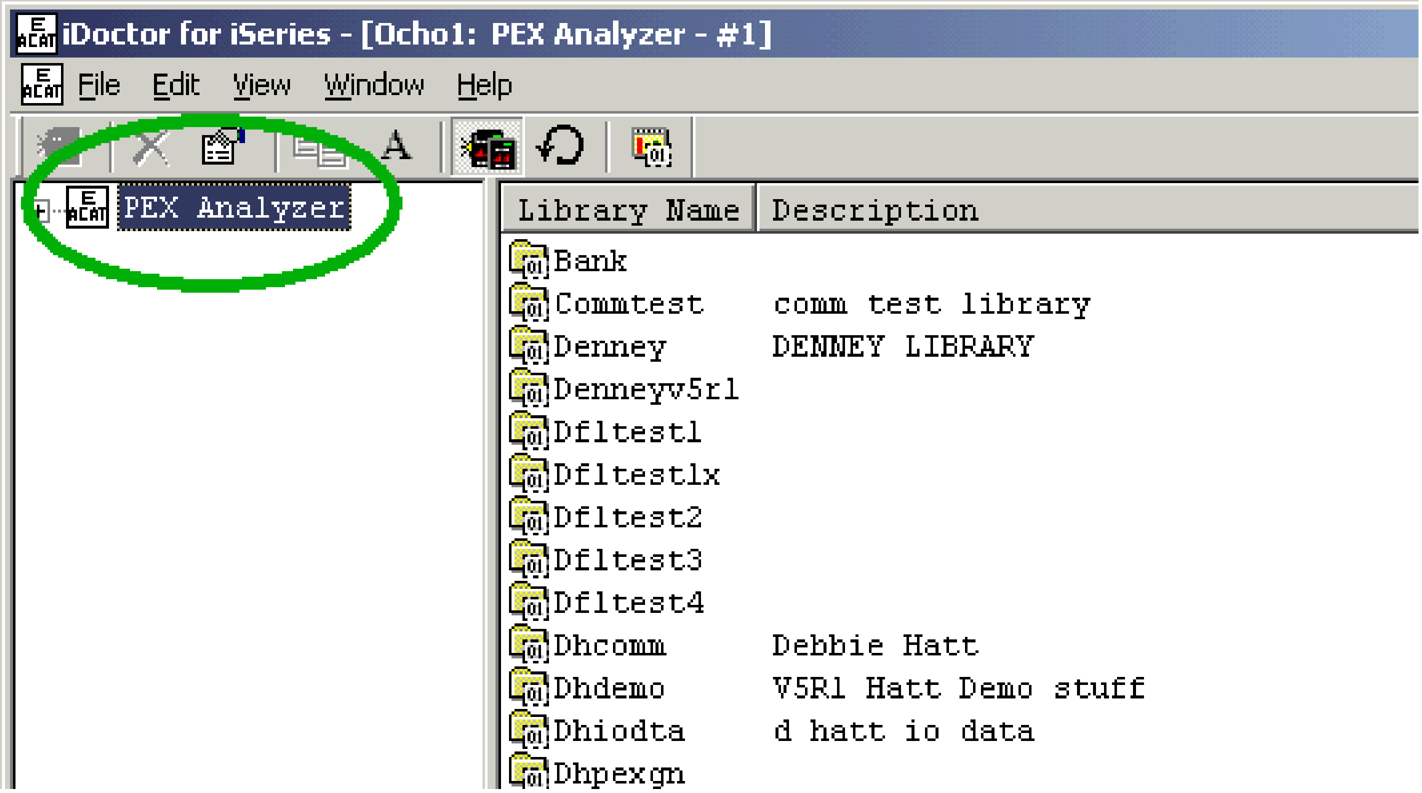
[The Data Viewer]

The Main Window and the Data Viewer as well as other interfaces for defining queries and graphs are common across all components. These interfaces will be defined in greater detail in the next sections.

Table of Contents | Previous | Next

1.0.1 Component Property Pages

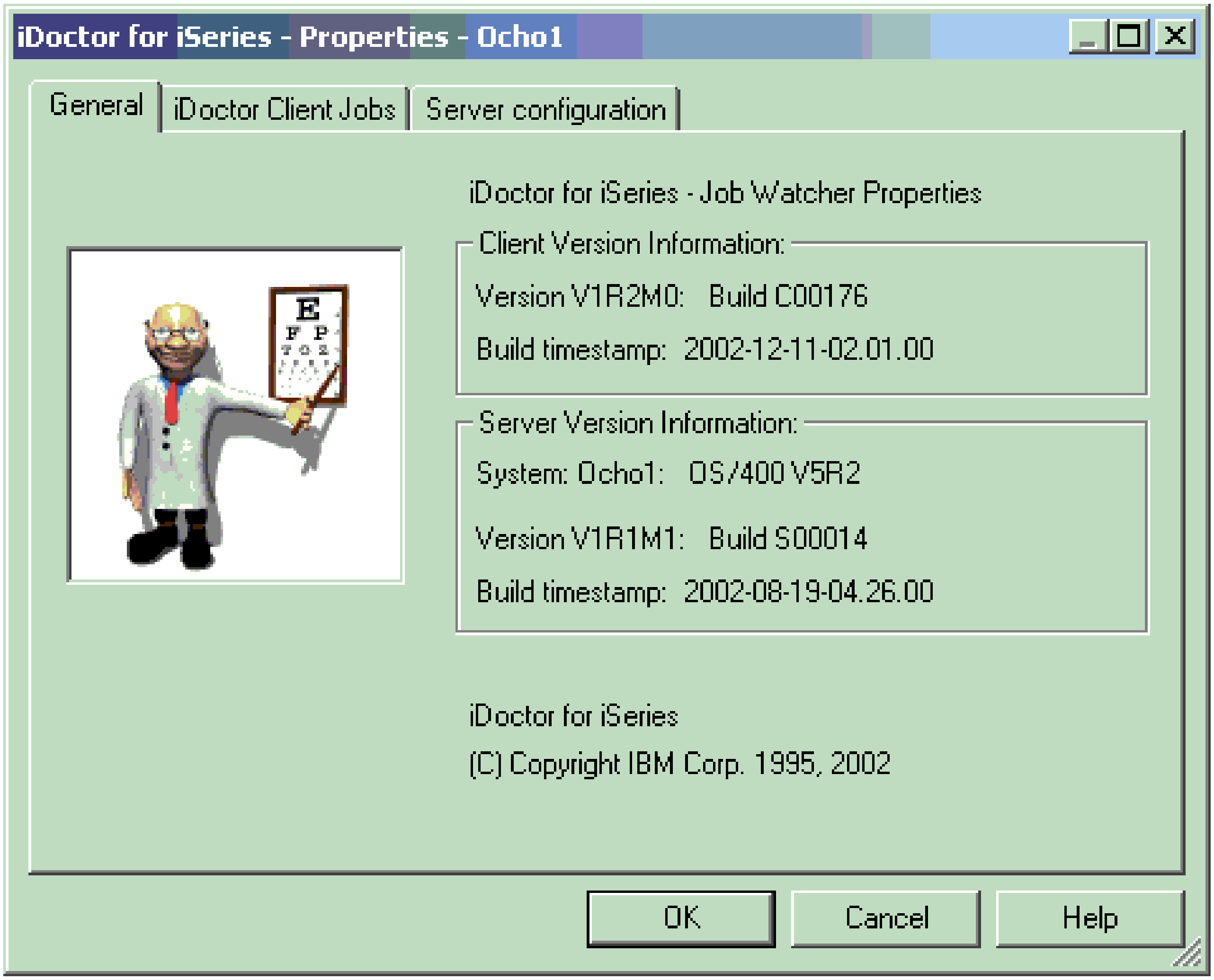
Each component view has a property page available by right-clicking on the component icon and choosing the Properties... menu. The component icon is either the 'PEX Analyzer', 'Job Watcher', 'Heap Analyzer' or 'Object Explorer' icon depending on the particular component view you are working with.



Library Name	Description
Bank	
Commtest	comm test library
Denney	DENNEY LIBRARY
Denneyv5r1	
Dfltest1	
Dfltest1x	
Dfltest2	
Dfltest3	
Dfltest4	
Dhcomm	Debbie Hatt
Dhdemo	V5R1 Hatt Demo stuff
Dhiodta	d hatt io data
Dhpexgn	

[PEX Analyzer component icon]

An example of a component property page for Job Watcher is shown below:



iDoctor for iSeries - Job Watcher Properties

Client Version Information:

Version V1R2M0: Build C00176

Build timestamp: 2002-12-11-02.01.00

Server Version Information:

System: Ocho1: OS/400 V5R2

Version V1R1M1: Build S00014

Build timestamp: 2002-08-19-04.26.00

iDoctor for iSeries

(C) Copyright IBM Corp. 1995, 2002

[Job Watcher component properties displaying client and server version information.]

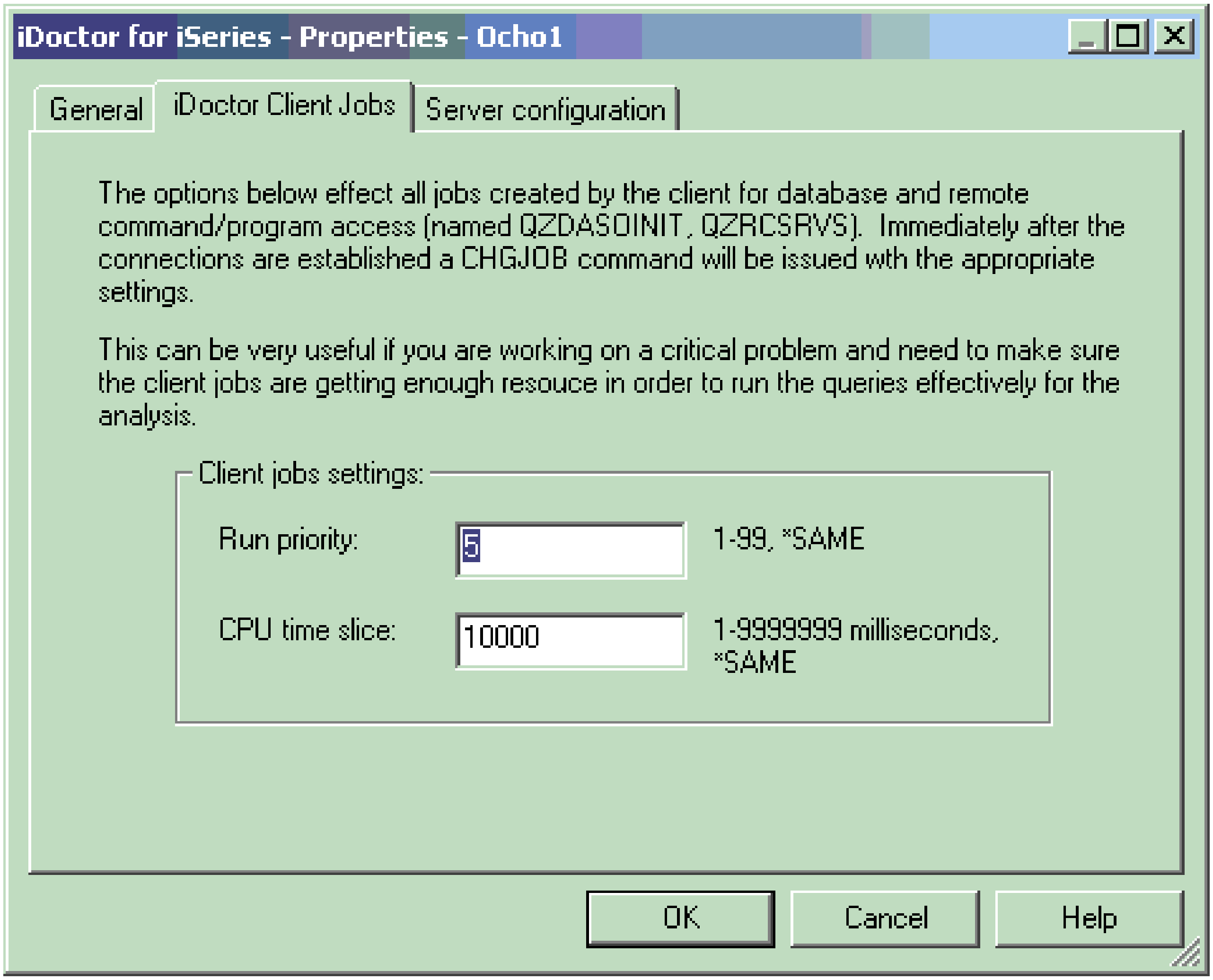
The following information is supplied within the General page of this window:

Client Version Information	**Description**
Version	The iDoctor for iSeries component version/release/mod level of the client. **Note:** The version and build number of the client is always the same regardless of the component you are currently working with.
Build	The build number of the client your are currently running. This number goes up everytime a new build is run for internal (IBM only) testing or for customer use. There will be "internal only" builds resulting in perceived gaps in the build number sequence for customers, but please disregard this.
Build timestamp	The date/time the client build was produced. This value is shown in yyyy-mm-dd-hh.mm.ss format.

Server Version Information	**Description**

System name	The system that the current component view is connected to.
OS/400	The version and release of OS/400 on the active system.
Version	The iDoctor for iSeries component version/release/mod level installed on the server. The version and build number are different for each component. Only the version for the component you launched this window from will be listed.
Build	Build number of the component installed on the server side
Build timestamp	The date/time the server build was produced. This value is shown in yyyy-mm-dd-hh.mm.ss format.

The following is an example of the iDoctor Client Jobs page:



The options below effect all jobs created by the client for database and remote command/program access (named QZDASOINIT, QZRCSRVS). Immediately after the connections are established a CHGJOB command will be issued with the appropriate settings.

This can be very useful if you are working on a critical problem and need to make sure the client jobs are getting enough resouce in order to run the queries effectively for the analysis.

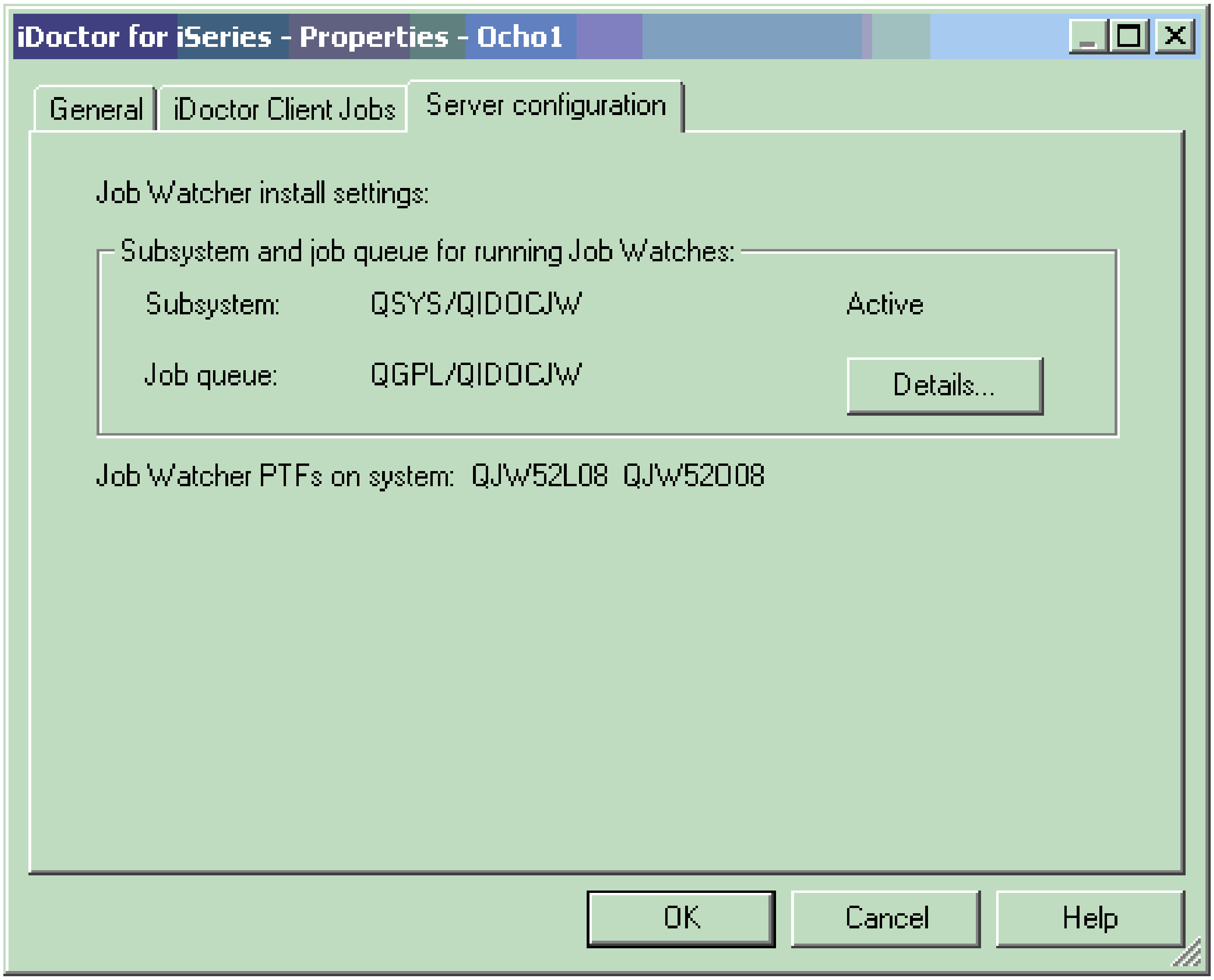
Client jobs settings:

Run priority: 5 1-99, *SAME

CPU time slice: 10000 1-9999999 milliseconds, *SAME

This page lets you set the run priority and CPU time slice of all iDoctor client jobs. This should only be set by advanced users. You must shut down the client and restart in order for these changes to take effect.

The following is an example of the Server configuration page.



Job Watcher install settings:

Subsystem and job queue for running Job Watches:

Subsystem: QSYS/QIDOCJW Active

Job queue: QGPL/QIDOCJW

Job Watcher PTFs on system: QJW52L08 QJW52008

The subsystem and job queue used for running Job Watches is shown on this page. If you were running PEX Analyzer you would see the subsystem and job queue used for running PEX Analysis jobs.

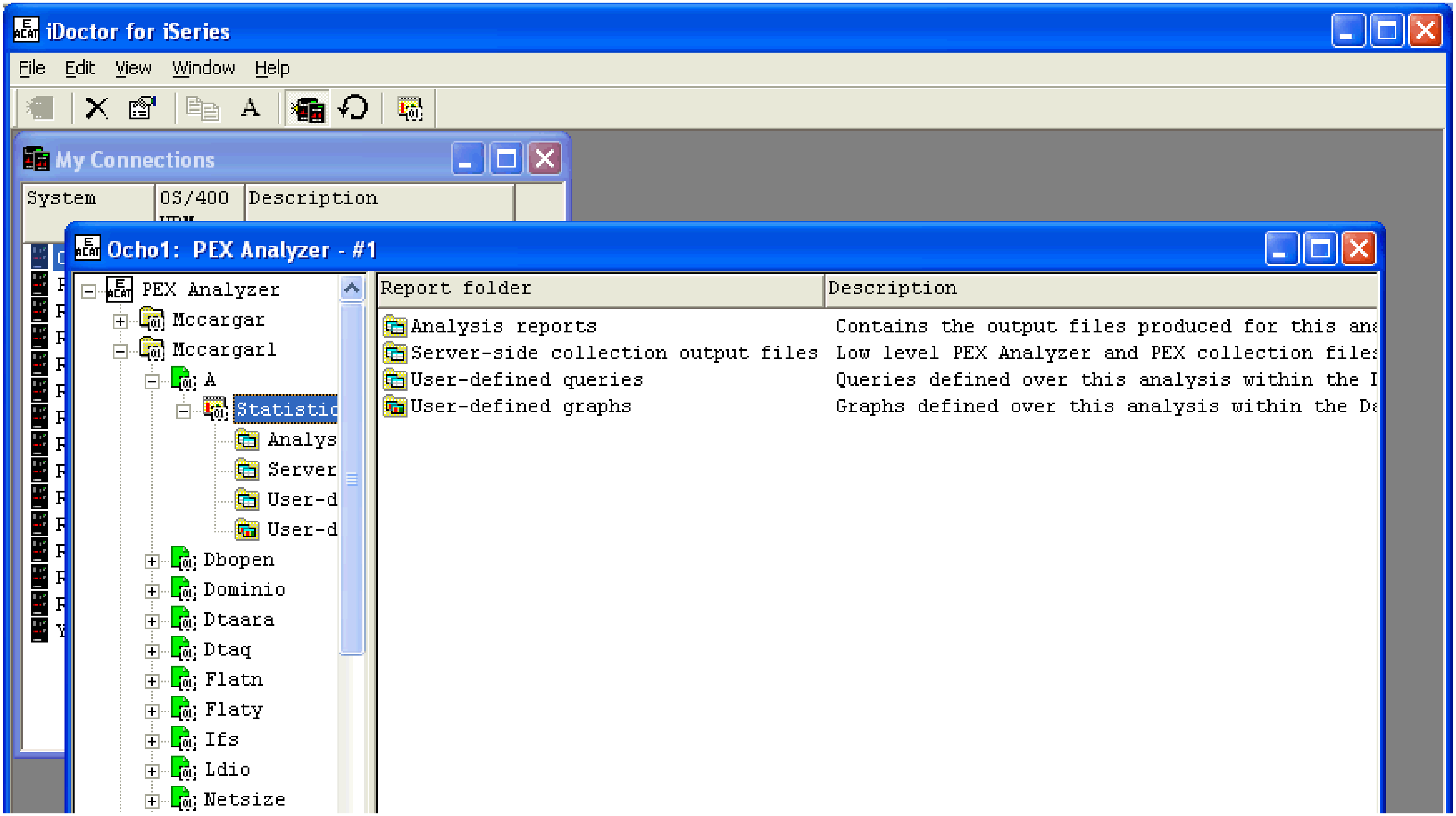
If you are running PEX Analyzer V4R5 this panel will also show fields where you can view/modify the location of the Java Toolbox JAR files (used when executing the Websphere analysis).

Table of Contents | Previous | Next

1.1 The Main Window

The Main Window displays the various component views as well as some additional views such as the "My Connections View". Each of these views within the Main Window will be discussed in greater detail in the next sections. This section will discuss the general use of the Main Window within iDoctor for iSeries.

The heart of the iDoctor for iSeries application is the interface shown below. All components are displayed within this GUI each within a seperate "tree/list" view. Each view can be resized, tiled or manipulated at runtime. You can have as many component (a.k.a "tree/list") views up at one time as you want.



Report folder	Description
Analysis reports	Contains the output files produced for this ana
Server-side collection output files	Low level PEX Analyzer and PEX collection file
User-defined queries	Queries defined over this analysis within the I
User-defined graphs	Graphs defined over this analysis within the D

Ocho1: PEX Analyzer\Mccargar1\A\Statistics flat analysis for all jobs/threads

1 - 4 of 4 objects

[The Main Window]

Main Window Menu Options

The table below outlines the different types of operations that may be performed within the Main Window of iDoctor for iSeries.

File Menu	**Description**
Open Component	Opens the specified component on the selected system. This menu is only enabled when the My Connections View is open and a system name is selected.
Open New Data Viewer	Opens an empty Data Viewer for the active component view. This menu is only enabled when a component view is open and has focus.
Close	This will close the active view within the Main Window.
Add Connection	Use this menu to add a connection to the My Connections View. This menu is only available when the My Connections View is active.
Delete	This will delete a conneciton from the My Connections View. This menu is only available when the My Connections View is active.
Exit	Exits the iDoctor for iSeries application. All open windows including Data Viewers will be closed down.

Edit Menu	**Description**
Copy	Copies the current selection from the active view to the clipboard. This is only enabled when the actve view is a list view (i.e. You can't copy text from the tree portion of the tree/list view).
Set Font	Displays a window allowing you to change the font used for the list views in the iDoctor for iSeries application.
Preferences	Displays the preferences window letting you work with iDoctor for iSeries user settings.

View Menu	**Description**
My Connections	This menu will either show or hide the My Connections view. If the view is already open there will be a checkmark next to the menu.
PEX Definitions	Brings up a window that lets you work with PEX definitions on your system. This is only available when a PEX Analyzer component view is open and has focus.
Remote Command Status	This menu will either show or hide the Remote Command Status view. If the view is already open there will be a checkmark next to the menu.
Toolbar	This menu will either show or hide the toolbar. If the toolbar is already visible there will be a checkmark next to the menu.
Status Bar	This menu will either show or hide the status bar. If the status bar is already visible there will be a checkmark next to the menu.

Refresh Selected	This menu will refresh the currently selected portion of a tree/list view. If a tree item is selected and this menu is clicked, everything underneath the tree item, including the tree item will have its data refreshed. If the list has focus and this menu is clicked, the entire list will be refreshed.

Window Menu	**Description**
Cascade	Use this menu to rearrange all views in the Main Window in an overlapping sequence starting in the upper left corner of the window.
Tile Horizontally	Use this menu to rearrange all views in the Main Window such that each view will have an equal distribution of the available height in the Main Window. The views will not overlap each other.
Tile Vertically	Use this menu to rearrange all views in the Main Window such that each view will have an equal distribution of the available width in the Main Window. The views will not overlap each other.

Help Menu	**Description**
Contents	This menu will launch your web browser and takes you to the online documentation start page.
iDoctor for iSeries Download Page	Launches your web browser and takes you to the iDoctor download page.
About	This displays version information for the iDoctor for iSeries client.

Table of Contents Previous Next

1.1.1 Component Views

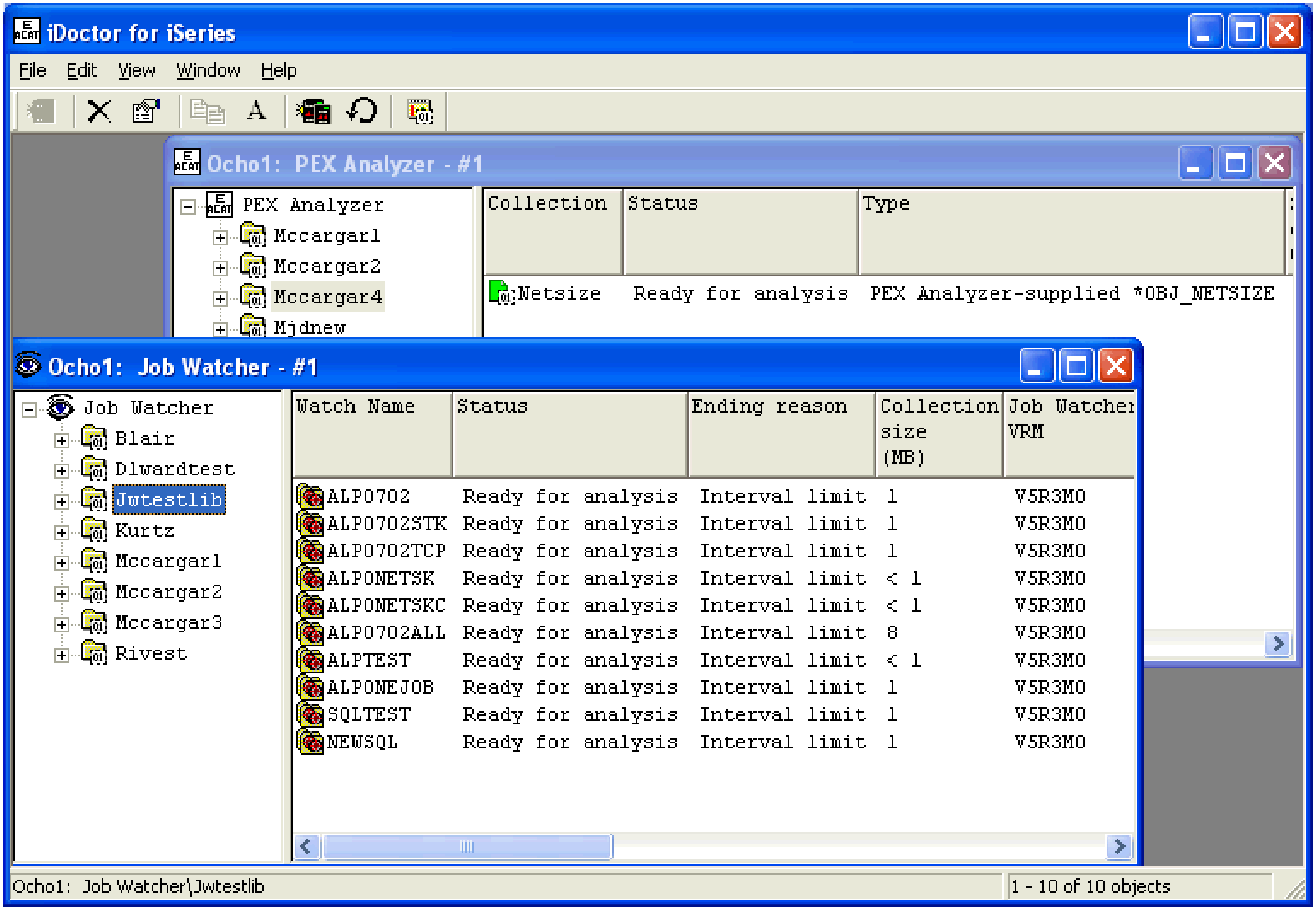
Component views are also known as the "tree/list" views and are the primary means of working with any of the iDoctor for iSeries components. You can have as many component views open within a Main Window as desired.

Component views look and feel consistently across the various components. The tree represents the hierarchy of libraries, collections, or other objects that you are dealing with on the system you are connected to. Moving down the tree lets you drill down into a specific library, job watch, etc. Your current selection in the tree is always displayed in the list portion of the tree/list.

The menus shown when right clicking a folder within a component view are also consistent. There will usually be an Explore menu letting you open the next branch of the tree or other menus that you'd expect like delete, copy, or properties.

Whenver a particular component view is active, the Data Viewer icon on the toolbar (far right) will be enabled. This lets you easily open files and run queries over data on the system you are connected to. Visit the Data Viewer documentation for more information on using the Data Viewer.

Because of the tendency to deal with large amounts of data and a desire to have the client perform optimally (reduce comm dependency, etc), refresh has been implemented in a way unlike most other applications. The refresh toolbar icon or menu will refresh only the contents of the selected tree branch. For example if a library is selected in the tree, only the contents of the library will be refreshed, not the list of libraries in the tree. Refreshing the list of libraries would require selecting the folder above the list of libraries (the component icon) before using the refresh option on the toolbar.



Collection	Status	Type
Netsize	Ready for analysis	PEX Analyzer-supplied *OBJ_NETSIZE

Watch Name	Status	Ending reason	Collection size (MB)	Job Watcher VRM
ALP0702	Ready for analysis	Interval limit	1	V5R3M0
ALP0702STK	Ready for analysis	Interval limit	1	V5R3M0
ALP0702TCP	Ready for analysis	Interval limit	1	V5R3M0
ALPONETSK	Ready for analysis	Interval limit	< 1	V5R3M0
ALPONETSKC	Ready for analysis	Interval limit	< 1	V5R3M0
ALP0702ALL	Ready for analysis	Interval limit	8	V5R3M0
ALPTEST	Ready for analysis	Interval limit	< 1	V5R3M0
ALPONEJOB	Ready for analysis	Interval limit	1	V5R3M0
SQLTEST	Ready for analysis	Interval limit	1	V5R3M0
NEWSQL	Ready for analysis	Interval limit	1	V5R3M0

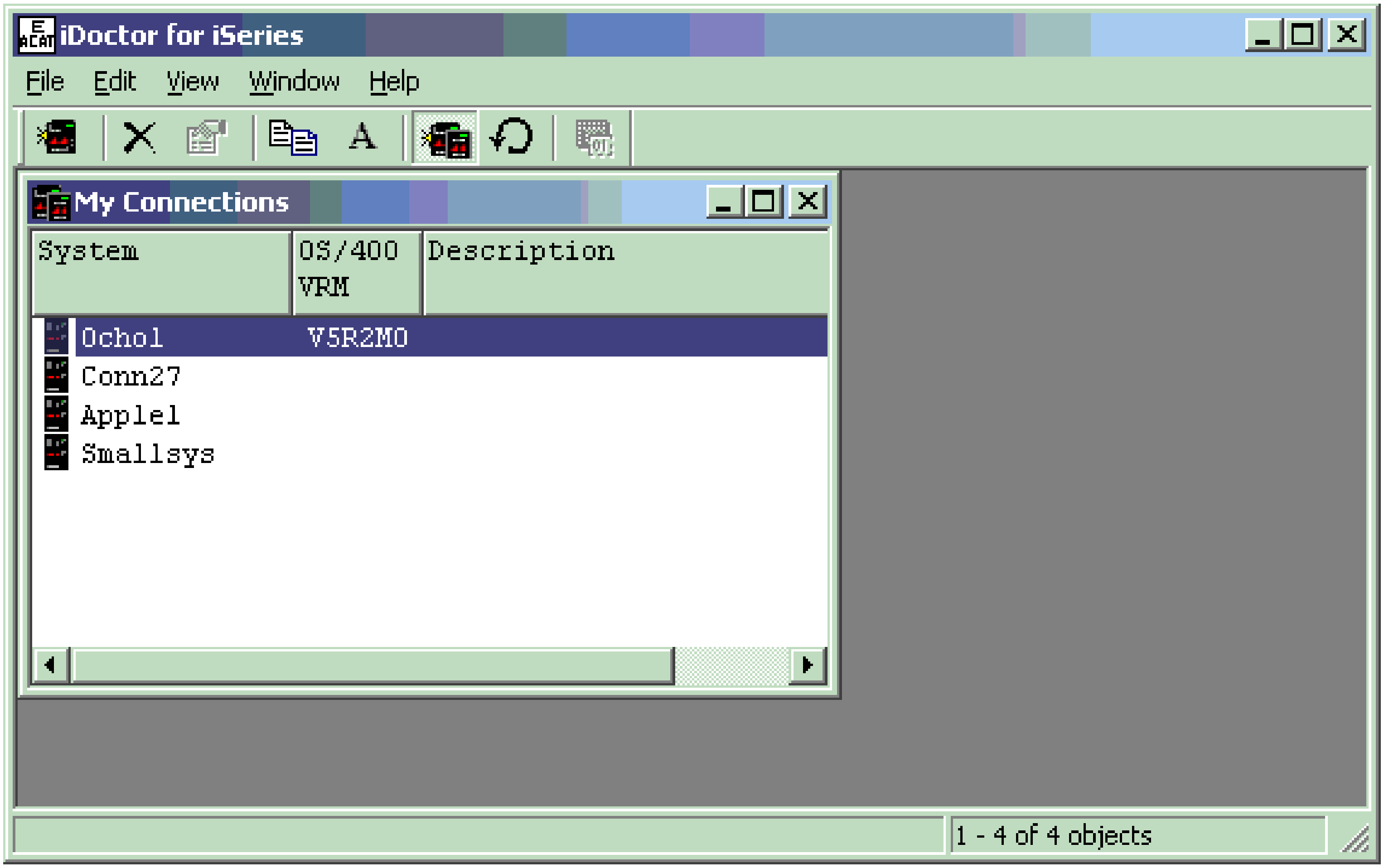
[Main Window - Showing an instance of PEX Analyzer and an instance of Job Watcher for system 'Ocho1']

Table of Contents | Previous | Next

1.1.2 My Connections View

The My Connections view allows you to work with connections to any system you have already defined on your PC through iSeries Navigator or via iDoctor for iSeries. You can easily add or remove connections to other systems through this view. The primary purpose of this view is to provide a quick and easy way to launch the iDoctor for iSeries components for any system desired.

The list of connections shown is for the currently active environment. You can change the currently active environment by right-clicking on the list and choosing the Change Environment... menu.

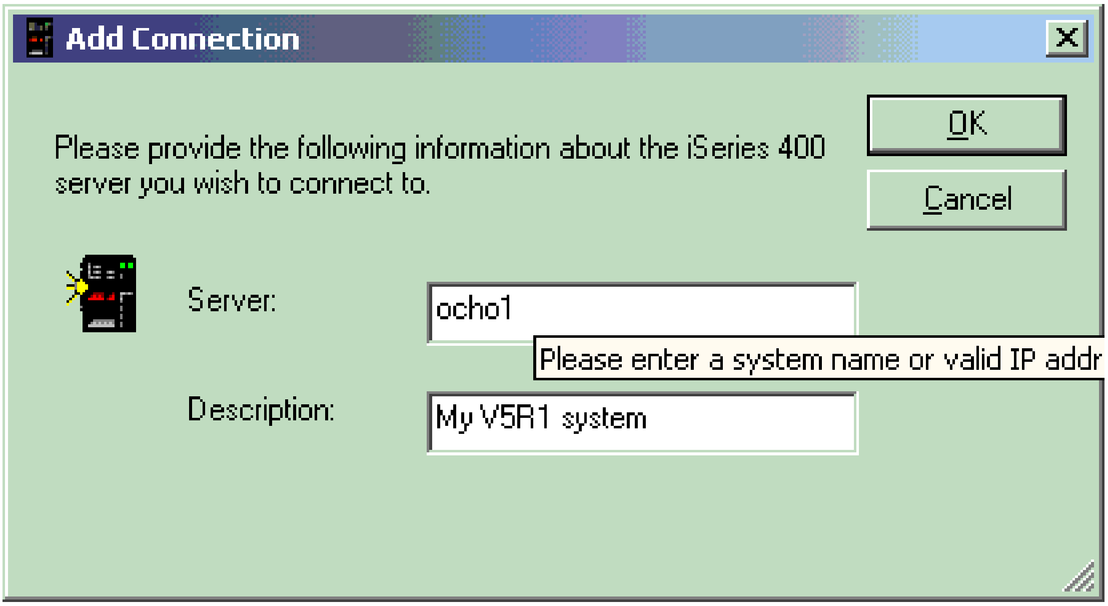


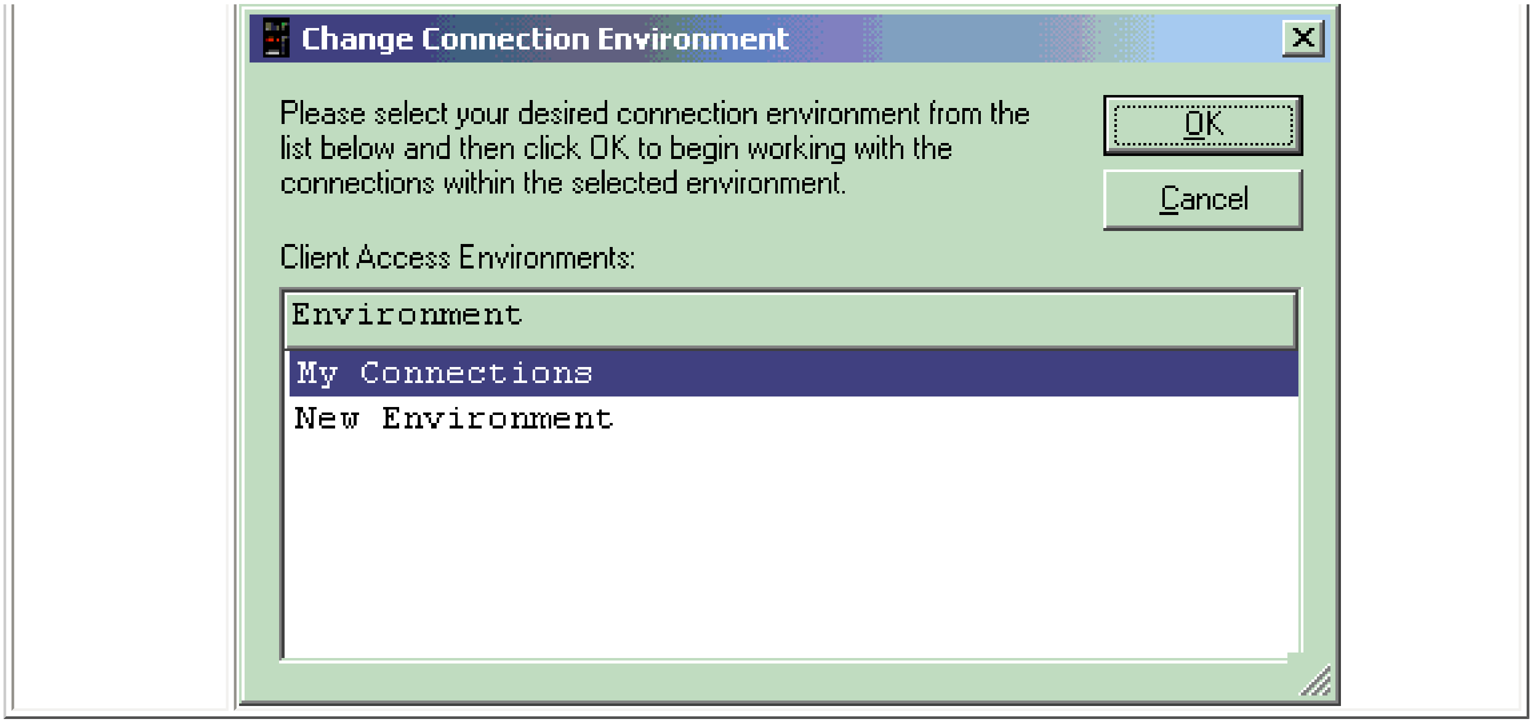
System	OS/400 VRM	Description
Ochol	V5R2M0	
Conn27		
Applel		
Smallsys		

[The Main Window displaying the My Connections View]

The following actions may be taken in the My Connections View by right-clicking on one of the systems in the list:

Popup Menu	Description

Component name	This will launch the appropriate component. If you need to provide your user profile and password to that system or provide an access code you will be prompted.
Add Connection	Use this menu to add a connection to the My Connections View. You will see the following dialog:  Add Connection Please provide the following information about the iSeries 400 server you wish to connect to. Server: ocho1 Please enter a system name or valid IP addr Description: My V5R1 system Please provide the system name or IP address and an optional description and click OK to register the system on your PC and add it to the connections list. Any systems you add will also be shown in iSeries Navigator.
Delete	This will delete a connection from the My Connections View. Any system connections you delete from this list will also be deleted in iSeries Navigator.
Change Environment...	This menu lets you change the currently active environment as desired. Each environment represents a list of connections. The environments may be created/imported through iSeries Navigator. Any connections you add or remove from the environment within iDoctor for iSeries take effect within iSeries Navigator.



Please select your desired connection environment from the list below and then click OK to begin working with the connections within the selected environment.

Client Access Environments:

Environment
My Connections
New Environment

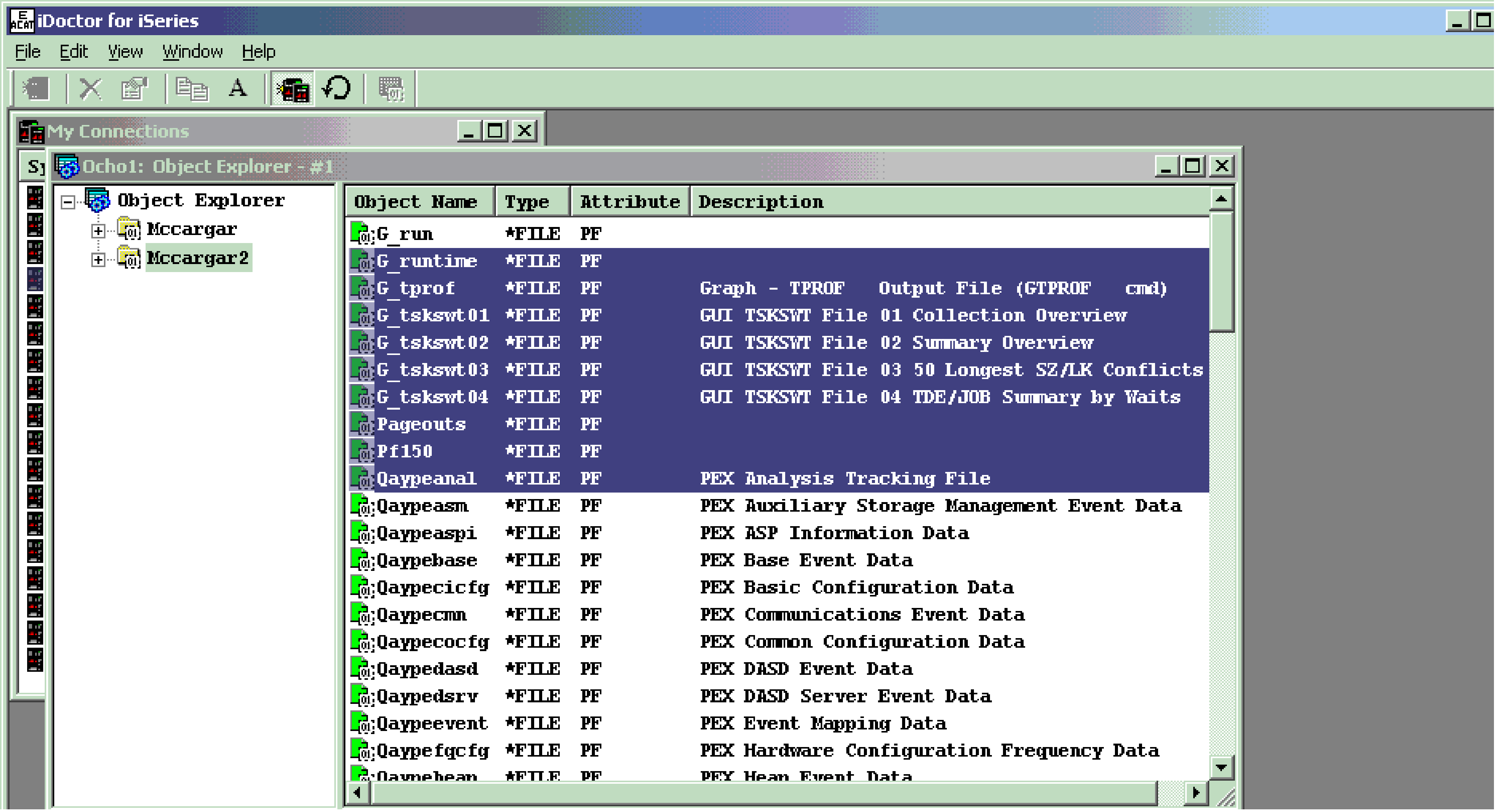
Table of Contents | Previous | Next

1.1.3 Remote Command Status View

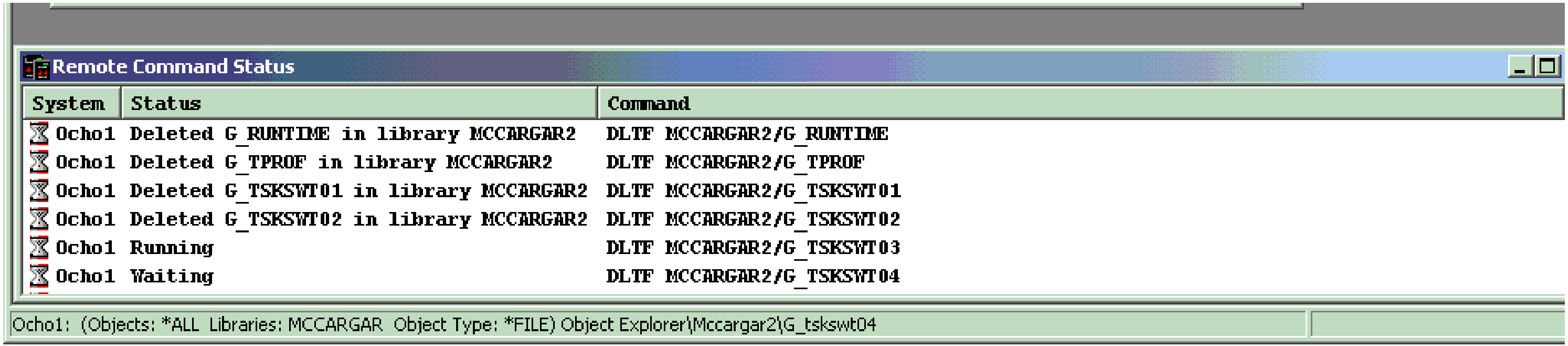
The Remote Command Status view shows you the status of certain remote commands being executed on a system. This allows you to perform lengthy operations like copy objects, or delete files without tying up the GUI (i.e. you can do other things like view a report while the remote commands are executing).

Depending on the function/feature being used (like copying objects in Object Explorer) you will see one or more commands in the remote command status view. As each command completes you will immediately see its result or error message in the view.

You can also close this window and reopen it later while commands are being executed to periodically check the status of the commands issued.



Object Name	Type	Attribute	Description
G_run	*FILE	PF	
G_runtime	*FILE	PF	
G_tprof	*FILE	PF	Graph - TPROF Output File (GTPROF cmd)
G_tskswt01	*FILE	PF	GUI TSKSWT File 01 Collection Overview
G_tskswt02	*FILE	PF	GUI TSKSWT File 02 Summary Overview
G_tskswt03	*FILE	PF	GUI TSKSWT File 03 50 Longest SZ/LK Conflicts
G_tskswt04	*FILE	PF	GUI TSKSWT File 04 TDE/JOB Summary by Waits
Pageouts	*FILE	PF	
Pf150	*FILE	PF	
Qaypeanal	*FILE	PF	PEX Analysis Tracking File
Qaypeasm	*FILE	PF	PEX Auxiliary Storage Management Event Data
Qaypeaspi	*FILE	PF	PEX ASP Information Data
Qaypebase	*FILE	PF	PEX Base Event Data
Qaypecicfg	*FILE	PF	PEX Basic Configuration Data
Qaypecmn	*FILE	PF	PEX Communications Event Data
Qaypecocfg	*FILE	PF	PEX Common Configuration Data
Qaypedasd	*FILE	PF	PEX DASD Event Data
Qaypedsrv	*FILE	PF	PEX DASD Server Event Data
Qaypeevent	*FILE	PF	PEX Event Mapping Data
Qaypefqcfg	*FILE	PF	PEX Hardware Configuration Frequency Data
Qaypeheap	*FILE	PF	PEX Heap Event Data



System	Status	Command
Ocho1	Deleted G_RUNTIME in library MCCARGAR2	DLTF MCCARGAR2/G_RUNTIME
Ocho1	Deleted G_TPROF in library MCCARGAR2	DLTF MCCARGAR2/G_TPROF
Ocho1	Deleted G_TSKSWT01 in library MCCARGAR2	DLTF MCCARGAR2/G_TSKSWT01
Ocho1	Deleted G_TSKSWT02 in library MCCARGAR2	DLTF MCCARGAR2/G_TSKSWT02
Ocho1	Running	DLTF MCCARGAR2/G_TSKSWT03
Ocho1	Waiting	DLTF MCCARGAR2/G_TSKSWT04

Ocho1: (Objects: *ALL Libraries: MCCARGAR Object Type: *FILE) Object Explorer\Mccargar2\G_tskswt04

[The Main Window displaying an Object Explorer View and the Remote Command Status View]

The following actions may be taken in the Remote Command Status View by right-clicking on one (or more) entries in the view:

Popup Menu	Description
Remove Completed	Use this menu to remove all remote command entries that have completed.
Remove Selected	Use this menu to remove all selected remote command entries from the view.
Remove All	Use this menu to remove all remote command entries from the view.

Table of Contents Previous Next

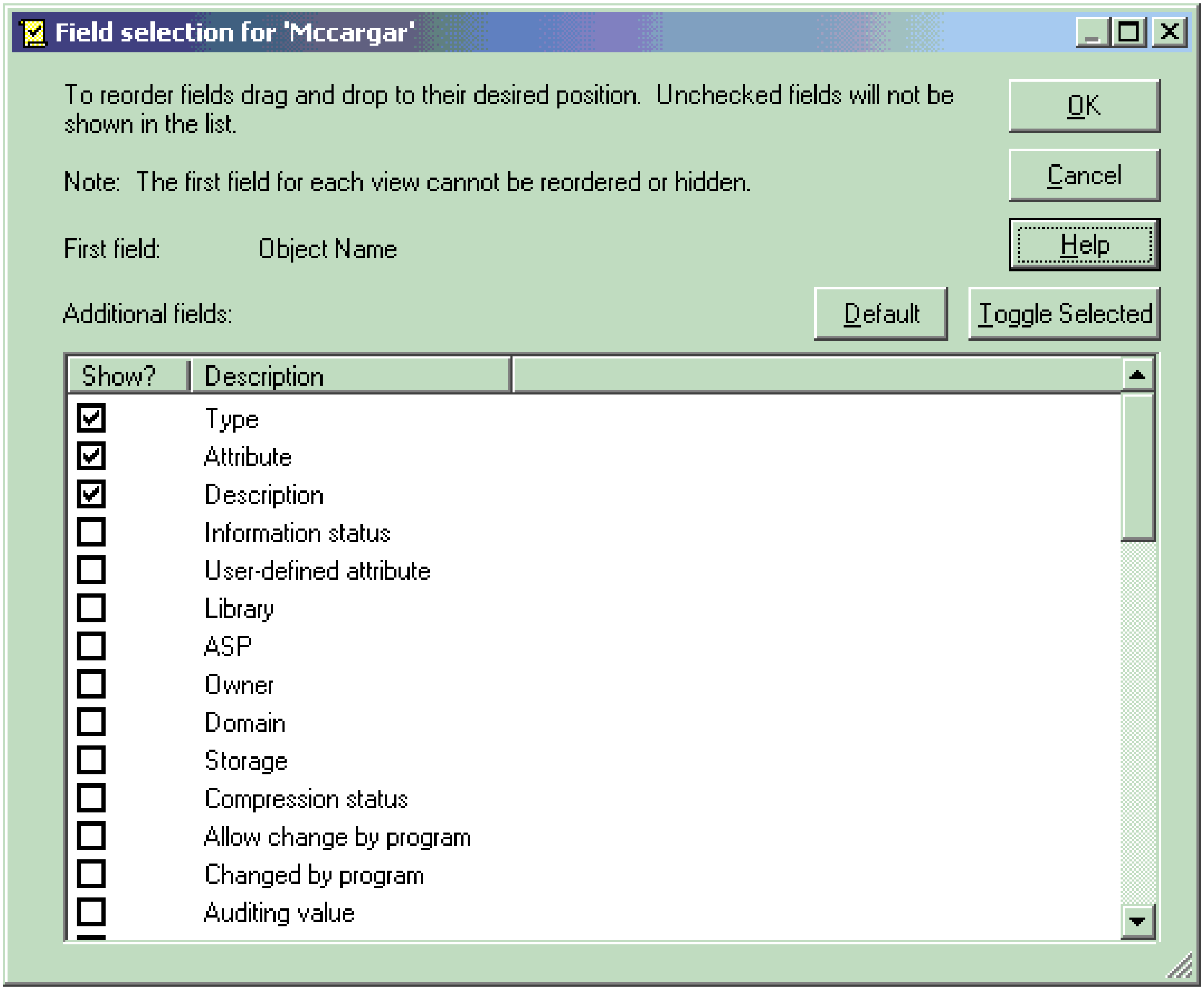
1.2 Field Selection Window

The Field Selection Window is a generic way to work with the fields shown in the list portion of a tree/list view. This window is available via the 'Select fields...' menu from any objects that has field selection enabled. Not all folders in the tree have field selection enabled, only those that have a large number of available fields to display.

The field selection window lets you drag fields into the desired positions. The first field in the list portion of a tree/list view is fixed, but the rest of the fields can be reordered or hidden as desired. An example of a Field Selection Window is shown below.

Any changes you make are saved to your system registry and reused the next time you open the view you are working with. To restore to the iDoctor-ship default ordering click the "Default" button. The "Toggle Selected" button is a fast way to toggle the show checkbox for several selected fields in the list at once. To select multiples hold down the ctrl or shift key while clicking your mouse on entries in the list.

All components make use of this interface to help make using iDoctor more convenient.



Field selection for 'Mccargar'

To reorder fields drag and drop to their desired position. Unchecked fields will not be shown in the list.

Note: The first field for each view cannot be reordered or hidden.

First field: Object Name

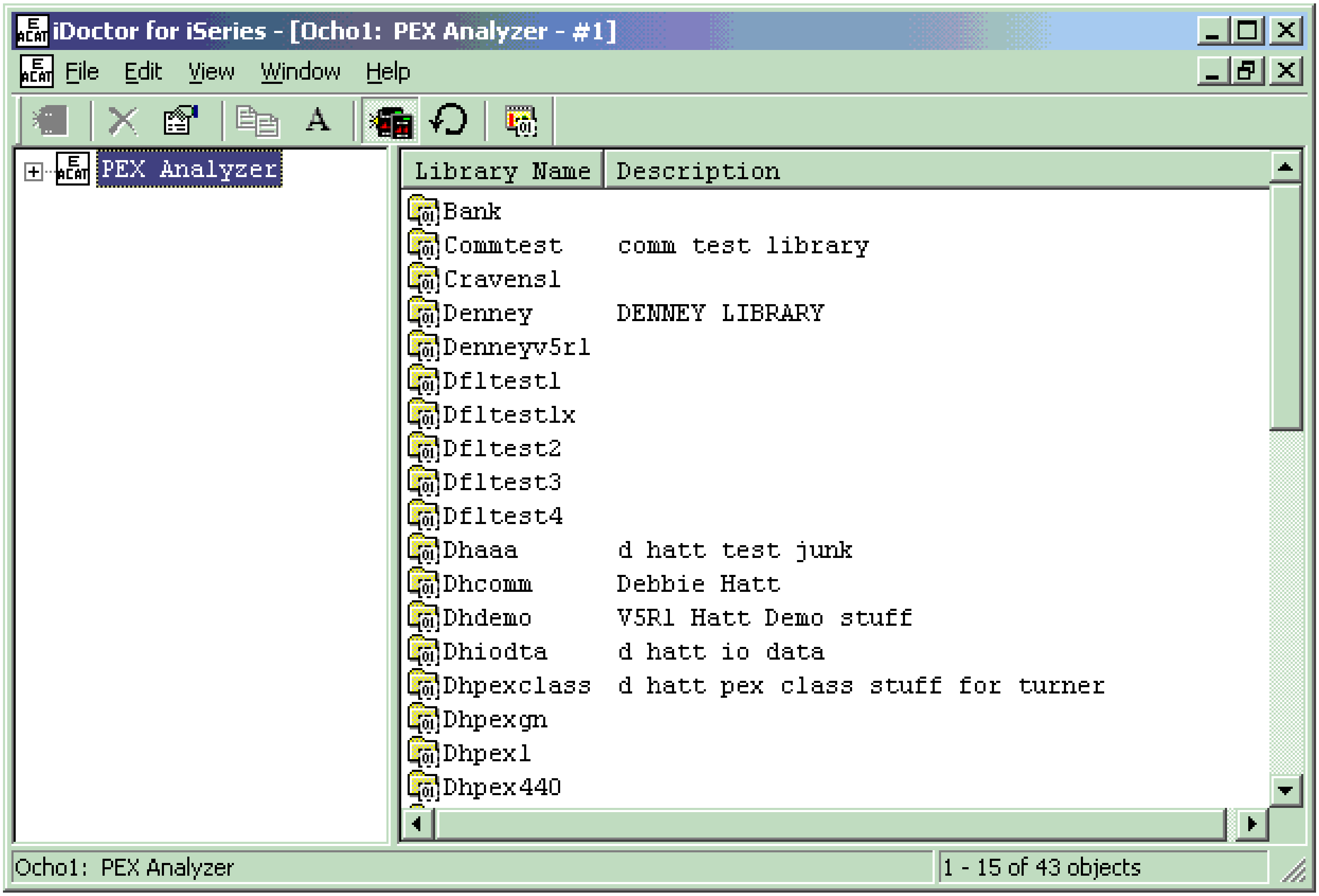
Additional fields:

Show?	Description
☑	Type
☑	Attribute
☑	Description
□	Information status
□	User-defined attribute
□	Library
□	ASP
□	Owner
□	Domain
□	Storage
□	Compression status
□	Allow change by program
□	Changed by program
□	Auditing value

[Field Selection Window - For displaying objects in a library within Object Explorer]

1.3 Libraries

A common characteristic of all component views in iDoctor for iSeries is that they all initially display a list of libraries on the system. The libraries shown depend on the particular component. PEX Analyzer shows libraries with PEX data, Job Watcher shows libraries with Job Watcher data, and Object Explorer shows all libraries on the system matching the current object filter criteria.



Library Name	Description
Bank	
Commtest	comm test library
Cravensl	
Denney	DENNEY LIBRARY
Denneyv5r1	
Dfltest1	
Dfltest1x	
Dfltest2	
Dfltest3	
Dfltest4	
Dhaaa	d hatt test junk
Dhcomm	Debbie Hatt
Dhdemo	V5R1 Hatt Demo stuff
Dhiodta	d hatt io data
Dhpexclass	d hatt pex class stuff for turner
Dhpexgm	
Dhpex1	
Dhpex440	

[A list of libraries in PEX Analyzer]

All libraries have detailed properties and a set of menu options available. This chapter will discuss each of the library property pages in iDoctor for iSeries as well as all the menu options for a library.

Table of Contents | Previous | Next

1.3.1 Menu Options

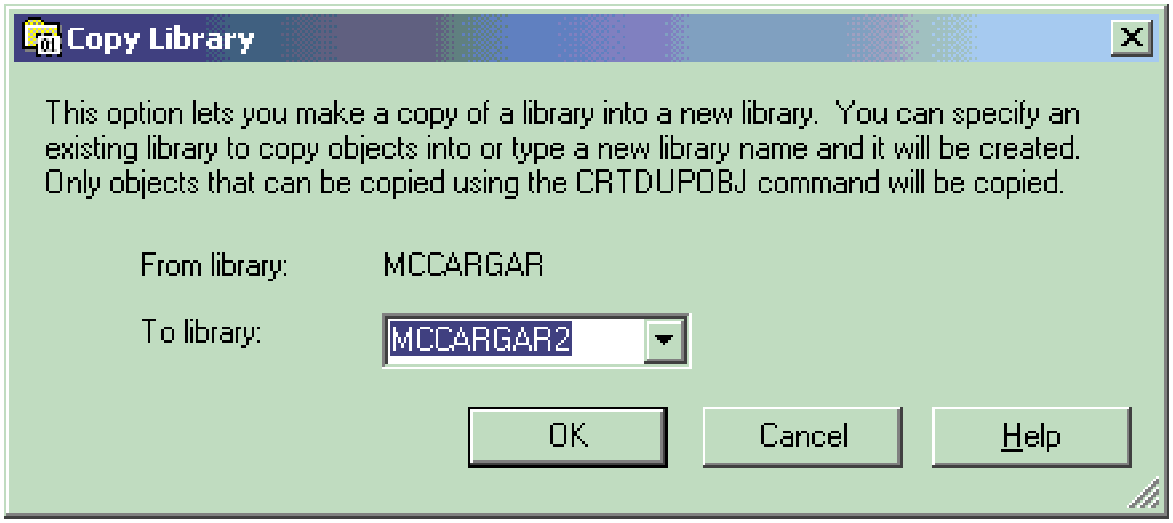
A library folder in iDoctor for iSeries has the following menu options available by right-clicking on the library icon:

Menu	Description
Explore	Show the contents of the library (the contents depends on the component).
Select fields...	Displays the Field Selection Window which allows you to reorder the fields that are displayed as the contents of the library. For example in Object Explorer, libraries contain objects and this menu lets you configure which fields for the objects are visible within the library.
Copy...	Allows you to copy the library's contents into a new library or into an existing one.
Save...	This option lets you save the library's contents into a save file.
Clear	This option clears a library (deletes all objects in the library).
Delete	Deletes the library.
Rename	Renames the library.
Properties	Displays the property pages for the library.

Depending on the component, a library folder will have other menu options available.

1.3.2 Copying

A library may have its contents copied into a new library or into an existing library by using the Copy... menu available by right-clicking on a library within iDoctor for iSeries. This option is an interface over the CPYLIB command. The progress of the library being copied may be viewed using the Remote Command Status View



This option lets you make a copy of a library into a new library. You can specify an existing library to copy objects into or type a new library name and it will be created. Only objects that can be copied using the CRTDUPOBJ command will be copied.

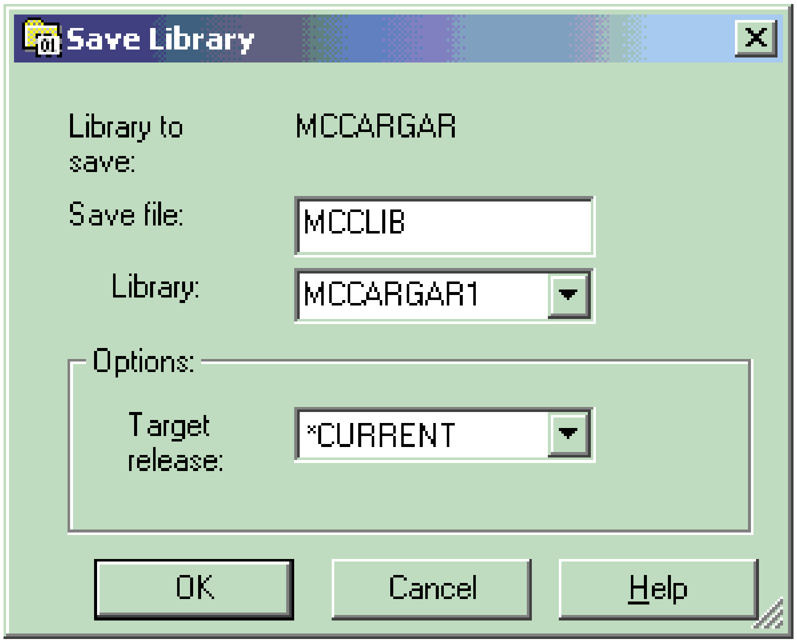
From library: MCCARGAR

To library: MCCARGAR2

Field	Description
From library	Displays the name of the library to be copied.
To library	The name of the library that will receive the contents of the from library. By clicking the down arrow you can choose from a list of all libraries on the system.

1.3.3 Saving

A library's contents can be saved using the Save... menu available by right-clicking on a library within iDoctor for iSeries. This option is a interface over the SAVLIB command. This interface is restricted to saving the library to a save file and is missing some of the advanced options on the command. The progress of the library being saved may be viewed using the Remote Command Status View



Field	Description
Library to save	The name of the library to be saved.
Save file/library	The name of the save file and library to save the contents of the library into. If the save file doesn't exist it is created. If the save file does exist, you will be asked for confirmation before continuing.
Target release	Specifies the release of the operating system on which you intend to restore and use the object.

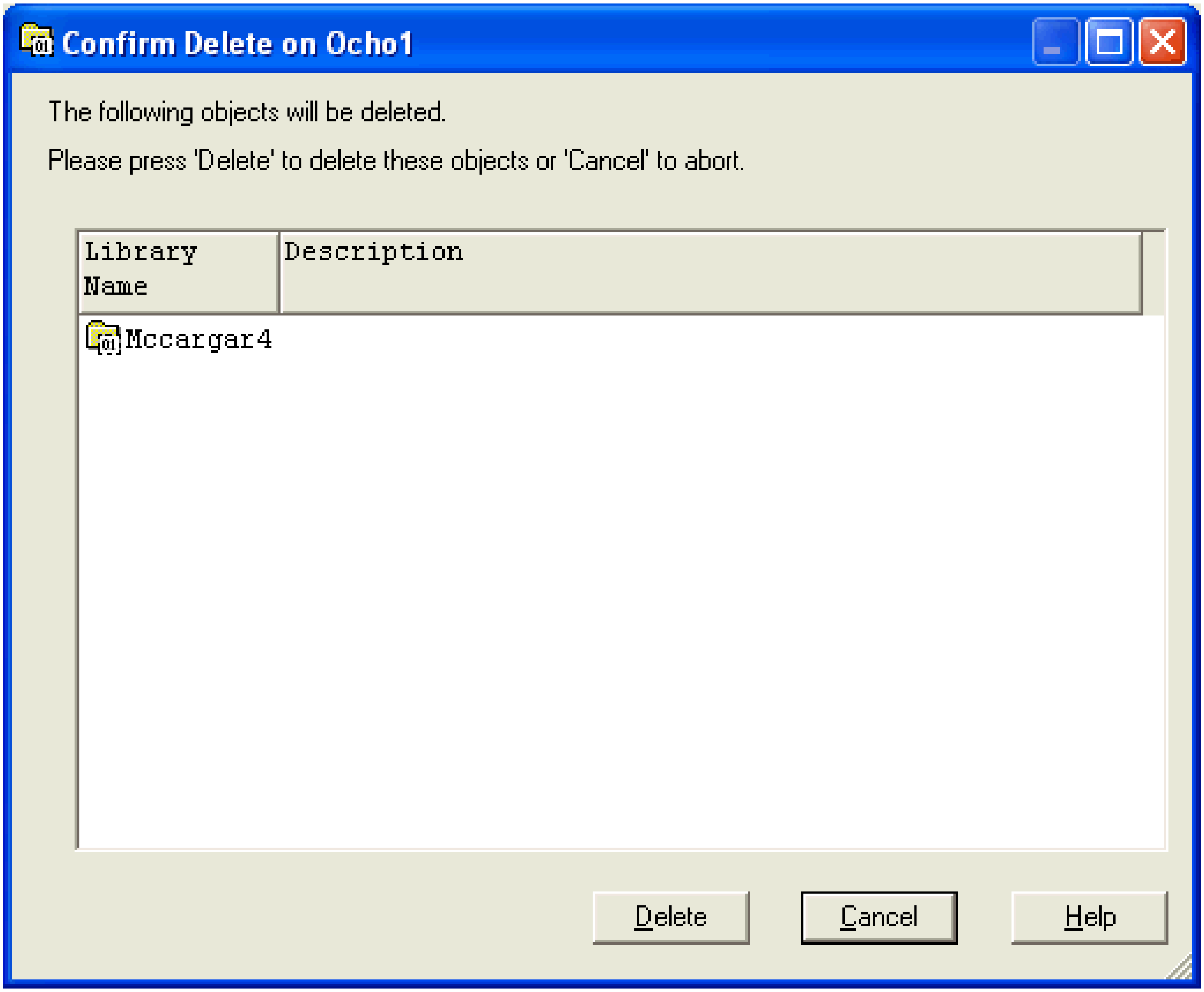
Table of Contents | Previous | Next

1.3.4 Clearing

A library's contents may be cleared using the Clear... menu available by right-clicking on a library within iDoctor for iSeries. This option will initiate (after confirmation) a CLRLIB command over the specified library. The progress of the library being cleared may be viewed using the Remote Command Status View

1.3.5 Deleting

A library's may be deleted using the Delete... menu available by right-clicking on a library within iDoctor for iSeries. This option is a interface over the DLTLIB command. The progress of the library being deleted may be viewed using the Remote Command Status View


Confirm Delete on Ocho1

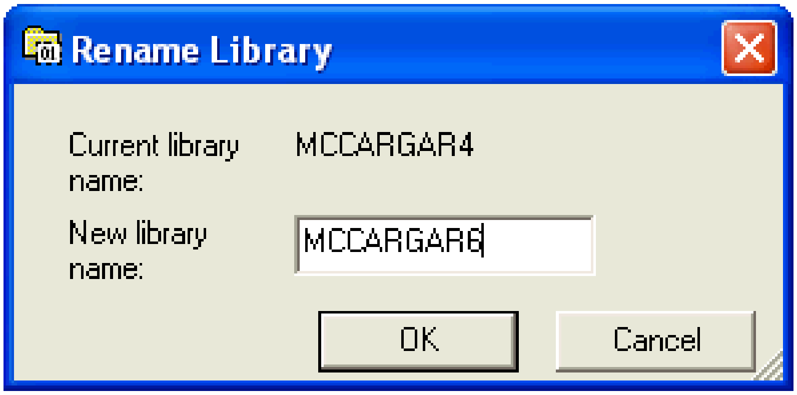
The following objects will be deleted.

Please press 'Delete' to delete these objects or 'Cancel' to abort.

Library Name	Description
Mccargar4	

1.3.6 Renaming

A library may be renamed using the Rename... menu available by right-clicking on a library within iDoctor for iSeries. This option is a interface over the RNMOBJ command.



Current library name: MCCARGAR4

New library name: MCCARGAR6

Field	Description
Current library name	The name of the library to be renamed.
New library name	The name to replace the current library name.

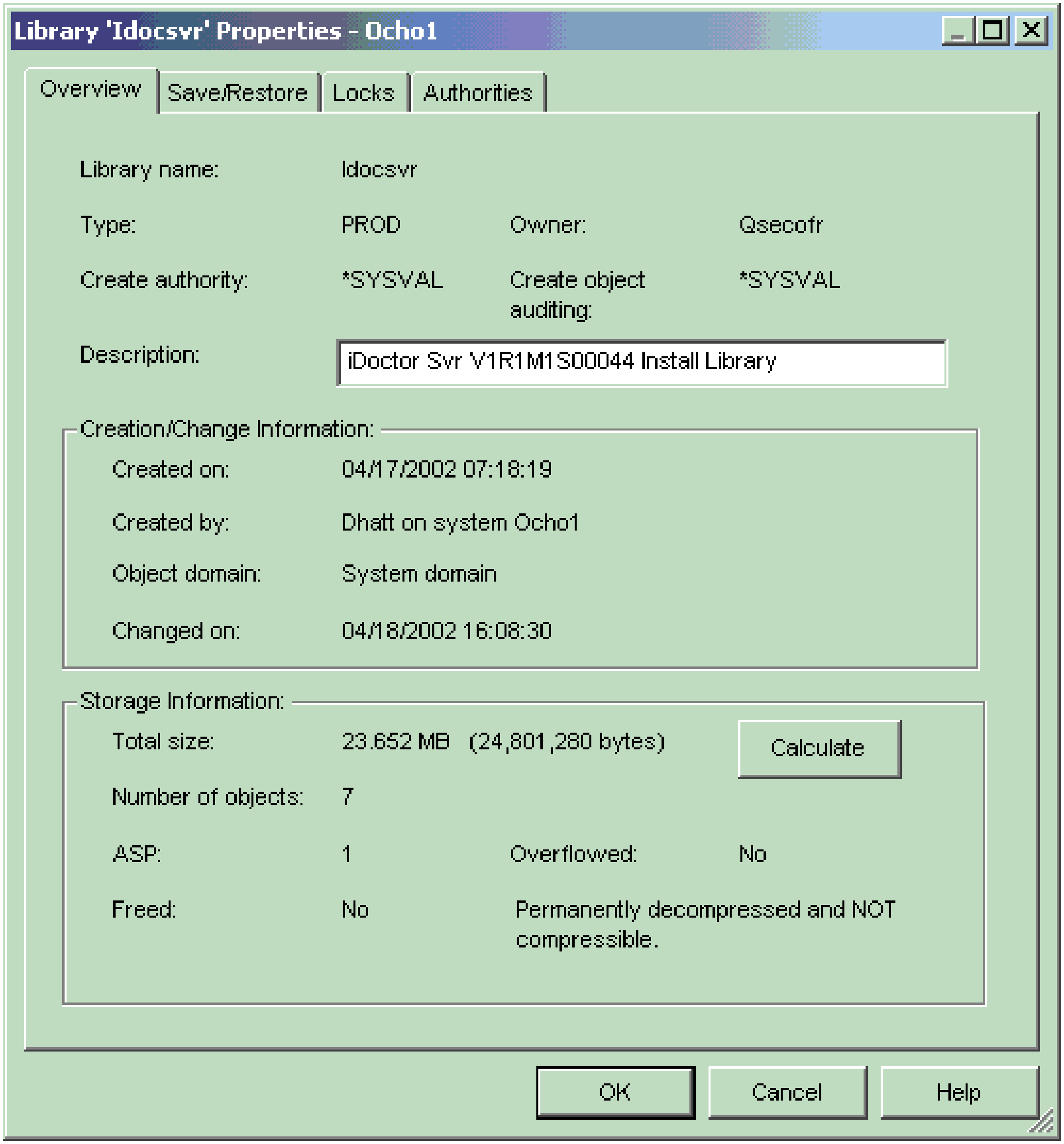
Table of Contents Previous Next

1.3.7 Properties

The library property pages are accessible by right-clicking on a library and choosing the Properties menu. The next sections discusses all of the library properties pages.

1.3.7.1 Overview

The overview property page for libraries displays basic information about the library, including the total size of all objects in the library.



Library name: Idocsvr

Type: PROD Owner: Qsecofr

Create authority: *SYSVAL Create object auditing: *SYSVAL

Description: iDoctor Svr V1R1M1S00044 Install Library

Creation/Change Information:

Created on: 04/17/2002 07:18:19

Created by: Dhatt on system Ocho1

Object domain: System domain

Changed on: 04/18/2002 16:08:30

Storage Information:

Total size: 23.652 MB (24,801,280 bytes)

Number of objects: 7

ASP: 1 Overflowed: No

Freed: No Permanently decompressed and NOT compressible.

The following information is listed on this page:

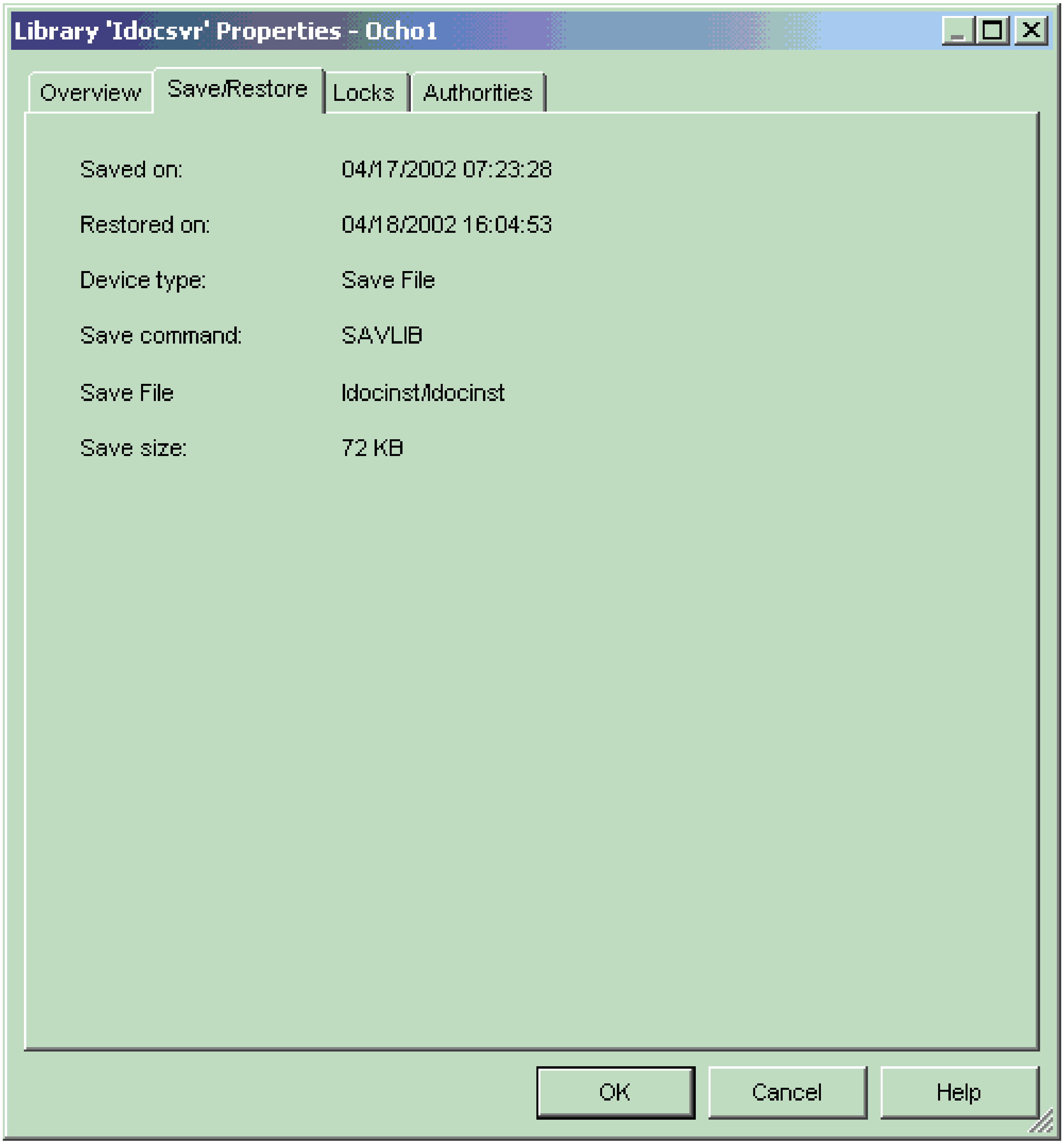
Field	Field Description
Library Name	Name of the library.
Type	Indicates the libraries type. **PROD** The library is a production library. Database files in production libraries cannot be opened for updating if a user is in debug mode and requested that production libraries be protected. **TEST** The library is a test library. All objects in a test library can be updated during test. See the STRDBG command for more details.
Owner	The name of the user profile which owns the library.
Create Authority	The default public authority used when an object is created into a library. This authority is given to the following users: - Users who do not have specific authority to the object. - Users who are not on the authorization list. - Users whose user group has no specific authority to the object. The valid values are: ***ALL** The user can perform all authorized operations on an object created in this library. ***CHANGE** The user can read the object description and has read, add, update, and delete authority to an object created in this library. ***EXCLUDE** The user is prevented from accessing an object created in this library. ***SYSVAL** The default authority for an object created in this library is determined by the value specified by the QCRTAUT system value. ***USE** The user can read the object and its description but cannot change them for an object created in this library. **Authorization list name** The name of the authorization list that secures an object created in this library. The default

	public authority is taken from the authorization list, and the public authority for the object is specified as *AUTL.
Create Object Auditing	The auditing value for objects created in this library. The valid values are: ***ALL** All change or read access to the object is logged. ***CHANGE** All change access to the object by all users is logged. ***NONE** Use or change access to the object is not logged (no audit entry is sent to the security journal). ***SYSVAL** The value specified in the system value QCRTOBJAUD is used. ***USRPRF** The user profile of the user who accesses the object is used to determine if an audit record is sent for this access. The OBJAUD parameter of the Change User Auditing (CHGUSRAUD) command is used to turn auditing on for a specific user.
Description	Library description. You can change this value if you want.
Created On	The date and time the library was created.
Created By	The name of the user who created the library and the system it was created on.
Object Domain	The domain that contains the object. The possible values are user domain or system domain.
Changed On	The date and time the library was changed.
Total Size	Total size of all objects in the library including the library itself. Click the calculate button to compute this value. Note: This calculation can take a long time (minutes) depending on the number of objects and members in the library.
Object Count	Total number of objects in the library.
ASP	Auxillary Storage Pool: A number indicating the identifier of the auxiliary storage pool from which storage space for the library was allocated.
Overflowed	Indicates if the object has overflowed the auxiliary storage pool it resides in.
Freed	Indicates the storage status of the object (Yes/No). If the storage status is freed, then the object is suspended, otherwise the object is not suspended.
Object Compression	Indicates the compression status of the object.

Table of Contents Previous Next

1.3.7.2 Save/Restore

The save/restore property page displays information about how and when the library was last saved or restored.



Saved on:	04/17/2002 07:23:28
Restored on:	04/18/2002 16:04:53
Device type:	Save File
Save command:	SAVLIB
Save File	Idocinst/Idocinst
Save size:	72 KB

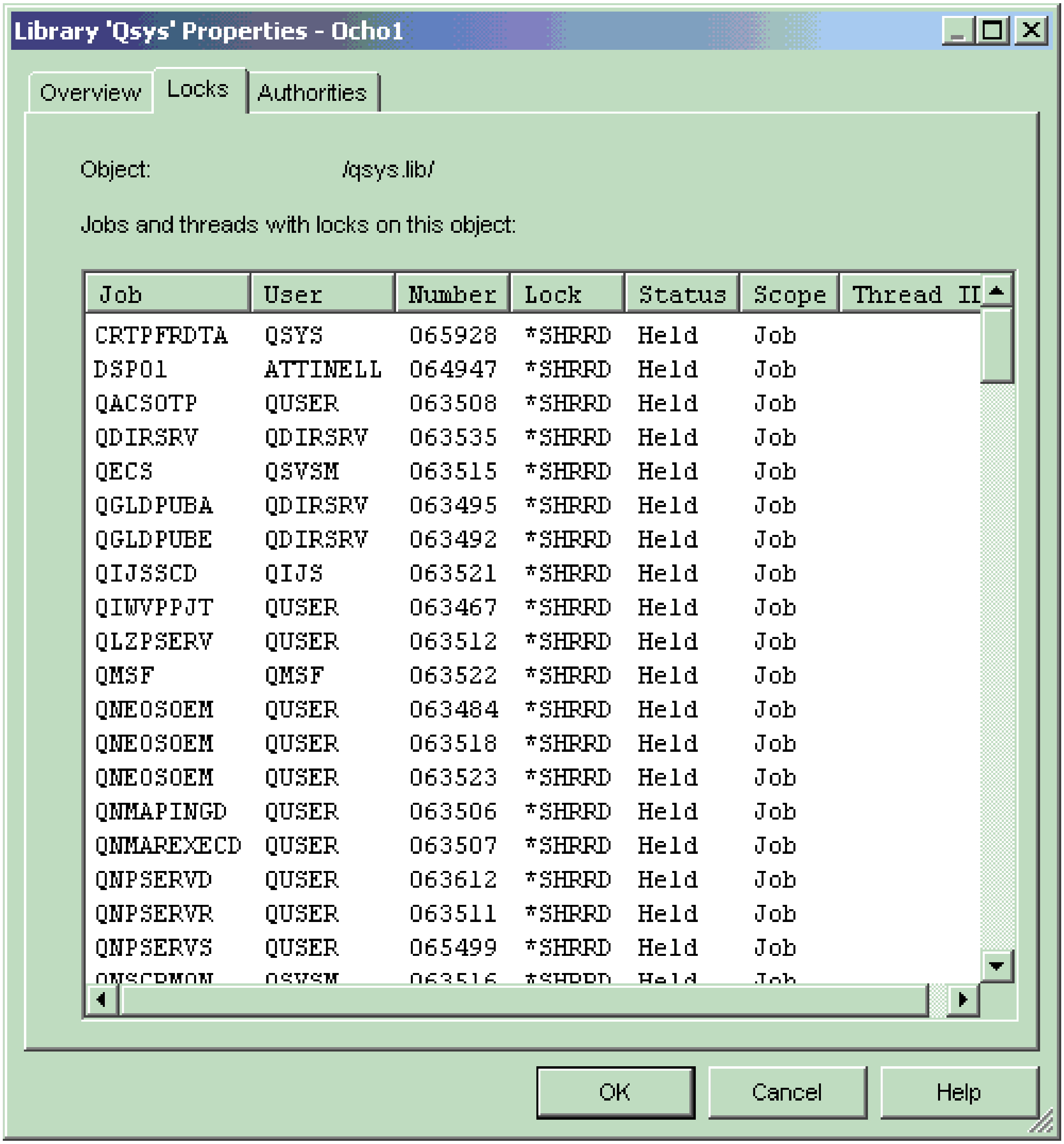
The following information is listed on this page:

Field	**Field Description**
Saved On	The date and time the library was last saved.
Restored On	The date and time the library was last restored.
Device Type	The type of the device to which the library was last saved. Valid values are: **Blank** The library was not saved. **Diskette** The library was saved to diskette. **Optical** The library was saved to optical. **Save file** The library was saved to a save file. **Tape** The library was saved to tape.
Save Command	The command used to save the object.
Label	The file label used when the object was saved. This value is not shown if the library was not saved to tape, diskette, or optical. The value of this field corresponds to the value specified for the LABEL or OPTFILE parameter on the command used to save the object.
Save File	Displays the library and name of the save file.
Volume ID	The tape, diskette, or optical volumes that are used for saving the library.
Save Size	Displays the size of the save file.
Save Sequence Number	The tape sequence number assigned when the library was saved on tape. If the library was not saved to tape, this value is not displayed.

Table of Contents | Previous | Next

1.3.7.3 Locks

The locks property page for libraries provides an interface similar to the Work Object Lock (WRKOBJLCK) command. This page will tell you which jobs (if any) have a lock on the library.



Object: /qsys.lib/

Jobs and threads with locks on this object:

Job	User	Number	Lock	Status	Scope	Thread II
CRTPFRDTA	QSYS	065928	*SHRRD	Held	Job	
DSP01	ATTINELL	064947	*SHRRD	Held	Job	
QACSOTP	QUSER	063508	*SHRRD	Held	Job	
QDIRSRV	QDIRSRV	063535	*SHRRD	Held	Job	
QECS	QSVSM	063515	*SHRRD	Held	Job	
QGLDPUBA	QDIRSRV	063495	*SHRRD	Held	Job	
QGLDPUBE	QDIRSRV	063492	*SHRRD	Held	Job	
QIJSSCD	QIJS	063521	*SHRRD	Held	Job	
QIWVPPJT	QUSER	063467	*SHRRD	Held	Job	
QLZPSERV	QUSER	063512	*SHRRD	Held	Job	
QMSF	QMSF	063522	*SHRRD	Held	Job	
QNEOSOEM	QUSER	063484	*SHRRD	Held	Job	
QNEOSOEM	QUSER	063518	*SHRRD	Held	Job	
QNEOSOEM	QUSER	063523	*SHRRD	Held	Job	
QNMAPINGD	QUSER	063506	*SHRRD	Held	Job	
QNMAREXECD	QUSER	063507	*SHRRD	Held	Job	
QNPSERVD	QUSER	063612	*SHRRD	Held	Job	
QNPSERVR	QUSER	063511	*SHRRD	Held	Job	
QNPSERVS	QUSER	065499	*SHRRD	Held	Job	
QNSCPMON	QSVSM	063516	*SHRRD	Held	Job	

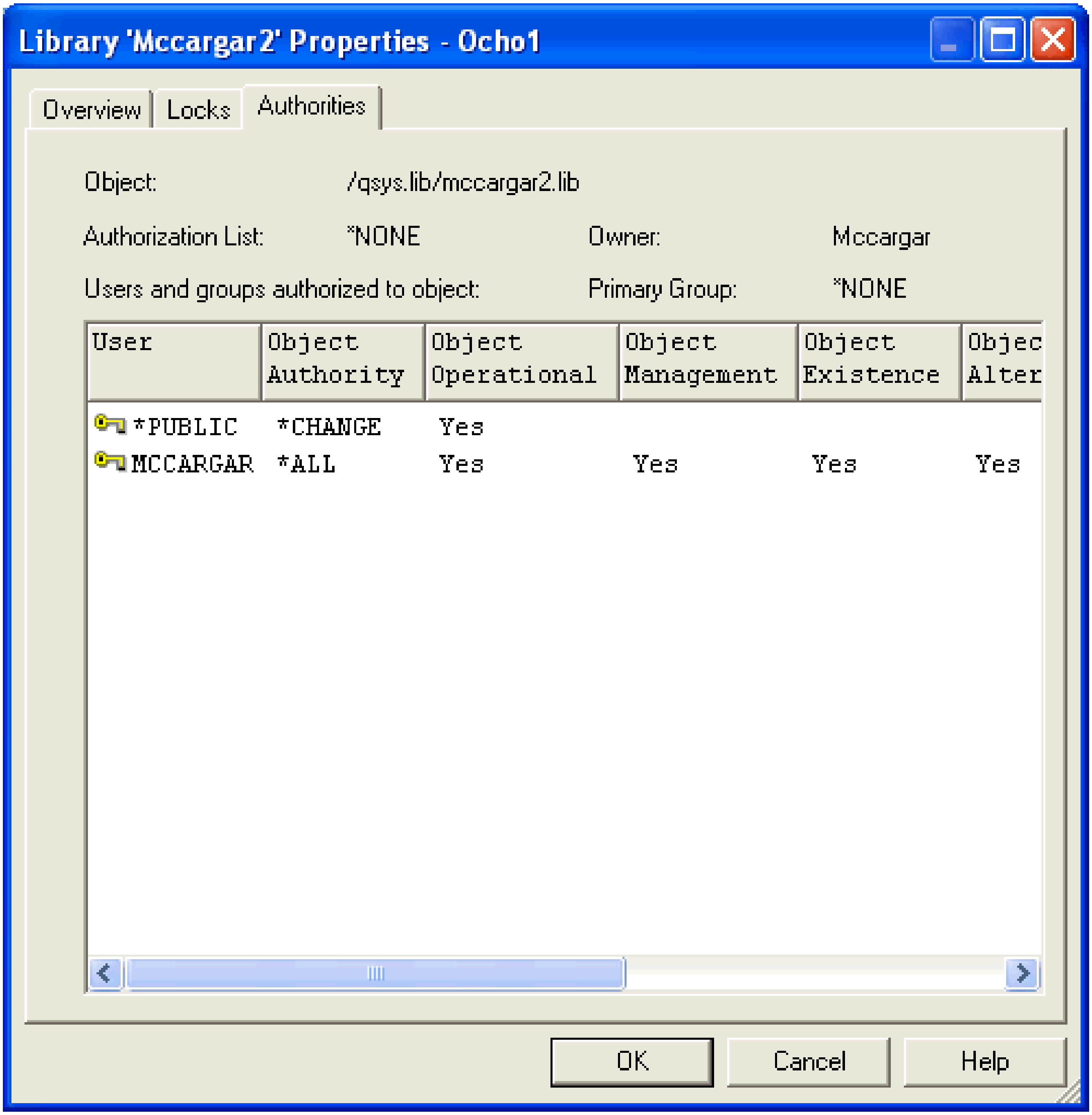
The following information is shown for each job in the list.

Field	**Field Description**
Job	The simple job name of the job that issued the lock request.
User	The user name of the job that issued the lock request.
Number	The number of the job that issued the lock request.
Lock	The lock condition for the request. The possible values are: ***SHRRD** Lock shared for read. ***SHRUPD** Lock shared for update. ***SHRNUP** Lock shared no update. ***EXCLRD** Lock exclusive allow read. ***EXCL** Lock exclusive no read. ***NONE** Lock entry has a null value and is used to select display of lower-level locks.
Status	The status of the lock. The possible values are: **HELD** The lock is currently held by the job. **WAIT** The job is waiting for the lock. **REQ** The job has a lock request outstanding for the object.
Scope	Specifies whether the lock is scoped to the job or scoped to the thread.
Thread ID	Specifies the thread that is associated with the lock. If a held lock is job scoped, the field is blank. If a held lock is thread scoped, the field contains the identifier for the thread holding the lock. If the lock is requested, but not yet available, this field contains the identifier of the thread requesting the lock.

Table of Contents Previous Next

1.3.7.4 Authorities

The Authorities property page shows a list of users that have authority to the library and the users' authorities. This interface is similar to the DSPOBJAUT command.



Object: /qsys.lib/mccargar2.lib

Authorization List: *NONE Owner: Mccargar

Users and groups authorized to object: Primary Group: *NONE

User	Object Authority	Object Operational	Object Management	Object Existence	Objec Alter
*PUBLIC	*CHANGE	Yes			
MCCARGAR	*ALL	Yes	Yes	Yes	Yes

The following information is shown on this page:

Field	Field Description
Object	The name of the object for which information is being displayed.
Authorization List	The name of the authorization list that is used to secure the named object. The value, *NONE, indicates that no authorization list is used in determining authority to the object.
Owner	The name of the user profile which owns the library.
Primary Group	The name of the user profile that is the primary group for the library. The primary group can be changed using the Change Object Primary Group (CHGOBJPGP) command.
User List	Displays each user authorized to the library and their detailed authorities to it.

The following information is shown for each user in the list.

Field	Field Description
User	The names of users who are authorized to use the object. The value *PUBLIC is used to indicate the authorities of users who are not specifically named and are not in the object's authorization list.
Group	A group from which the user receives authority.
Obj Authority	The user's authority to the object. This field contains one of the following values: ***ALL** The user has all object (operational, management, existence, alter, and reference) and data (read, add, update, delete, and execute) authorities to the object. ***CHANGE** The user has object operational and all data authorities to the object. ***USE** The user has object operational and data read and execute authorities to the object. ***EXCLUDE** The user has none of the object or data authorities to the object, or authorization list management authority to the authorization list. ***AUTL** The public authority for the object comes from the public authority on the authorization list securing the object. This value can only be returned if there is an authorization list securing the object and the authorized user is *PUBLIC.

	USER DEF The user has some combination of object and data authorities that do not relate to a special value. The individual authorities for the user should be checked to determine what authority the user has to the object.
Obj Opr	Object operational authority provides authority to look at the object's attributes and to use the object as specified by the data authorities that the user has to the object.
Obj Mgmt	Object management authority provides authority to specify security, to move or rename the object, and to add members if the object is a database file.
Obj Exist	Object existence authority provides authority to control the object's existence and ownership.
Obj Alter	Object alter authority provides authority to change the attributes of an object, such as adding or removing triggers for a database file.
Obj Ref	Object reference authority provides authority to specify the object as the first level in a referential constraint.
Data Read	Read authority provides authority to access the contents of the object.
Data Add	Add authority provides authority to add entries to the object.
Data Update	Update authority provides authority to change the content of existing entries in the object.
Data Delete	Delete authority provides authority to remove entries from the object.
Data Execute	Execute authority provides authority to run a program or search a library or directory.

Table of Contents Previous Next

1.5 The Data Viewer

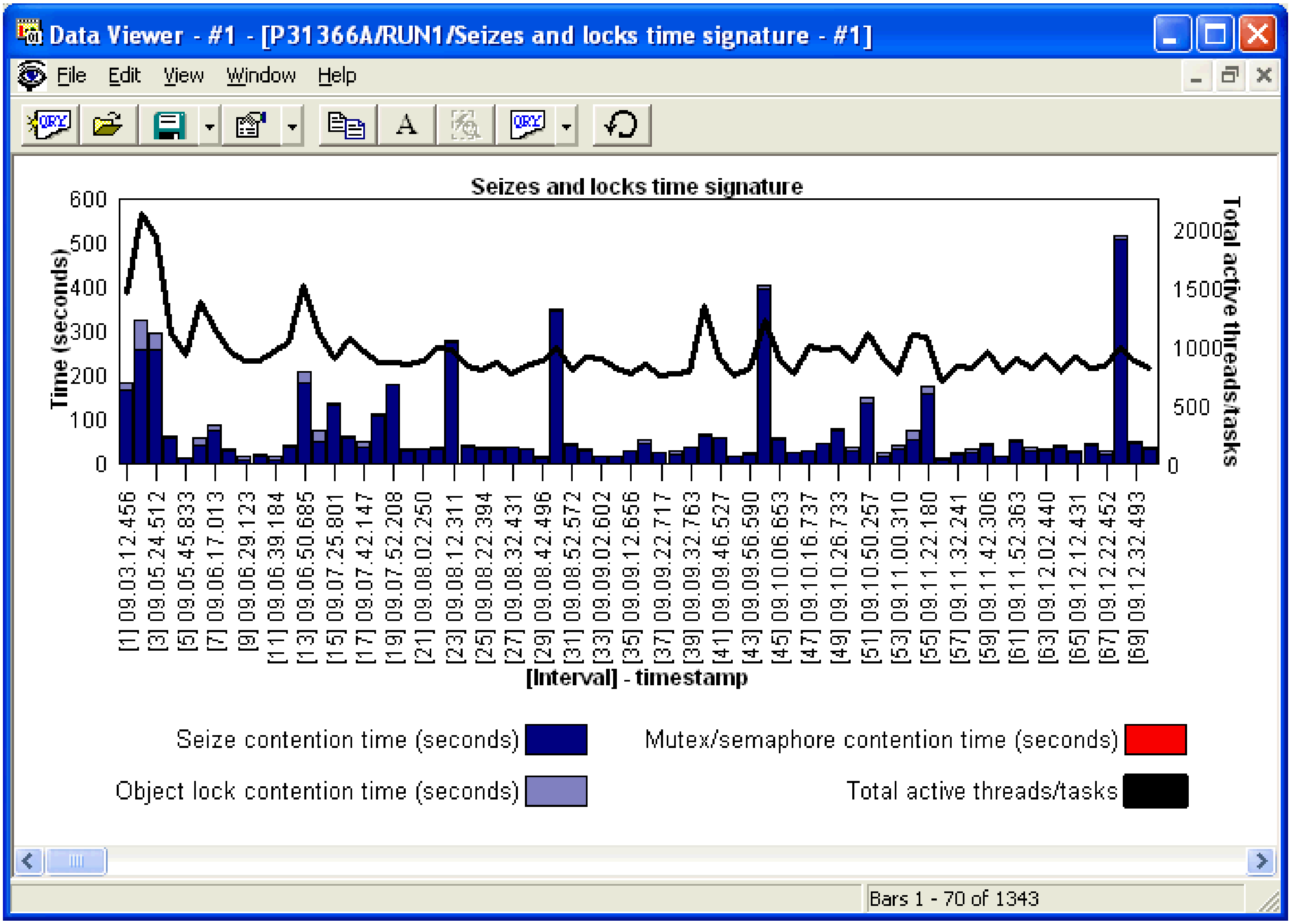
The Data Viewer is a window within iDoctor for iSeries used for displaying tables and graphs over data on an iSeries. You can have as many Data Viewer windows open at one time as you want. The data behind the views within a Data Viewer may come from any number of systems desired.

These views are maniuplated in essentially the same way as within the Main Window (they can be tiled, cascaded, etc). Tables and graphs are typically (but not always) opened from the component views in the Main Window. There are also interfaces to open tables and graphs directly from the Data Viewer window instead of going back to the Main Window.

The data behind a table or graph in iDoctor for iSeries is produced by running an SQL query over one or more database files. This query is supplied by iDoctor for iSeries whenever working with analyses within Job Watcher or PEX Analyzer. The query itself can be viewed and manipulated in the query definition interface. Queries can be saved and restored for future use.

In addition to database files the Data Viewer also has support for viewing spool files containing information such as job logs.

An example of a graph from Job Watcher is shown below.

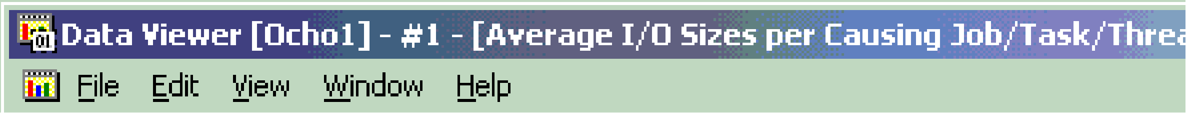


[The Data Viewer]

Table of Contents | Previous | Next

1.5.1 Menu Options

This section discusses the menu options available within the iDoctor for iSeries Data Viewer. This only covers the menus available at the top of the Data Viewer window and does not cover the popup menus available within views displayed inside the Data Viewer.


Data Viewer [Ocho1] - #1 - [Average I/O Sizes per Causing Job/Task/Threa
File
Edit
View
Window
Help

[The Data Viewer Menus]

The table below outlines the different types of operations that may be performed within the Data Viewer.

File Menu	Description
New SQL Query	Opens a new instance of an SQL Query View. The SQL Query View is used to create a query using Structured Query Language (SQL). The top portion of the view is an area where you can enter an SQL statement and the bottom portion is the result or output from the statement above. You can either edit the statement directly or use the query definition interface to change it. Any changes you make via query definition will be immediately visible in the top portion of this view.
Open File/Member	This option allows you to open any library/file/member on the system using the Open File Window. This window lets you browse for the physical or logical file you want to open. If you do not specify the member, you will be prompted to select the member from a list if the file is a multiple member file.
Save | View As...	This option allows you to save the contents of a table or graph view to a file. When using this option the entire contents of the table or graph view are saved. When saving a table, you can choose between rich text, comma separated and tab separated text formats. When saving a graph view, the file created will be a jpeg image of the current visible page of the graph.
Save | Selection As...	The option allows you to save the **selected** contents of a table to a file. When using this option only the selected records or block of cells are written to the file. When using this option you can choose between rich text, comma separated and tab separated text formats. This option is not available for graph views.
Save | Query As...	This option allows you to save the current table's query (SQL) to a file on the server. This feature allows you to save iDoctor for iSeries queries so they can be used later A window will be displayed asking for a description of the query (50 characters max).
Close	This menu will close the active view in the Data Viewer.

Print	This menu allows you to print the active graph view. This option is only available for graph views.
Close Data Viewer	Use the menu to immediately close the current data viewer and all open views and windows that are open within it.

Edit Menu	Description
Copy	Copies the current selection from the active table or graph view to the clipboard. If a table view is active, this is only enabled when something in the table has been selected. For graph views this will copy an image of the current graph to the clipboard.
Find...	This menu allows you to reposition the current record position in a table view, based on some input you supply. The Find Dialog will be displayed and you can use it search through a column for a specific value.
Set Font...	This menu displays a window allowing you to set the font used for the table views in the iDoctor for iSeries application. This option does not apply to the graph views. The font sizes used in the graph views are controlled in the Preferences window.
Preferences...	This menu displays a window allowing you to set user preferences for the iDoctor for iSeries application.

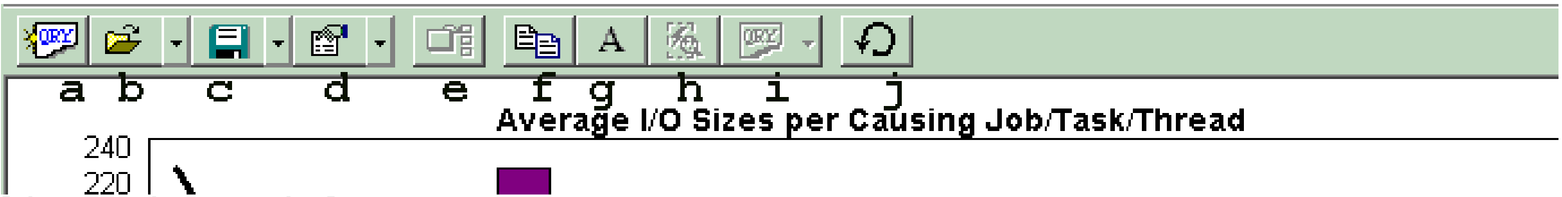
View Menu	Description
Record Quick View	This menu will display horizontally the currently selected record in the active table view if one is available. This can be very useful to see all the details for a specific record of data without having to scroll as much.
Toolbar	This menu will either show or hide the toolbar. If the toolbar is already visible then there will be a checkmark next to the menu.
Status Bar	This menu will either show or hide the status bar. If the status bar is already visible then there will be a checkmark next to the menu.
Field Descriptions	Use this menu to toggle the display of fields in table views from short names to long descriptions and vice versa. If a checkbox is next to this menu then long field descriptions are displayed, otherwise short field names are displayed. Changing this setting will effect all future table views opened with iDoctor for iSeries.
Auto-refresh real-time data	Use this menu to toggle the setting for auto-refreshing real-time views (like those in Job Watcher). A checkmark next to this menu means that this feature is turned on.
Always show new real time data	Use this menu to toggle the setting for always showing real-time data in real-time views (like those in Job Watcher). A checkmark next to this menu means that this feature is turned on. If this option is turned on the newest data will always be scrolled into view whenever possible. If the sort order changes from that of the original, this feature won't work properly.
Refresh	Refresh the currently active table or graph view.

Window Menu	Description

Cascade	Use this menu to rearrange all views in the Data Viewer in an overlapping sequence starting in the upper left corner of the window.
Tile Horizontally	Use this menu to rearrange all views in the Data Viewer such that each view will have an equal distribution of the available height in the Data Viewer. The views will not overlap each other.
Tile Vertically	Use this menu to rearrange all views in the Data Viewer such that each view will have an equal distribution of the available width in the Data Viewer. The views will not overlap each other.
Close All	Closes all open views within the Data Viewer. The Data Viewer will remain open.

Help Menu	**Description**
Contents	This menu will launch your web browser and takes you to the online documentation.
iDoctor for iSeries Download Page	Launches your web browser and takes you to the iDoctor download page.
About	This will display version information for the iDoctor for iSeries client.

1.5.2 Toolbar



[The Data Viewer Toolbar]

The following table outlines the different toolbar icons available in the Data Viewer and their purpose.

Alpha Identifier	Description
"a"	Opens a new instance of an SQL Query View. The SQL Query View is used to create a query using Structured Query Language (SQL). The top portion of the view is an area where you can enter an SQL statement and the bottom portion is the result or output from the statement above. You can either edit the statement directly or use the query definition interface to change it. Any changes you make via query definition will of course be immediately visible in the top portion of this view.
"b"	This option allows you to open any library/file/member on the system using the Open File Window. This window lets you browse for the physical or logical file you want to open. If you do not specify the member, you will be prompted to select the member from a list.
"c"	This option allows you to save the contents of a table or graph view to a file. When using this option the entire contents of the table or graph view are saved. When saving a table, you can choose between rich text, comma separated and tab separated text formats. When saving a graph view, the file created will be a jpeg image of the current visible page of the graph.
"d"	Use this option to view the Query (SQL statement) behind the active graph or table view. From these properties you can also see what file/library/member you are working with. You can copy the SQL to the clipboard and paste it into an SQL Query View to be modified directly if you want.
"f"	Copies the current selection from the active table or graph view to the clipboard. If a table view is active, this is only enabled when something in the table has been selected. For graph views this will copy an image of the current graph to the clipboard.
"g"	This menu displays a window allowing you to set the font used for the table views in the iDoctor for iSeries application.
"h"	This menu allows you to reposition the current record position in a table view, based on some input you supply. The Find Dialog will be displayed and you can use it search through a column for a specific value.
"i"	The query definition menu has a large number of submenus each letting you work with the query behind the active table view or in some cases graph views.
"j"	Refresh the currently active table or graph view.

Table of Contents Previous Next

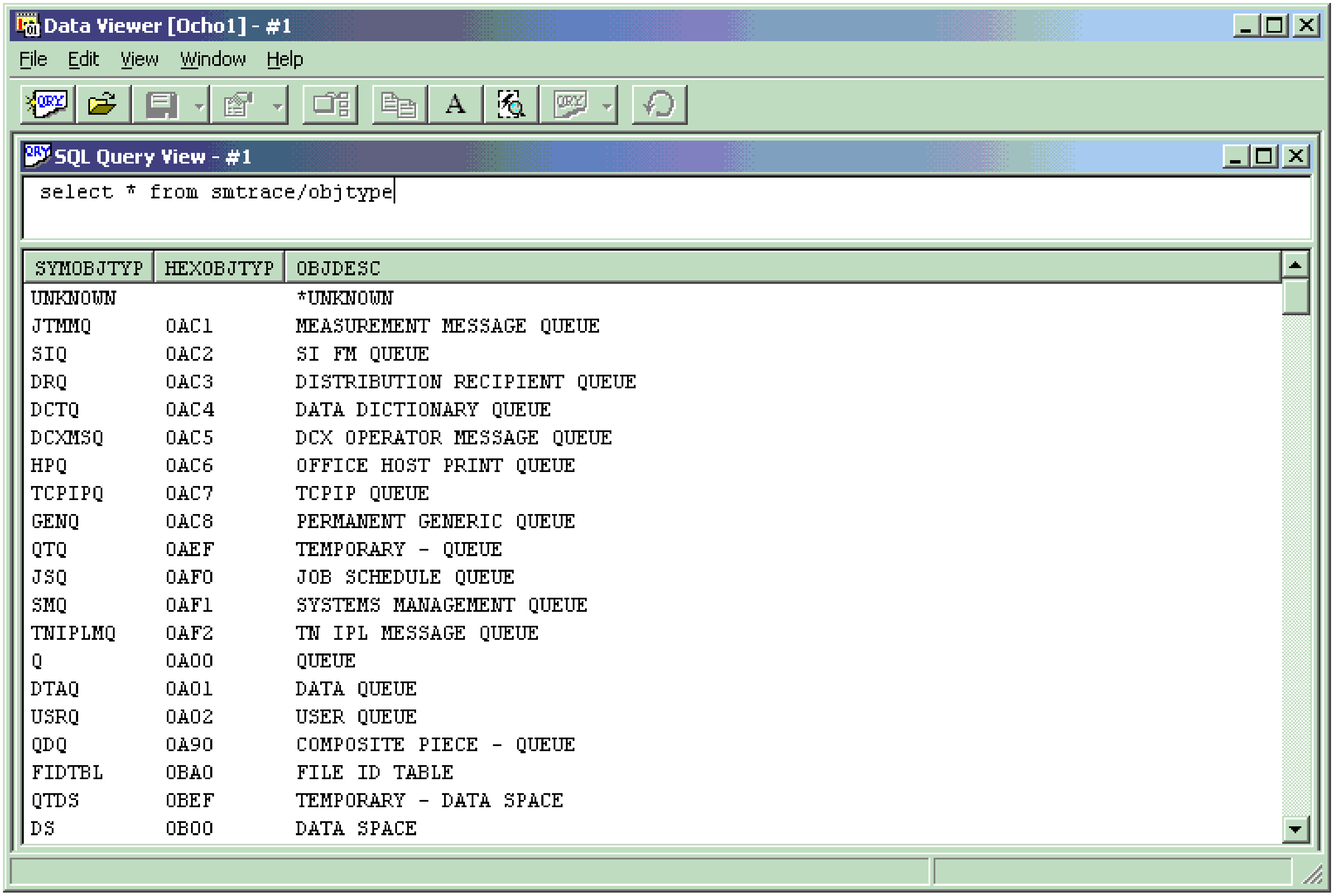
1.5.3 SQL Query View

The SQL Query View lets you dynamically execute and display the results of a query using Structured Query Language (SQL). The top portion of the view is an area where you can enter an SQL statement and the bottom portion is the result or output from the statement above. The bottom portion is a table view and allows the typical copy to clipboard, query definition, export, find functions that are standard in a table view.

The queries you create with this view may be saved and restored for later use and their definitions can be viewed and manipulated using the query definition interface. A benefit of using query definition in concert with the SQL Query View is any changes to the query definition (which changes the SQL statement) will be immediately visible in the top portion of this window.

In order to execute your quey after typing it in, right-click on the top portion of the view and choose the Execute menu or press the F4 key. You can reexecute your query at any time.

An example of this interface is shown below:



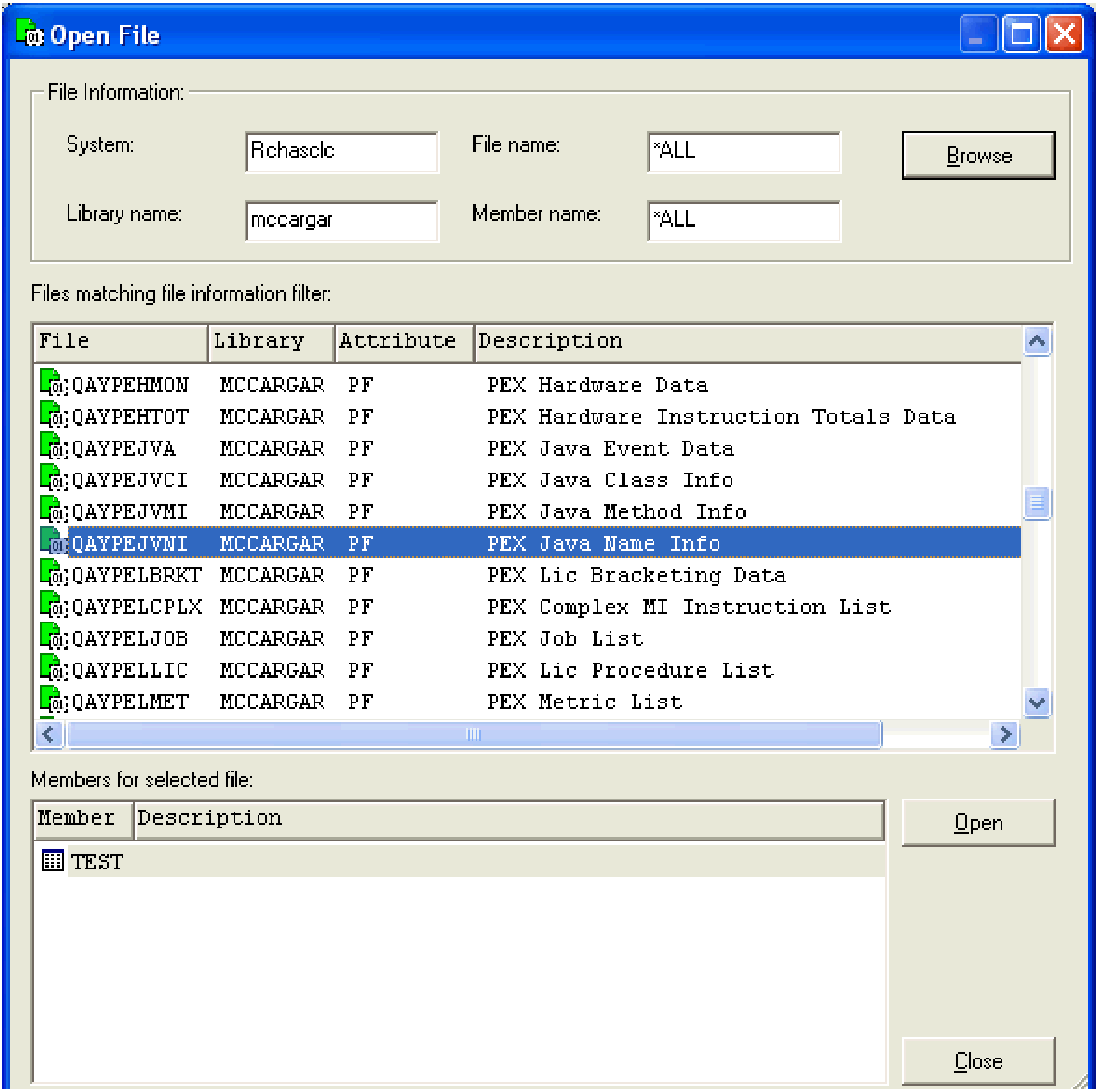
```
select * from smtrace/objtype
```

SYMOBJTYP	HEXOBJTYP	OBJDESC
UNKNOWN		*UNKNOWN
JTMMQ	0AC1	MEASUREMENT MESSAGE QUEUE
SIQ	0AC2	SI FM QUEUE
DRQ	0AC3	DISTRIBUTION RECIPIENT QUEUE
DCTQ	0AC4	DATA DICTIONARY QUEUE
DCXMSQ	0AC5	DCX OPERATOR MESSAGE QUEUE
HPQ	0AC6	OFFICE HOST PRINT QUEUE
TCPIPQ	0AC7	TCPIP QUEUE
GENQ	0AC8	PERMANENT GENERIC QUEUE
QTQ	0AEF	TEMPORARY - QUEUE
JSQ	0AF0	JOB SCHEDULE QUEUE
SMQ	0AF1	SYSTEMS MANAGEMENT QUEUE
TNIPLMQ	0AF2	TN IPL MESSAGE QUEUE
Q	0A00	QUEUE
DTAQ	0A01	DATA QUEUE
USRQ	0A02	USER QUEUE
QDQ	0A90	COMPOSITE PIECE - QUEUE
FIDTBL	0BA0	FILE ID TABLE
QTDS	0BEF	TEMPORARY - DATA SPACE
DS	0B00	DATA SPACE

Table of Contents Previous Next

1.5.4 Open File Window

This option allows the user to open any library/file/member on the system. A window is displayed where the user can browse for the physical or logical file to open. When the file selection changes, the list of members shown is also updated based on the selection.



Open File

File Information:

System:	Rchasclc	File name:	*ALL	Browse
Library name:	mccargar	Member name:	*ALL	

Files matching file information filter:

File	Library	Attribute	Description
QAYPEHMON	MCCARGAR	PF	PEX Hardware Data
QAYPEHTOT	MCCARGAR	PF	PEX Hardware Instruction Totals Data
QAYPEJVA	MCCARGAR	PF	PEX Java Event Data
QAYPEJVCI	MCCARGAR	PF	PEX Java Class Info
QAYPEJVMI	MCCARGAR	PF	PEX Java Method Info
QAYPEJVNI	MCCARGAR	PF	PEX Java Name Info
QAYPELBRKT	MCCARGAR	PF	PEX Lic Bracketing Data
QAYPELCPLX	MCCARGAR	PF	PEX Complex MI Instruction List
QAYPELJOB	MCCARGAR	PF	PEX Job List
QAYPELLIC	MCCARGAR	PF	PEX Lic Procedure List
QAYPELMET	MCCARGAR	PF	PEX Metric List

Members for selected file:

Member	Description
TEST	

The following table describes the interface elements within this window.

Inteface Element	**Description**
Library name	Type the name of the library to look for files in. Generic library names are not supported for this field. *ALL may also be used for the library name parameter. However if *ALL is used a generic file name should be used otherwise it could take several minutes in order for the list of files to be built.
File name	Type the specific name or generic name of the file to open.
Member name	Type the specific name or generic name of the member you are looking for.
Browse button	This button updates the lists based on the library/file/member information specified.
Files matching file information list	The list of files matching the file library/file specified.
Members for selected file list	The list of members for the selected file in the file list and that match the member name filter.
Open button	Opens the selected library/file/member.

Table of Contents | Previous | Next

1.5.5 Table Views

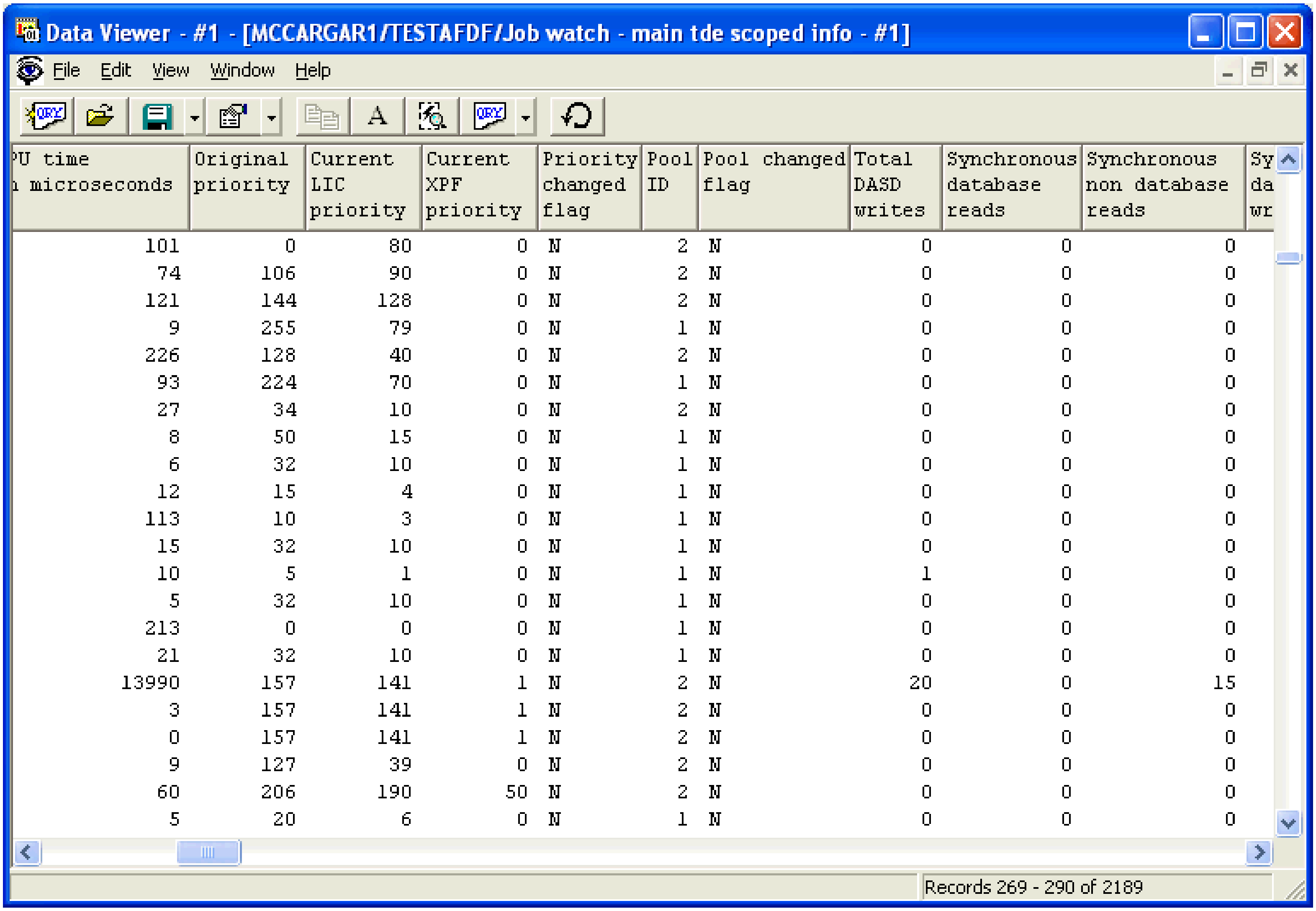
A table view shows data from database files on the iSeries. The user can display millions of records in a table view and use the scroll bar to quickly move to the records desired. For performance reasons, only records actually needed to be displayed are loaded into the client via blocking methods allowing quick relative scrolling.

The SQL behind the table view can be modified at any time using the Query Definition interface. The data may also be sorted by clicking the desired column to sort by. Clicking again on the same column heading will resort the data in the reverse order.

Data in a table view may be selected for copy and paste to a file or to the clipboard. A set of records -or- a block of cells may be selected at any one time. Click the left mouse button and drag across the cells desired in order to make a block selection. Once a selection is made, use the Edit | Copy to copy the current selection to the clipboard. Use the File | Save Selection As... menu to write the selection to a file.

The record indicator in the status bar will show which records are currently being viewed out of the total possible in the active view.

An example of a table view is the following:



Data Viewer - #1 - [MCCARGAR1/TESTAFDF/Job watch - main tde scoped info - #1]

PU time n microseconds	Original priority	Current LIC priority	Current XPF priority	Priority changed flag	Pool ID	Pool changed flag	Total DASD writes	Synchronous database reads	Synchronous non database reads	Sy da wr
101	0	80	0	N	2	N	0	0	0	
74	106	90	0	N	2	N	0	0	0	
121	144	128	0	N	2	N	0	0	0	
9	255	79	0	N	1	N	0	0	0	
226	128	40	0	N	2	N	0	0	0	
93	224	70	0	N	1	N	0	0	0	
27	34	10	0	N	2	N	0	0	0	
8	50	15	0	N	1	N	0	0	0	
6	32	10	0	N	1	N	0	0	0	
12	15	4	0	N	1	N	0	0	0	
113	10	3	0	N	1	N	0	0	0	
15	32	10	0	N	1	N	0	0	0	
10	5	1	0	N	1	N	1	0	0	
5	32	10	0	N	1	N	0	0	0	
213	0	0	0	N	1	N	0	0	0	
21	32	10	0	N	1	N	0	0	0	
13990	157	141	1	N	2	N	20	0	15	
3	157	141	1	N	2	N	0	0	0	
0	157	141	1	N	2	N	0	0	0	
9	127	39	0	N	2	N	0	0	0	
60	206	190	50	N	2	N	0	0	0	
5	20	6	0	N	1	N	0	0	0	

Records 269 - 290 of 2189

A popup menu is available by right-clicking anywhere on the table. The following options are available via the popup menu:

Popup Menu	Description

Record Quick View	This option will display in the property pages a vertical view of the current record(s) selected. If multiples are selected this option can be used to show a comparison between two records in a side-by-side view.
Copy	Copies the current text selection to the clipboard.
Find...	This menu allows the user to reposition the current record in a table view, based on input supplied if matching information is found.
Graph Definition	This menu contains an option to create a new user-defined graph from the current report.
Query Definition	This menu has a large number of submenus each letting the user work with the query behind the active table view.
Set Font...	This menu displays a window allowing customization of the font used for all table views.
Preferences...	This menu displays a window allowing the user to set customized settings for the iDoctor for iSeries application.
Properties	Displays the properties for the current report. The information shown in the property pages varies based on the type of report being viewed.

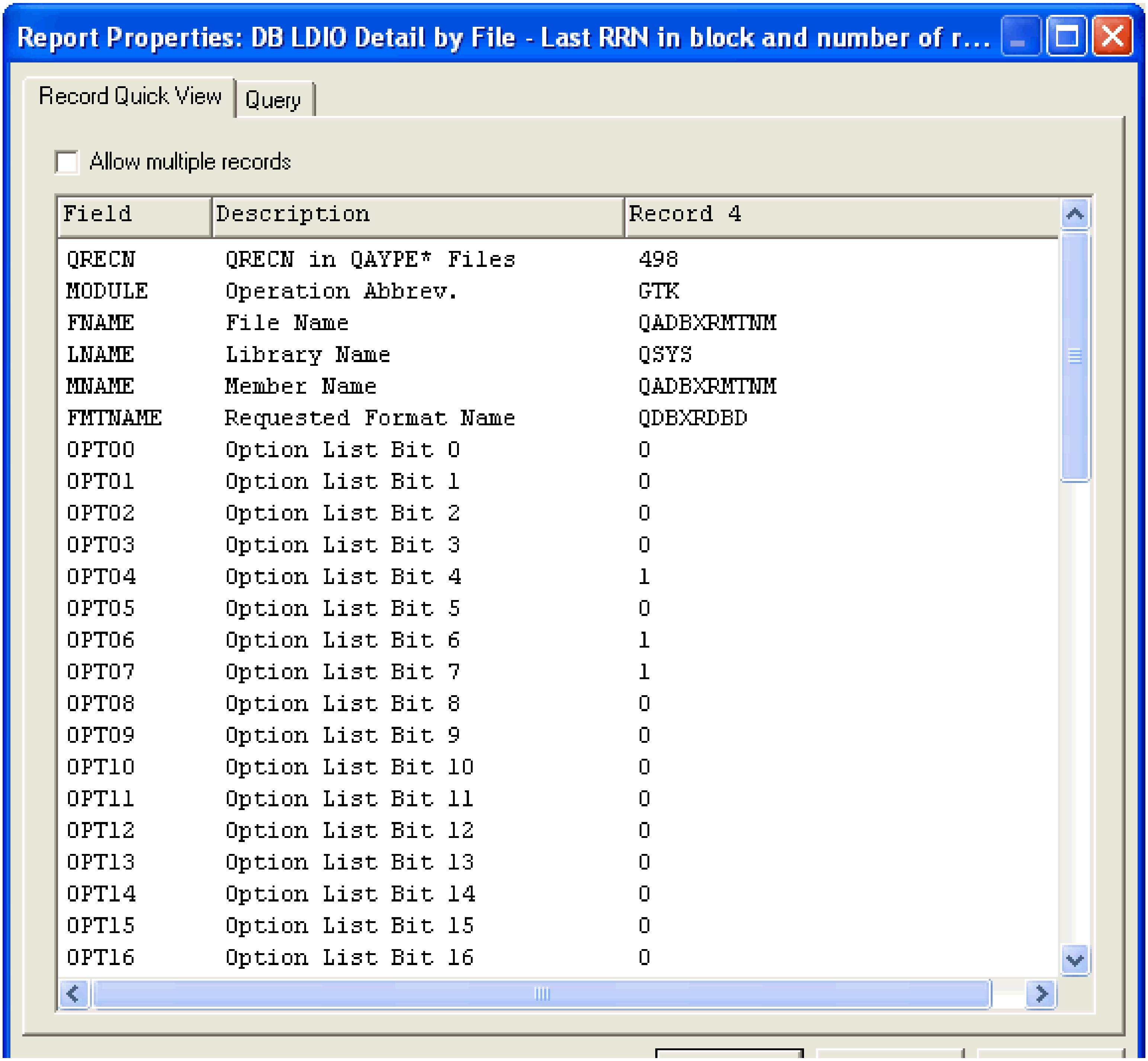
Depending on the type of report shown in a table view, other menu applicable to that report type will be shown. These are mentioned in the Job Watcher and PEX Analyzer documentation for the applicable report types.

Table of Contents | Previous | Next

1.5.5.1 Report Properties - Record Quick View

This window is part of the property pages for a table view. The Record Quick View page shows all of the data for the selected record from the table in a vertical list. This can make it easier to see all the data for a single record if many fields exist in the table. Access this window by double-clicking on any record in a table view or by using the Properties menu.

An example of this window for a PEX Analyzer table view is the following:


Report Properties: DB LDIO Detail by File - Last RRN in block and number of r...
Record Quick View
Query

□ Allow multiple records

Field	Description	Record 4
QRECN	QRECN in QAYPE* Files	498
MODULE	Operation Abbrev.	GTK
FNAME	File Name	QADBXRMTNM
LNAME	Library Name	QSYS
MNAME	Member Name	QADBXRMTNM
FMTNAME	Requested Format Name	QDBXRDBD
OPT00	Option List Bit 0	0
OPT01	Option List Bit 1	0
OPT02	Option List Bit 2	0
OPT03	Option List Bit 3	0
OPT04	Option List Bit 4	1
OPT05	Option List Bit 5	0
OPT06	Option List Bit 6	1
OPT07	Option List Bit 7	1
OPT08	Option List Bit 8	0
OPT09	Option List Bit 9	0
OPT10	Option List Bit 10	0
OPT11	Option List Bit 11	0
OPT12	Option List Bit 12	0
OPT13	Option List Bit 13	0
OPT14	Option List Bit 14	0
OPT15	Option List Bit 15	0
OPT16	Option List Bit 16	0

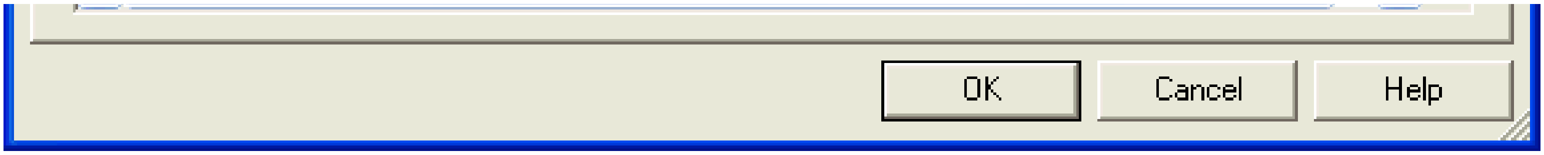
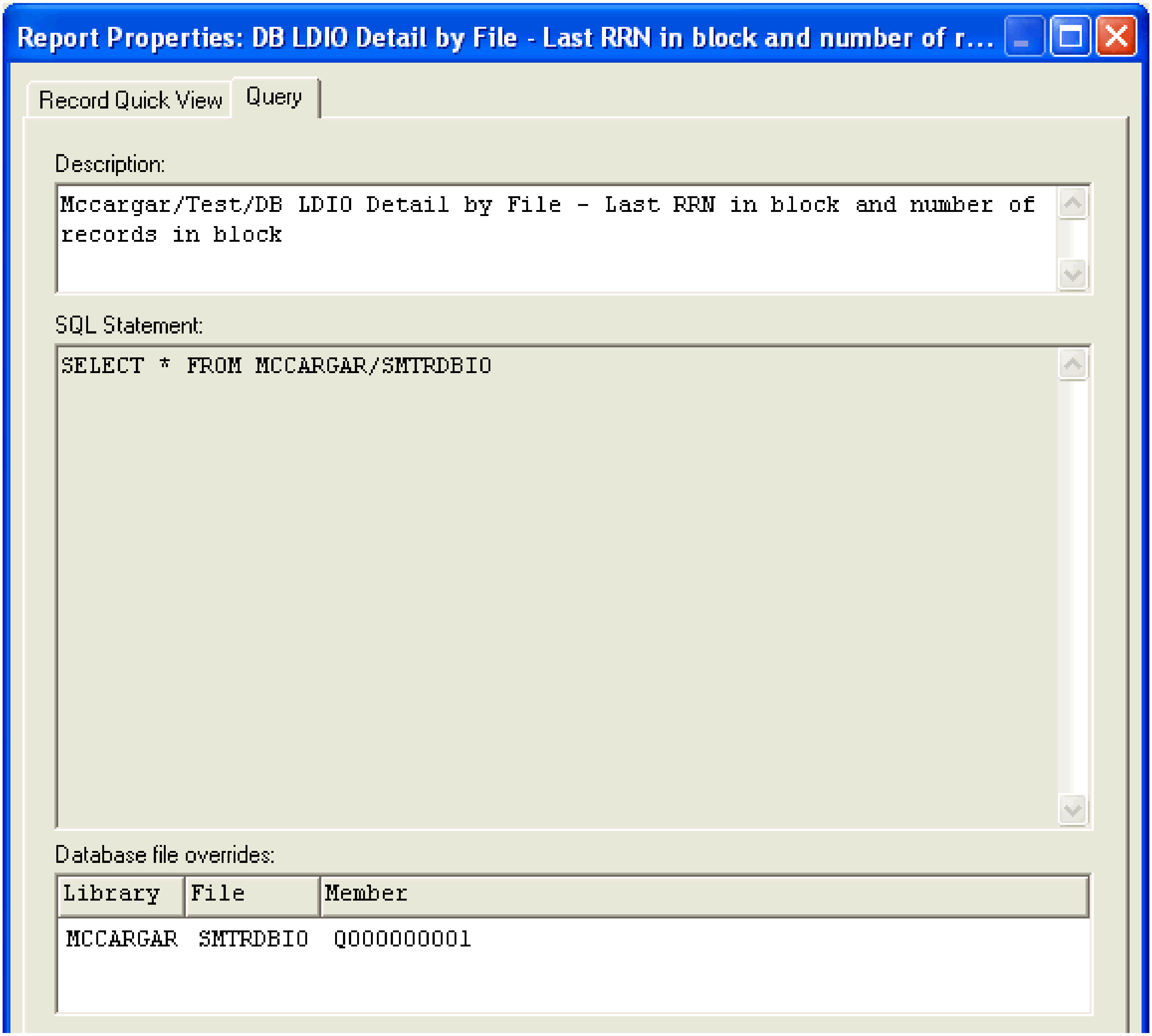

OK
Cancel
Help

Table of Contents Previous Next

1.5.5.2 Report Properties - Query

The Query page of the Report Properties window displays the SQL statement used to produce the current table view. This window also displays the title of the table view and the overrides used to produce the current table view. Because SQL does not support multiple member tables, overrides (see the OVRDBF command) are issued before the query is executed to select which file(s)/member(s) should be used when running the SQL statements.

An example of this page is the following:


Report Properties: DB LDIO Detail by File - Last RRN in block and number of r...
Record Quick View
Query

Description:

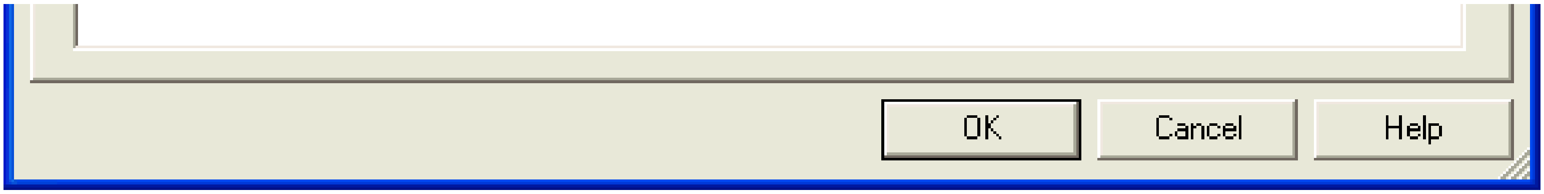
Mccargar/Test/DB LDIO Detail by File - Last RRN in block and number of records in block

SQL Statement:

```
SELECT * FROM MCCARGAR/SMTRDBIO
```

Database file overrides:

Library	File	Member
MCCARGAR	SMTRDBIO	Q000000001



The interface elements within this window are described in more detail in the table below:

Interface Element	**Description**
Description	The text description identifying the report.
SQL statement	The complete SQL statement for the query definition. Any changes made to the query using the Query Definition interfaces (field selection, record selection, sort by, etc) will be reflected in this SQL statement.
Database file overrides	This list identifies all of the members used for each file in the query. An override is used to point to a specific member when executing the query since this cannot be indicated by a SQL statement.

Table of Contents Previous Next

1.5.6 Graph Views

The graph views show visual representations of any data queryable on the system. The only types of graphs supported currently are bar graphs. There are 4 different types of bar graphs supported: stacked horizontal, horizontal bar, stacked vertical bar, vertical bar. The vertical bar graph type and horizontal bar graph type shows bars side-by-side instead of stacked for the same data point. Each color in the graph represents a different value or field in the data. Most graphs will have a legend identifying the information shown at the bottom of the window.

Use the scroll bars to navigate through the information in the graphs. Due to the potential to view vast amounts of data at one time, the graph data is shown a page at a time. The number of bars shown per page is configurable through the Preferences interface. When scrolling through the data the scale of the axes can be set to adjust automatically. This is another option on the Preferences interface. If automatic scaling is disabled then the graph scale will be set to the maximum/minimum values of the first page shown in the graph.

Additional information about each piece of data in the graph is avaialble by moving the mouse over the bar of interest. A flyover help window will appear in yellow providing this information. Some of this information is also displayed in the status bar as the mouse moves from bar to bar. The user can also click on any bar to get a complete look at all the information for that particular piece of the graph and any other applicable data that goes with it (interval, job, etc)..

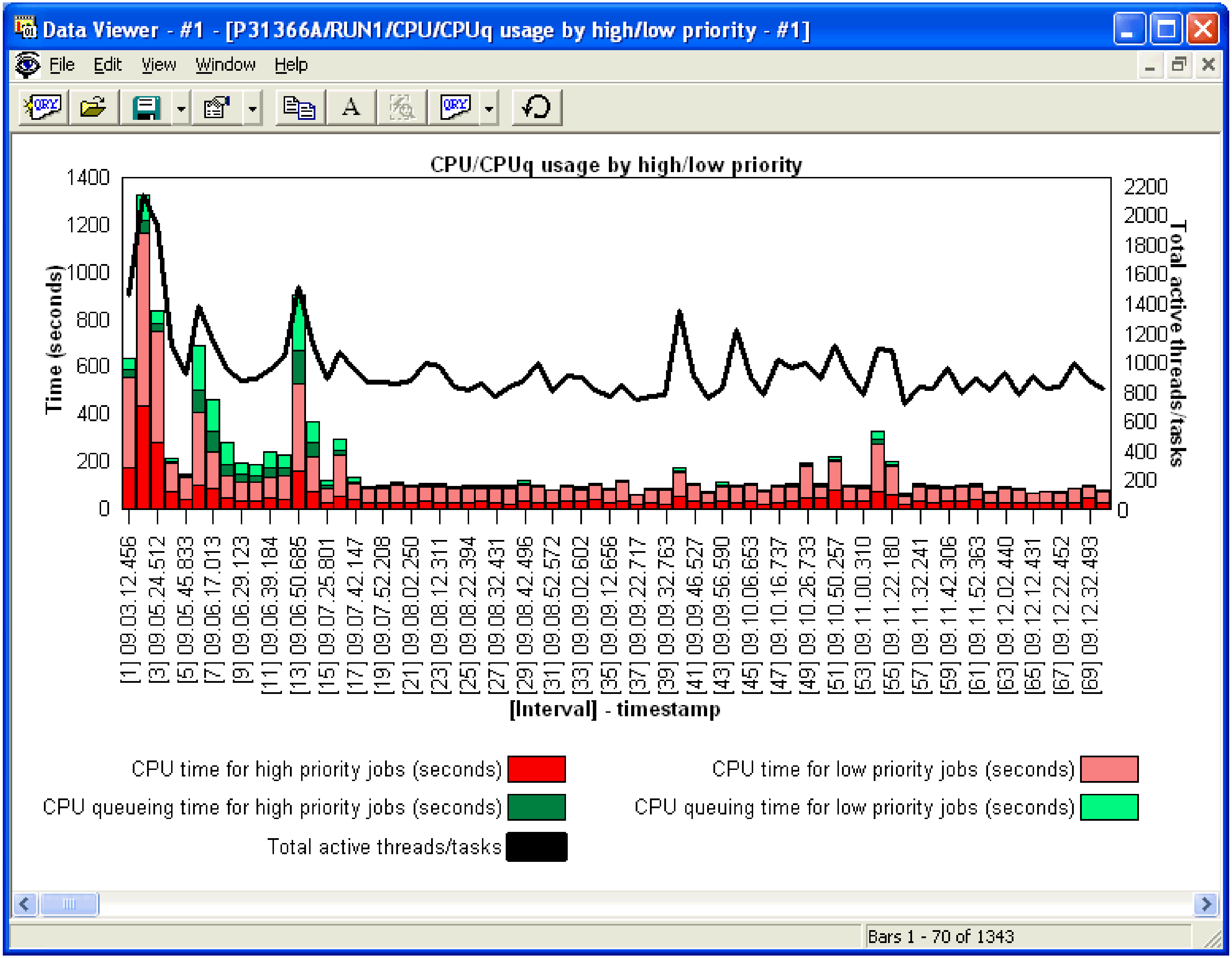
The position indicator in the status bar indicates exactly which bars are being viewed out of the total possible.

There are two types of graphs in iDoctor for iSeries: iDoctor-supplied graphs and user-defined graphs.

iDoctor-supplied graphs

iDoctor-supplied graphs are graphs shipped with iDoctor for iSeries. There are several iDoctor-supplied graphs that come with PEX Analyzer and many more in Job Watcher.

An example of an iDoctor-supplied graph from Job Watcher is:



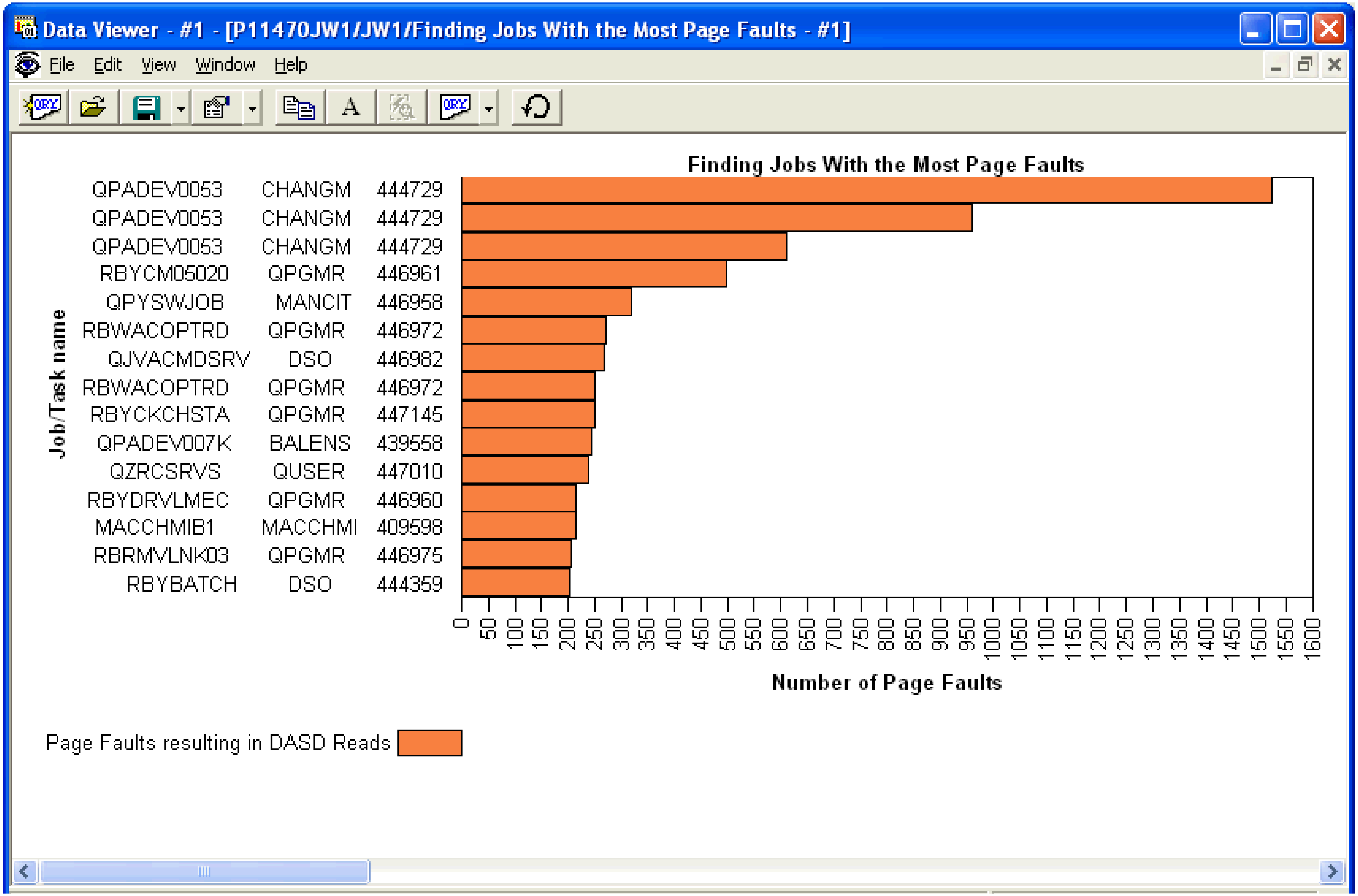
There is a difference between PEX Analyzer graphs and Job Watcher graphs. The Job Watcher graphs allow the user to modify the query definition behind the graph at any time, the PEX Analyzer iDoctor-supplied graphs does not allow this due to the way the PEX Analyzer graphs are constructed.

User-defined graphs

User-defined graphs are created by the user and saved into a definition on the server. The settings for a user-defined graph is called its graph definition.

A user-defined graph is created from a table view. The SQL statement (query) behind the table view is what is used for the query behind the graph. Creating a graph is done using the Graph Definition | Define New... popup menu of a table view.

An example of a user-defined graph over Job Watcher data is shown below:



Popup Menu

Both types of graphs offer the following features via the right-mouse click popup menu:

Popup Menu	Description
Copy	Copy the current graph view as an image (windows bitmap) to the clipboard.
Save As...	Save the current graph view as an image (.jpeg).
Preferences	Displays the preferences window. With this interface the user can change the number of bars shown per page and customize font and other graph settings.
Properties	Displays the properties for the currently selected point on the graph as well as other information such as the SQL statement behind the graph view.

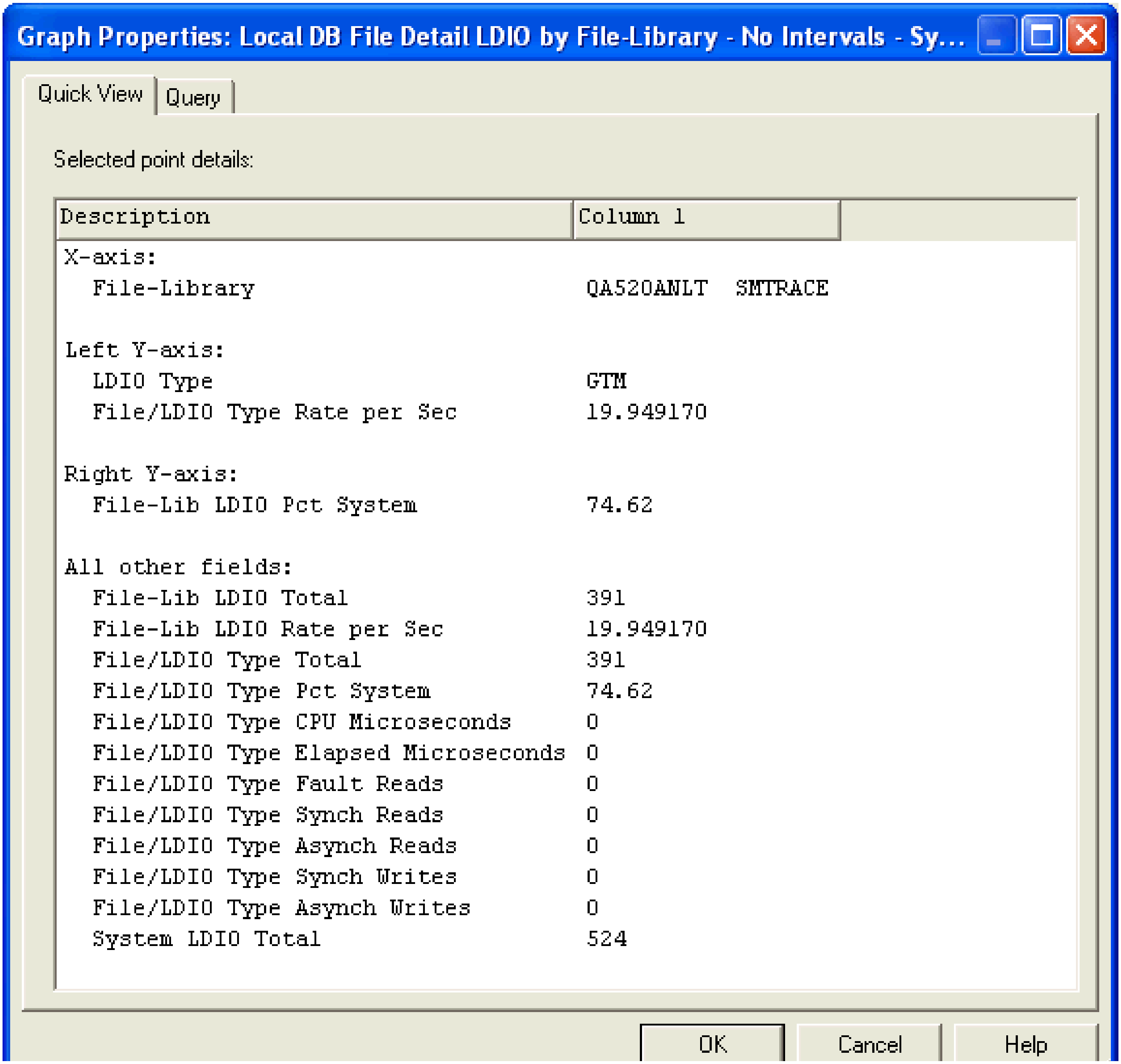
Other popup menu items are shown depending on the type of data/analysis being viewed. These additional options are covered under the documentation for the appropriate analysis type.

Table of Contents Previous Next

1.5.6.1 Graph Properties - Quick View

This window is part of the property pages for a graph view. The Quick View page contains all of the information about a particular bar in the graph from the fields available in the query used to created the graph. Access this window by clicking on any bar in a graph view.

An example of this window for a PEX Analyzer graph is the following:


Graph Properties: Local DB File Detail LDIO by File-Library - No Intervals - Sy...
Quick View
Query
OK
Cancel
Help

Selected point details:

```
Description                                Column 1
X-axis:
  File-Library                             QA520ANLT  SMTRACE

Left Y-axis:
  LDIO Type                                GTM
  File/LDIO Type Rate per Sec              19.949170

Right Y-axis:
  File-Lib LDIO Pct System                 74.62

All other fields:
  File-Lib LDIO Total                      391
  File-Lib LDIO Rate per Sec               19.949170
  File/LDIO Type Total                     391
  File/LDIO Type Pct System                74.62
  File/LDIO Type CPU Microseconds          0
  File/LDIO Type Elapsed Microseconds      0
  File/LDIO Type Fault Reads               0
  File/LDIO Type Synch Reads               0
  File/LDIO Type Asynch Reads              0
  File/LDIO Type Synch Writes              0
  File/LDIO Type Asynch Writes             0
  System LDIO Total                        524
```

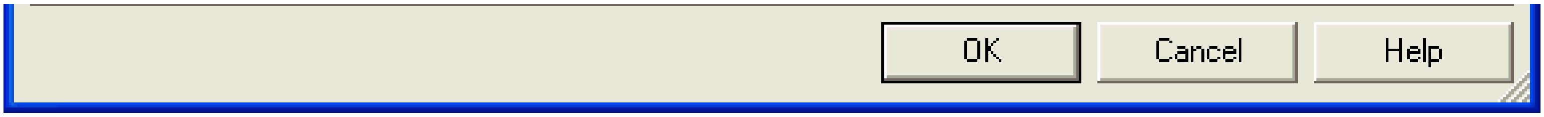
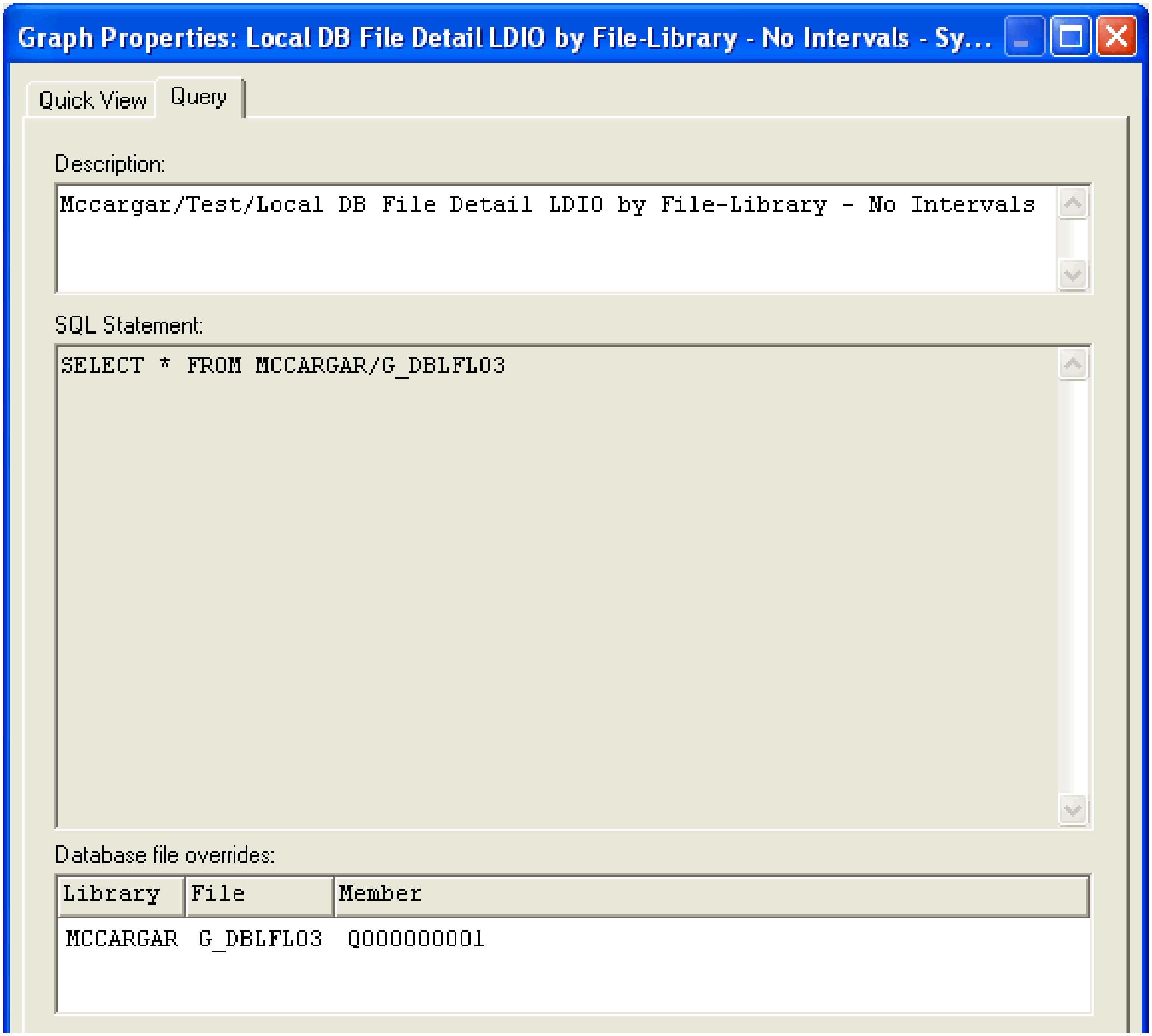

OK
Cancel
Help

Table of Contents Previous Next

1.5.6.2 Graph Properties - Query

The Query page of the Graph Properties window displays the SQL statement used to produce the current graph view. This window also displays the title of the graph view and the overrides used to produce the current view. Because SQL does not support multiple member tables, overrides (see the OVRDBF command) are issued before the query is executed to select which file(s)/member(s) should be used when running the SQL statements.

An example of this page is the following:


Graph Properties: Local DB File Detail LDIO by File-Library - No Intervals - Sy...
Quick View
Query

Description:

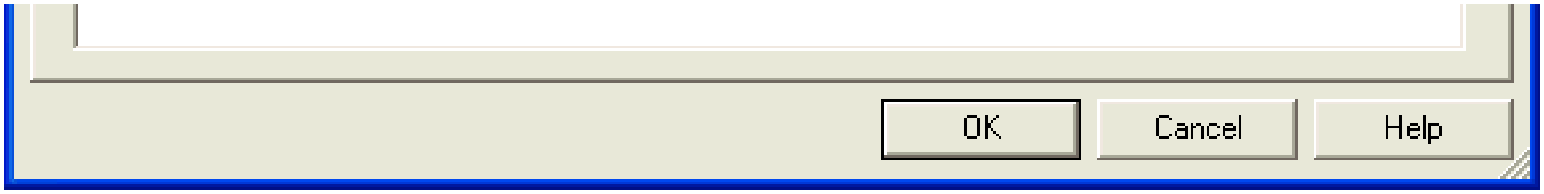
Mccargar/Test/Local DB File Detail LDIO by File-Library - No Intervals

SQL Statement:

```
SELECT * FROM MCCARGAR/G_DBLFL03
```

Database file overrides:

Library	File	Member
MCCARGAR	G_DBLFL03	Q000000001



The interface elements within this window are described in more detail in the table below:

Interface Element	**Description**
Description	The text description identifying the report.
SQL statement	The complete SQL statement for the query definition. Any changes made to the query using the Query Definition interfaces (field selection, record selection, sort by, etc) will be reflected in this SQL statement.
Database file overrides	This list identifies all of the members used for each file in the query. An override is used to point to a specific member when executing the query since this cannot be indicated by a SQL statement.

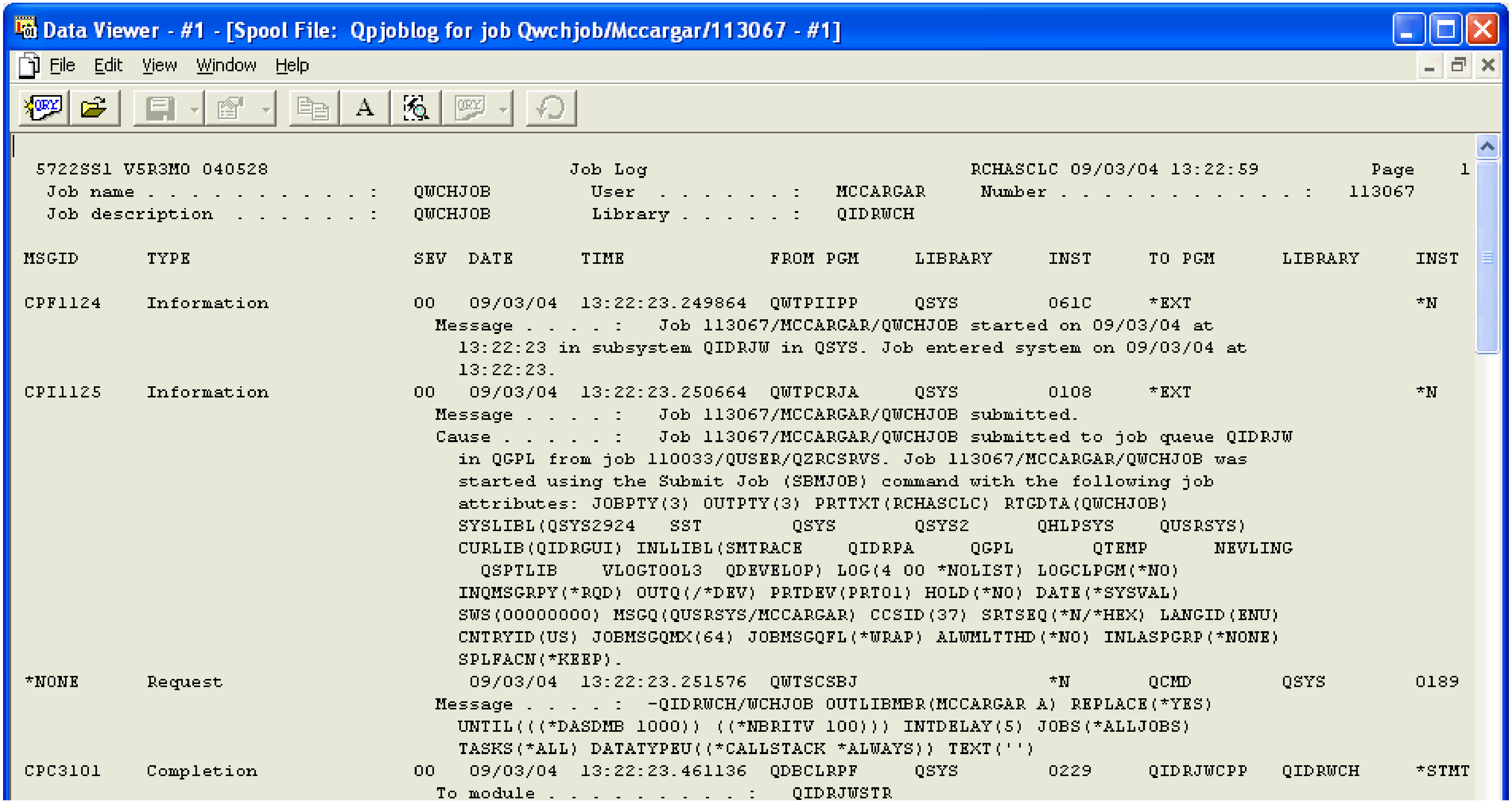
Table of Contents Previous Next

1.5.7 Spool File Views

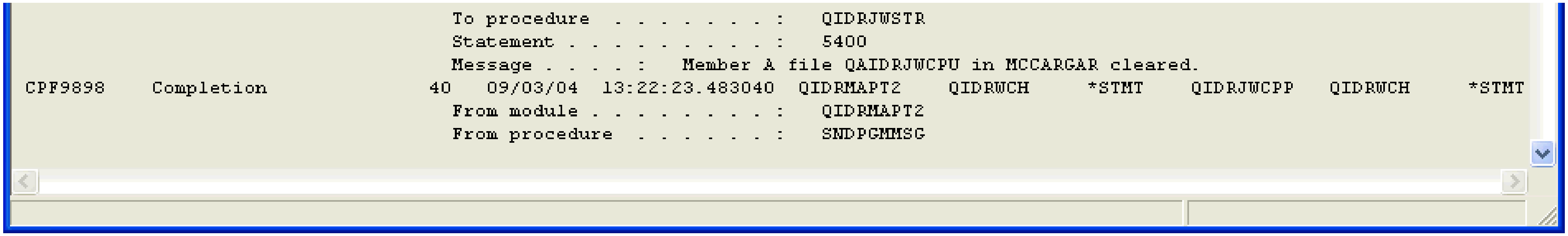
The Data Viewer can be used to display the contents of spool files on the server. Whenever a job log for a collection is viewed that has already ended the job log is displayed in this viewer.

The spool file viewer will read in the entire contents of the spool file into the viewer. Although this will cause delays when reading large files this allows the user to more quickly perform a text search using the Find feature on the toolbar after the data is loaded into the client.

Other types of spool files besides job logs are displayable. However, they can only be opened using the Object Explorer component. An example of a Spool File View is shown below:


Data Viewer - #1 - [Spool File: Qpjoblog for job Qwchjob/Mccargar/113067 - #1]
File
Edit
View
Window
Help

```
 5722SS1 V5R3M0 040528                              Job Log                             RCHASCLC 09/03/04 13:22:59          Page    1
  Job name . . . . . . . . . . :   QWCHJOB         User  . . . . . . :   MCCARGAR     Number . . . . . . . . . . . :   113067
  Job description  . . . . . . :   QWCHJOB         Library . . . . . :   QIDRWCH

MSGID       TYPE                    SEV  DATE       TIME              FROM PGM     LIBRARY     INST      TO PGM      LIBRARY     INST

CPF1124     Information             00   09/03/04   13:22:23.249864   QWTPIIPP     QSYS        061C      *EXT                    *N
                                      Message . . . . :   Job 113067/MCCARGAR/QWCHJOB started on 09/03/04 at
                                        13:22:23 in subsystem QIDRJW in QSYS. Job entered system on 09/03/04 at
                                        13:22:23.
CPI1125     Information             00   09/03/04   13:22:23.250664   QWTPCRJA     QSYS        0108      *EXT                    *N
                                      Message . . . . :   Job 113067/MCCARGAR/QWCHJOB submitted.
                                      Cause . . . . . :   Job 113067/MCCARGAR/QWCHJOB submitted to job queue QIDRJW
                                        in QGPL from job 110033/QUSER/QZRCSRVS. Job 113067/MCCARGAR/QWCHJOB was
                                        started using the Submit Job (SBMJOB) command with the following job
                                        attributes: JOBPTY(3) OUTPTY(3) PRTTXT(RCHASCLC) RTGDTA(QWCHJOB)
                                        SYSLIBL(QSYS2924   SST        QSYS       QSYS2      QHLPSYS    QUSRSYS)
                                        CURLIB(QIDRGUI) INLLIBL(SMTRACE    QIDRPA     QGPL       QTEMP      NEVLING
                                          QSPTLIB    VLOGTOOL3  QDEVELOP) LOG(4 00 *NOLIST) LOGCLPGM(*NO)
                                        INQMSGRPY(*RQD) OUTQ(/*DEV) PRTDEV(PRT01) HOLD(*NO) DATE(*SYSVAL)
                                        SWS(00000000) MSGQ(QUSRSYS/MCCARGAR) CCSID(37) SRTSEQ(*N/*HEX) LANGID(ENU)
                                        CNTRYID(US) JOBMSGQMX(64) JOBMSGQFL(*WRAP) ALWMLTTHD(*NO) INLASPGRP(*NONE)
                                        SPLFACN(*KEEP).
*NONE       Request                      09/03/04   13:22:23.251576   QWTSCSBJ                 *N        QCMD        QSYS        0189
                                      Message . . . . :  -QIDRWCH/WCHJOB OUTLIBMBR(MCCARGAR A) REPLACE(*YES)
                                        UNTIL(((*DASDMB 1000)) ((*NBRITV 100))) INTDELAY(5) JOBS(*ALLJOBS)
                                        TASKS(*ALL) DATATYPEU((*CALLSTACK *ALWAYS)) TEXT('')
CPC3101     Completion              00   09/03/04   13:22:23.461136   QDBCLRPF     QSYS        0229      QIDRJWCPP   QIDRWCH     *STMT
                                      To module . . . . . . . . . :   QIDRJWSTR
```



```
                                         To procedure  . . . . . . . :   QIDRJWSTR
                                         Statement . . . . . . . . . :   5400
                                         Message . . . . :   Member A file QAIDRJWCPU in MCCARGAR cleared.
CPF9898    Completion             40   09/03/04  13:22:23.483040  QIDRMAPT2    QIDRWCH     *STMT    QIDRJWCPP   QIDRWCH     *STMT
                                         From module . . . . . . . . :   QIDRMAPT2
                                         From procedure  . . . . . . :   SNDPGMMSG
```

Table of Contents | Previous | Next

1.6 Query Definitions

Tables and graphs are created via an underlining query definition. The query definition defines exactly how data is to be retrieved and from what file(s). The Query Definition Interface is an interface over an SQL statement. Nearly all table and graph views have a query definition menu available that lets the user work with the query definition for that particular table or graph view. The only exception is the graph views in the PEX Analyzer component. Modifying the query definition for a PEX Analyzer graph is not supported and therefore this interface is not available in that case.

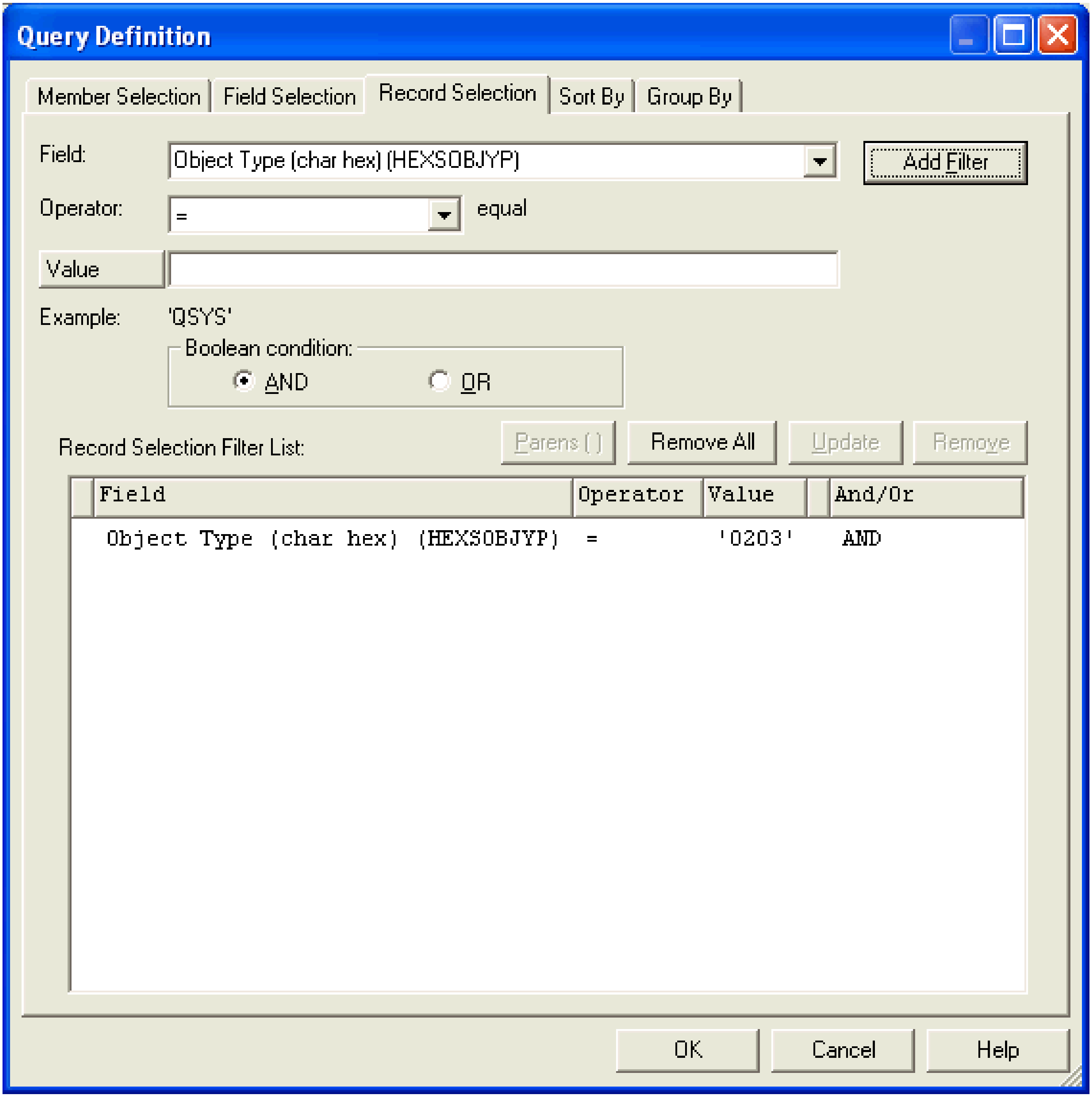
The Query Definition Interface allows a user to customize the query for the active table or graph within the Data Viewer. Right-click on the view and use the Query Definition... popup menu to start the Query Definition Interface. The Query Definition window contains several different tabs each for defining a specific part of the query.

The tabs within the query definition interface are:

Member selection - define which members from the files in the query should be overriden to in order to produce the correct data
Field selection - indicates the order and content of the data in the view
Record selection - used to filter out or include records that meet certain characteristics
Sort by - indicates which field(s) the data should be sorted by
Group by - indicates if the query is a group by query and if it is the group by fields and the having by clause of the SQL statement are definable on this page

In order to use the more advanced features of the Query Definition Interface like the "group by" page, the user needs to have understanding of SQL statement syntax. However, most features like field selection, record selection and sort by have been designed to be understandable by anyone.

An example of the query definition interface is shown below:



Field: Object Type (char hex) (HEXSOBJYP)

Operator: = equal

Value

Example: 'QSYS'

Boolean condition: ● AND ○ OR

Record Selection Filter List:

Field	Operator	Value	And/Or
Object Type (char hex) (HEXSOBJYP)	=	'0203'	AND

Limitations

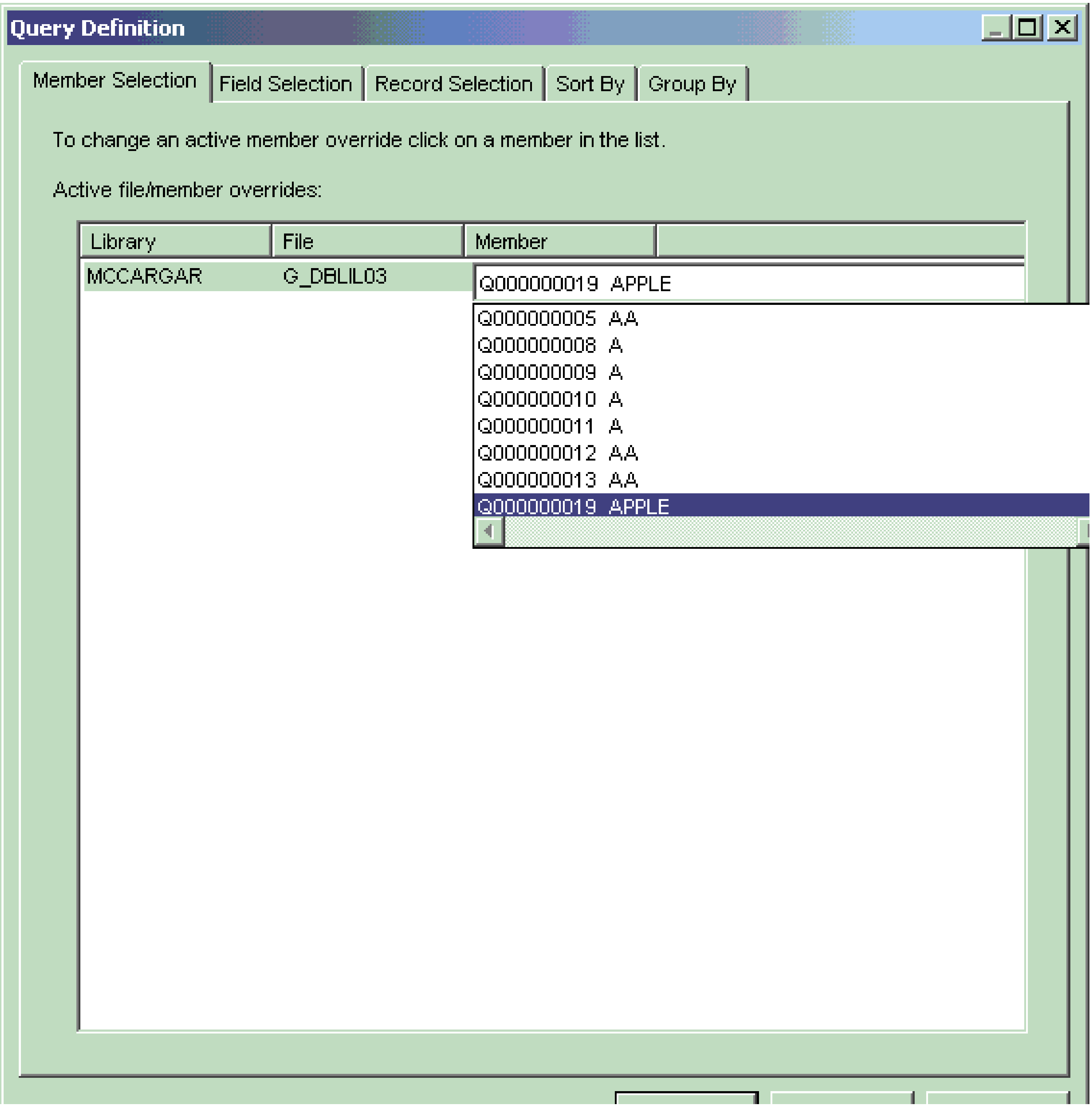
The Query Definition interface is built by parsing the contents of an SQL statement. This parsing works well for many queries but it does not acknowledge all types of SQL syntax. It will parse most SQL select statements containing "joins" but there are some very complex statements that are not parseable (such as UNIONS). Although a query can be parsed that contains joins the types of joins, and the files being joined are not changeable through the interface. In order to do this through the iDoctor for iSeries GUI, the user should use the SQL Query view and type in the SQL statement as needed. The query definition can be used to adjust the where clause, order by and group by clauses of the outermost part of the SQL statement.

Any order by clauses , where clauses, or group by clauses for subqueries within the SQL statement are not configurable through this interface.

Table of Contents Previous Next

1.6.1 Member Selection

Use the member selection page to select the member you want to work with from any of the files referenced in the query. To change the member for any file simply click on any member and a drop down list will appear. The drop down list shows the name and description for all members in the library/file.



To change an active member override click on a member in the list.

Active file/member overrides:

Library	File	Member
MCCARGAR	G_DBLIL03	Q000000019 APPLE

```
Q000000005  AA
Q000000008  A
Q000000009  A
Q000000010  A
Q000000011  A
Q000000012  AA
Q000000013  AA
Q000000019  APPLE
```

OK
Cancel
Help

Table of Contents | Previous | Next

1.6.2 Field Selection

The field selection panel allows you to hide or reorder the fields in the associated table view. You may also use this panel to create your own fields by typing an SQL expression directly into the SQL expression column in the field list.

The following types of operations can be performed on the field selection page:

- Working with field visibility
- Reordering fields
- Creating new fields

Instructions for performing each of these types of operations follows:

Working with field visibility

Visible fields are indicated by a checkmark in the Show? column within the Field List. If a field is not checked, then it will not be shown.

You may use the Toggle Selected button to check/uncheck the checkbox for the selected fields. This can be very handy when you want to hide or show a large number of fields at once.

Reording fields

The order that the fields are displayed in the Field List, directly effects the order that the fields are displayed in the table view.

To reorder fields:

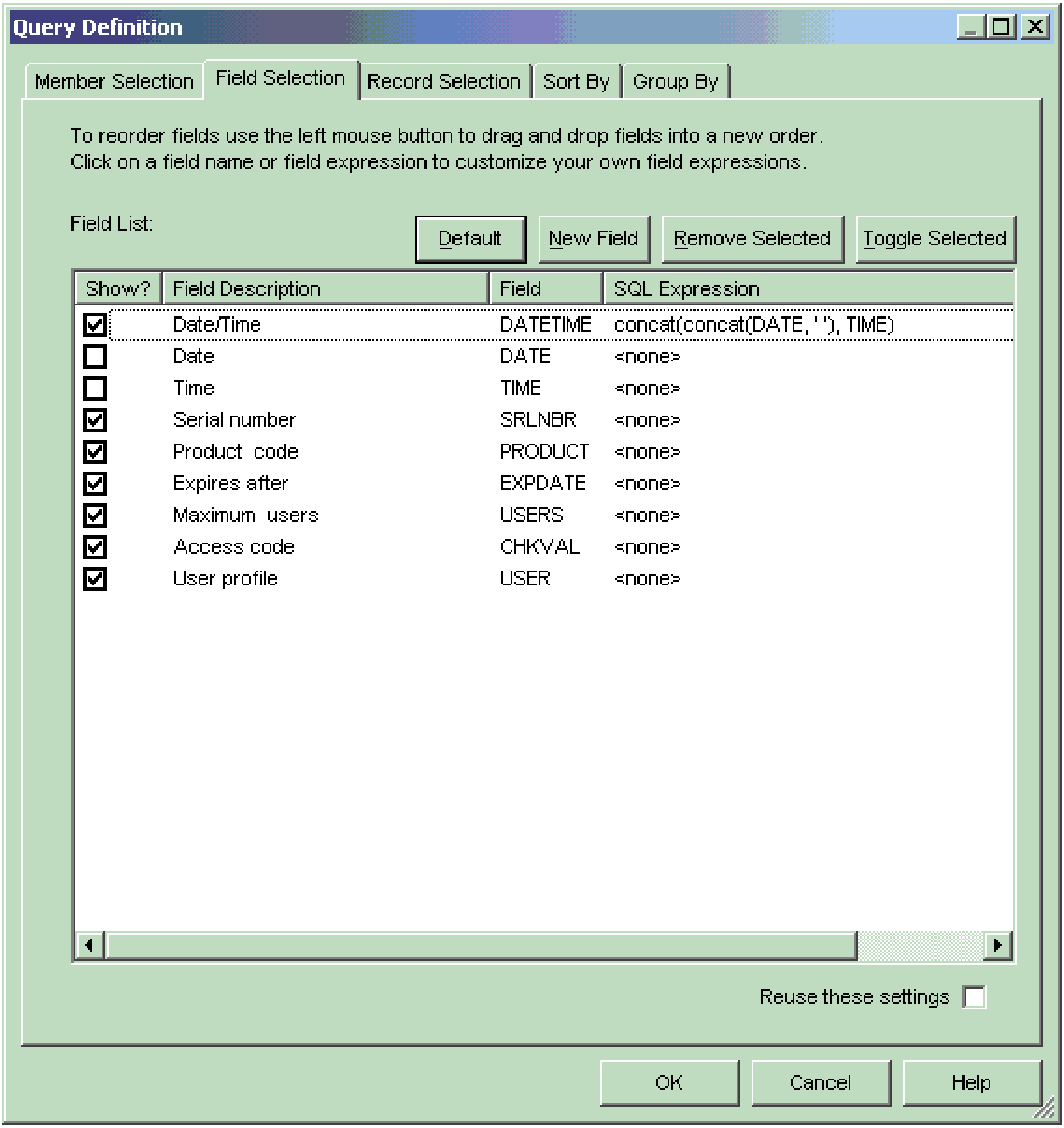
1. Select the fields you want to reorder using the mouse and ctrl/shift keys.
2. Press the left mouse button over one of the selected fields and hold it down.
3. Drag the selection to your desired position in the list. You can scroll through to the bottom of the list if desired.
4. Release the left mouse button.

Creating new fields

To create a new field:

1. Click the New Field button. After doing this a new row will be added to the list.
2. Click on the description cell for the new row in the list. Type a description for the new field.
3. Click the field cell in the list and provide a value. This value is the short SQL field name in your SQL field list. This value comes after the " AS " in the SQL expression **2 * count AS DBLCOUNT**
4. Click on the SQL expression cell in the list. Type in your SQL expression for this field. Do not include the " AS fieldname" part of the expression as this is done for you using the field value you already specified.

An example of the field selection page is the following:



To reorder fields use the left mouse button to drag and drop fields into a new order.
Click on a field name or field expression to customize your own field expressions.

Field List:

Show?	Field Description	Field	SQL Expression
☑	Date/Time	DATETIME	concat(concat(DATE, ' '), TIME)
□	Date	DATE	<none>
□	Time	TIME	<none>
☑	Serial number	SRLNBR	<none>
☑	Product code	PRODUCT	<none>
☑	Expires after	EXPDATE	<none>
☑	Maximum users	USERS	<none>
☑	Access code	CHKVAL	<none>
☑	User profile	USER	<none>

Reuse these settings □

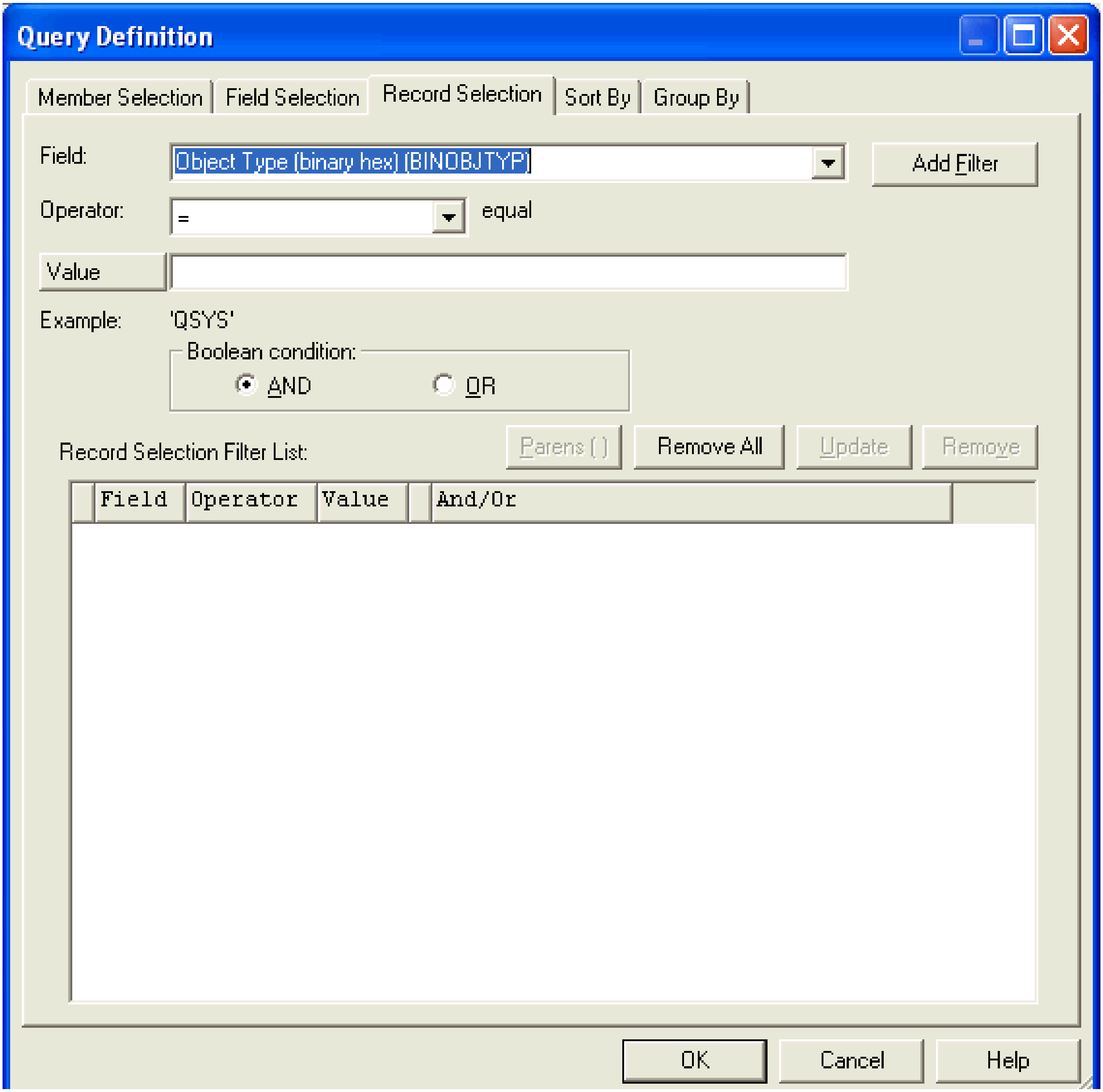
Note: The "reuse these settings" option at the bottom of the page is only enabled for PEX Analyzer report files. This allows the user to reorder the fields and this order saved onto the current PC (in the Windows Registry). The next time the analysis report is opened the same field ordering will be used.

Table of Contents | Previous | Next

1.6.3 Record Selection

The Record Selection Tab allows a user to limit the number of records or data returned in the active table or graph view. Any fields in the report may be filtered on using this interface.

An example of the Record Selection Page is shown below:



Query Definition

Field: Object Type (binary hex) (BINOBJTYP)

Operator: = equal

Value

Example: 'QSYS'

Boolean condition: ● AND ○ OR

Record Selection Filter List:

Field	Operator	Value	And/Or

To Add a Filter:

1. First select the field or type in its short name into the field drop down list.
2. Depending on the type of the field selected, various operators available to that field will be displayed in the operator list.
3. Select the desired operator from the operator list.
4. Type in the value that the operator should test for. For example, to specify only records where CPUTIME field is greater than 10 the operator selected would be > (greater than) and the value would be 10.
5. Press the Add Filter button to add the filter to the list.
6. Press the OK button to close this interface and run the query using the new filter.

By selecting more than one concurrent records in the list and pressing the 'Parens ( )' button the user can add or remove a set of parentheses. To remove parentheses around multiple filters, select the range of records that contain the starting and ending parentheses and click the 'Parens ( )' button. Parentheses are necessary in order to make complex evaluations in the where clause of an SQL statement such as:
CPUTIME >10 OR (IO > 1000 AND CPUTIME >= 1)

As the selection changes in the list, the interface objects above the list will change based on the current selection. This allows the user to quickly change values in the filter list by selecting any item in the list, changing any values from the fields above the list, and clicking the 'Update' button. The 'Update' button will update the selected row in the filter list.

A description of all the GUI elements on this panel follows:

GUI element name	**Description**
Field drop-down list	This is a list of every field in the current report. Select a field to filter by before clicking the 'Add Filter' or 'Update' buttons. The short name of a field may also be entered.
Operator list	This is a list of every operator available for the currently selected field. A text field has a different set of available operators than does a numeric field. The set of operators is also different for a timestamp field. The operators 'Field contains', 'Field starts with', 'Field ends with', 'Field xxx', etc are not valid for numeric and timestamp fields. The following operators are supported on this page: Equal Less than Less than or equal to Greater than Greater than or equal to Not equal Is null Is Not null Range List Not List

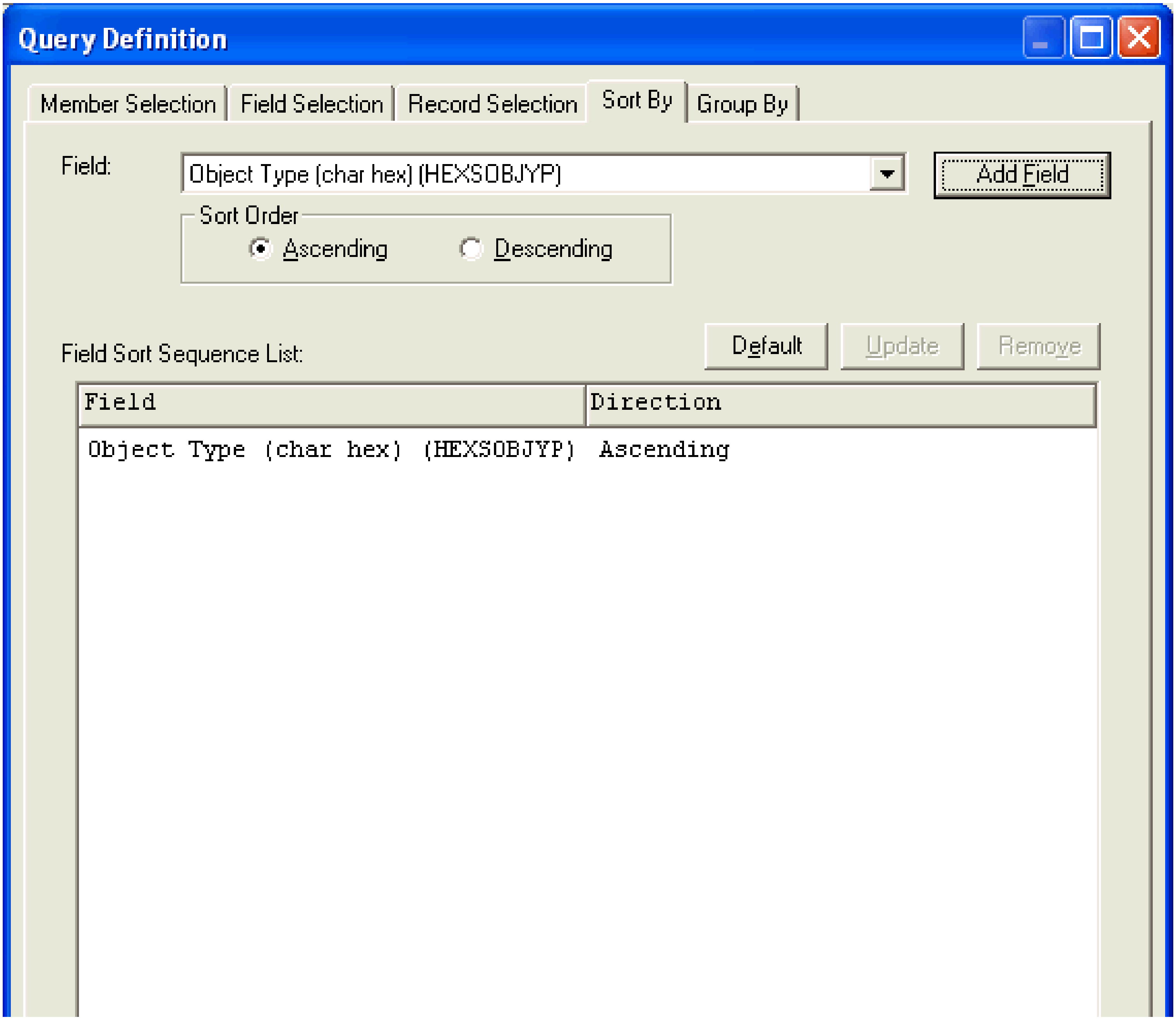
	Field contains Field starts with Field ends with Field does not contain Field does not start with Field does not end with
Value text box	Use this textbox to enter the value to apply to the current field using the selected operator. The value should match the format presented by the 'Example' label directly beneath the text box. Text fields should have their values enclosed in 'single quotation marks' and if the operator is 'Range', 'List' or 'Not list' then more than one values each separated by a space is expected. Whenever entering a value, follow the example provided.
Add Filter button	This button creates a new filter and adds the filter to the Record Selection Filter List.
Value/Expression button	This button allows the user to enter a valid SQL expression instead of a single value. This provides greater flexibility but requires that you know SQL syntax. Any errors in the SQL statement will prevent the query from running and will cause an SQL error message.
AND/OR options	Use this to indicate whether two filters should be ANDed together or OR'd together.
Parens ( ) button	The 'Parens ( )' button allows grouping of multiple filters in the Record Selection Filter List into a single logical expression by placing parentheses around the set of filters. If parentheses already exist for the starting and ending record in the selected range, the parenthese will be removed by pressing this button.
Remove All	This button will clear the list of filters.
Update button	This provides the ability to change the selected filter from the Record Selection Filter List.
Remove button	This button allows the user to remove one or more records from the Record Selection Filter List.
Record Selection Filter List	This is a list of all of the active filters to be applied to the report. Use the 'Add Filter' button to add a filter to the list. Press the OK button on the bottom of the Query Definition dialog to close the dialog and display the report using the filters from the list.

1.6.4 Sort By

The Sort By Page allows a user to change the order in which records are sorted in a table or graph view. This screen displays a list of fields to sort by and the sort direction for each field. The field at the top of the list has highest precedence in the sort sequence.

To add a field to the sort sequence list:

1. Select the field add to the list using the Field drop-down list.
2. Select the sort order: ascending or descending.
3. Press the Add Field button. The new field will be added to the Sort Sequence list.

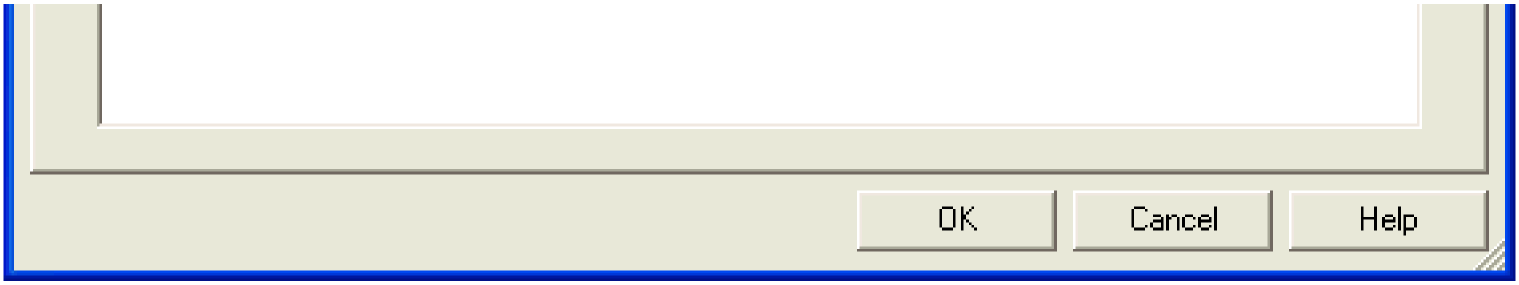


Field: Object Type (char hex) (HEXSOBJYP)

Sort Order
● Ascending ○ Descending

Field Sort Sequence List:

Field	Direction
Object Type (char hex) (HEXSOBJYP)	Ascending



The GUI elements on this page are described in the table below:

GUI element name	**Description**
Field drop-down list	This is a list of every field and its sort direction to use in the active view. The field at the top of the list has highest precedence in the sort sequence. Choose the field to add to the 'Field Sort Sequence' list before clicking the 'Add Field' or 'Update'
Add Field button	This button adds a field to the sort sequence indicated by the list, using the sort order currently specified.
Sort order options	Each field may be sorted in ascending (A-Z) or descending (Z-A) order. Choose the desired sort order before clicking 'Add Field' or 'Update'.
Default button	The default button changes the active sort order to whatever the default sort sequence is for the active report. In most cases, this will clear the sort sequence to nothing. In this case the sort order will be based on an ascending sort by relative record number of the raw data in the file.
Update button	The update button will change the sort sequence definition for the currently selected item in the list.
Remove button	This button allows the removal of one or more sort definitions from the list.
Field sort sequence list	This list represents the current sort order to apply to the active table or graph. Selecting any item in the list allows the option to change the value for the selected item using the 'Update' button.

Table of Contents | Previous | Next

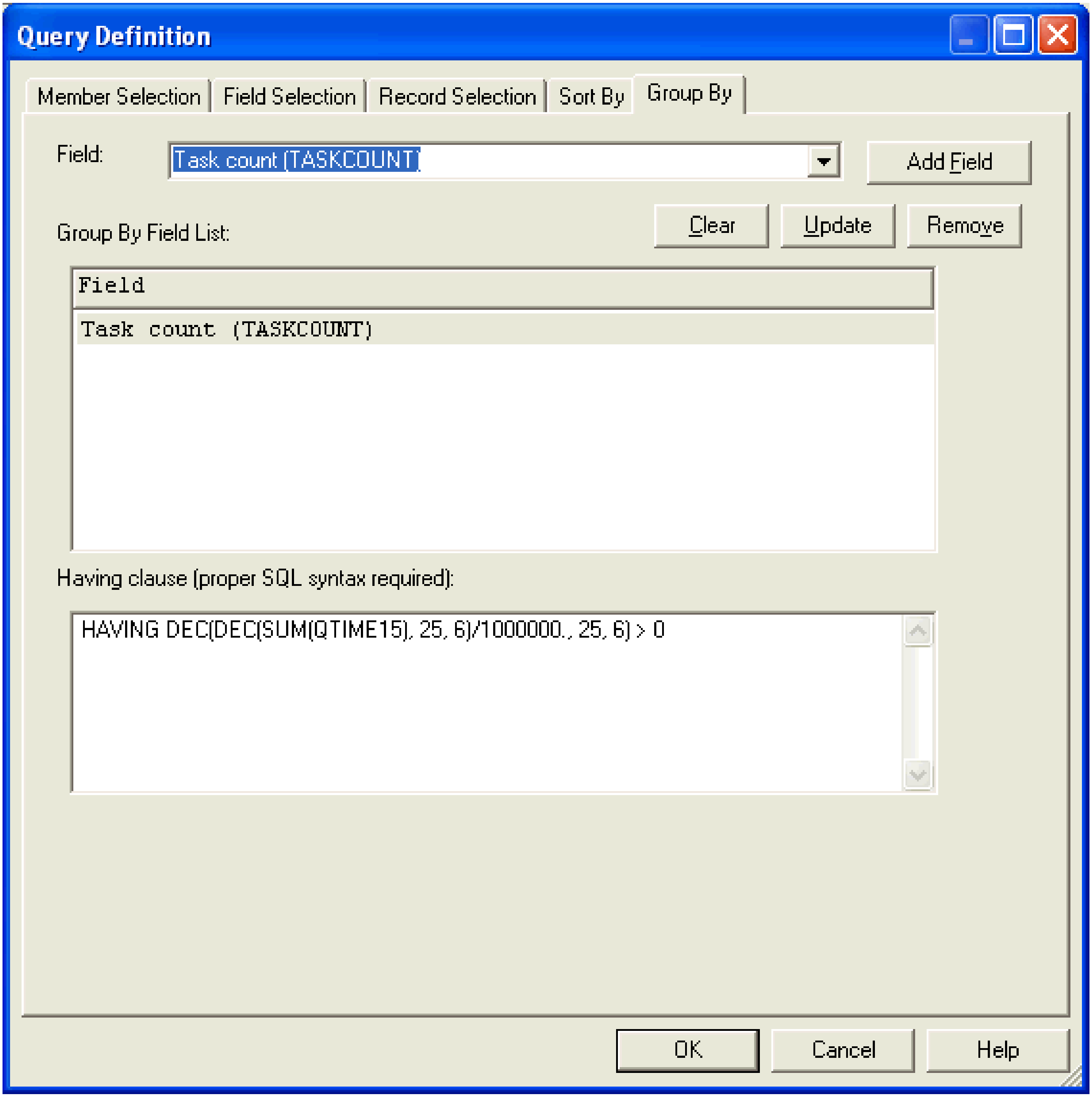
1.6.5 Group By

The Group By Page allows a user to define the fields that should be used as part of a Group by query. The field at the top of the list has highest precedence in the SQL GROUP BY clause.

Group by queries are only valid when the fields on the field selection page comply with the rules SQL has wth running group by SQL statements. Any fields that are not part of the group by clause must be summarized in order to exist in the field selection or the query will not run.

To add a field to the group by list:

1. Select the field to add to the list using the Field drop-down list.
2. Press the Add Field button. The new field will be added to the Group By Field List.



Field: Task count (TASKCOUNT)

Group By Field List:

Field
Task count (TASKCOUNT)

Having clause (proper SQL syntax required):

```
HAVING DEC(DEC(SUM(QTIME15), 25, 6)/1000000., 25, 6) > 0
```

The GUI elements on this page are described in the table below:

GUI element name	Description
Field drop-down list	This is a list of every field in the active view. Choose the field to add to the 'Group By Field' list before clicking the 'Add Field' or 'Update' button

Add Field button	This button adds a field to the group by list.
Clear button	Removes all fields from the list.
Update button	Use the update button to change the selected field in the list to match the selected field in the drop-down list.
Remove button	The remove button will delete all selected fields from the Group By Field List.
Group By Field List	This list represents the GROUP BY clause in the Group By query. Selecting any item in the list allows the option to change the value for the selected item using the 'Update' button.
Having clause	This is the exact syntax to use for the Having clause for the group by query. Speciyfing a Having clause is not required.

Table of Contents | Previous | Next

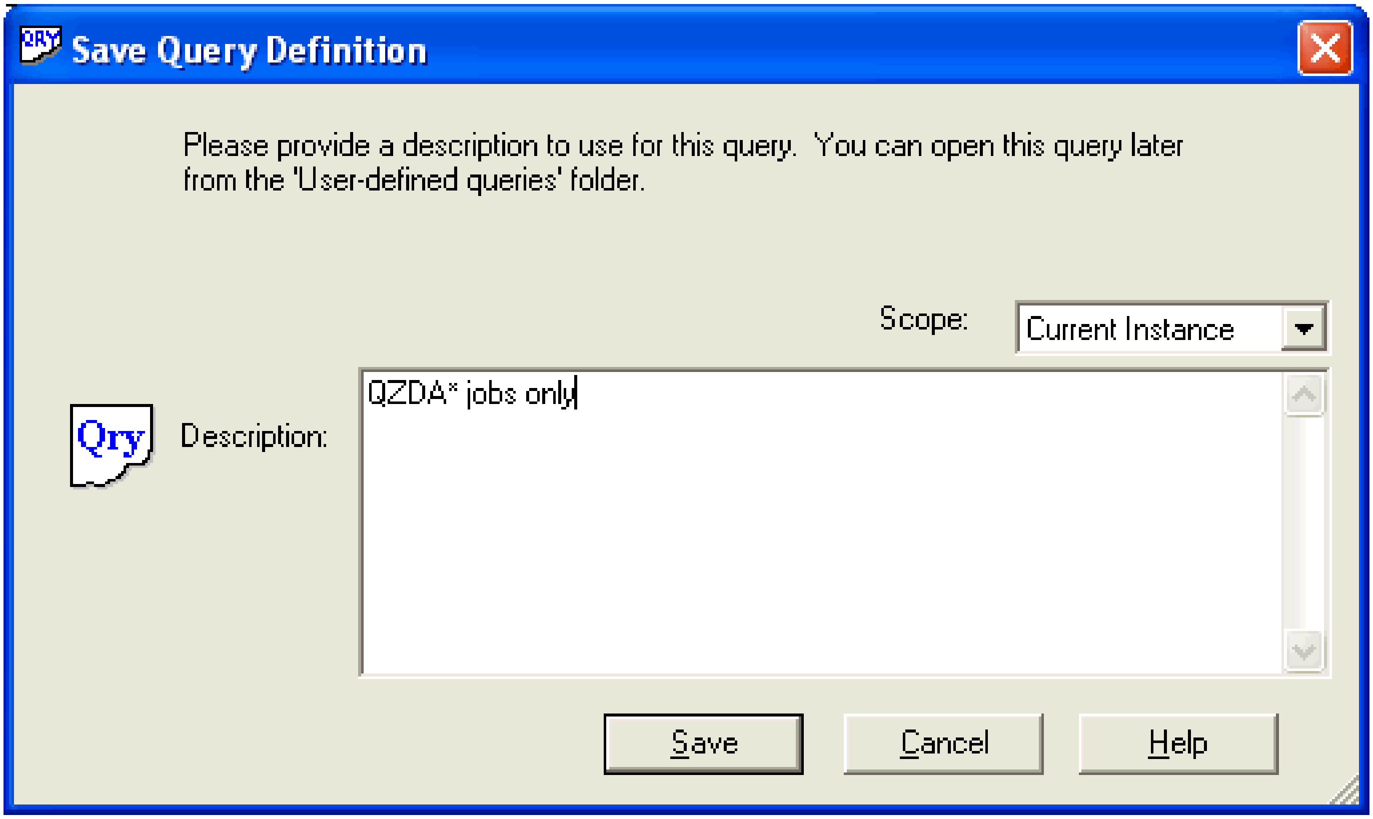
1.6.6 Resetting

The reset submenu under the query definition popup menu may be used to reset a query back to its original state. Whenever a table or graph view is loaded the initial SQL statement is saved. If at some point it is desired to to discard the changes made to the report, use the Reset menu.

1.6.7 Saving

Query Definitions are saved using the Query Definition -> Save As... menu for an active table view. The query definition behind a graph view is saved automatically using the Graph Definition -> Save As... menu for a graph view. All Query Definitions are saved into the file QAIDRSQL04. The library where the query is saved into depends on the scope of the query (explained below).

An example of the Save Query Definition interface is shown below:



Save Query Definition

Please provide a description to use for this query. You can open this query later from the 'User-defined queries' folder.

Scope: Current Instance

Description: QZDA* jobs only

The interface objects within this window are described in more detail below:

Filter	Description
Replace existing query definition option	Select this box to replace the saved query definition with the one currently being used. This checkbox is only visible if the table view was created from a user-defined query definition.
Description	The user-defined description for the definition. This description can be up to 250 characters long.

Scope	Use this option to set the scope of the query. This determines at which level (system, library, or current instance) the query should be visible. If the scope is library or current instance then the file QAIDRSQL04 in the same library the current collection is in will contain the query definition. System scoped query definitions are stored in library QUSRSYS. System scoped queries can be used against any libraries on the system that have the same file(s) that the current query uses. Query definitions can be opened from the user-defined queries folder underneath a collection.

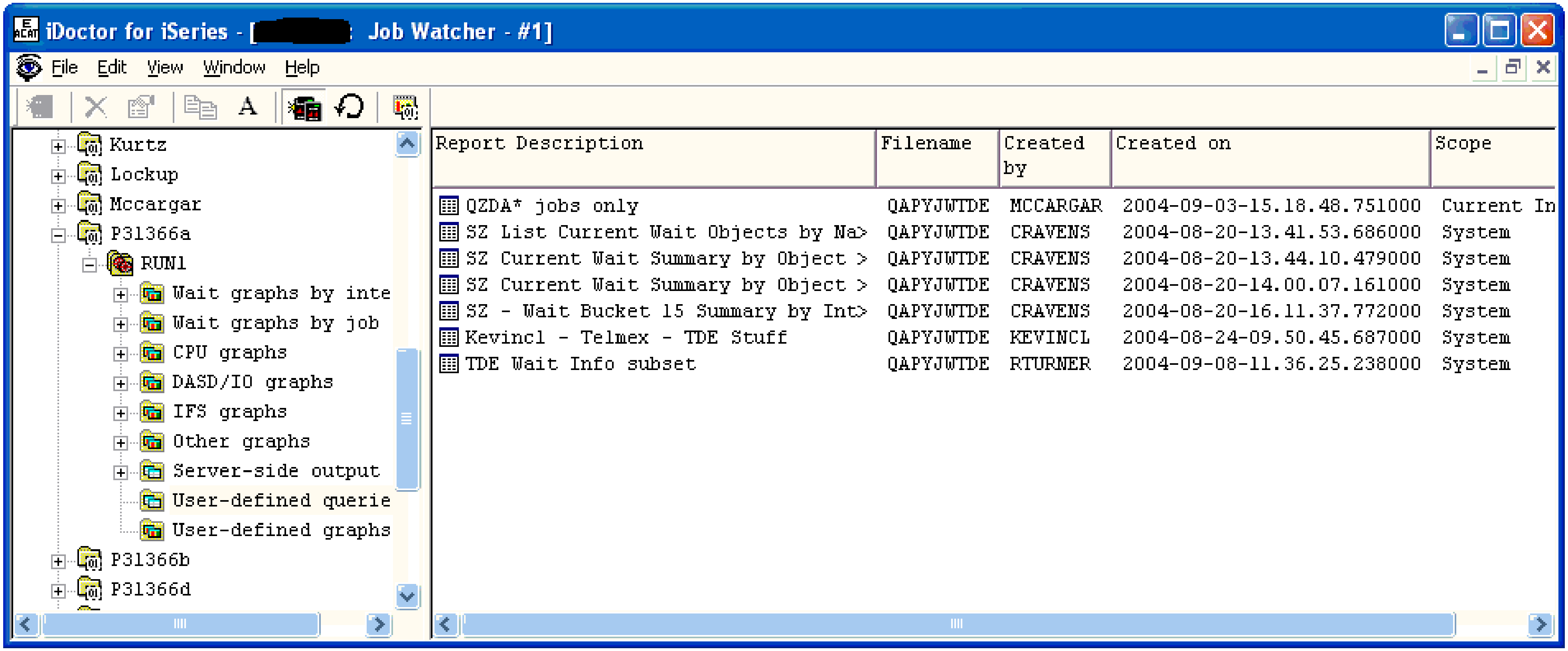
Table of Contents | Previous | Next

1.6.8 Working with Query Definitions

After a query definition has been created it will be displayed in the user-defined queries folder within a collection in either PEX Analyzer, Job Watcher or Data Explorer.

This folder lists the query definitions available for the current collection. Only queries that were saved over the same type of files as the current selection will be visible. For example, PEX Analyzer queries are not visible when using Job Watcher within the user-defined queries folder.

An example of the user-defined queries folder is:



Report Description	Filename	Created by	Created on	Scope
QZDA* jobs only	QAPYJWTDE	MCCARGAR	2004-09-03-15.18.48.751000	Current In
SZ List Current Wait Objects by Na>	QAPYJWTDE	CRAVENS	2004-08-20-13.41.53.686000	System
SZ Current Wait Summary by Object >	QAPYJWTDE	CRAVENS	2004-08-20-13.44.10.479000	System
SZ Current Wait Summary by Object >	QAPYJWTDE	CRAVENS	2004-08-20-14.00.07.161000	System
SZ - Wait Bucket 15 Summary by Int>	QAPYJWTDE	CRAVENS	2004-08-20-16.11.37.772000	System
Kevincl - Telmex - TDE Stuff	QAPYJWTDE	KEVINCL	2004-08-24-09.50.45.687000	System
TDE Wait Info subset	QAPYJWTDE	RTURNER	2004-09-08-11.36.25.238000	System

The fields shown in the user-defined queries folder are:

Report description - name of the query when it was last saved.
Filename - the primary file (or first file) in the SQL statement
Created by - user profile that last saved this query definition
Created on - the date and time when the query definition was last changed or created
Scope - describes the visibility of the query definition
Current instance - the query definition is only visible for the library and collection or analysis for which it was created
Library - the query definition is visible for all collections/analyses in the same library of the same type for which the query was created

System - the query definition is visible for all collections/analyses in all libraries of the same type for which the query was created

The menu options available for a query definition are:

Menu	**Description**
Open Table	Opens the selected query definition as a table view in a new or existing data viewer.
Add to Playlist...	Not yet implemented.
Delete...	Removes the selected query definition(s) from the system.
Set Scope	Modifies the scope of the selected query definition to the specified type.

Table of Contents Previous Next

1.7 Graph Definitions

In iDoctor for iSeries, users can define graphs over data within any table view or graph view desired. This is done by creating a graph definition which is the information that builds a user-defined or iDoctor-supplied graph. Like query definitions, graph definitions are stored on the server in a database file.

A graph definition defines everything needed to display the graph including a reference to the query definition needed to produce the graph's data. Whenever a graph view is saved, the current query definition as well as the current graph definition is saved.

The interface is accessible by either defining a new graph definition using the Graph Definition | Define New... popup menu within a table view or by using the Graph Definition popup menu in a user-defined graph view.

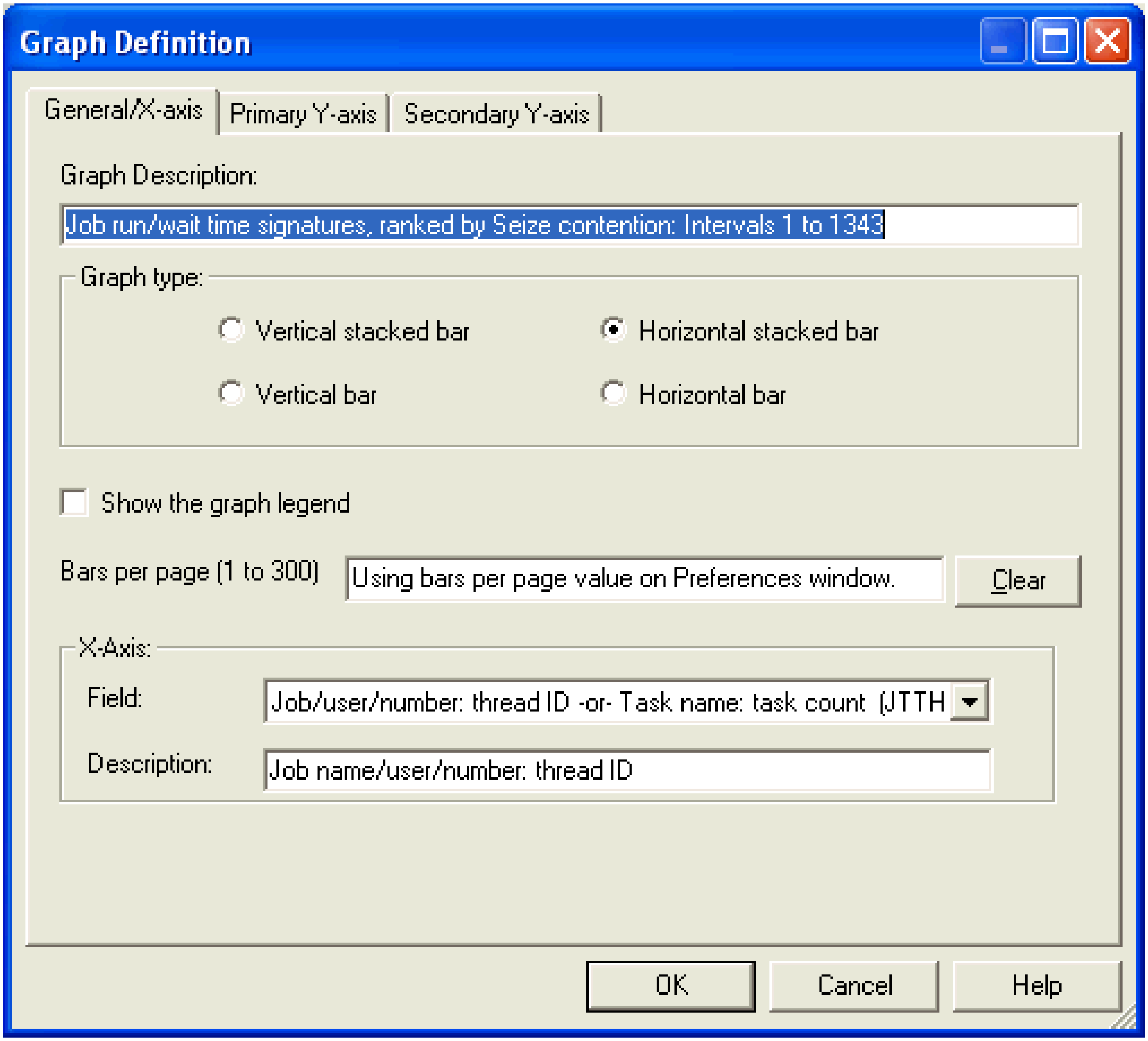
There are a sets of pages which let the user define the graph definition. These pages are discussed in greater detail in the next sections. A summary of the pages that make up a graph definition is shown below:

Page Name	**Description**
General/X-Axis	Defines the general features of the graph, like the type of graph, as well as information about the X-Axis.
Primary Y-Axis	This page defines the fields, colors and desciptions to use for the bars in the graph. Up to 32 different fields/colors may be defined in the graph definition.
Secondary Y-Axis	This page identifies the secondary Y-axis. This axis is always a single black line which overlays the rest of the graph. The secondary Y-axis is optional.

1.7.1 General Page

The general page lets the user define the graph title, type, bars per page as well as the field used for the X-axis.

A example of the General Page is shown below:



Graph Description: Job run/wait time signatures, ranked by Seize contention: Intervals 1 to 1343

Graph type:

○ Vertical stacked bar ● Horizontal stacked bar

○ Vertical bar ○ Horizontal bar

□ Show the graph legend

Bars per page (1 to 300) Using bars per page value on Preferences window.

X-Axis:

Field: Job/user/number: thread ID -or- Task name: task count (JTTH

Description: Job name/user/number: thread ID

The GUI elements on the General Page are described in detail within the table below:

Element name	**Description**
Graph Description	A description of the graph. (50 characters max)
Graph type	Indicates the layout of the graph. Vertical bar graph will produce a graph with side-by-side vertical bars. Horizontal bar graph will produce a graph with side-by-side horizontal bars.
Show the graph legend	This checkbox indicates whether the graph legend should be displayed. On some graphs with many colors/graphs this option is turned off by default in order to allocate more screen space to the data itself.
Bars per page	This value can be used to optionally specify the number of bars to show per page on this graph. If a value is not specified on this page then the applicable bars per page value on the Preferences window will be used instead.
X-Axis Field	The field information to use when for the X-Axis on the graph. Typically this is something like Interval number or Job name. This can be a numeric field or a character field and is a required field.
X-Axis Description	The description to display under the X-Axis on the graph. This value defaults to the description of the field, but it can be changed to something else if desired.

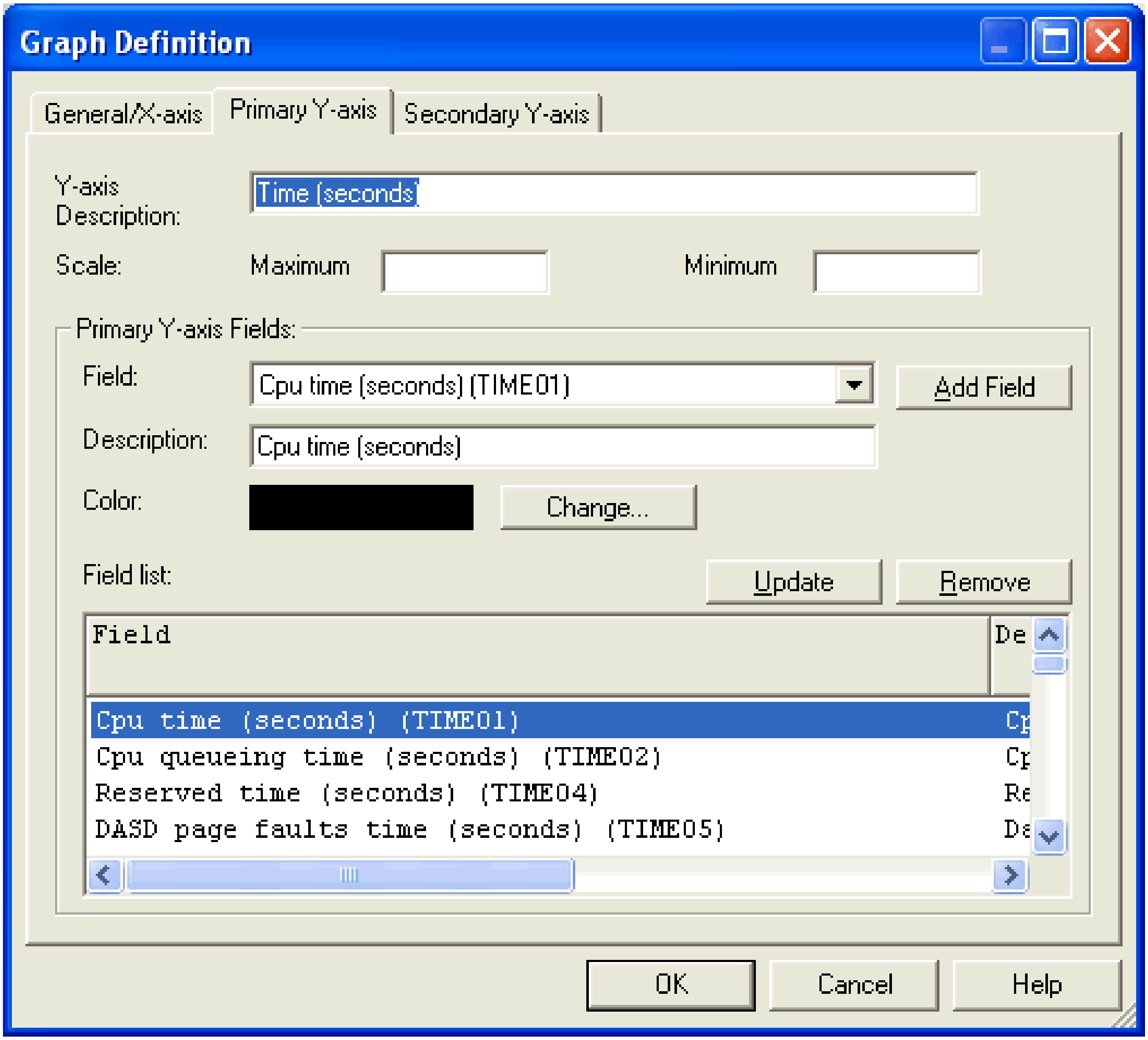
Table of Contents | Previous | Next

1.7.2 Primary Y-axis

Use the Primary Y-axis page to define the fields that should be displayed on the graph. Each field represents a bar on the graph and can have a different color and customized description.

To Add a Field.

1. Select the field you want to use for the new field from the Field drop-down list.
2. If desired, modify the description of the field from the default field description.
3. If desired, define a color for this Y-Axis field. If this is not done, a color will be automatically assigned.
4. Click the Add Field button to the add the field to the list of fields.



Y-axis Description: Time (seconds)

Scale: Maximum Minimum

Primary Y-axis Fields:

Field: Cpu time (seconds) (TIME01)

Description: Cpu time (seconds)

Color: ■

Field list:

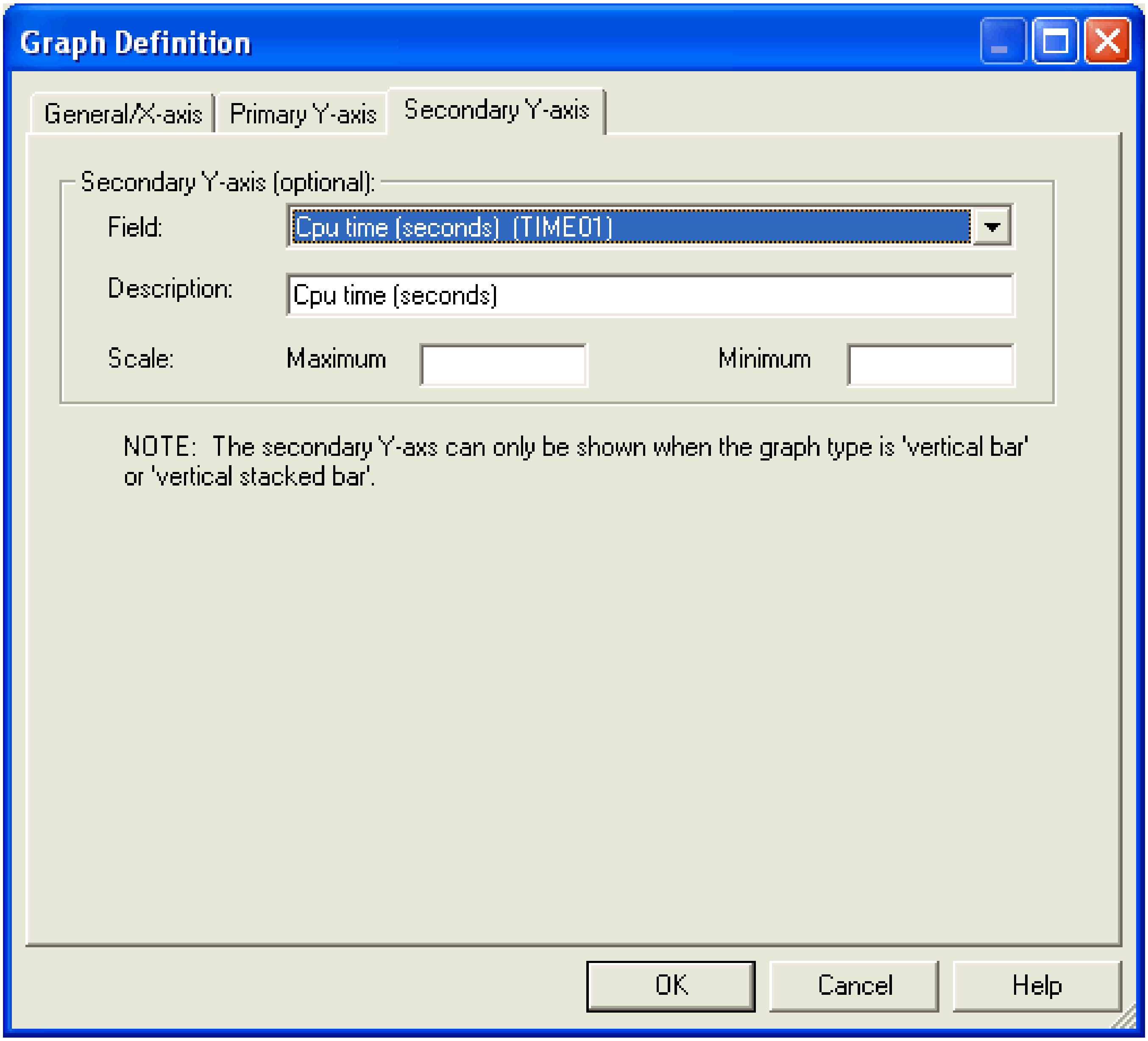
```
Field                                        De
Cpu time (seconds) (TIME01)                  Cp
Cpu queueing time (seconds) (TIME02)         Cp
Reserved time (seconds) (TIME04)             Re
DASD page faults time (seconds) (TIME05)     Da
```

The GUI elements on this page are described in detail within the table below:

Element name	**Description**
Y-axis Description	A description of the primary Y-axis. (50 characters max)
Scale	Allows the primary Y-axis scale to be set to a specific maximum and/or minimum value. If any numbers are greater or less than the values specified the bar will be truncated. Using this feature is optional.
Field	Allows selection of a field to add to the field list. Changing the field will update the description to match the field description for the selected field.
Description	The 50 character description that identifies the data in the graph. This description will be displayed in the graph's legend.
Color	The color to use for the field selected. If no color is selected a unique color will be provided automatically.
Update	The update button is used to modify the selected field in the field list. For example this option could be used to change the color of an existing field in the field list.
Remove	This option will remove the selected fiels from the field list.
Field list	Displays the field names, descriptions and colors to use for the bars in the bar graph.

1.7.3 Secondary Y-axis

This page allows the user to define a secondary Y-axis on a bar graph. The secondary Y-axis is always a black line over the bar graph with a Y axis on the right side of the graph. This axis is only visible for horizontal bar graphs.



Secondary Y-axis (optional):

Field: Cpu time (seconds) (TIME01)

Description: Cpu time (seconds)

Scale: Maximum Minimum

NOTE: The secondary Y-axs can only be shown when the graph type is 'vertical bar' or 'vertical stacked bar'.

The GUI elements on this page are described in detail within the table below:

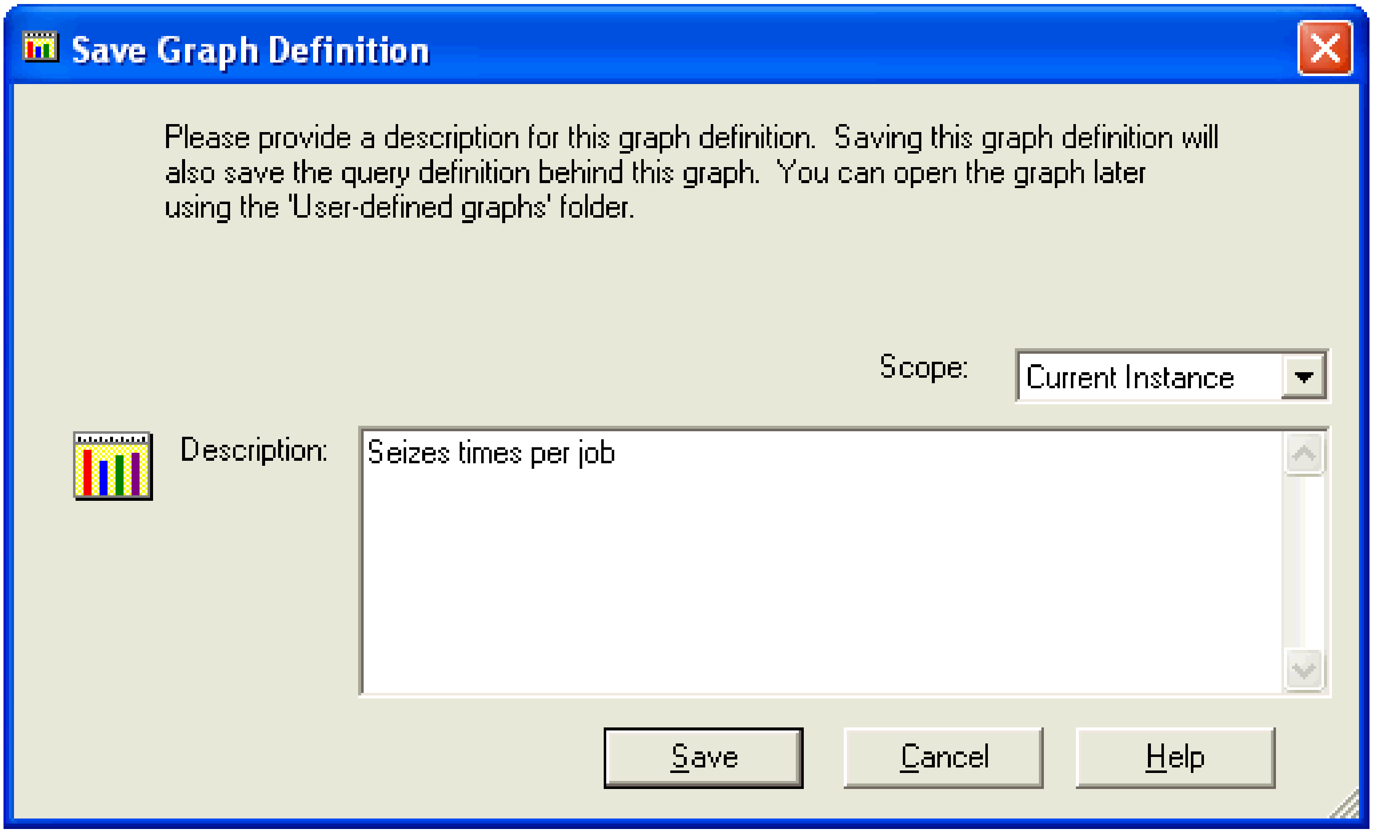
Element name	**Description**
Field	The name of the field to use for the secondary Y-axis. This value should be blank (first selection in the list) if no secondary Y-axis is desired.
Description	A description of the seconary Y-axis. (50 characters max)
Scale	Allows the secondary Y-axis scale to be set to a specific maximum and/or minimum value. If any numbers are greater or less than the values specified they will not be visible. This feature is optional.

Table of Contents | Previous | Next

1.7.4 Saving

Graph Definitions are saved using the Graph Definition -> Save As... menu for the active graph view. All Graph Definitions are saved into the file QAIDRGPH08. The library where the graph is saved into depends on the scope of the graph.

An example of the Save Graph Definition interface is shown below:



Save Graph Definition

Please provide a description for this graph definition. Saving this graph definition will also save the query definition behind this graph. You can open the graph later using the 'User-defined graphs' folder.

Scope: Current Instance

Description: Seizes times per job

The interface elements within this window are described in more detail below:

Interface element	Description
Replace existing graph definition option	Select this box to replace the saved graph definition with the one currently being used. This checkbox is only visible if the graph view was created from a user-defined graph definition.
Description	The description for the Graph Definition. This description can be up to 250 characters long.

Scope	Use this option to set the scope of the graph. This determines at which level (system, library, or current instance) the graph should be visible. If the graph scope is library or current instance then the file QAIDRGPH08 in the same library the current collection is in will contain the graph definition. System scoped graph definitions are stored in library QUSRSYS. System scoped graphs can be used against any libraries on the system that have the same underlining file(s) that the current graph uses. Graph definitions can be opened from the user-defined graphs folder underneath a collection.

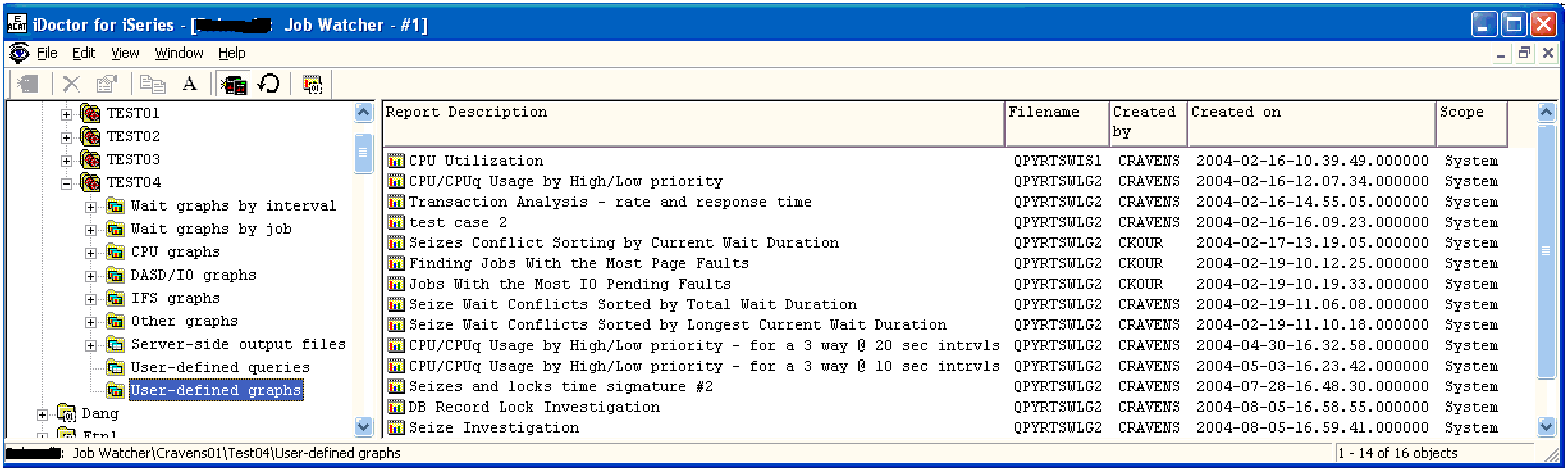
Table of Contents | Previous | Next

1.7.5 Working with Graph Definitions

After a graph definition has been created it will be displayed in the user-defined graphs folder within a collection in either PEX Analyzer, Job Watcher or Data Explorer.

This folder lists the graph definitions available for the current collection. Only graphs that were saved over the same type of files as the current selection will be visible. For example, PEX Analyzer graphs are not visible when using Job Watcher within the user-defined graphs folder.

An example of the user-defined graphs folder is:



- TEST01
- TEST02
- TEST03
- TEST04
  - Wait graphs by interval
  - Wait graphs by job
  - CPU graphs
  - DASD/IO graphs
  - IFS graphs
  - Other graphs
  - Server-side output files
  - User-defined queries
  - User-defined graphs
- Dang

Report Description	Filename	Created by	Created on	Scope
CPU Utilization	QPYRTSWIS1	CRAVENS	2004-02-16-10.39.49.000000	System
CPU/CPUq Usage by High/Low priority	QPYRTSWLG2	CRAVENS	2004-02-16-12.07.34.000000	System
Transaction Analysis - rate and response time	QPYRTSWLG2	CRAVENS	2004-02-16-14.55.05.000000	System
test case 2	QPYRTSWLG2	CRAVENS	2004-02-16-16.09.23.000000	System
Seizes Conflict Sorting by Current Wait Duration	QPYRTSWLG2	CKOUR	2004-02-17-13.19.05.000000	System
Finding Jobs With the Most Page Faults	QPYRTSWLG2	CKOUR	2004-02-19-10.12.25.000000	System
Jobs With the Most IO Pending Faults	QPYRTSWLG2	CKOUR	2004-02-19-10.19.33.000000	System
Seize Wait Conflicts Sorted by Total Wait Duration	QPYRTSWLG2	CRAVENS	2004-02-19-11.06.08.000000	System
Seize Wait Conflicts Sorted by Longest Current Wait Duration	QPYRTSWLG2	CRAVENS	2004-02-19-11.10.18.000000	System
CPU/CPUq Usage by High/Low priority - for a 3 way @ 20 sec intrvls	QPYRTSWLG2	CRAVENS	2004-04-30-16.32.58.000000	System
CPU/CPUq Usage by High/Low priority - for a 3 way @ 10 sec intrvls	QPYRTSWLG2	CRAVENS	2004-05-03-16.23.42.000000	System
Seizes and locks time signature #2	QPYRTSWLG2	CRAVENS	2004-07-28-16.48.30.000000	System
DB Record Lock Investigation	QPYRTSWLG2	CRAVENS	2004-08-05-16.58.55.000000	System
Seize Investigation	QPYRTSWLG2	CRAVENS	2004-08-05-16.59.41.000000	System

[redacted]: Job Watcher\Cravens01\Test04\User-defined graphs

1 - 14 of 16 objects

The fields shown in the user-defined graphs folder are:

Report description - name of the graph when it was last saved.
Filename - the primary file (or first file) for the SQL statement used behind the graph
Created by - user profile that last saved this graph definition
Created on - the date and time when the graph definition was last changed or created
Scope - describes the visibility of the graph definition
Current instance - the graph definition is only visible for the library and collection or analysis for which it was created
Library - the graph definition is visible for all collections/analyses in the same library of the same type for which the graph was created
System - the graph definition is visible for all collections/analyses in all libraries of the same type for which the graph was created

The menu options available for a graph definition are:

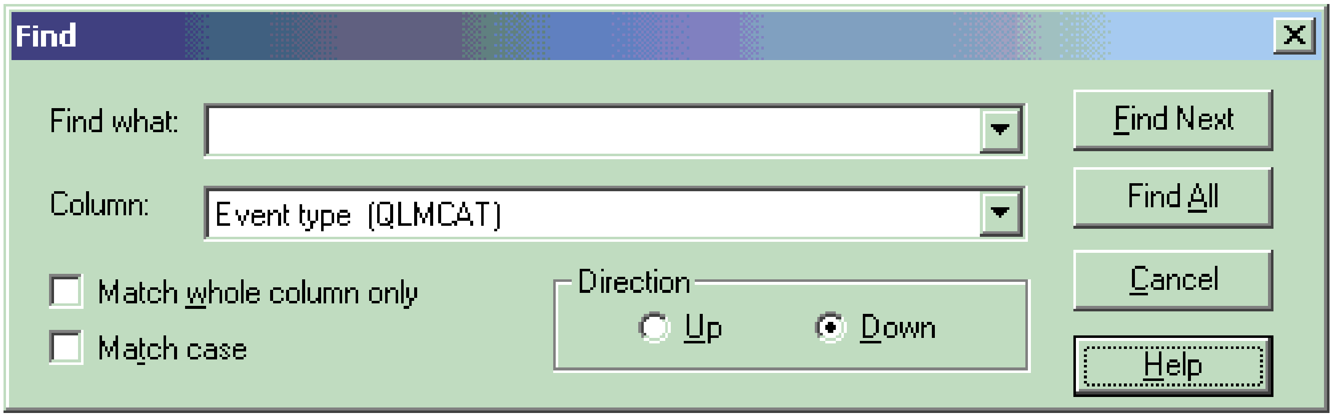
Menu	Description
Open Graph	Opens the selected graph definition as a graph view in a new or existing data viewer.
Add to Playlist...	Not yet implemented.
Delete...	Removes the selected graph definition(s) from the system.
Set Scope	Modifies the scope of the selected graph definition to the specified type.

Table of Contents | Previous | Next

1.8 Find Window

The Find Window allows a user to perform a search over a column in a Table View. Use the Edit | Find... menu or right-click on a Table View and choose the Find... menu to use the Find Window. Find allows the user to search for a text string within a specfic column.

An example of the find window is shown below:



Find what:

Column: Event type (QLMCAT)

□ Match whole column only

□ Match case

Direction: ○ Up ● Down

The following table summarizes the interface elements on the Find window.

Interface Element	Field Description
Find what textbox	Enter the string you would like to search on in this textbox.
Column drop down list	This list contains the name of every column in the active Table View. Select the column you would like to search on from this list.
Find Next button	Clicking this button will perform a search over the active Table View for the next occurence (depending on direction up or down) of the string in the Find what textbox in the specified column.
Find All button	Click this button to find and select all matching occurrences of the Find what textbox in the specified column.
Match whole column only	Select this to indicate that records should only match if the value in the find what text box matches exactly with the searched on column value.
Match case	Select this to perform a case-sensitive search.
Direction (Up/Down)	These fields indicate whether the search should be performed over the remaining data in the Table View or the preceding data.

Table of Contents Previous Next

1.9 Preferences

The Preferences window allows a user to work with the customizable options in the iDoctor for iSeries client. Several different categories of options are available and each category is presented on a different page.

The Preferences window is accessible via the Edit | Preferences menu in the Data Viewer or from the iDoctor for Series Main Window. See the next sections for information on each page in the Preferences window.

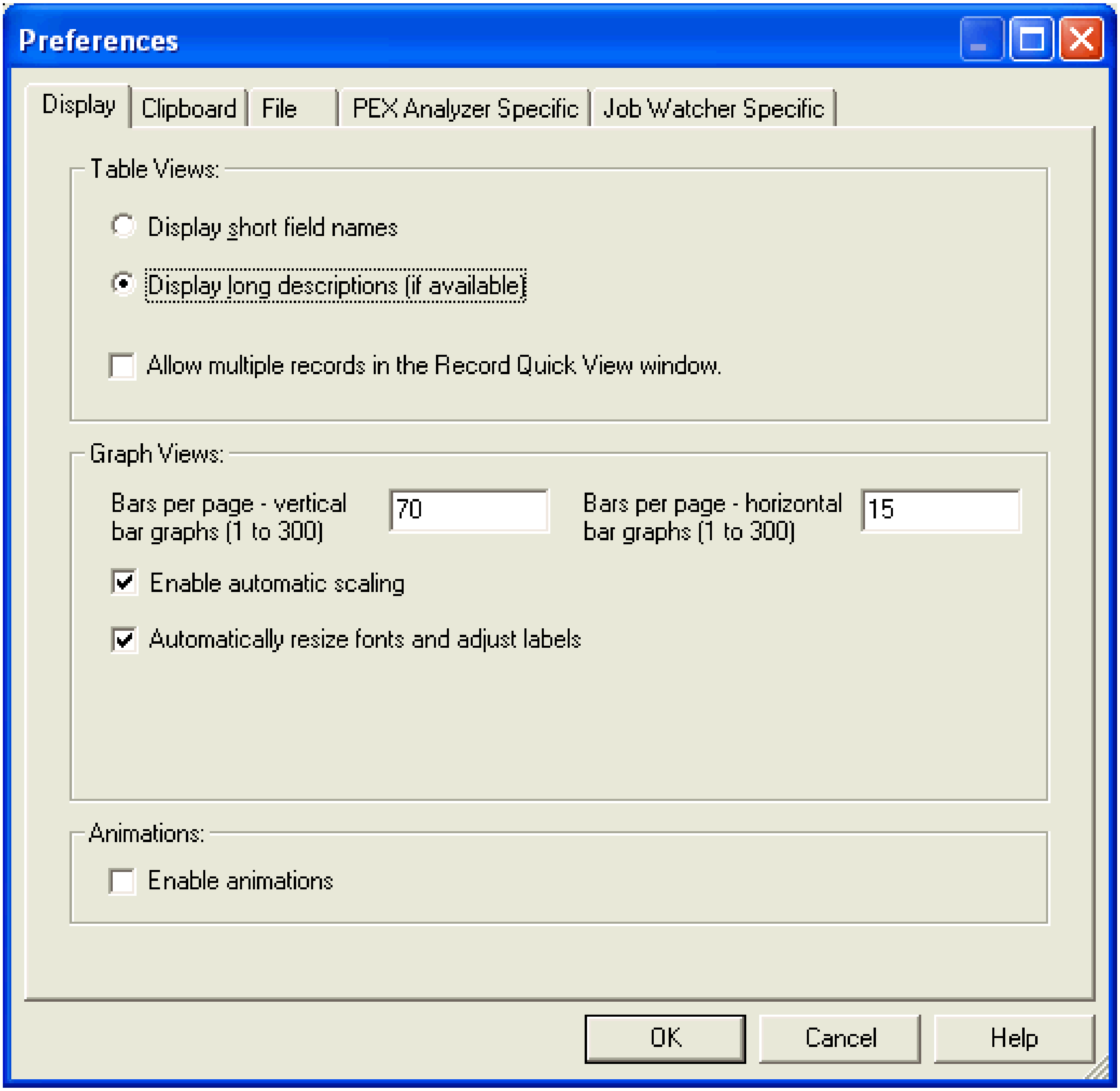


Table Views:

○ Display short field names

● Display long descriptions (if available)

□ Allow multiple records in the Record Quick View window.

Graph Views:

Bars per page - vertical bar graphs (1 to 300) 70

Bars per page - horizontal bar graphs (1 to 300) 15

☑ Enable automatic scaling

☑ Automatically resize fonts and adjust labels

Animations:

□ Enable animations

Table of Contents | Previous | Next

1.9.1 Display

The Display page on the Preferences window lets the user work with options that effect the visible presentation of table or graph views in the iDoctor for iSeries client.

An example of this interface is shown below:

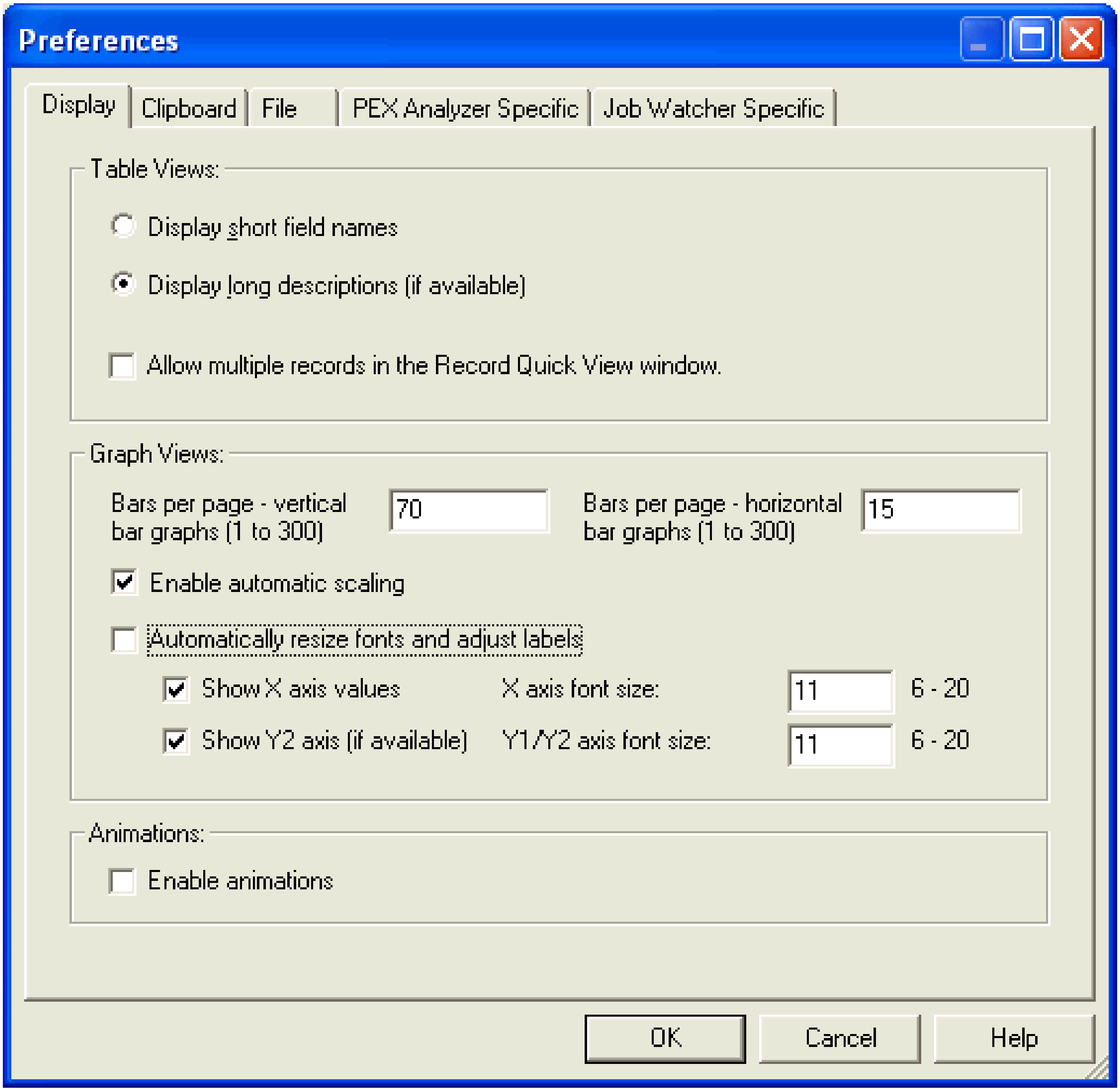


Table Views:

○ Display short field names

● Display long descriptions (if available)

☐ Allow multiple records in the Record Quick View window.

Graph Views:

Bars per page - vertical bar graphs (1 to 300): 70

Bars per page - horizontal bar graphs (1 to 300): 15

☑ Enable automatic scaling

☐ Automatically resize fonts and adjust labels

☑ Show X axis values — X axis font size: 11 — 6 - 20

☑ Show Y2 axis (if available) — Y1/Y2 axis font size: 11 — 6 - 20

Animations:

☐ Enable animations

The options available on this page are summarized in the tables below:

Table View Options	**Description**
Table View Display short or long name option	This option lets the user decide if long field descriptions should be displayed or short field names (10 characters or less) in the column headings of all table views. **Note:** Short field names will be displayed if the long descriptions are not available.
Allow multiple records in the Record Quick View window	This option allows the user to select multiple records in a table and use the record quick view option to compare multiple records side by side.

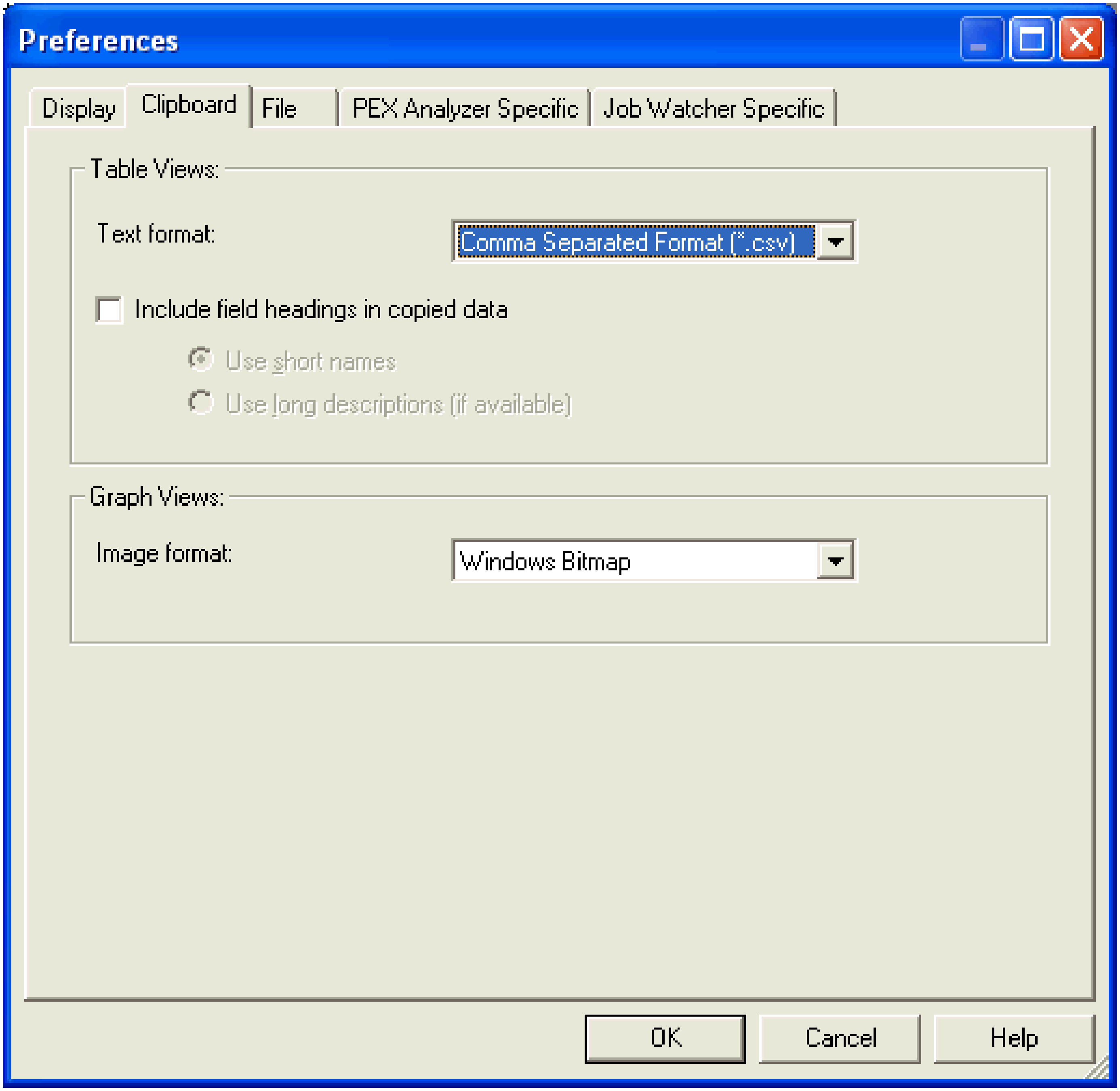
Graph View Options	**Description**
Bars per page - vertical	Indicates how many bars should be displayed per page in a vertical bar, or stacked vertical bar graph.
Bars Per Page - horizontal	Indicates how many bars should be displayed per page in a horizontal bar, or stacked horizontal bar graph.
Enable automatic scaling	Indicates if the graph should automaticly resize the scale on the Y axis each time the current position in the graph changes. If this option is turned off the scale will be fixed based on the maximum and minimum values of the first page of the graph when it is opened.
Automatically resize fonts and adjust labels	This option controls whether the fonts and labels should be automatically resized and adjusted (recommended on).
Show X-Axis values	Indicates if labels for the X-Axis values should be displayed.
Show Y2-Axis (if available)	Indicates if the Y2-Axis (the secondary Y-Axis) should be displayed. This axis is not used on all graphs.
X-Axis font size	Indicates the font size to use for values on the X-Axis. The higher the number the larger the font will appear.
Y-Axis font size	Indicates the font size to use for values on the Y-Axis. The higher the number the larger the font will appear.

Animation Option	**Description**
Enable animations	Turns animations in the Wizards on or off.

1.9.2 Clipboard

The Clipboard page on the Preferences window lets the user work with the 'Copy to Clipboard' options available for table or graph views in the iDoctor for iSeries client.

An example of this interface is shown below:



The options available on this page are summarized in the tables below:

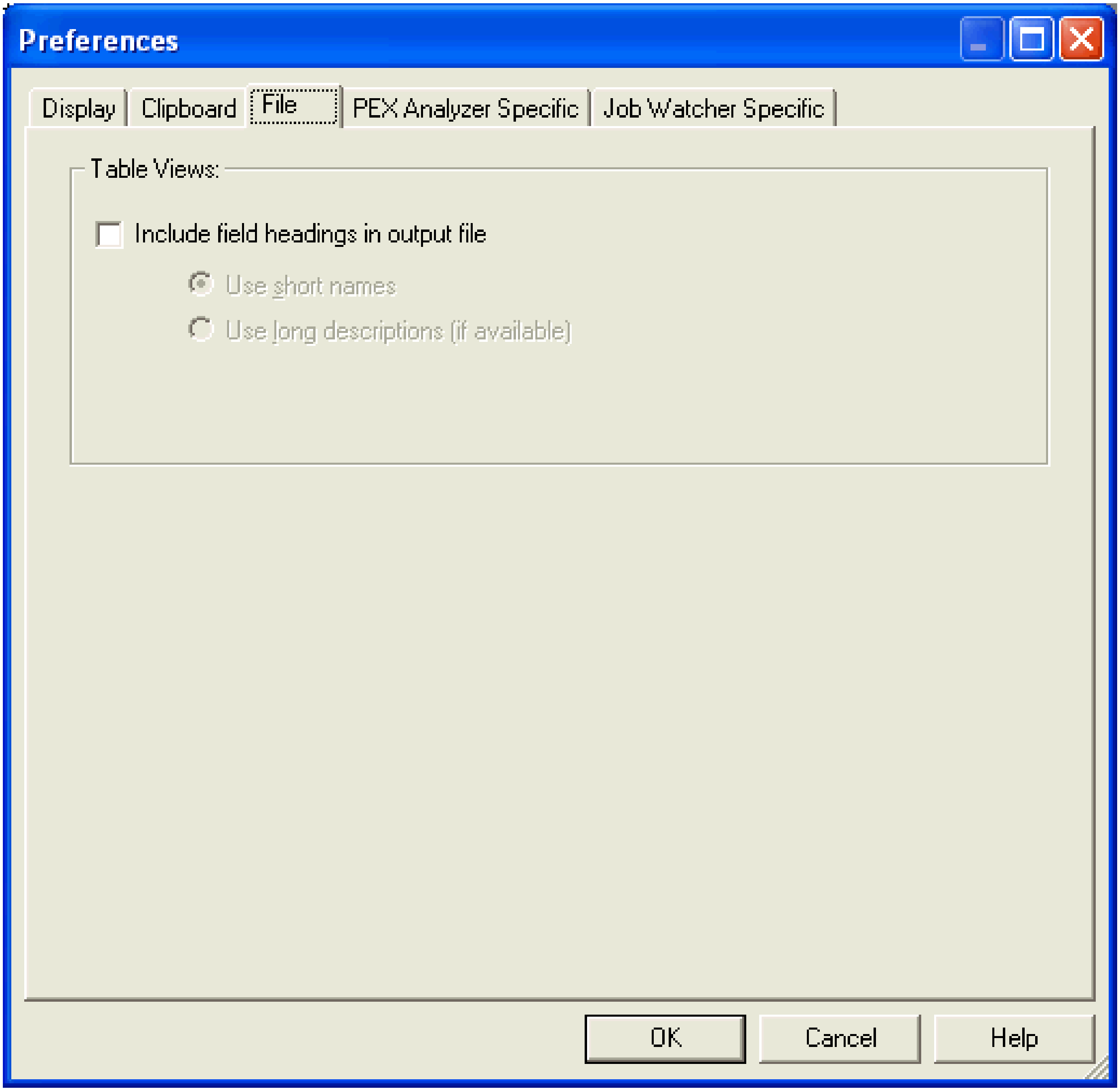
Table View Options	**Description**
Text format	Select the desired text format when copying records or cell selections to the clipboard. The possible choices are: comma separated, tab separated and rich text format.
Include field headings in copied data	Select this option to indicate that field headings should be included as the first record of data when copying data to the clipboard. If this option is selected you can choose to use short field names or long descriptions for the copied output.

Graph View Options	**Description**
Image format	Select the desired image format when copying a graph view to the clipboard. This option will copy the currently active graph view to the clipboard as a bitmap.

1.9.3 File

The File page on the Preferences window lets the user work with options related to creating output files from a table view's data.

An example of this interface is shown below:



□ Include field headings in output file

● Use short names

○ Use long descriptions (if available)

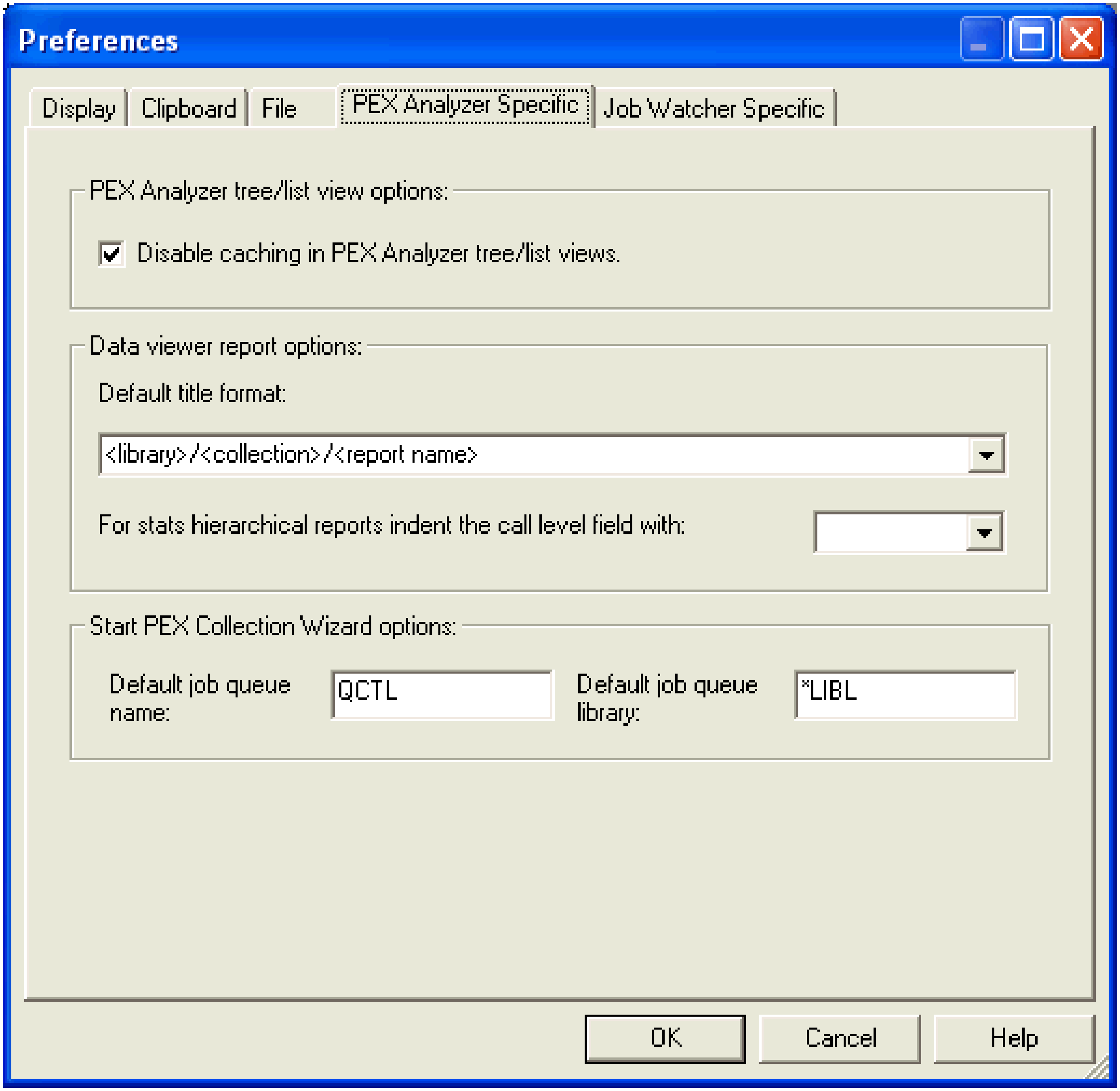
The options available on this page are:

Table View Options	Description
Include field headings in output file	Select this option to indicate that field headings should be included as the first record of data when generating the output file. If this option is selected the user may choose to use short field names or long descriptions in the output file. To generate an output file use the File | Save As... menu when for an active Table View.

1.9.4 PEX Analyzer

The PEX Analyzer page on the Preferences window lets the user work with options related to the PEX Analyzer component of iDoctor for iSeries.

An example of this interface is shown below:



PEX Analyzer tree/list view options:

☑ Disable caching in PEX Analyzer tree/list views.

Data viewer report options:

Default title format: <library>/<collection>/<report name>

For stats hierarchical reports indent the call level field with:

Start PEX Collection Wizard options:

Default job queue name: QCTL

Default job queue library: *LIBL

The options available on this page are summarized below:

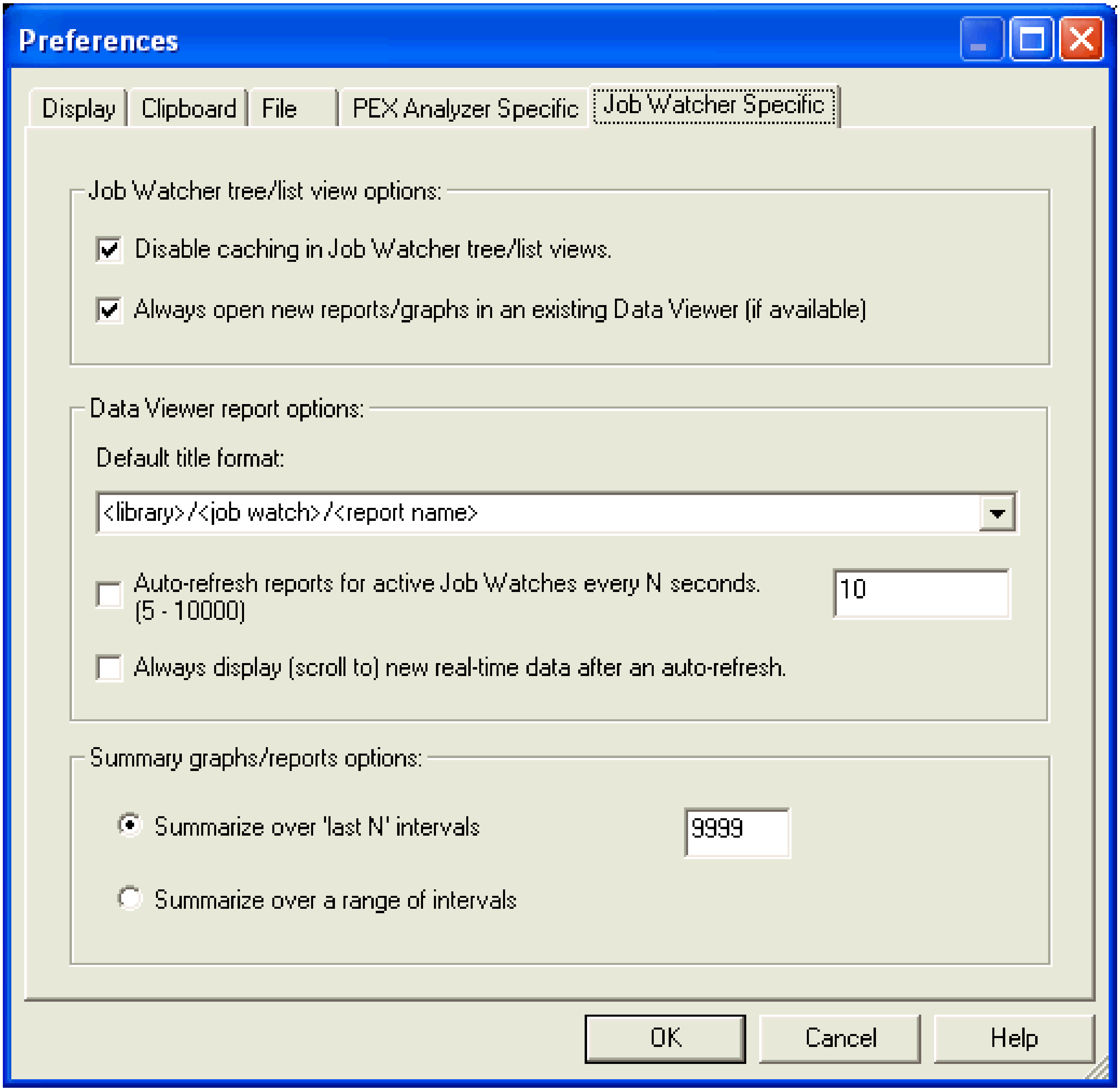
Option	Description
Disable caching	Turn caching on or off in the PEX Analyzer component view. If caching is on and a folder in the tree is clicked that has already been previously clicked it will NOT go to the server to get the latest information. However, this feature can be useful if the system or connection is slow and it's only desired to get data on a "refresh (F5)".
Default Title Format	Use this option to identify how the titles of PEX Analyzer reports should be named. The dropdown list contains several different possible name formats. Other possible titles are available by modifying the value in the list and including any of the tabs listed below in <>. <library> - Library name for the collection <collection> - Collection name <file> - The filename for the report. Each PEX Analyzer report is typically over a single file. <report name> - The long report description. <analysis description> - The analysis description. <analysis member> - The member name for all report files for this analysis. This member name will also match the member of the primary file you are viewing, unless you have changed it using the Query Definition -> Member selection page.
Indent call level with	This option effects the Call Level field for a Statistical hierarchical report (table view). The call level will be indented with the character selected in the drop-down list.
Default job queue name	Indicates the value to use for the default job queue advanced option in the Start PEX Collection Wizard.
Default job queue library	Indicates the value to use for the default job queue library name advanced option in the Start PEX Collection Wizard.

Table of Contents | Previous | Next

1.9.5 Job Watcher

The Job Watcher - Specific page on the Preferences window lets the user work with options only related to the Job Watcher component of iDoctor for iSeries.

An example of this interface is shown below:



Job Watcher tree/list view options:

☑ Disable caching in Job Watcher tree/list views.

☑ Always open new reports/graphs in an existing Data Viewer (if available)

Data Viewer report options:

Default title format: <library>/<job watch>/<report name>

□ Auto-refresh reports for active Job Watches every N seconds. (5 - 10000) 10

□ Always display (scroll to) new real-time data after an auto-refresh.

Summary graphs/reports options:

● Summarize over 'last N' intervals 9999

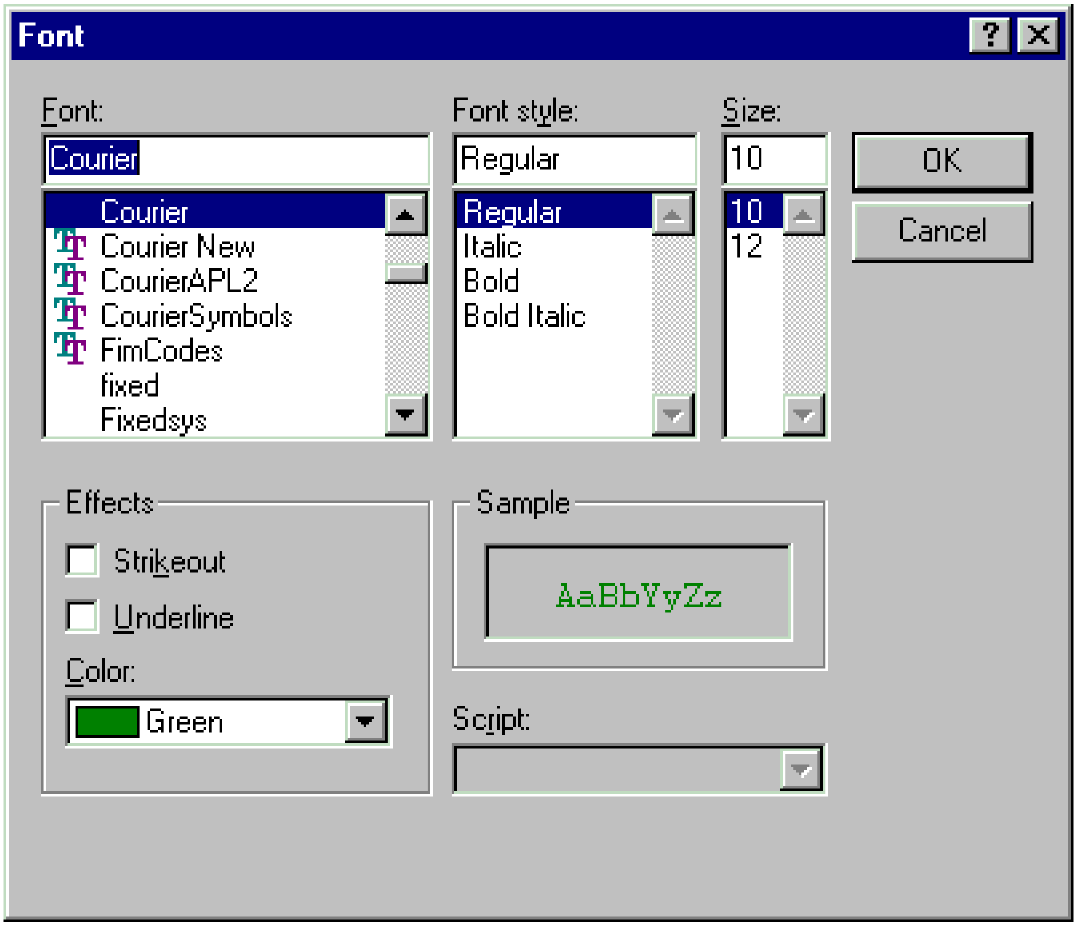
○ Summarize over a range of intervals

The options available on this page are summarized in the tables below:

Options	**Description**
Disable caching in tree/list views	Turn caching on or off in the Job Watcher component view. If caching is on and a folder in the tree is clicked that has already been previously clicked it will NOT go to the server to get the latest information. However, this feature can be useful if the system or connection is slow and it's only desired to get data on a "refresh (F5)".
Always open new reports/graphs in an existing Data Viewer	This option lets the user choose to open every Job Watcher report into an existing Data Viewer or into a new Data Viewer.
Default title format	Use this option to identify how Job Watcher reports should be named. The dropdown list contains several different possible name formats. Other possible titles are available by modifying the value in the list and including any of the tabs listed below in <>. <library> - Library name for the watch <job watch> - Job Watch name <report name> - The long report description.
Auto refresh reports for active Job Watches every N seconds	This option lets the user specify how often to auto refresh reports in the Data Viewer that are over currently active Job Watches. **Note:** only the report with the current focus will get refreshed every N seconds.
Always scroll to new data after refresh	This option indicates that after an auto refresh occurs the scrollbar should be adjusted to scroll to the end of the table or graph. This can be useful if new data is consistently being added to the end of the report.
Summarize over a range or last N intervals	Indicates for the Job Watcher job summary graphs and reports, how many intervals should be included in the report. Either the last N intervals or a range of intervals may be specified.

1.10 Set Font Dialog

Another feature of the Data Viewer is the ability to customize the font used in the Table Views. The Set Font dialog provides the user with this flexibility. To change the Table View font, right-click on an active Table View and choose the Set Font... menu. The font will be saved for use in future iDoctor for iSeries sessions. In addition to table views this font is used in all tree/list views and list views elsewhere in the application.



Font

Font: Courier

Font style: Regular

Size: 10

Effects

□ Strikeout

□ Underline

Color: 🟩 Green

Sample

AaBbYyZz

Script:

Table of Contents | Previous | Next

Chapter 2 Job Watcher

This chapter provides an overview of the interfaces within the iDoctor for iSeries - Job Watcher component.

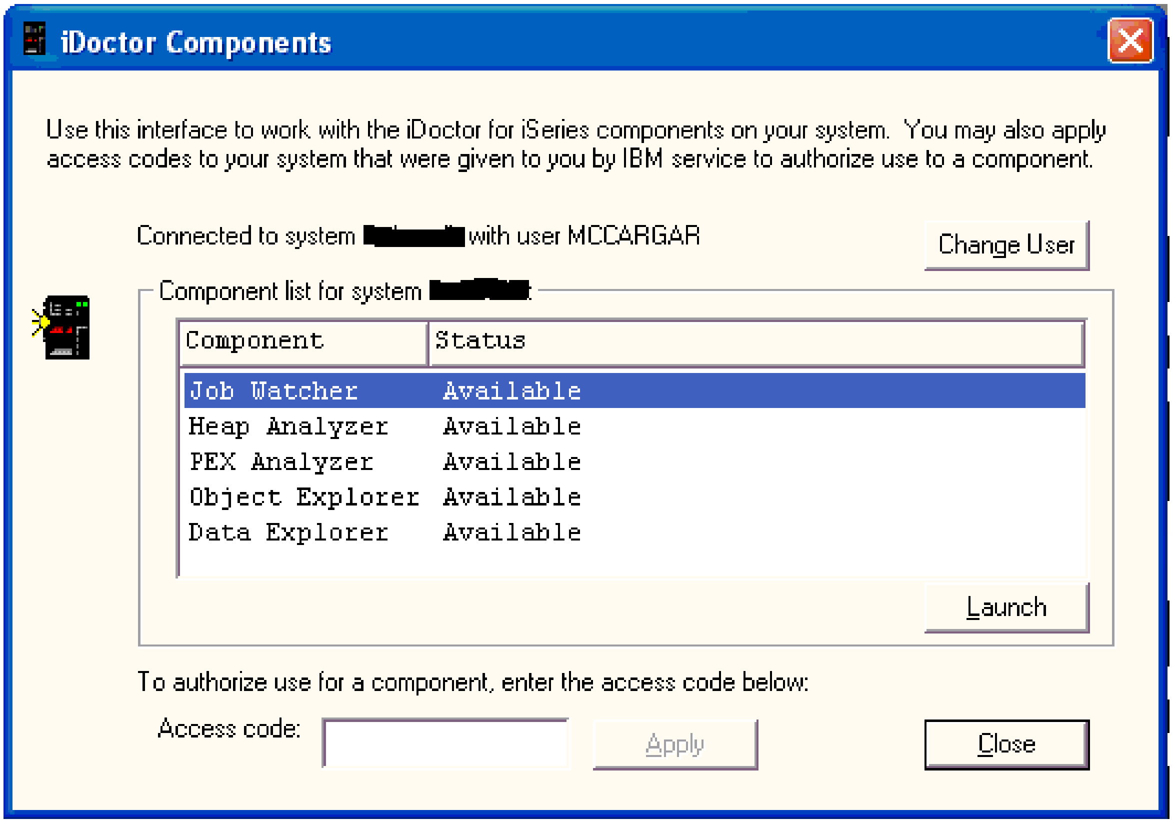
2.1 Job Watcher Basics

The Job Watcher component provides a number of interfaces designed to help the user more easily determine what a job is doing and more importantly why it is not running effectively.

Starting Job Watcher

Job Watcher is a component of the iDoctor for iSeries suite of tools. iDoctor for iSeries can be started using the Start menu: Start->Programs->iDoctor for iSeries. Once the iDoctor for iSeries application appears, the Job Watcher component is started from the Connection List View by double-clicking on the desired system.

A list of available components will appear on the next window. Double-click on the Job Watcher component or select Job Watcher and click the Launch button in order to continue



Use this interface to work with the iDoctor for iSeries components on your system. You may also apply access codes to your system that were given to you by IBM service to authorize use to a component.

Connected to system [redacted] with user MCCARGAR

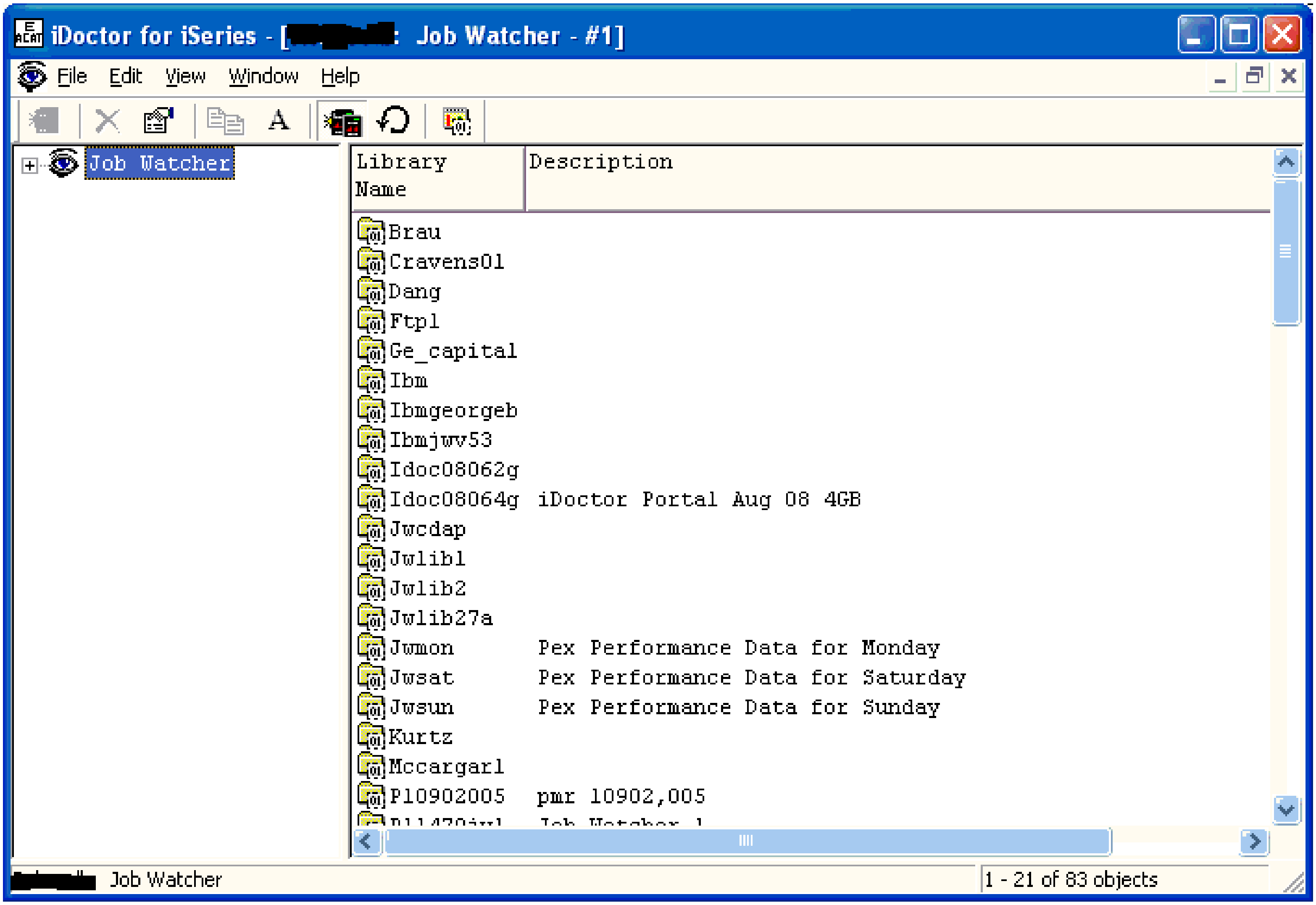
Component list for system [redacted]

Component	Status
Job Watcher	Available
Heap Analyzer	Available
PEX Analyzer	Available
Object Explorer	Available
Data Explorer	Available

To authorize use for a component, enter the access code below:

Access code:

After clicking the Launch button the following window will appear. This window is called the Job Watcher component view.



Library Name	Description
Brau	
Cravens01	
Dang	
Ftpl	
Ge_capital	
Ibm	
Ibmgeorgeb	
Ibmjwv53	
Idoc08062g	
Idoc08064g	iDoctor Portal Aug 08 4GB
Jwcdap	
Jwlib1	
Jwlib2	
Jwlib27a	
Jwmon	Pex Performance Data for Monday
Jwsat	Pex Performance Data for Saturday
Jwsun	Pex Performance Data for Sunday
Kurtz	
Mccargarl	
P10902005	pmr 10902,005

[redacted] Job Watcher

1 - 21 of 83 objects

[The Job Watcher component displaying a list of libraries containing Job Watcher data on a system.]

The 'Job Watcher' folder contains a list of library folders, each representing a library on the iSeries system that contains Job Watcher database files (job watches). The list displays each library's name and description.

Job Watcher Objects

There are four types of objects within the tree/list views of Job Watcher in the following order: **Libraries**, **job watches**, **job watch report folders**, and **reports**. Each of these will covered in more detail in the next sections.

Job Watcher Menu Options

The following menu options are available by right-clicking on the 'Job Watcher' folder in the tree/list view above.

Menu Item	Description
Explore	Displays the contents of the Job Watcher folder (list of libraries on the system containing Job Watcher data) in the right pane of the tree/list window.
Start Job Watch	This menu will open the Start Job Watch Wizard where the user can define and run a Job Watch.
Migrate query and graph definitions	This action will migrate any old iDoctor for iSeries query and graph definitions into the latest format so they will be visible in the client.
Open New Data Viewer	Opens a new Data Viewer window. This window is used to display tables and graphs on the system. You can open Job Watcher reports into this window or you can also open any other type of physical file and view as a graph or table.

Properties	Use this menu to display Job Watcher version information installed on the current system.

Table of Contents | Previous | Next

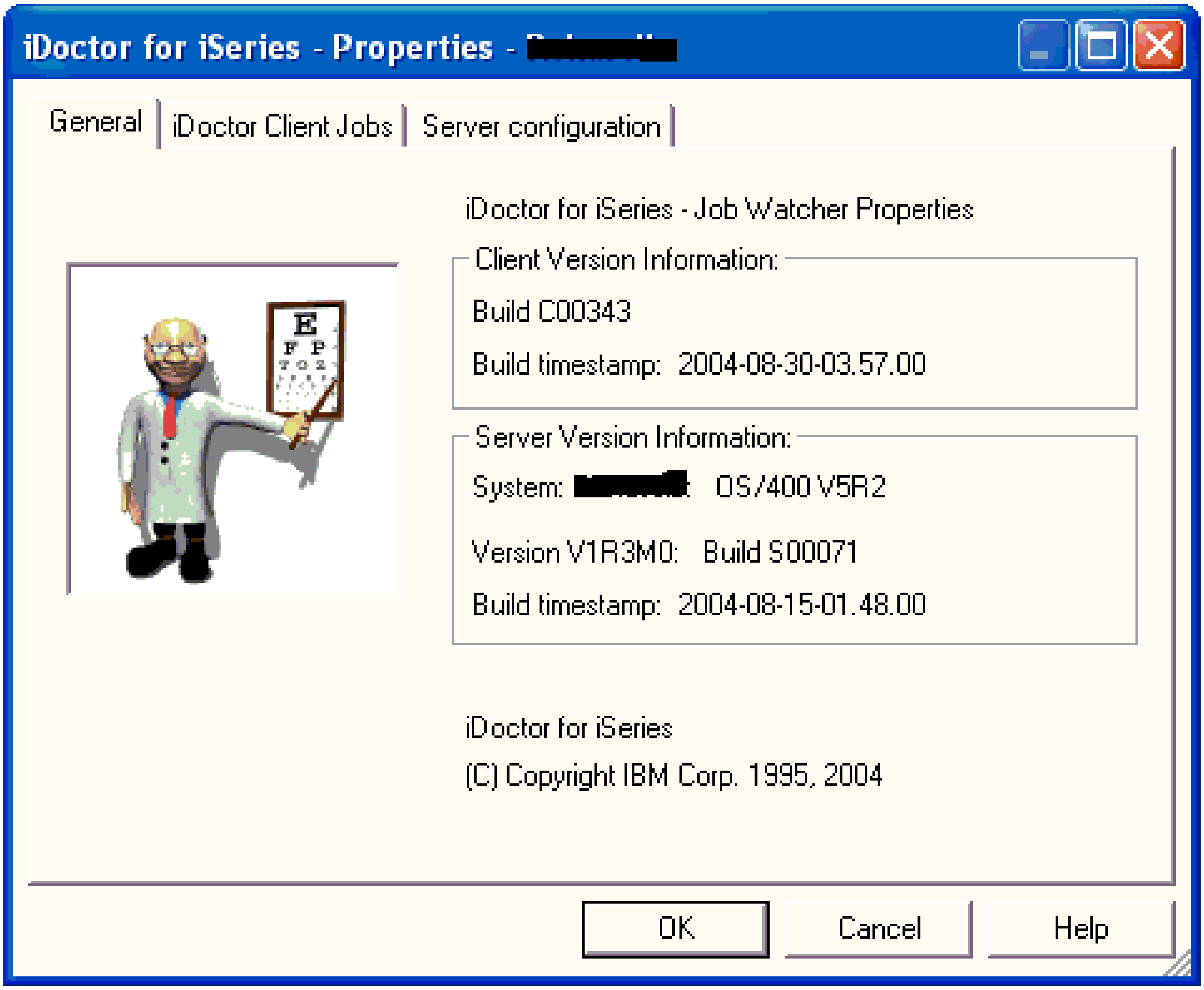
2.2 Job Watcher Component Property Pages

Each component view including Job Watcher has a set of property pages available by right-clicking on the component icon and choosing the Properties... menu. The Job Watcher component property pages contain basic information about the level of Job Watcher installed, as well as the job queue and subsystem that has been configured to run Job Watcher collections on the server.

The next sections cover the different property pages for the Job Watcher component.

Table of Contents | Previous | Next

2.2.1 General



iDoctor for iSeries - Properties - [redacted]

iDoctor for iSeries - Job Watcher Properties

Client Version Information:

Build C00343

Build timestamp: 2004-08-30-03.57.00

Server Version Information:

System: [redacted] OS/400 V5R2

Version V1R3M0: Build S00071

Build timestamp: 2004-08-15-01.48.00

iDoctor for iSeries

(C) Copyright IBM Corp. 1995, 2004

[Job Watcher component properties General page.]

This page displays information about the version of Job Watcher installed on both the client and the server. The following information is supplied within the General page of this window:

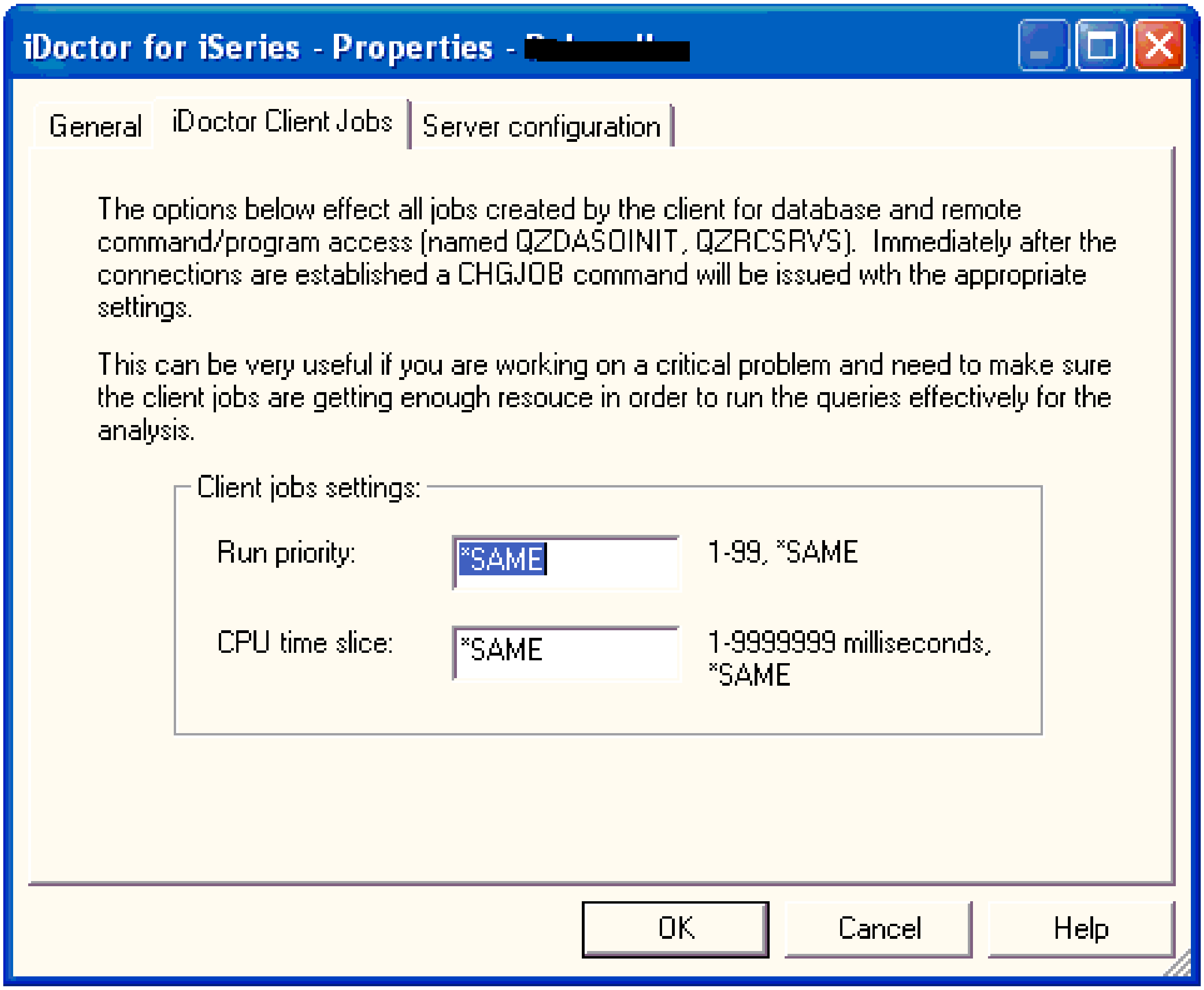
Client Version Information	Description
Build	The build number of the client your are currently running. This number goes up everytime a new build is run for internal (IBM only) testing or for customer use. There will be "internal only" builds resulting in perceived gaps in the build number sequence for customers, but please disregard this.

Build timestamp	The date/time the client build was produced. This value is shown in yyyy-mm-dd-hh.mm.ss format.

Server Version Information	**Description**
System name	The system that the current component view is connected to.
OS/400	The version and release of OS/400 on the active system.
Version	The version and release of Job Watcher on the system. For Job Watcher running at release V5R2 this value is usually V1R3M0.
Build	Build number of the component installed on the server side
Build timestamp	The date/time the server build was produced. This value is shown in yyyy-mm-dd-hh.mm.ss format.

Table of Contents Previous Next

2.2.2 iDoctor Client Jobs



The options below effect all jobs created by the client for database and remote command/program access (named QZDASOINIT, QZRCSRVS). Immediately after the connections are established a CHGJOB command will be issued with the appropriate settings.

This can be very useful if you are working on a critical problem and need to make sure the client jobs are getting enough resouce in order to run the queries effectively for the analysis.

Client jobs settings:

Run priority: *SAME 1-99, *SAME

CPU time slice: *SAME 1-9999999 milliseconds, *SAME

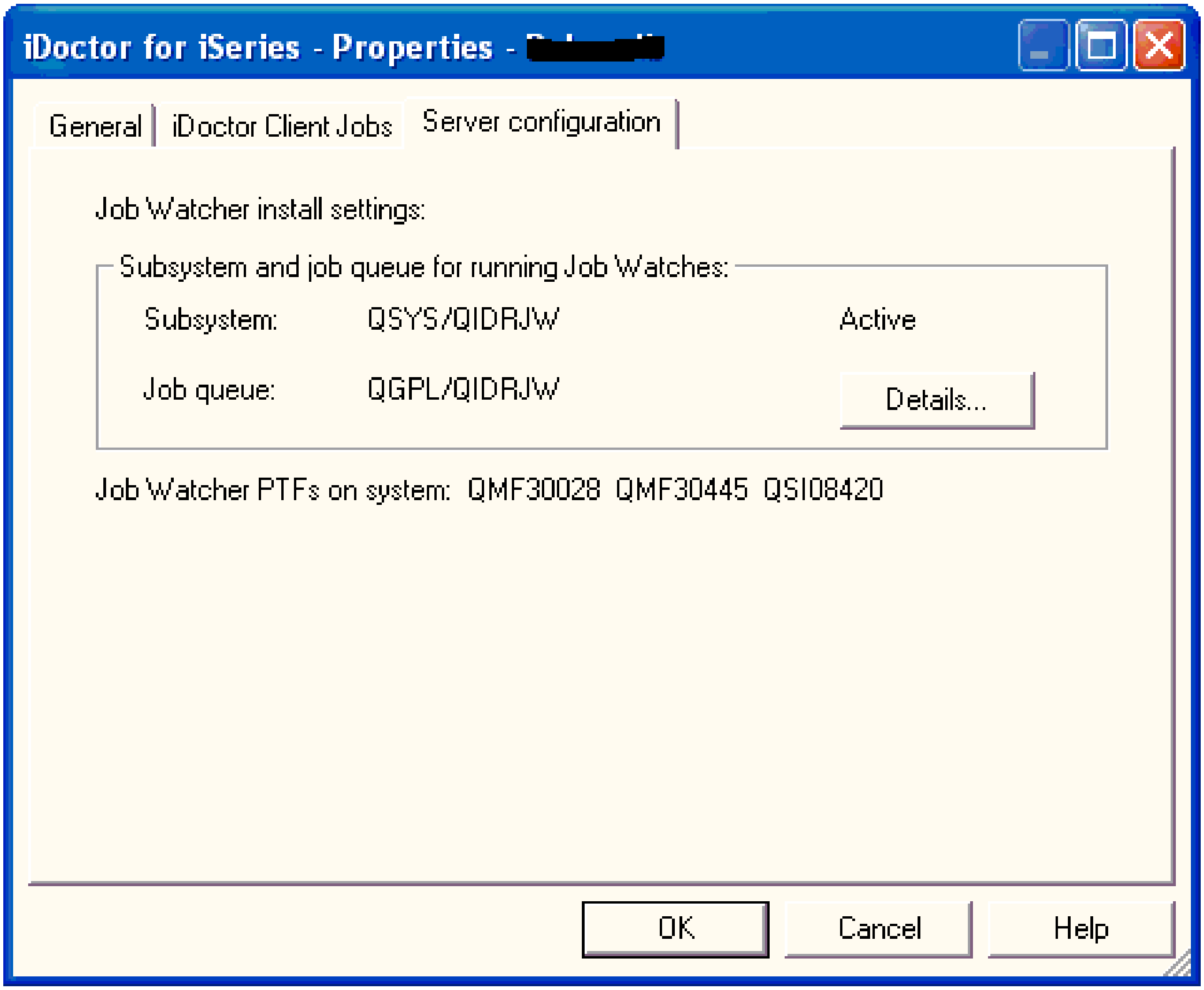
[Job Watcher component properties displaying the iDoctor Client Jobs page.]

The following information is supplied within the iDoctor Client Jobs page of this window:

This page lets the user set the run priority and CPU time slice of all iDoctor client jobs. This should only be set by advanced users. You must shut down the client and restart in order for these changes to take effect.

Table of Contents | Previous | Next

2.2.3 Server configuration



iDoctor for iSeries - Properties - [redacted]

Job Watcher install settings:

Subsystem and job queue for running Job Watches:

Subsystem: QSYS/QIDRJW Active

Job queue: QGPL/QIDRJW

Job Watcher PTFs on system: QMF30028 QMF30445 QSI08420

[Job Watcher component properties displaying the Server configuration page.]

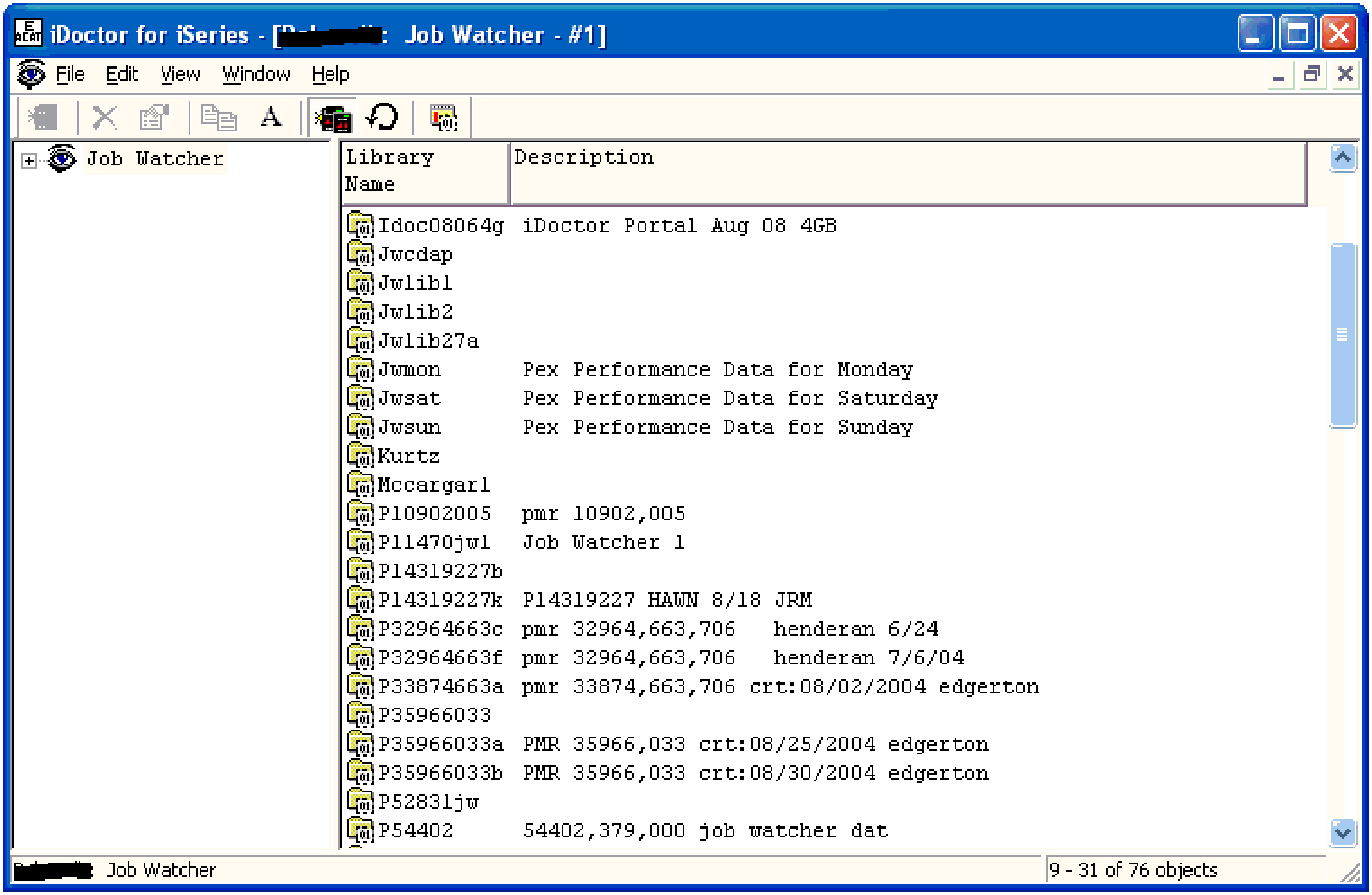
The subsystem and job queue used for running Job Watches is shown on this page. Information about the status of the subsystem and the PTFs installed on the system are also shown on this page.

Table of Contents | Previous | Next

2.3 Libraries

The 'Job Watcher' folder contains a list of library folders, each representing a library on the iSeries system that contains Job Watcher database files. The list displays each library's name and description.

By clicking on a library in the tree you will see its contents (a list of job watches).



Library Name	Description
Idoc08064g	iDoctor Portal Aug 08 4GB
Jwcdap	
Jwlibl	
Jwlib2	
Jwlib27a	
Jwmon	Pex Performance Data for Monday
Jwsat	Pex Performance Data for Saturday
Jwsun	Pex Performance Data for Sunday
Kurtz	
Mccargarl	
P10902005	pmr 10902,005
P11470jwl	Job Watcher 1
P14319227b	
P14319227k	P14319227 HAWN 8/18 JRM
P32964663c	pmr 32964,663,706 henderan 6/24
P32964663f	pmr 32964,663,706 henderan 7/6/04
P33874663a	pmr 33874,663,706 crt:08/02/2004 edgerton
P35966033	
P35966033a	PMR 35966,033 crt:08/25/2004 edgerton
P35966033b	PMR 35966,033 crt:08/30/2004 edgerton
P52831jw	
P54402	54402,379,000 job watcher dat

[redacted] Job Watcher | 9 - 31 of 76 objects

[The Job Watcher component displaying a list of libraries containing Job Watcher data.]

Library Menu Options

The following menu options are available by right-clicking on a library in the tree/list view above.

Menu Item	Description
Explore	Displays the contents of the library (list of job watches within the library) in the right pane of the tree/list window.
Select fields...	Brings up a window that lets the user modify what fields are shown when displaying the contents of the library. The contents of a library are Job Watcher collections. This option provides allows the user to hide/display/reorder fields that are relevant to a Job Watch.
Start Job Watch...	This menu will open the Start Job Watch Wizard so the user can create a Job Watch in the selected library.
Copy...	Gives the user the option to copy the library's contents into a new library or into an existing one.

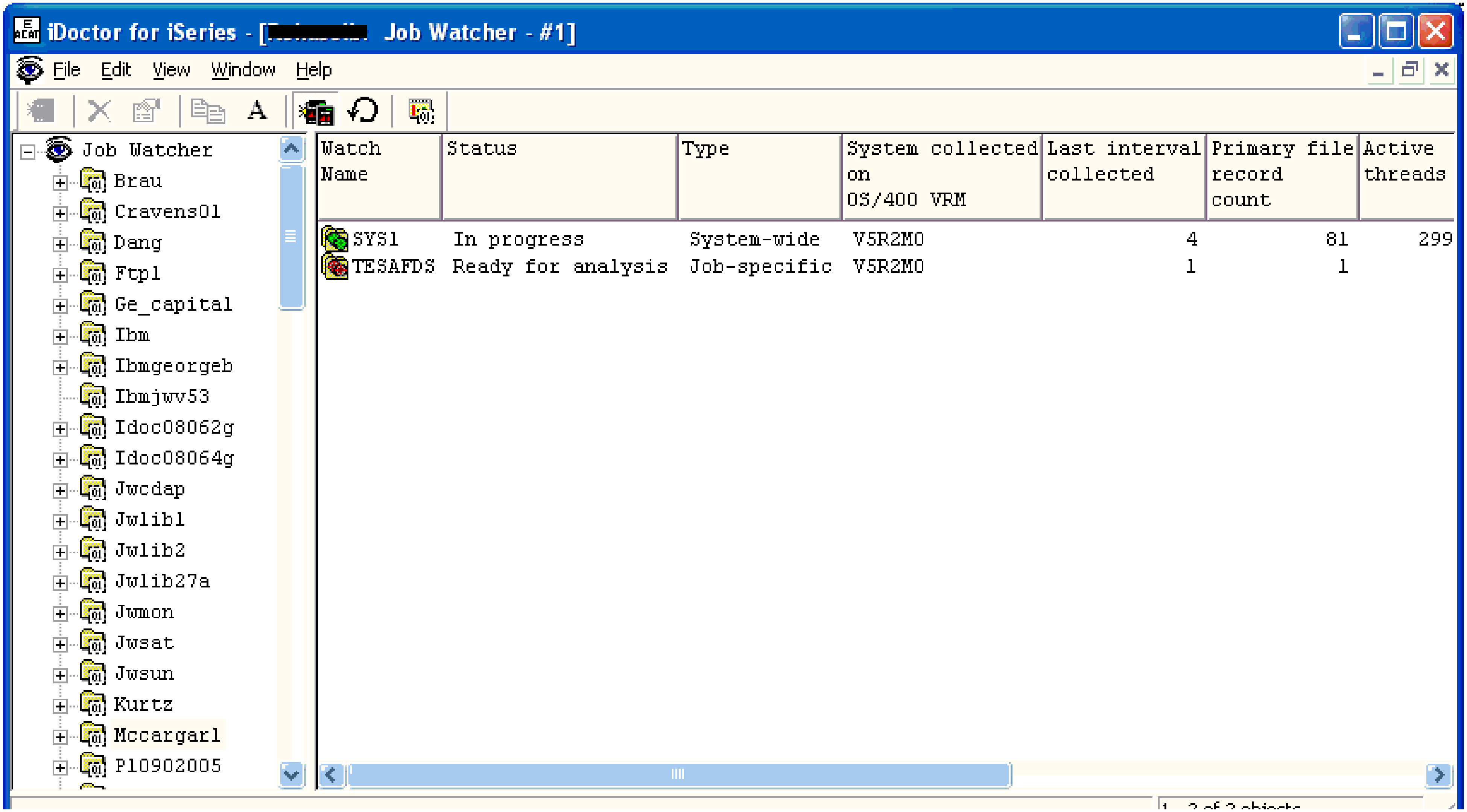
Save...	This option provides the option to save the library's contents into a save file.
Clear	This option clears a library (deletes all objects in the library).
Delete	Deletes the library.
Rename	Renames the library.
Properties	Use this menu to display the library property pages. Basic information similar to that provided by DSPOBJD is available through these property pages.

Table of Contents | Previous | Next

2.4 Job Watches

Moving down the tree within each Library folder are one or more Job Watches that have been created (or are currently being created) within the current library. The green icons indicate active Job Watches and red icons indicate Job Watches which have completed. The status field is used to indicate if any errors occurred during collection or the current status of an active job watch.

There are two types of Job Watches: system-wide and job-specific. The type of watch will be shown in the 3rd column in the list by default (see screen capture below). The type of watch indicates what views/reports are available within the Job Watch and over which jobs the data is being collected.



Watch Name	Status	Type	System collected on OS/400 VRM	Last interval collected	Primary file record count	Active threads
SYS1	In progress	System-wide	V5R2M0	4	81	299
TESAFDS	Ready for analysis	Job-specific	V5R2M0	1	1	

1 - 2 of 2 objects

[Job Watcher displaying the list of Job Watches within a library on a system.]

Job Watch Status

Each Watch has a status field indicating whether it is currently running. You can also tell the status by the color of the icon: Green = active, Red = not active.

Job Watch Fields

The list of job watches displays the watch name, description, status, number of records, as well as several additional fields.

Each Job Watch in the list has a set of fields available which can be optionally reordered and displayed. To change the current field selections for the Job Watch list, use the Select fields... menu from the library folder. A listing of the available fields and a short description is provided in the table below:

Field	**Description**
Watch name	Name of the Job Watch. This name matches the member name used in the output files named QPYRT* that exist on the system.
Type	The type of Job Watch. Possible values are System-wide and Job-specific.
Status	The status field indicates the status of the job on the system running the Job Watch (if active) or if not active the status indicates whether the watch can be analyzed and/or has data available.
System collected on OS/400 VRM	The version of OS/400 that was used to create this Job Watch. On a V5R2 system this potentially could read V5R1M0 if the system was upgraded to V5R2 since the collection was originally created, or if the collection was moved from a V5R1 system to the V5R2 system.
Job Watcher VRM	The version of Job Watcher that was used to create this Job Watch. If the System collected on OS/400 VRM value is V5R2M0 then this value will normally read V1R3M0.
Last interval collected	This value shows the last interval collected. If the Job Watch is not running, this value indicates the total number of intervals that were collected in the Job Watch.
Primary file record count	For a **job-specific** Job Watch this indicates the current number of records found in the main output file QPYRTJWA in the Job Watch library. For a **system-wide** Job Watch this indicates the current number of records found in the log/exception file QPYRTSWLG1 in the Job Watch library. If the Job Watch is active and collecting this value will increase over time as the list of collections is refreshed.
Active threads/tasks	The total number of active jobs/threads detected in this Job Watch as of the last refresh. This value is only shown when the Job Watch is actively running.
Interval duration (secs)	The size of each interval sample (in seconds).
Time last updated	The date/time the Job Watch status information was last refreshed.
Description	A description for the Job Watch specified at creation time.
Rule member	The member name of the rule definition file used to create this Job Watch. Additional information about the rule is available in the property pages for the Job Watch.

Table of Contents | Previous | Next

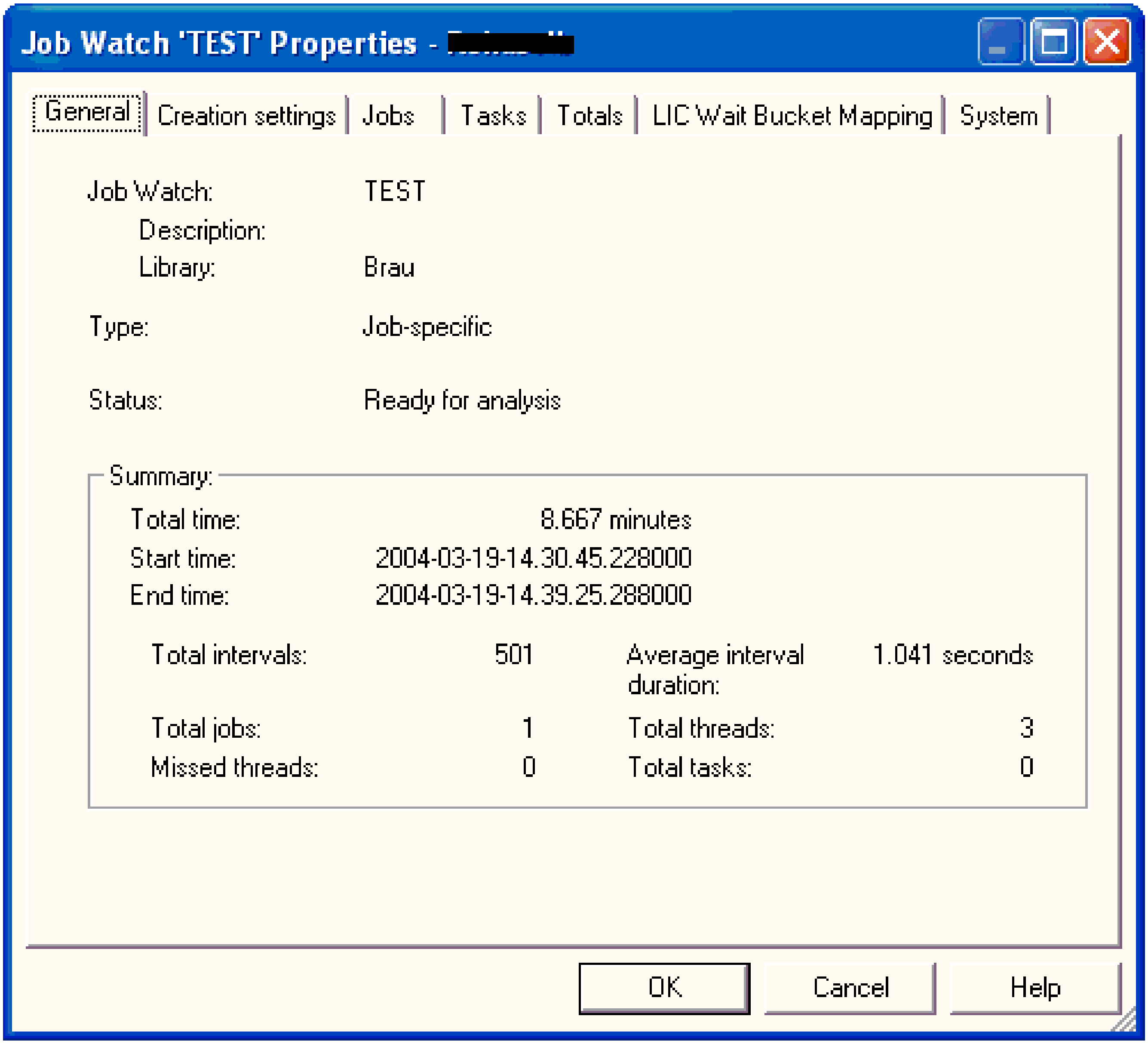
2.4.1 Menu options

The table below outlines the different types of operations that may be performed on a Job Watch by right-clicking on a Job Watch from the Job Watcher component view.

Menu Item	**Description**
Explore	Displays the contents of the Job Watch (the report folders in the Job Watch) in the right pane of the tree/list window.
Record Quick View	Displays the fields for a Job Watch in the list view vertically for easier viewing. Not available from the tree side, only the list side.
Graphs and reports menus	Several menus are available with different graphs and reports over the Job Watch. An example is the CPU graphs menu. The graphs and reports available under each category will differ slightly depending on the type of Job Watch selected.
Delete...	Deletes a Job Watch. Select multiple Job Watches in order to delete more than one at a time. Selecting multiples is only available from the list side of the tree/list view.
Stop	Permanently stops an active Job Watch. This function sends a request to the server to stop the Watch. The watch will only end after the current interval the Job Watch is in when the command is issued completes. Once a Job Watch is stopped it cannot be restarted again.
Properties	Use this menu to display the Job Watch's property pages. The property pages provide quick access to additional information about the Job Watch.

2.4.2 Job-specific Property Pages

This section covers the property pages for a job-specific Job Watch. Access the property pages by right-clicking on a job-specific job watch and choosing the Properties menu.



Job Watch 'TEST' Properties - [redacted]

Job Watch: TEST

Description:

Library: Brau

Type: Job-specific

Status: Ready for analysis

Summary:

Total time: 8.667 minutes

Start time: 2004-03-19-14.30.45.228000

End time: 2004-03-19-14.39.25.288000

Total intervals: 501 Average interval duration: 1.041 seconds

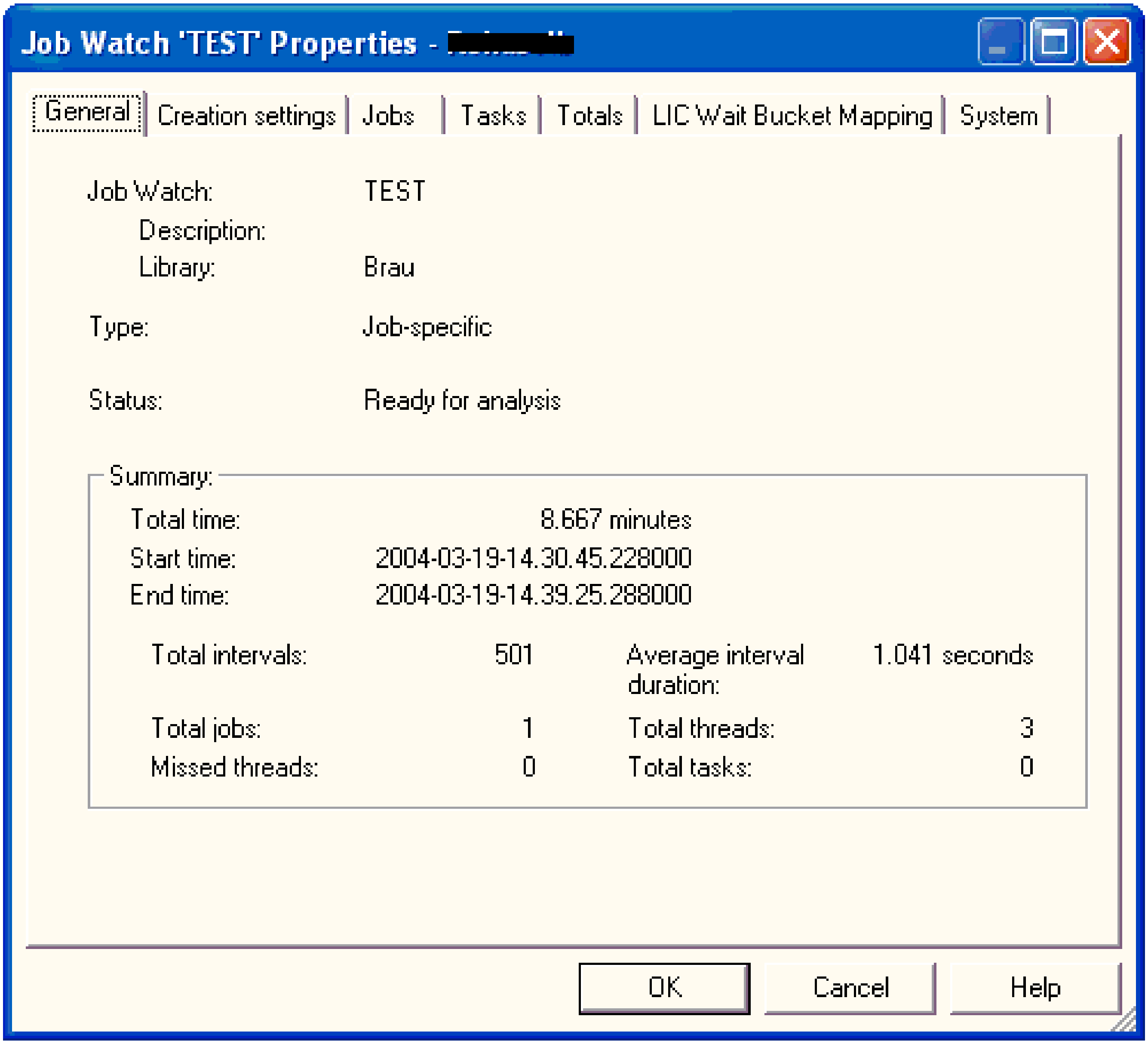
Total jobs: 1 Total threads: 3

Missed threads: 0 Total tasks: 0

Table of Contents | Previous | Next

2.4.2.1 General

The General property page provides a summary of the most basic information for a job-specific Job Watch.



Job Watch 'TEST' Properties - [redacted]

Job Watch: TEST
Description:
Library: Brau

Type: Job-specific

Status: Ready for analysis

Summary:

Total time: 8.667 minutes
Start time: 2004-03-19-14.30.45.228000
End time: 2004-03-19-14.39.25.288000

Total intervals: 501
Average interval duration: 1.041 seconds
Total jobs: 1
Total threads: 3
Missed threads: 0
Total tasks: 0

The following information is displayed on the General property page:

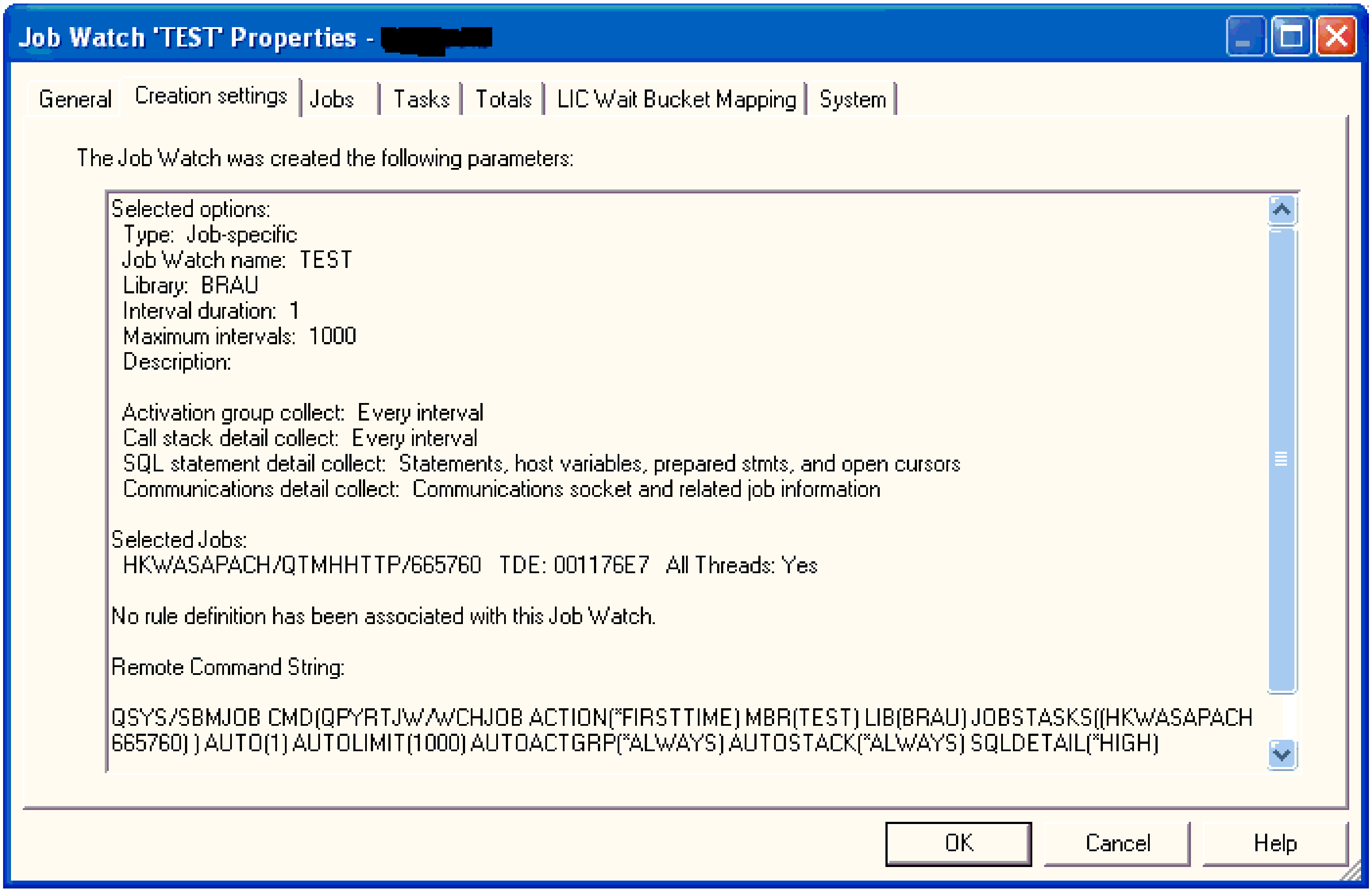
Field Name	Field Description

Job Watch	Name of the job watch. This also matches the member name used in the QPYRTJW* files on the server in the library specified.
Description	Description of the job watch set when the job watch was created.
Library	Library the job watch is stored in.
Type	The type of job watch.
Status	The status of the job watch. This will either be "in progress", "ready for analysis" or some type of error message if the job that created the job watch failed.
Total time	The total time the job watch was running.
Start time	Timestamp of when the collection started.
End time	Timestamp of when the collection ended (if collection has completed).
Total intervals	Number of intervals collected.
Average interval duration	The average time it took to collect an interval of data over all jobs/threads in the Job Watch.
Total jobs	The total number of primary threads collected by the job watch. This is also the number of unique process handles in the job watch.
Total threads	The total number of primary and secondary threads collected by the job watch.
Missed threads	Indicates the total number of threads not collected (due to the 500 thread limit) each interval added together.
Total tasks	Indicates the total number of tasks collected by the job watch.

Table of Contents | Previous | Next

2.4.2.2 Creation settings

The Creation settings property page provides details about the parameters on the WCHJOB command that were used to create the job-specific Job Watch.



The Job Watch was created the following parameters:

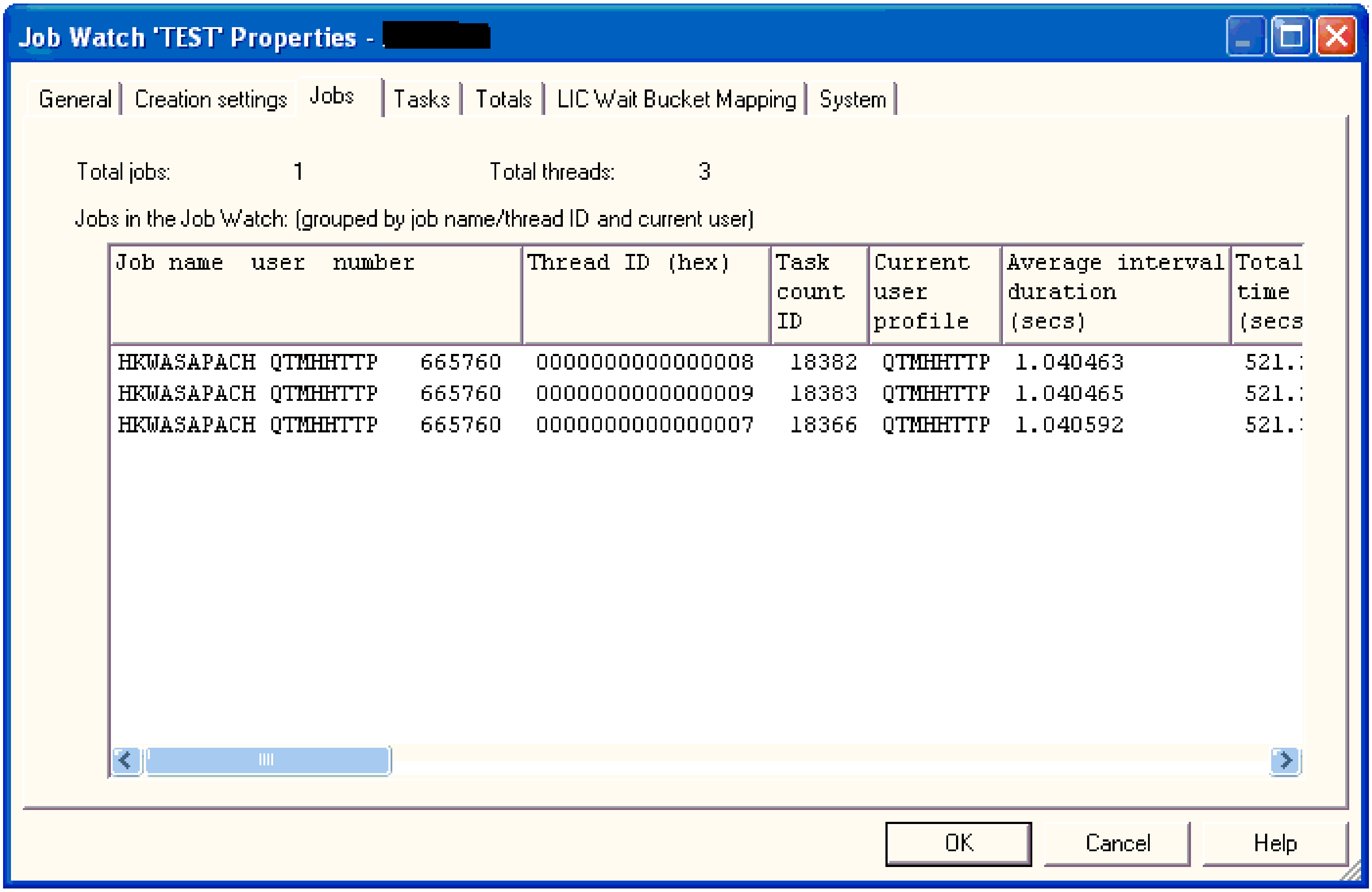
```
Selected options:
  Type:  Job-specific
  Job Watch name:  TEST
  Library:  BRAU
  Interval duration:  1
  Maximum intervals:  1000
  Description:

  Activation group collect:  Every interval
  Call stack detail collect:  Every interval
  SQL statement detail collect:  Statements, host variables, prepared stmts, and open cursors
  Communications detail collect:  Communications socket and related job information

Selected Jobs:
  HKWASAPACH/QTMHHTTP/665760   TDE: 001176E7   All Threads: Yes

No rule definition has been associated with this Job Watch.

Remote Command String:

QSYS/SBMJOB CMD(QPYRTJW/WCHJOB ACTION(*FIRSTTIME) MBR(TEST) LIB(BRAU) JOBSTASKS((HKWASAPACH
665760) ) AUTO(1) AUTOLIMIT(1000) AUTOACTGRP(*ALWAYS) AUTOSTACK(*ALWAYS) SQLDETAIL(*HIGH)
```

The information shown on this window matches the summary page of the Start Job Watch Wizard when the Job Watch was created. **Note:** This page is only shown if the job watch was originally created using the client interface.

Table of Contents Previous Next

2.4.2.3 Jobs

The Jobs property page displays the list of all jobs that were collected by the job watch.



Job Watch 'TEST' Properties - [redacted]

Total jobs: 1 Total threads: 3

Jobs in the Job Watch: (grouped by job name/thread ID and current user)

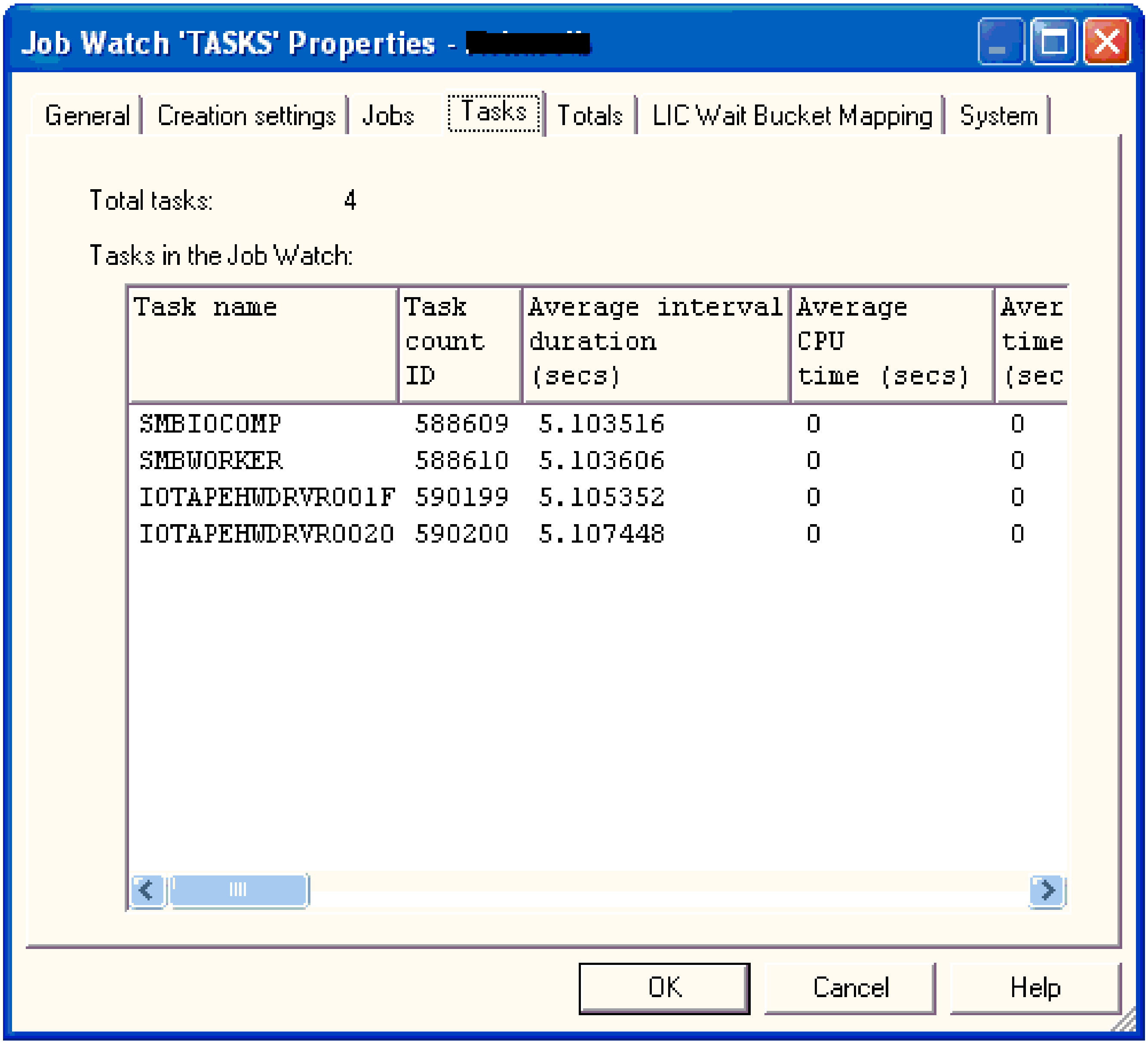
Job name user number	Thread ID (hex)	Task count ID	Current user profile	Average interval duration (secs)	Total time (secs
HKWASAPACH QTMHHTTP 665760	0000000000000008	18382	QTMHHTTP	1.040463	521.
HKWASAPACH QTMHHTTP 665760	0000000000000009	18383	QTMHHTTP	1.040465	521.
HKWASAPACH QTMHHTTP 665760	0000000000000007	18366	QTMHHTTP	1.040592	521.

The list contains all threads with 1 record shown per current user profile per thread. Each job/thread record contains summarized details about the threads that were sampled by the job watch. This tab is only available if the job watch has completed.

Table of Contents | Previous | Next

2.4.2.4 Tasks

The Tasks property page provides the list of tasks that were collected by the job watch. Each task record contains summarized details about the task that were sampled by the job watch. This tab is only available if the job watch has completed.



Job Watch 'TASKS' Properties - [redacted]

Total tasks: 4

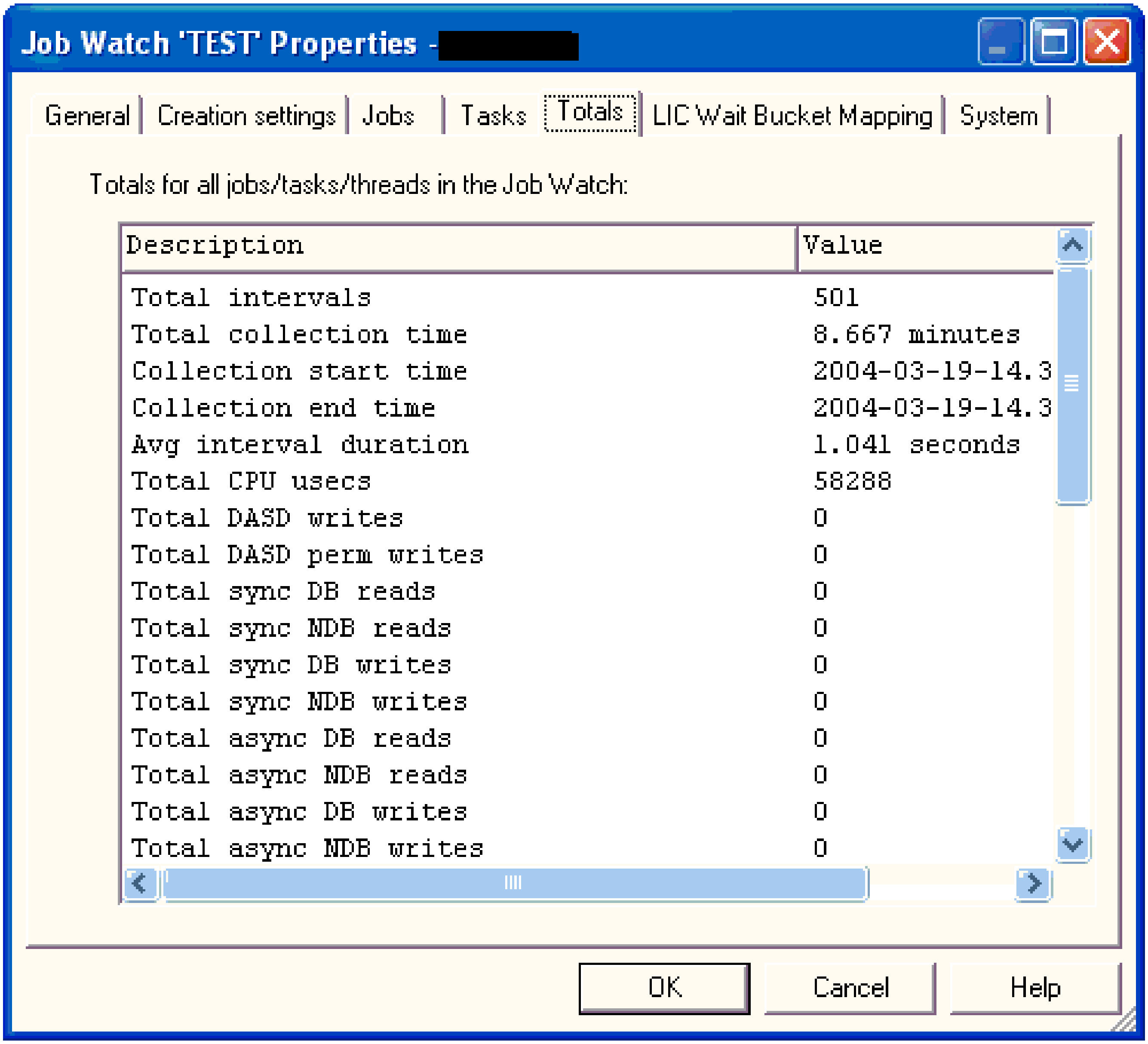
Tasks in the Job Watch:

Task name	Task count ID	Average interval duration (secs)	Average CPU time (secs)	Aver time (sec
SMBIOCOMP	588609	5.103516	0	0
SMBWORKER	588610	5.103606	0	0
IOTAPEHWDRVR001F	590199	5.105352	0	0
IOTAPEHWDRVR0020	590200	5.107448	0	0

Table of Contents | Previous | Next

2.4.2.5 Totals

The Totals property page provides a summary of many different types of performance statistics over all data collected by the Job Watch. The totals provided on this page only pertain to job/thread/task information that was collected while the job watch was running. This page is only available for collections that have completed.



Job Watch 'TEST' Properties - [redacted]

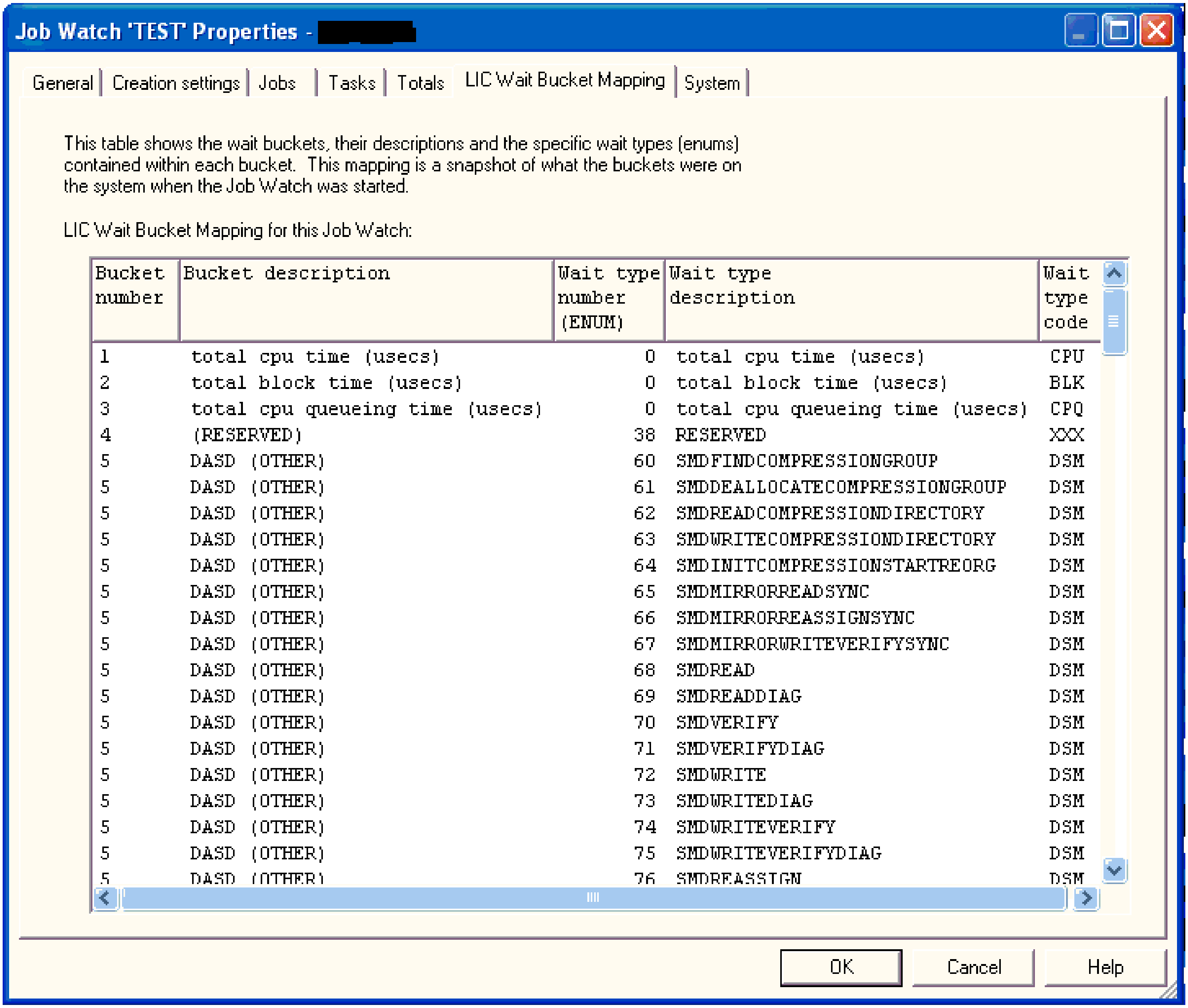
Totals for all jobs/tasks/threads in the Job Watch:

Description	Value
Total intervals	501
Total collection time	8.667 minutes
Collection start time	2004-03-19-14.3
Collection end time	2004-03-19-14.3
Avg interval duration	1.041 seconds
Total CPU usecs	58288
Total DASD writes	0
Total DASD perm writes	0
Total sync DB reads	0
Total sync NDB reads	0
Total sync DB writes	0
Total sync NDB writes	0
Total async DB reads	0
Total async NDB reads	0
Total async DB writes	0
Total async NDB writes	0

Table of Contents | Previous | Next

2.4.2.6 LIC wait bucket mapping

This page shows the wait buckets and the wait points within each bucket at the time the job watch was created.



Job Watch 'TEST' Properties - [redacted]

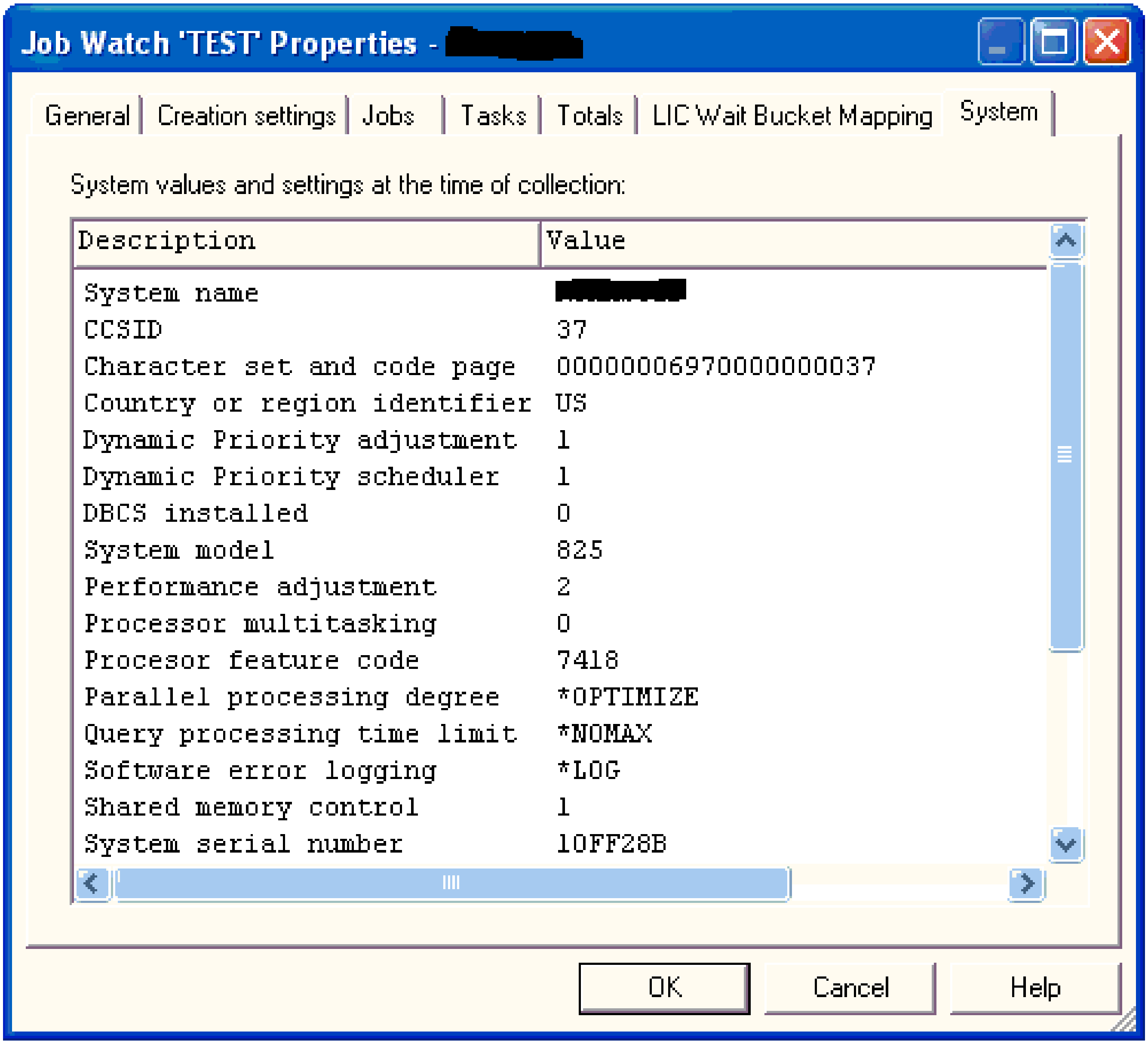
This table shows the wait buckets, their descriptions and the specific wait types (enums) contained within each bucket. This mapping is a snapshot of what the buckets were on the system when the Job Watch was started.

LIC Wait Bucket Mapping for this Job Watch:

Bucket number	Bucket description	Wait type number (ENUM)	Wait type description	Wait type code
1	total cpu time (usecs)	0	total cpu time (usecs)	CPU
2	total block time (usecs)	0	total block time (usecs)	BLK
3	total cpu queueing time (usecs)	0	total cpu queueing time (usecs)	CPQ
4	(RESERVED)	38	RESERVED	XXX
5	DASD (OTHER)	60	SMDFINDCOMPRESSIONGROUP	DSM
5	DASD (OTHER)	61	SMDDEALLOCATECOMPRESSIONGROUP	DSM
5	DASD (OTHER)	62	SMDREADCOMPRESSIONDIRECTORY	DSM
5	DASD (OTHER)	63	SMDWRITECOMPRESSIONDIRECTORY	DSM
5	DASD (OTHER)	64	SMDINITCOMPRESSIONSTARTREORG	DSM
5	DASD (OTHER)	65	SMDMIRRORREADSYNC	DSM
5	DASD (OTHER)	66	SMDMIRRORREASSIGNSYNC	DSM
5	DASD (OTHER)	67	SMDMIRRORWRITEVERIFYSYNC	DSM
5	DASD (OTHER)	68	SMDREAD	DSM
5	DASD (OTHER)	69	SMDREADDIAG	DSM
5	DASD (OTHER)	70	SMDVERIFY	DSM
5	DASD (OTHER)	71	SMDVERIFYDIAG	DSM
5	DASD (OTHER)	72	SMDWRITE	DSM
5	DASD (OTHER)	73	SMDWRITEDIAG	DSM
5	DASD (OTHER)	74	SMDWRITEVERIFY	DSM
5	DASD (OTHER)	75	SMDWRITEVERIFYDIAG	DSM
5	DASD (OTHER)	76	SMDREASSIGN	DSM

2.4.2.7 System

This page shows information about the system the job watch was collected on including many system values.



System values and settings at the time of collection:

Description	Value
System name	[redacted]
CCSID	37
Character set and code page	00000006970000000037
Country or region identifier	US
Dynamic Priority adjustment	1
Dynamic Priority scheduler	1
DBCS installed	0
System model	825
Performance adjustment	2
Processor multitasking	0
Procesor feature code	7418
Parallel processing degree	*OPTIMIZE
Query processing time limit	*NOMAX
Software error logging	*LOG
Shared memory control	1
System serial number	10FF28B

Table of Contents | Previous | Next

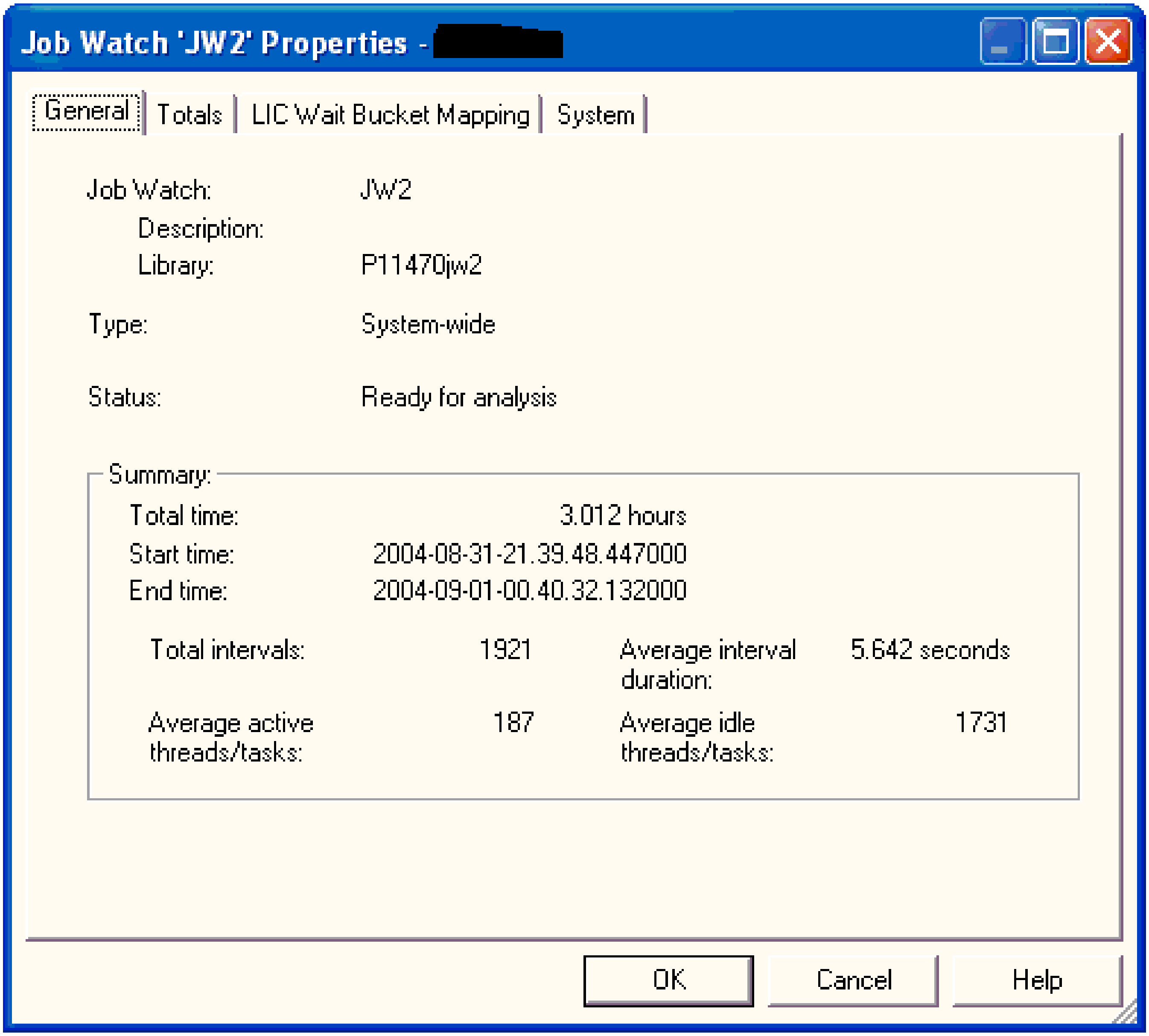
2.4.3 System-wide Property Pages

This section covers the property pages for a system-wide Job Watch. Access the property pages by right-clicking on a system-wide job watch and choosing the Properties menu.

Table of Contents | Previous | Next

2.4.3.1 General

The General property page provides a summary of the most basic information for a system-wide Job Watch.



Job Watch: JW2
Description:
Library: P11470jw2

Type: System-wide

Status: Ready for analysis

Summary:
Total time: 3.012 hours
Start time: 2004-08-31-21.39.48.447000
End time: 2004-09-01-00.40.32.132000

Total intervals: 1921
Average interval duration: 5.642 seconds

Average active threads/tasks: 187
Average idle threads/tasks: 1731

The following information is displayed on the General property page:

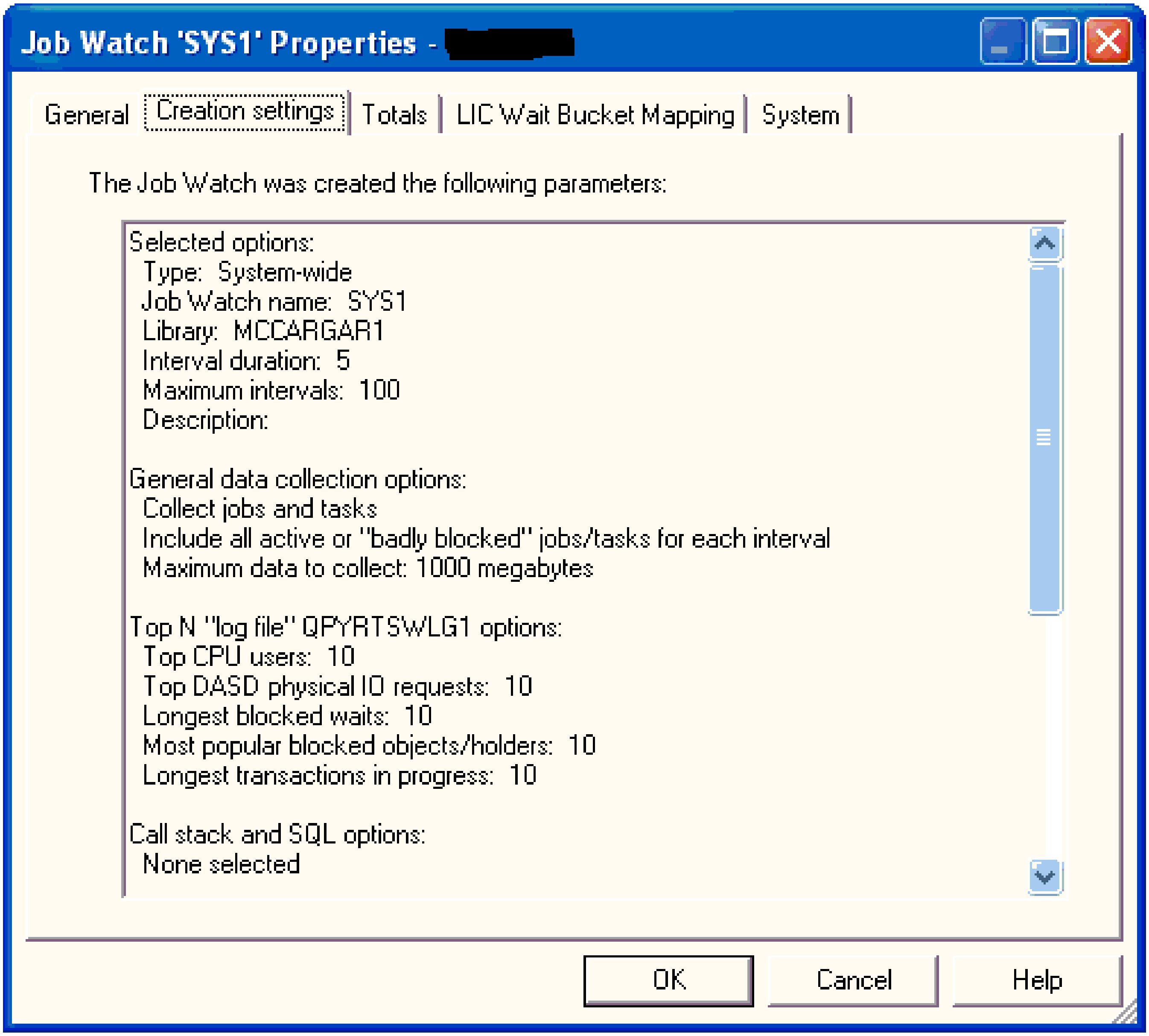
Field Name	Field Description

Job Watch	Name of the job watch. This also matches the member name used in the QPYRTSW* files on the server in the library specified.
Description	Description of the job watch set when the job watch was created.
Library	Library the job watch is stored in.
Type	The type of job watch.
Status	The status of the job watch. This will either be "in progress", "ready for analysis" or some type of error message if the job that created the job watch failed.
Total time	The total time the job watch was running.
Start time	Timestamp of when the collection started.
End time	Timestamp of when the collection ended (if collection has completed).
Total intervals	Number of intervals collected.
Average interval duration	The average time it took to collect an interval of data over all jobs/threads in the Job Watch.
Average active threads/tasks	The average number of threads/tasks that were using CPU each interval over the entire Job Watch.
Average idle threads/tasks	The average number of threads/tasks that were NOT using CPU each interval over the entire Job Watch.

Table of Contents | Previous | Next

2.4.3.2 Creation settings

The Creation settings property page provides details about the parameters on the WCHSYS command that were used to create the system-wide Job Watch.



The Job Watch was created the following parameters:

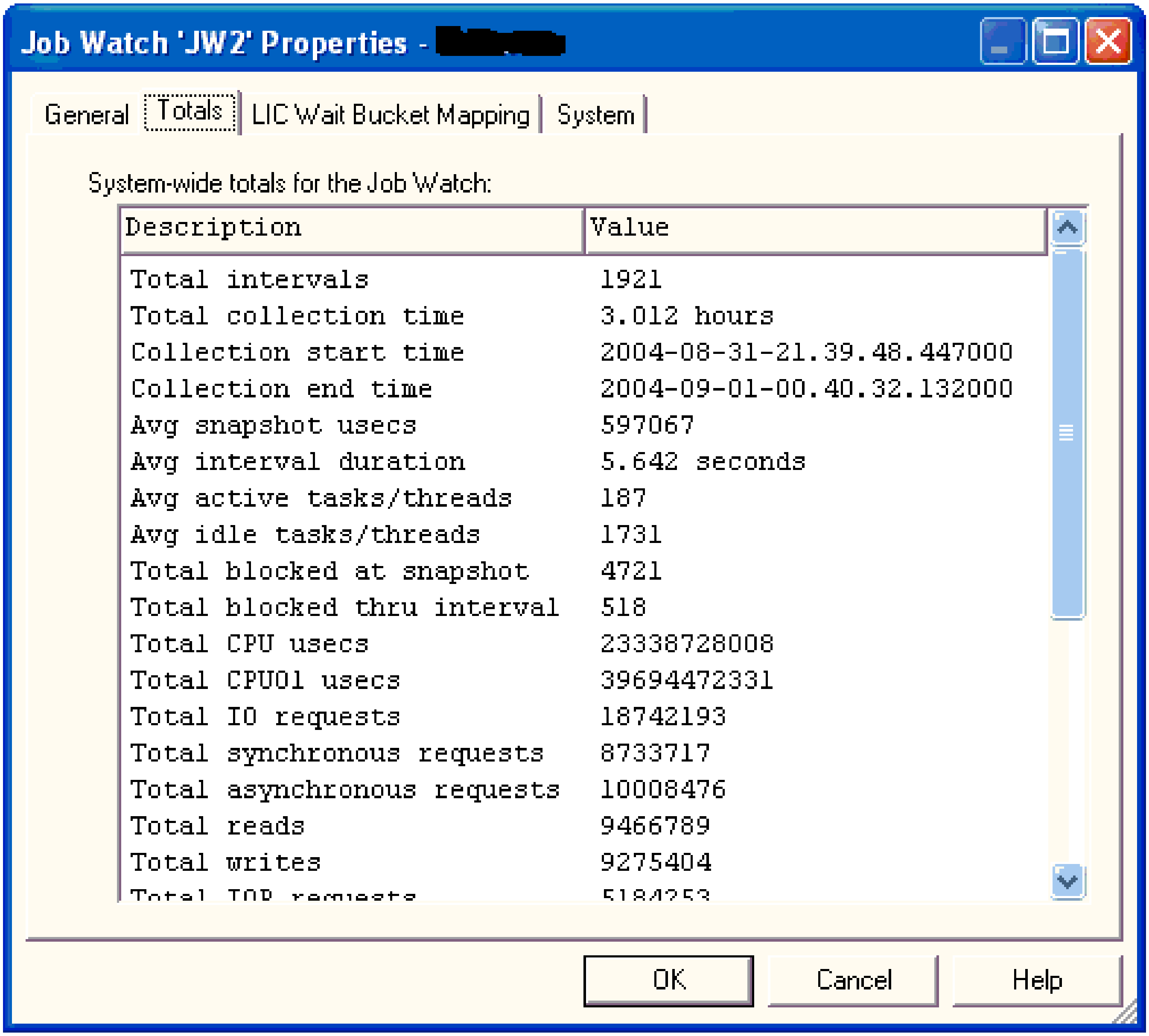
```
Selected options:
  Type:  System-wide
  Job Watch name:  SYS1
  Library:  MCCARGAR1
  Interval duration:  5
  Maximum intervals:  100
  Description:

General data collection options:
  Collect jobs and tasks
  Include all active or "badly blocked" jobs/tasks for each interval
  Maximum data to collect: 1000 megabytes

Top N "log file" QPYRTSWLG1 options:
  Top CPU users:  10
  Top DASD physical IO requests:  10
  Longest blocked waits:  10
  Most popular blocked objects/holders:  10
  Longest transactions in progress:  10

Call stack and SQL options:
  None selected
```

The information shown on this window matches the summary page of the Start Job Watch Wizard when the Job Watch was created. **Note:** This page is only shown if the job watch was originally created using the client interface.

Table of Contents | Previous | Next

2.4.3.3 Totals

The Totals property page provides a summary of many different types of performance statistics over the system-wide Job Watch. This page is only available for collections that have completed.



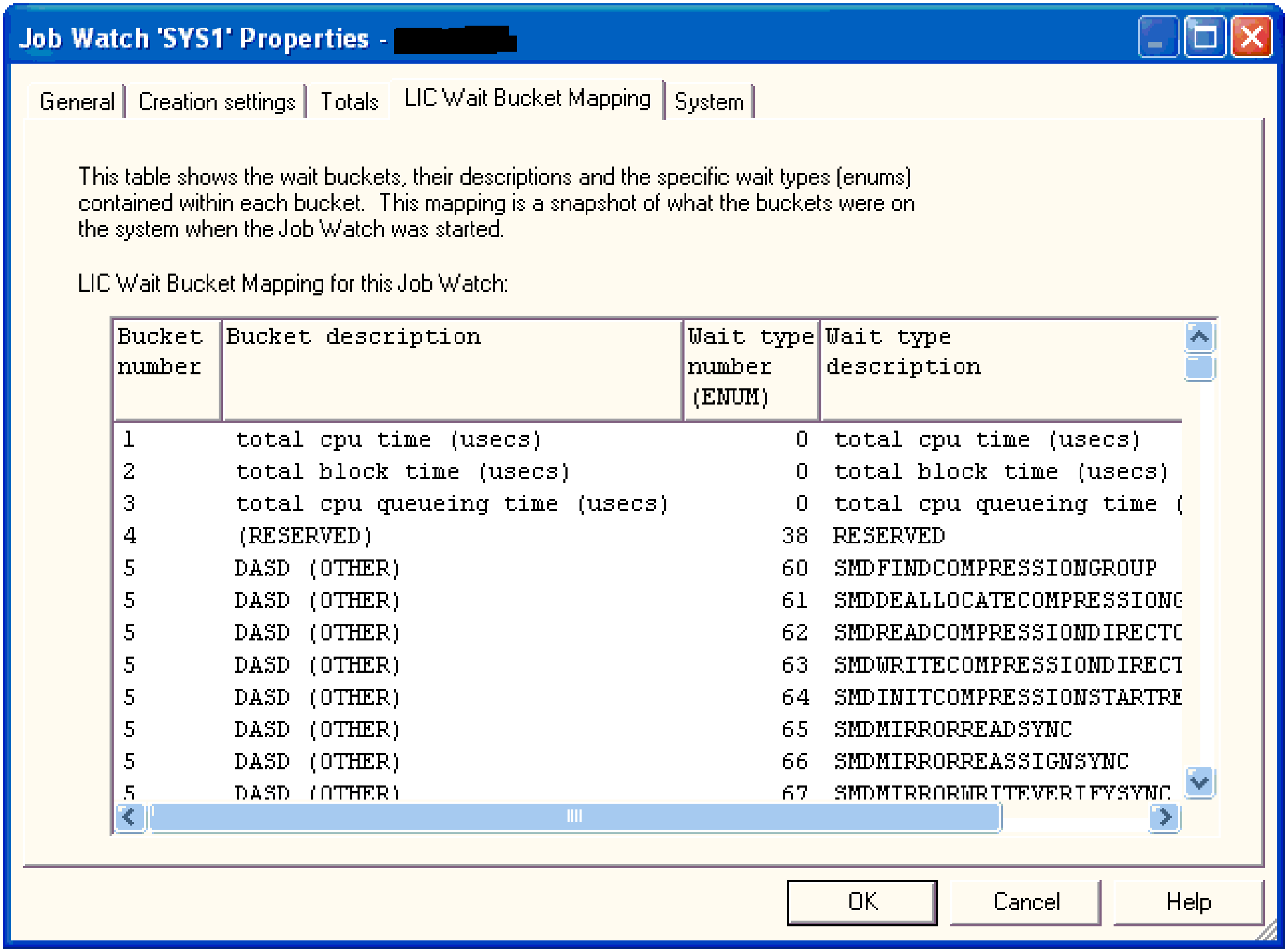
System-wide totals for the Job Watch:

Description	Value
Total intervals	1921
Total collection time	3.012 hours
Collection start time	2004-08-31-21.39.48.447000
Collection end time	2004-09-01-00.40.32.132000
Avg snapshot usecs	597067
Avg interval duration	5.642 seconds
Avg active tasks/threads	187
Avg idle tasks/threads	1731
Total blocked at snapshot	4721
Total blocked thru interval	518
Total CPU usecs	23338728008
Total CPU01 usecs	39694472331
Total IO requests	18742193
Total synchronous requests	8733717
Total asynchronous requests	10008476
Total reads	9466789
Total writes	9275404
Total IOP requests	5184253

Table of Contents Previous Next

2.4.3.4 LIC wait bucket mapping

This page shows the wait buckets and the wait points within each bucket at the time the job watch was created.



This table shows the wait buckets, their descriptions and the specific wait types (enums) contained within each bucket. This mapping is a snapshot of what the buckets were on the system when the Job Watch was started.

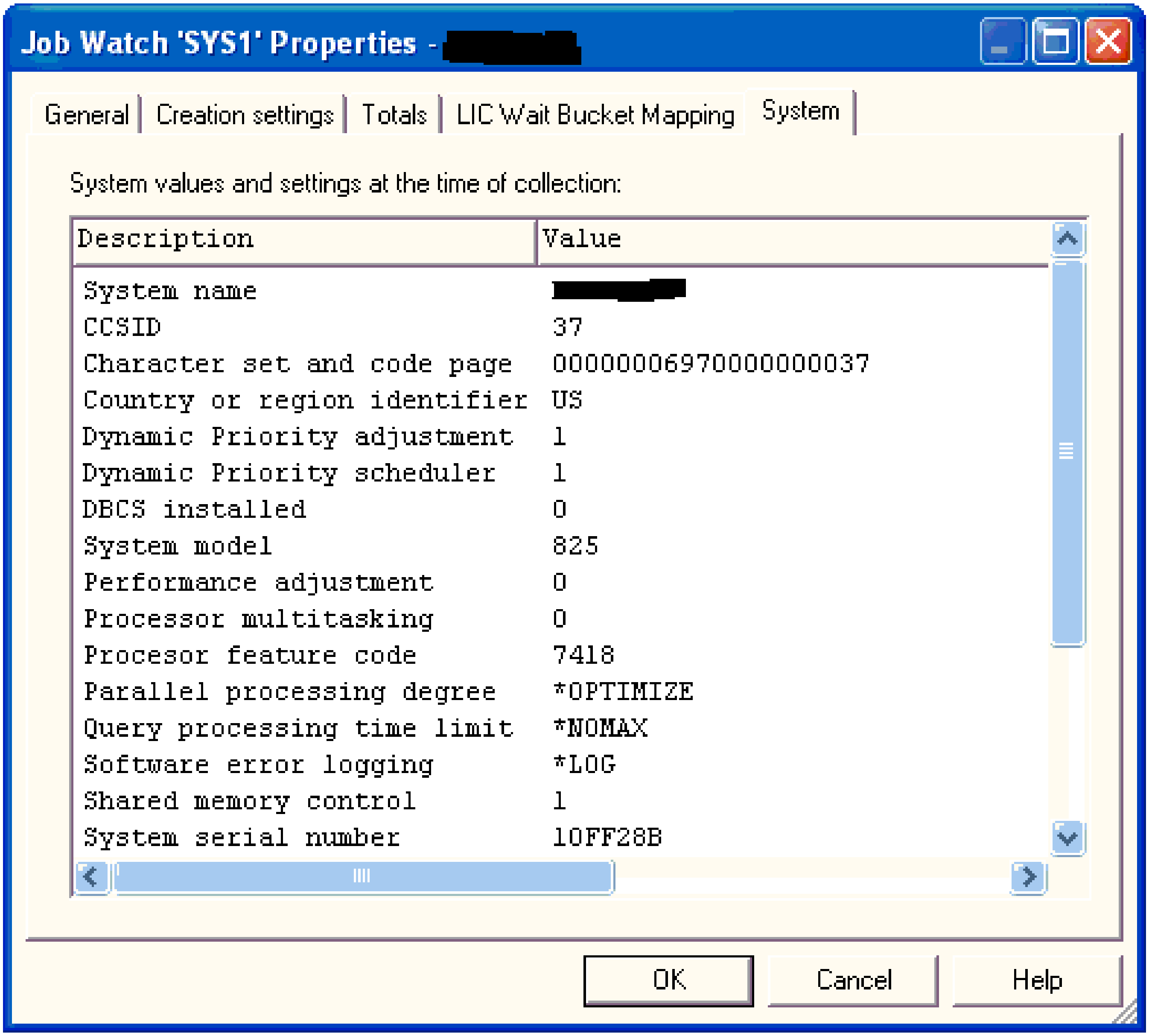
LIC Wait Bucket Mapping for this Job Watch:

Bucket number	Bucket description	Wait type number (ENUM)	Wait type description
1	total cpu time (usecs)	0	total cpu time (usecs)
2	total block time (usecs)	0	total block time (usecs)
3	total cpu queueing time (usecs)	0	total cpu queueing time (
4	(RESERVED)	38	RESERVED
5	DASD (OTHER)	60	SMDFINDCOMPRESSIONGROUP
5	DASD (OTHER)	61	SMDDEALLOCATECOMPRESSIONG
5	DASD (OTHER)	62	SMDREADCOMPRESSIONDIRECTC
5	DASD (OTHER)	63	SMDWRITECOMPRESSIONDIRECT
5	DASD (OTHER)	64	SMDINITCOMPRESSIONSTARTRE
5	DASD (OTHER)	65	SMDMIRRORREADSYNC
5	DASD (OTHER)	66	SMDMIRRORREASSIGNSYNC
5	DASD (OTHER)	67	SMDMIRRORWRITEVERIFYSYNC

Table of Contents | Previous | Next

2.4.3.5 System

This page shows information about the system the job watch was collected on including many system values.



System values and settings at the time of collection:

Description	Value
System name	[redacted]
CCSID	37
Character set and code page	00000006970000000037
Country or region identifier	US
Dynamic Priority adjustment	1
Dynamic Priority scheduler	1
DBCS installed	0
System model	825
Performance adjustment	0
Processor multitasking	0
Procesor feature code	7418
Parallel processing degree	*OPTIMIZE
Query processing time limit	*NOMAX
Software error logging	*LOG
Shared memory control	1
System serial number	10FF28B

Table of Contents | Previous | Next

2.4.4 Creating - Start Job Watch Wizard - Job-specific mode

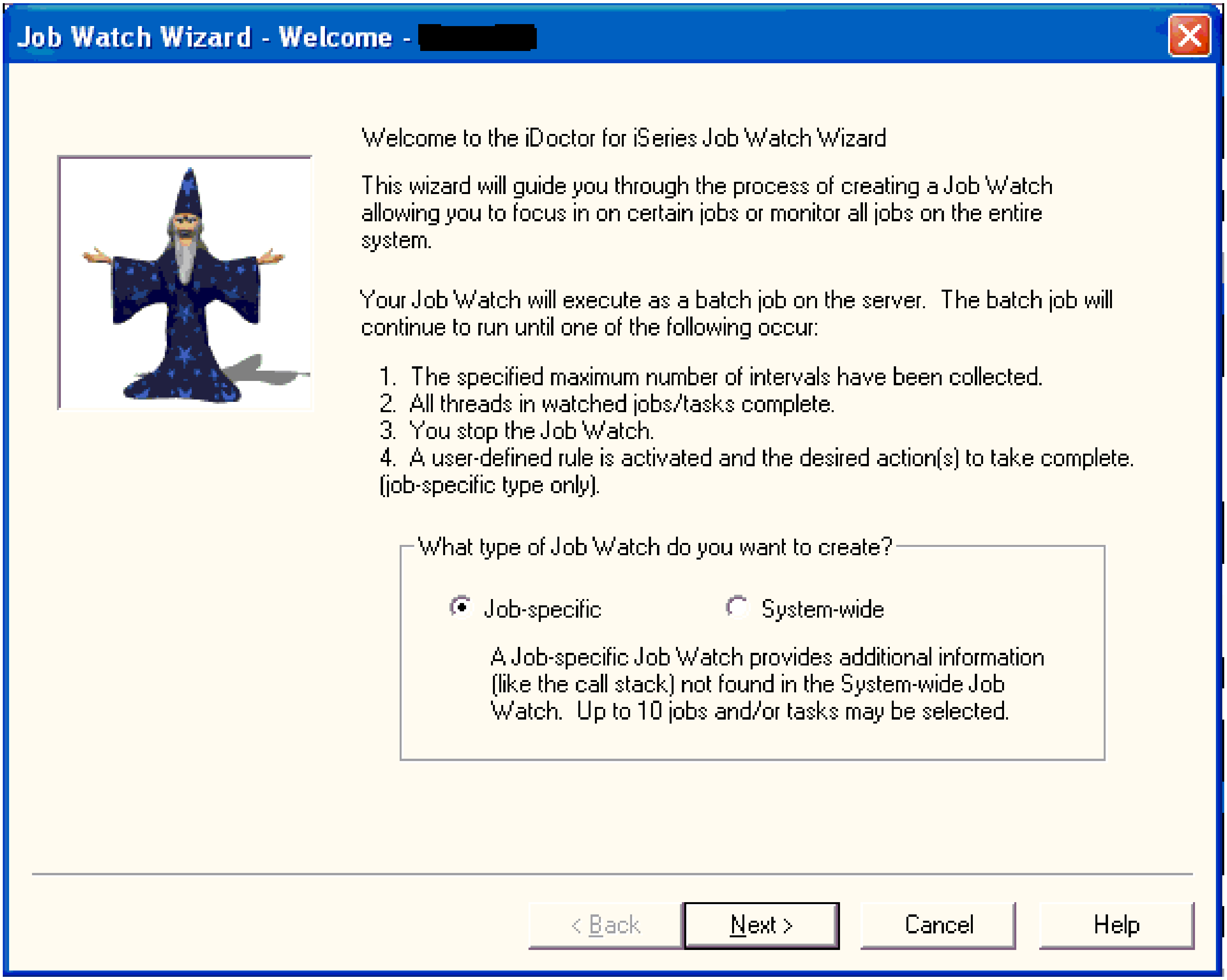
Job Watcher provides the capability to collect detailed information about any currently running job/task/thread on the system or to collect information about all jobs/tasks/threads on the system. There are two different modes of collection in Job Watcher: system-wide and job-specific.

System-wide is a good choice to use if the problem jobs on the system are unknown. If however, the jobs are known to the user that should be watched, then job-specific mode is a better choice because it provides more detailed information such as the call stacks and active SQL statements every interval of collection.

This section covers the creation of a job-specific job watch using the Wizard. To create a new Job Watch use the Start Job Watch Wizard. The Wizard is accessible via the Start Job Watch menu on the Job Watcher or library folder icons.

The Job Watch Wizard guides the user step by step through the process of creating the Job Watch. Each page is covered in detailed within the next sections.

An example of the Wizard is shown below:



Job Watch Wizard - Welcome - [redacted]

Welcome to the iDoctor for iSeries Job Watch Wizard

This wizard will guide you through the process of creating a Job Watch allowing you to focus in on certain jobs or monitor all jobs on the entire system.

Your Job Watch will execute as a batch job on the server. The batch job will continue to run until one of the following occur:

1. The specified maximum number of intervals have been collected.
2. All threads in watched jobs/tasks complete.
3. You stop the Job Watch.
4. A user-defined rule is activated and the desired action(s) to take complete. (job-specific type only).

What type of Job Watch do you want to create?

● Job-specific ○ System-wide

A Job-specific Job Watch provides additional information (like the call stack) not found in the System-wide Job Watch. Up to 10 jobs and/or tasks may be selected.

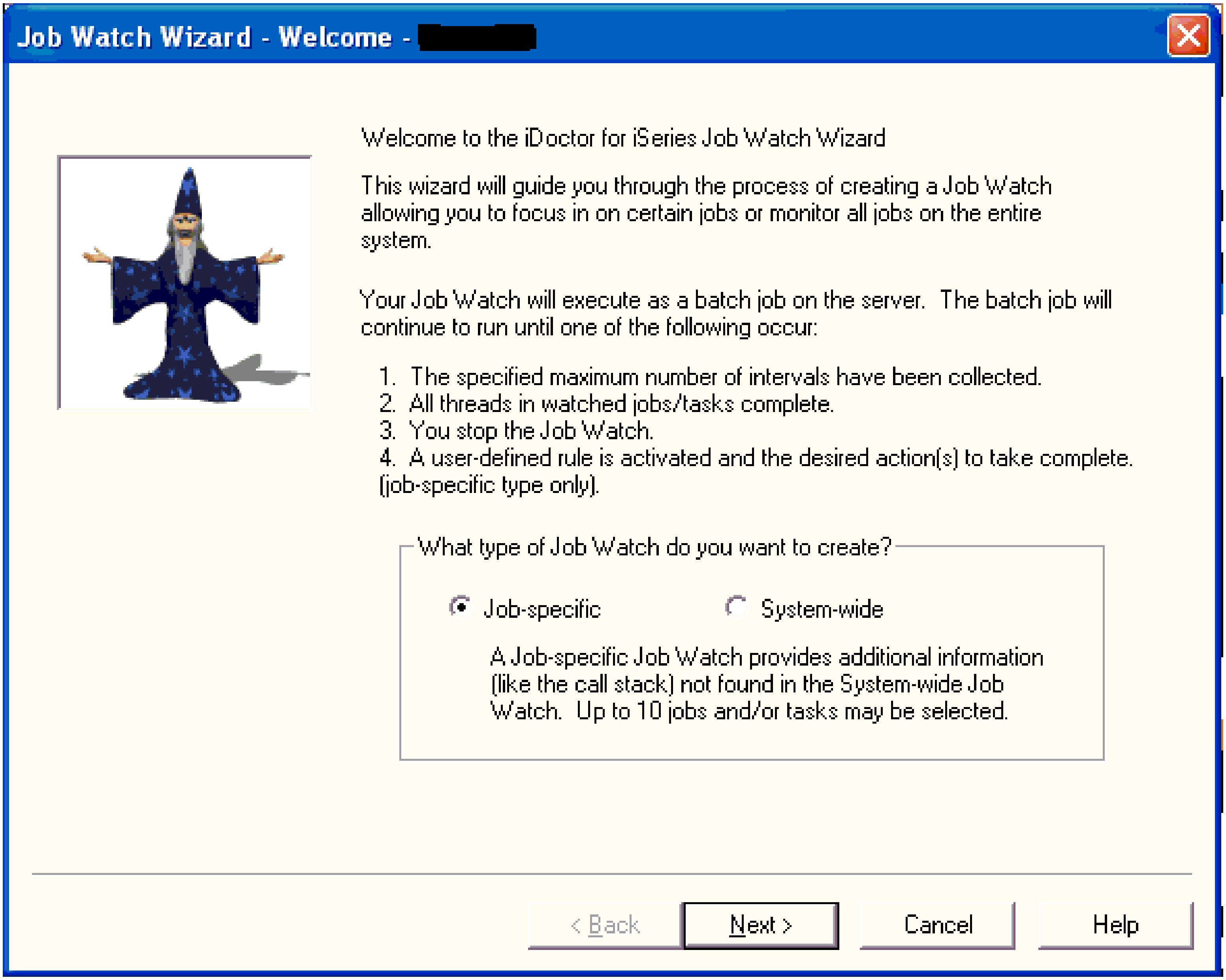
[The Start Job Watch Wizard]

Table of Contents | Previous | Next

2.4.4.1 Welcome

The Welcome page in the Job Watch Wizard introduces the user to the wizard and offers information about what the wizard will do. The page also explains for what conditions the Job Watch will end execution.

On this window the user must specify the type of Job Watch to create. System-wide is a good choice for first time users or if the job names that should be analyzed are unknown. Job-specific is better when the jobs that are running poorly are known to the user poorly because it can provide additional information that system-wide cannot.



Job Watch Wizard - Welcome - [redacted]

Welcome to the iDoctor for iSeries Job Watch Wizard

This wizard will guide you through the process of creating a Job Watch allowing you to focus in on certain jobs or monitor all jobs on the entire system.

Your Job Watch will execute as a batch job on the server. The batch job will continue to run until one of the following occur:

1. The specified maximum number of intervals have been collected.
2. All threads in watched jobs/tasks complete.
3. You stop the Job Watch.
4. A user-defined rule is activated and the desired action(s) to take complete. (job-specific type only).

What type of Job Watch do you want to create?

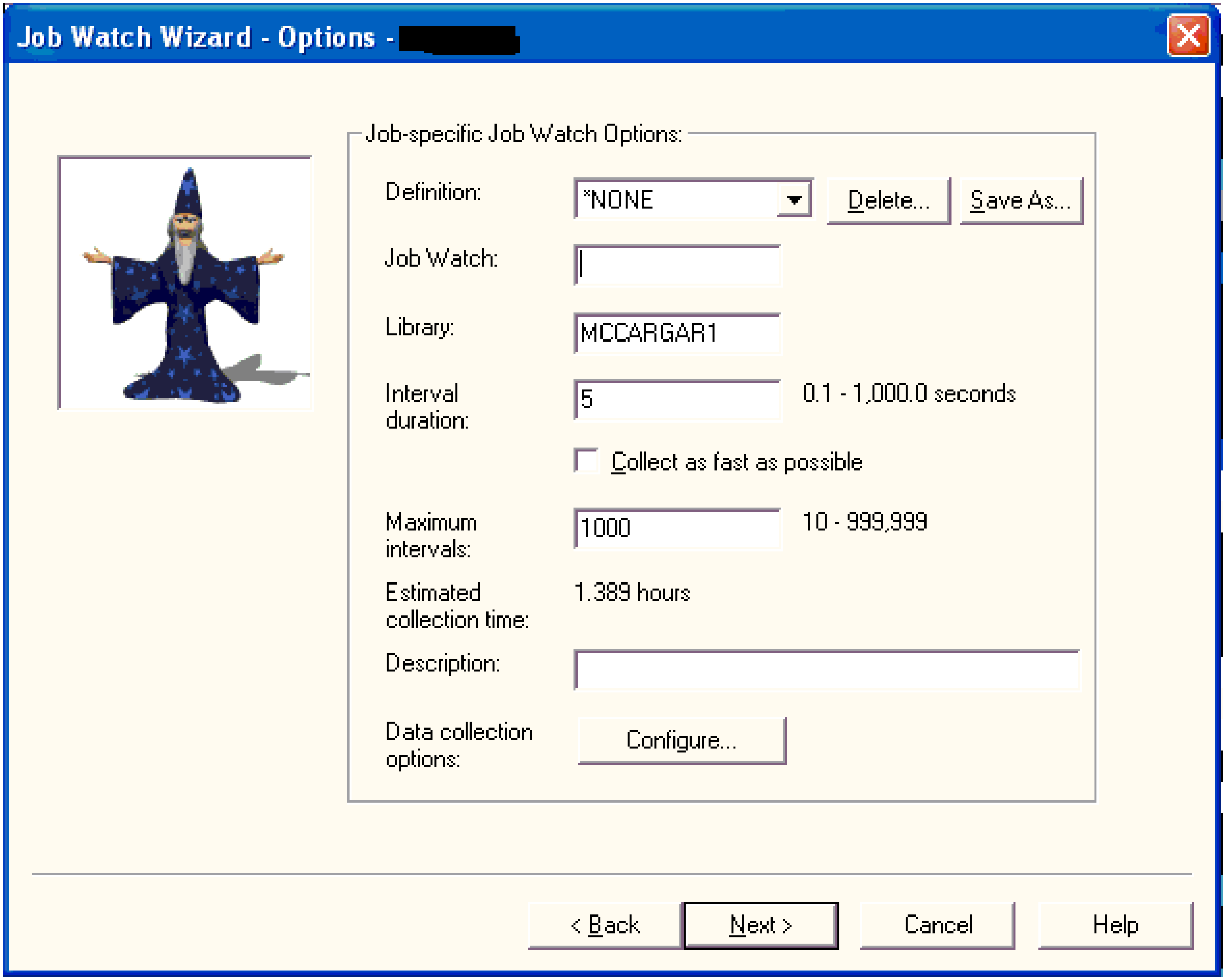
● Job-specific ○ System-wide

A Job-specific Job Watch provides additional information (like the call stack) not found in the System-wide Job Watch. Up to 10 jobs and/or tasks may be selected.

[Job Watch Wizard - Welcome Page]

2.4.4.2 Options

The Options Page allows the user to specify the most basic pieces of information about a Job Watch such as its name, library, duration and description. The following is an example of this page of the Start Job Watch Wizard.



Job Watch Wizard - Options - [redacted]

Job-specific Job Watch Options:

Definition: *NONE

Job Watch:

Library: MCCARGAR1

Interval duration: 5 0.1 - 1,000.0 seconds

□ Collect as fast as possible

Maximum intervals: 1000 10 - 999,999

Estimated collection time: 1.389 hours

Description:

Data collection options:

[Job Watch Wizard - Options Page]

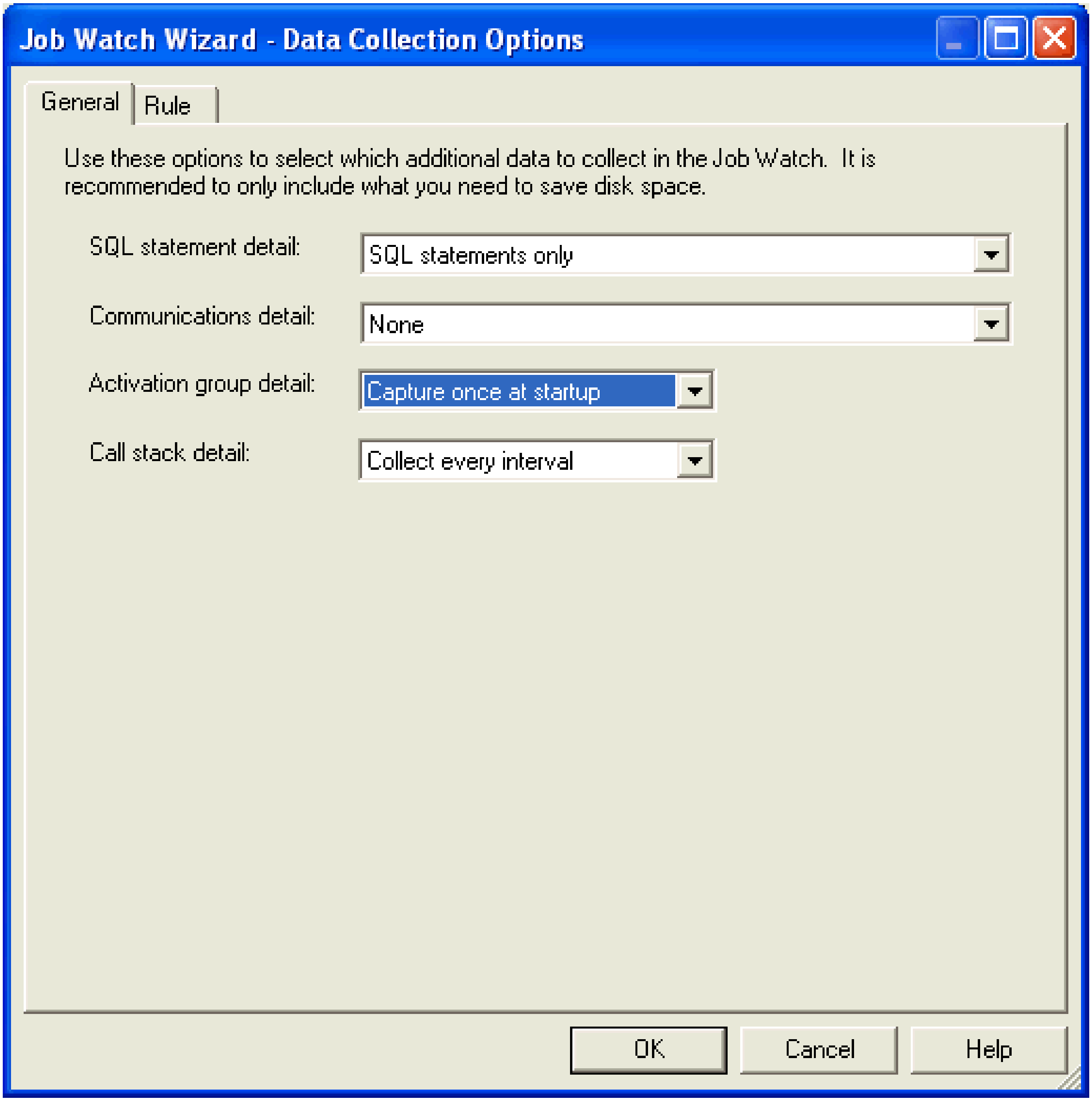
The following table provides details about each of the parameters on this page:

Field	Description
Definition	The name of the definition to create the job watch with. A Job Watcher definition is a set of parameter settings that can be used to more quickly set up and run multiple Job Watches having the same data collection options. A definition can be created by using the Save As... button on the options page once the desired data collection options have been set on the data collection options section of the Wizard. By selecting a definition from the list of definitions, the parameters for the definition will be loaded into the Wizard for use.
Job Watch	The name of the Job Watch (10 chars max). This name matches the member name used when creating the output files on the server. The output file names start with QPYRT* and will exist in the library specified on this page of the Wizard.
Library	The name of the library to create the Job Watch in. If the library does not exist, the client will ask for confirmation before creating it.
Interval duration	The size of each sample of data in seconds. For job-specific mode this field is not available if the "collect as fast as possible" option is selected.
Collect as fast as possible	Select this box to set the delay between intervals to zero. **Warning:** Using this option with a very high maximum intervals value will generate a lot of data very quickly, and as a result could stop your system if you are already low on disk space.
Maximum intervals	The total number of intervals to collect data for.
Estimated collection time	The total estimated time the collection will run if it is not stopped prematurely based on the interval duration and maximum intervals parameter values provided.
Description	A description to give the Job Watch being created.
Data collection options button	Click this button to select additional types of information to collect.

Table of Contents | Previous | Next

2.4.4.3 Data collection options - General page

This page allows the user to specify which additional types of information should be collected about each job/task/thread in the job watch. The types of optional data available for a job-specific job watch are: SQL statements, communications, activation groups, call stacks.



Use these options to select which additional data to collect in the Job Watch. It is recommended to only include what you need to save disk space.

SQL statement detail: SQL statements only

Communications detail: None

Activation group detail: Capture once at startup

Call stack detail: Collect every interval

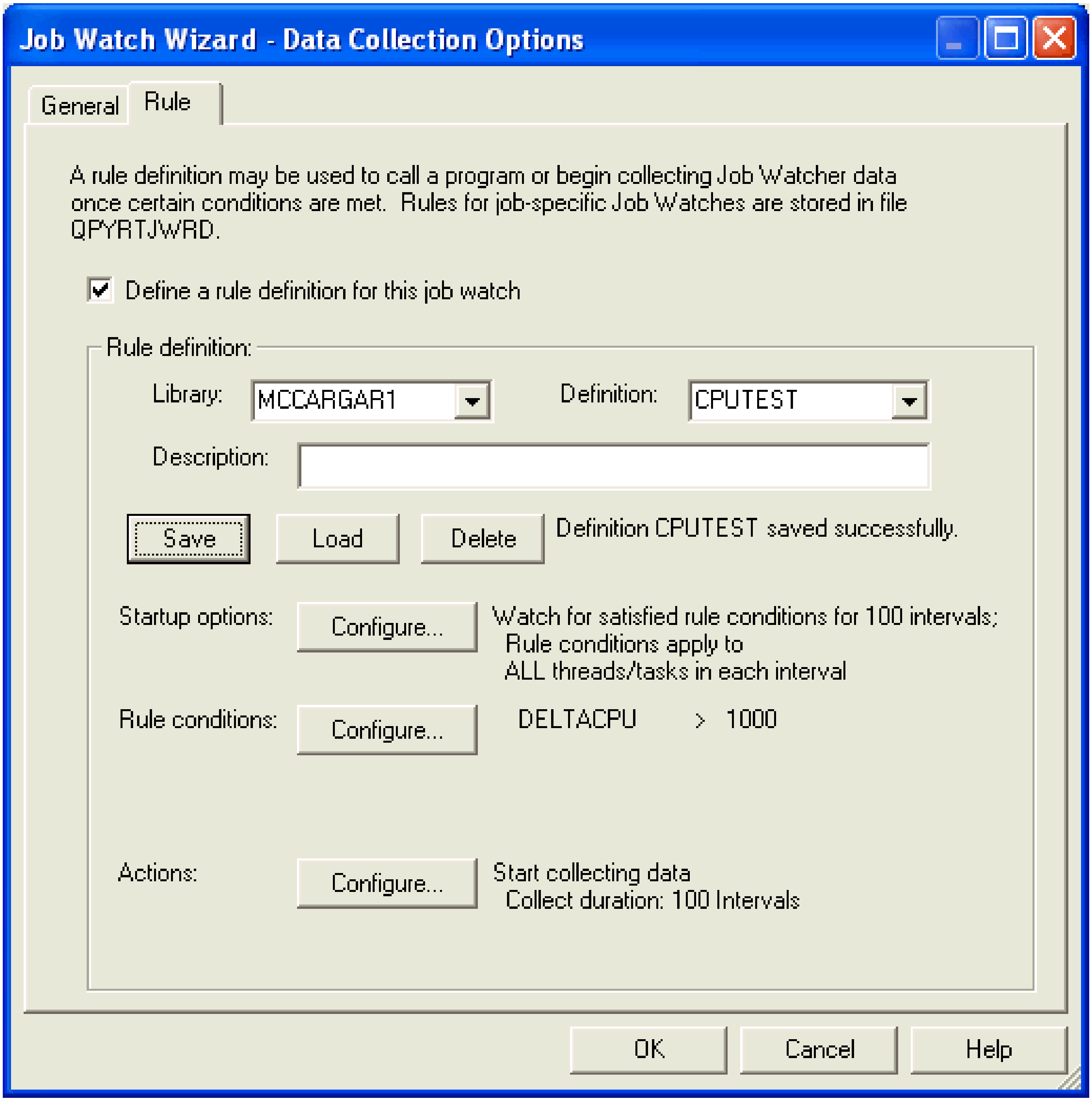
[Job Watch Wizard - Data Collection Options Page]

The following table describes the parameters available on this page of the Wizard. The default values for each parameter are **bold**.

Collection Option	**Description**
SQL statements	Possible values are: **None** SQL statements only - The last (or currently running) SQL statement executed per job is captured every interval of the job watch. SQL statements and host variables - The last (or currently running) SQL statement executed along with its host variables is captured every interval of the job watch. Statements, host variables, prepared statements, open cursors - All available SQL information is captured per job every interval.
Communications	Possible values are: **None** Communications socket and IP information - details about all open socket connections for all jobs being watched in the job watch. Communications detail and related job information - contains the same information as the previous option but also includes an extra file identifying the job associated with each socket.
Activation groups	Possible values are: **None** Capture once at startup - activation group details are captured for each job for interval 1 only. Capture every interval - activation group details captured for each job every interval. Capture every Nth interval - activation group details captured for each job every Nth interval. The value for N can be filled in within the text box provided next to the list.
Call stacks	Possible values are: None **Capture every interval** - call stacks captured for each job/task/thread every interval. Capture every Nth interval - call stacks captured for each job/task/thread every Nth interval. The value for N can be filled in within the text box provided next to the list.

2.4.4.4 Data collection options - Rule page

This page allows the user to define or reuse a rule definition to use during execution of the job-specific job watch. A rule definition can be used to call a program or delay collection of data until specific conditions in the data are met.



A rule definition may be used to call a program or begin collecting Job Watcher data once certain conditions are met. Rules for job-specific Job Watches are stored in file QPYRTJWRD.

☑ Define a rule definition for this job watch

Rule definition:

Library: MCCARGAR1

Definition: CPUTEST

Description:

Definition CPUTEST saved successfully.

Startup options: Watch for satisfied rule conditions for 100 intervals;
Rule conditions apply to
ALL threads/tasks in each interval

Rule conditions: DELTACPU > 1000

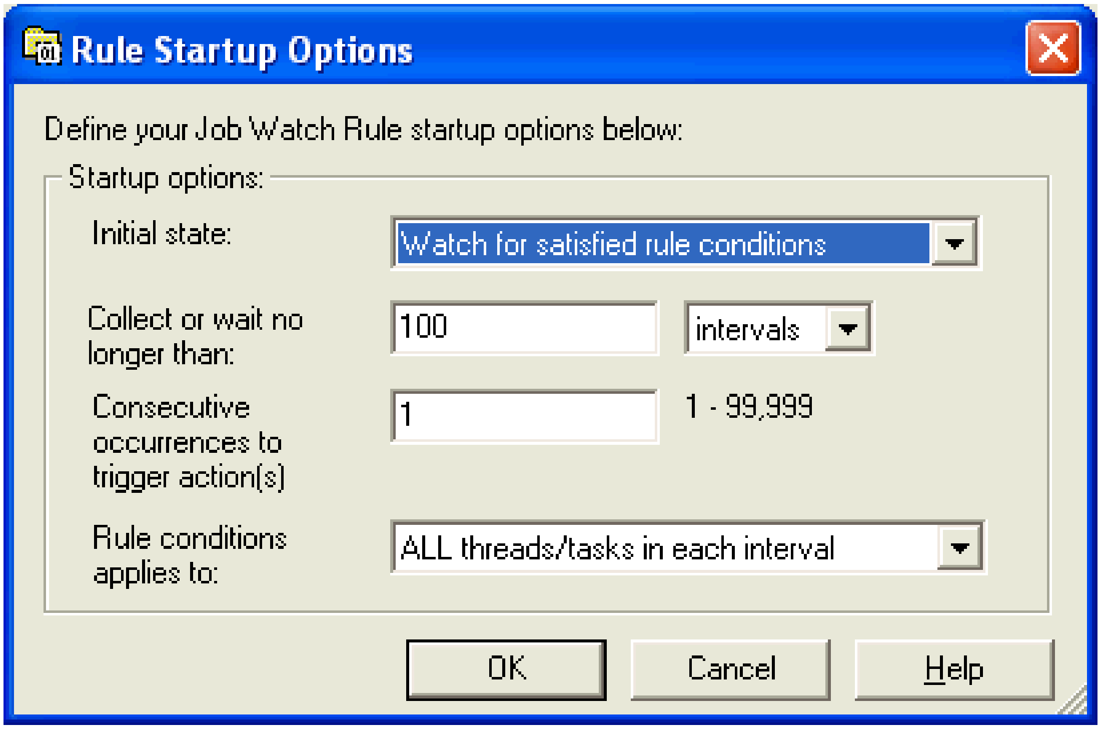
Actions: Start collecting data
Collect duration: 100 Intervals

The following table describes the parameters available on this page of the Wizard.

Option	Description
Define a rule checkbox	This option can be used to quickly disable/reenable use of a rule definition for the current job watch. If a rule has been defined in the current session, unselecting the checkbox will cause the rule definition to not to be used when the job watch is submitted.
Library	The name of the library the rule definition is located in. The library list box contains all libraries on the system where file QPYRTJWRD (the rule definition file for job-specific job watches) exists.
Definition	The name of the rule definition. This name matches the member name in file QPYRTJWRD in the library specified. After a library is selected, clicking the arrow on the definition drop down list will show all members that exist in file QPYRTJWRD in the library specified. Select a library/member from the lists if a rule definition needs to be loaded into the wizard or deleted using the Load, Delete buttons on the page.
Description	The description of the rule definition. Type in a description if desired to associate with the rule definition being created.
Save button	Saves the current settings on this page of the wizard to the library/definition name indicated.
Load button	Loads the rule definition in the library/definition name indicated. The current settings on the rule page will be replaced with the information obtained from the rule definition specified.
Delete button	Deletes the rule definition in the library/definition name indicated.

Startup options

Clicking on the startup options button will display the following interface for configuring the rule definition:



Define your Job Watch Rule startup options below:

Startup options:

Initial state: Watch for satisfied rule conditions

Collect or wait no longer than: 100 intervals

Consecutive occurrences to trigger action(s): 1 1 - 99,999

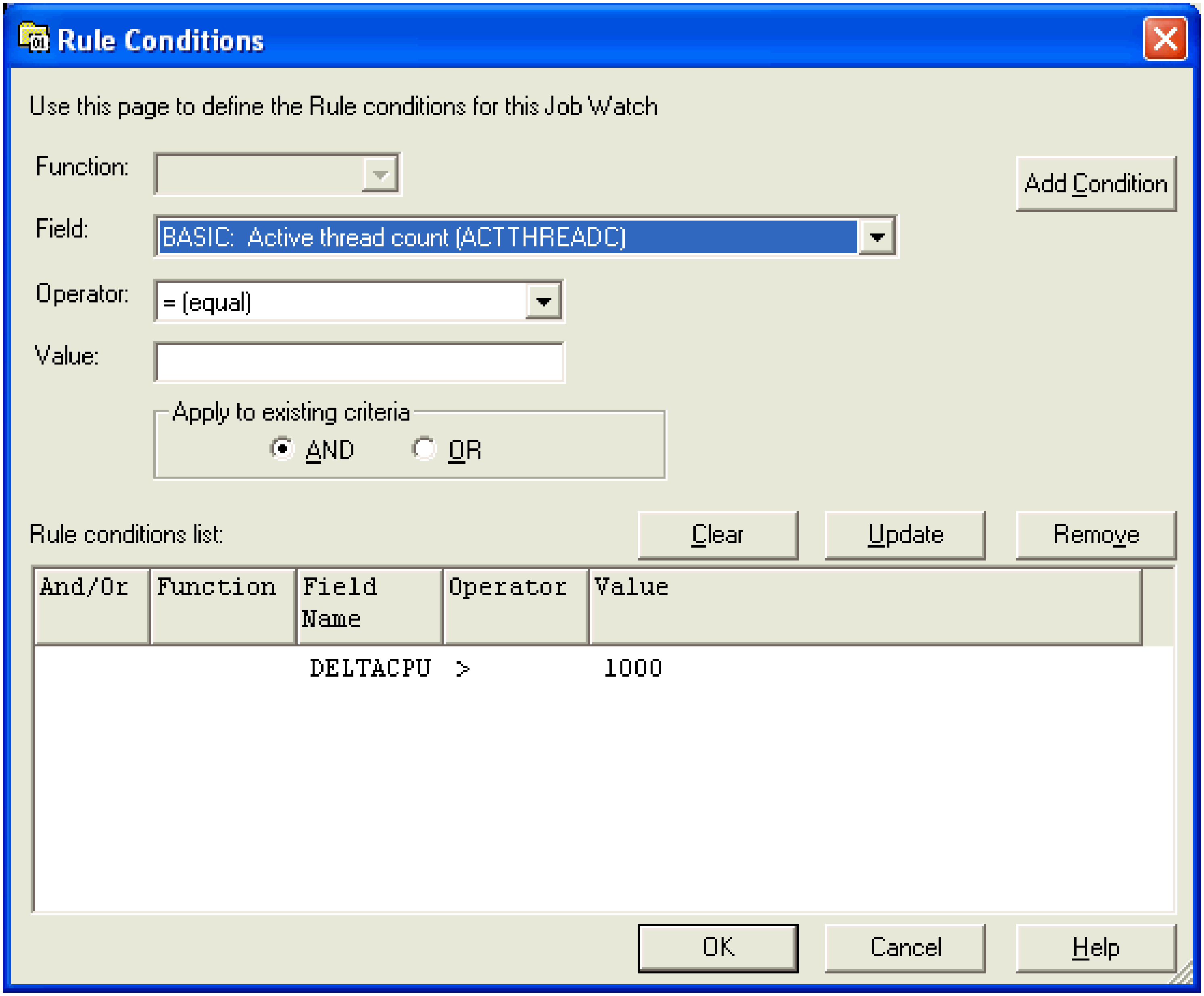
Rule conditions applies to: ALL threads/tasks in each interval

The following table describes the startup options available:

Option	**Description**
Initial state	Indicates whether the job watch is collecting data when the collection is started. If the job watch is collecting data at startup then the only "action" allowed is to call a program.
Collect or wait no longer than	Indicates how many intervals (or seconds) the job watch should look for the rule conditions to be met before ending. If the rule condition is never met this value indicates how long the job watch will run.
Consecutive occurrences to trigger action(s)	Indicates how many consecutive intervals the rule conditions must be met before the action is fired.
Rule conditions applies to	Indicates if the rule conditions must apply to ALL threads/tasks in the job watch or ANY thread/task in the job watch. For example if ALL threads is used for this value and the condition is CPU > 1000 and some jobs exceed this amount of CPU but other jobs don't, the rule condition will NOT be met. If this value is ANY threads under the same circumstances then the condition will be met.

Rule conditions

Clicking on the rule conditions button will display the following interface for configuring the rule definitions conditions to check for.



Rule Conditions

Use this page to define the Rule conditions for this Job Watch

Function:

Field: BASIC: Active thread count (ACTTHREADC)

Operator: = (equal)

Value:

Apply to existing criteria

● AND ○ OR

Rule conditions list:

And/Or	Function	Field Name	Operator	Value
		DELTACPU	>	1000

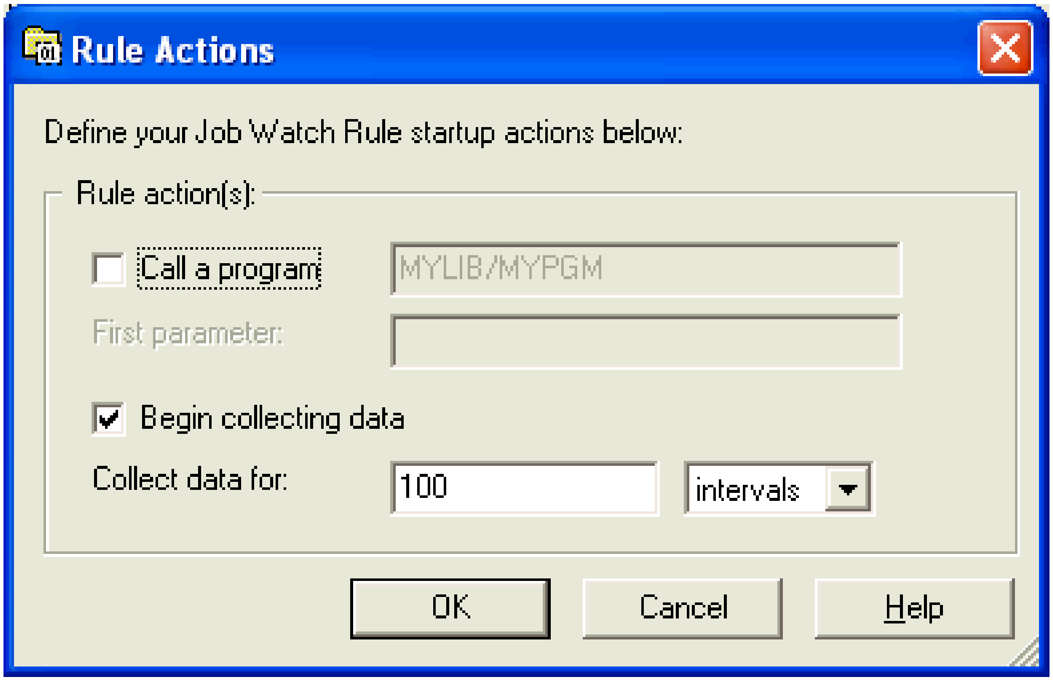
The following table describes the options available within this interface:

Option	Description
Function	This drop down list can be used to apply a function to the selected field when the Add Condition button is pressed. If the selected field changes the functions available in the list will be updated. Some functions are only available when certain fields are selected. The list of functions and fields available is in the example rule conditions file QPYRTJWRD in library QPYRTJW.

Field	The list of possible fields the rule definition may be defined over. Most (but not all) of these fields exist in files QPYRTJWA and QPYRTJWAPR. The fields are organized by category. The categories are: Basic - Disk/IO fields, performance counters, current user profile Buckets - The 32 wait bucket counts and times Call stack - A special case field which can be used to setup a rule definition that allows a search on a specific program/procedure/module name in the call stack. Comm - communications statistics Database - SQL statement or LDIO fields Exceptions - numeric data type exceptions Holder - fields related to the holding job IFS - IFS statistics Java - Java heap sizes and thread name fields Other - single level store stack size State transitions - active to wait, active to ineligible, wait to ineligible Transactions - transaction counters and times Wait object - fields related to the wait object
Operator	The types of operations that can be performed on the selected field. The list of operators is automatically updated everytime the selected field changes.
Value	The numeric or character value that will be compared against the field using the operator specified.
And/Or	Indicates if the condition added next should have an AND or OR applied to the beginning of it. This can be used to set up a conditional expression such as CPU > 1000 OR WAITOBJ = 'FRED'
Add Condition	Adds the selected field, operator, value and And/Or value to the list of conditions. If the condition being added is the first one in the list, the And/Or value is not visible.
Clear button	Removes all entries from the rule conditions list.
Update button	Updates the selected entry in the rule conditions list with the selected field, operator and value on this window.
Remove button	Removes all selected entries from the rule conditions list.
Rule conditions list	The list of conditions which will be compared against the job watcher data to see if the conditions are met. If the conditions are met the specified actions on the actions window will be triggered.

Actions

Clicking on the actions button will display the following interface for configuring the actions to fire if the rule conditions are met:


Rule Actions

Define your Job Watch Rule startup actions below:

Rule action(s):

□ Call a program: MYLIB/MYPGM

First parameter:

☑ Begin collecting data

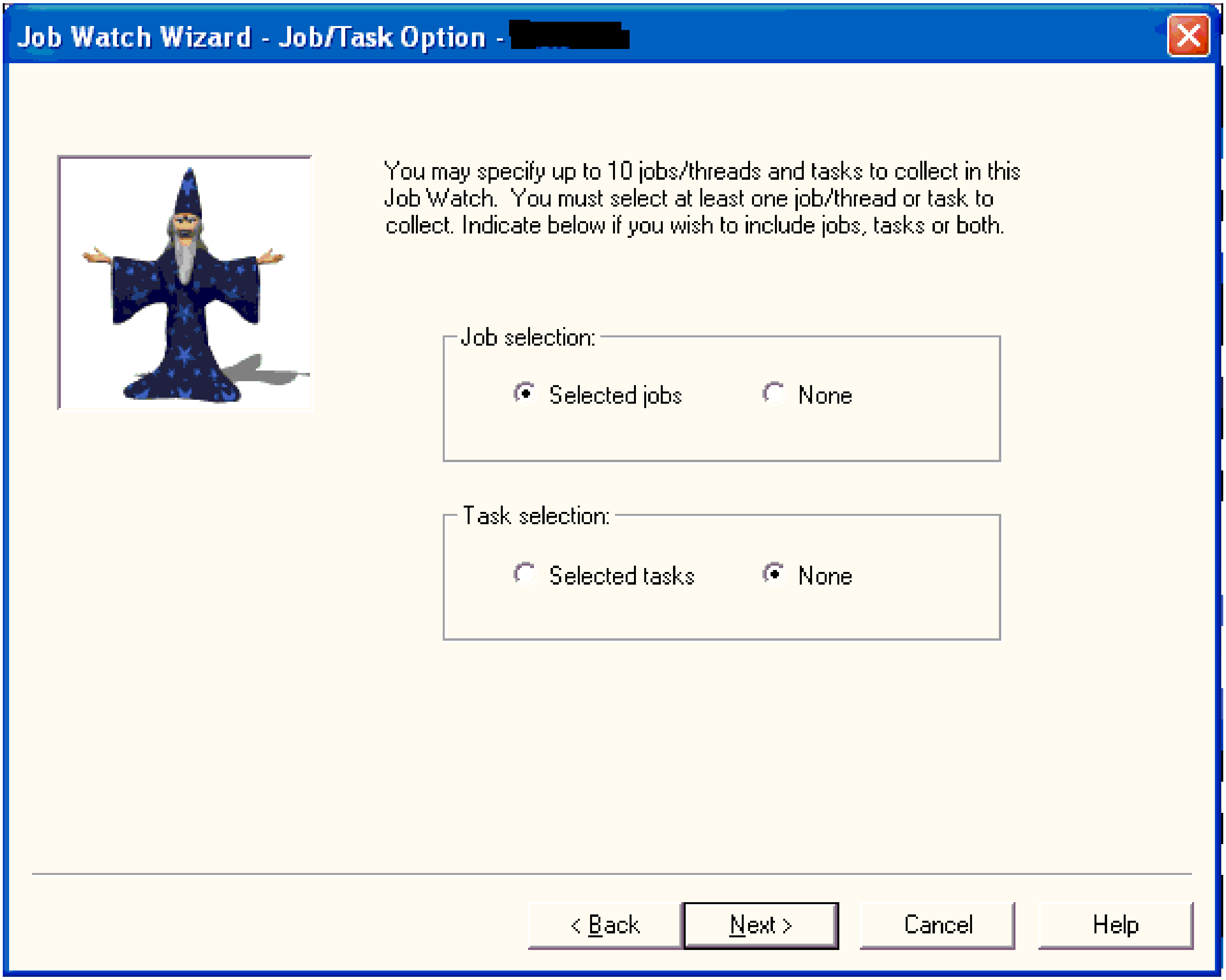
Collect data for: 100 intervals

The following table describes the options available in this window:

Option	Description
Call a program	Indicates the library and program name to call if the rule conditions are met. An up to 50 character string will be passed to the program. In order for this program call to work, the program must have 7 parameters and have the following layout: 1 - user defined character string (char 50) 2 - library name of the job watch session (char 10) 3 - member name of the job watch session (char 10) 4 - the job or task name that satisfied the conditions (char 32 - matches the NAME field in file QPYRTJWA) 5 - the thread id of the job that satisfied the conditions (packed (20,0)) 6 - the time of day the program was called (char 28) 7 - the interval number (packed (20,0))
Collect data	Indicates how long to collect data for once the rule conditions are met. This value may be specified in intervals or seconds and is only available if the initial state on the startup options window is set to "Wait for rule conditions". If this value is specified the maximum intervals parameter on the options page is ignored.

2.4.4.5 Job/Task Option

This page lets the user determine if jobs or tasks should be collected (or both). The selections indicated on this page determine what pages come next in the wizard. The job selection page and/or the task selection page will be displayed after clicking the 'Next' button. The value of None cannot be specified for both job selection and task selection.



Job Watch Wizard - Job/Task Option - [redacted]

You may specify up to 10 jobs/threads and tasks to collect in this Job Watch. You must select at least one job/thread or task to collect. Indicate below if you wish to include jobs, tasks or both.

Job selection:
● Selected jobs ○ None

Task selection:
○ Selected tasks ● None

[Job Watch Wizard - Job/Task Option Page]

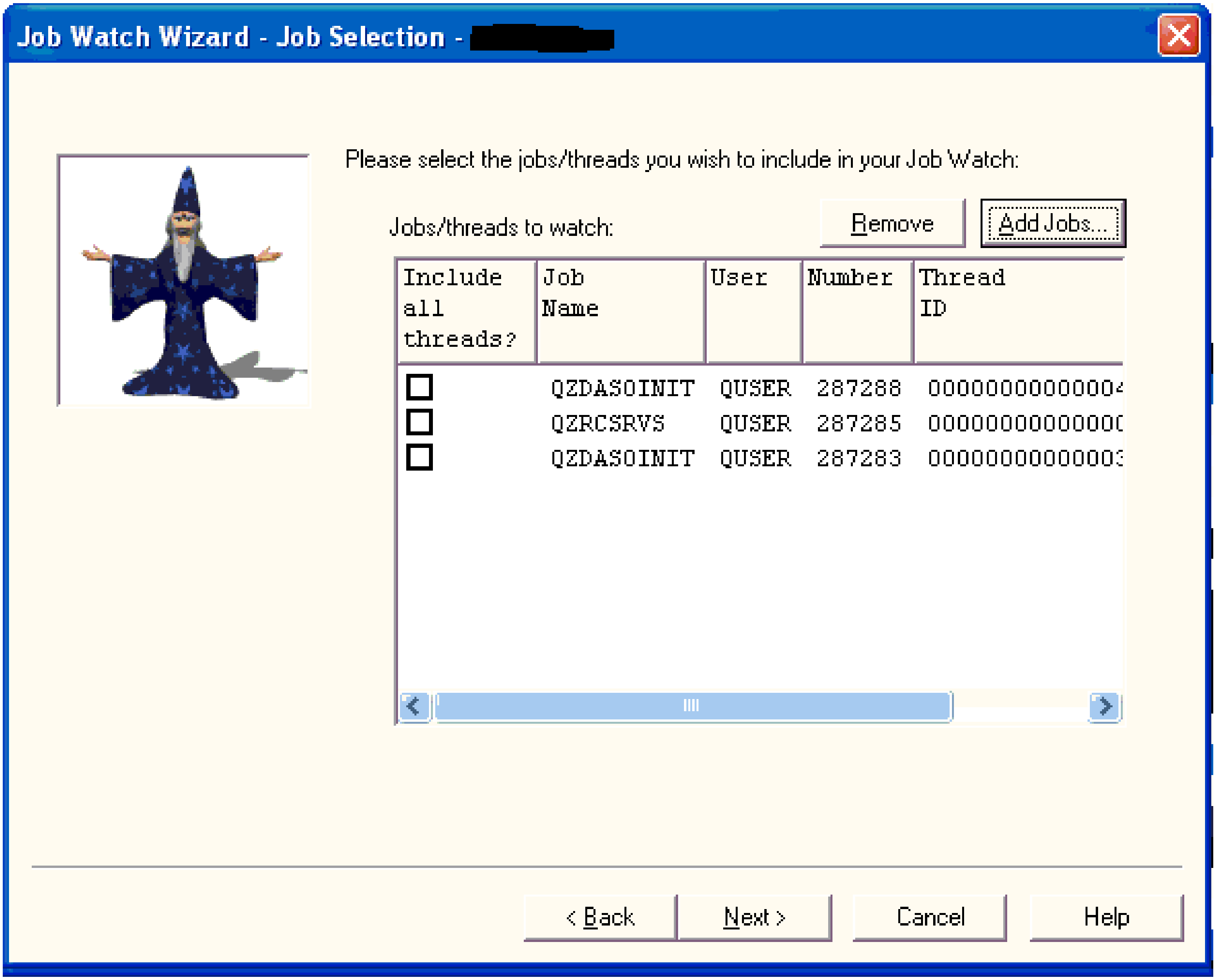
Table of Contents Previous Next

2.4.4.6 Job Selection

The job selection page lets the user add or remove jobs/threads to the watch. There is a limit of 10 jobs/threads per Job Watch, but if primary threads are selected an option is available to also collect statistics for all secondary threads within the primary thread. A maximum of 500 threads can be collected each interval during the job watch. If this limit is exceeded only the 1st 500 threads each interval are captured and the total missing threads each interval are noted in the missed threads file QPYRTJWAMT.

The following table provides information about each of the criteria on this page:

Field	**Description**
Remove button	Removes selected jobs from the list
Add Jobs button	Displays the Add Jobs window. From this window the user can select which jobs (among the curently active jobs) to include in the job watch.
Job/threads to collect list	List of jobs/threads to collect. Select the "Include all threads?" field in order to collect all threads if the job/thread listed is a primary thread. The combined total of jobs and tasks from the job selection and task selection pages can be no more than 10 entries.



Job Watch Wizard - Job Selection - [redacted]

Please select the jobs/threads you wish to include in your Job Watch:

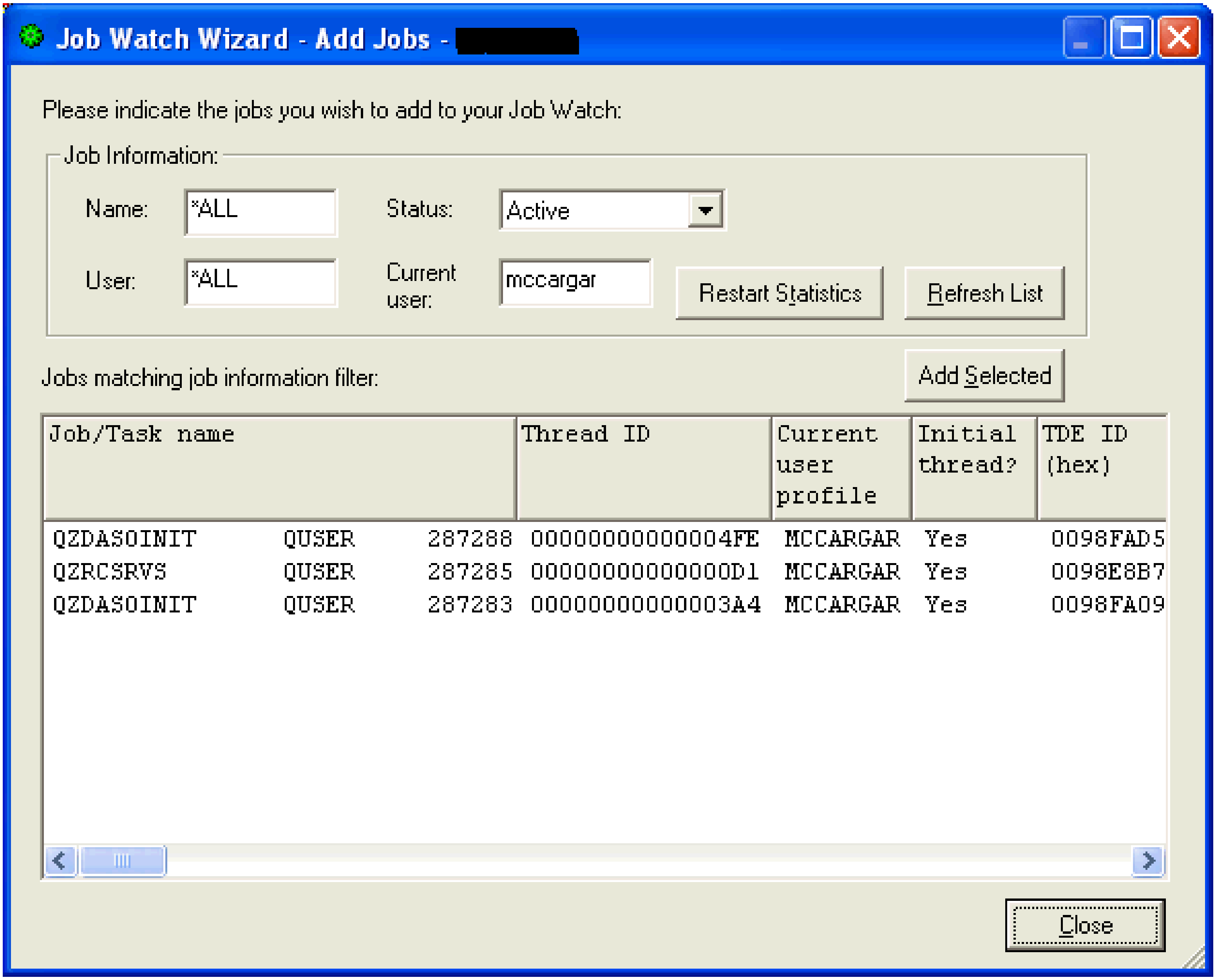
Jobs/threads to watch:

Include all threads?	Job Name	User	Number	Thread ID
□	QZDASOINIT	QUSER	287288	00000000000004
□	QZRCSRVS	QUSER	287285	00000000000000
□	QZDASOINIT	QUSER	287283	00000000000003

[Job Watch Wizard - Job Selection Page]

Add Jobs Window

The add jobs window is displayed when the job selection page is first opened. This window allows the user to select which jobs/threads should be included in the job watch. This screen also provide the ability to display numerous statistics (waits, IOs, cpu) about the jobs specified in the job information filter. An example of the Add Jobs window is shown below:



Job Watch Wizard - Add Jobs - [redacted]

Please indicate the jobs you wish to add to your Job Watch:

Job Information:

Name: *ALL

Status: Active

User: *ALL

Current user: mccargar

Jobs matching job information filter:

Job/Task name	Thread ID	Current user profile	Initial thread?	TDE ID (hex)
QZDASOINIT QUSER 287288	00000000000004FE	MCCARGAR	Yes	0098FAD5
QZRCSRVS QUSER 287285	00000000000000D1	MCCARGAR	Yes	0098E8B7
QZDASOINIT QUSER 287283	00000000000003A4	MCCARGAR	Yes	0098FA09

[Job Watch Wizard - The Add Jobs window for the Job Selection Page]

The following table provides information about each of the fields on this page:

Field	Description
Name	Indicates the generic job name that should be used when the list of jobs is next refreshed.
User	Indicates the generic user name that should be used when the list of jobs is next refreshed.
Current user profile	The name of the current user profile to filter the list of jobs by when the list of jobs is next refreshed. If this value is blank, all jobs matching the rest of the job filter are included.
Status	Indicates the status of jobs that should be included in the list of jobs. The possible values are Active and On Job Queue. This option can be used to start a job watch over one or more jobs that have not started yet and are waiting on a job queue.
Restart Statistics	The list of jobs is produced by performing two snapshots over the jobs matching the job information filter. This option indicates to refresh the list by producing two immediate snapshots and comparing the values between the two snapshots to produce a delta.

Refresh List	The list of jobs is produced by performing a computation from two snapshots over the jobs matching the job information filter. This option indicates that only 1 snapshot should be taken and the first snapshot taken in this window (or since the last use of "refresh statistics") should be used to make the comparison.
Jobs matching job information filter	This is the list of jobs that meet the job information filter as produced by the last refresh. This list also contains many performance statistics which can be used to get an indication of which jobs on the systems are performing the most IO, using the most CPU, etc and are good candidates to be watched by Job Watcher.
Add selected	This button will add the selected jobs from the list to the job selection page of the Wizard.

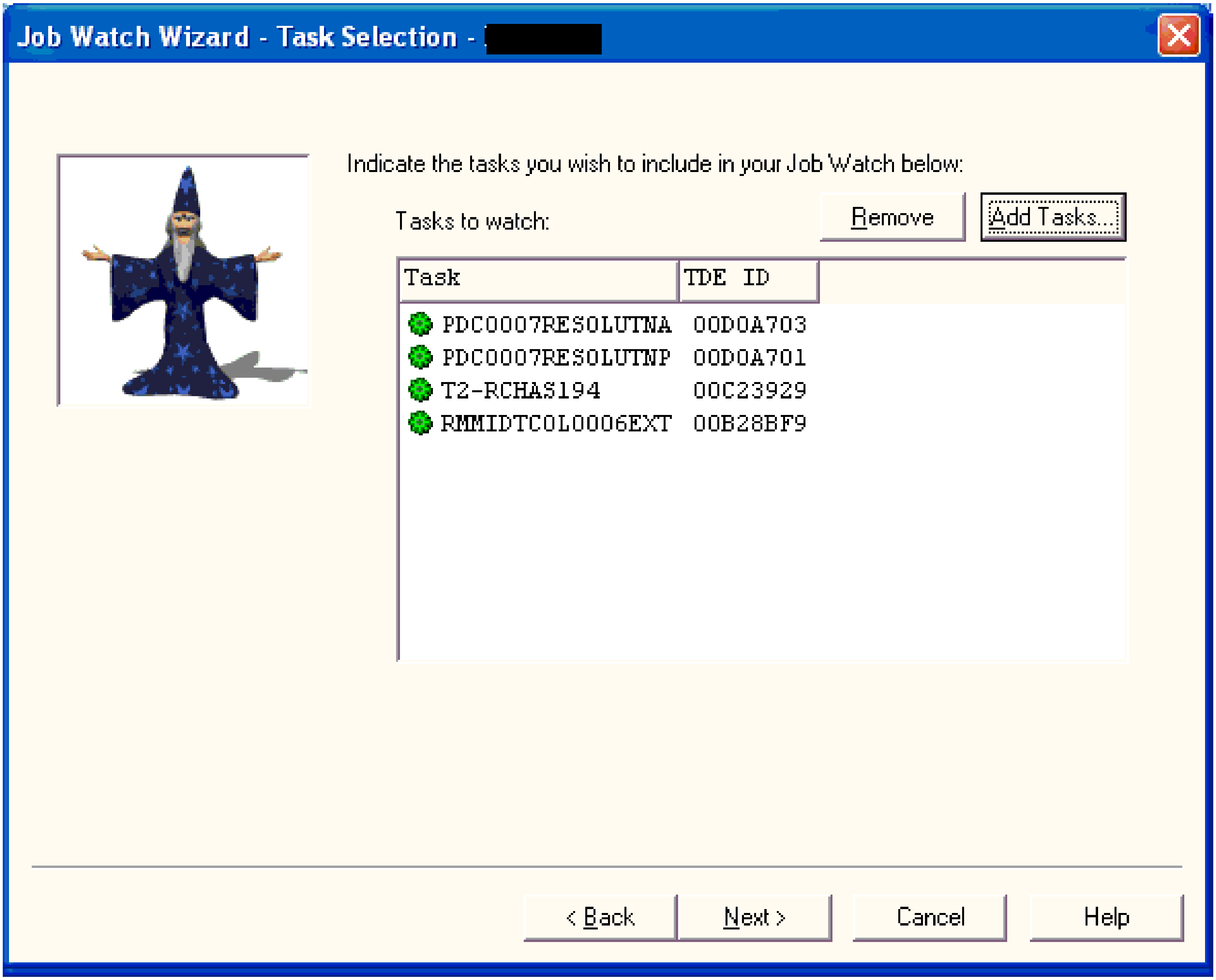
Table of Contents | Previous | Next

2.4.4.7 Task Selection

The task selection page lets the user add or remove tasks to the watch. There is a limit of 10 jobs/threads/tasks per Job Watch. Therefore if 4 jobs have already been selected on the job selection page, then only 6 tasks may be selected on the task selection page.

The following table provides information about each of the criteria on this page:

Field	**Description**
Remove button	Removes selected tasks from the list
Add Tasks button	Displays the Add Tasks dialog with a list of currently active tasks on the system to select from.
Tasks to watch list	List of tasks to collect.



Job Watch Wizard - Task Selection - [redacted]

Indicate the tasks you wish to include in your Job Watch below:

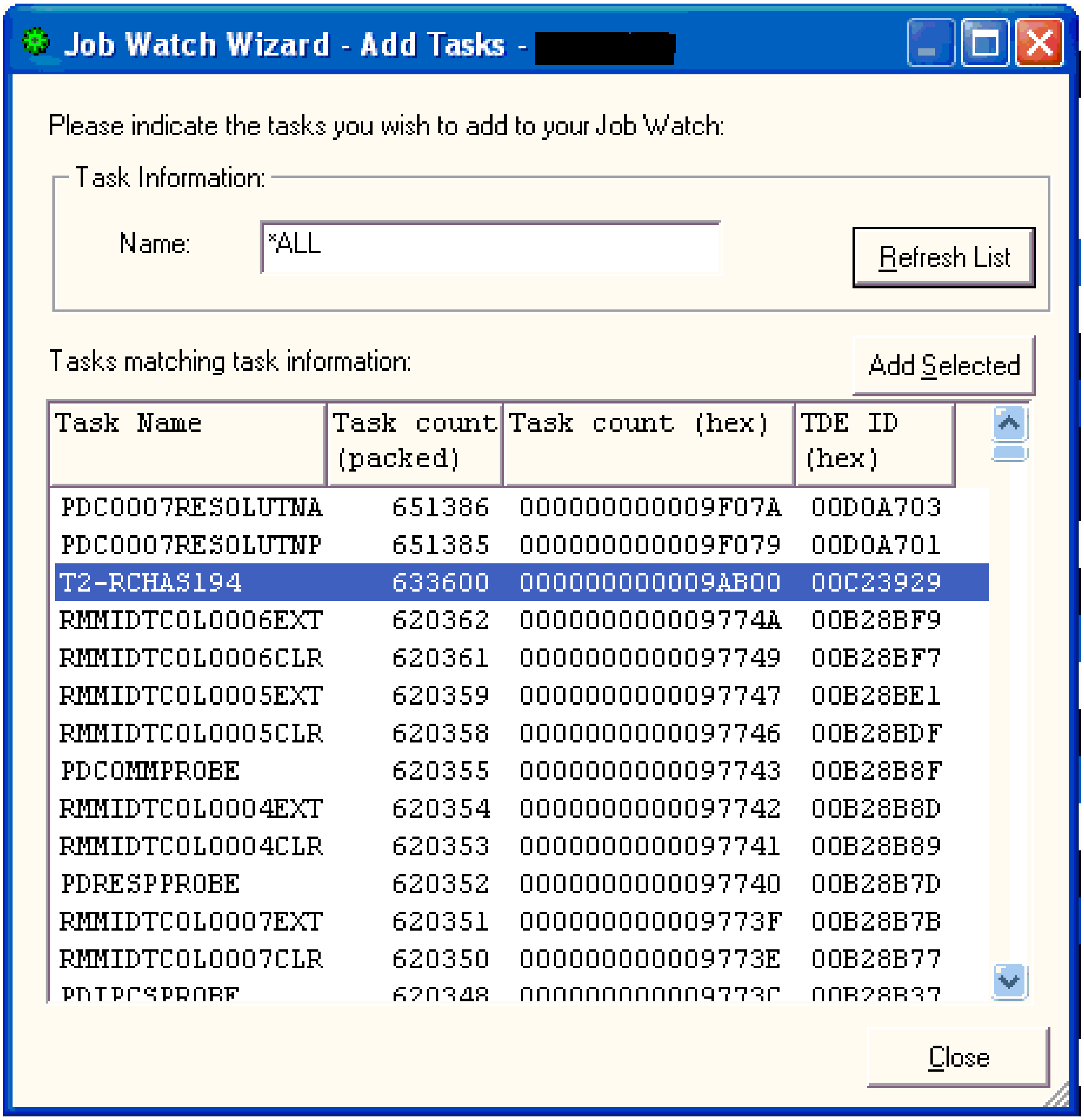
Tasks to watch:

Task	TDE ID
PDC0007RESOLUTNA	00D0A703
PDC0007RESOLUTNP	00D0A701
T2-RCHAS194	00C23929
RMMIDTCOL0006EXT	00B28BF9

[Job Watch Wizard - Task Selection Page]

Add Tasks Window

The add tasks window is displayed when the task selection page is first opened. This window allows the user to select which tasks should be included in the job watch. An example of the Add Tasks window is shown below:



Job Watch Wizard - Add Tasks - [redacted]

Please indicate the tasks you wish to add to your Job Watch:

Task Information:

Name: *ALL

Tasks matching task information:

Task Name	Task count (packed)	Task count (hex)	TDE ID (hex)
PDC0007RESOLUTNA	651386	000000000009F07A	00D0A703
PDC0007RESOLUTNP	651385	000000000009F079	00D0A701
T2-RCHAS194	633600	000000000009AB00	00C23929
RMMIDTCOL0006EXT	620362	000000000009774A	00B28BF9
RMMIDTCOL0006CLR	620361	0000000000097749	00B28BF7
RMMIDTCOL0005EXT	620359	0000000000097747	00B28BE1
RMMIDTCOL0005CLR	620358	0000000000097746	00B28BDF
PDCOMMPROBE	620355	0000000000097743	00B28B8F
RMMIDTCOL0004EXT	620354	0000000000097742	00B28B8D
RMMIDTCOL0004CLR	620353	0000000000097741	00B28B89
PDRESPPROBE	620352	0000000000097740	00B28B7D
RMMIDTCOL0007EXT	620351	000000000009773F	00B28B7B
RMMIDTCOL0007CLR	620350	000000000009773E	00B28B77
PDIPCSPROBE	620348	000000000009773C	00B28B37

[Job Watch Wizard - The Add Tasks window for the Task Selection Page]

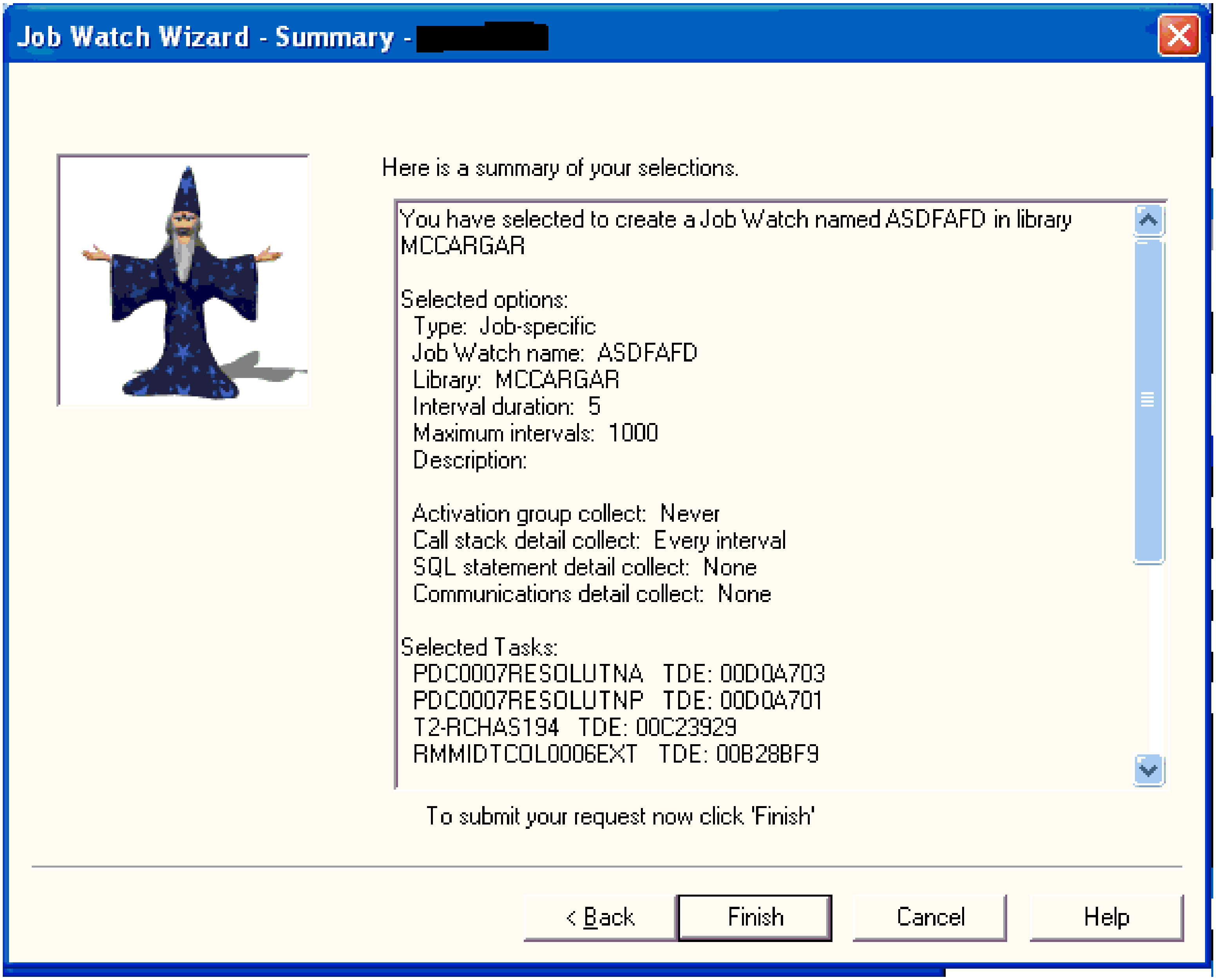
The following table provides information about each of the fields on this page:

Field	Description
Name	Indicates the generic task name that should be used when the list of jobs is next refreshed.
Refresh list	This option will build/rebuild the list of tasks based on the current tasks detected on the system.
Tasks matching job information filter	This is the list of tasks that meet the task information filter as produced by the last refresh.
Add selected	This button will add the selected tasks from the list to the task selection page of the Wizard.

2.4.4.8 Summary

The Summary page provides complete details about all selections made in the wizard. If anything listed doesn't look right, use the Back button to go back and make any changes necessary. After clicking 'Finish' a SBMJOB command will be issued to start the job watch. This command is listed at the bottom this page, and can be copied to a green screen session and modified if necessary.

After the job watch is submitted it will take several seconds before anything appears in the GUI while the files are being allocated. Use F5 to refresh the list of job watches in the current library in order to work with the newly created job watch.



Job Watch Wizard - Summary - [redacted]

Here is a summary of your selections.

You have selected to create a Job Watch named ASDFAFD in library MCCARGAR

Selected options:
  Type: Job-specific
  Job Watch name: ASDFAFD
  Library: MCCARGAR
  Interval duration: 5
  Maximum intervals: 1000
  Description:

  Activation group collect: Never
  Call stack detail collect: Every interval
  SQL statement detail collect: None
  Communications detail collect: None

Selected Tasks:
  PDC0007RESOLUTNA   TDE: 00D0A703
  PDC0007RESOLUTNP   TDE: 00D0A701
  T2-RCHAS194   TDE: 00C23929
  RMMIDTCOL0006EXT   TDE: 00B28BF9

To submit your request now click 'Finish'

[Job Watch Wizard - Summary Page]

Table of Contents Previous Next

2.4.5 Creating - Start Job Watch Wizard - System-wide mode

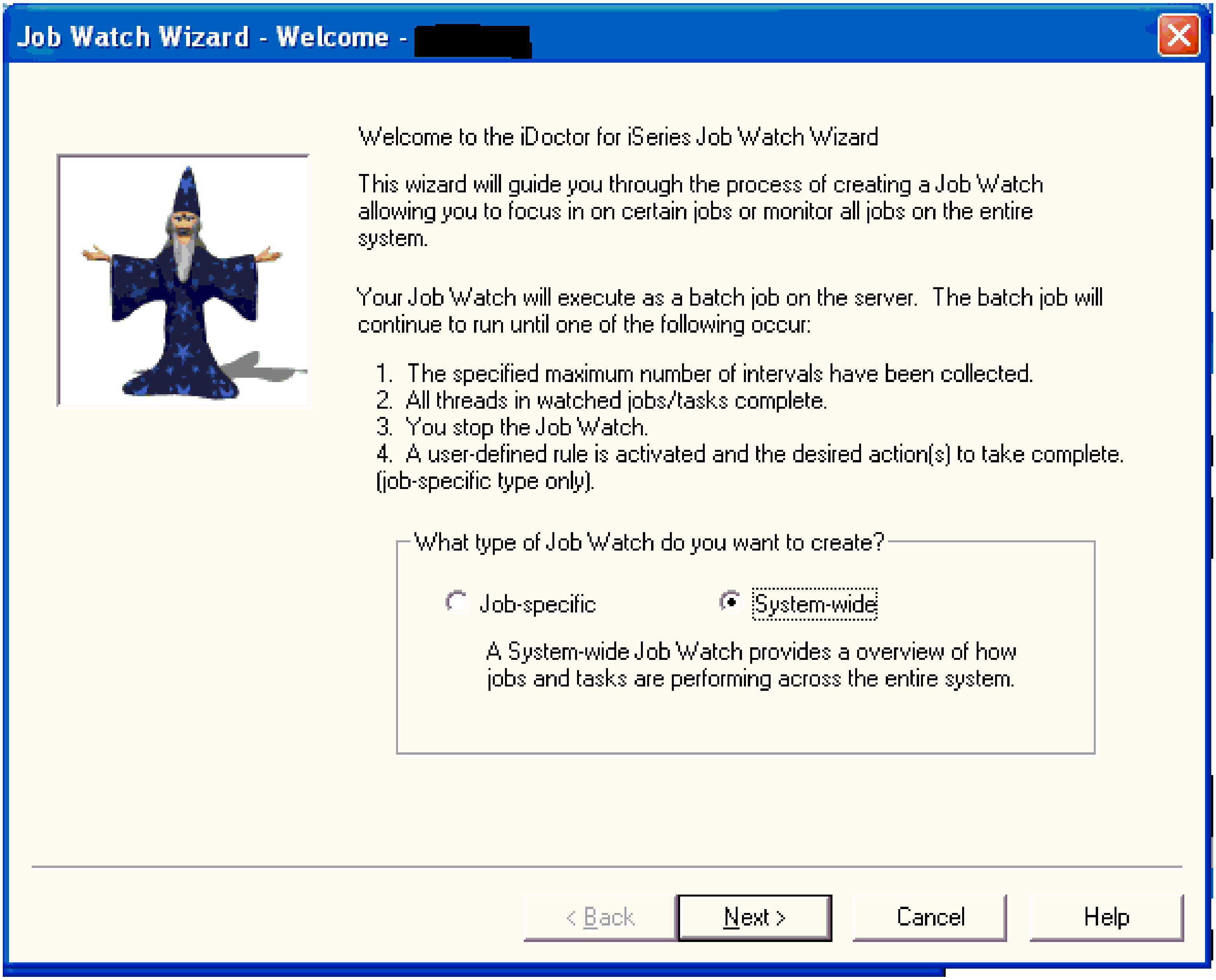
The system-wide mode of Job Watcher provides the capability to collect detailed information about all jobs and tasks on the system.

System-wide is a good choice to use if the problem jobs on the system are unknown or if more than 10 jobs need to be examined in one collection. Job-specific is the other mode available in Job Watcher.

This section covers the creation of a system-wide job watch using the Wizard. To create a new Job Watch use the Start Job Watch Wizard. The Wizard is accessible via the Start Job Watch menu on the Job Watcher or library folder icons.

The Job Watch Wizard guides the user step by step through the process of creating the Job Watch. Each page is covered in detailed within the next sections.

An example of the Wizard is shown below:



Job Watch Wizard - Welcome - [redacted]

Welcome to the iDoctor for iSeries Job Watch Wizard

This wizard will guide you through the process of creating a Job Watch allowing you to focus in on certain jobs or monitor all jobs on the entire system.

Your Job Watch will execute as a batch job on the server. The batch job will continue to run until one of the following occur:

1. The specified maximum number of intervals have been collected.
2. All threads in watched jobs/tasks complete.
3. You stop the Job Watch.
4. A user-defined rule is activated and the desired action(s) to take complete. (job-specific type only).

What type of Job Watch do you want to create?

○ Job-specific ● System-wide

A System-wide Job Watch provides a overview of how jobs and tasks are performing across the entire system.

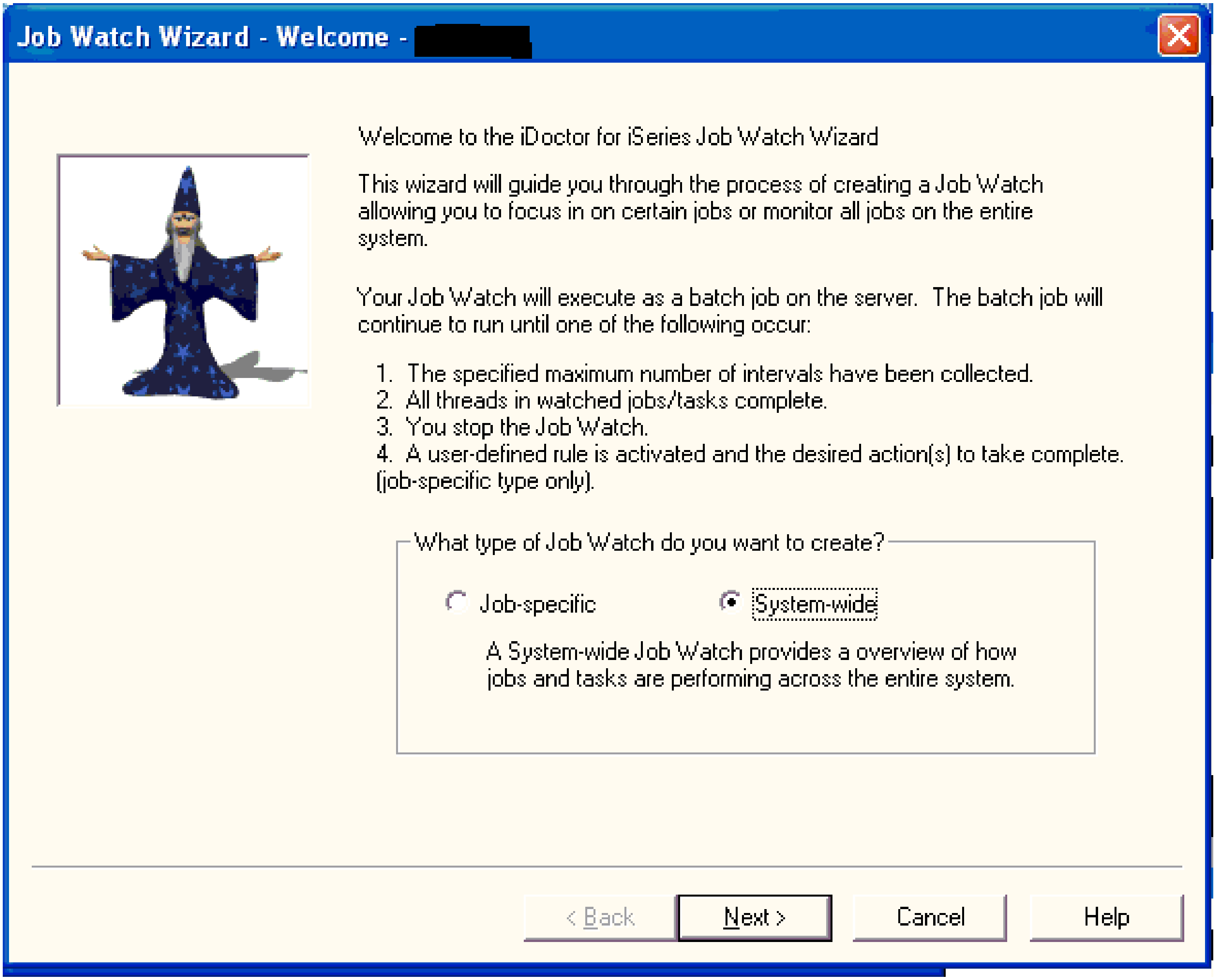
[The Start Job Watch Wizard]

Table of Contents | Previous | Next

2.4.5.1 Welcome

The Welcome page in the Job Watch Wizard introduces the user to the wizard and offers information about what the wizard will do. The page also explains for what conditions the Job Watch will end execution.

On this window the user must specify the type of Job Watch to create. System-wide is a good choice for first time users or if the job names that should be analyzed are unknown. Job-specific is better when the jobs that are running poorly are known to the user poorly because it can provide additional information that system-wide cannot.



Job Watch Wizard - Welcome - [redacted]

Welcome to the iDoctor for iSeries Job Watch Wizard

This wizard will guide you through the process of creating a Job Watch allowing you to focus in on certain jobs or monitor all jobs on the entire system.

Your Job Watch will execute as a batch job on the server. The batch job will continue to run until one of the following occur:

1. The specified maximum number of intervals have been collected.
2. All threads in watched jobs/tasks complete.
3. You stop the Job Watch.
4. A user-defined rule is activated and the desired action(s) to take complete. (job-specific type only).

What type of Job Watch do you want to create?

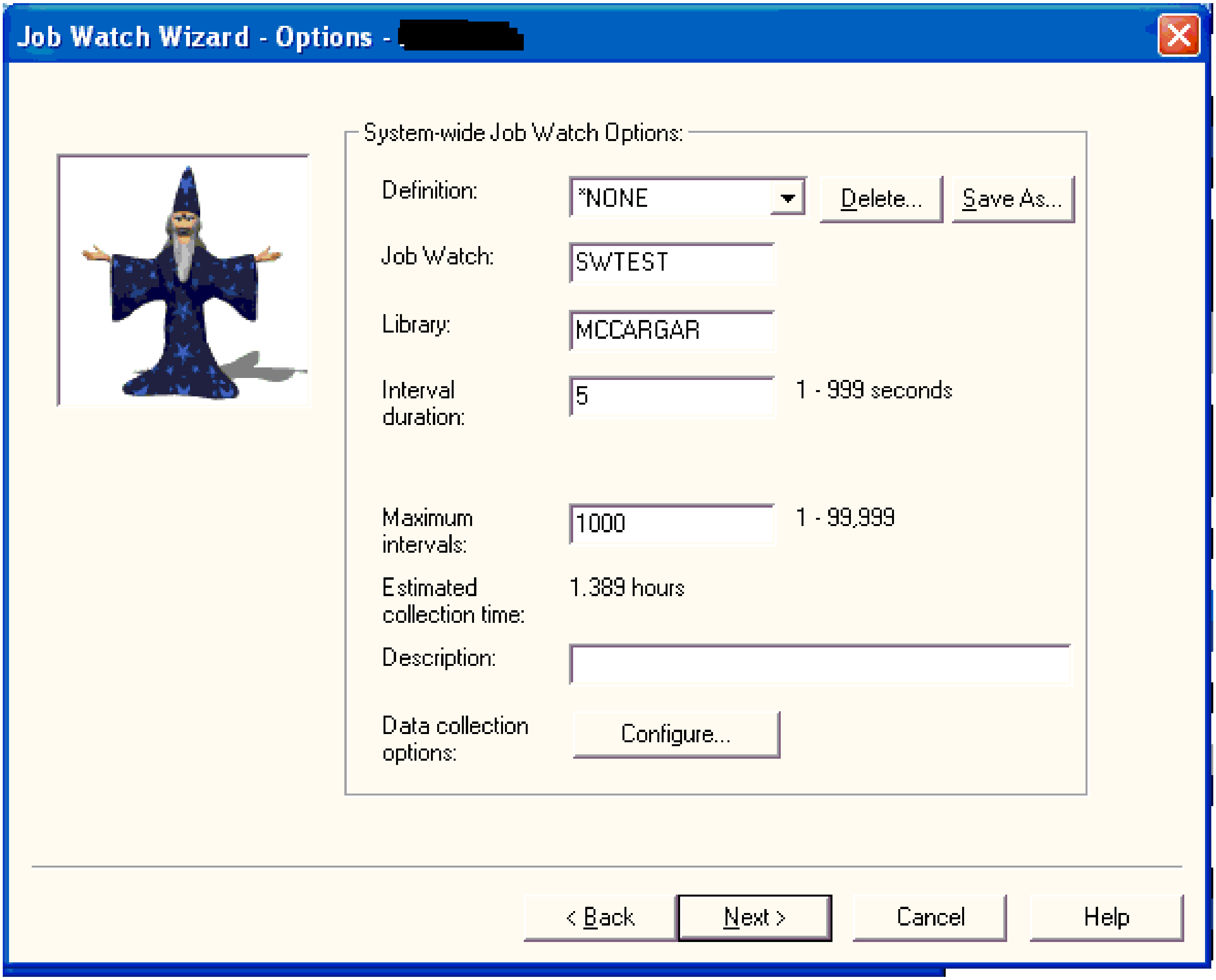
○ Job-specific ● System-wide

A System-wide Job Watch provides a overview of how jobs and tasks are performing across the entire system.

[Job Watch Wizard - Welcome Page]

2.4.5.2 Options

The Options Page allows the user to specify the most basic pieces of information about a Job Watch such as its name, library, duration and description. The following is an example of this page of the Start Job Watch Wizard.



Job Watch Wizard - Options - [redacted]

System-wide Job Watch Options:

Definition: *NONE

Job Watch: SWTEST

Library: MCCARGAR

Interval duration: 5 1 - 999 seconds

Maximum intervals: 1000 1 - 99,999

Estimated collection time: 1.389 hours

Description:

Data collection options:

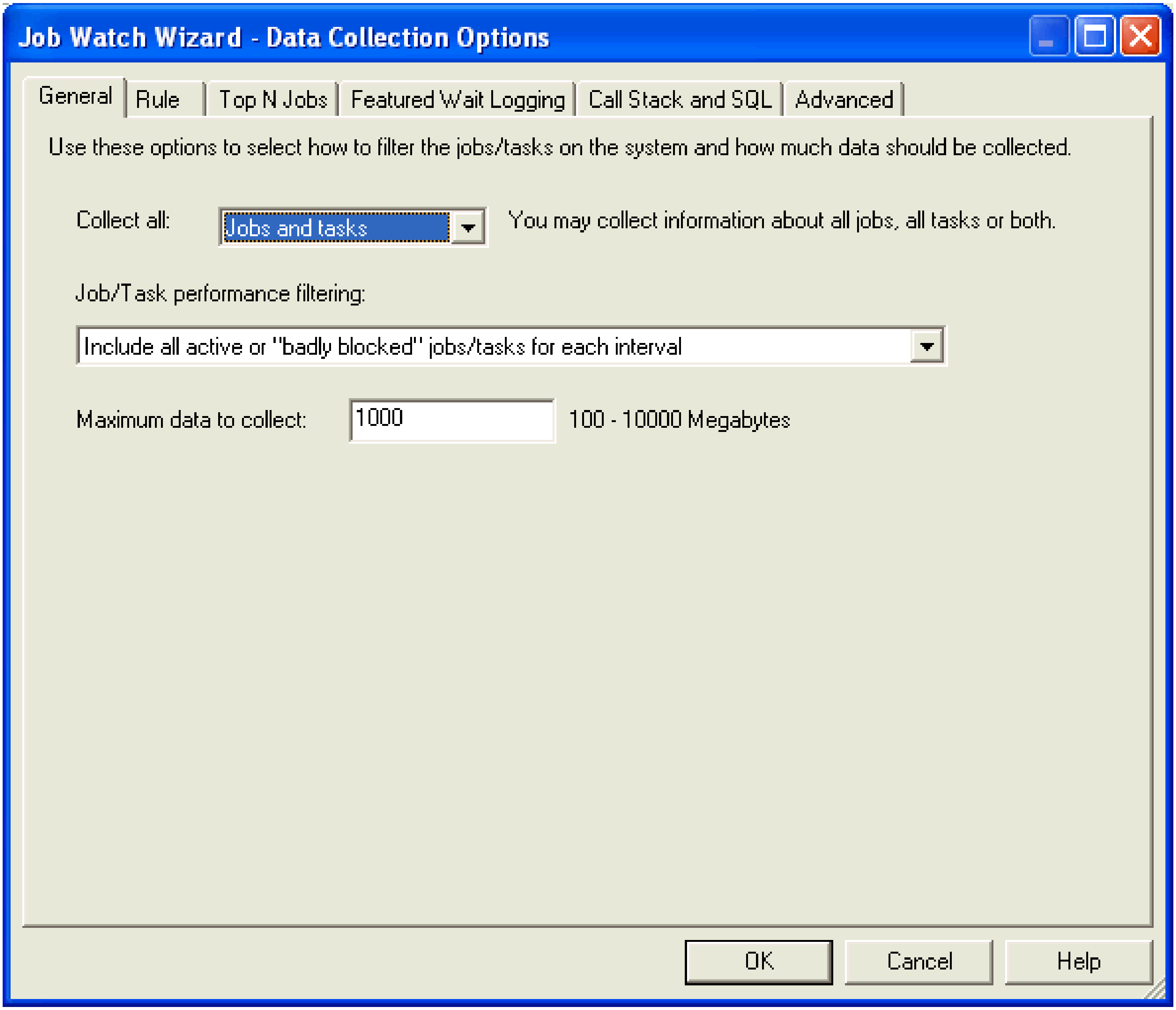
[Job Watch Wizard - Options Page]

The following table provides details about each of the parameters on this page:

Field	Description
Definition	The name of the definition to create the job watch with. A Job Watcher definition is a set of parameter settings that can be used to more quickly set up and run multiple Job Watches having the same data collection options. A definition can be created by using the Save As... button on the options page once the desired data collection options have been set on the data collection options section of the Wizard. By selecting a definition from the list of definitions, the parameters for the definition will be loaded into the Wizard for use.
Job Watch	The name of the Job Watch (10 chars max). This name matches the member name used when creating the output files on the server. The output file names start with QPYRT* and will exist in the library specified on this page of the Wizard.
Library	The name of the library to create the Job Watch in. If the library does not exist, the client will ask for confirmation before creating it.
Interval duration	The size of each sample of data in seconds.
Maximum intervals	The total number of intervals to collect data for.
Estimated collection time	The total estimated time the collection will run if it is not stopped prematurely based on the interval duration and maximum intervals parameter values provided.
Description	A description to give the Job Watch being created.
Data collection options button	Click this button to select additional types of information to collect. The options available vary greatly depending on the type of Job Watch being created.

2.4.5.3 Data collection options - General page

This page allows the user to specify the scope of collection for the system-wide job watch and the maximum data to collect.



Use these options to select how to filter the jobs/tasks on the system and how much data should be collected.

Collect all: Jobs and tasks

You may collect information about all jobs, all tasks or both.

Job/Task performance filtering:

Include all active or "badly blocked" jobs/tasks for each interval

Maximum data to collect: 1000 100 - 10000 Megabytes

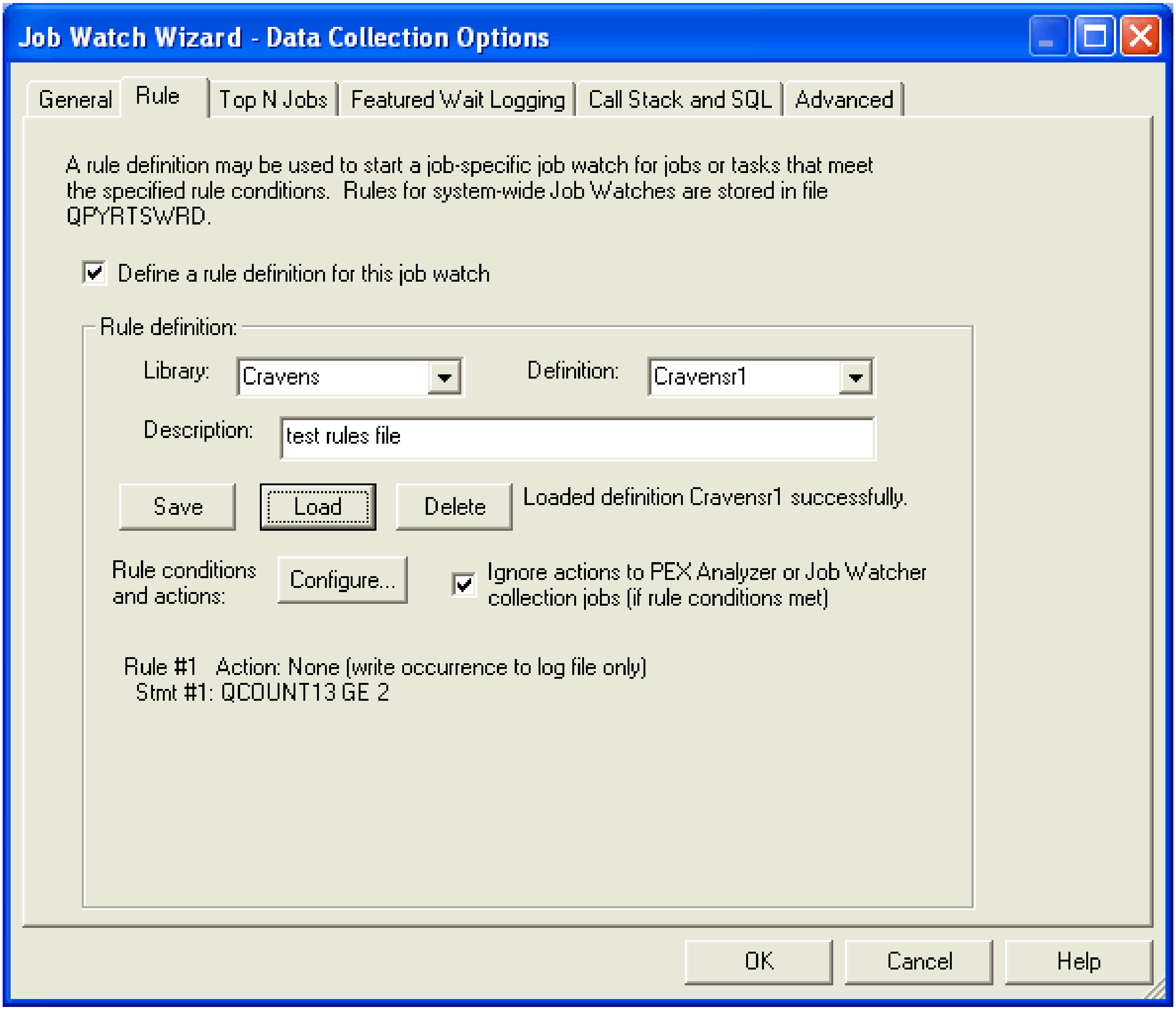
[Job Watch Wizard - Data Collection Options Page]

The following table describes the parameters available on this page of the Wizard. The default values for each parameter are **bold**.

Collection Option	**Description**
Collect all	Possible values are: **Jobs and tasks** Jobs only Tasks only
Job/Task performance filtering	Possible values are: Include only exception or top N jobs/tasks each interval - Produces file QPYRTSWLG1 **Include all active or "badly blocked" jobs/tasks for each interval -** Produces files QPYRTSWLG1, QPYRTSWLG2 Include all jobs/tasks each interval (Not Recommended) - Produces file QPYRTSWLG1, QPYRTSWLG2, QPYRTSWLG3
Maximum data to collect	The maximum number of megabytes of Job Watcher data to collect before the collection should end. The collection will end when either the total amount of disk space consumed by the job watch exceeds the value specified or when the maximum intervals parameter specified on the options page has been reached. Default value is **1000 MB.**

2.4.5.4 Data collection options - Rule page

This page allows the user to define or reuse a rule definition to use during execution of the system-wide job watch. A rule definition can be used to start a job-specific job watch over the job(s) matching the rule conditions.



A rule definition may be used to start a job-specific job watch for jobs or tasks that meet the specified rule conditions. Rules for system-wide Job Watches are stored in file QPYRTSWRD.

☑ Define a rule definition for this job watch

Rule definition:

Library: Cravens

Definition: Cravensr1

Description: test rules file

Loaded definition Cravensr1 successfully.

Rule conditions and actions:

☑ Ignore actions to PEX Analyzer or Job Watcher collection jobs (if rule conditions met)

Rule #1 Action: None (write occurrence to log file only)
Stmt #1: QCOUNT13 GE 2

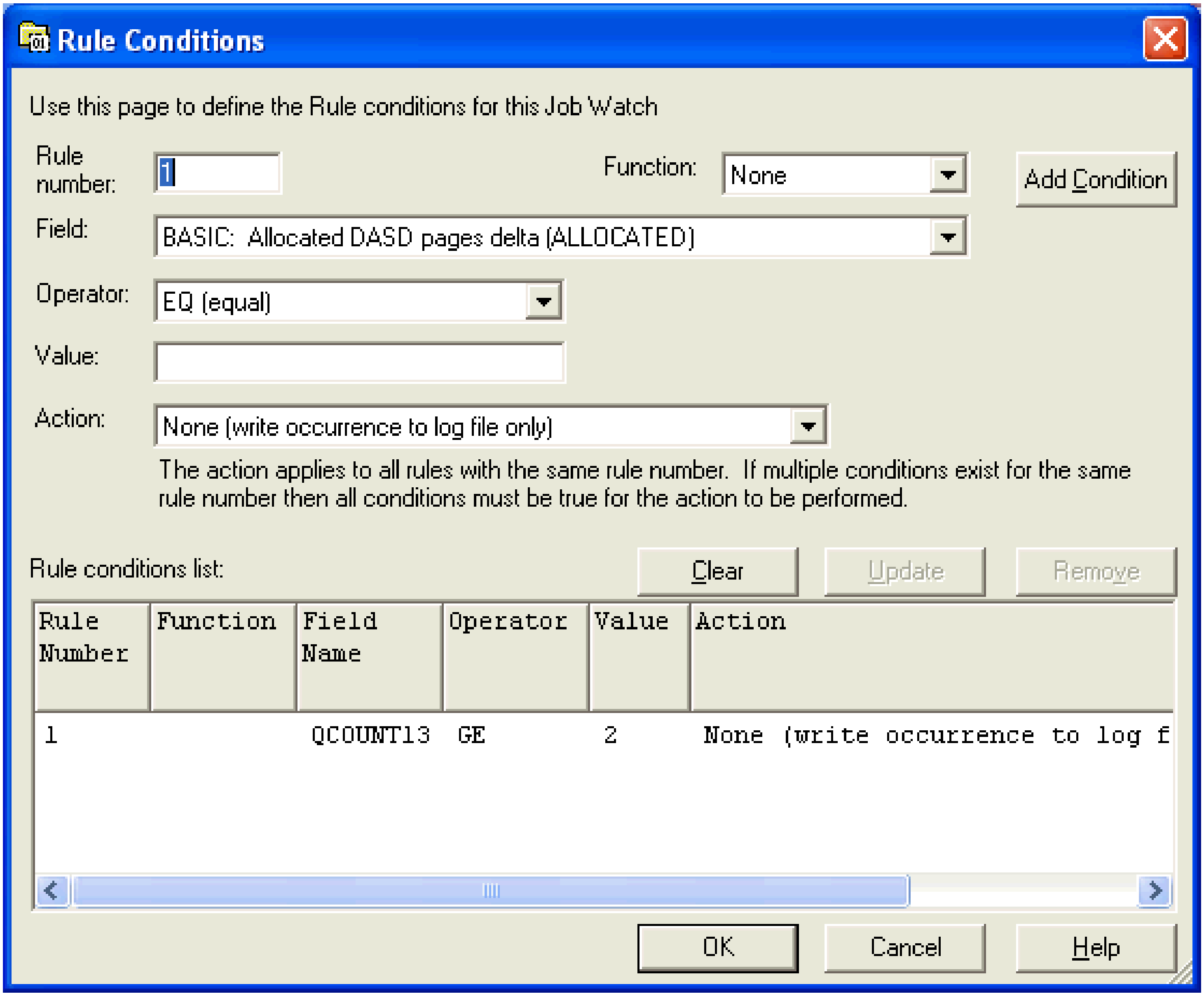
[Job Watch Wizard - Data Collection Options Page]

The following table describes the parameters available on this page of the Wizard.

Option	Description
Define a rule checkbox	This option can be used to quickly disable/reenable use of a rule definition for the current job watch. If a rule has been defined in the current session, unselecting the checkbox will cause the rule definition to not to be used when the job watch is submitted.
Library	The name of the library the rule definition is located in. The library list box contains all libraries on the system where file QPYRTSWRD (the rule definition file for system-wide job watches) exists.
Definition	The name of the rule definition. This name matches the member name in file QPYRTSWRD in the library specified. After a library is selected, clicking the arrow on the definition drop down list will show all members that exist in file QPYRTJWRD in the library specified. Select a library/member from the lists if a rule definition needs to be loaded into the wizard or deleted using the Load, Delete buttons on the page.
Description	The description of the rule definition. Type in a description if desired to associate with the rule definition being created.
Save button	Saves the current settings on this page of the wizard to the library/definition name indicated.
Load button	Loads the rule definition in the library/definition name indicated. The current settings on the rule page will be replaced with the information obtained from the rule definition specified.
Delete button	Deletes the rule definition in the library/definition name indicated.
Ignore actions to iDoctor jobs	This option (if selected) indicates that any PEX Analyzer collection job (named IDOCCOL) or Job Watcher collection jobs (named QPYJWJOB, QPYSWJOB) will be ignored and not evaluated as part as the rule condition checking.

Rule conditions and actions

Clicking on the rule conditions and actions button will display the following interface for configuring the rule definitions conditions/actions:



Use this page to define the Rule conditions for this Job Watch

Rule number: 1

Function: None

Field: BASIC: Allocated DASD pages delta (ALLOCATED)

Operator: EQ (equal)

Value:

Action: None (write occurrence to log file only)

The action applies to all rules with the same rule number. If multiple conditions exist for the same rule number then all conditions must be true for the action to be performed.

Rule conditions list:

Rule Number	Function	Field Name	Operator	Value	Action
1		QCOUNT13	GE	2	None (write occurrence to log f

The following table describes the options available within this interface:

Option	Description
Rule number	Identifies the rule number to use when adding the next condition in the list. All conditions having the same rule number will also share the same action.
Function	This drop down list can be used to apply a function to the selected field when the Add Condition button is pressed. If the selected field changes the functions available in the list will be updated. Some functions are only available when certain fields are selected.

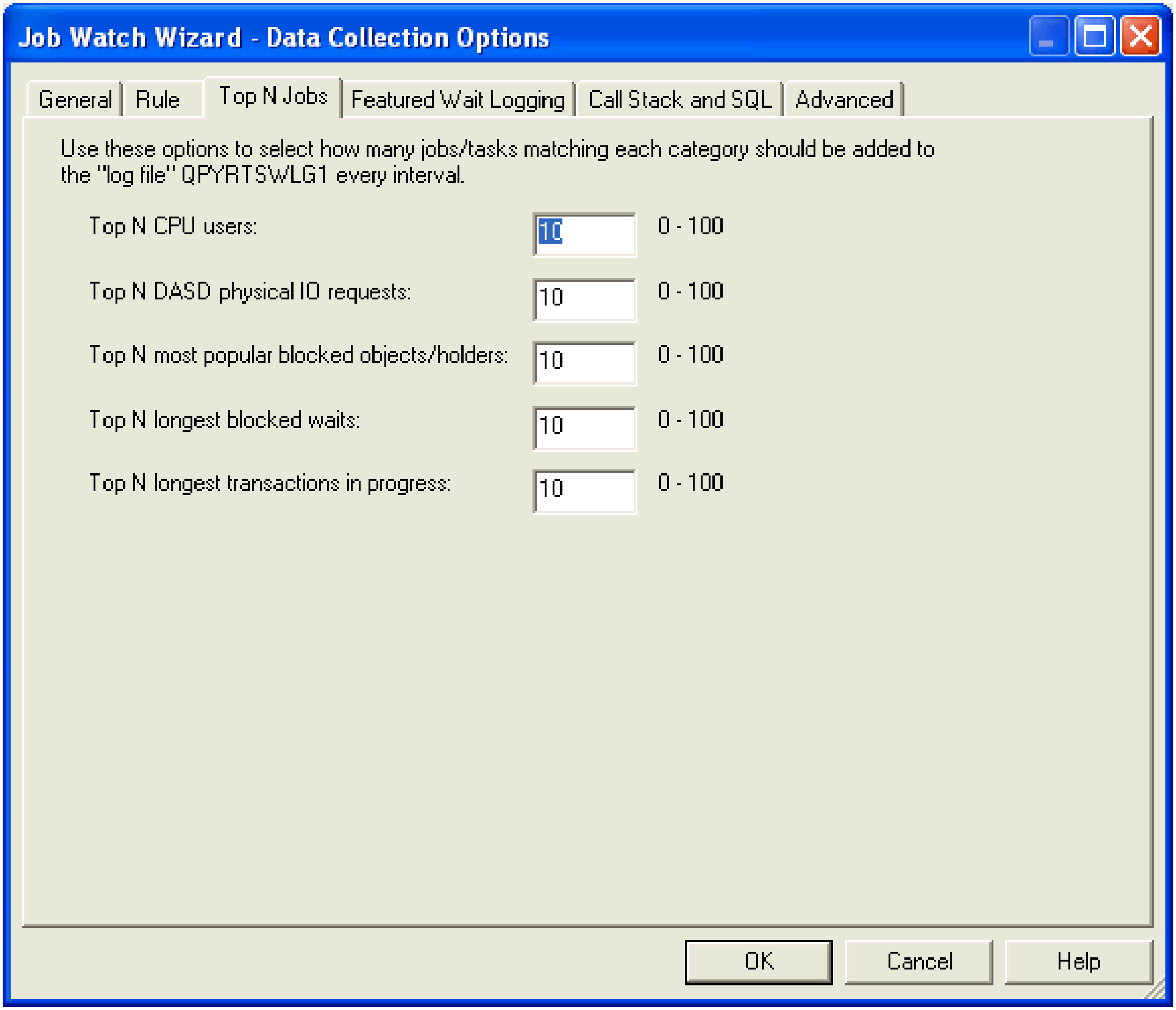
Field	The list of possible fields the rule definition may be defined over. Most of these fields exist in file QPYRTSWLG1. The fields are organized by category. The categories are: Basic - Disk/IO fields, performance counters, current user profile Buckets - The 32 wait bucket counts and times Comm - communications statistics Database - SQL statement or LDIO fields Exceptions - numeric data type exceptions Holder - fields related to the holding job IFS - IFS statistics State transitions - active to wait, active to ineligible, wait to ineligible Transactions - transaction counters and times Wait object - fields related to the wait object
Operator	The types of operations that can be performed on the selected field. The list of operators is automatically updated everytime the selected field changes.
Value	The numeric or character value that will be compared against the field using the operator specified.
Action	The action to take if all conditions having the same rule number are met. The action defined by the 1st rule number will be used if the actions differ for statements within the same rule number. Possible actions are: None (write occurrence to log file only) Start a job watch Start a job wtch (over the holding job)
Add Condition	Adds the selected field, operator and value to the list of conditions.
Clear button	Removes all entries from the rule conditions list.
Update button	Updates the selected entry in the rule conditions list with the selected field, operator and value on this window.
Remove button	Removes all selected entries from the rule conditions list.
Rule conditions list	The list of conditions which will be compared against the Job Watcher data to see if the conditions are met. If the conditions are met the specified actions for each rule number will be implemented.

Table of Contents | Previous | Next

2.4.5.5 Data collection options - Top N jobs

This page allows the user to specify the maximum number of jobs/threads/tasks that should be written out to the log file (QPYRTSWLG1) every interval for the following categories of jobs: Top CPU users, Top DASD physical IO requests, most popular blocked objects/holders, longest blocked waits, and longest transactions in progress. Each catetgory allows a value from 0 to 100.

These options only effect the output in file QPYRTSWLG1 and not file QPYRTSWLG2.



Use these options to select how many jobs/tasks matching each category should be added to the "log file" QPYRTSWLG1 every interval.

Top N CPU users: 10 0 - 100

Top N DASD physical IO requests: 10 0 - 100

Top N most popular blocked objects/holders: 10 0 - 100

Top N longest blocked waits: 10 0 - 100

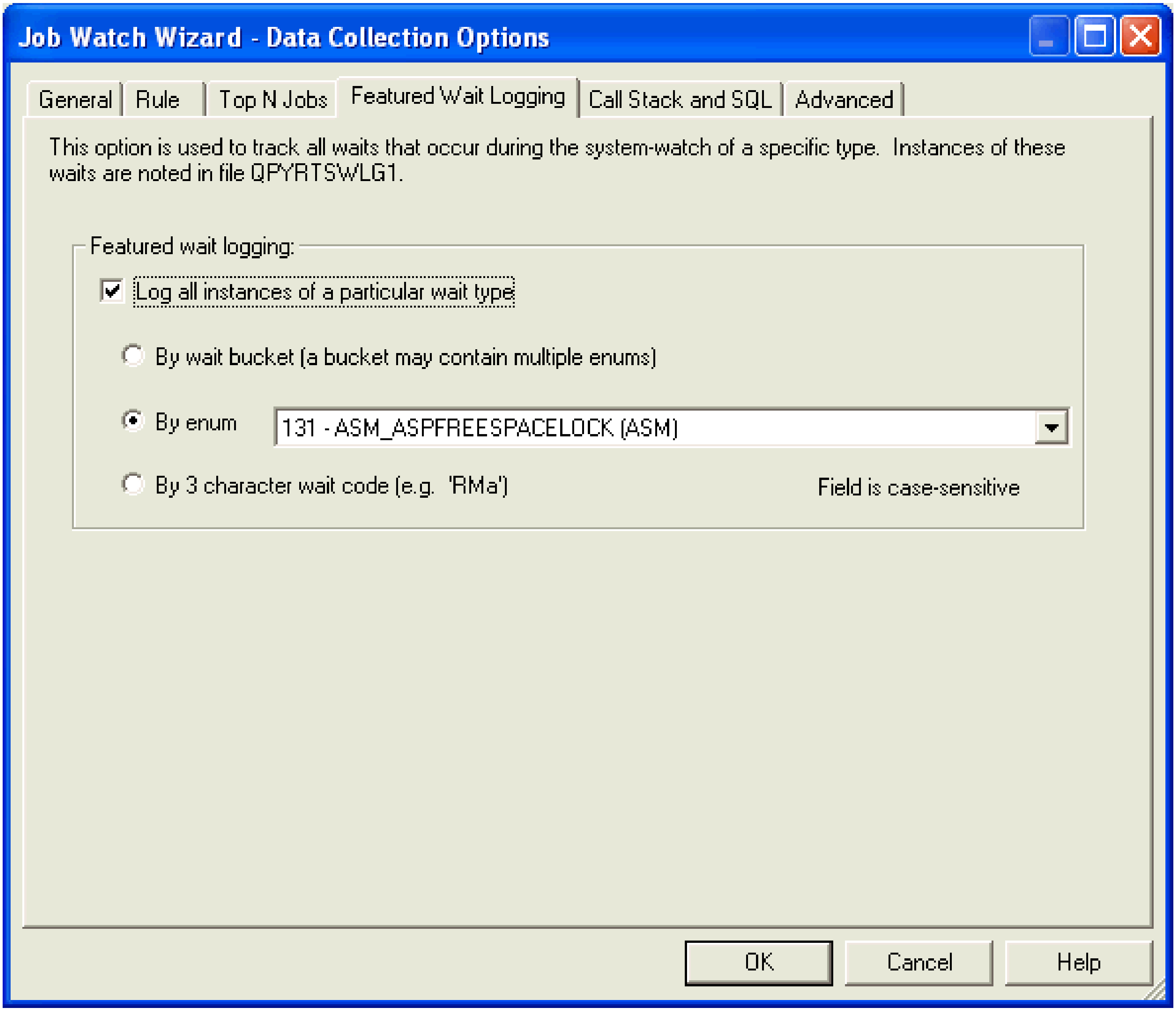
Top N longest transactions in progress: 10 0 - 100

Table of Contents | Previous | Next

2.4.5.6 Data collection options - Featured waiting logging

This page allows the user to log every occurrence where a specific type of wait happens at the moment the snapshot is taken into the log file (QPYRTSWLG1).

An example of this window is shown below:



Job Watch Wizard - Data Collection Options

This option is used to track all waits that occur during the system-watch of a specific type. Instances of these waits are noted in file QPYRTSWLG1.

Featured wait logging:

☑ Log all instances of a particular wait type

○ By wait bucket (a bucket may contain multiple enums)

● By enum 131 - ASM_ASPFREESPACELOCK (ASM)

○ By 3 character wait code (e.g. 'RMa') Field is case-sensitive

The following table describes the parameters available on this page of the Wizard.

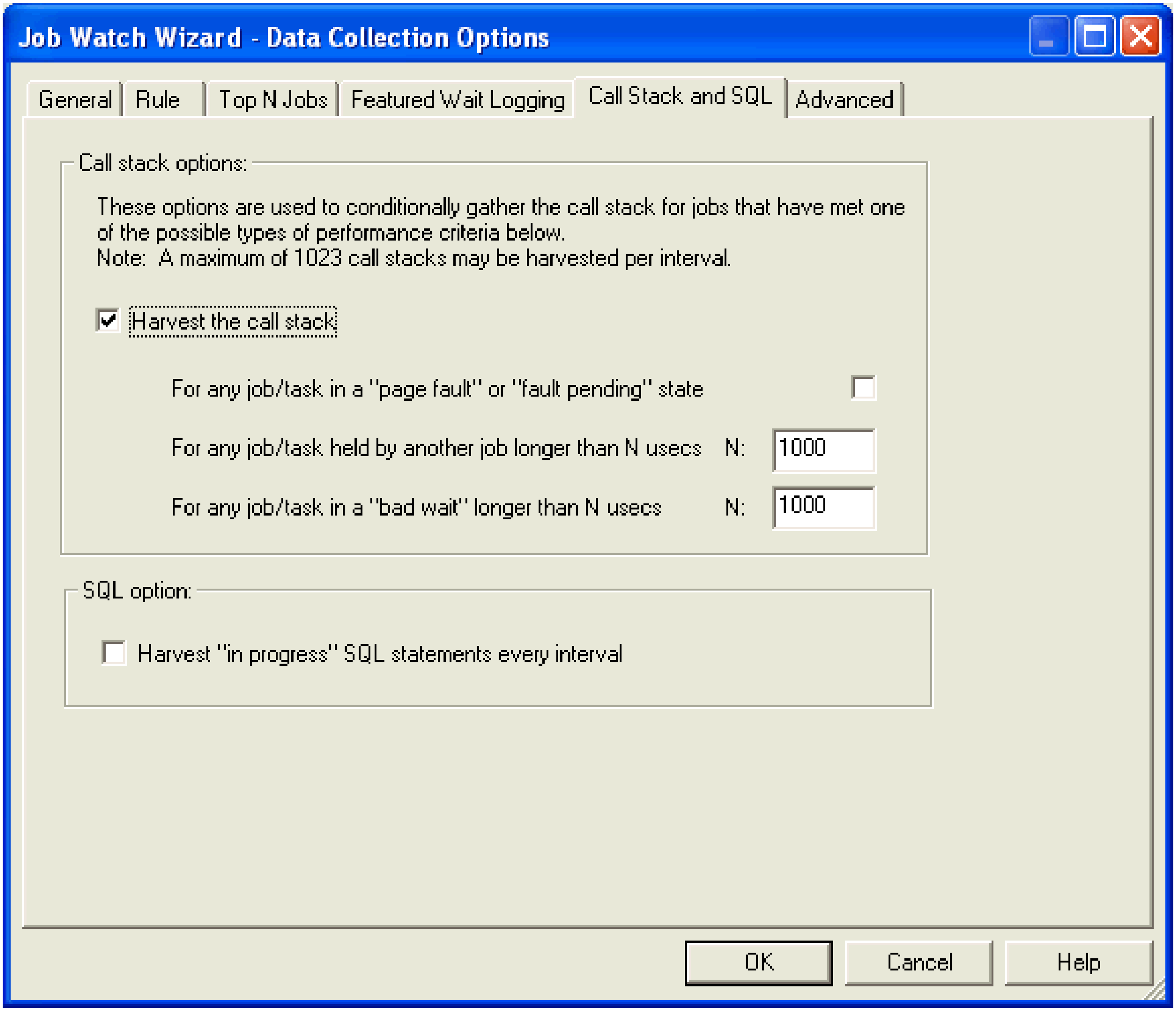
Option	Description
Log all instances of a particular wait type	This option can be used to quickly disable/reenable use of the featured wait logging support.
By wait bucket	This option allows the user to pick a specific wait bucket which should be logged.
By enum	This option allows the user to pick the specific wait point (within a wait bucket) which should be logged.
By 3 character wait code	Allows the user to pick a specific wait code (LIC wait type) which should be logged.

Table of Contents | Previous | Next

2.4.5.7 Data collection options - Call stack and SQL

This page allows the user to optionally collect call stacks and actively running SQL statements.

An example of this window is shown below:



Call stack options:

These options are used to conditionally gather the call stack for jobs that have met one of the possible types of performance criteria below.
Note: A maximum of 1023 call stacks may be harvested per interval.

☑ Harvest the call stack

For any job/task in a "page fault" or "fault pending" state □

For any job/task held by another job longer than N usecs N: 1000

For any job/task in a "bad wait" longer than N usecs N: 1000

SQL option:

□ Harvest "in progress" SQL statements every interval

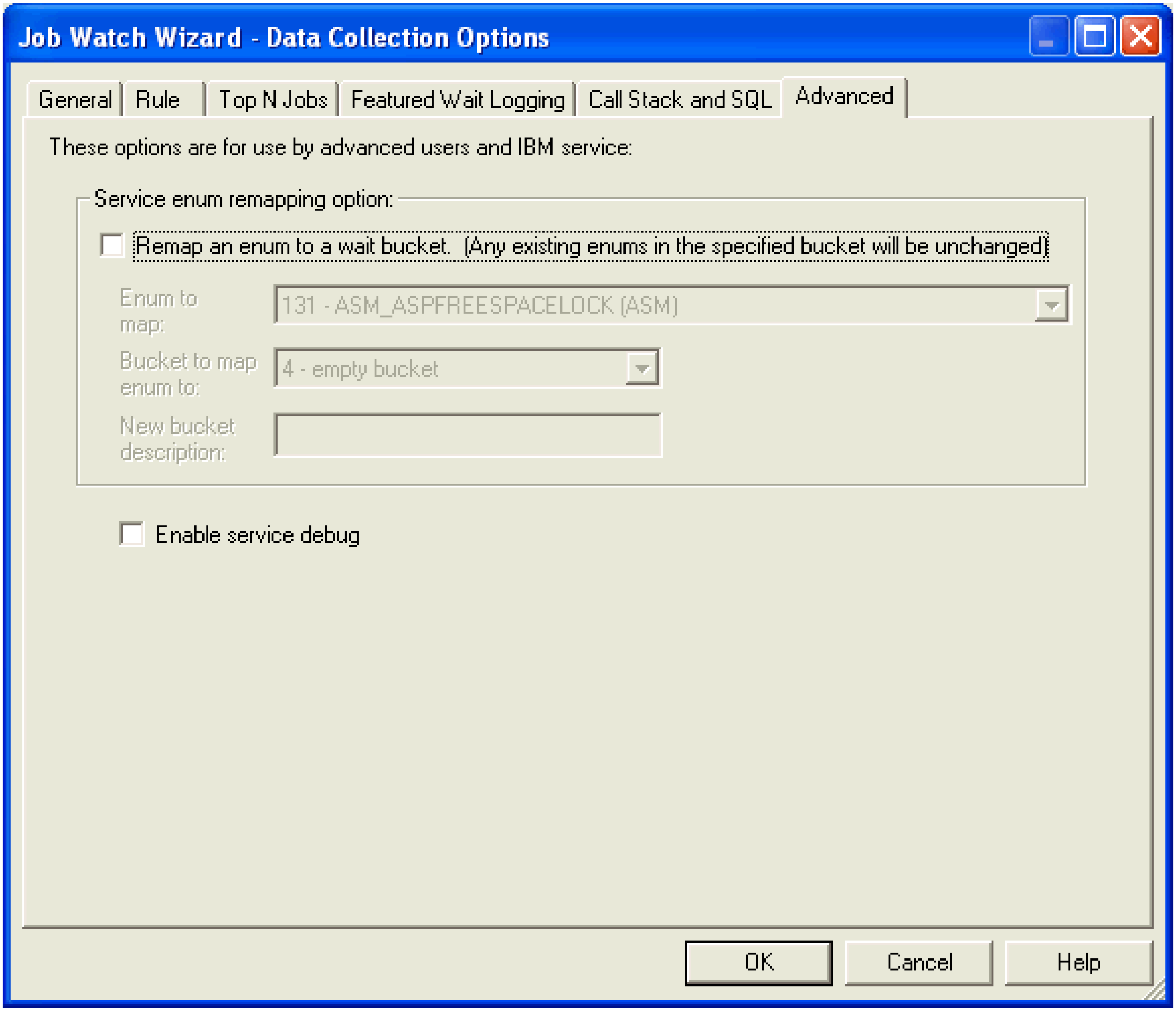
The following table describes the parameters available on this page of the Wizard.

Option	Description
Harvest the call stack	This checkbox can be used to quickly disable/reenable the call stack collection option.
For any/job task in a "page fault" or "fault pending" state	Indicates that the job's call stack should be collected if the job is in a "page fault" or "fault pending" state at the moment the job's condition is evaluated each interval.
For any job/task held by another job longer than N usecs	Indicates that the job's call stack should be collected if the job has been held longer than a user specified number of microseconds at the moment the job's condition is evaluated each interval.
For any job/task in a "bad wait" longer than N usecs	Indicates that the job's call stack should be collected if the job has been in a "bad wait" (like a seize) longer than a user specified number of microseconds at the moment the job's condition is evaluated each interval.
Harvest "in progress" SQL	Collects the SQL statement (without host variables) for any job actively running an SQL statement at the moment the job's condition is evaluated each interval.

Table of Contents | Previous | Next

2.4.5.8 Data collection options - Advanced

This page is intended for IBM service use only.



These options are for use by advanced users and IBM service:

Service enum remapping option:

□ Remap an enum to a wait bucket. (Any existing enums in the specified bucket will be unchanged)

Enum to map: 131 - ASM_ASPFREESPACELOCK (ASM)

Bucket to map enum to: 4 - empty bucket

New bucket description:

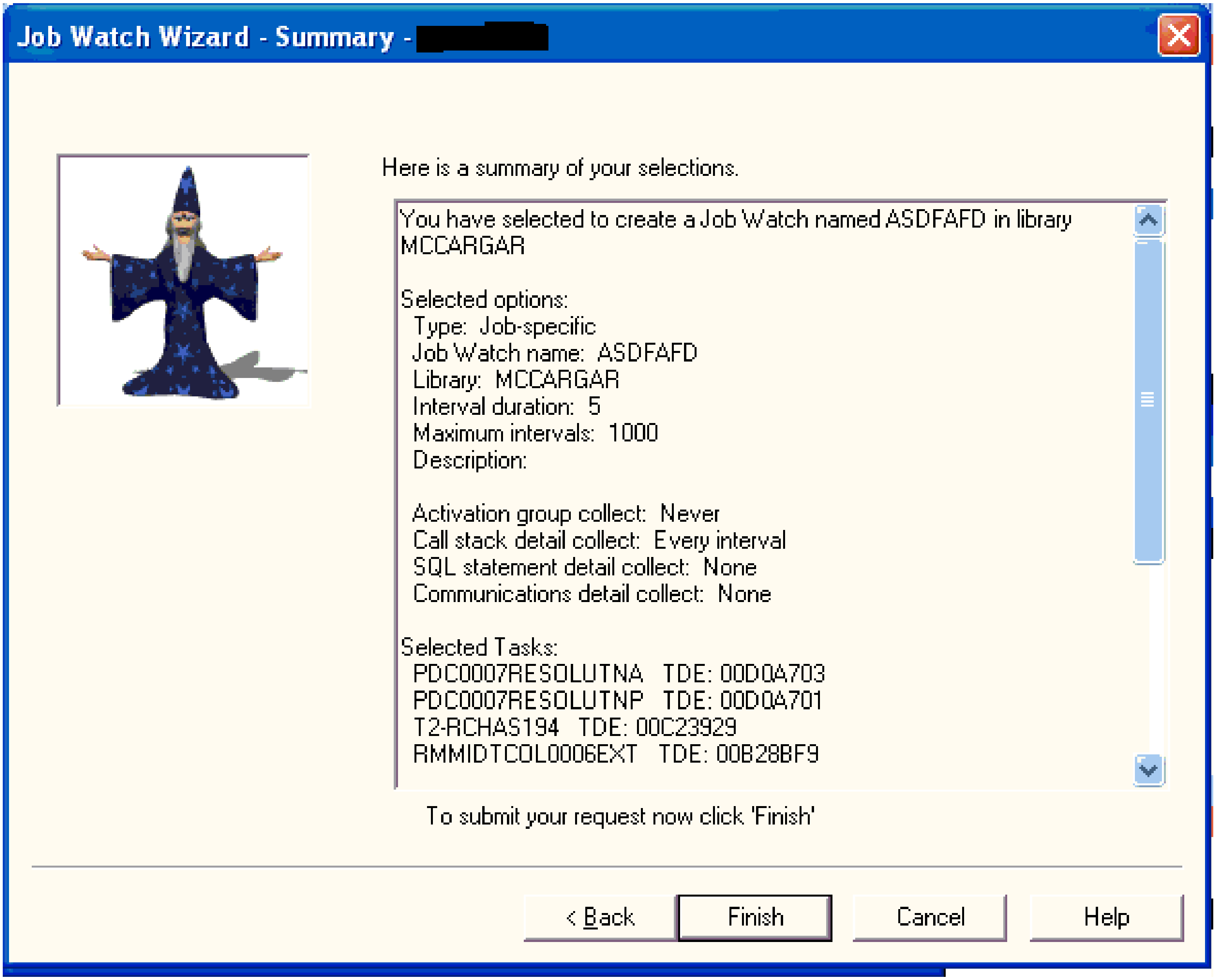
□ Enable service debug

Table of Contents | Previous | Next

2.4.5.9 Summary

The Summary page provides complete details about all selections made in the wizard. If anything listed doesn't look right, use the Back button to go back and make any changes necessary. After clicking 'Finish' a SBMJOB command will be issued to start the job watch. This command is listed at the bottom this page, and can be copied to a green screen session and modified if necessary.

After the job watch is submitted it will take several seconds before anything appears in the GUI while the files are being allocated. Use F5 to refresh the list of job watches in the current library in order to work with the newly created job watch.



Job Watch Wizard - Summary - [redacted]

Here is a summary of your selections.

You have selected to create a Job Watch named ASDFAFD in library MCCARGAR

Selected options:
  Type: Job-specific
  Job Watch name: ASDFAFD
  Library: MCCARGAR
  Interval duration: 5
  Maximum intervals: 1000
  Description:

  Activation group collect: Never
  Call stack detail collect: Every interval
  SQL statement detail collect: None
  Communications detail collect: None

Selected Tasks:
  PDC0007RESOLUTNA   TDE: 00D0A703
  PDC0007RESOLUTNP   TDE: 00D0A701
  T2-RCHAS194   TDE: 00C23929
  RMMIDTCOL0006EXT   TDE: 00B28BF9

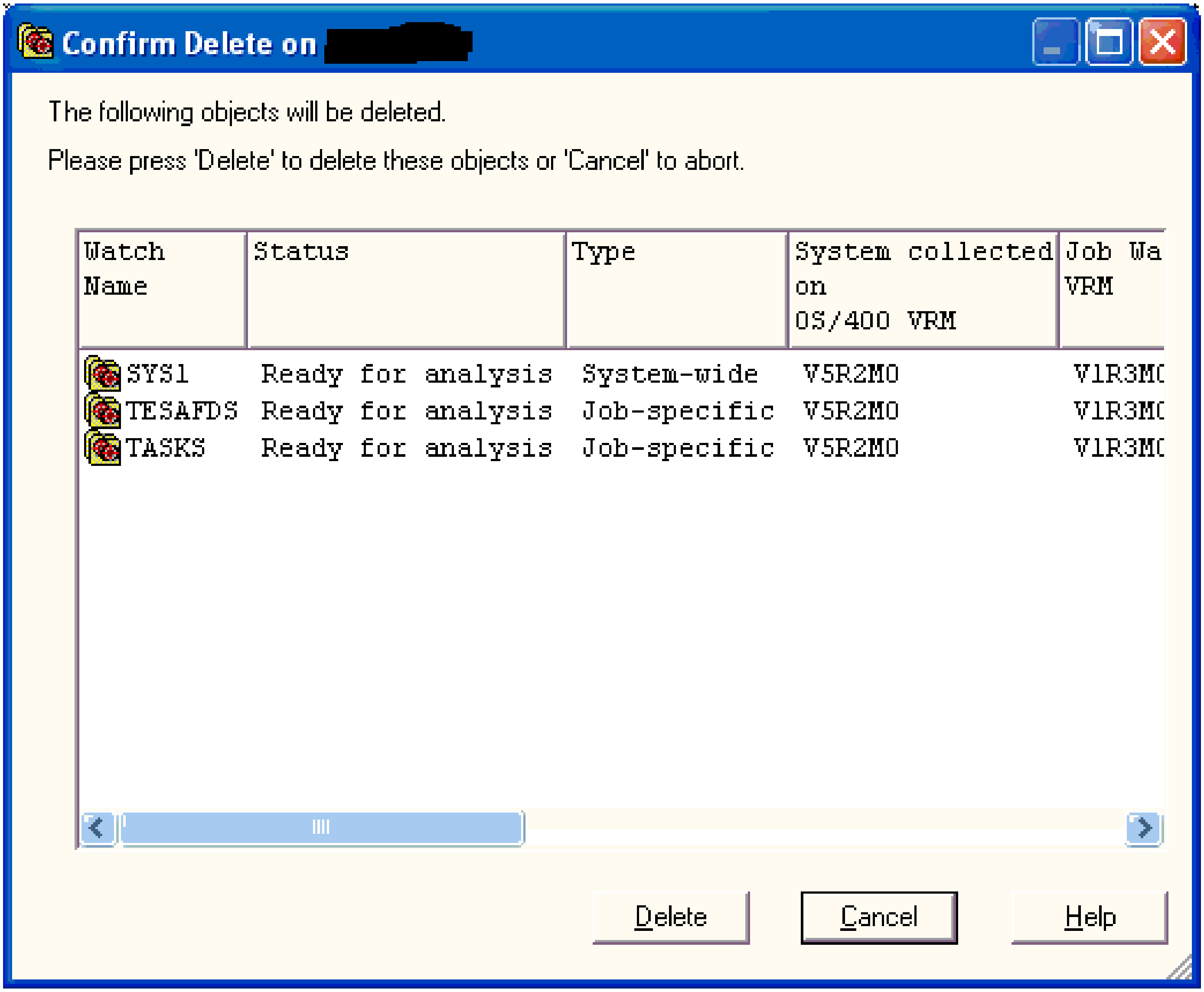
To submit your request now click 'Finish'

[Job Watch Wizard - Summary Page]

Table of Contents | Previous | Next

2.4.6 Deleting

A job watch can be deleted by using the Delete... menu found by right-clicking on a Job Watch from the Job Watcher component view.
This option will execute the DLTWCH green screen command. Deleting a job watch that is still running is not allowed. The job watch must be stopped before trying to delete it.



Confirm Delete on [redacted]

The following objects will be deleted.

Please press 'Delete' to delete these objects or 'Cancel' to abort.

Watch Name	Status	Type	System collected on OS/400 VRM	Job Wa VRM
SYS1	Ready for analysis	System-wide	V5R2M0	V1R3M0
TESAFDS	Ready for analysis	Job-specific	V5R2M0	V1R3M0
TASKS	Ready for analysis	Job-specific	V5R2M0	V1R3M0

Table of Contents Previous Next

2.4.7 Stopping

An active job watch can be stopped by using the Stop menu found by right-clicking on a Job Watch from the Job Watcher component view.

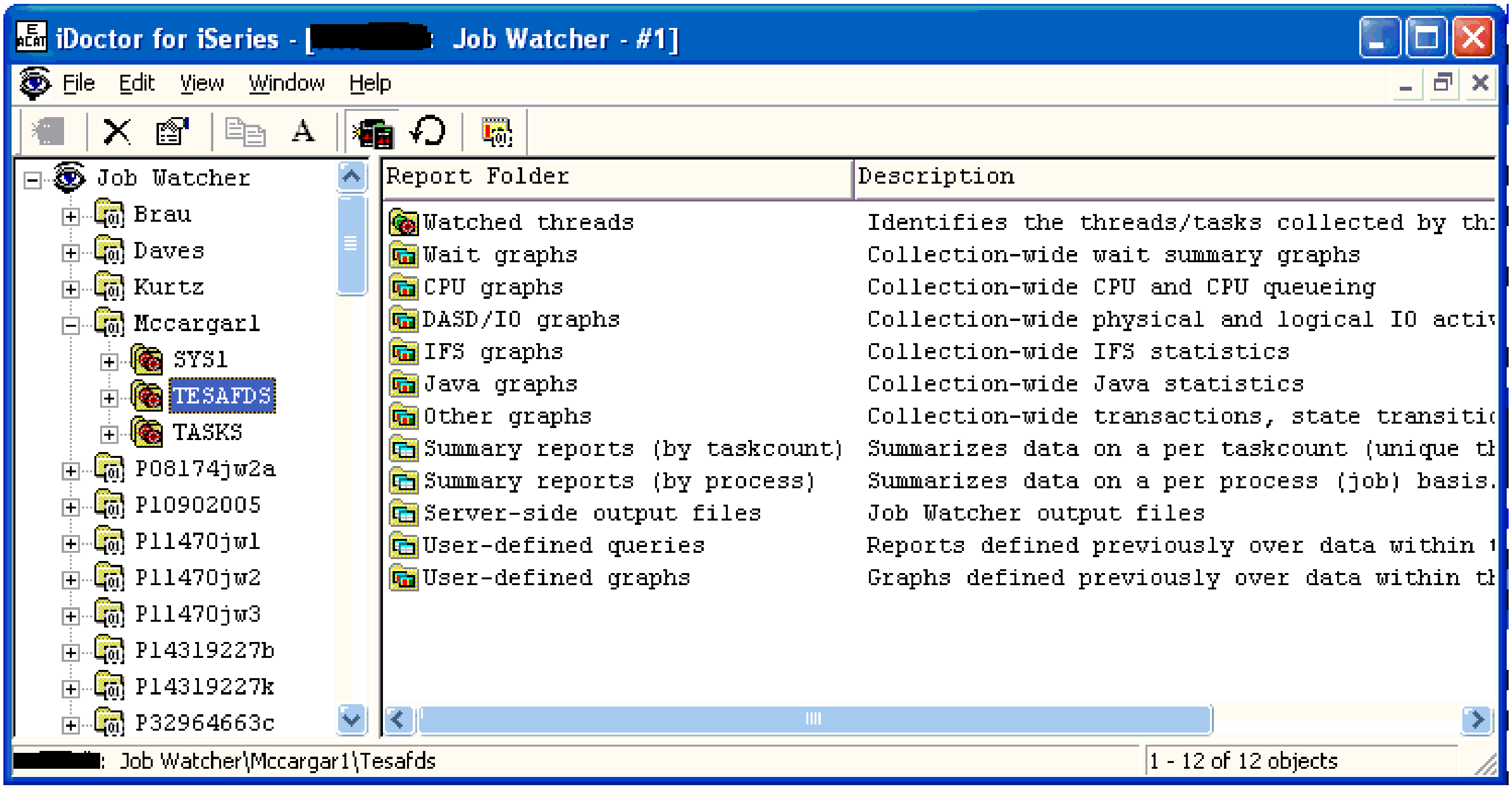
This option will issue the ENDWCHSYS (system-wide) or ENDWCHJOB (job-specific) command depending on the type of job watch selected. These commands will issue a request to have the job watch end once the current interval of collection has been completed. If the interval duration is very large, the job watch will not end until the beginning of the next interval is reached so there could be a signficant delay in the change of the status of the job watch from "In progress" to "Ready for analysis".

Table of Contents Previous Next

2.5 Job-specific Job Watch summary reports

Moving down the tree within each job-specific Job Watch are several report folders. These folders contain many different ways to look at the Job Watcher data. Right-clicking on a job watch provides options to view graphs and reports over many of the same report folders listed underneath a job-specific job watch. These popup menu options allow the user to open graphs and tables without having to expand the report folders.

An example of the contents of a job-specific Job Watch is shown below:



Report Folder	Description
Watched threads	Identifies the threads/tasks collected by th:
Wait graphs	Collection-wide wait summary graphs
CPU graphs	Collection-wide CPU and CPU queueing
DASD/IO graphs	Collection-wide physical and logical IO acti
IFS graphs	Collection-wide IFS statistics
Java graphs	Collection-wide Java statistics
Other graphs	Collection-wide transactions, state transiti
Summary reports (by taskcount)	Summarizes data on a per taskcount (unique tl
Summary reports (by process)	Summarizes data on a per process (job) basis.
Server-side output files	Job Watcher output files
User-defined queries	Reports defined previously over data within 1
User-defined graphs	Graphs defined previously over data within tl

[redacted]: Job Watcher\Mccargar1\Tesafds   1 - 12 of 12 objects

The following table describes the contents of each report folder:

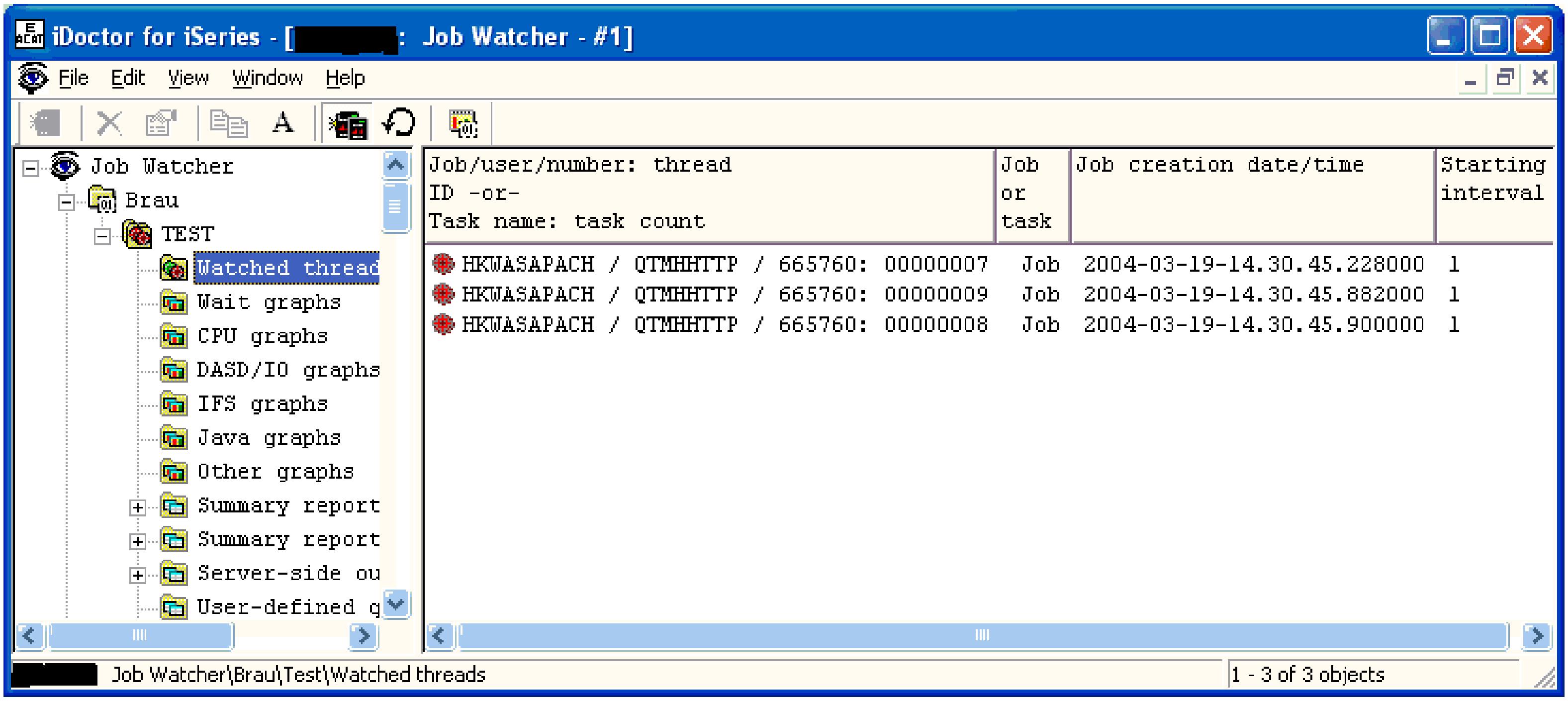
Folder	Description
Watched threads	A list of all threads and tasks that have been collected in the job-specific job watch. This folder provides quick access to the detailed reports which show information for a single job in the job watch.
Wait graphs	Contains the summary graphs related to wait analysis over the job watch. The run/wait signature graphs within this folder are a common starting point to get a high-level overview of what types of work the job's in the job watch were performing while the collection was being executed.
CPU graphs	This folder contains summary graphs illustrating CPU and CPU queueing usage for all jobs in the job watch.
DASD/IO graphs	This folder contains summary graphs showing IO operations and disk activity occurring within the job watch.
IFS graphs	This folder contains summary graphs showing IFS activity occurring within the job watch.
Java graphs	This folder contains summary graphs showing Java heap sizes for each job running Java within the job watch.
Other graphs	This folder contains summary graphs showing other types of information such as state transitions and transactions that occurred during the job watch.

Summary reports (by taskcount)	Contains summarized tables over waits, IOs, state transitions and more within the job watch. A summary record is produced per thread/task.
Summary reports (by process)	Contains summarized tables over waits, IOs, state transitions and more within the job watch. A summary record is produced for each job's primary thread.
Server-side output files	A complete list of each job-specific Job Watcher output file produced for the collection. These files are queryable through the query definition interface.
User-defined queries	Contains table views over the Job Watcher output files previously defined on the current system. These queries are stored in file QAIDRSQL04 in the current library and library QUSRSYS for system scoped queries. This folder only contains queries for data defined over job-specific Job Watcher output files.
User-defined graphs	Contains graph views over the Job Watcher output files that were previously defined on the current system. These graphs are stored in file QAIDRGPH08 in the current library and library QUSRSYS for system scoped graphs. This folder only contains graphs for data defined over job-specific Job Watcher output files.

Table of Contents | Previous | Next

2.5.1 Watched threads

The watched threads folder provides quick access to the list of jobs/threads/tasks that were collected by the job-specific job watch. Only basic information about each thread/task is available such as the job/thread/task name, the time the job was created and the interval when the job/thread was initially captured. In most cases the starting interval will be 1. However, if threads are being created for the jobs being watched as collection is in progress the starting interval will be greater than 1.



iDoctor for iSeries - [[redacted]: Job Watcher - #1]

Job/user/number: thread ID -or- Task name: task count	Job or task	Job creation date/time	Starting interval
HKWASAPACH / QTMHHTTP / 665760: 00000007	Job	2004-03-19-14.30.45.228000	1
HKWASAPACH / QTMHHTTP / 665760: 00000009	Job	2004-03-19-14.30.45.882000	1
HKWASAPACH / QTMHHTTP / 665760: 00000008	Job	2004-03-19-14.30.45.900000	1

[redacted] Job Watcher\Brau\Test\Watched threads   1 - 3 of 3 objects

The following table illustrates the menu options available by right-clicking on a job/thread/task within the Watched threads folder. The job/thread/task right-clicked on is referred to as the 'selected job' in these descriptions.

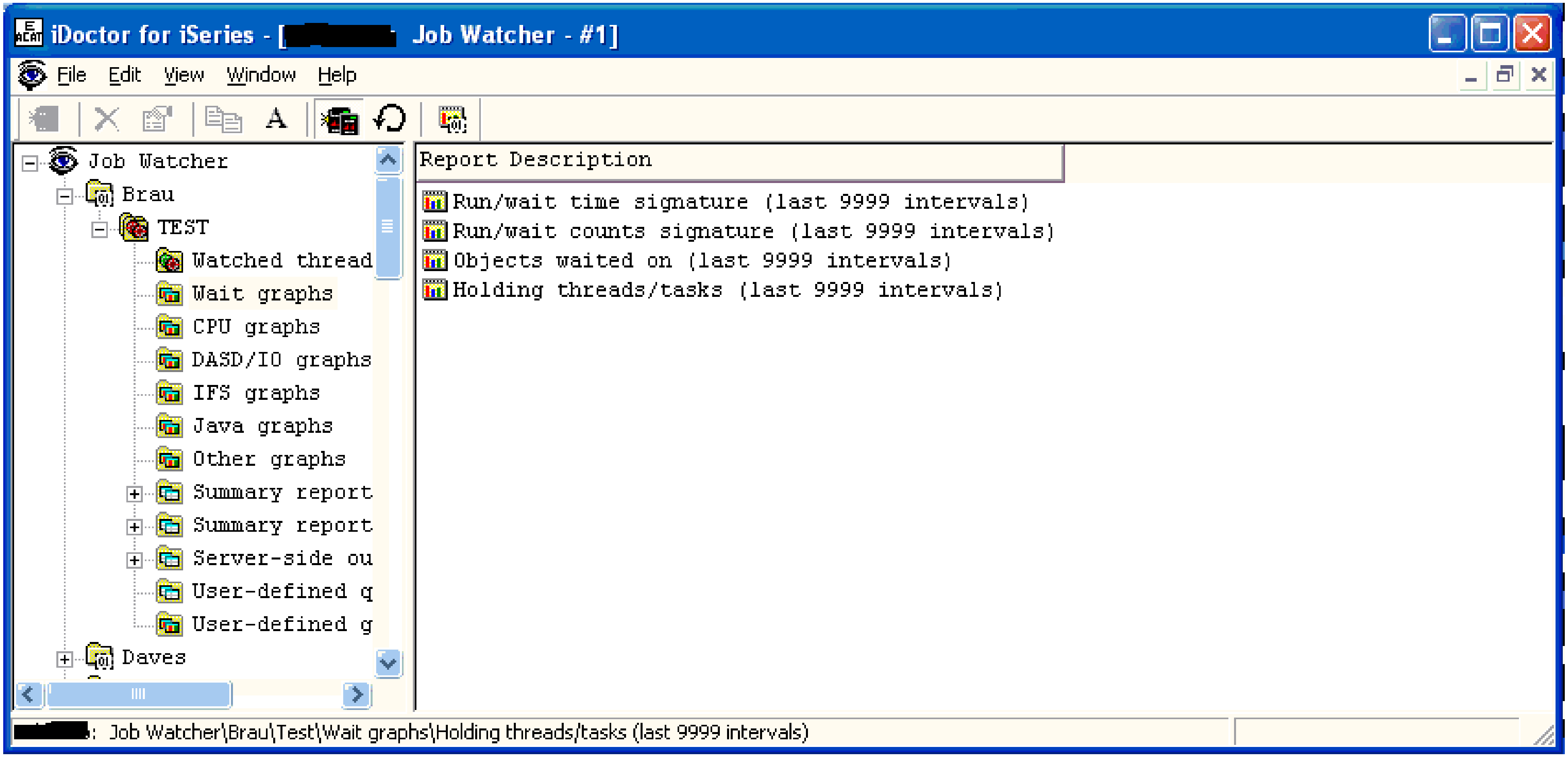
Menu	Description
Record quick view	Displays a view showing all of the selected job's fields.
Display call stack	Shows the interval details property pages for the selected job with the call stack page of interval details having initial focus.
Wait graphs	Contains detailed graphs showing information about the types of waits, wait objects and holding jobs that were detected for the selected job.
CPU graphs	Contains detailed graphs showing CPU and CPU queueing use for the selected job.
DASD/IO graphs	Contains detailed graphs showing IO and disk usage for the selected job.
IFS graphs	Contains detailed graphs showing IFS statistics for the selected job.
Java graphs	Contains detailed graphs showing Java heap sizes (if any active JVM is present) for the selected job.
Other graphs	Contains detailed graphs for other areas of interest such as transactions, state transitions, and numeric exceptions for the selected job.
Details reports	Contains a list of table views each showing a specific type of performance information related to the selected job.

Table of Contents | Previous | Next

2.5.2 Wait graphs - summarized

This folder contains a list of graphs which contain summary data over the job-specific job watch relating to waits. These graphs include the run/wait time signature graph which paints a visual picture of how much time the jobs in the job watch spent in various types of waits IO, lock conflict, journalling. These graphs also include reports on the objects waited on (which are objects on the system that job's in the job watch are waiting to use).

An example of the contents of the wait graphs folder for a job-specific job watch is:



Report Description

- Run/wait time signature (last 9999 intervals)
- Run/wait counts signature (last 9999 intervals)
- Objects waited on (last 9999 intervals)
- Holding threads/tasks (last 9999 intervals)

[redacted]: Job Watcher\Brau\Test\Wait graphs\Holding threads/tasks (last 9999 intervals)

The following table illustrates the menu options available by right-clicking on a graph in the list.

Menu	Description
Open graph(s)	Opens the selected graph(s) into a Data Viewer. If a Data Viewer has already been opened submenus will appear underneath this menu in order to give the option of opening the graph(s) into an already open Data Viewer.

Table of Contents Previous Next

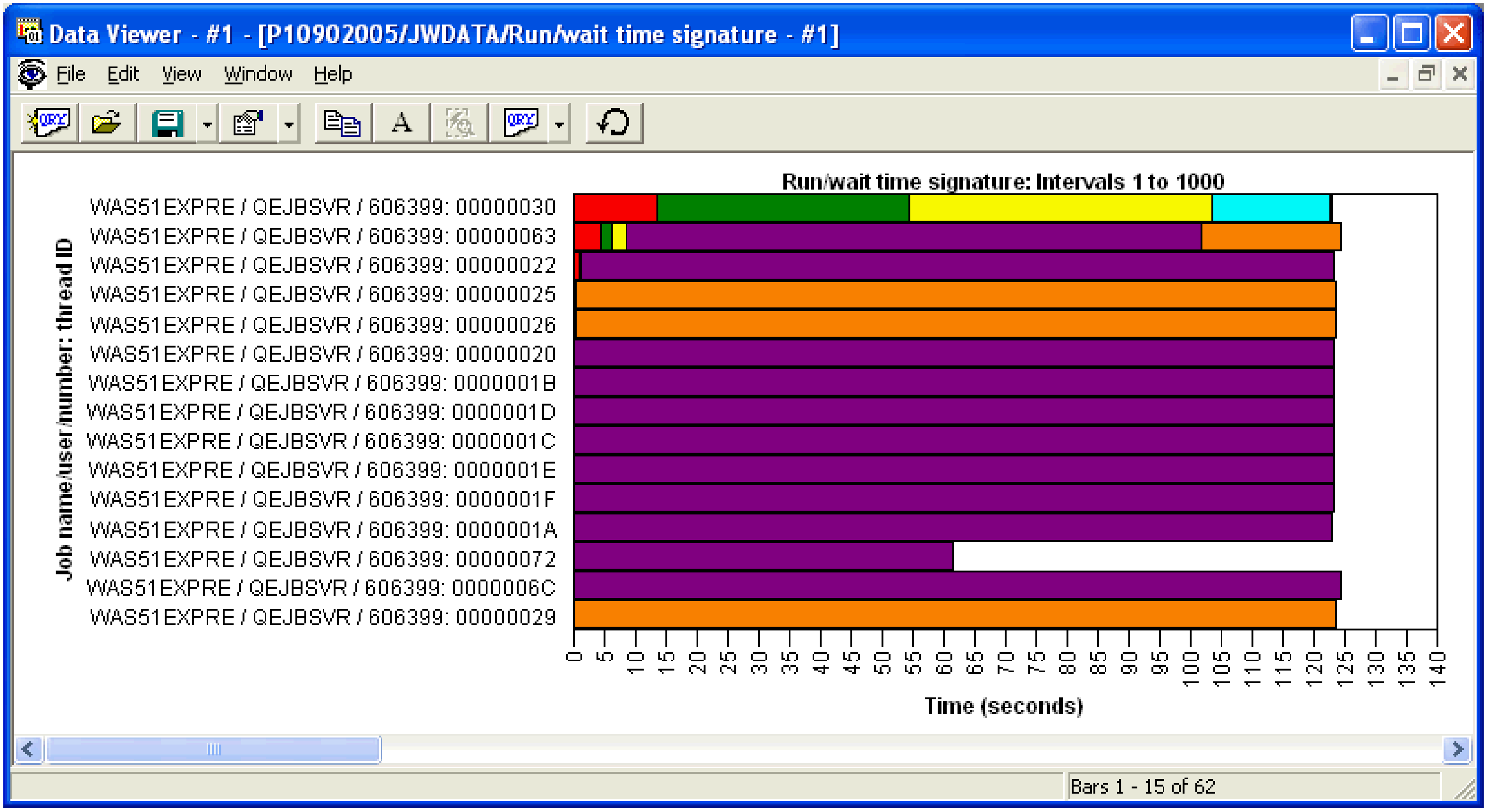
2.5.2.1 Run/wait time signature

Description: This graph shows a **summary** of the total time a job spent in each type of wait over the complete interval range specified within the job watch. The graph makes use of the 32 run/wait buckets used in Job Watcher. To determine what a color represents, place the mouse pointer of a color/bar of interest and a description of the wait as well as other statistics will be displayed.

This graph is a good starting point because it provides a high-level overview of what a job was waiting on or doing during the collection. Each bar in the graph represents a job in the job watch. If a particular job has a higher percentage of seize conflict for example than others, additional details about that job can be determined by right-clicking on the bar and choosing one of the detail graph menus. The detail graphs provide information about a single job over time (1 bar per interval).

Graph Type: summarized job-specific (stacked horizontal bar)
X-axis: Fully qualified job name and thread id or taskname. One bar is produced per job/thread or task.
Y-Axis: Each color represents the amount of time the job/thread/task spent in one of the 32 different wait buckets. Time is listed in seconds.

Example:


Data Viewer - #1 - [P10902005/JWDATA/Run/wait time signature - #1]
File Edit View Window Help
Run/wait time signature: Intervals 1 to 1000
Job name/user/number: thread ID
WAS51EXPRE / QEJBSVR / 606399: 00000030
WAS51EXPRE / QEJBSVR / 606399: 00000063
WAS51EXPRE / QEJBSVR / 606399: 00000022
WAS51EXPRE / QEJBSVR / 606399: 00000025
WAS51EXPRE / QEJBSVR / 606399: 00000026
WAS51EXPRE / QEJBSVR / 606399: 00000020
WAS51EXPRE / QEJBSVR / 606399: 0000001B
WAS51EXPRE / QEJBSVR / 606399: 0000001D
WAS51EXPRE / QEJBSVR / 606399: 0000001C
WAS51EXPRE / QEJBSVR / 606399: 0000001E
WAS51EXPRE / QEJBSVR / 606399: 0000001F
WAS51EXPRE / QEJBSVR / 606399: 0000001A
WAS51EXPRE / QEJBSVR / 606399: 00000072
WAS51EXPRE / QEJBSVR / 606399: 0000006C
WAS51EXPRE / QEJBSVR / 606399: 00000029
0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100 105 110 115 120 125 130 135 140
Time (seconds)
Bars 1 - 15 of 62

Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.
CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.
DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.
Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.
Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.
Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.

Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

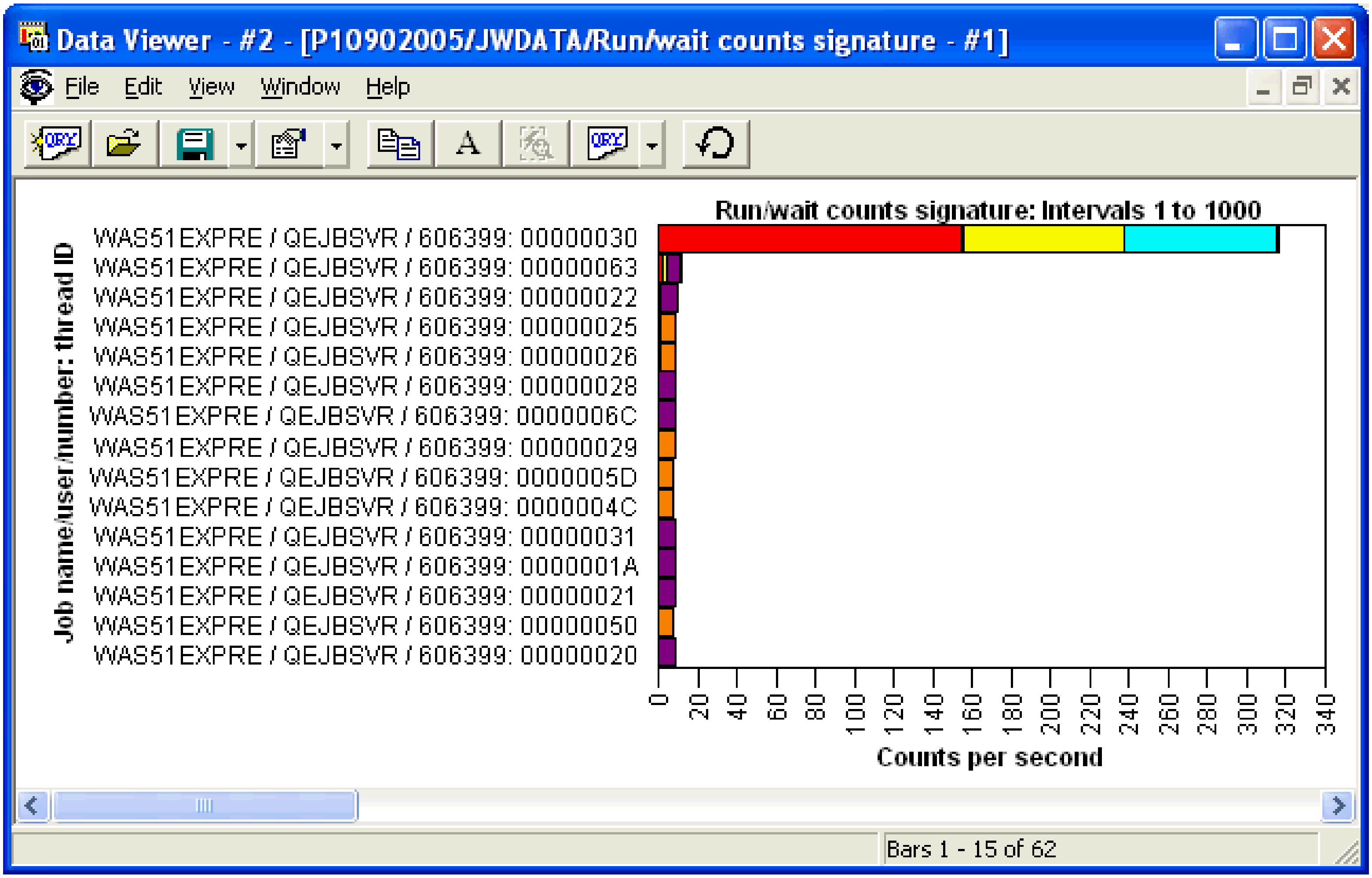
2.5.2.2 Run/wait counts signature

Description: This graph shows a **summary** of the rate of the total occurrences of each type of wait per job over the complete interval range specified within the job watch. Every time a job goes into a wait (or uses CPU) this is 'counted' into the appropriate wait bucket it belogs to. When the job changes to a different type of wait/CPU the counter is updated again. The graph makes use of the 32 run/wait buckets used in Job Watcher. To determine what a color represents, place the mouse pointer of a color/bar of interest and a description of the wait as well as other statistics will be displayed.

This graph shows jobs that have a high amount of activity. A very short bar indicates that transitions from one type of a wait to another did not occur very often.

Graph Type: summarized job-specific (stacked horizontal bar)
X-axis: Fully qualified job name and thread id or taskname. One bar is produced per job/thread or task.
Y-Axis: Each color represents the rate of occurrences per second for one of the 32 different wait buckets. The rate is calculated using the total time the job was active within the job watch.

Example:


Data Viewer - #2 - [P10902005/JWDATA/Run/wait counts signature - #1]
File
Edit
View
Window
Help
Run/wait counts signature: Intervals 1 to 1000
Job name/user/number: thread ID
WAS51EXPRE / QEJBSVR / 606399: 00000030
WAS51EXPRE / QEJBSVR / 606399: 00000063
WAS51EXPRE / QEJBSVR / 606399: 00000022
WAS51EXPRE / QEJBSVR / 606399: 00000025
WAS51EXPRE / QEJBSVR / 606399: 00000026
WAS51EXPRE / QEJBSVR / 606399: 00000028
WAS51EXPRE / QEJBSVR / 606399: 0000006C
WAS51EXPRE / QEJBSVR / 606399: 00000029
WAS51EXPRE / QEJBSVR / 606399: 0000005D
WAS51EXPRE / QEJBSVR / 606399: 0000004C
WAS51EXPRE / QEJBSVR / 606399: 00000031
WAS51EXPRE / QEJBSVR / 606399: 0000001A
WAS51EXPRE / QEJBSVR / 606399: 00000021
WAS51EXPRE / QEJBSVR / 606399: 00000050
WAS51EXPRE / QEJBSVR / 606399: 00000020
0
20
40
60
80
100
120
140
160
180
200
220
240
260
280
300
320
340
Counts per second
Bars 1 - 15 of 62

Right-clicking on a bar in this graph provides the following menu options:

Menu	**Description**
Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.
CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.
DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.
Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.
Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.
Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

Table of Contents Previous Next

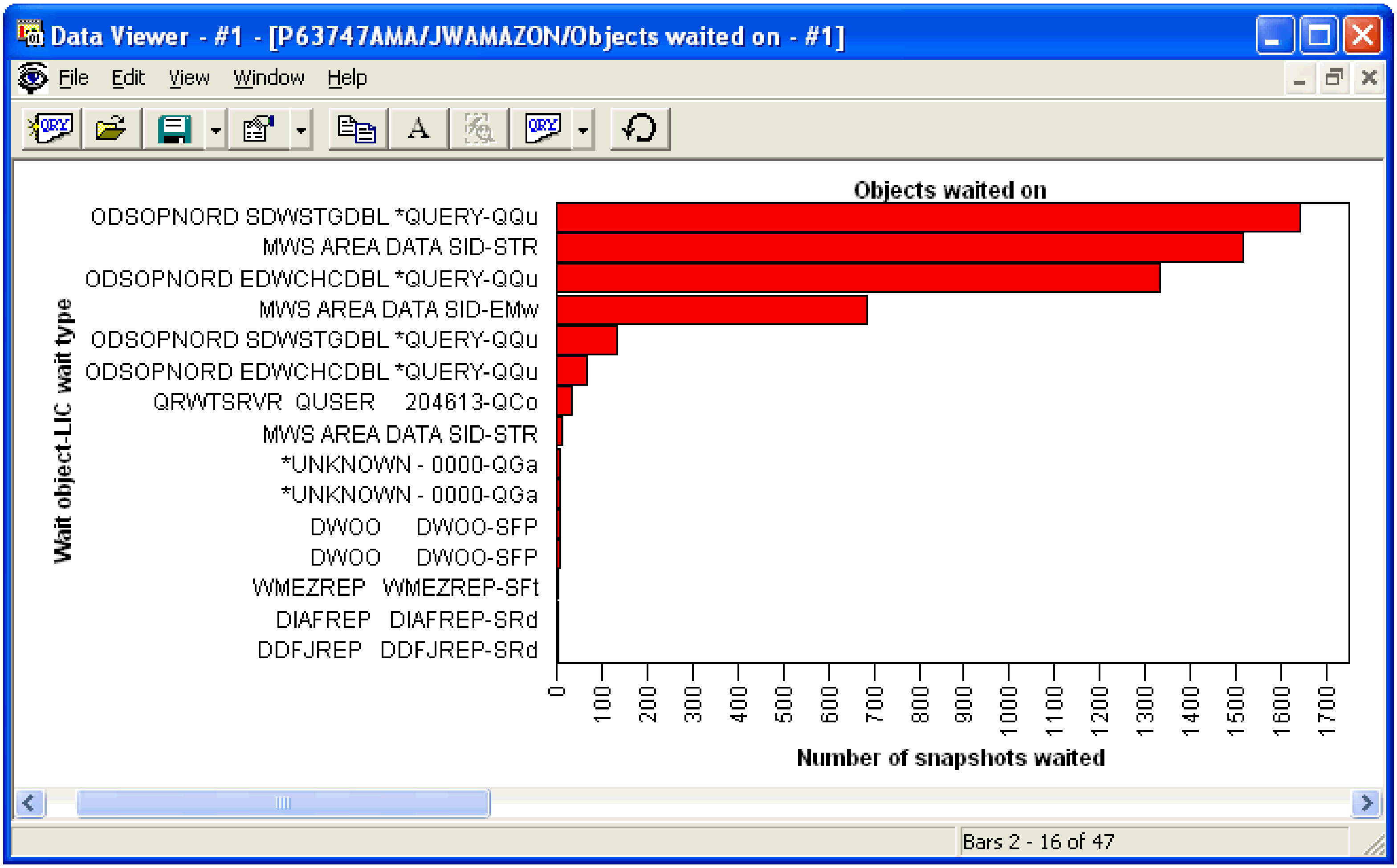
2.5.2.3 Objects waited on

Description: This graph shows a **summary** of the objects (including low level system objects) that were waited on by any jobs/threads/tasks collected by the job watch. Unlike the other summary graphs this graph does not summarize the data over a specific range of intervals. This graph's data is a summary over the entire job watch.

This graph shows the object information and low level wait type for threads that when sampled had the highest occurrences of the listed object being flagged as the "wait object".

Graph Type: summarized job-specific (horizontal bar)
X-axis: Wait object name and the 3 character LIC wait type. One bar is produced per unique object address per job/thread found in the data.
Y-Axis: The total number of snapshots that the object was listed as being the waiter. Placing the mouse pointer over a bar will identify the job/thread/task that waited on the selected object.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.
CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.

DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.
Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.
Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.
Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

Table of Contents Previous Next

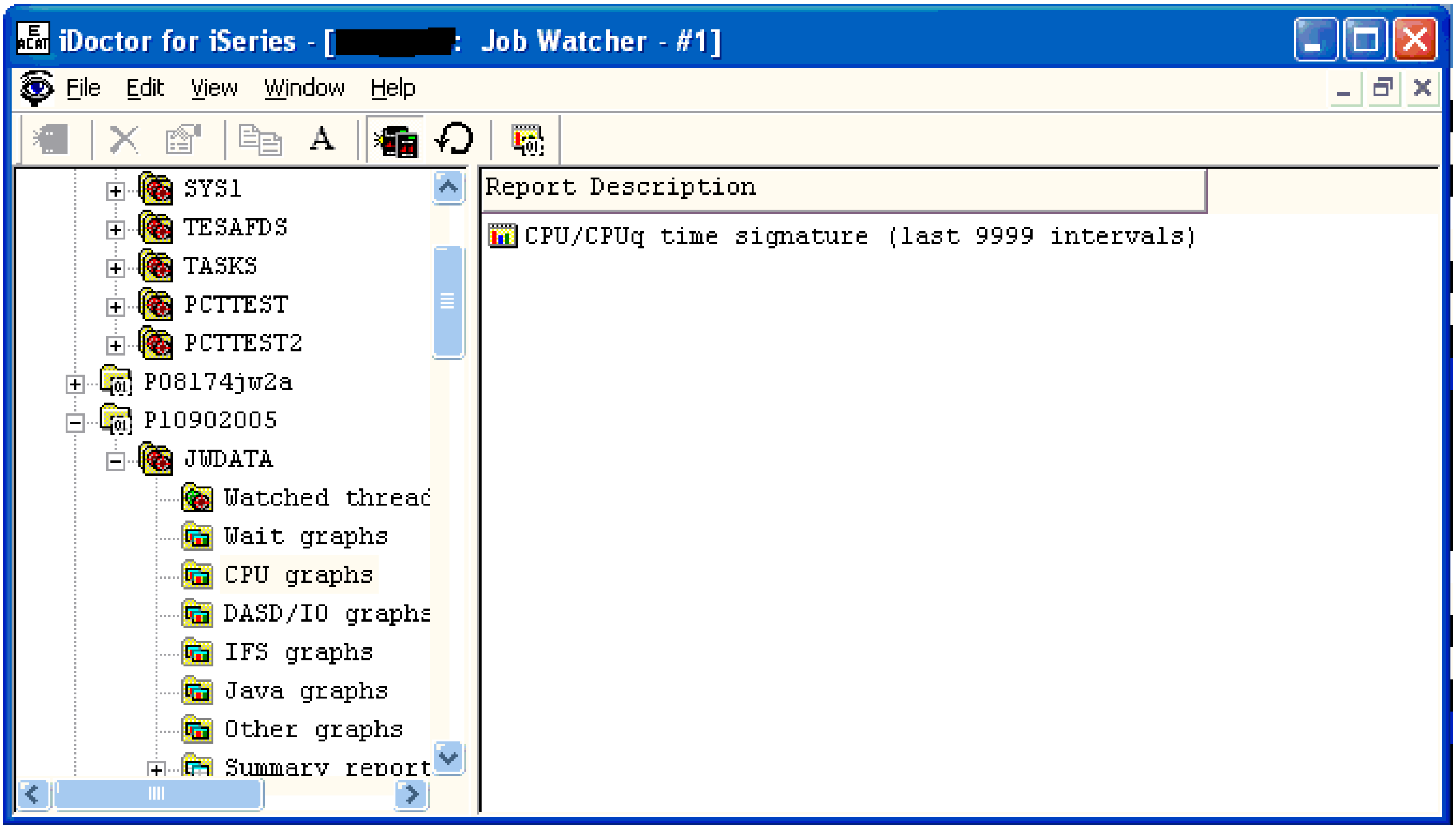
2.5.2.4 Holding threads/tasks

Table of Contents | Previous | Next

2.5.3 CPU graphs - summarized

This folder contains a list of graphs which contain summary data over the job-specific job watch relating to CPU and CPU queueing. Currently this folder contains a single graph showing the the list of jobs/threads/tasks in the job watch ranked by those using the highest CPU.

An example of the contents of the CPU graphs folder for a job-specific job watch is:



Report Description
CPU/CPUq time signature (last 9999 intervals)

The following table illustrates the menu options available by right-clicking on a graph in the list.

Menu	Description
Open graph(s)	Opens the selected graph(s) into a Data Viewer. If a Data Viewer has already been opened submenus will appear underneath this menu in order to give the option of opening the graph(s) into an already open Data Viewer.

Table of Contents | Previous | Next

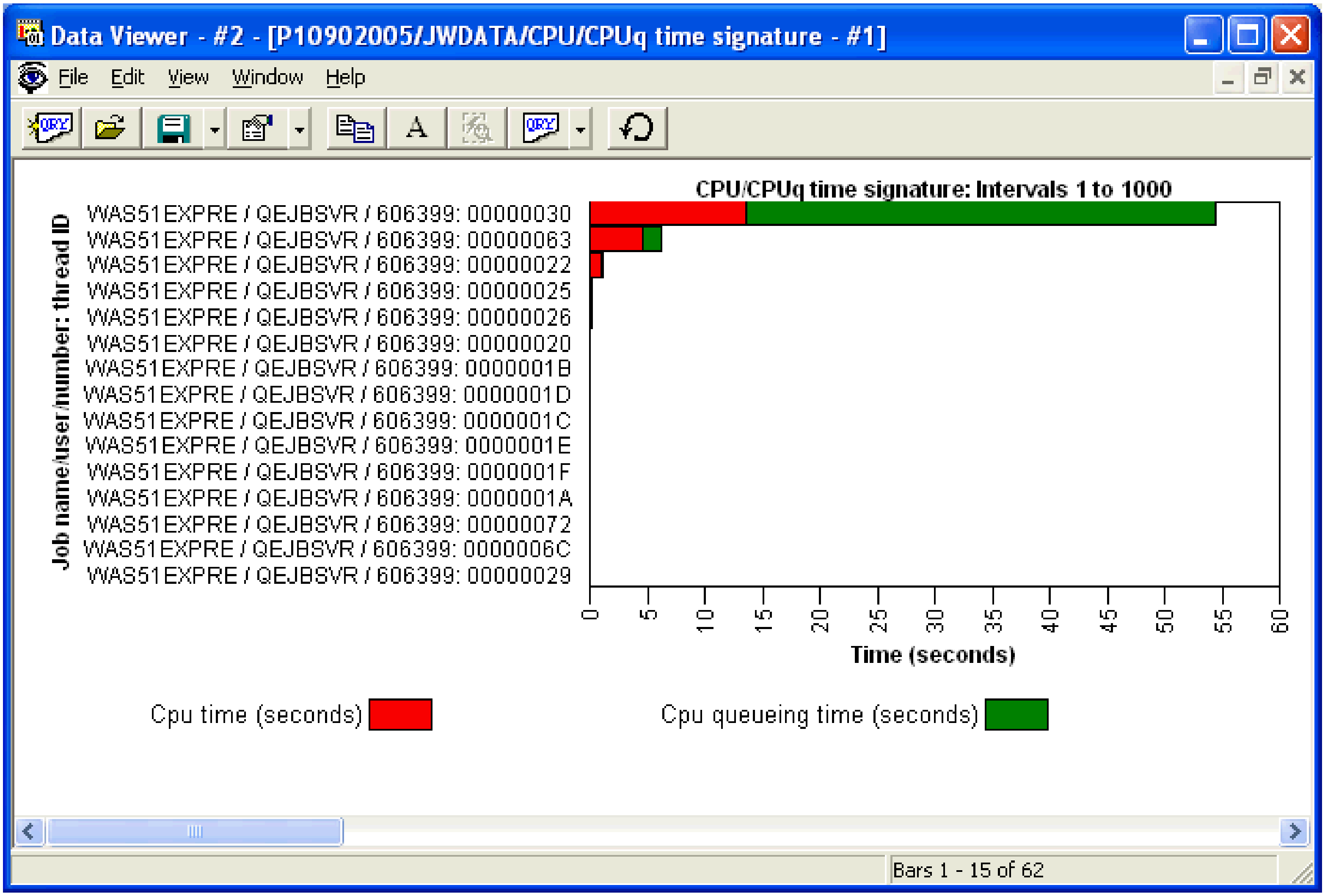
2.5.3.1 CPU/CPUq time signature

Description: This graph shows a **summary** of the total CPU and CPU queueing usage per job for all jobs/threads/tasks in the job watch over the interval range specified. The CPU and CPU queueing is provided as the total time spent for each job/thread or task. The red color represents CPU usage and the green color represents CPU queueing.

This graph can be useful to quickly see from the entire job watch the threads or tasks that used the most CPU.

Graph Type: summarized job-specific (stacked horizontal bar)
X-axis: Fully qualified job name and thread id or taskname. One bar is produced per job/thread or task.
Y-Axis: Each color represents the amount of CPU time or CPU queueing time the job/thread/task had. Time is listed in seconds.

Example:


Data Viewer - #2 - [P10902005/JWDATA/CPU/CPUq time signature - #1]
File
Edit
View
Window
Help
CPU/CPUq time signature: Intervals 1 to 1000
Job name/user/number: thread ID
WAS51EXPRE / QEJBSVR / 606399: 00000030
WAS51EXPRE / QEJBSVR / 606399: 00000063
WAS51EXPRE / QEJBSVR / 606399: 00000022
WAS51EXPRE / QEJBSVR / 606399: 00000025
WAS51EXPRE / QEJBSVR / 606399: 00000026
WAS51EXPRE / QEJBSVR / 606399: 00000020
WAS51EXPRE / QEJBSVR / 606399: 0000001B
WAS51EXPRE / QEJBSVR / 606399: 0000001D
WAS51EXPRE / QEJBSVR / 606399: 0000001C
WAS51EXPRE / QEJBSVR / 606399: 0000001E
WAS51EXPRE / QEJBSVR / 606399: 0000001F
WAS51EXPRE / QEJBSVR / 606399: 0000001A
WAS51EXPRE / QEJBSVR / 606399: 00000072
WAS51EXPRE / QEJBSVR / 606399: 0000006C
WAS51EXPRE / QEJBSVR / 606399: 00000029
0
5
10
15
20
25
30
35
40
45
50
55
60
Time (seconds)
Cpu time (seconds) 🟥
Cpu queueing time (seconds) 🟩
Bars 1 - 15 of 62

Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.
CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.

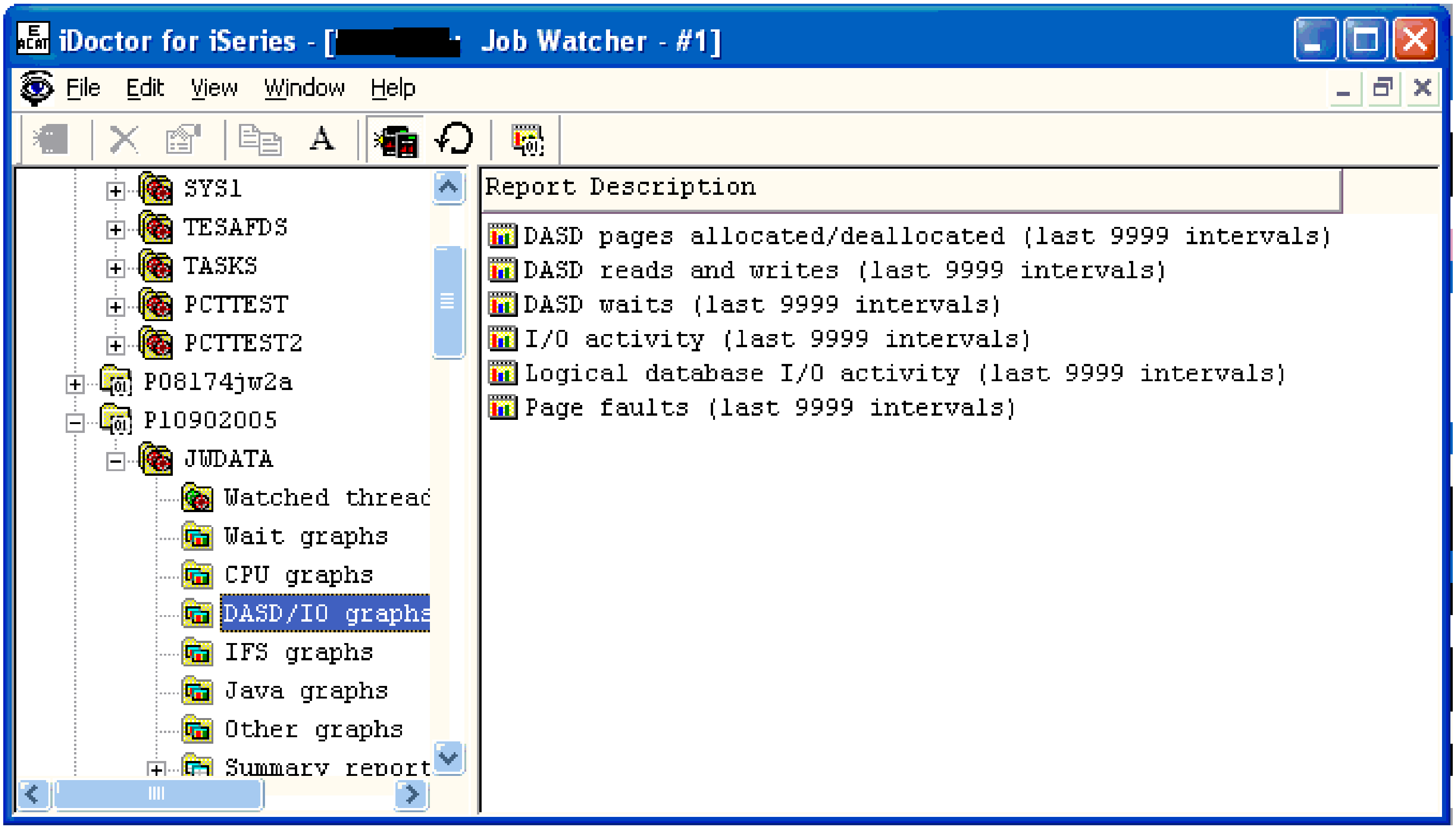
DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.
Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.
Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.
Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

Table of Contents Previous Next

2.5.4 DASD/IO graphs - summarized

This folder contains a list of graphs which contain summary data over the job-specific job watch relating to DASD and I/O. These graphs include the DASD pages allocated/deallocated graph, DASD reads and writes, I/O activity, LDIO activity and page faults.

An example of the contents of the DASD/IO graphs folder for a job-specific job watch is:



iDoctor for iSeries - [[redacted] Job Watcher - #1]

Report Description
DASD pages allocated/deallocated (last 9999 intervals)
DASD reads and writes (last 9999 intervals)
DASD waits (last 9999 intervals)
I/O activity (last 9999 intervals)
Logical database I/O activity (last 9999 intervals)
Page faults (last 9999 intervals)

The following table illustrates the menu options available by right-clicking on a graph in the list.

Menu	Description
Open graph(s)	Opens the selected graph(s) into a Data Viewer. If a Data Viewer has already been opened submenus will appear underneath this menu in order to give the option of opening the graph(s) into an already open Data Viewer.

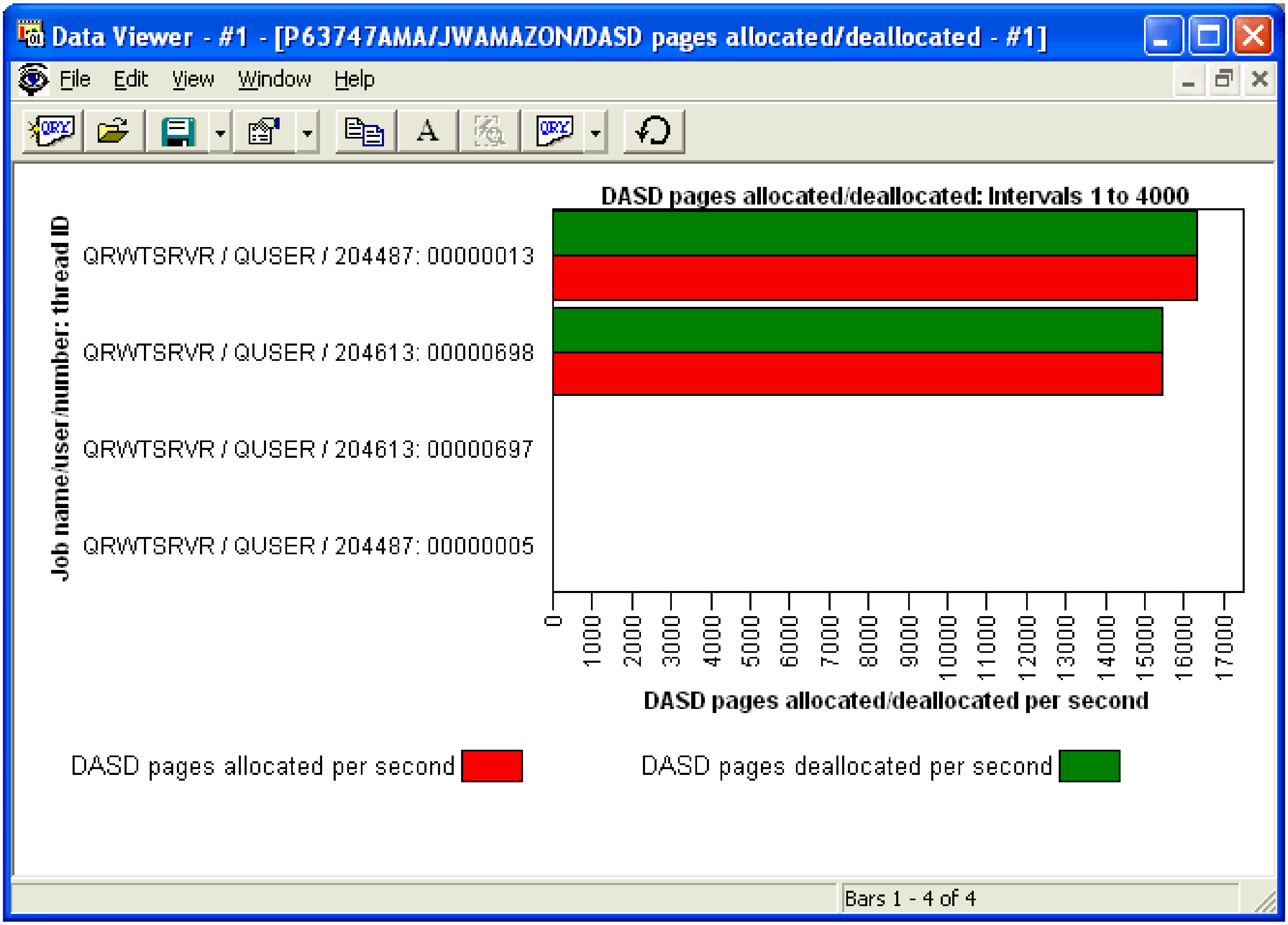
Table of Contents Previous Next

2.5.4.1 DASD pages allocated/deallocated

Description: This graph shows a **summary** of the rate of pages allocated and deallocated for each job/thread in the job watch over the interval range specified. Each job/thread/task has a red bar (allocations) and a green bar (deallocations) associated with it showing a comparison between the rate of allocations vs deallocations. A "DASD page" is a 4k (4096 bytes) block of data.
This graph can be useful to quickly see from the entire job watch the threads or tasks that performed the most allocations or deallocations of DASD space.

Graph Type: summarized job-specific (horizontal bar)
X-axis: Fully qualified job name and thread id or taskname. One set of bars is produced per job/thread or task.
Y-Axis: Each color represents number of DASD pages allocated or deallocated per second. Red is used for allocations and green is used for deallocations.

Example:


Data Viewer - #1 - [P63747AMA/JWAMAZON/DASD pages allocated/deallocated - #1]
File
Edit
View
Window
Help
DASD pages allocated/deallocated: Intervals 1 to 4000
Job name/user/number: thread ID
QRWTSRVR / QUSER / 204487: 00000013
QRWTSRVR / QUSER / 204613: 00000698
QRWTSRVR / QUSER / 204613: 00000697
QRWTSRVR / QUSER / 204487: 00000005
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
11000
12000
13000
14000
15000
16000
17000
DASD pages allocated/deallocated per second
DASD pages allocated per second 🟥
DASD pages deallocated per second 🟩
Bars 1 - 4 of 4

Right-clicking on a bar in this graph provides the following menu options:

Menu	**Description**
Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.
CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.
DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.
Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.
Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.
Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

Table of Contents | Previous | Next

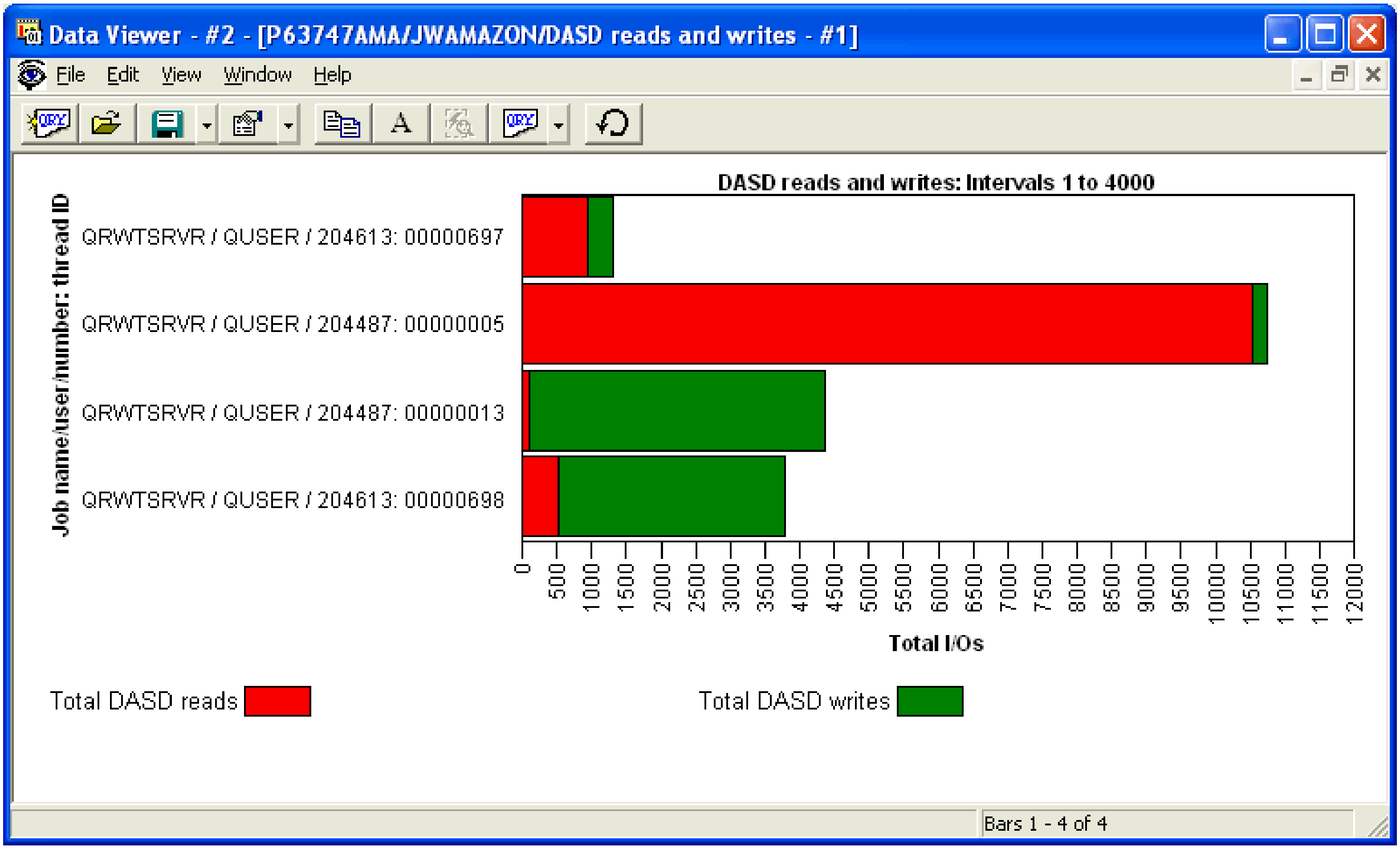
2.5.4.2 DASD reads and writes

Description: This graph shows a **summary** of the total number of reads and writes performed by each job/thread in the job watch over the interval range specified. The values shown are counts and not rates.

This graph can be useful to quickly see within the entire job watch the threads or tasks that performed the most reads and/or writes.

Graph Type: summarized job-specific (stacked horizontal bar)
X-axis: Fully qualified job name and thread id or taskname. One set of bars is produced per job/thread or task.
Y-Axis: Each color represents total DASD reads or total DASD writes for each job/thread in the job watch.

Example:


Data Viewer - #2 - [P63747AMA/JWAMAZON/DASD reads and writes - #1]
File
Edit
View
Window
Help
DASD reads and writes: Intervals 1 to 4000
Job name/user/number: thread ID
QRWTSRVR / QUSER / 204613: 00000697
QRWTSRVR / QUSER / 204487: 00000005
QRWTSRVR / QUSER / 204487: 00000013
QRWTSRVR / QUSER / 204613: 00000698
0
500
1000
1500
2000
2500
3000
3500
4000
4500
5000
5500
6000
6500
7000
7500
8000
8500
9000
9500
10000
10500
11000
11500
12000
Total I/Os
Total DASD reads
Total DASD writes
Bars 1 - 4 of 4

Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.
CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.
DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.
Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.
Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.

Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

Table of Contents Previous Next

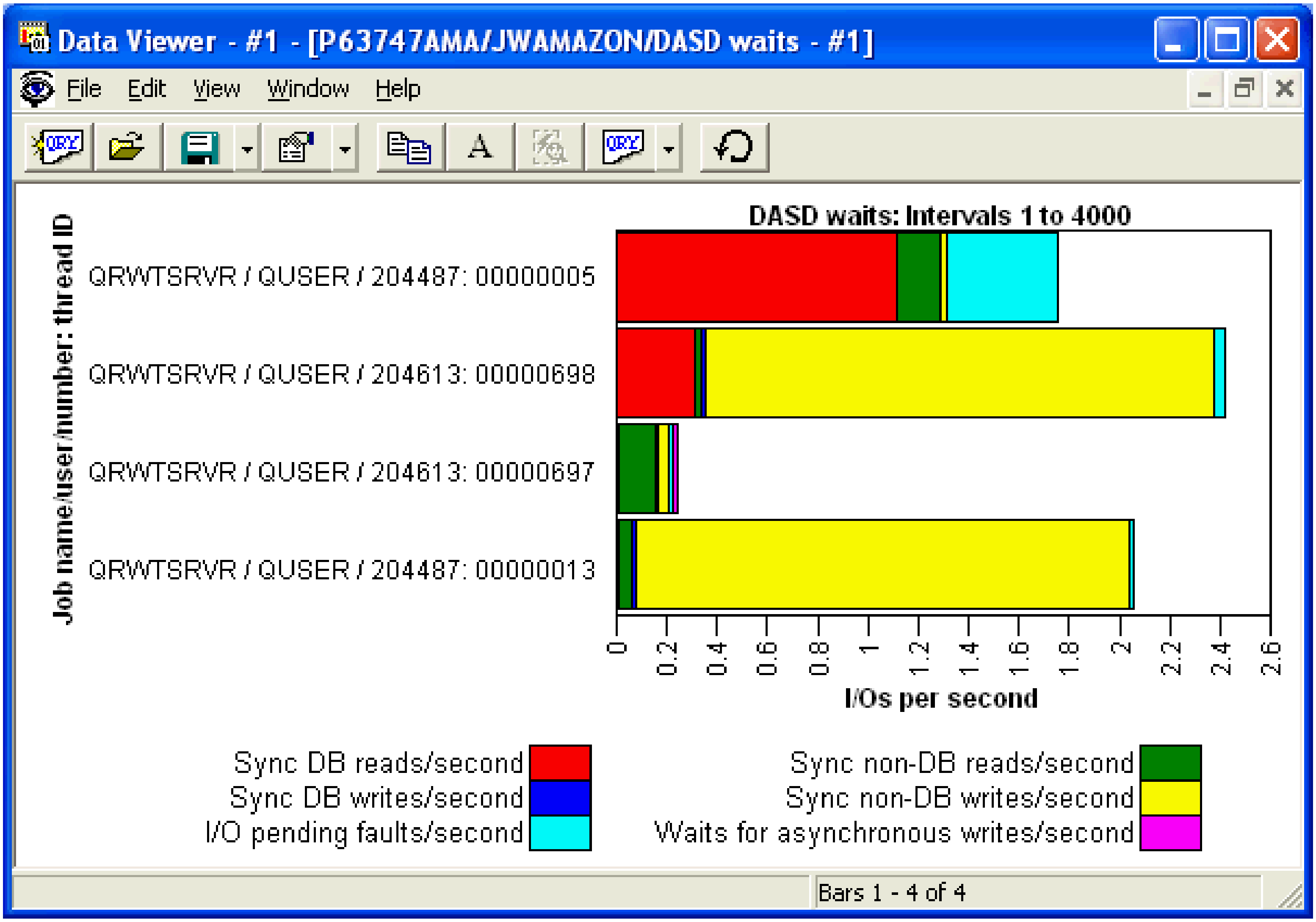
2.5.4.3 DASD waits

Description: This graph shows a **summary** of the rates of synchronous and asynchronous reads and writes as the page fault rate for each job/thread in the job watch over the interval range specified. Each type of statistics has a different color represented by the legend on this graph.

This graph can be useful to quickly see from the entire job watch which jobs had higher rates of the statistics shown.

Graph Type: summarized job-specific (stacked horizontal bar)
X-axis: Fully qualified job name and thread id or taskname. One set of bars is produced per job/thread or task.
Y-Axis: Each color represents a different type of I/O rate per second.

Example:


Data Viewer - #1 - [P63747AMA/JWAMAZON/DASD waits - #1]
File
Edit
View
Window
Help
QRY
A
QRY
DASD waits: Intervals 1 to 4000
Job name/user/number: thread ID
QRWTSRVR / QUSER / 204487: 00000005
QRWTSRVR / QUSER / 204613: 00000698
QRWTSRVR / QUSER / 204613: 00000697
QRWTSRVR / QUSER / 204487: 00000013
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
2.2
2.4
2.6
I/Os per second
Sync DB reads/second 🟥
Sync DB writes/second 🟦
I/O pending faults/second 🟦
Sync non-DB reads/second 🟩
Sync non-DB writes/second 🟨
Waits for asynchronous writes/second 🟪
Bars 1 - 4 of 4

Right-clicking on a bar in this graph provides the following menu options:

Menu	**Description**
Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.
CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.
DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.
Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.
Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.
Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

Table of Contents Previous Next

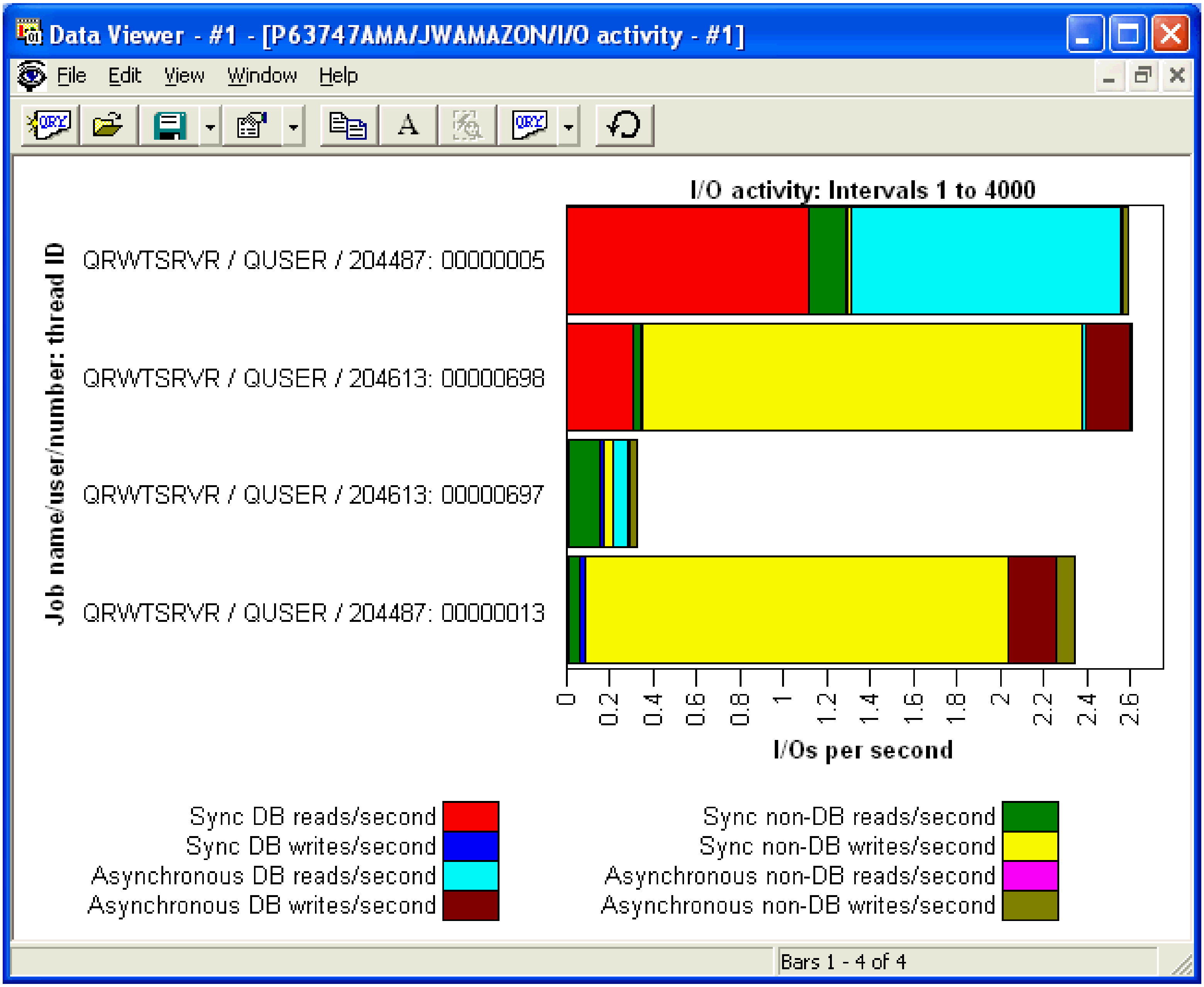
2.5.4.4 I/O activity

Description: This graph shows a **summary** of the rates of I/O activity including database and non-database synchronous and asynchronous reads and writes for each job/thread in the job watch over the interval range specified. Each bar in the graph shows the I/O activity for a specific job

This graph can be useful to quickly see from the entire job watch the threads or tasks that performed the most reads or writes and of each type.

Graph Type: summarized job-specific (stacked horizontal bar)
X-axis: Fully qualified job name and thread id or taskname. One set of bars is produced per job/thread or task.
Y-Axis: Each color represents a different type of I/O rate per second.

Example:


Data Viewer - #1 - [P63747AMA/JWAMAZON/I/O activity - #1]
File
Edit
View
Window
Help
I/O activity: Intervals 1 to 4000
Job name/user/number: thread ID
QRWTSRVR / QUSER / 204487: 00000005
QRWTSRVR / QUSER / 204613: 00000698
QRWTSRVR / QUSER / 204613: 00000697
QRWTSRVR / QUSER / 204487: 00000013
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
2.2
2.4
2.6
I/Os per second
Sync DB reads/second
Sync DB writes/second
Asynchronous DB reads/second
Asynchronous DB writes/second
Sync non-DB reads/second
Sync non-DB writes/second
Asynchronous non-DB reads/second
Asynchronous non-DB writes/second
Bars 1 - 4 of 4

Right-clicking on a bar in this graph provides the following menu options:

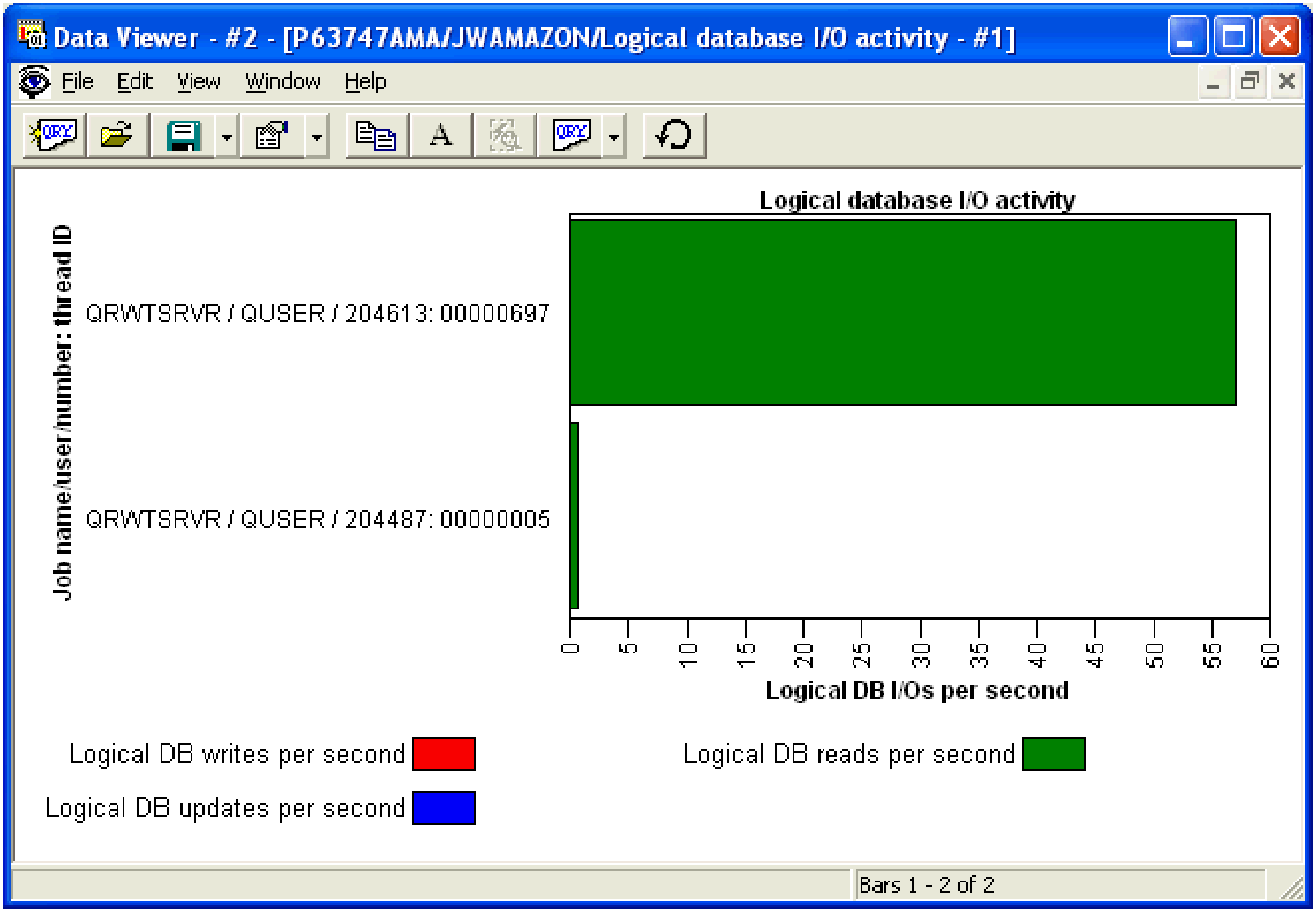
Menu	**Description**
Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.
CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.
DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.
Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.
Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.
Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

2.5.4.5 Logical database I/O activity

Description: This graph shows a **summary** of the rate of logical database I/O activity including writes, reads and updates for each job/thread in the job watch over the interval range specified.
This graph can be useful to quickly see from the entire job watch the threads or tasks that performed the most logical database IO.

Graph Type: summarized job-specific (stacked horizontal bar)
X-axis: Fully qualified job name and thread id or taskname. One set of bars is produced per job/thread or task.
Y-Axis: Each color represents a different type of LDIO rate per second.

Example:


Data Viewer - #2 - [P63747AMA/JWAMAZON/Logical database I/O activity - #1]
File
Edit
View
Window
Help
Logical database I/O activity
Job name/user/number: thread ID
QRWTSRVR / QUSER / 204613: 00000697
QRWTSRVR / QUSER / 204487: 00000005
0
5
10
15
20
25
30
35
40
45
50
55
60
Logical DB I/Os per second
Logical DB writes per second 🟥
Logical DB reads per second 🟩
Logical DB updates per second 🟦
Bars 1 - 2 of 2

Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.

CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.
DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.
Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.
Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.
Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

Table of Contents | Previous | Next

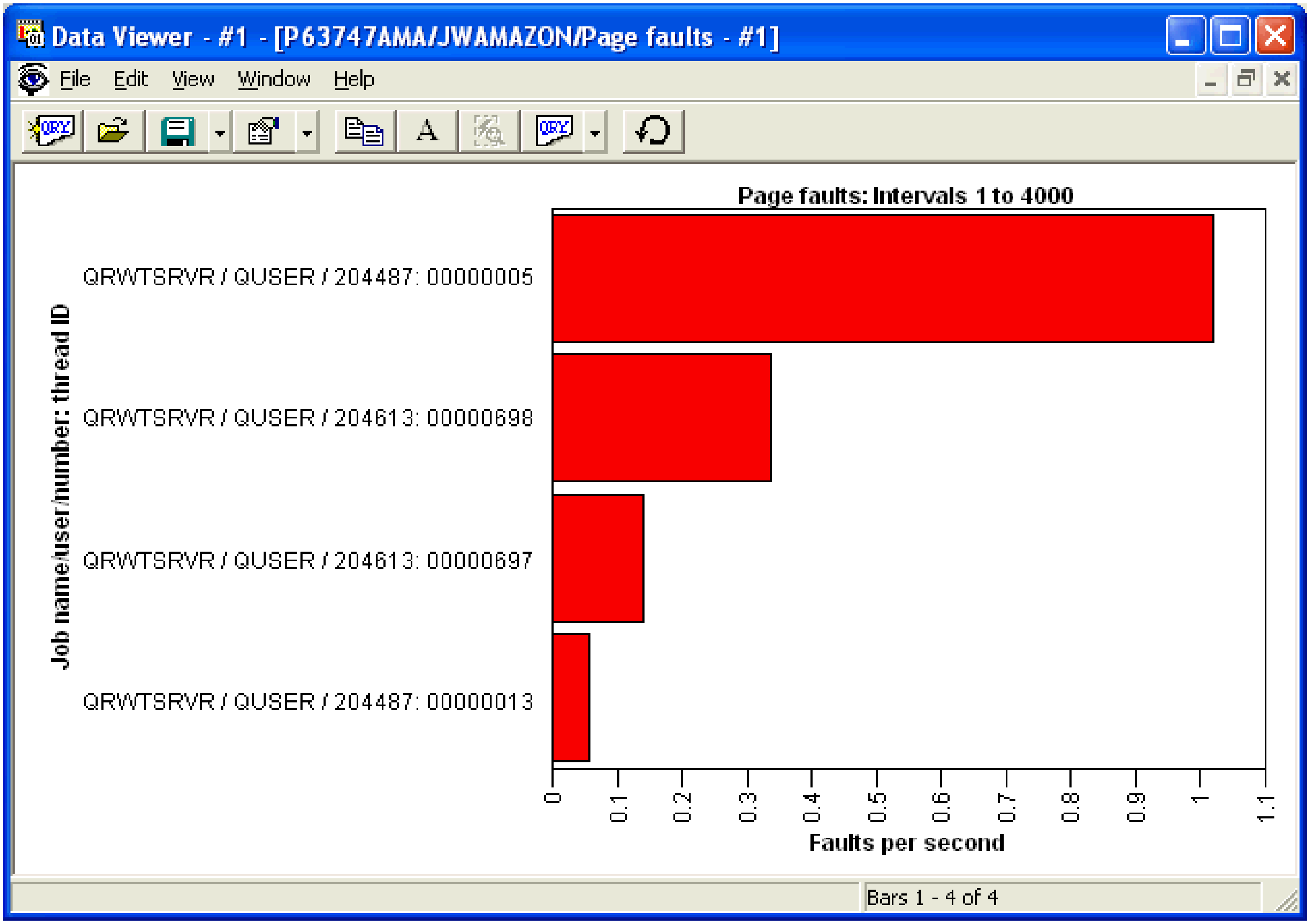
2.5.4.6 Page faults

Description: This graph shows a **summary** of the rate of page faults for each job/thread in the job watch over the interval range specified.

This graph can be useful to quickly see within the entire job watch the threads or tasks that had the highest fault rate.

Graph Type: summarized job-specific (stacked horizontal bar)
X-axis: Fully qualified job name and thread id or taskname. One set of bars is produced per job/thread or task.
Y-Axis: Faults per second for each job/thread.

Example:


Data Viewer - #1 - [P63747AMA/JWAMAZON/Page faults - #1]
File
Edit
View
Window
Help
Page faults: Intervals 1 to 4000
Job name/user/number: thread ID
QRWTSRVR / QUSER / 204487: 00000005
QRWTSRVR / QUSER / 204613: 00000698
QRWTSRVR / QUSER / 204613: 00000697
QRWTSRVR / QUSER / 204487: 00000013
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
1.1
Faults per second
Bars 1 - 4 of 4

Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.

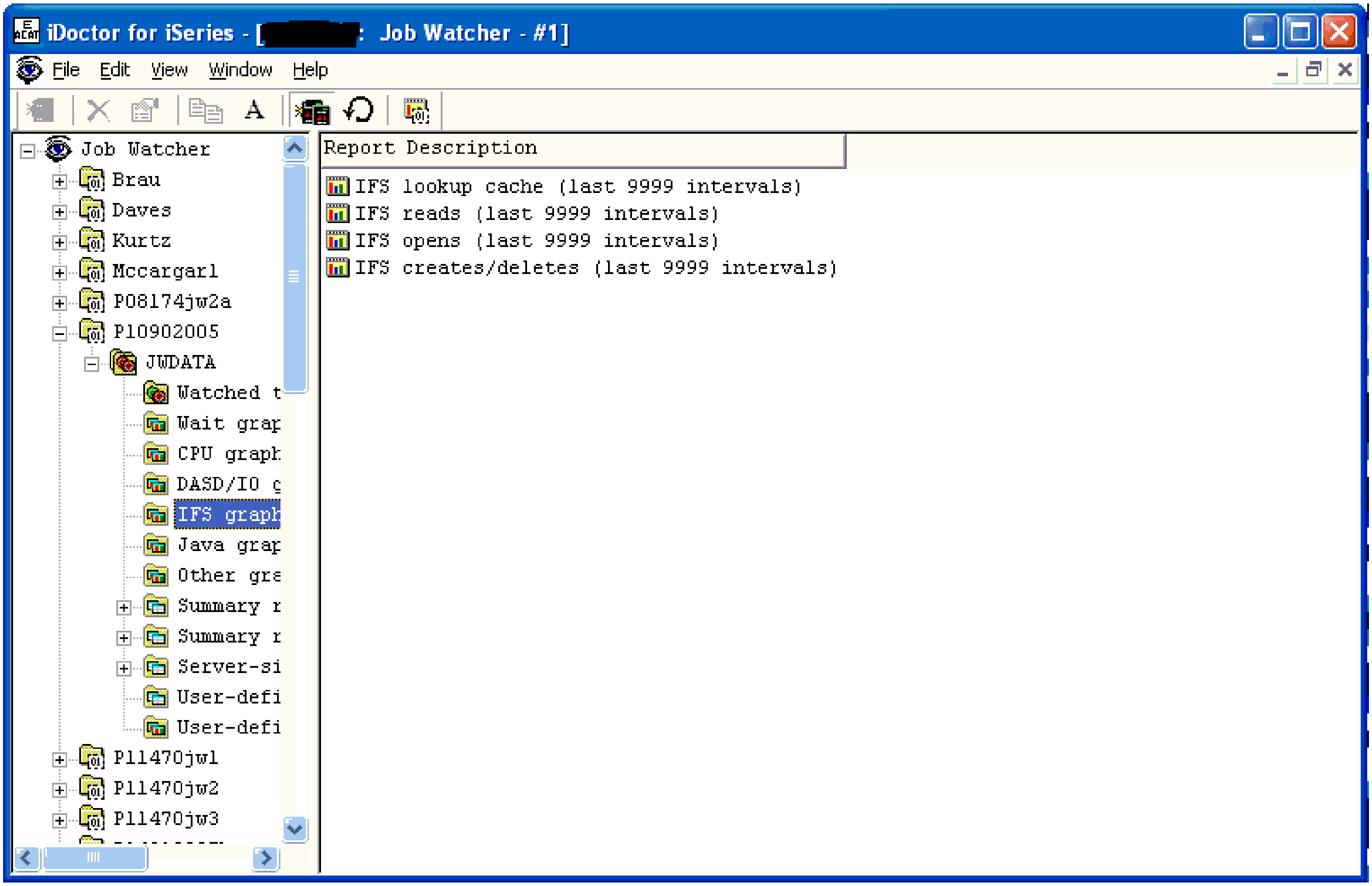
CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.
DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.
Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.
Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.
Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

Table of Contents Previous Next

2.5.5 IFS graphs - summarized

This folder contains a list of graphs which contain summary data over the job-specific job watch relating to IFS statistics. These graphs cover IFS lookup cache, reads, opens and creates/deletes.

An example of the contents of the IFS graphs folder for a job-specific job watch is:



IFS lookup cache (last 9999 intervals)
IFS reads (last 9999 intervals)
IFS opens (last 9999 intervals)
IFS creates/deletes (last 9999 intervals)

The following table illustrates the menu options available by right-clicking on a graph in the list.

Menu	Description
Open graph(s)	Opens the selected graph(s) into a Data Viewer. If a Data Viewer has already been opened submenus will appear underneath this menu in order to give the option of opening the graph(s) into an already open Data Viewer.

Table of Contents Previous Next

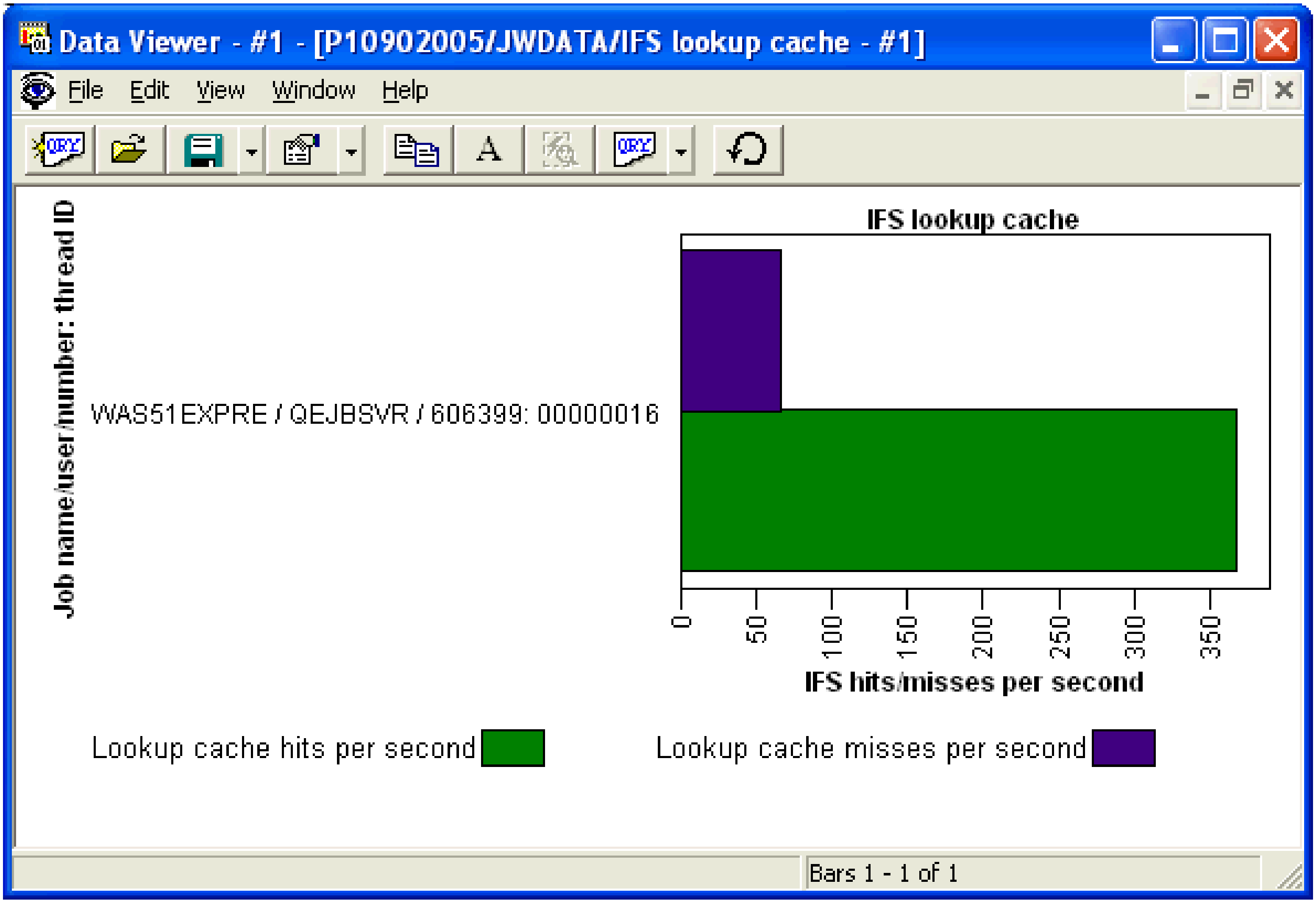
2.5.5.1 IFS lookup cache

Description: This graph shows a **summary** of the rate of IFS lookup cache hits and misses for each job/thread in the job watch over the interval range specified. Each job/thread/task has a green bar showing the hit rate, and a blue bar showing the lookup cache miss rate.

This graph can be useful to quickly see from the entire job watch which jobs/threads had the most lookup cache hits and misses. Once a job of interest is located, the detailed graphs may be used to drill down into specific interval by interval information about a specific job/thread.

Graph Type: summarized job-specific (stacked horizontal bar)
X-axis: Fully qualified job name and thread id or taskname. One set of bars is produced per job/thread or task.
Y-Axis: Rates of IFS lookup cache hits and misses per second.

Example:


Data Viewer - #1 - [P10902005/JWDATA/IFS lookup cache - #1]
File Edit View Window Help
QRY
A
QRY
IFS lookup cache
Job name/user/number: thread ID
WAS51EXPRE / QEJBSVR / 606399: 00000016
0
50
100
150
200
250
300
350
IFS hits/misses per second
Lookup cache hits per second 🟩
Lookup cache misses per second 🟪
Bars 1 - 1 of 1

Right-clicking on a bar in this graph provides the following menu options:

Menu	**Description**
Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.
CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.
DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.
Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.
Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.
Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

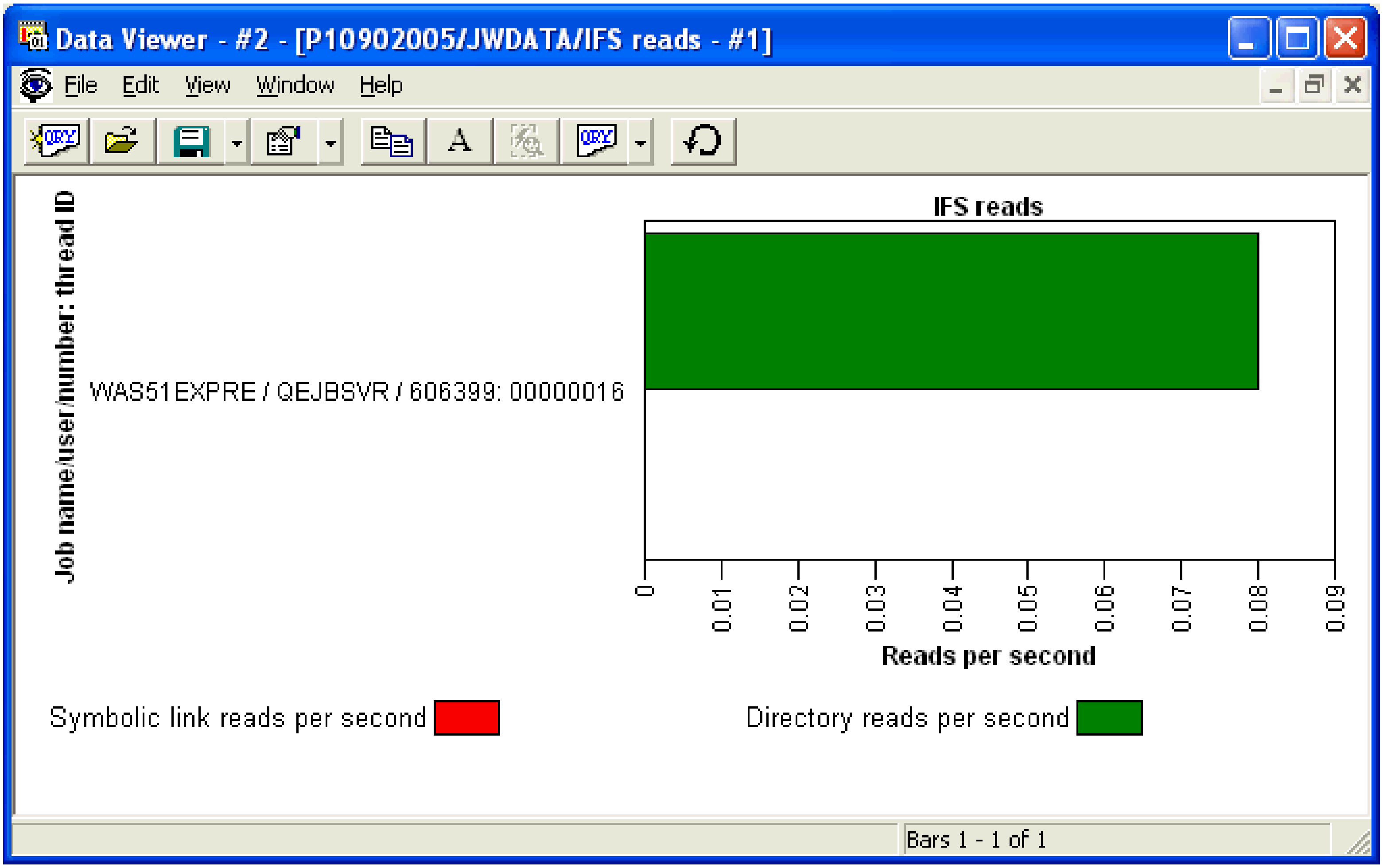
2.5.5.2 IFS reads

Description: This graph shows a **summary** of the rate of IFS symbolic link reads and directory reads for each job/thread in the job watch over the interval range specified. Each job/thread/task has a red bar showing the symbolic link reads rate, and a green bar showing the rate of directory reads. All rates are computed in seconds using the total time the job/thread was active while the job watch was running.

This graph can be useful to quickly see from the entire job watch which jobs/threads had the most lookup cache hits and misses. Once a job of interest is located, the detailed graphs may be used to drill down into specific interval by interval information about a specific job/thread.

Graph Type: summarized job-specific (stacked horizontal bar)
X-axis: Fully qualified job name and thread id or taskname. One set of bars is produced per job/thread or task.
Y-Axis: Symbolic link reads per second and directory reads per second.

Example:


Data Viewer - #2 - [P10902005/JWDATA/IFS reads - #1]
File
Edit
View
Window
Help
IFS reads
Job name/user/number: thread ID
WAS51EXPRE / QEJBSVR / 606399: 00000016
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
Reads per second
Symbolic link reads per second 🟥
Directory reads per second 🟩
Bars 1 - 1 of 1

Right-clicking on a bar in this graph provides the following menu options:

Menu	Description

Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.
CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.
DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.
Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.
Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.
Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

Table of Contents Previous Next

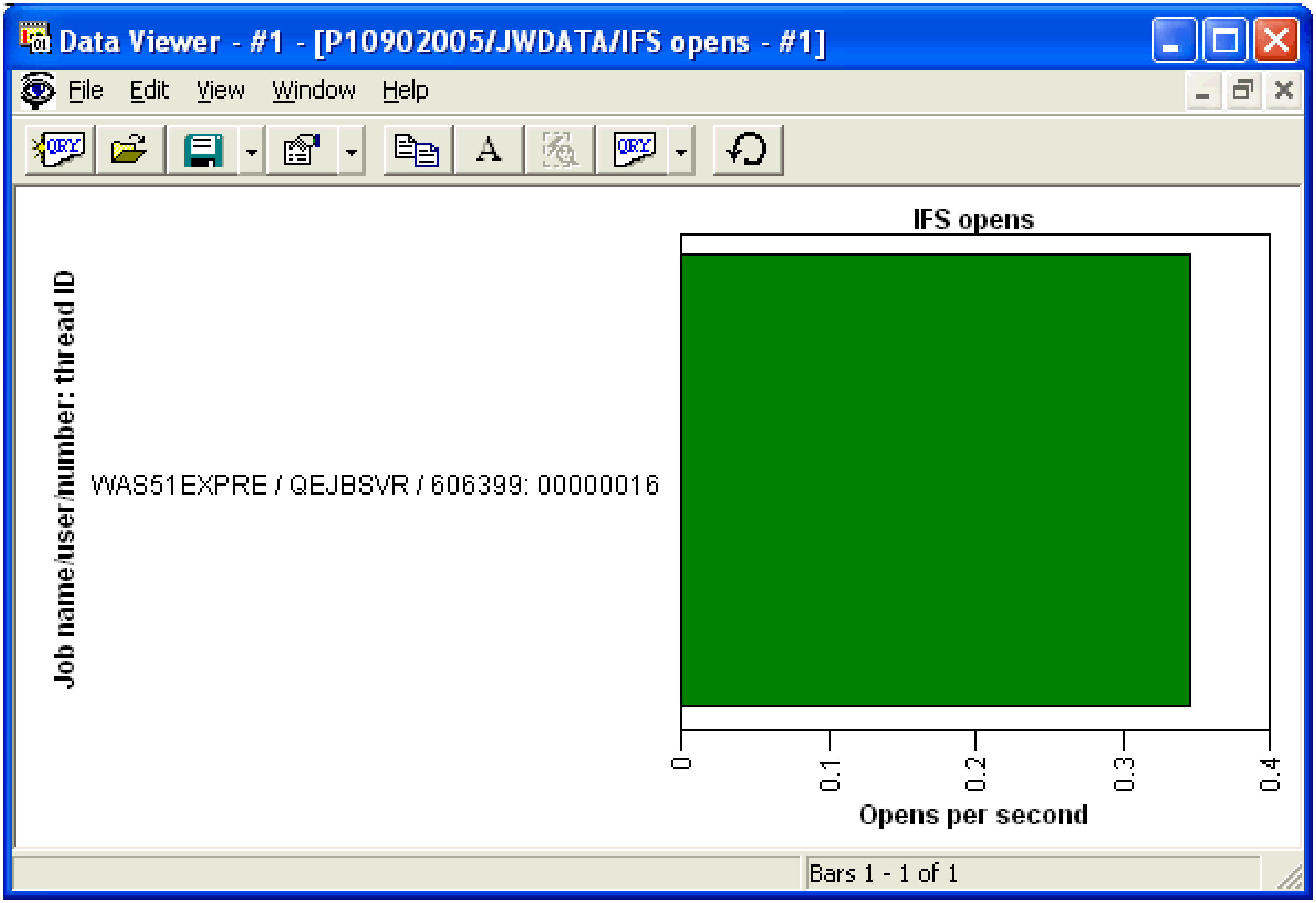
2.5.5.3 IFS opens

Description: This graph shows a **summary** of the rate of IFS opens for each job/thread in the job watch over the interval range specified. Each job/thread/task has a green bar showing the number of IFS file opens per second.

This graph can be useful to quickly see from the entire job watch which jobs/threads had the highest rate of files in the IFS being opened. Once a job of interest is located, the detailed graphs may be used to drill down into specific interval by interval information about a specific job/thread.

Graph Type: summarized job-specific (stacked horizontal bar)
X-axis: Fully qualified job name and thread id or taskname. One set of bars is produced per job/thread or task.
Y-Axis: Rate of opens over files in the IFS per second for the job/thread/task specified on the X-axis.

Example:


Data Viewer - #1 - [P10902005/JWDATA/IFS opens - #1]
File Edit View Window Help
QRY
A
IFS opens
Job name/user/number: thread ID
WAS51EXPRE / QEJBSVR / 606399: 00000016
0
0.1
0.2
0.3
0.4
Opens per second
Bars 1 - 1 of 1

Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.
CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.
DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.
Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.
Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.
Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

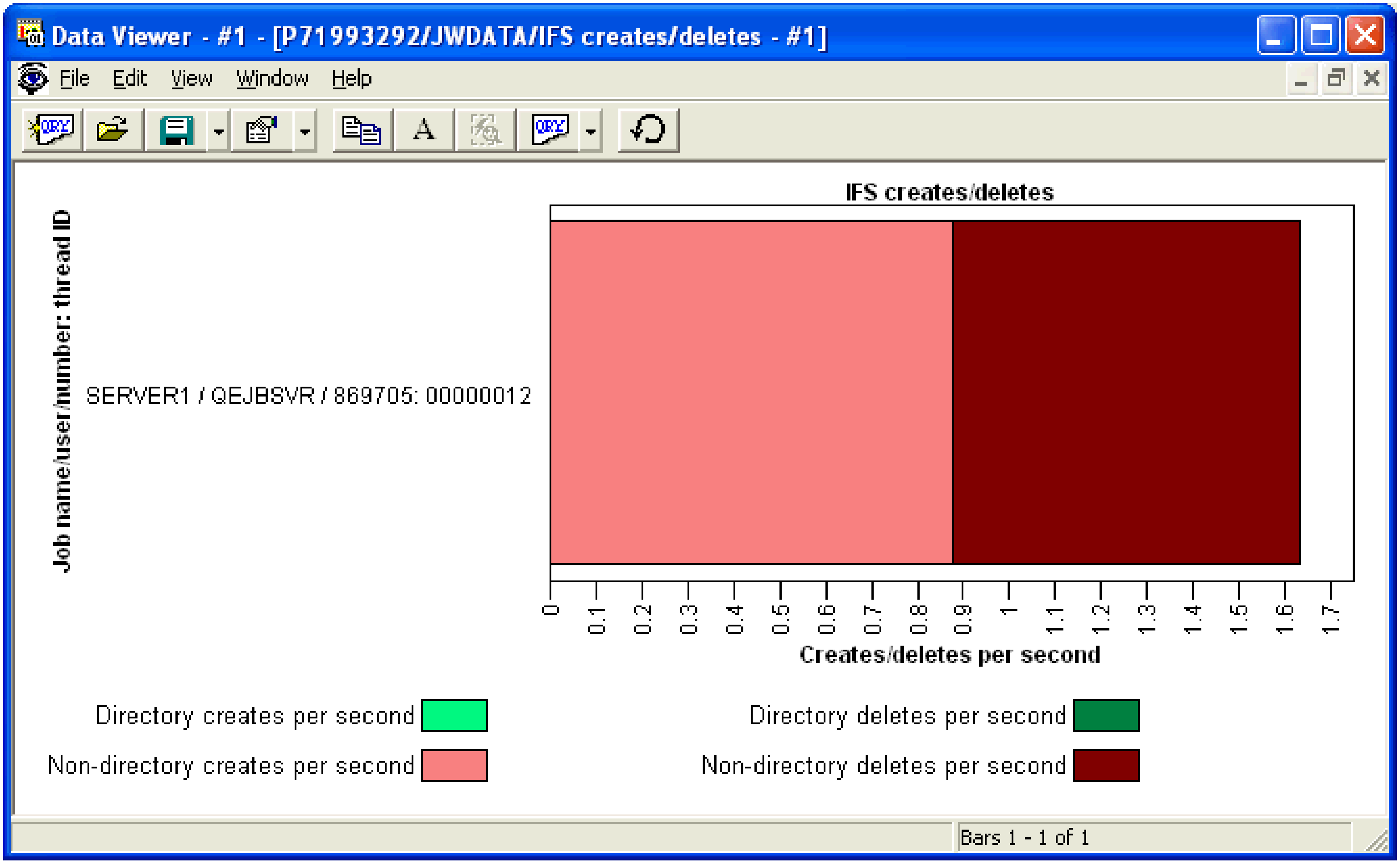
Table of Contents Previous Next

2.5.5.4 IFS creates/deletes

Description: This graph shows a **summary** of the rate of IFS creates and deletes for each job/thread in the job watch over the interval range specified. Each job/thread/task has a set of colors showing the rates of IFS creates and deletes that occurred.

Graph Type: summarized job-specific (stacked horizontal bar)
X-axis: Fully qualified job name and thread id or taskname. One set of bars is produced per job/thread or task.
Y-Axis: Contains rates of directory and non-directory creates and deletes for the IFS. Green colors are used to display directory creates/deletes. Red colors are for non-directory creates and deletes.

Example:


Data Viewer - #1 - [P71993292/JWDATA/IFS creates/deletes - #1]
File Edit View Window Help
IFS creates/deletes
Job name/user/number: thread ID
SERVER1 / QEJBSVR / 869705: 00000012
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1 1.1 1.2 1.3 1.4 1.5 1.6 1.7
Creates/deletes per second
Directory creates per second
Directory deletes per second
Non-directory creates per second
Non-directory deletes per second
Bars 1 - 1 of 1

Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.
CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.
DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.
Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.

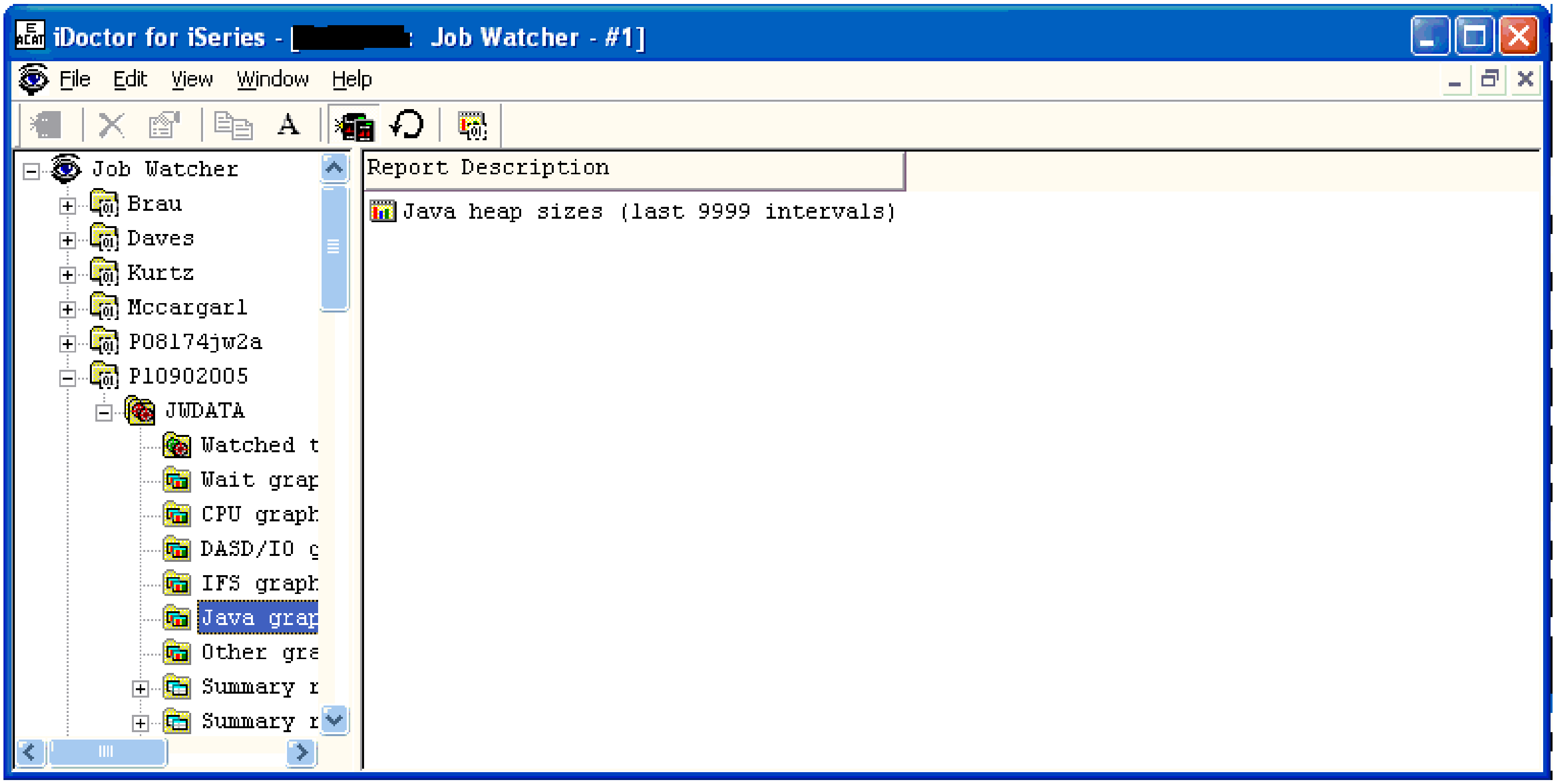
Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.
Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

Table of Contents Previous Next

2.5.6 Java graphs - summarized

This folder contains a list of graphs which contain summary data over the job-specific job watch relating to Java. Currently the only Java related graph available is the Java heap sizes graph.

An example of the contents of the Java graphs folder for a job-specific job watch is:



Report Description
Java heap sizes (last 9999 intervals)

The following table illustrates the menu options available by right-clicking on a graph in the list.

Menu	Description
Open graph(s)	Opens the selected graph(s) into a Data Viewer. If a Data Viewer has already been opened submenus will appear underneath this menu in order to give the option of opening the graph(s) into an already open Data Viewer.

Table of Contents Previous Next

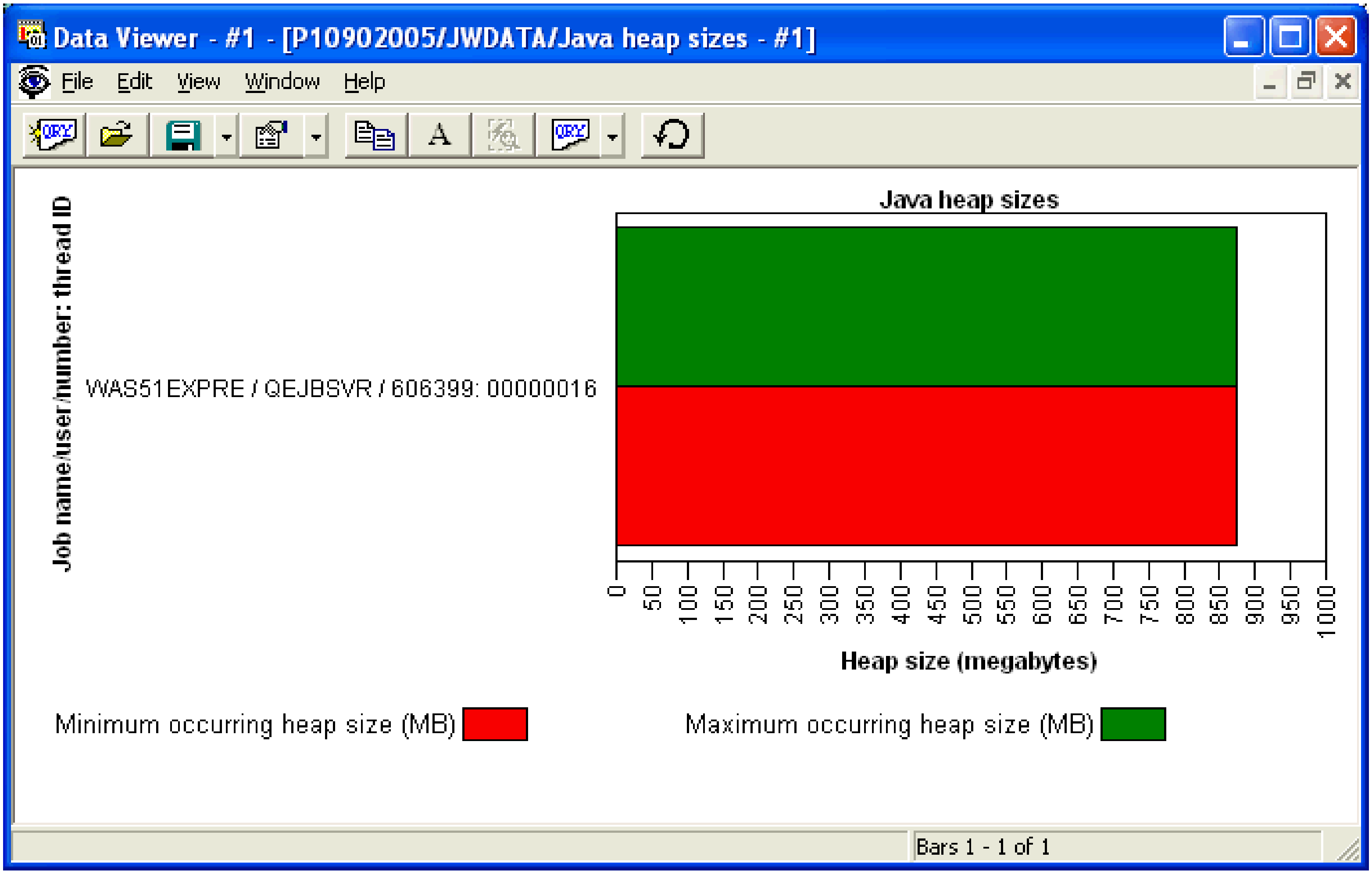
2.5.6.1 Java heap sizes

Description: This graph shows a **summary** of the heap sizes of the JVM for every job in the job watch over the interval range specified. The red bar shows the minimum occuring Java heap size (in megabytes) for each job and the green bar shows the maximum Java heap size (in megabytes).

This graph can be useful to judge the size of an application's heap and using the Java heap graph detail menu (by right-clicking on the bar and choosing the Java graphs option) the user can see how the heap size changed over time.

Graph Type: summarized job-specific (horizontal bar)
X-axis: Fully qualified job name and thread id or taskname. One set of bars is produced per job/thread or task.
Y-Axis: Minimum and maximum size of the JVM (in megabytes).

Example:


Data Viewer - #1 - [P10902005/JWDATA/Java heap sizes - #1]
File Edit View Window Help
Java heap sizes
Job name/user/number: thread ID
WAS51EXPRE / QEJBSVR / 606399: 00000016
0 50 100 150 200 250 300 350 400 450 500 550 600 650 700 750 800 850 900 950 1000
Heap size (megabytes)
Minimum occurring heap size (MB) 🟥
Maximum occurring heap size (MB) 🟩
Bars 1 - 1 of 1

Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.
CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.

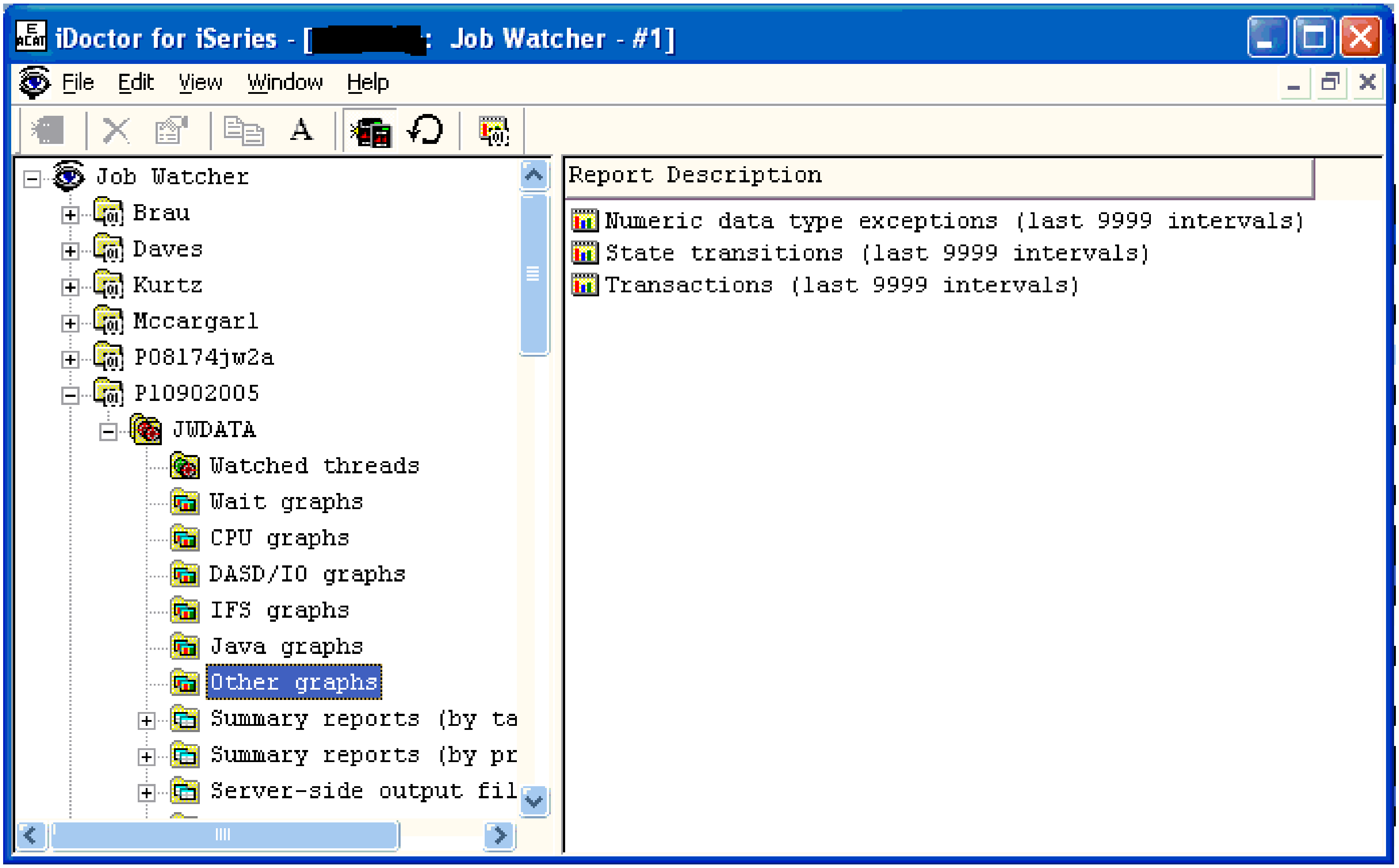
DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.
Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.
Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.
Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

Table of Contents Previous Next

2.5.7 Other graphs - summarized

This folder contains a list of graphs which contain **summary** data over other types of data not covered in the previous graph categories. These graphs include data over transactions, numeric data type exceptions and job state transitions.

An example of the contents of the other graphs folder for a job-specific job watch is:



The following table illustrates the menu options available by right-clicking on a graph in the list.

Menu	Description
Open graph(s)	Opens the selected graph(s) into a Data Viewer. If a Data Viewer has already been opened submenus will appear underneath this menu in order to give the option of opening the graph(s) into an already open Data Viewer.

Table of Contents | Previous | Next

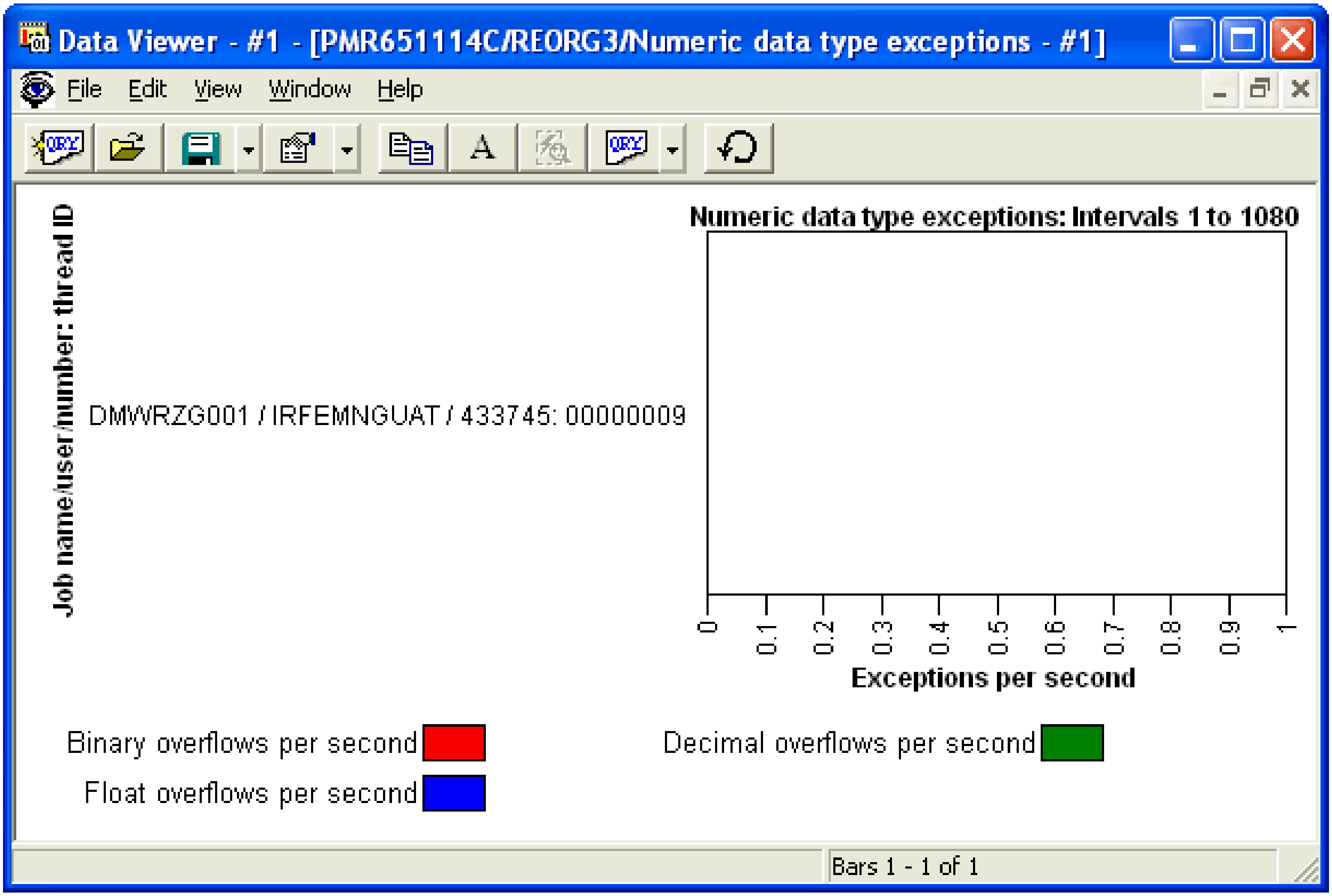
2.5.7.1 Numeric data type exceptions

Description: This graph shows a **summary** of the rates of the numeric exceptions that occurred for every job/thread in the job watch over the interval range specified.

This graph can be useful to quickly see if any overflow exceptions are occurring in an application's code.

Graph Type: summarized job-specific (stacked horizontal bar)
X-axis: Fully qualified job name and thread id or taskname. One bar is produced per job/thread or task.
Y-Axis: Rates of numeric data type exceptions including binary overflows, decimal overflows and float overflows.

Example:


Data Viewer - #1 - [PMR651114C/REORG3/Numeric data type exceptions - #1]
File
Edit
View
Window
Help
QRY
Numeric data type exceptions: Intervals 1 to 1080
Job name/user/number: thread ID
DMWRZG001 / IRFEMNGUAT / 433745: 00000009
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Exceptions per second
Binary overflows per second 🟥
Decimal overflows per second 🟩
Float overflows per second 🟦
Bars 1 - 1 of 1

Right-clicking on a bar in this graph provides the following menu options:

Menu	Description

Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.
CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.
DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.
Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.
Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.
Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

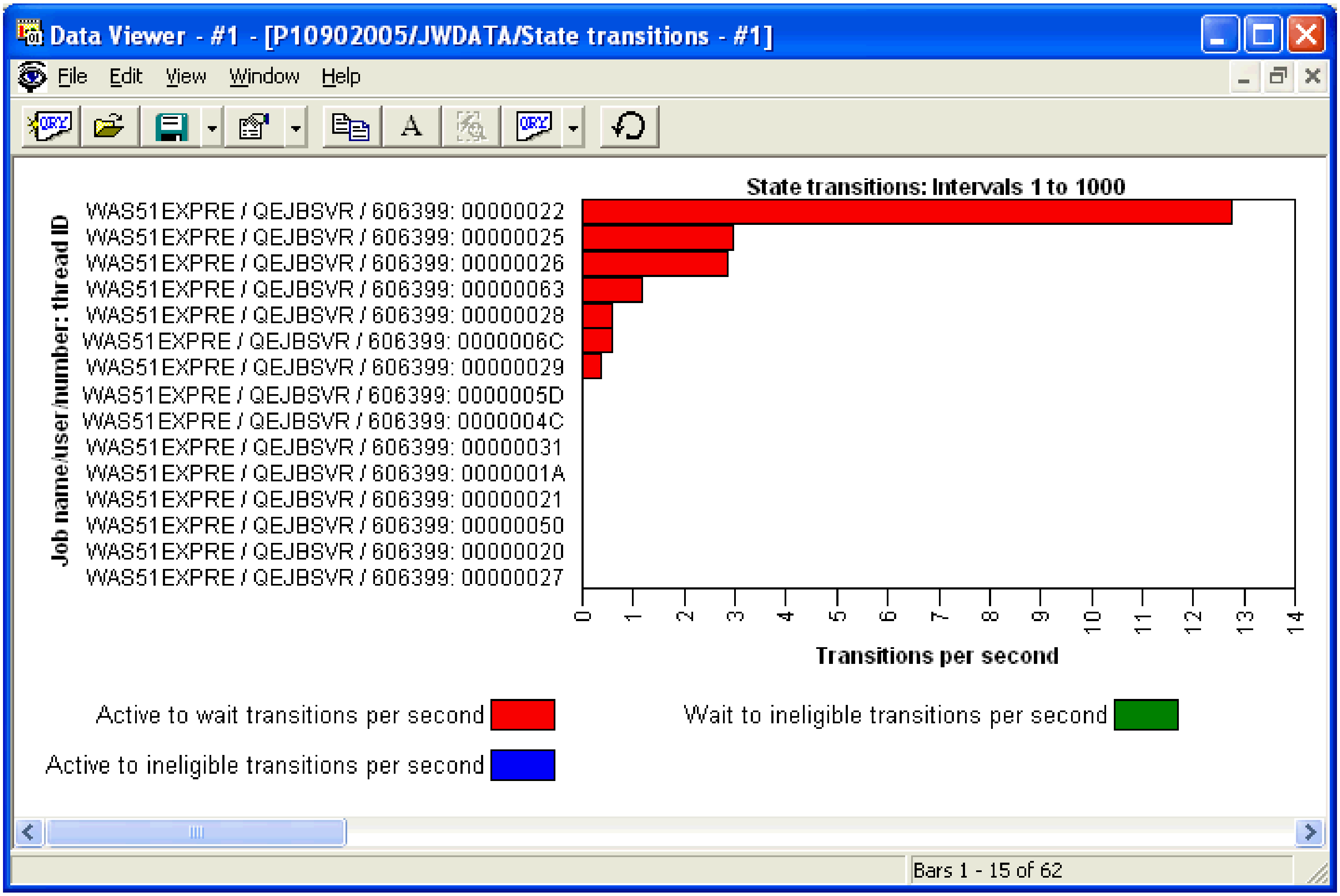
Table of Contents Previous Next

2.5.7.2 State transitions

Description: This graph shows a **summary** of the job state transitions for each job/thread/task in the job watch over the interval range specified.

Graph Type: summarized job-specific (stacked horizontal bar)
X-axis: Fully qualified job name and thread id or taskname. One stacked bar is produced per job/thread or task.
Y-Axis: Active to wait transitions per second in red, wait to ineligible transitions per second in green and active to ineligible transitions per second in blue.

Example:


Data Viewer - #1 - [P10902005/JWDATA/State transitions - #1]
File
Edit
View
Window
Help
State transitions: Intervals 1 to 1000
Job name/user/number: thread ID
WAS51EXPRE / QEJBSVR / 606399: 00000022
WAS51EXPRE / QEJBSVR / 606399: 00000025
WAS51EXPRE / QEJBSVR / 606399: 00000026
WAS51EXPRE / QEJBSVR / 606399: 00000063
WAS51EXPRE / QEJBSVR / 606399: 00000028
WAS51EXPRE / QEJBSVR / 606399: 0000006C
WAS51EXPRE / QEJBSVR / 606399: 00000029
WAS51EXPRE / QEJBSVR / 606399: 0000005D
WAS51EXPRE / QEJBSVR / 606399: 0000004C
WAS51EXPRE / QEJBSVR / 606399: 00000031
WAS51EXPRE / QEJBSVR / 606399: 0000001A
WAS51EXPRE / QEJBSVR / 606399: 00000021
WAS51EXPRE / QEJBSVR / 606399: 00000050
WAS51EXPRE / QEJBSVR / 606399: 00000020
WAS51EXPRE / QEJBSVR / 606399: 00000027
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Transitions per second
Active to wait transitions per second 🟥
Wait to ineligible transitions per second 🟩
Active to ineligible transitions per second 🟦
Bars 1 - 15 of 62

Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.
CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.
DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.

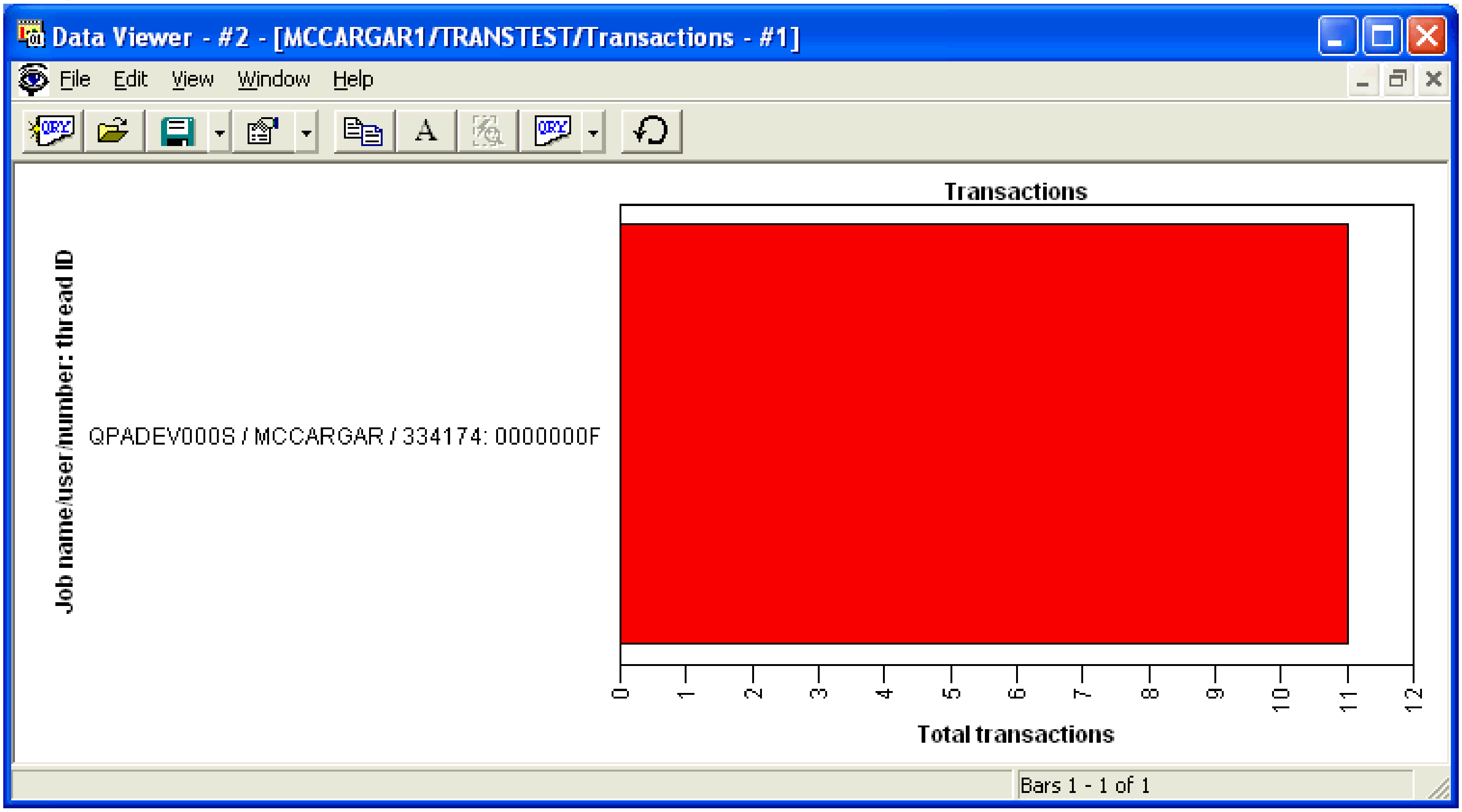
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.
Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.
Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.
Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

2.5.7.3 Transactions

Description: This graph shows a **summary** of the total transactions occuring for each job in the job watch over the interval range specified.

Graph Type: summarized job-specific (horizontal bar)
X-axis: Fully qualified job name and thread id or taskname. One bar is produced per job/thread or task.
Y-Axis: Total transactions occurring per job/task/thread.

Example:


Data Viewer - #2 - [MCCARGAR1/TRANSTEST/Transactions - #1]
File
Edit
View
Window
Help
Transactions
Job name/user/number: thread ID
QPADEV0008 / MCCARGAR / 334174: 0000000F
0
1
2
3
4
5
6
7
8
9
10
11
12
Total transactions
Bars 1 - 1 of 1

Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.
CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.
DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.
Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.
Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.
Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.

Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

Table of Contents | Previous | Next

2.6 Job-specific Job Watch detail reports

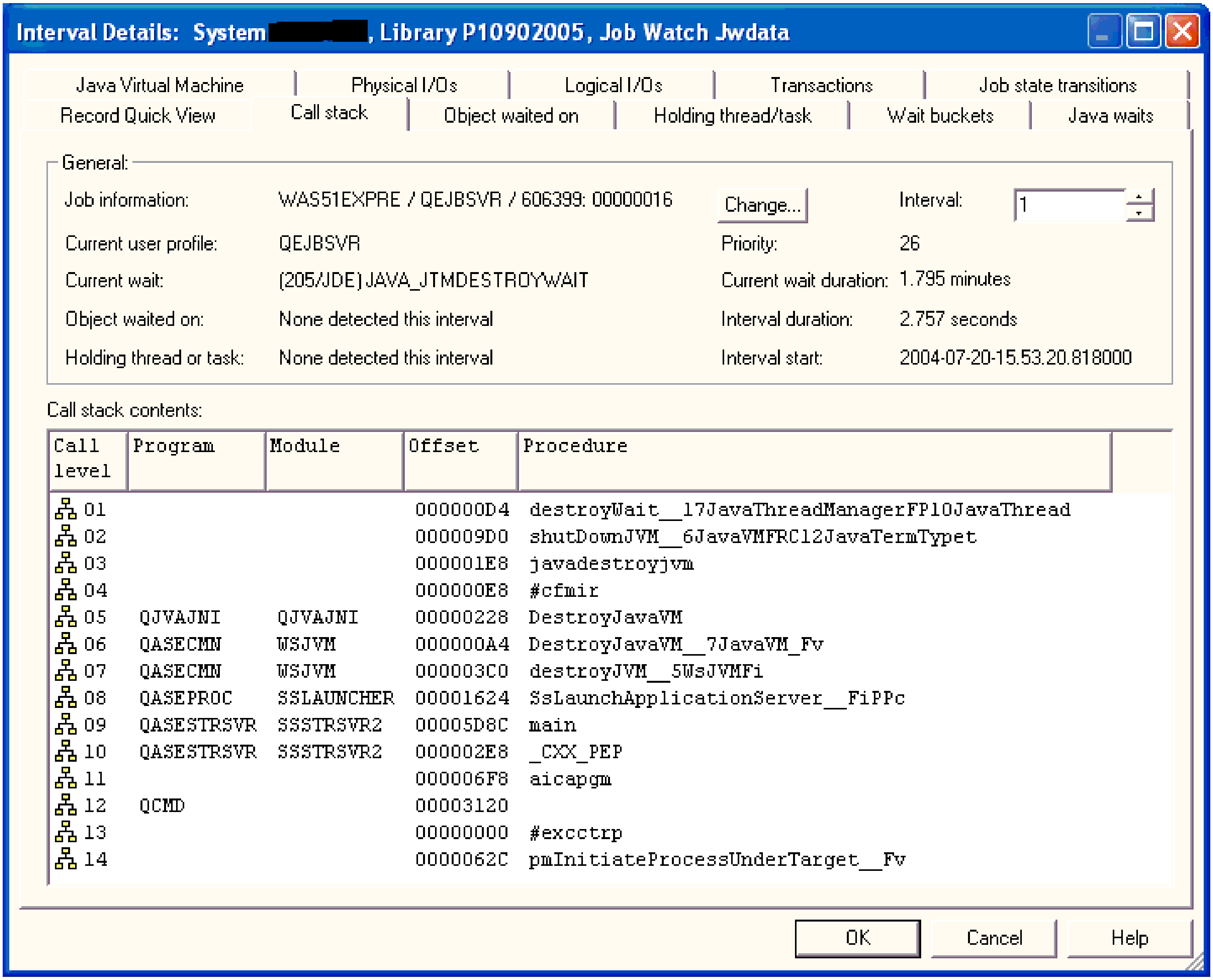
This section covers the detailed property pages, graphs and tables available in a job-specific job watch. These reports are available from by selecting any job/thread/task in a table view or graph and right-clicking on the record or bar representing the job. For example, right-clicking on a bar in a summarized graph provides many different detailed reports showing information about just the specific job selected. In most cases each bar in a detail graph represents some value in the job's data reported every interval.

Table of Contents Previous Next

2.6.1 Interval Properties

The interval properties window displays information about a job in a job-specific job watch for a specific interval. These properties contain several different panels which vary depending on the data available. Some panels are not shown if data has not been collected.

Getting to the interval details panel can be done by clicking on a bar in most job-specific Job Watcher interval detail graphs or by double-clicking on a record in a report/table in a job-specific Job Watcher output file. These output files are listed in the server-side output files folder under a job watch.



Interval Details: System [redacted], Library P10902005, Job Watch Jwdata

General:

Job information: WAS51EXPRE / QEJBSVR / 606399: 00000016

Interval: 1

Current user profile: QEJBSVR

Priority: 26

Current wait: (205/JDE) JAVA_JTMDESTROYWAIT

Current wait duration: 1.795 minutes

Object waited on: None detected this interval

Interval duration: 2.757 seconds

Holding thread or task: None detected this interval

Interval start: 2004-07-20-15.53.20.818000

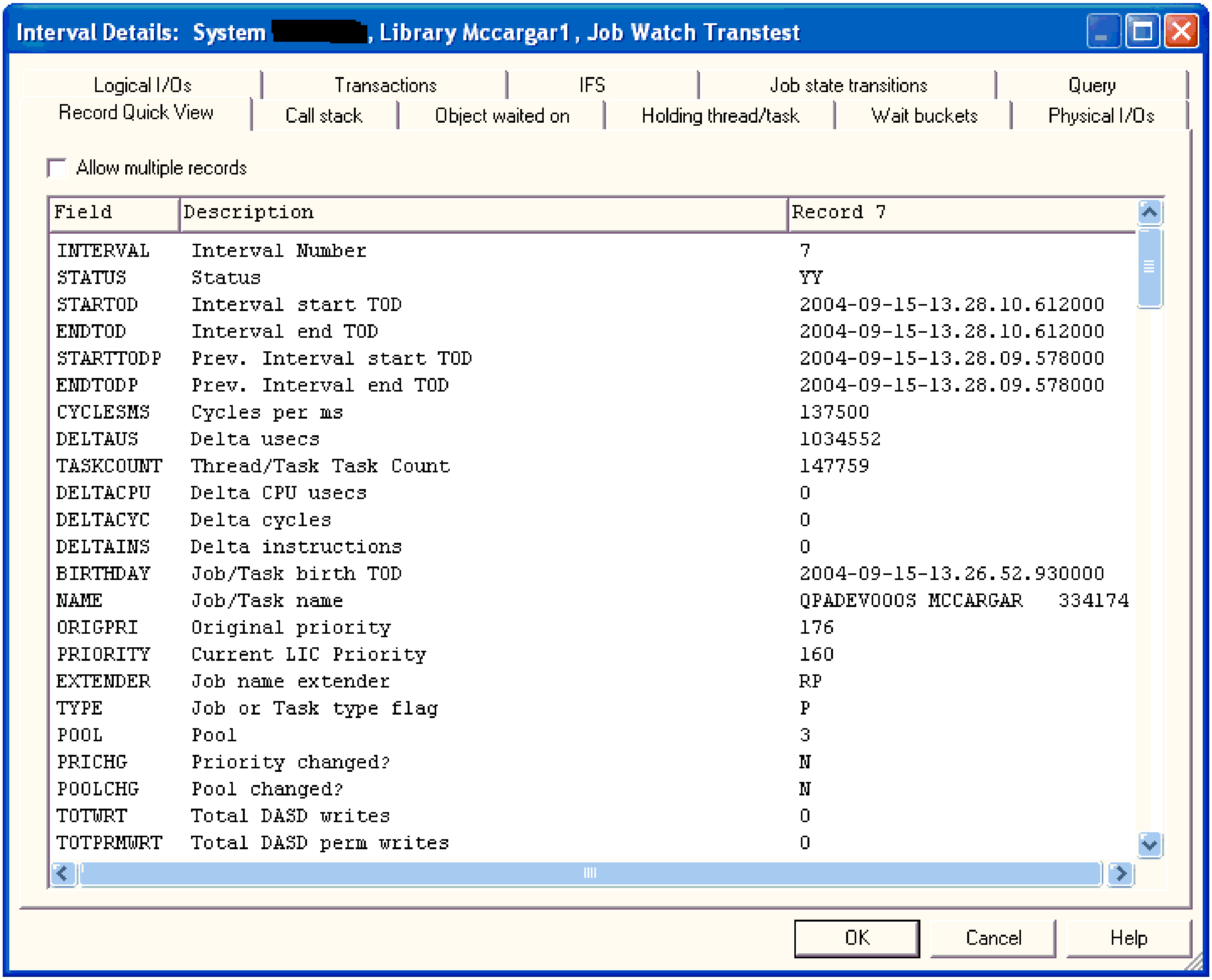
Call stack contents:

Call level	Program	Module	Offset	Procedure
01			000000D4	destroyWait__17JavaThreadManagerFP10JavaThread
02			000009D0	shutDownJVM__6JavaVMFRC12JavaTermTypet
03			000001E8	javadestroyjvm
04			000000E8	#cfmir
05	QJVAJNI	QJVAJNI	00000228	DestroyJavaVM
06	QASECMN	WSJVM	000000A4	DestroyJavaVM__7JavaVM_Fv
07	QASECMN	WSJVM	000003C0	destroyJVM__5WsJVMFi
08	QASEPROC	SSLAUNCHER	00001624	SsLaunchApplicationServer__FiPPc
09	QASESTRSVR	SSSTRSVR2	00005D8C	main
10	QASESTRSVR	SSSTRSVR2	000002E8	_CXX_PEP
11			000006F8	aicapgm
12	QCMD		00003120	
13			00000000	#excctrp
14			0000062C	pmInitiateProcessUnderTarget__Fv

Table of Contents | Previous | Next

2.6.1.1 Record Quick View

This page of the interface is found by double-clicking on one of the Job Watcher output files from a table view. The window displays a record vertically from a table view. The other tabs in the view display additional information about the currently selected job and interval.

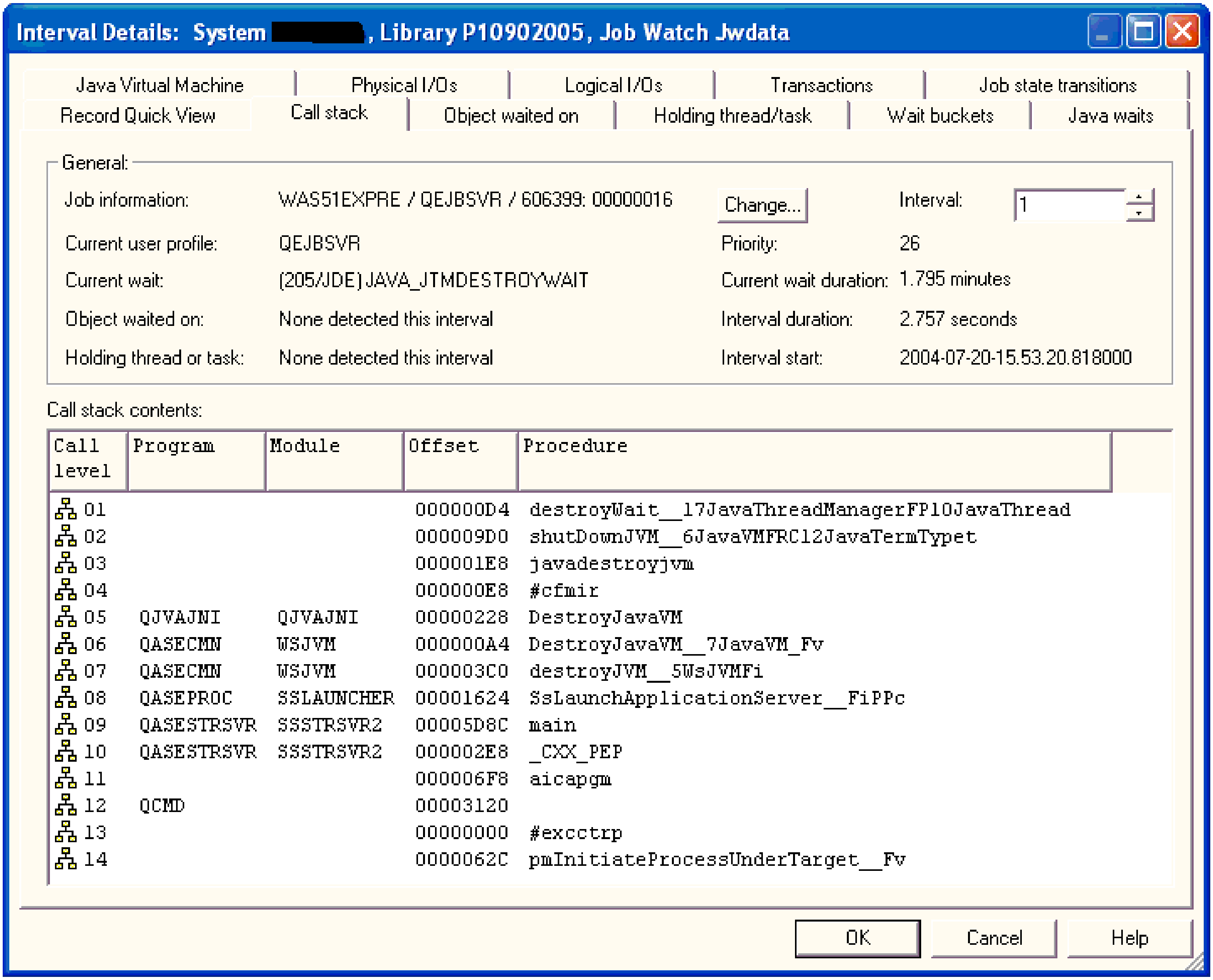


Field	Description	Record 7
INTERVAL	Interval Number	7
STATUS	Status	YY
STARTOD	Interval start TOD	2004-09-15-13.28.10.612000
ENDTOD	Interval end TOD	2004-09-15-13.28.10.612000
STARTTODP	Prev. Interval start TOD	2004-09-15-13.28.09.578000
ENDTODP	Prev. Interval end TOD	2004-09-15-13.28.09.578000
CYCLESMS	Cycles per ms	137500
DELTAUS	Delta usecs	1034552
TASKCOUNT	Thread/Task Task Count	147759
DELTACPU	Delta CPU usecs	0
DELTACYC	Delta cycles	0
DELTAINS	Delta instructions	0
BIRTHDAY	Job/Task birth TOD	2004-09-15-13.26.52.930000
NAME	Job/Task name	QPADEV000S MCCARGAR 334174
ORIGPRI	Original priority	176
PRIORITY	Current LIC Priority	160
EXTENDER	Job name extender	RP
TYPE	Job or Task type flag	P
POOL	Pool	3
PRICHG	Priority changed?	N
POOLCHG	Pool changed?	N
TOTWRT	Total DASD writes	0
TOTPRMWRT	Total DASD perm writes	0

Table of Contents | Previous | Next

2.6.1.2 Call stack

This page shows the call stack (if available) for a job in a job-specific job watch at the moment the snapshot for the interval was taken. The call stack can be up to 50 levels deep, and is shown "bottom up". The call stack includes LIC information below the MI.



Interval Details: System [redacted], Library P10902005, Job Watch Jwdata

General:

Job information:	WAS51EXPRE / QEJBSVR / 606399: 00000016	Interval:	1
Current user profile:	QEJBSVR	Priority:	26
Current wait:	(205/JDE) JAVA_JTMDESTROYWAIT	Current wait duration:	1.795 minutes
Object waited on:	None detected this interval	Interval duration:	2.757 seconds
Holding thread or task:	None detected this interval	Interval start:	2004-07-20-15.53.20.818000

Call stack contents:

Call level	Program	Module	Offset	Procedure
01			000000D4	destroyWait__17JavaThreadManagerFP10JavaThread
02			000009D0	shutDownJVM__6JavaVMFRC12JavaTermTypet
03			000001E8	javadestroyjvm
04			000000E8	#cfmir
05	QJVAJNI	QJVAJNI	00000228	DestroyJavaVM
06	QASECMN	WSJVM	000000A4	DestroyJavaVM__7JavaVM_Fv
07	QASECMN	WSJVM	000003C0	destroyJVM__5WsJVMFi
08	QASEPROC	SSLAUNCHER	00001624	SsLaunchApplicationServer__FiPPc
09	QASESTRSVR	SSSTRSVR2	00005D8C	main
10	QASESTRSVR	SSSTRSVR2	000002E8	_CXX_PEP
11			000006F8	aicapgm
12	QCMD		00003120	
13			00000000	#excctrp
14			0000062C	pmInitiateProcessUnderTarget__Fv

This window contains the following fields:

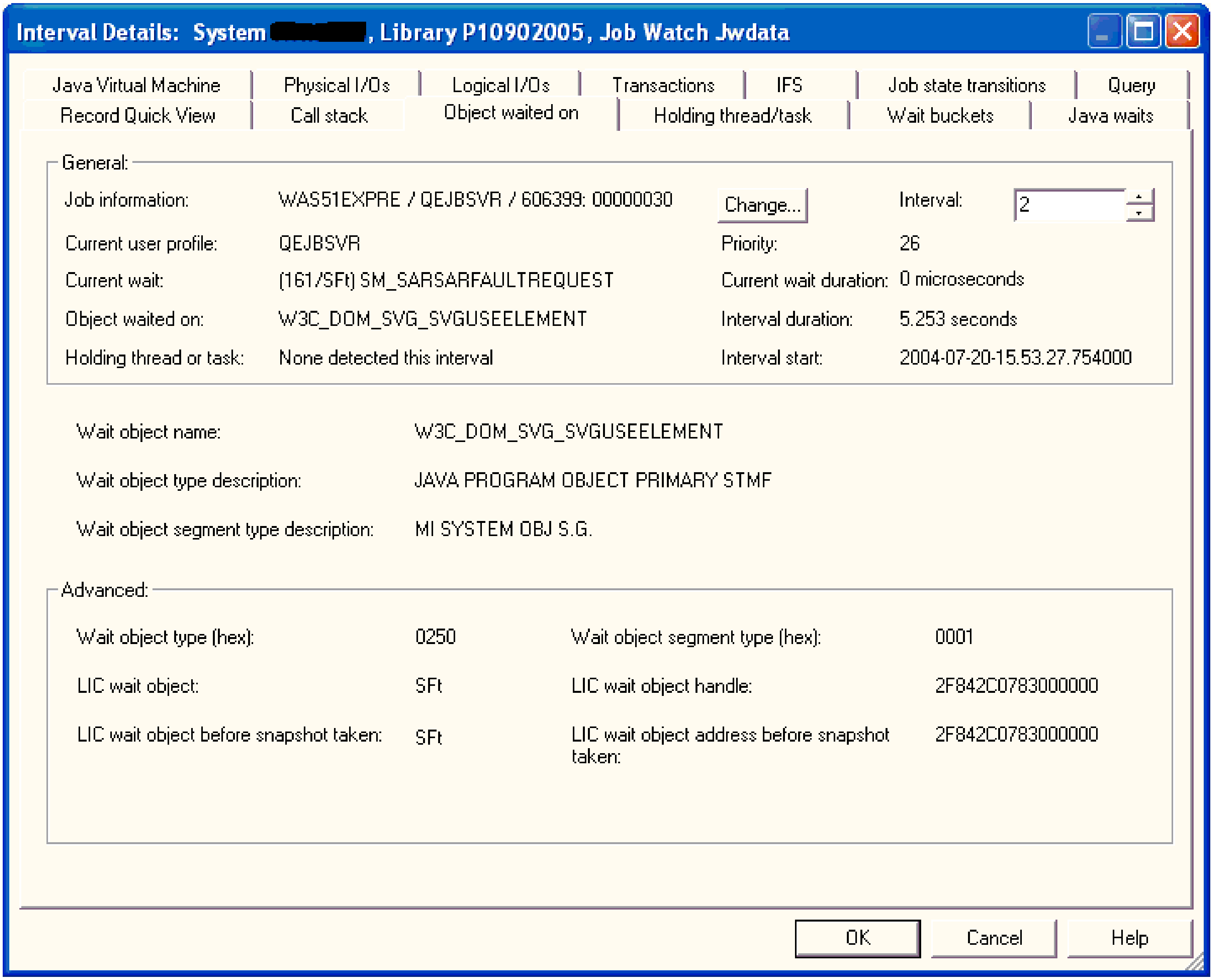
Field name	Description
Job information	The fully qualified job/thread id or task information this interface applies to. The Change... button allows the user to change the current job/thread/task to another. Changing the selected job will update the interface information on this panel but the information on the record quick view panel will still show the data for the original job.

Interval	Displays the selected interval the data shown applies to. The up and down buttons allow the user to change the selected interval. If the interval specified does not exist for the selected job no data will be shown.
Current user profile	The user profile currently active when the job was running at the interval specified.
Priority	The priority of the job for the specified interval.
Current wait	The current wait information contains information about the wait point (numeric identifier and 3 character wait code) and the description of the wait point from file QPYRTJWAT2. This field indicates the type of wait (or CPU) that was occurring when the snapshot to gather information about the job was taken. The wait point shown belongs to a wait bucket. Information about the current bucket is available on the "Wait buckets" panel in this interface.
Current wait duration	The total time the job has been in the current wait type. Everytime the wait type changes or CPU is used, this value is reset at 0. The current wait duration could be days for an idle job doing nothing and this would not indicate any problem. However if the current wait was a seize lock condition and the current wait duration was 30 seconds, this could flag a potential problem. For more information on understanding wait analysis, see the white paper available on the home page of the iDoctor for iSeries <a href="http://www-912.ibm.com/i_dir/idoctor.nsf">Web site</a>.
Object waited on	This field contains information about the object the job/thread/task waited on for the current interval (if any) as well as the type of object being waited on. If an object is specified on this field, additional information about the object is available on the "Object waited on" panel in this interface
Interval duration	The time that elapsed when collecting the selected interval.
Holding thread or task	This field contains information about the job/thread/task that was holding the selected job during the selected interval and preventing it from doing work. If a holding thread or task is specified within this field, additional information about the holder is available on the "Holding thread/task" panel in this interface
Interval start	The date/time the current interval began for the specified job.
Call stack	This contains program, module and procedure names within the current job's call stack for the specified interval. The call stack is gathered at the end of the interval. It is a snapshot of what programs/procedures were running in the job at this particular moment in time. The call stack can contain up to 50 levels of information.

Table of Contents | Previous | Next

2.6.1.3 Object waited on

This page of the interval details shows information about the object the current job was waiting on at the moment the snapshot for the specified interval was taken.



Interval Details: System [redacted], Library P10902005, Job Watch Jwdata

General:

Job information: WAS51EXPRE / QEJBSVR / 606399: 00000030
Interval: 2
Current user profile: QEJBSVR
Priority: 26
Current wait: (161/SFt) SM_SARSARFAULTREQUEST
Current wait duration: 0 microseconds
Object waited on: W3C_DOM_SVG_SVGUSEELEMENT
Interval duration: 5.253 seconds
Holding thread or task: None detected this interval
Interval start: 2004-07-20-15.53.27.754000

Wait object name: W3C_DOM_SVG_SVGUSEELEMENT
Wait object type description: JAVA PROGRAM OBJECT PRIMARY STMF
Wait object segment type description: MI SYSTEM OBJ S.G.

Advanced:

Wait object type (hex): 0250
Wait object segment type (hex): 0001
LIC wait object: SFt
LIC wait object handle: 2F842C0783000000
LIC wait object before snapshot taken: SFt
LIC wait object address before snapshot taken: 2F842C0783000000

This window contains the following fields:

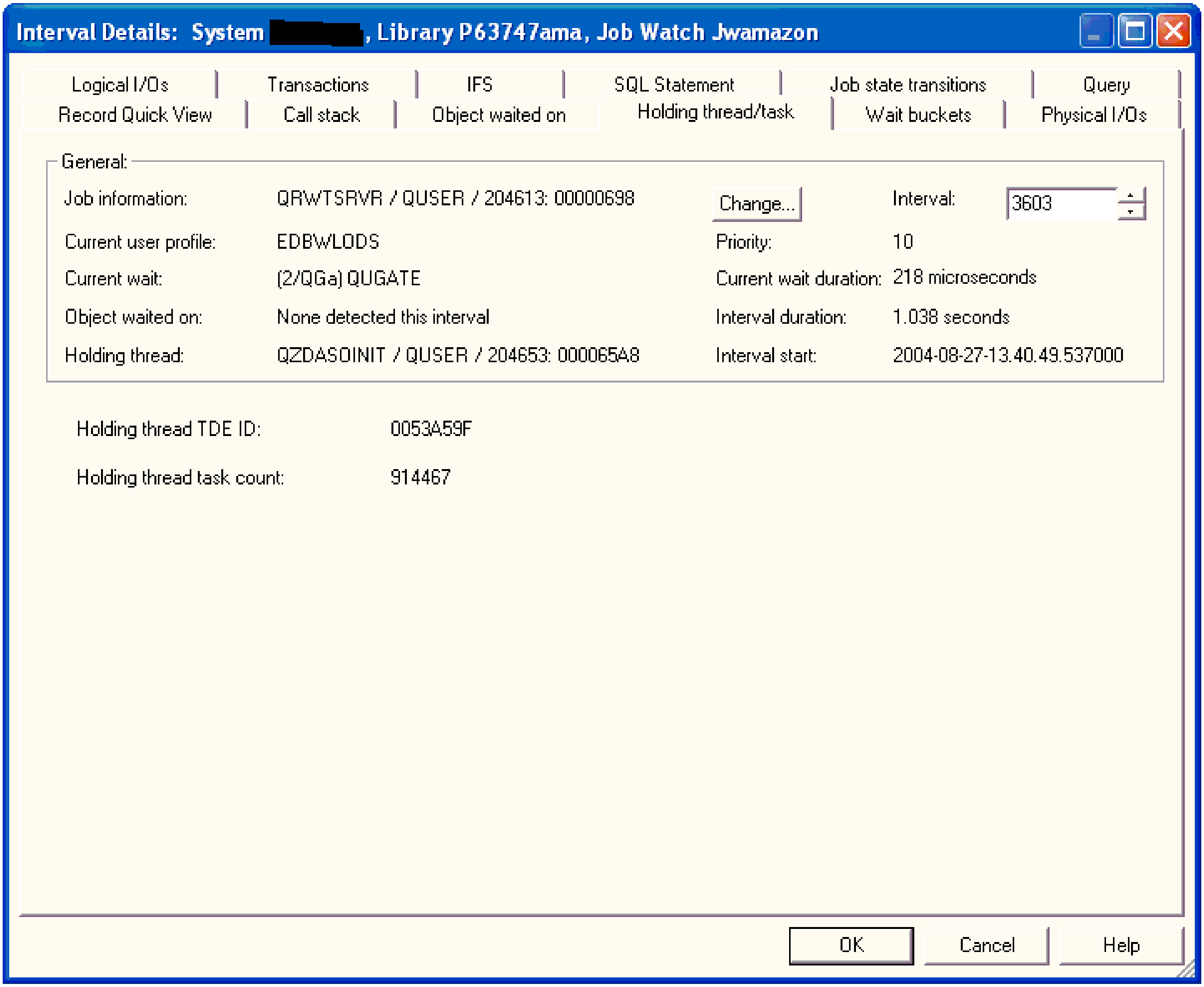
Field name	Description
Job information	The fully qualified job/thread id or task information this interface applies to. The Change... button allows the user to change the current job/thread/task to another. Changing the selected job will update the interface information on this panel but the information on the record quick view panel will still show the data for the original job.

Interval	Displays the selected interval the data shown applies to. The up and down buttons allow the user to change the selected interval. If the interval specified does not exist for the selected job no data will be shown.
Current user profile	The user profile currently active when the job was running at the interval specified.
Priority	The priority of the job for the specified interval.
Current wait	The current wait information contains information about the wait point (numeric identifier and 3 character wait code) and the description of the wait point from file QPYRTJWAT2. This field indicates the type of wait (or CPU) that was occurring when the snapshot to gather information about the job was taken. The wait point shown belongs to a wait bucket. Information about the current bucket is available on the "Wait buckets" panel in this interface.
Current wait duration	The total time the job has been in the current wait type. Everytime the wait type changes or CPU is used, this value is reset at 0. The current wait duration could be days for an idle job doing nothing and this would not indicate any problem. However if the current wait was a seize lock condition and the current wait duration was 30 seconds, this could flag a potential problem. For more information on understanding wait analysis, see the white paper available on the home page of the iDoctor for iSeries <a href="http://www-912.ibm.com/i_dir/idoctor.nsf">Web site</a>.
Object waited on	This field contains information about the object the job/thread/task waited on for the current interval (if any) as well as the type of object being waited on. If an object is specified on this field, additional information about the object is available on the "Object waited on" panel in this interface
Interval duration	The time that elapsed when collecting the selected interval.
Holding thread or task	This field contains information about the job/thread/task that was holding the selected job during the selected interval and preventing it from doing work. If a holding thread or task is specified within this field, additional information about the holder is available on the "Holding thread/task" panel in this interface
Interval start	The date/time the current interval began for the specified job.

Table of Contents Previous Next

2.6.1.4 Holding thread/task

This page shows information about the job holding the current job.



Interval Details: System [redacted], Library P63747ama, Job Watch Jwamazon

Job information: QRWTSRVR / QUSER / 204613: 00000698

Interval: 3603

Current user profile: EDBWLODS

Priority: 10

Current wait: (2/QGa) QUGATE

Current wait duration: 218 microseconds

Object waited on: None detected this interval

Interval duration: 1.038 seconds

Holding thread: QZDASOINIT / QUSER / 204653: 000065A8

Interval start: 2004-08-27-13.40.49.537000

Holding thread TDE ID: 0053A59F

Holding thread task count: 914467

This window contains the following fields:

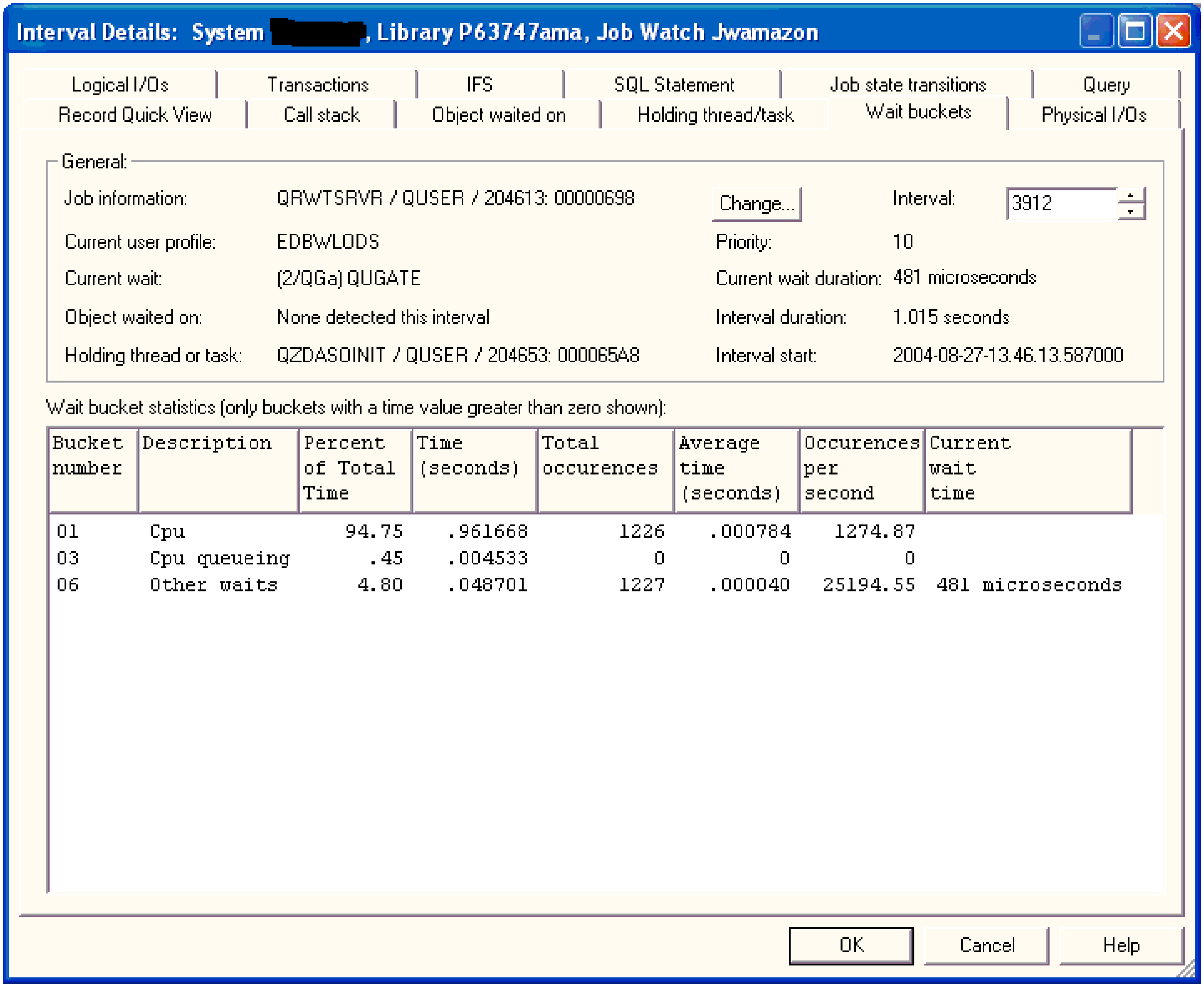
Field name	Description
Job information	The fully qualified job/thread id or task information this interface applies to. The Change... button allows the user to change the current job/thread/task to another. Changing the selected job will update the interface information on this panel but the information on the record quick view panel will still show the data for the original job.

Interval	Displays the selected interval the data shown applies to. The up and down buttons allow the user to change the selected interval. If the interval specified does not exist for the selected job no data will be shown.
Current user profile	The user profile currently active when the job was running at the interval specified.
Priority	The priority of the job for the specified interval.
Current wait	The current wait information contains information about the wait point (numeric identifier and 3 character wait code) and the description of the wait point from file QPYRTJWAT2. This field indicates the type of wait (or CPU) that was occurring when the snapshot to gather information about the job was taken. The wait point shown belongs to a wait bucket. Information about the current bucket is available on the "Wait buckets" panel in this interface.
Current wait duration	The total time the job has been in the current wait type. Everytime the wait type changes or CPU is used, this value is reset at 0. The current wait duration could be days for an idle job doing nothing and this would not indicate any problem. However if the current wait was a seize lock condition and the current wait duration was 30 seconds, this could flag a potential problem. For more information on understanding wait analysis, see the white paper available on the home page of the iDoctor for iSeries <a href="http://www-912.ibm.com/i_dir/idoctor.nsf">Web site</a>.
Object waited on	This field contains information about the object the job/thread/task waited on for the current interval (if any) as well as the type of object being waited on. If an object is specified on this field, additional information about the object is available on the "Object waited on" panel in this interface
Interval duration	The time that elapsed when collecting the selected interval.
Holding thread or task	This field contains information about the job/thread/task that was holding the selected job during the selected interval and preventing it from doing work. If a holding thread or task is specified within this field, additional information about the holder is available on the "Holding thread/task" panel in this interface
Interval start	The date/time the current interval began for the specified job.
Holding thread TDE ID	TDE (task dispatching element) ID of the holding thread or task. This field is displayed as an 8 character hex string.
Holding thread task count	A number uniquely identifing the holding thread.

Table of Contents | Previous | Next

2.6.1.5 Wait buckets

This page shows the waits that occurred during the specified interval for the job indicated. The wait buckets are shown along with the percentage of time spent in each bucket. The final column in the list shows which bucket the current wait (listed in the 3rd line of the general section) belongs to. In the example below the wait point QUGATE (a type of normal wait state when a job is waiting in line to get its turn for work) is shown to belong to bucket 6 (other waits).



Interval Details: System [redacted], Library P63747ama, Job Watch Jwamazon

General:

Job information:	QRWTSRVR / QUSER / 204613: 00000698	Change... Interval:	3912
Current user profile:	EDBWLODS	Priority:	10
Current wait:	(2/QGa) QUGATE	Current wait duration:	481 microseconds
Object waited on:	None detected this interval	Interval duration:	1.015 seconds
Holding thread or task:	QZDASOINIT / QUSER / 204653: 000065A8	Interval start:	2004-08-27-13.46.13.587000

Wait bucket statistics (only buckets with a time value greater than zero shown):

Bucket number	Description	Percent of Total Time	Time (seconds)	Total occurences	Average time (seconds)	Occurences per second	Current wait time
01	Cpu	94.75	.961668	1226	.000784	1274.87	
03	Cpu queueing	.45	.004533	0	0	0	
06	Other waits	4.80	.048701	1227	.000040	25194.55	481 microseconds

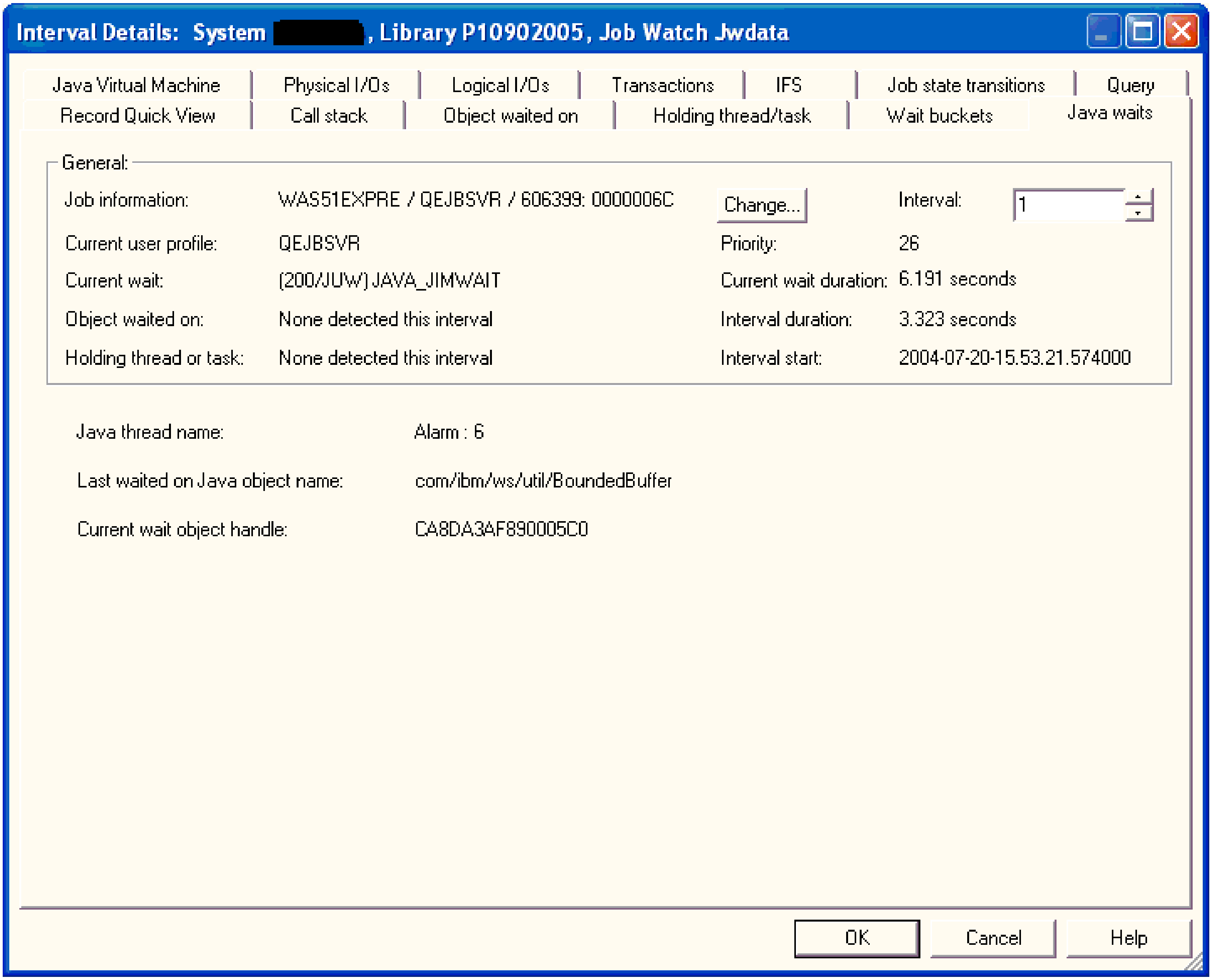
This window contains the following fields:

Field name	Description

Job information	The fully qualified job/thread id or task information this interface applies to. The Change... button allows the user to change the current job/thread/task to another. Changing the selected job will update the interface information on this panel but the information on the record quick view panel will still show the data for the original job.
Interval	Displays the selected interval the data shown applies to. The up and down buttons allow the user to change the selected interval. If the interval specified does not exist for the selected job no data will be shown.
Current user profile	The user profile currently active when the job was running at the interval specified.
Priority	The priority of the job for the specified interval.
Current wait	The current wait information contains information about the wait point (numeric identifier and 3 character wait code) and the description of the wait point from file QPYRTJWAT2. This field indicates the type of wait (or CPU) that was occurring when the snapshot to gather information about the job was taken. The wait point shown belongs to a wait bucket. Information about the current bucket is available on the "Wait buckets" panel in this interface.
Current wait duration	The total time the job has been in the current wait type. Everytime the wait type changes or CPU is used, this value is reset at 0. The current wait duration could be days for an idle job doing nothing and this would not indicate any problem. However if the current wait was a seize lock condition and the current wait duration was 30 seconds, this could flag a potential problem. For more information on understanding wait analysis, see the white paper available on the home page of the iDoctor for iSeries <a href="http://www-912.ibm.com/i_dir/idoctor.nsf">Web site</a>.
Object waited on	This field contains information about the object the job/thread/task waited on for the current interval (if any) as well as the type of object being waited on. If an object is specified on this field, additional information about the object is available on the "Object waited on" panel in this interface
Interval duration	The time that elapsed when collecting the selected interval.
Holding thread or task	This field contains information about the job/thread/task that was holding the selected job during the selected interval and preventing it from doing work. If a holding thread or task is specified within this field, additional information about the holder is available on the "Holding thread/task" panel in this interface
Interval start	The date/time the current interval began for the specified job.
Wait bucket statistics	Contains the time spent in each type of wait as well as the number of occurrences of each wait bucket during the specified interval. The occurrence count goes up everytime a transition is made from one type of wait to another.

2.6.1.6 Java waits

This page of the interval details window provides statistics information about the java thread within the Java application that was in a wait during the specified interval. The effected object name where the wait is occuring in the application as well as the object handle within the JVM is provided on this window.



Interval Details: System [redacted], Library P10902005, Job Watch Jwdata

General:

Job information: WAS51EXPRE / QEJBSVR / 606399: 0000006C

Interval: 1

Current user profile: QEJBSVR

Priority: 26

Current wait: (200/JUW) JAVA_JIMWAIT

Current wait duration: 6.191 seconds

Object waited on: None detected this interval

Interval duration: 3.323 seconds

Holding thread or task: None detected this interval

Interval start: 2004-07-20-15.53.21.574000

Java thread name: Alarm : 6

Last waited on Java object name: com/ibm/ws/util/BoundedBuffer

Current wait object handle: CA8DA3AF890005C0

This window contains the following fields:

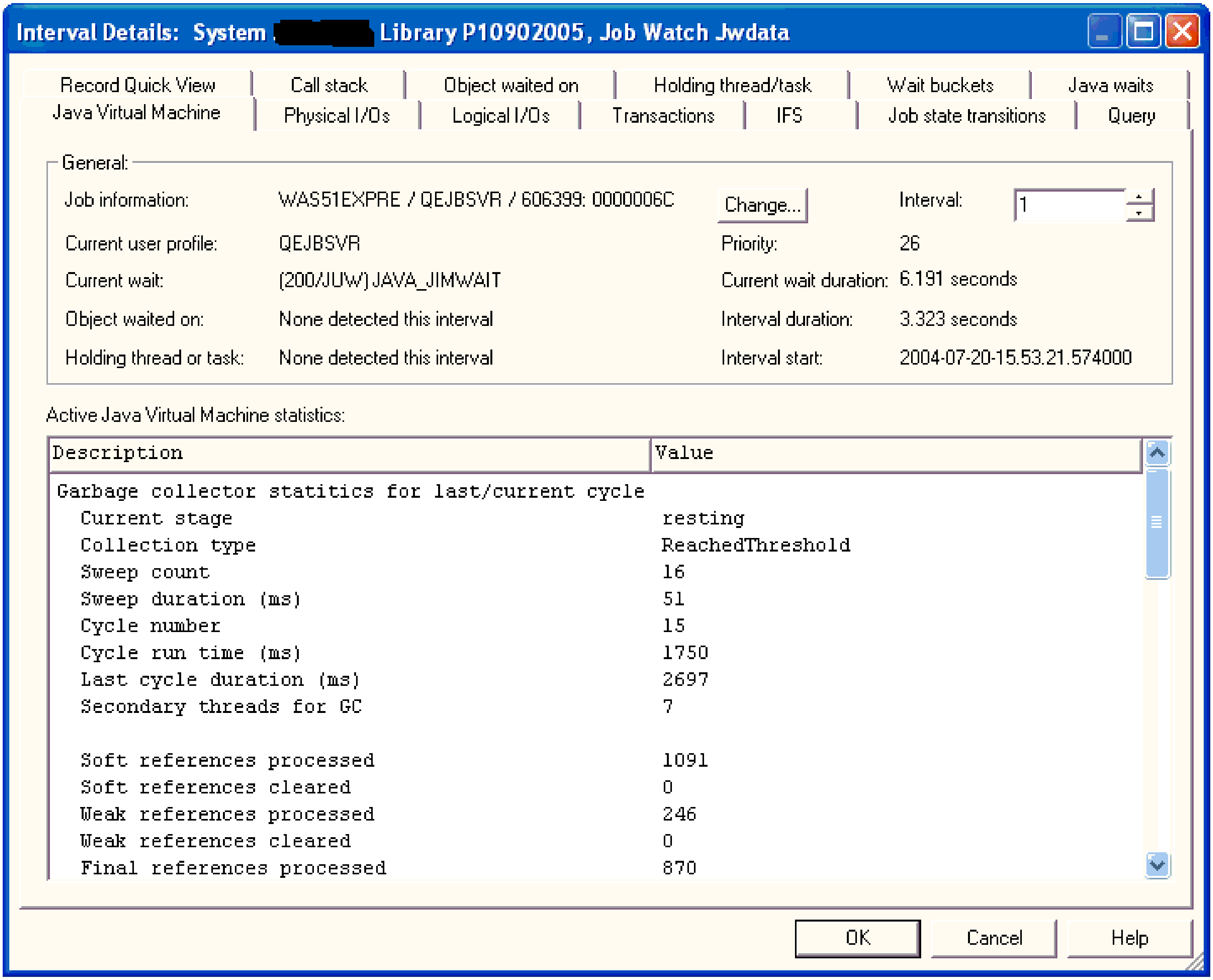
Field name	Description
Job information	The fully qualified job/thread id or task information this interface applies to. The Change... button allows the user to change the current job/thread/task to another. Changing the selected job will update the interface information on this panel but the information on the record quick view panel will still show the data for the original job.

Interval	Displays the selected interval the data shown applies to. The up and down buttons allow the user to change the selected interval. If the interval specified does not exist for the selected job no data will be shown.
Current user profile	The user profile currently active when the job was running at the interval specified.
Priority	The priority of the job for the specified interval.
Current wait	The current wait information contains information about the wait point (numeric identifier and 3 character wait code) and the description of the wait point from file QPYRTJWAT2. This field indicates the type of wait (or CPU) that was occurring when the snapshot to gather information about the job was taken. The wait point shown belongs to a wait bucket. Information about the current bucket is available on the "Wait buckets" panel in this interface.
Current wait duration	The total time the job has been in the current wait type. Everytime the wait type changes or CPU is used, this value is reset at 0. The current wait duration could be days for an idle job doing nothing and this would not indicate any problem. However if the current wait was a seize lock condition and the current wait duration was 30 seconds, this could flag a potential problem. For more information on understanding wait analysis, see the white paper available on the home page of the iDoctor for iSeries <a href="http://www-912.ibm.com/i_dir/idoctor.nsf">Web site</a>.
Object waited on	This field contains information about the object the job/thread/task waited on for the current interval (if any) as well as the type of object being waited on. If an object is specified on this field, additional information about the object is available on the "Object waited on" panel in this interface
Interval duration	The time that elapsed when collecting the selected interval.
Holding thread or task	This field contains information about the job/thread/task that was holding the selected job during the selected interval and preventing it from doing work. If a holding thread or task is specified within this field, additional information about the holder is available on the "Holding thread/task" panel in this interface
Interval start	The date/time the current interval began for the specified job.

Table of Contents | Previous | Next

2.6.1.7 Java Virtual Machine

This page of the interval details window provides statistics for the Java Virtual Machine within the Java application running in the specified job when the interval's snapshot was taken.



Interval Details: System [redacted] Library P10902005, Job Watch Jwdata

General:

Job information:	WAS51EXPRE / QEJBSVR / 606399: 0000006C	Interval:	1
Current user profile:	QEJBSVR	Priority:	26
Current wait:	(200/JUW) JAVA_JIMWAIT	Current wait duration:	6.191 seconds
Object waited on:	None detected this interval	Interval duration:	3.323 seconds
Holding thread or task:	None detected this interval	Interval start:	2004-07-20-15.53.21.574000

Active Java Virtual Machine statistics:

```
Description                                                Value
Garbage collector statitics for last/current cycle
  Current stage                                             resting
  Collection type                                           ReachedThreshold
  Sweep count                                               16
  Sweep duration (ms)                                       51
  Cycle number                                              15
  Cycle run time (ms)                                       1750
  Last cycle duration (ms)                                  2697
  Secondary threads for GC                                  7

  Soft references processed                                 1091
  Soft references cleared                                   0
  Weak references processed                                 246
  Weak references cleared                                   0
  Final references processed                                870
```

This window contains the following fields:

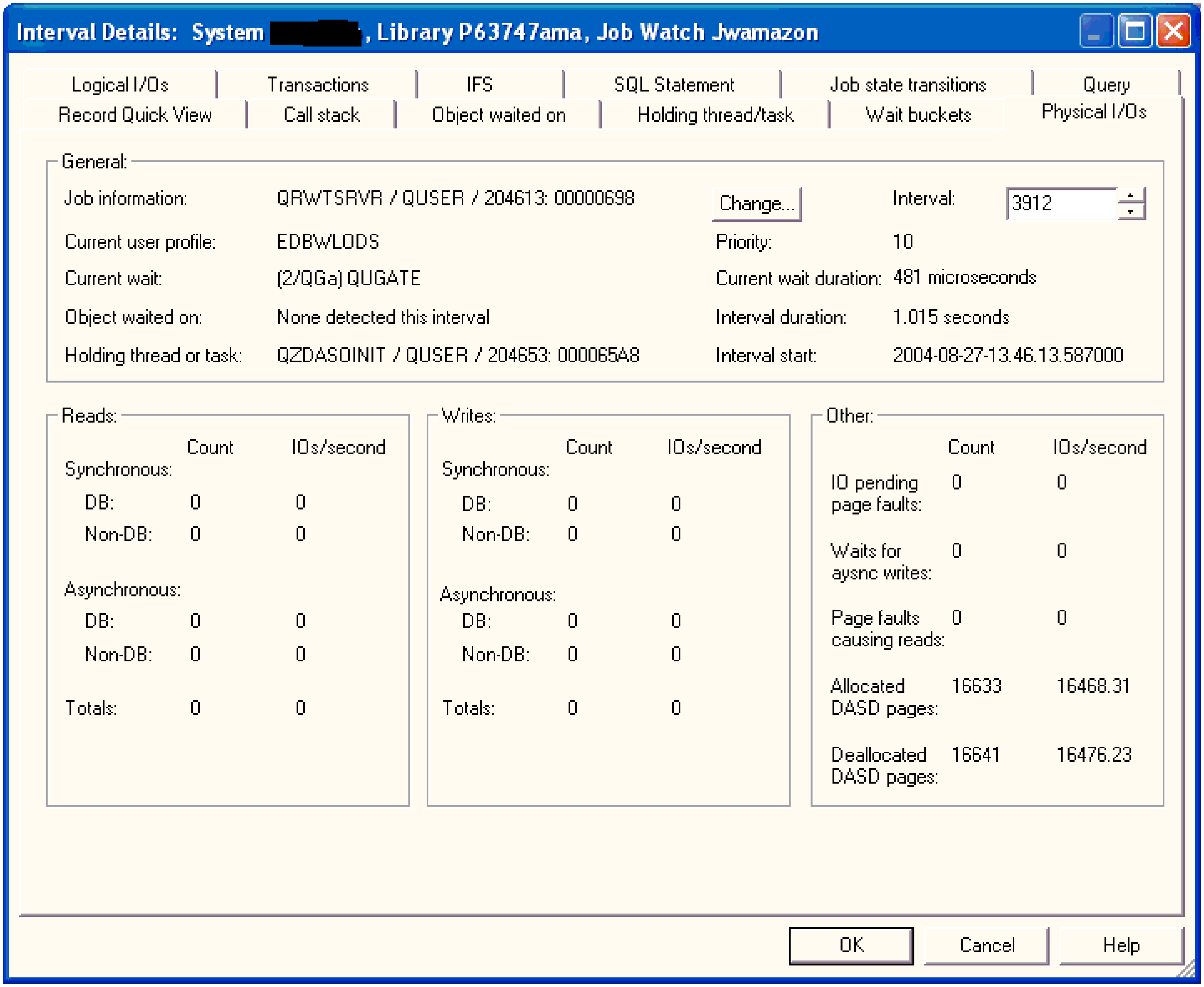
Field name	Description
Job information	The fully qualified job/thread id or task information this interface applies to. The Change... button allows the user to change the current job/thread/task to another. Changing the selected job will update the interface information on this panel but the information on the record quick view panel will still show the data for the original job.

Interval	Displays the selected interval the data shown applies to. The up and down buttons allow the user to change the selected interval. If the interval specified does not exist for the selected job no data will be shown.
Current user profile	The user profile currently active when the job was running at the interval specified.
Priority	The priority of the job for the specified interval.
Current wait	The current wait information contains information about the wait point (numeric identifier and 3 character wait code) and the description of the wait point from file QPYRTJWAT2. This field indicates the type of wait (or CPU) that was occurring when the snapshot to gather information about the job was taken. The wait point shown belongs to a wait bucket. Information about the current bucket is available on the "Wait buckets" panel in this interface.
Current wait duration	The total time the job has been in the current wait type. Everytime the wait type changes or CPU is used, this value is reset at 0. The current wait duration could be days for an idle job doing nothing and this would not indicate any problem. However if the current wait was a seize lock condition and the current wait duration was 30 seconds, this could flag a potential problem. For more information on understanding wait analysis, see the white paper available on the home page of the iDoctor for iSeries <a href="http://www-912.ibm.com/i_dir/idoctor.nsf">Web site</a>.
Object waited on	This field contains information about the object the job/thread/task waited on for the current interval (if any) as well as the type of object being waited on. If an object is specified on this field, additional information about the object is available on the "Object waited on" panel in this interface
Interval duration	The time that elapsed when collecting the selected interval.
Holding thread or task	This field contains information about the job/thread/task that was holding the selected job during the selected interval and preventing it from doing work. If a holding thread or task is specified within this field, additional information about the holder is available on the "Holding thread/task" panel in this interface
Interval start	The date/time the current interval began for the specified job.

Table of Contents Previous Next

2.6.1.8 Physical I/Os

This window contains many details about the various types of physical I/Os that occurred for the specified job and interval. Statistics for reads and writes as well as page faults and disk space allocations/deallocations are provided on this panel.



Interval Details: System [redacted], Library P63747ama, Job Watch Jwamazon

General:

Job information: QRWTSRVR / QUSER / 204613: 00000698 Change... Interval: 3912
Current user profile: EDBWLODS Priority: 10
Current wait: (2/QGa) QUGATE Current wait duration: 481 microseconds
Object waited on: None detected this interval Interval duration: 1.015 seconds
Holding thread or task: QZDASOINIT / QUSER / 204653: 000065A8 Interval start: 2004-08-27-13.46.13.587000

Reads:

	Count	IOs/second
Synchronous:		
DB:	0	0
Non-DB:	0	0
Asynchronous:		
DB:	0	0
Non-DB:	0	0
Totals:	0	0

Writes:

	Count	IOs/second
Synchronous:		
DB:	0	0
Non-DB:	0	0
Asynchronous:		
DB:	0	0
Non-DB:	0	0
Totals:	0	0

Other:

	Count	IOs/second
IO pending page faults:	0	0
Waits for aysnc writes:	0	0
Page faults causing reads:	0	0
Allocated DASD pages:	16633	16468.31
Deallocated DASD pages:	16641	16476.23

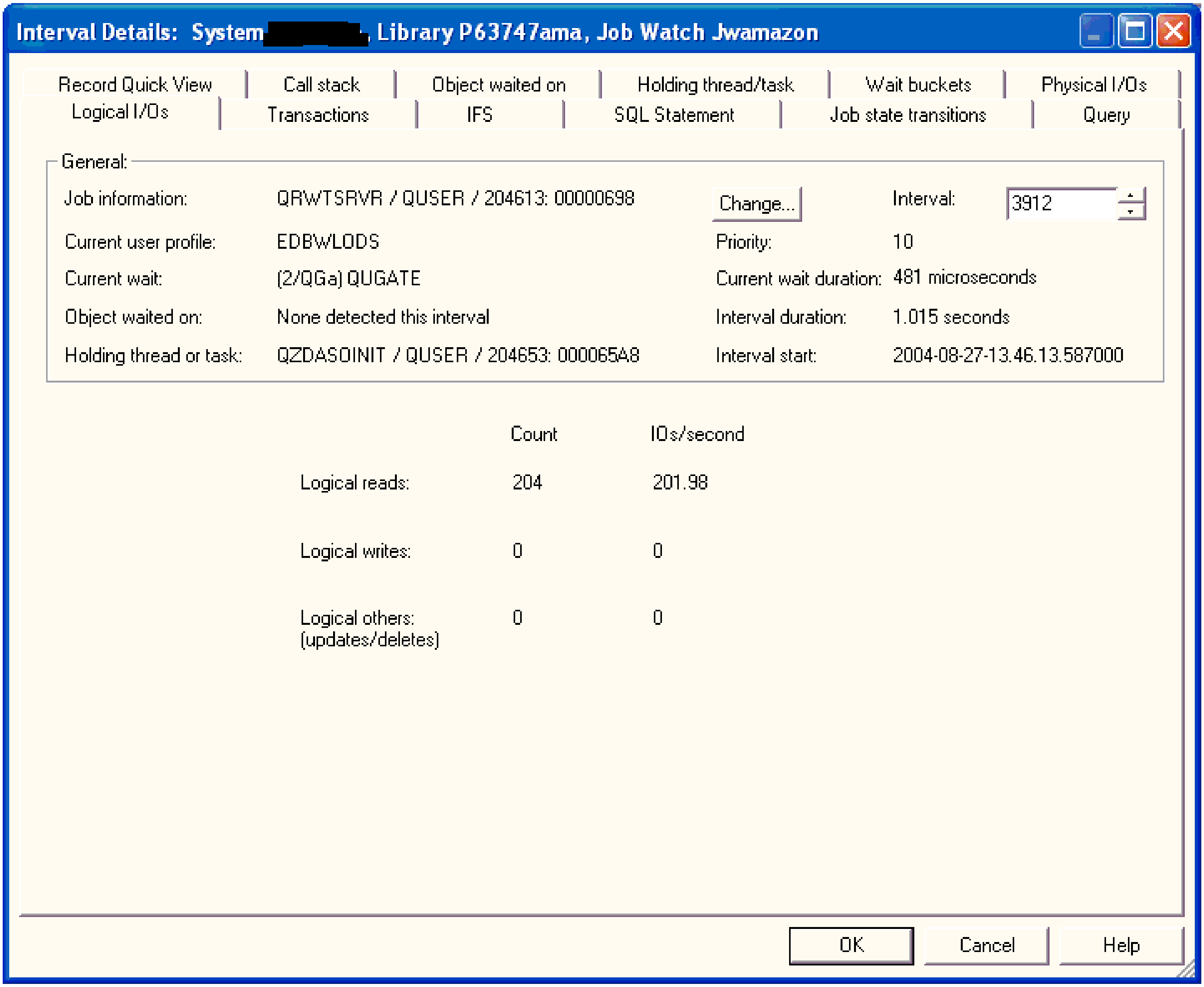
This window contains the following fields:

Field name	Description
Job information	The fully qualified job/thread id or task information this interface applies to. The Change... button allows the user to change the current job/thread/task to another. Changing the selected job will update the interface information on this panel but the information on the record quick view panel will still show the data for the original job.

Interval	Displays the selected interval the data shown applies to. The up and down buttons allow the user to change the selected interval. If the interval specified does not exist for the selected job no data will be shown.
Current user profile	The user profile currently active when the job was running at the interval specified.
Priority	The priority of the job for the specified interval.
Current wait	The current wait information contains information about the wait point (numeric identifier and 3 character wait code) and the description of the wait point from file QPYRTJWAT2. This field indicates the type of wait (or CPU) that was occurring when the snapshot to gather information about the job was taken. The wait point shown belongs to a wait bucket. Information about the current bucket is available on the "Wait buckets" panel in this interface.
Current wait duration	The total time the job has been in the current wait type. Everytime the wait type changes or CPU is used, this value is reset at 0. The current wait duration could be days for an idle job doing nothing and this would not indicate any problem. However if the current wait was a seize lock condition and the current wait duration was 30 seconds, this could flag a potential problem. For more information on understanding wait analysis, see the white paper available on the home page of the iDoctor for iSeries <a href="http://www-912.ibm.com/i_dir/idoctor.nsf">Web site</a>.
Object waited on	This field contains information about the object the job/thread/task waited on for the current interval (if any) as well as the type of object being waited on. If an object is specified on this field, additional information about the object is available on the "Object waited on" panel in this interface
Interval duration	The time that elapsed when collecting the selected interval.
Holding thread or task	This field contains information about the job/thread/task that was holding the selected job during the selected interval and preventing it from doing work. If a holding thread or task is specified within this field, additional information about the holder is available on the "Holding thread/task" panel in this interface
Interval start	The date/time the current interval began for the specified job.

2.6.1.9 Logical I/Os

This page contains statistics for logical database I/Os occurred for the specified job and interval. The LDIO statistics includes rates and totals of reads, writes and updates/deletes.


Interval Details: System [redacted], Library P63747ama, Job Watch Jwamazon

General:

Job information: QRWTSRVR / QUSER / 204613: 00000698

Interval: 3912

Current user profile: EDBWLODS

Priority: 10

Current wait: (2/QGa) QUGATE

Current wait duration: 481 microseconds

Object waited on: None detected this interval

Interval duration: 1.015 seconds

Holding thread or task: QZDASOINIT / QUSER / 204653: 000065A8

Interval start: 2004-08-27-13.46.13.587000

	Count	IOs/second
Logical reads:	204	201.98
Logical writes:	0	0
Logical others: (updates/deletes)	0	0

This window contains the following fields:

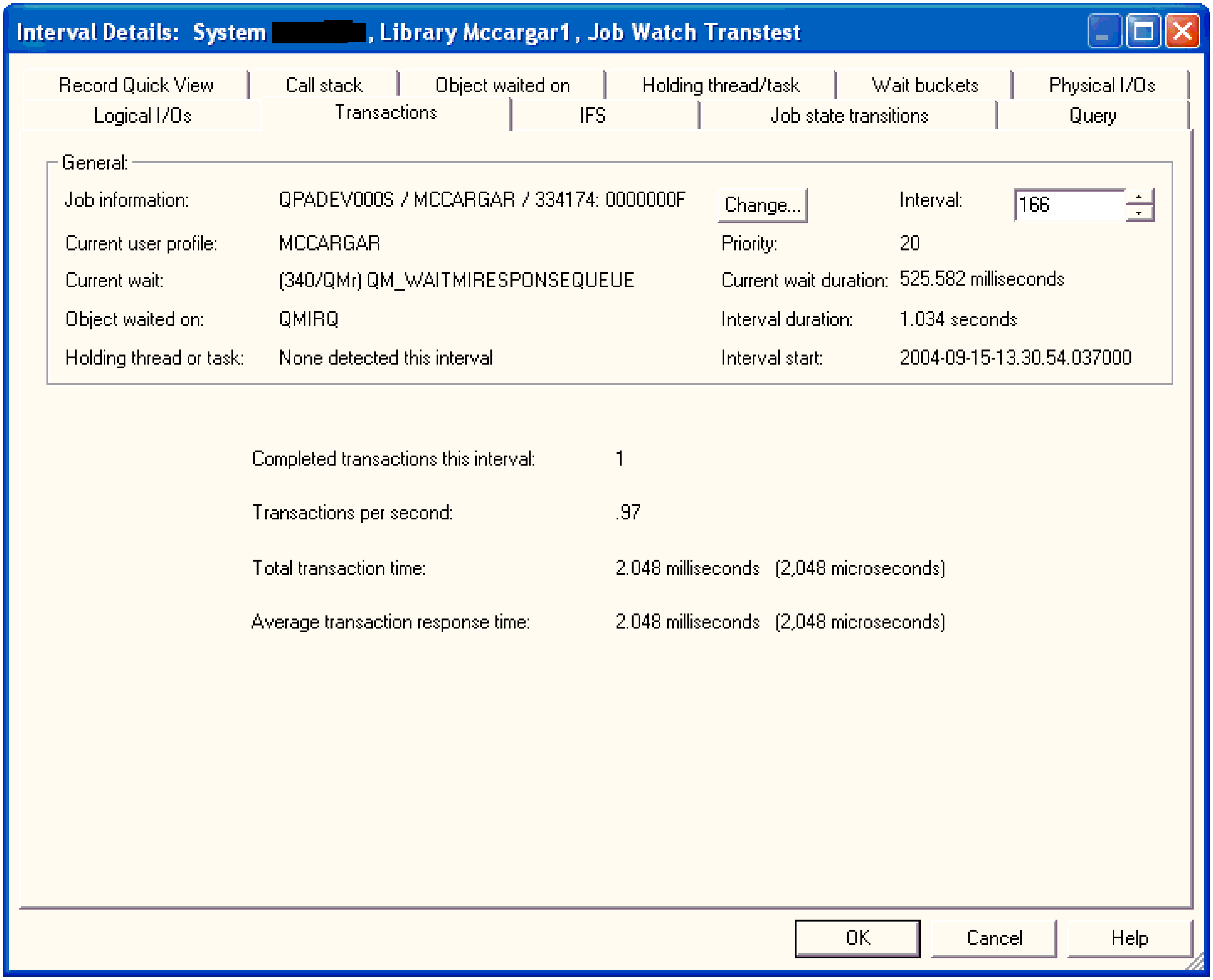
Field name	Description
Job information	The fully qualified job/thread id or task information this interface applies to. The Change... button allows the user to change the current job/thread/task to another. Changing the selected job will update the interface information on this panel but the information on the record quick view panel will still show the data for the original job.

Interval	Displays the selected interval the data shown applies to. The up and down buttons allow the user to change the selected interval. If the interval specified does not exist for the selected job no data will be shown.
Current user profile	The user profile currently active when the job was running at the interval specified.
Priority	The priority of the job for the specified interval.
Current wait	The current wait information contains information about the wait point (numeric identifier and 3 character wait code) and the description of the wait point from file QPYRTJWAT2. This field indicates the type of wait (or CPU) that was occurring when the snapshot to gather information about the job was taken. The wait point shown belongs to a wait bucket. Information about the current bucket is available on the "Wait buckets" panel in this interface.
Current wait duration	The total time the job has been in the current wait type. Everytime the wait type changes or CPU is used, this value is reset at 0. The current wait duration could be days for an idle job doing nothing and this would not indicate any problem. However if the current wait was a seize lock condition and the current wait duration was 30 seconds, this could flag a potential problem. For more information on understanding wait analysis, see the white paper available on the home page of the iDoctor for iSeries <a href="http://www-912.ibm.com/i_dir/idoctor.nsf">Web site</a>.
Object waited on	This field contains information about the object the job/thread/task waited on for the current interval (if any) as well as the type of object being waited on. If an object is specified on this field, additional information about the object is available on the "Object waited on" panel in this interface
Interval duration	The time that elapsed when collecting the selected interval.
Holding thread or task	This field contains information about the job/thread/task that was holding the selected job during the selected interval and preventing it from doing work. If a holding thread or task is specified within this field, additional information about the holder is available on the "Holding thread/task" panel in this interface
Interval start	The date/time the current interval began for the specified job.

Table of Contents Previous Next

2.6.1.10 Transactions

This page of the interval details window provides statistics on the transactions that ocurred during the specified job and interval. The total number of completed transactions, transaction rate, and transaction times are provided on this window.



Interval Details: System [redacted], Library Mccargar1 , Job Watch Transtest

General:

Job information:	QPADEV000S / MCCARGAR / 334174: 0000000F	Interval:	166
Current user profile:	MCCARGAR	Priority:	20
Current wait:	(340/QMr) QM_WAITMIRESPONSEQUEUE	Current wait duration:	525.582 milliseconds
Object waited on:	QMIRQ	Interval duration:	1.034 seconds
Holding thread or task:	None detected this interval	Interval start:	2004-09-15-13.30.54.037000

Completed transactions this interval: 1

Transactions per second: .97

Total transaction time: 2.048 milliseconds (2,048 microseconds)

Average transaction response time: 2.048 milliseconds (2,048 microseconds)

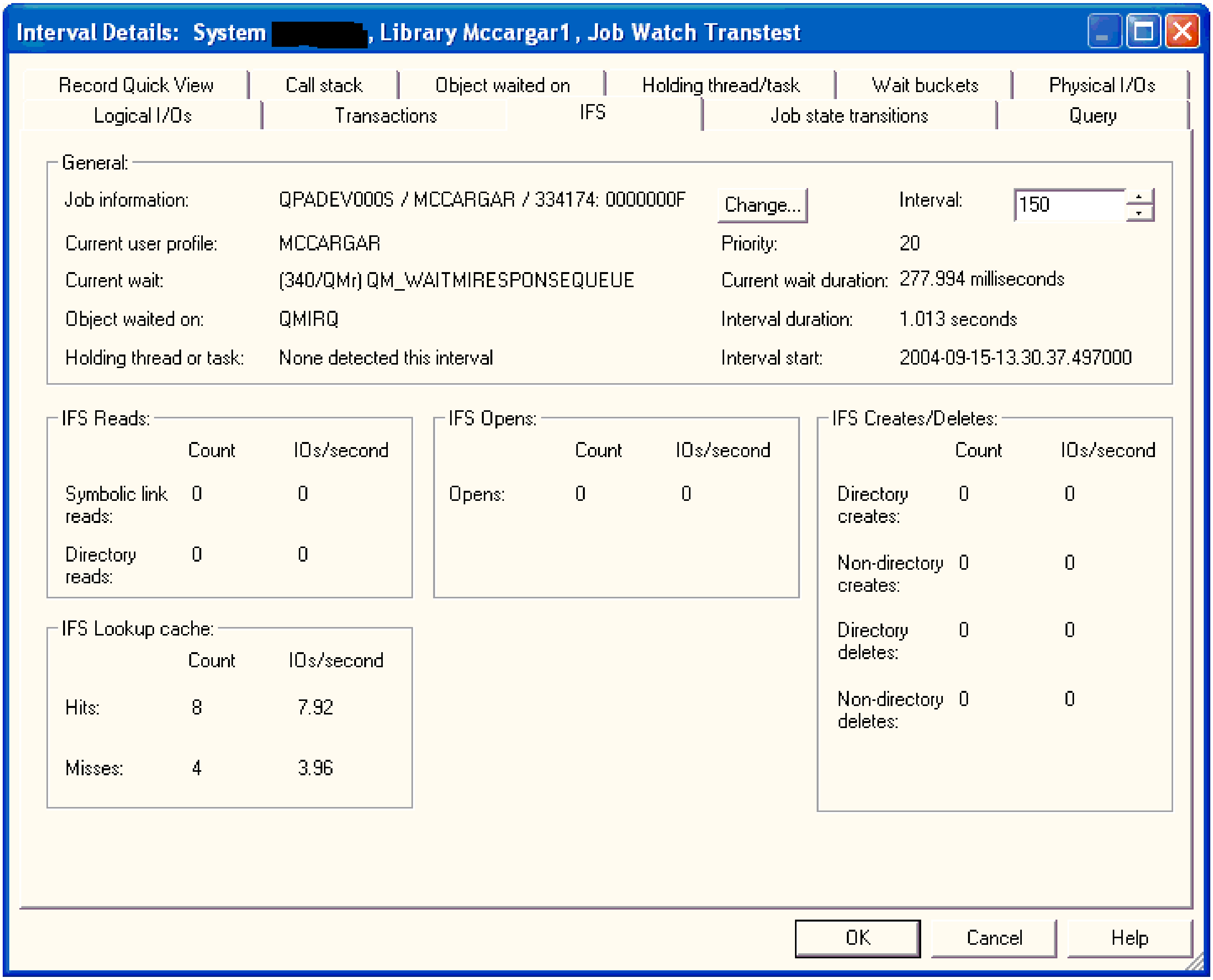
This window contains the following fields:

Field name	Description
Job information	The fully qualified job/thread id or task information this interface applies to. The Change... button allows the user to change the current job/thread/task to another. Changing the selected job will update the interface information on this panel but the information on the record quick view panel will still show the data for the original job.

Interval	Displays the selected interval the data shown applies to. The up and down buttons allow the user to change the selected interval. If the interval specified does not exist for the selected job no data will be shown.
Current user profile	The user profile currently active when the job was running at the interval specified.
Priority	The priority of the job for the specified interval.
Current wait	The current wait information contains information about the wait point (numeric identifier and 3 character wait code) and the description of the wait point from file QPYRTJWAT2. This field indicates the type of wait (or CPU) that was occurring when the snapshot to gather information about the job was taken. The wait point shown belongs to a wait bucket. Information about the current bucket is available on the "Wait buckets" panel in this interface.
Current wait duration	The total time the job has been in the current wait type. Everytime the wait type changes or CPU is used, this value is reset at 0. The current wait duration could be days for an idle job doing nothing and this would not indicate any problem. However if the current wait was a seize lock condition and the current wait duration was 30 seconds, this could flag a potential problem. For more information on understanding wait analysis, see the white paper available on the home page of the iDoctor for iSeries <a href="http://www-912.ibm.com/i_dir/idoctor.nsf">Web site</a>.
Object waited on	This field contains information about the object the job/thread/task waited on for the current interval (if any) as well as the type of object being waited on. If an object is specified on this field, additional information about the object is available on the "Object waited on" panel in this interface
Interval duration	The time that elapsed when collecting the selected interval.
Holding thread or task	This field contains information about the job/thread/task that was holding the selected job during the selected interval and preventing it from doing work. If a holding thread or task is specified within this field, additional information about the holder is available on the "Holding thread/task" panel in this interface
Interval start	The date/time the current interval began for the specified job.

2.6.1.11 IFS

This page of the interval details window provides statistics on IFS activity that occurred during the specified job and interval. IFS reads, opens, creates/deletes and lookup cache hits and misses are provided on this window.



Interval Details: System [redacted], Library Mccargar1 , Job Watch Transtest

General:

Job information: QPADEV000S / MCCARGAR / 334174: 0000000F

Interval: 150

Current user profile: MCCARGAR

Priority: 20

Current wait: (340/QMr) QM_WAITMIRESPONSEQUEUE

Current wait duration: 277.994 milliseconds

Object waited on: QMIRQ

Interval duration: 1.013 seconds

Holding thread or task: None detected this interval

Interval start: 2004-09-15-13.30.37.497000

IFS Reads:

	Count	IOs/second
Symbolic link reads:	0	0
Directory reads:	0	0

IFS Opens:

	Count	IOs/second
Opens:	0	0

IFS Creates/Deletes:

	Count	IOs/second
Directory creates:	0	0
Non-directory creates:	0	0
Directory deletes:	0	0
Non-directory deletes:	0	0

IFS Lookup cache:

	Count	IOs/second
Hits:	8	7.92
Misses:	4	3.96

This window contains the following fields:

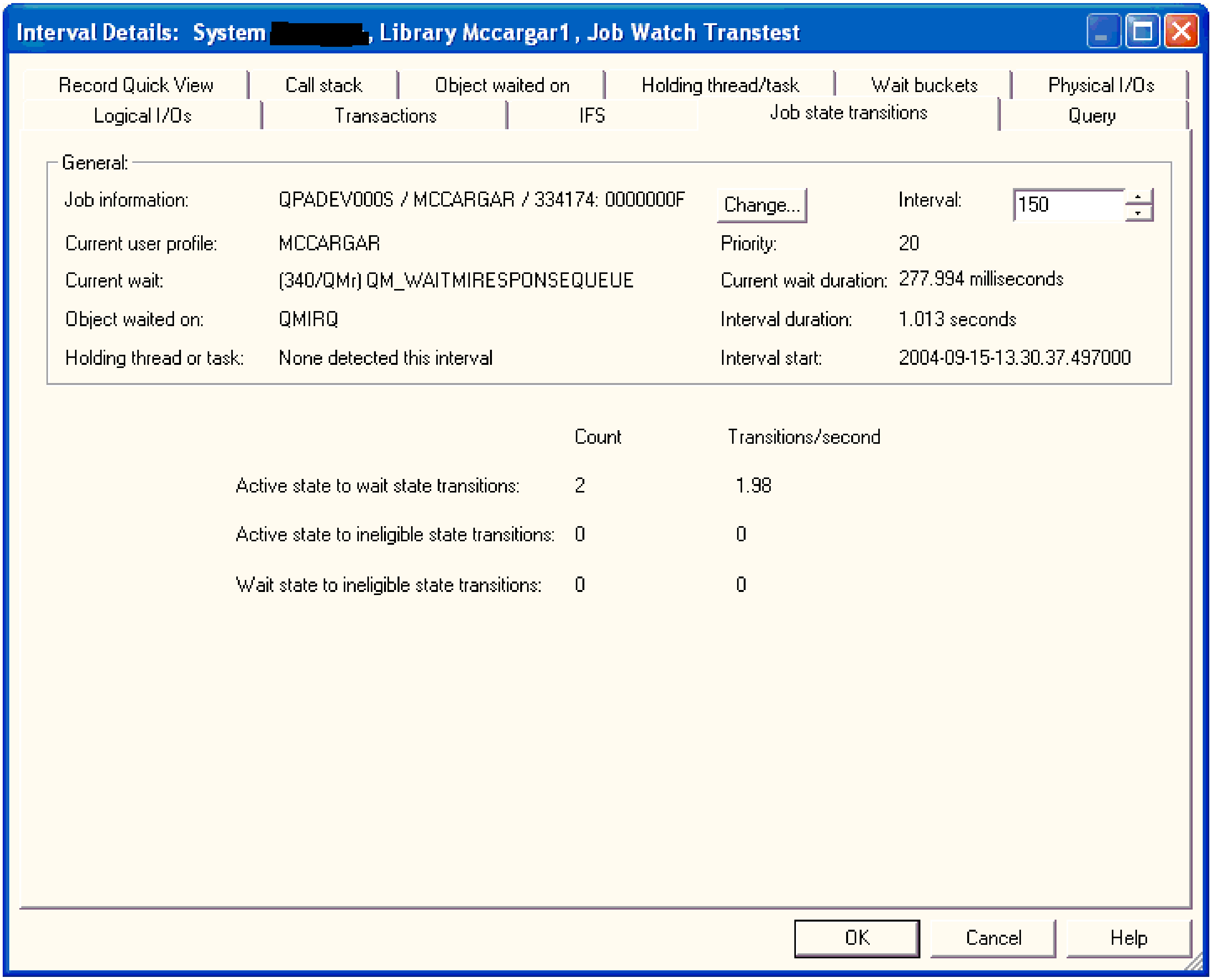
Field name	Description
Job information	The fully qualified job/thread id or task information this interface applies to. The Change... button allows the user to change the current job/thread/task to another. Changing the selected job will update the interface information on this panel but the information on the record quick view panel will still show the data for the original job.

Interval	Displays the selected interval the data shown applies to. The up and down buttons allow the user to change the selected interval. If the interval specified does not exist for the selected job no data will be shown.
Current user profile	The user profile currently active when the job was running at the interval specified.
Priority	The priority of the job for the specified interval.
Current wait	The current wait information contains information about the wait point (numeric identifier and 3 character wait code) and the description of the wait point from file QPYRTJWAT2. This field indicates the type of wait (or CPU) that was occurring when the snapshot to gather information about the job was taken. The wait point shown belongs to a wait bucket. Information about the current bucket is available on the "Wait buckets" panel in this interface.
Current wait duration	The total time the job has been in the current wait type. Everytime the wait type changes or CPU is used, this value is reset at 0. The current wait duration could be days for an idle job doing nothing and this would not indicate any problem. However if the current wait was a seize lock condition and the current wait duration was 30 seconds, this could flag a potential problem. For more information on understanding wait analysis, see the white paper available on the home page of the iDoctor for iSeries <a href="http://www-912.ibm.com/i_dir/idoctor.nsf">Web site</a>.
Object waited on	This field contains information about the object the job/thread/task waited on for the current interval (if any) as well as the type of object being waited on. If an object is specified on this field, additional information about the object is available on the "Object waited on" panel in this interface
Interval duration	The time that elapsed when collecting the selected interval.
Holding thread or task	This field contains information about the job/thread/task that was holding the selected job during the selected interval and preventing it from doing work. If a holding thread or task is specified within this field, additional information about the holder is available on the "Holding thread/task" panel in this interface
Interval start	The date/time the current interval began for the specified job.

Table of Contents | Previous | Next

2.6.1.12 Job state transitions

This page of the interval details window provides statistics on the job state transitions that ocurred during the specified job and interval.



Interval Details: System [redacted], Library Mccargar1 , Job Watch Transtest

General:

Job information:	QPADEV000S / MCCARGAR / 334174: 0000000F	Interval:	150
Current user profile:	MCCARGAR	Priority:	20
Current wait:	(340/QMr) QM_WAITMIRESPONSEQUEUE	Current wait duration:	277.994 milliseconds
Object waited on:	QMIRQ	Interval duration:	1.013 seconds
Holding thread or task:	None detected this interval	Interval start:	2004-09-15-13.30.37.497000

	Count	Transitions/second
Active state to wait state transitions:	2	1.98
Active state to ineligible state transitions:	0	0
Wait state to ineligible state transitions:	0	0

This window contains the following fields:

Field name	Description
Job information	The fully qualified job/thread id or task information this interface applies to. The Change... button allows the user to change the current job/thread/task to another. Changing the selected job will update the interface information on this panel but the information on the record quick view panel will still show the data for the original job.
Interval	Displays the selected interval the data shown applies to. The up and down buttons allow the user to change the selected interval. If the interval specified does not exist for the selected job no data will be shown.

Current user profile	The user profile currently active when the job was running at the interval specified.
Priority	The priority of the job for the specified interval.
Current wait	The current wait information contains information about the wait point (numeric identifier and 3 character wait code) and the description of the wait point from file QPYRTJWAT2. This field indicates the type of wait (or CPU) that was occurring when the snapshot to gather information about the job was taken. The wait point shown belongs to a wait bucket. Information about the current bucket is available on the "Wait buckets" panel in this interface.
Current wait duration	The total time the job has been in the current wait type. Everytime the wait type changes or CPU is used, this value is reset at 0. The current wait duration could be days for an idle job doing nothing and this would not indicate any problem. However if the current wait was a seize lock condition and the current wait duration was 30 seconds, this could flag a potential problem. For more information on understanding wait analysis, see the white paper available on the home page of the iDoctor for iSeries <a href="http://www-912.ibm.com/i_dir/idoctor.nsf">Web site</a>.
Object waited on	This field contains information about the object the job/thread/task waited on for the current interval (if any) as well as the type of object being waited on. If an object is specified on this field, additional information about the object is available on the "Object waited on" panel in this interface
Interval duration	The time that elapsed when collecting the selected interval.
Holding thread or task	This field contains information about the job/thread/task that was holding the selected job during the selected interval and preventing it from doing work. If a holding thread or task is specified within this field, additional information about the holder is available on the "Holding thread/task" panel in this interface
Interval start	The date/time the current interval began for the specified job.

Table of Contents Previous Next

2.6.2 Wait graphs - detailed

The detailed wait graphs provide information about waiting that occurred in a job within a job-specific Job Watch.

2.6.2.1 Run/wait time signature

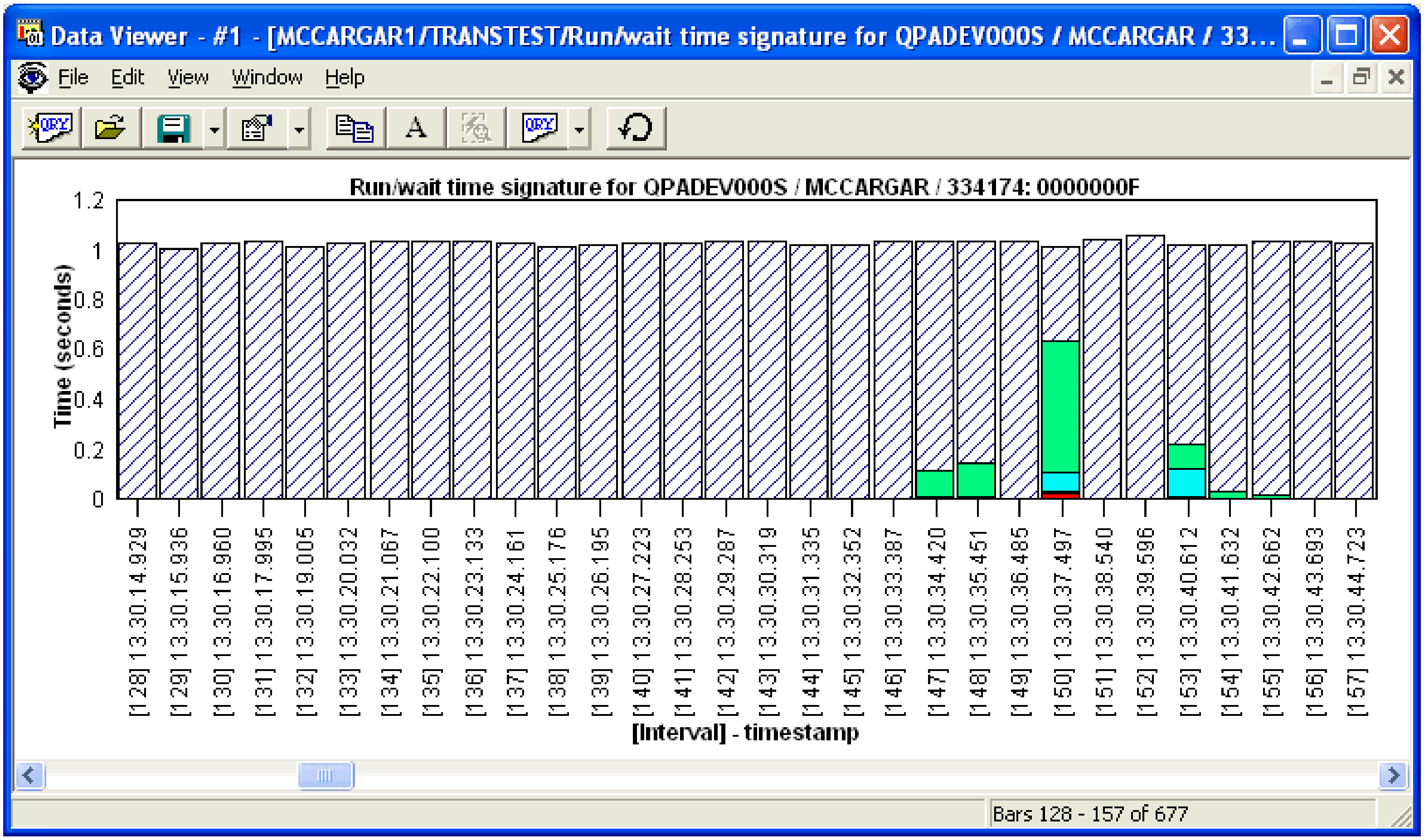
Description: This graph shows a **detailed** look at the time a job spent in various types of waits throughout its existence during the job watch. The graph makes use of the 32 run/wait buckets. To determine which type of wait a color represents, place the mouse pointer of a color/bar of interest and a description of the wait as well as other statistics about the interval/wait will be displayed. Due to screen space limitations and the high number of colors the legend is not shown by default but it is available by using the Show Legend option on the Graph Definition -> General page.

This graph can help pinpoint within a specific job where the job started to perform work and how efficiently it was performing that work. If a interval in the graph suddenly becomes suspicious (high seize time for example) clicking on the bar will provide other valuable information through the interval details window.

The total height of the stacked bar (which includes all waits for that interval and CPU) indicates the elapsed interval time.

Graph Type: detailed job-specific (stacked vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Each color represents the amount of time the job/thread/task spent in one of the 32 different wait buckets within an interval. Time is listed in seconds.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	Description

Display Stack	Shows the interval details window and brings focus to the call stack tab. The call stack may not always be available if it was indicated at collection time to not collect the call stack. In this case the call stack window will not show a call stack but other information about the job's performance during this interval will be available.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in the interval details window. This window will also appear by left-clicking on any bar in the graph.

Table of Contents Previous Next

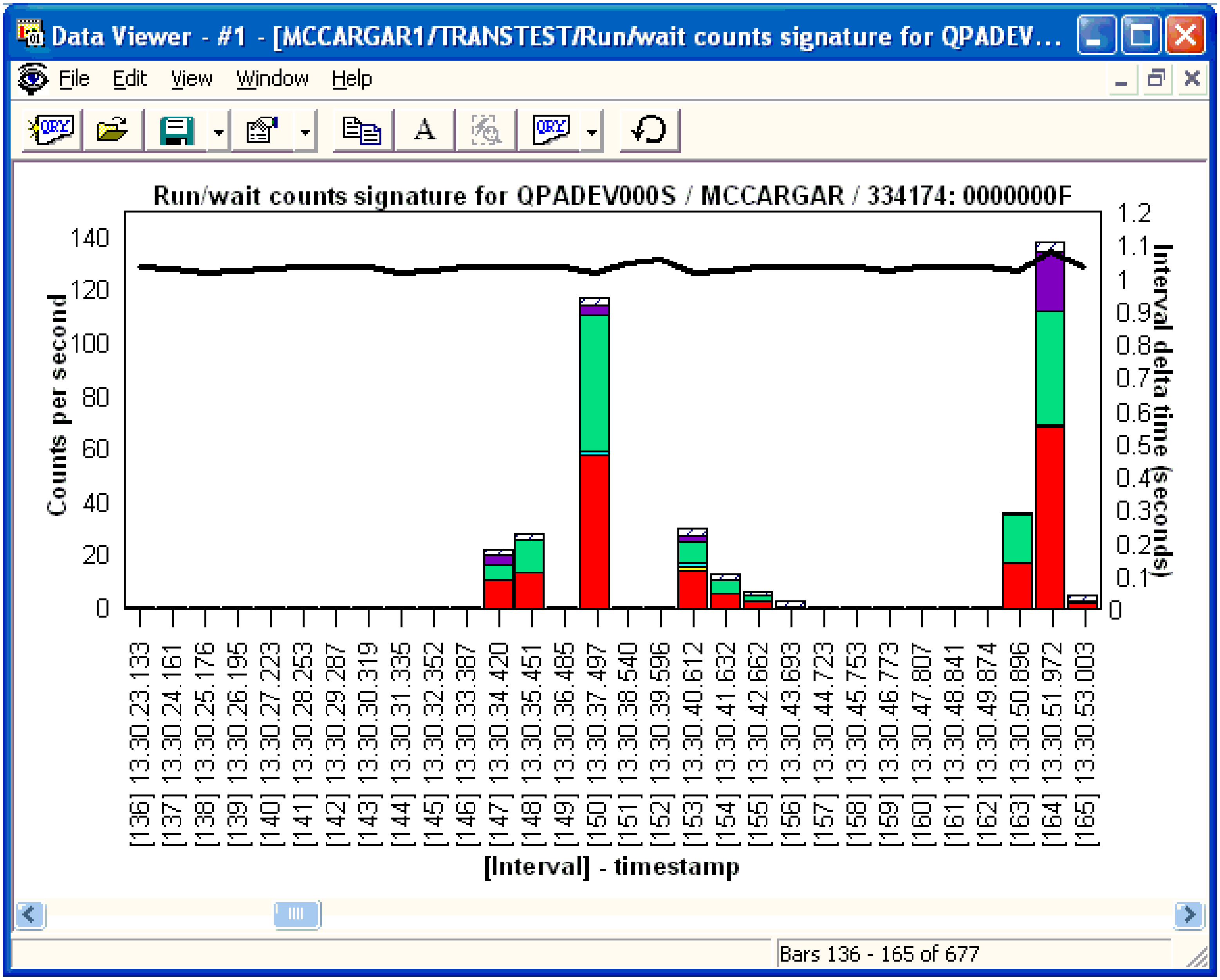
2.6.2.2 Run/wait counts signature

Description: This graph shows a **detailed** look at the transitions from one wait type to another throughout a job's existence during the job watch. The graph makes use of the counts provided by the 32 run/wait buckets. To determine which type of wait a color represents, place the mouse pointer of a color/bar of interest and a description of the wait as well as other statistics about the interval/wait will be displayed. Due to screen space limitations and the high number of colors the legend is not shown by default but it is available by using the Show Legend option on the Graph Definition -> General page.

This graph can help pinpoint within a specific job where things happened where a job was idle vs not idle. If a job is stuck in the way wait for an entire interval the size of the bar will be very small (1 count per interval time). On the other hand if the job is performing work the counts will be noticeably high.

Graph Type: detailed job-specific (stacked vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Each color represents the amount of time the job/thread/task spent in one of the 32 different wait buckets within an interval. Time is listed in seconds.
Second Y-Axis: This line represents the elapsed time for each interval in seconds. The interval time could vary depending on the overall performance of the system which can cause delays in the Job Watcher engine. For example, these delays may cause an interval of 1 second to actually take 1.5 seconds to complete.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Display Stack	Shows the interval details window and brings focus to the call stack tab. The call stack may not always be available if it was indicated at collection time to not collect the call stack. In this case the call stack window will not show a call stack but other information about the job's performance during this interval will be available.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.

Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in the interval details window. This window will also appear by left-clicking on any bar in the graph.

Table of Contents | Previous | Next

2.6.2.3 Objected waited on

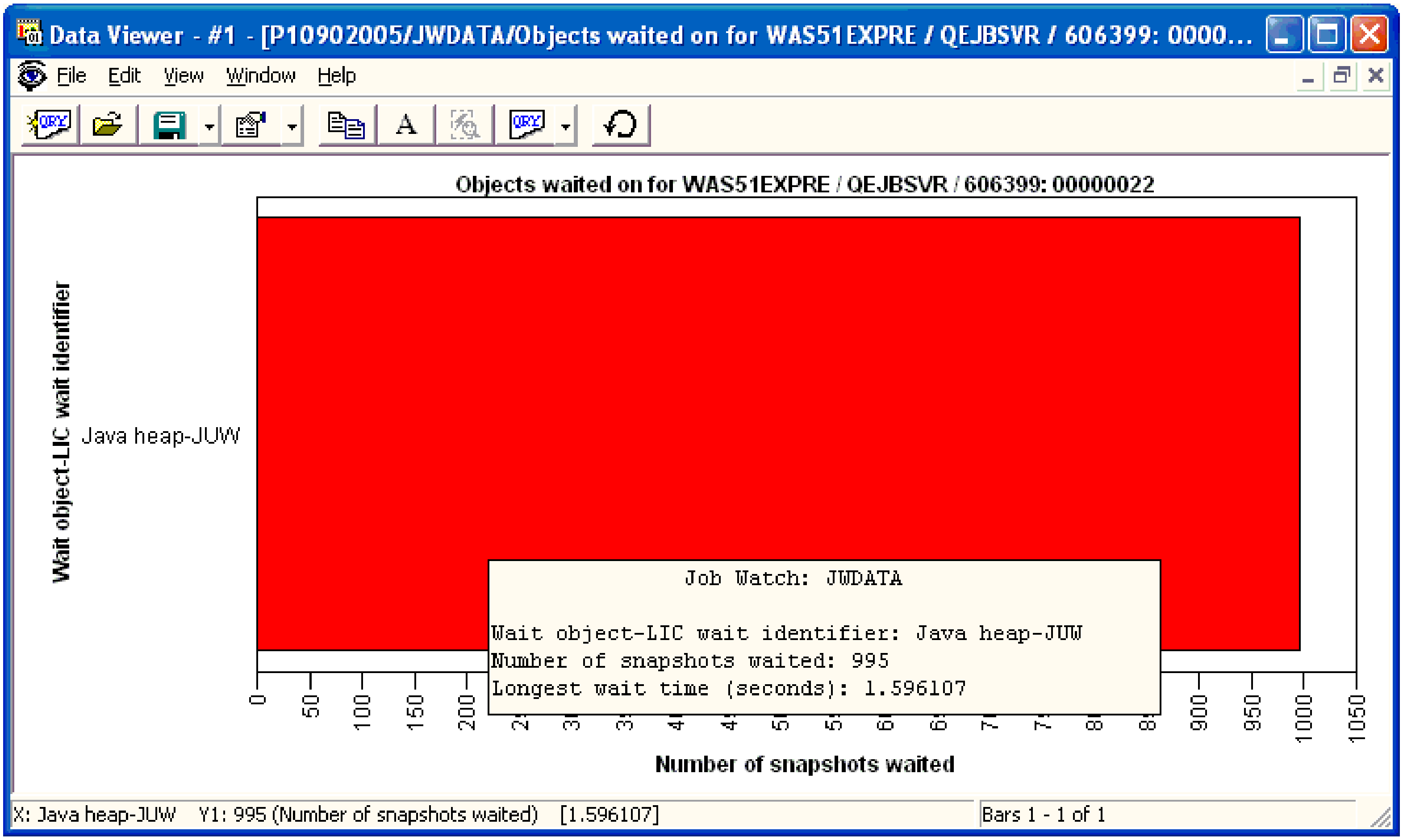
Description: This graph shows the list of objects a job has waited on throughout its existence during the job watch. The graph shows the number of snapshots taken where the job was waiting to use an object, the name/type information for each object and the type of wait.

The objects waited on does not necessarily indicate a problem, depending on the type of work the job is performing as well as what is normal for the application. However in other cases especially situations where a program is waiting to use a file, the objected waited on feature can help identify locking conflicts.

Placing the mouse over a bar will also show the longest chunk of time spent waiting on the object during the collection.

Graph Type: detailed job-specific (horizontal bar)
X-axis: The name of the object being waited on, and the 3 character LIC wait identifier showing the type of wait that occurred. 1 bar is created per object being waited on for the job.
Y-Axis: The number of snapshots taken where the object waited on was the same.

Example:



Data Viewer - #1 - [P10902005/JWDATA/Objects waited on for WAS51EXPRE / QEJBSVR / 606399: 0000...
File Edit View Window Help
Objects waited on for WAS51EXPRE / QEJBSVR / 606399: 00000022
Wait object-LIC wait identifier
Java heap-JUW
0 50 100 150 200 ... 900 950 1000 1050
Number of snapshots waited

Job Watch: JWDATA

Wait object-LIC wait identifier: Java heap-JUW
Number of snapshots waited: 995
Longest wait time (seconds): 1.596107

X: Java heap-JUW Y1: 995 (Number of snapshots waited) [1.596107]
Bars 1 - 1 of 1

Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Display Stack	Not available for this graph.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.

Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window.

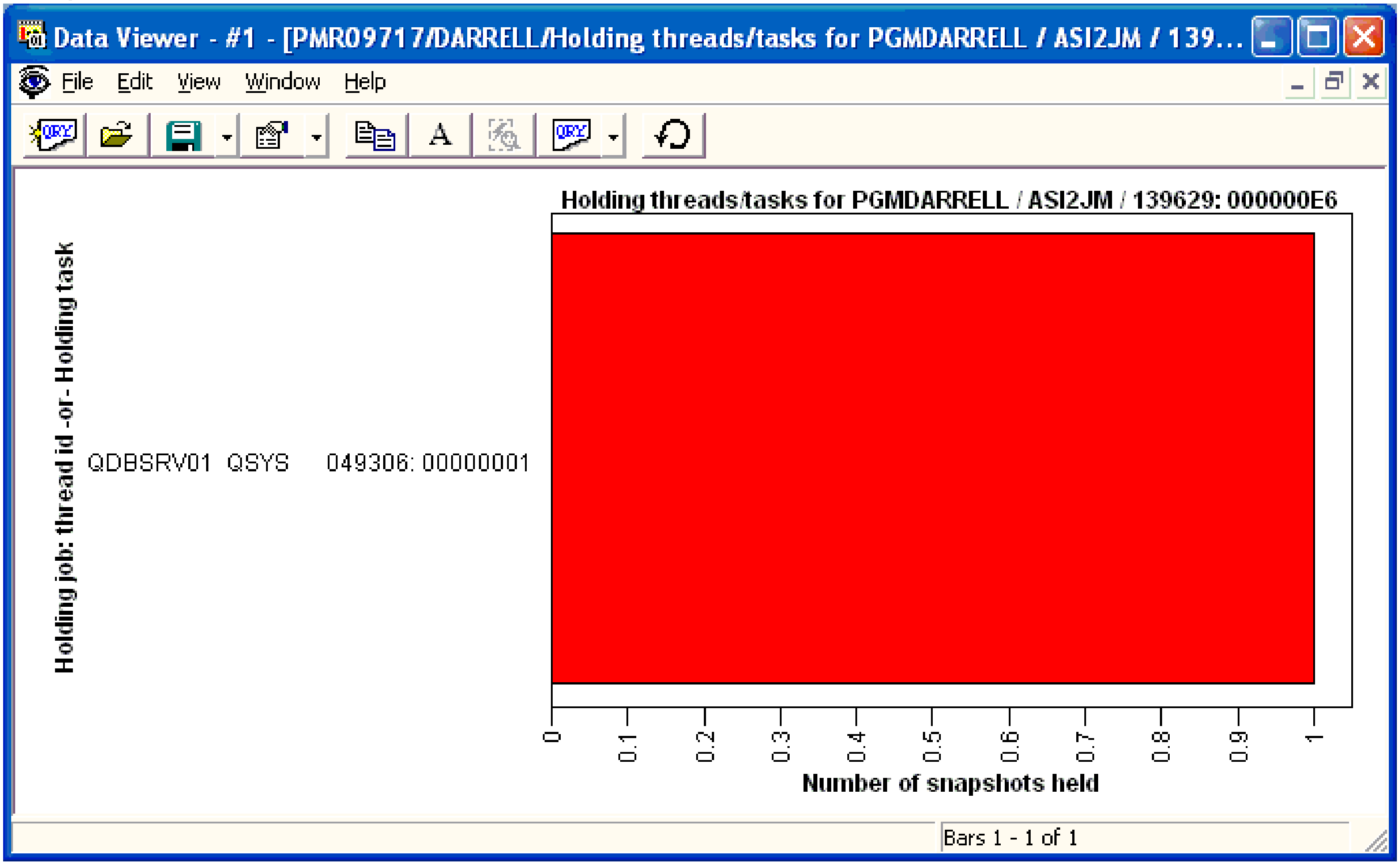
Table of Contents | Previous | Next

2.6.2.4 Holding threads/tasks

Description: This graph displays a list of jobs/threads/tasks that were holding a specific job in the job watch.

Graph Type: detailed job-specific (horizontal bar)
X-axis: Each bar represents a job that was holding the job listed in the title of the graph (in the example below the holder is QDBSRV01 and the waiter job is PGMDARRELL)
Y-Axis: The number of snapshots where the holder was detected. This does NOT mean the job was held for this many intervals consecutively. This value is a count of how many times the holder was detected for the waiting job.

Example:


Data Viewer - #1 - [PMR09717/DARRELL/Holding threads/tasks for PGMDARRELL / ASI2JM / 139...
File Edit View Window Help
Holding threads/tasks for PGMDARRELL / ASI2JM / 139629: 000000E6
Holding job: thread id -or- Holding task
QDBSRV01 QSYS 049306: 00000001
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Number of snapshots held
Bars 1 - 1 of 1

Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Display Stack	Not available for this graph.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.

Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window.

Table of Contents | Previous | Next

2.6.3 CPU graphs - detailed

The detailed CPU graphs provide statistics for the CPU and CPU queueing that occurred for a job in a job-specific Job Watch.

Table of Contents | Previous | Next

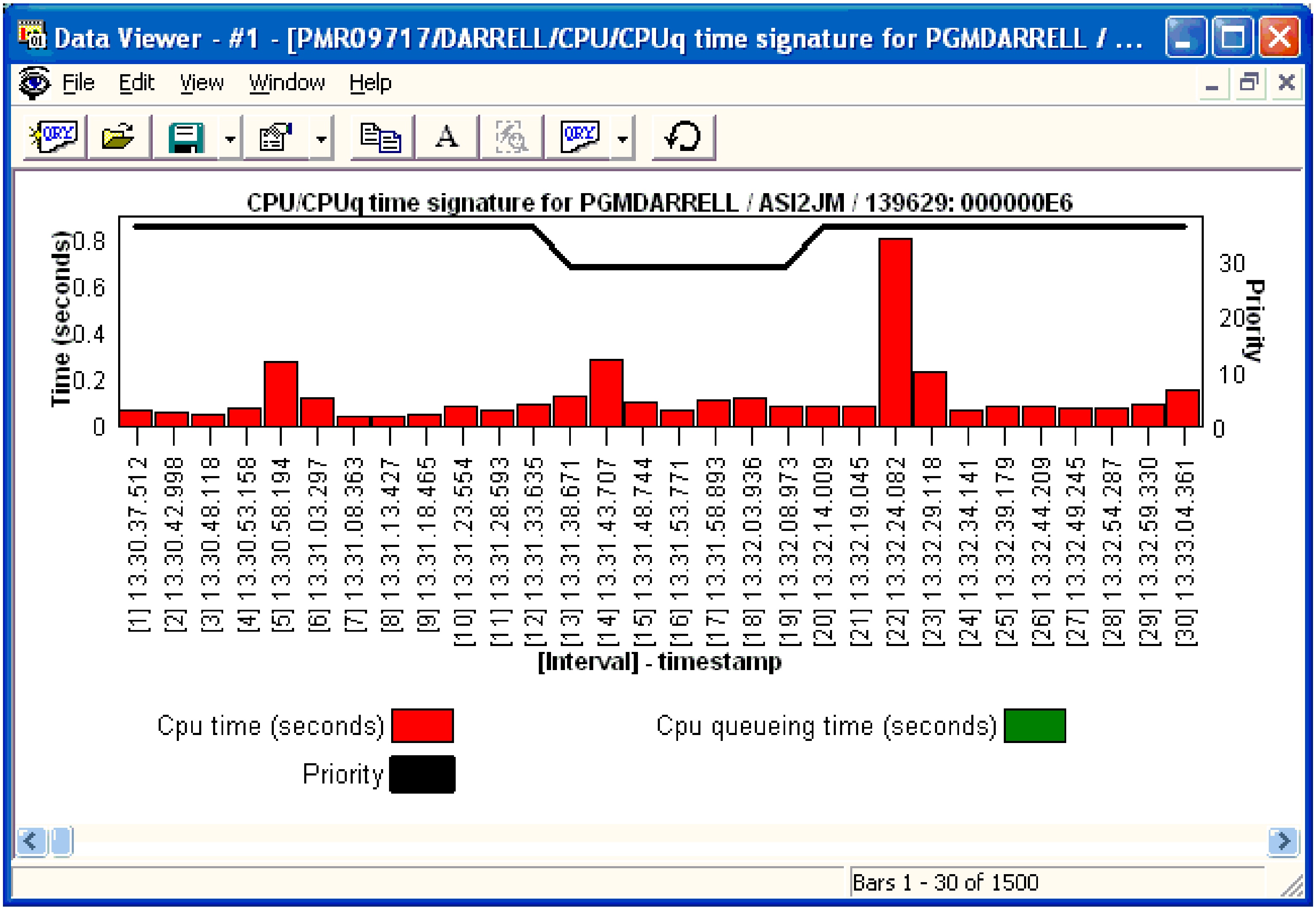
2.6.3.1 CPU/CPUq time signature

Description: This graph shows a **detailed** look at the time the job spent using CPU and waiting for CPU throughout its existence during the job watch. Red colors indicate CPU use, and green colors represent CPU queueing time.

This graph can help locate where high CPU, or CPU queueing is occurring for a specific job.

Graph Type: detailed job-specific (stacked vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Each color represents the amount of time the job/thread/task spent using CPU or CPU queueing. All times are listed in seconds.
Second Y-Axis: The line shows the job's priority during the collection.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	**Description**
Display Stack	Shows the interval details window and brings focus to the call stack tab. The call stack may not always be available if it was indicated at collection time to not collect the call stack. In this case the call stack window will not show a call stack but other information about the job's performance during this interval will be available.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in the interval details window. This window will also appear by left-clicking on any bar in the graph.

Table of Contents | Previous | Next

2.6.4 DASD/IO graphs - detailed

The detailed DASD/IO graphs provide statistics for disk space allocation/deallocations, as well all physical and logical IOs that occurred for a job in a job-specific Job Watch.

Table of Contents | Previous | Next

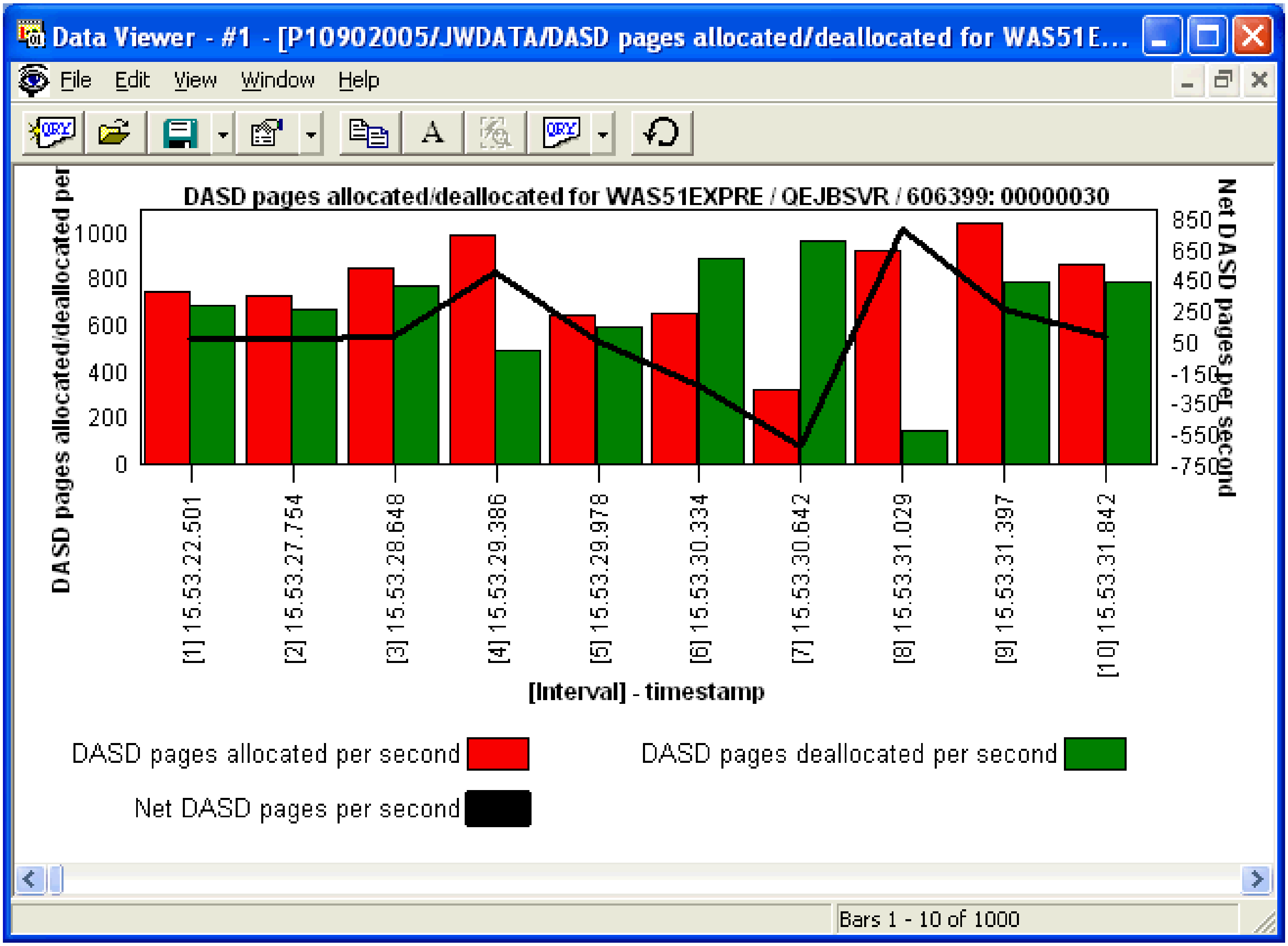
2.6.4.1 DASD pages allocated/deallocated

Description: This graph shows a **detailed** look at the amount of disk space allocated and deallocated by a job throughout its existence during the job watch. Red colors indicate space allocations and green colors indicate space deallocations. Allocations and deallocations listed are the number of 4K dasd pages allocated/deallocated per second. A "DASD page" is a 4k (4096 bytes) block of data.

This graph can help locate where high CPU, or CPU queueing is occurring for a specific job.

Graph Type: detailed job-specific (vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Contains the rate of DASD pages allocations and deallocations for each interval.
Second Y-Axis: This line represents the rate of allocations minus the rate of deallocations per interval (the net dasd pages per second)

Example:


Data Viewer - #1 - [P10902005/JWDATA/DASD pages allocated/deallocated for WAS51E...
File
Edit
View
Window
Help
QRY
A
QRY
DASD pages allocated/deallocated for WAS51EXPRE / QEJBSVR / 606399: 00000030
DASD pages allocated/deallocated per
1000
800
600
400
200
0
850
650
450
250
50
-150
-350
-550
-750
Net DASD pages per second
[1] 15.53.22.501
[2] 15.53.27.754
[3] 15.53.28.648
[4] 15.53.29.386
[5] 15.53.29.978
[6] 15.53.30.334
[7] 15.53.30.642
[8] 15.53.31.029
[9] 15.53.31.397
[10] 15.53.31.842
[Interval] - timestamp
DASD pages allocated per second 🟥
DASD pages deallocated per second 🟩
Net DASD pages per second ■
Bars 1 - 10 of 1000

Right-clicking on a bar in this graph provides the following menu options:

Menu	**Description**
Display Stack	Shows the interval details window and brings focus to the call stack tab. The call stack may not always be available if it was indicated at collection time to not collect the call stack. In this case the call stack window will not show a call stack but other information about the job's performance during this interval will be available.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in the interval details window. This window will also appear by left-clicking on any bar in the graph.

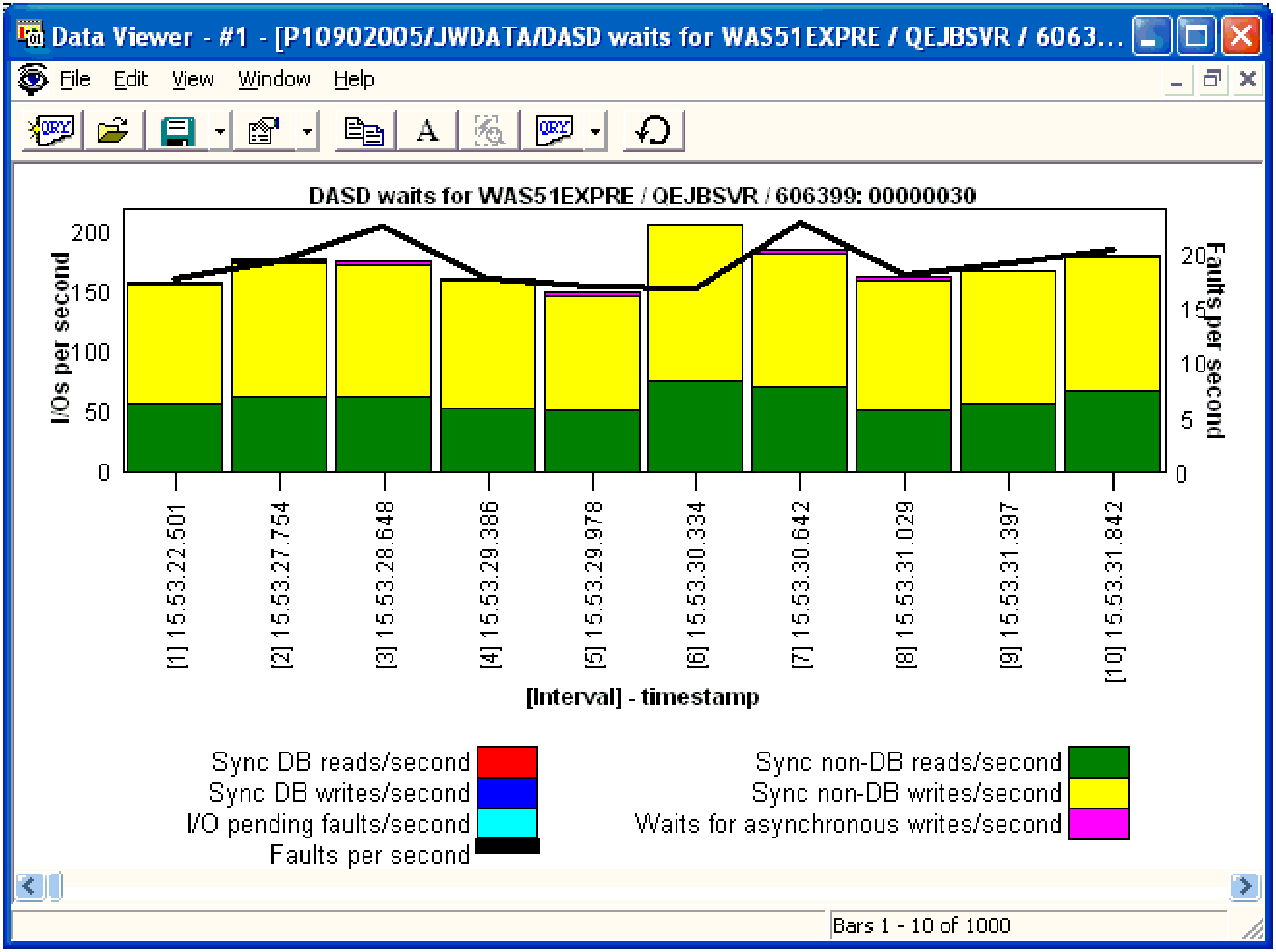
Table of Contents | Previous | Next

2.6.4.2 DASD waits

Description: This graph shows a **detailed** look at the disk IO operations and page faulting occuring by a job throughout its existence during the job watch. Each color in the graph represents a different type of IO operation.

Graph Type: detailed job-specific (stacked vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Contains rates of various types of disk operations for a specific job.
Second Y-Axis: This line shows the faults per second over time.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	Description

Display Stack	Shows the interval details window and brings focus to the call stack tab. The call stack may not always be available if it was indicated at collection time to not collect the call stack. In this case the call stack window will not show a call stack but other information about the job's performance during this interval will be available.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in the interval details window. This window will also appear by left-clicking on any bar in the graph.

Table of Contents Previous Next

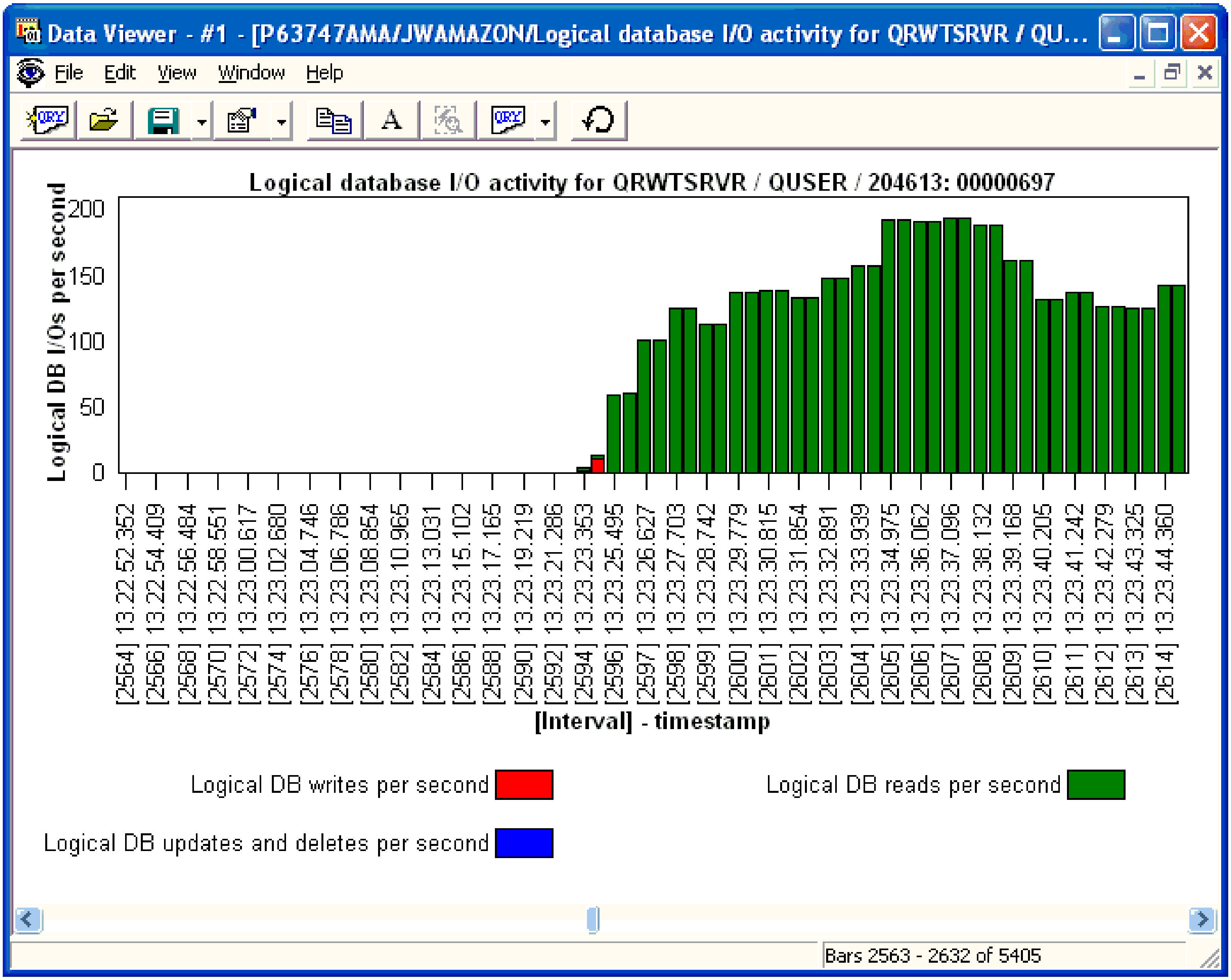
2.6.4.4 Logical database I/O activity

Description: This graph shows a **detailed** look at the rates of logical database I/O activity including writes, reads and updates/deletes for a job in the job-specific job watch.

This graph may be used to locate where LDIO activity is occurring within a specific job.

Graph Type: detailed job-specific (vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Contains the rate of logical database I/Os for writes, reads and updates/deletes.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Display Stack	Shows the interval details window and brings focus to the call stack tab. The call stack may not always be available if it was indicated at collection time to not collect the call stack. In this case the call stack window will not show a call stack but other information about the job's performance during this interval will be available.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in the interval details window. This window will also appear by left-clicking on any bar in the graph.

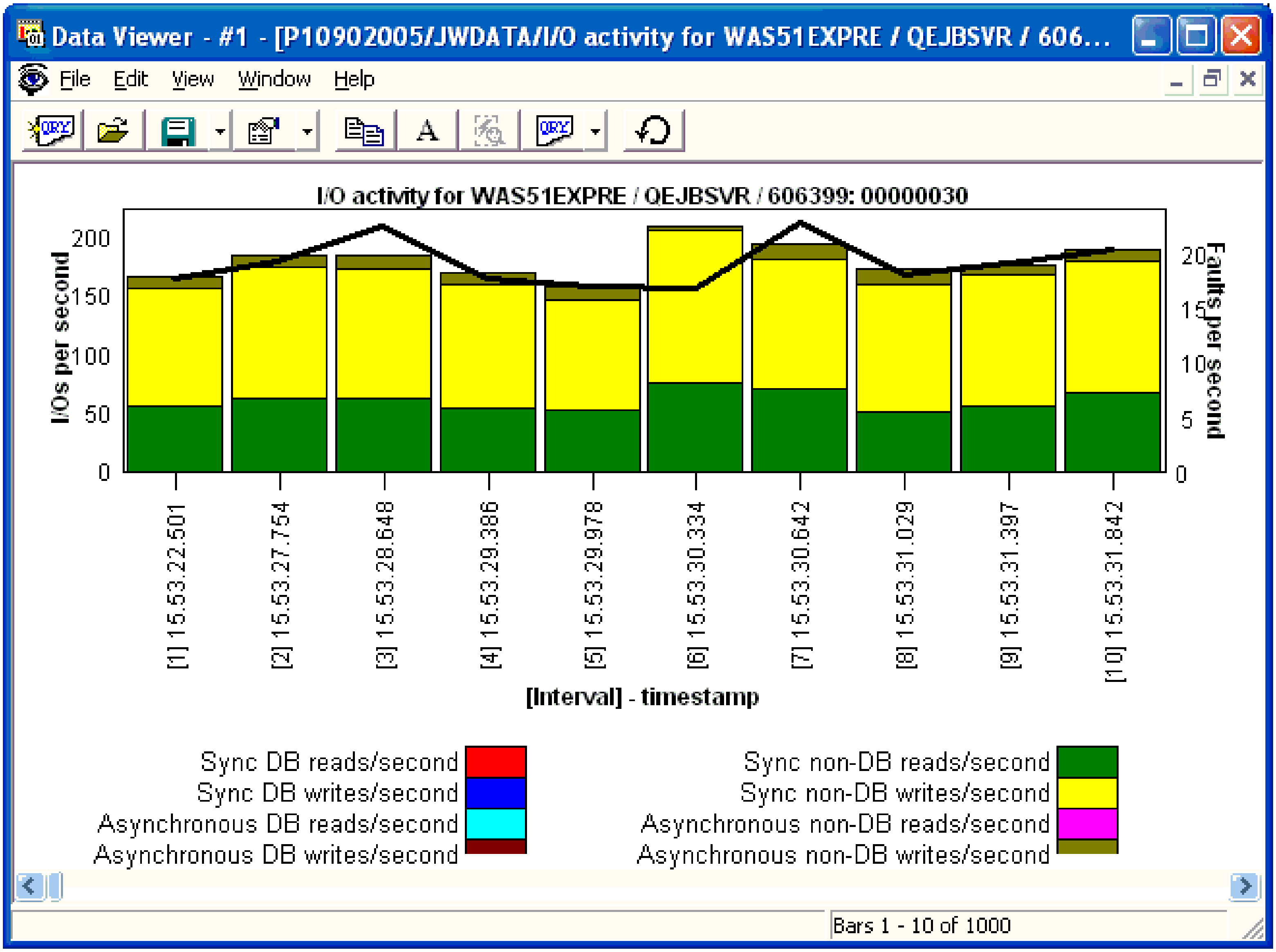
Table of Contents Previous Next

2.6.4.3 I/O Activity

Description: This graph shows a **detailed** look at the physical I/Os that were made by a job throughout its existence during the job watch. The graph provides rates of database and non-database reads and writes. Page fault rates are listed on the secondary Y-axis.

Graph Type: detailed job-specific (stacked vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Contains the rate of async and synchronous DB and non-DB reads and writes.
Second Y-Axis: The line represents the page fault rate for the job over time.

Example:


Data Viewer - #1 - [P10902005/JWDATA/I/O activity for WAS51EXPRE / QEJBSVR / 606...
File Edit View Window Help
QRY
A
QRY
I/O activity for WAS51EXPRE / QEJBSVR / 606399: 00000030
I/Os per second
200
150
100
50
0
Faults per second
20
15
10
5
0
[1] 15.53.22.501
[2] 15.53.27.754
[3] 15.53.28.648
[4] 15.53.29.386
[5] 15.53.29.978
[6] 15.53.30.334
[7] 15.53.30.642
[8] 15.53.31.029
[9] 15.53.31.397
[10] 15.53.31.842
[Interval] - timestamp
Sync DB reads/second 🟥
Sync DB writes/second 🟦
Asynchronous DB reads/second
Asynchronous DB writes/second
Sync non-DB reads/second 🟩
Sync non-DB writes/second 🟨
Asynchronous non-DB reads/second
Asynchronous non-DB writes/second
Bars 1 - 10 of 1000

Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Display Stack	Shows the interval details window and brings focus to the call stack tab. The call stack may not always be available if it was indicated at collection time to not collect the call stack. In this case the call stack window will not show a call stack but other information about the job's performance during this interval will be available.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in the interval details window. This window will also appear by left-clicking on any bar in the graph.

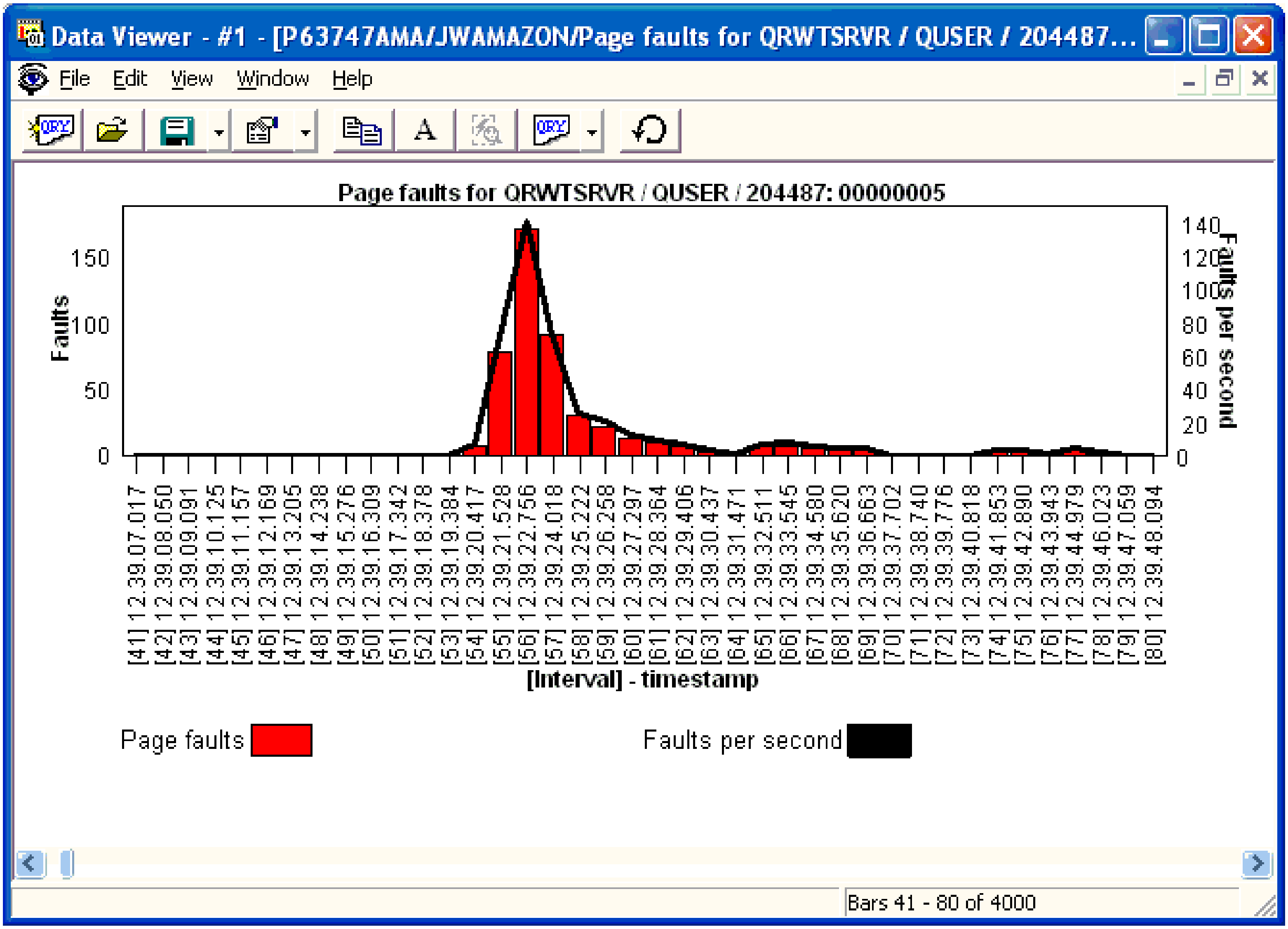
Table of Contents | Previous | Next

2.6.4.5 Page faults

Description: This graph shows a **detailed** look at the total number of pages faults and the rate of faults occurring per second for a job in a job-specific job watch.

Graph Type: detailed job-specific (vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Each bar represents the total page faults that occurred for the job in an interval.
Second Y-Axis: The line represents the page fault rate for each interval over time.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	Description

Display Stack	Shows the interval details window and brings focus to the call stack tab. The call stack may not always be available if it was indicated at collection time to not collect the call stack. In this case the call stack window will not show a call stack but other information about the job's performance during this interval will be available.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in the interval details window. This window will also appear by left-clicking on any bar in the graph.

Table of Contents | Previous | Next

2.6.5 IFS graphs - detailed

The detailed IFS graphs provide IFS statistics for a job in a job-specific Job Watch.

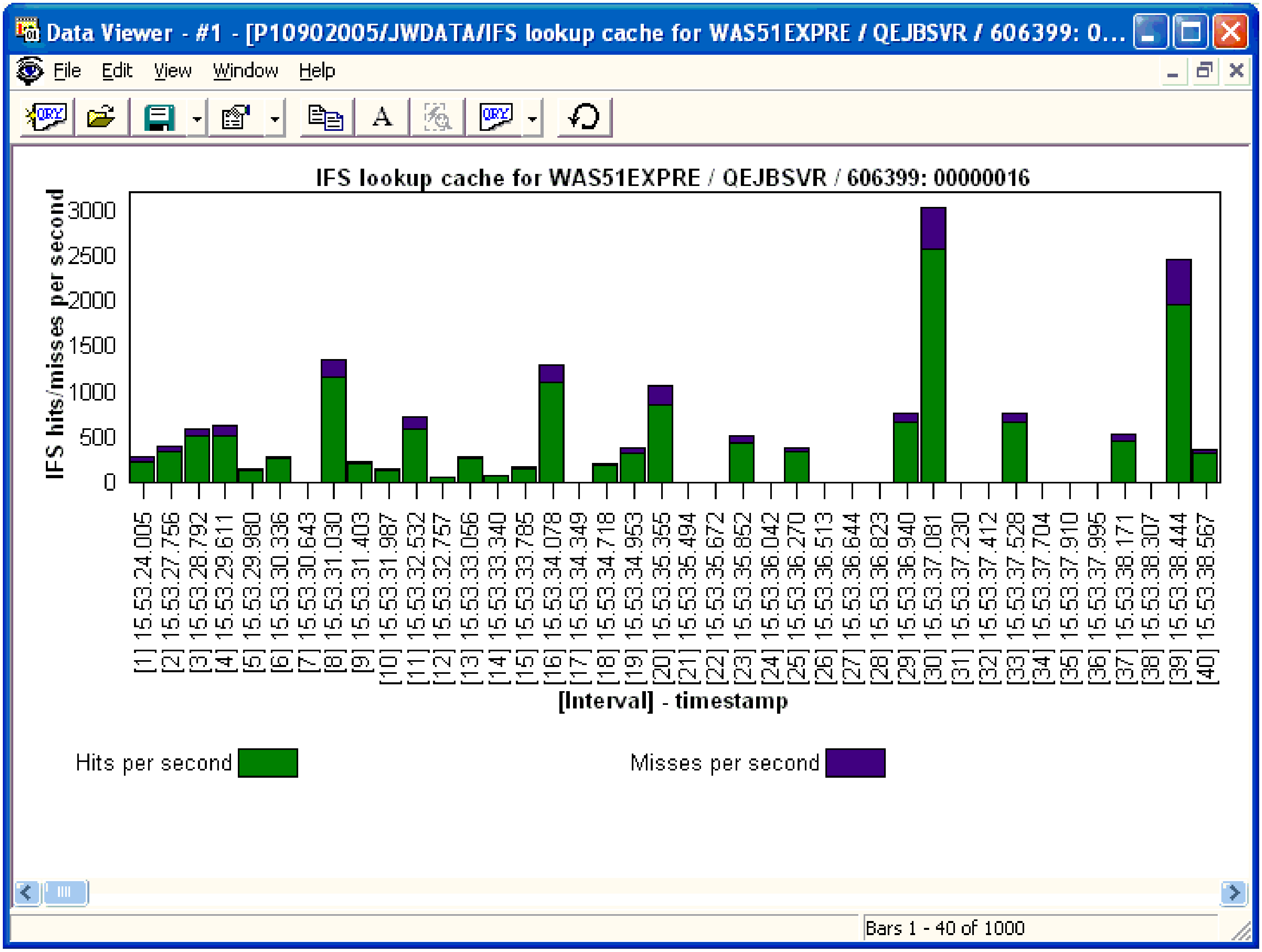
Table of Contents | Previous | Next

2.6.5.1 IFS lookup cache

Description: This graph shows a **detailed** look at the rate of hits/misses for the IFS lookup cache by a job throughout its existence during the job watch. Lookup cache hit rate is indicated in green and misses are shown in blue.

Graph Type: detailed job-specific (stacked vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Contains the rate of lookup cache hits and misses.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Display Stack	Shows the interval details window and brings focus to the call stack tab. The call stack may not always be available if it was indicated at collection time to not collect the call stack. In this case the call stack window will not show a call stack but other information about the job's performance during this interval will be available.

Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in the interval details window. This window will also appear by left-clicking on any bar in the graph.

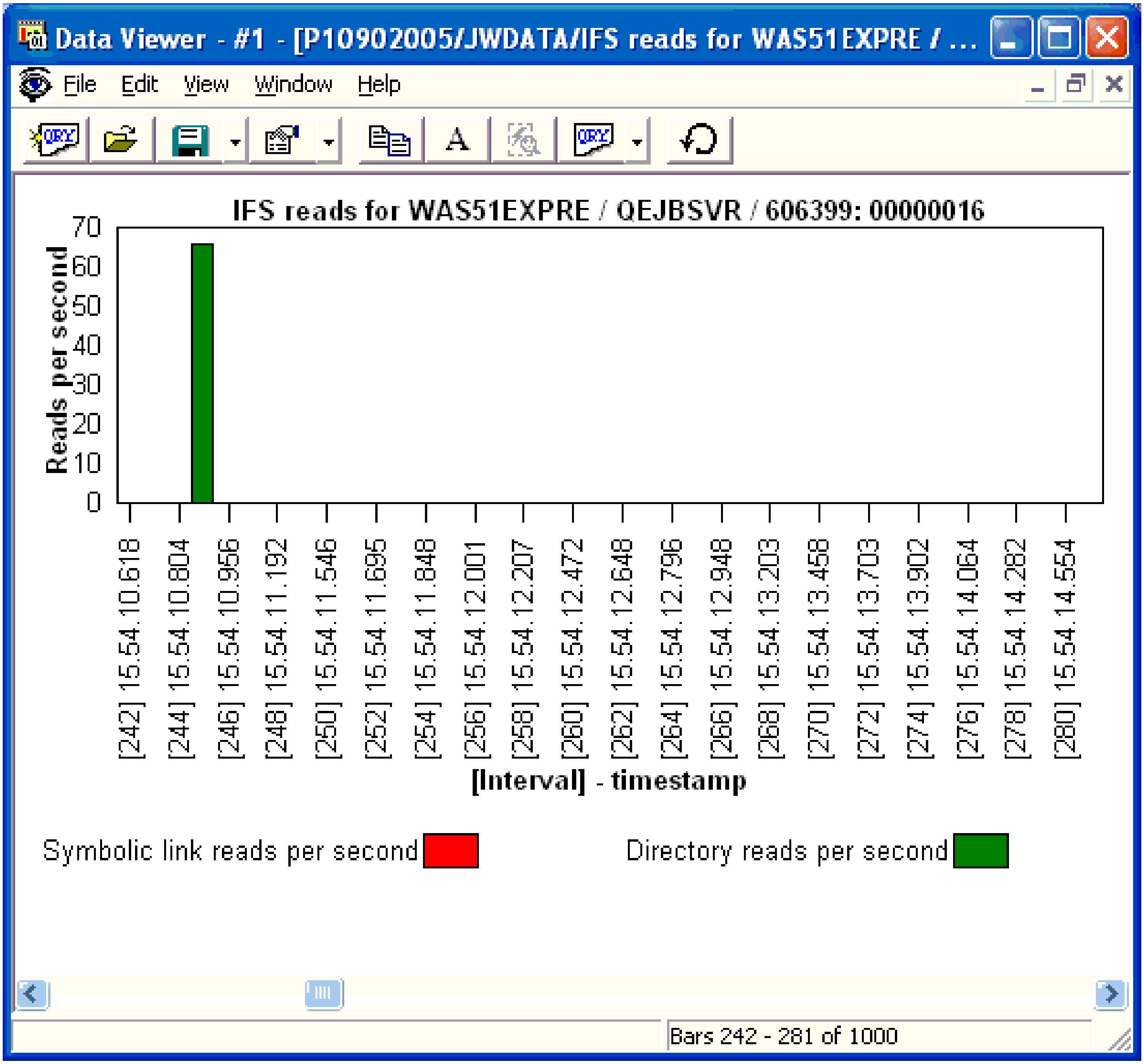
Table of Contents | Previous | Next

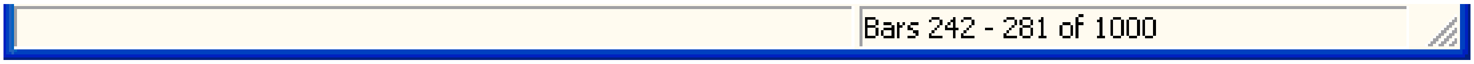
2.6.5.2 IFS reads

Description: This graph shows a **detailed** look at the rate of IFS directory reads and symbolic link reads for a job throughout its existence during the job watch.

Graph Type: detailed job-specific (stacked vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Contains the rate of symbolic link reads per second in red and directory reads per second in green.

Example:





Bars 242 - 281 of 1000

Right-clicking on a bar in this graph provides the following menu options:

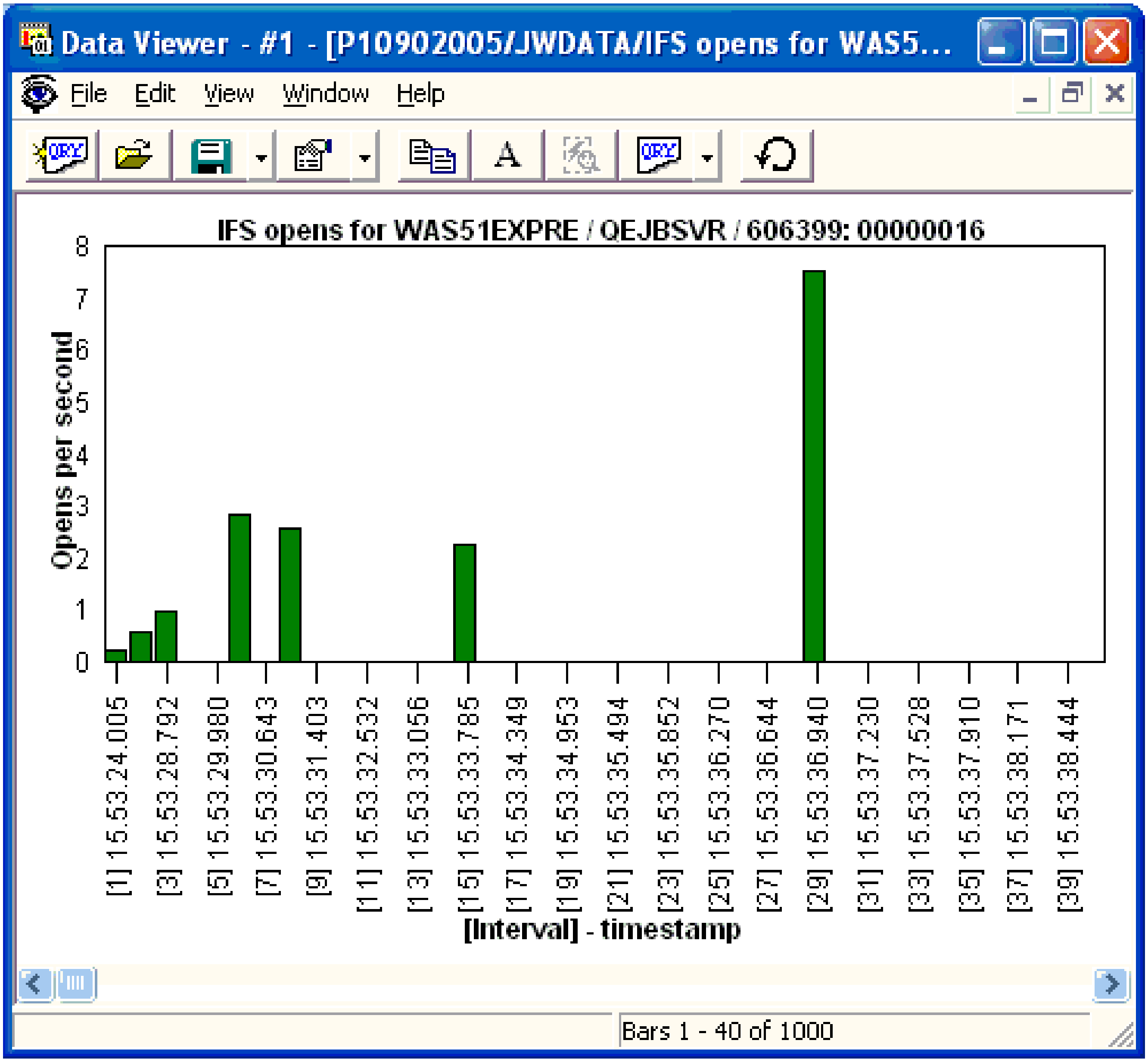
Menu	**Description**
Display Stack	Shows the interval details window and brings focus to the call stack tab. The call stack may not always be available if it was indicated at collection time to not collect the call stack. In this case the call stack window will not show a call stack but other information about the job's performance during this interval will be available.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in the interval details window. This window will also appear by left-clicking on any bar in the graph.

2.6.5.3 IFS opens

Description: This graph shows a **detailed** look at the rate of IFS file opens by a job throughout its existence during the job watch.

Graph Type: detailed job-specific (vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Contains the number of IFS file opens occuring for each interval per second.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Display Stack	Shows the interval details window and brings focus to the call stack tab. The call stack may not always be available if it was indicated at collection time to not collect the call stack. In this case the call stack window will not show a call stack but other information about the job's performance during this interval will be available.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in the interval details window. This window will also appear by left-clicking on any bar in the graph.

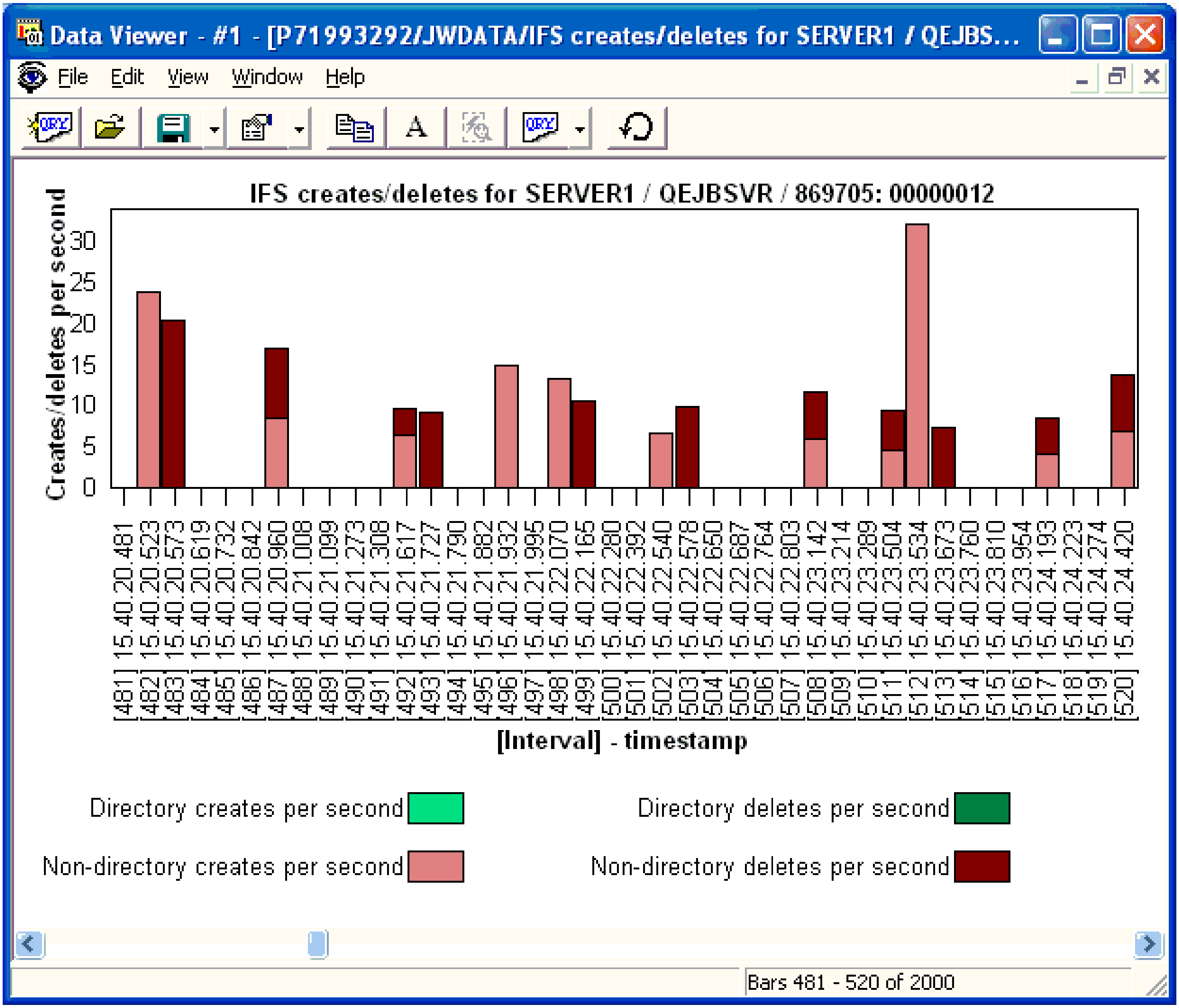
Table of Contents Previous Next

2.6.5.4 IFS creates/deletes

Description: This graph shows a **detailed** look at the rate of hits/misses for the IFS creates and deletes performed by a job throughout its existence during the job watch. The creates and deletes are categorized into two types: directory and non-directory.

Graph Type: detailed job-specific (stacked vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Contains the rates of IFS directory and non-directory creates and deletes per second. Green colors represent directory operations and red colors represent non-directory operations.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	**Description**
Display Stack	Shows the interval details window and brings focus to the call stack tab. The call stack may not always be available if it was indicated at collection time to not collect the call stack. In this case the call stack window will not show a call stack but other information about the job's performance during this interval will be available.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in the interval details window. This window will also appear by left-clicking on any bar in the graph.

Table of Contents | Previous | Next

2.6.6 Java graphs - detailed

The detailed Java graphs provide statistics for Java applications running in a job in a job-specific Job Watch.

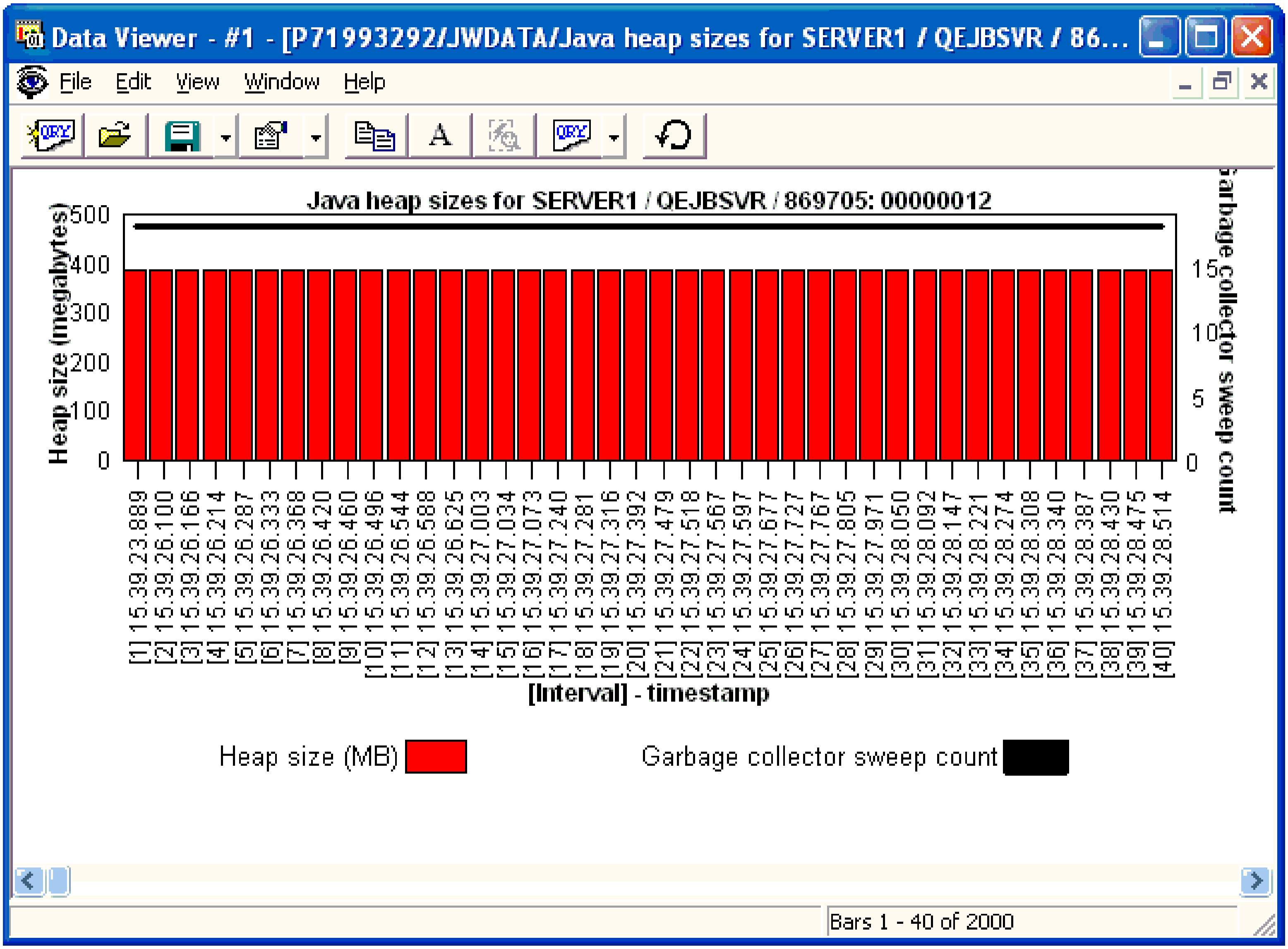
Table of Contents Previous Next

2.6.6.1 Java heap sizes

Description: This graph shows a **detailed** look at the size of the Java heap over time for an application running Java within a job in the job watch.

Graph Type: detailed job-specific (vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Each bar shows the size of the JVM's heap at the moment the interval snapshot was taken.
Second Y-Axis: The line shows the JVM's garbage collector sweep count for each interval.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	Description

Display Stack	Shows the interval details window and brings focus to the call stack tab. The call stack may not always be available if it was indicated at collection time to not collect the call stack. In this case the call stack window will not show a call stack but other information about the job's performance during this interval will be available.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in the interval details window. This window will also appear by left-clicking on any bar in the graph.

Table of Contents | Previous | Next

2.6.7 Other graphs - detailed

The detailed "other" graphs provide graphs that were not covered in the other categories for a job in a job-specific Job Watch.

Table of Contents Previous Next

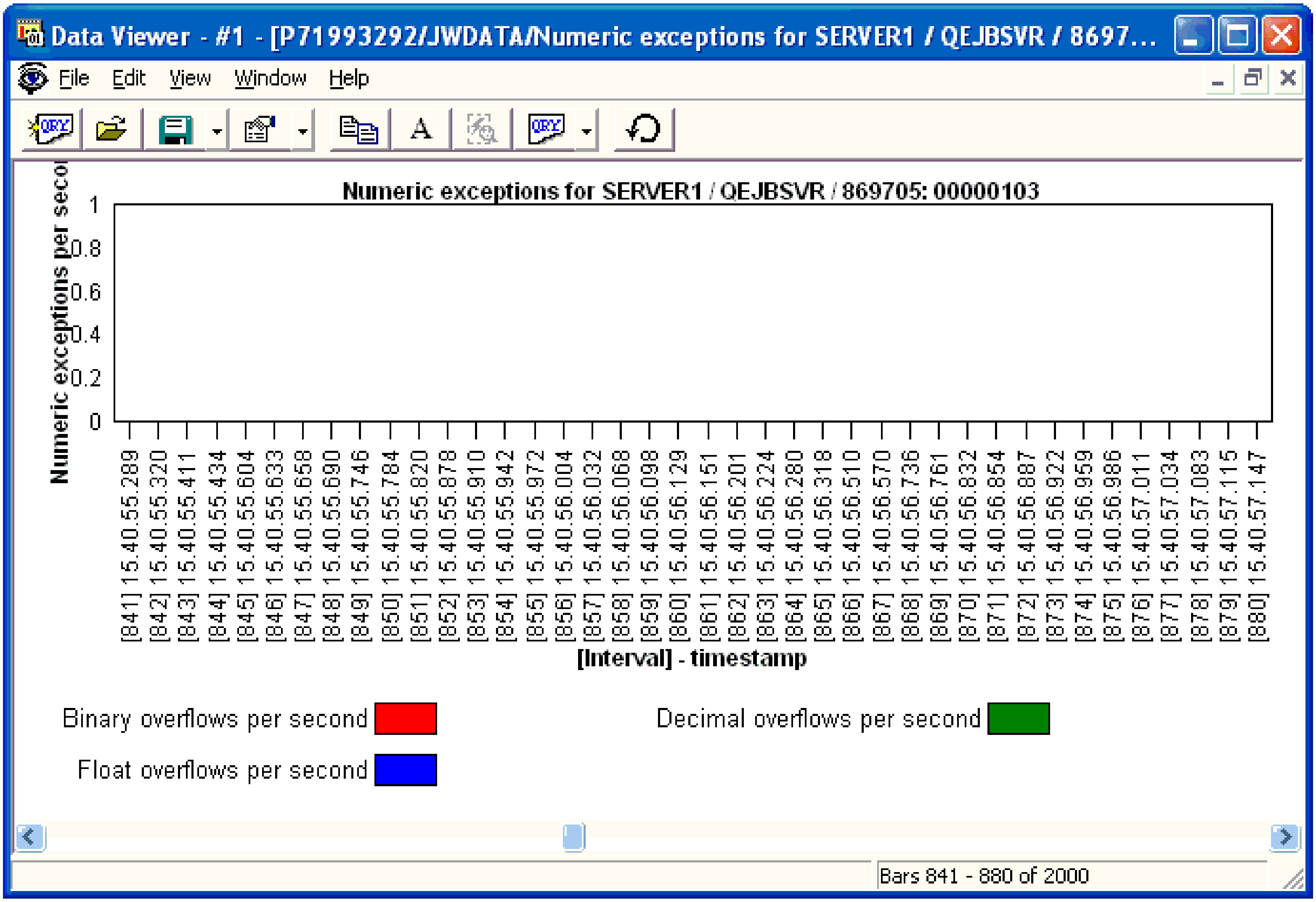
2.6.7.1 Numeric data type exceptions

Description: This graph shows a **detailed** look at the numeric data type exceptions that occurred for a job in the job watch.

This graph can be useful to see where any numeric exceptions are occurring in an application's code (by using this graph in conjunction with the interval details/call stack panel).

Graph Type: detailed job-specific (stacked vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Each bar shows the rates of the following types of exceptions: binary overflows, decimal overflows and float overflows.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	Description

Display Stack	Shows the interval details window and brings focus to the call stack tab. The call stack may not always be available if it was indicated at collection time to not collect the call stack. In this case the call stack window will not show a call stack but other information about the job's performance during this interval will be available.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in the interval details window. This window will also appear by left-clicking on any bar in the graph.

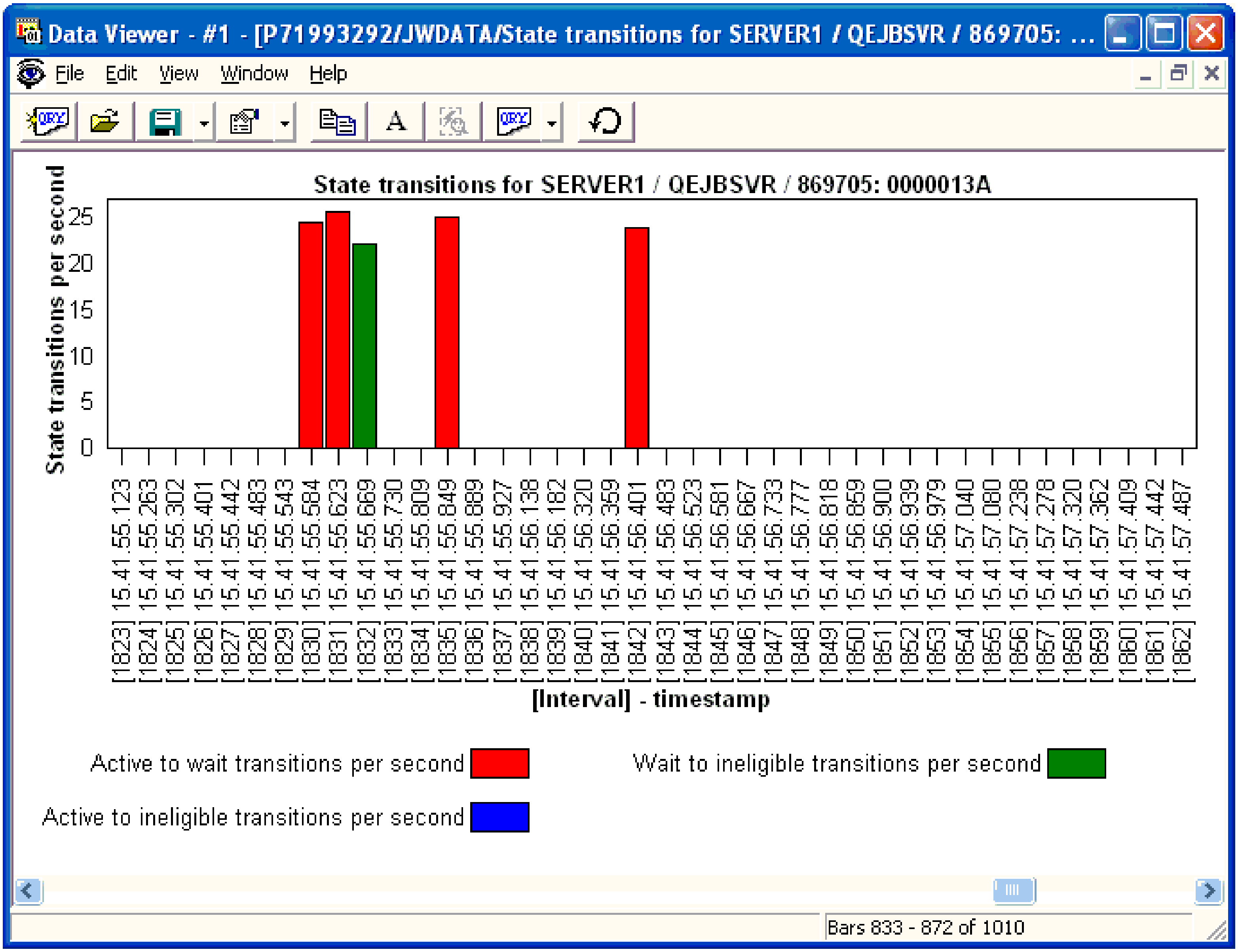
Table of Contents | Previous | Next

2.6.7.2 State transitions

Description: This graph shows a **detailed** look at the job state transitions occurring for a job in the job watch.

Graph Type: detailed job-specific (stacked vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Each bar shows the rates of the following types of job state transitions: active to wait, wait to ineligible, and active to ineligible.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	Description

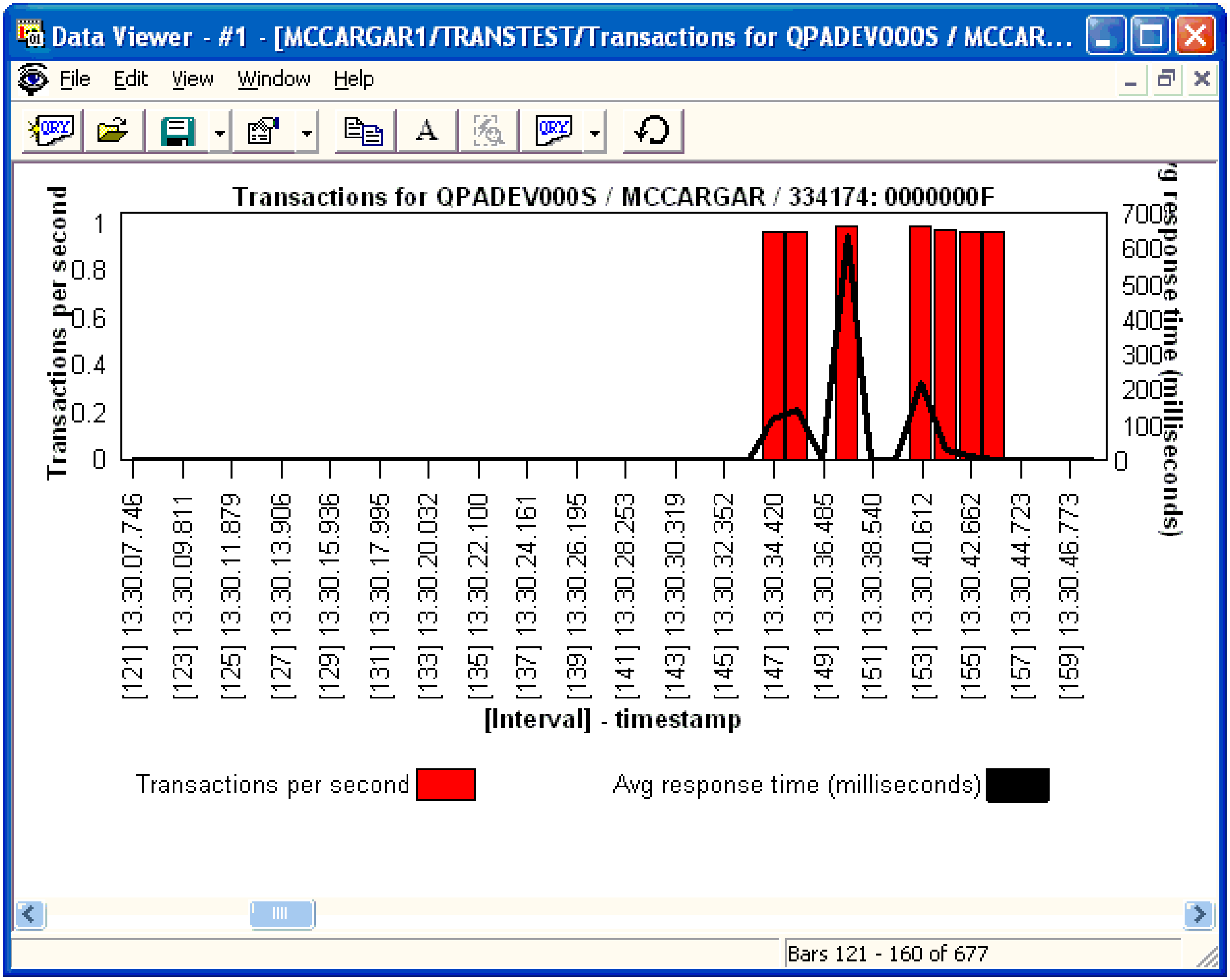
Display Stack	Shows the interval details window and brings focus to the call stack tab. The call stack may not always be available if it was indicated at collection time to not collect the call stack. In this case the call stack window will not show a call stack but other information about the job's performance during this interval will be available.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in the interval details window. This window will also appear by left-clicking on any bar in the graph.

2.6.7.3 Transactions

Description: This graph shows a **detailed** look at the transactions occurring over time for a job in the job watch. The transactions rate (per second) for each interval is shown as each bar in the graph with the average transaction response time on the second Y-axis.

Graph Type: detailed job-specific (stacked vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Each bar shows the transaction per second occuring for each interval.
Second Y-Axis: The line shows the transaction average response time (in milliseconds).

Example:



Right-clicking on a bar in this graph provides the following menu options:

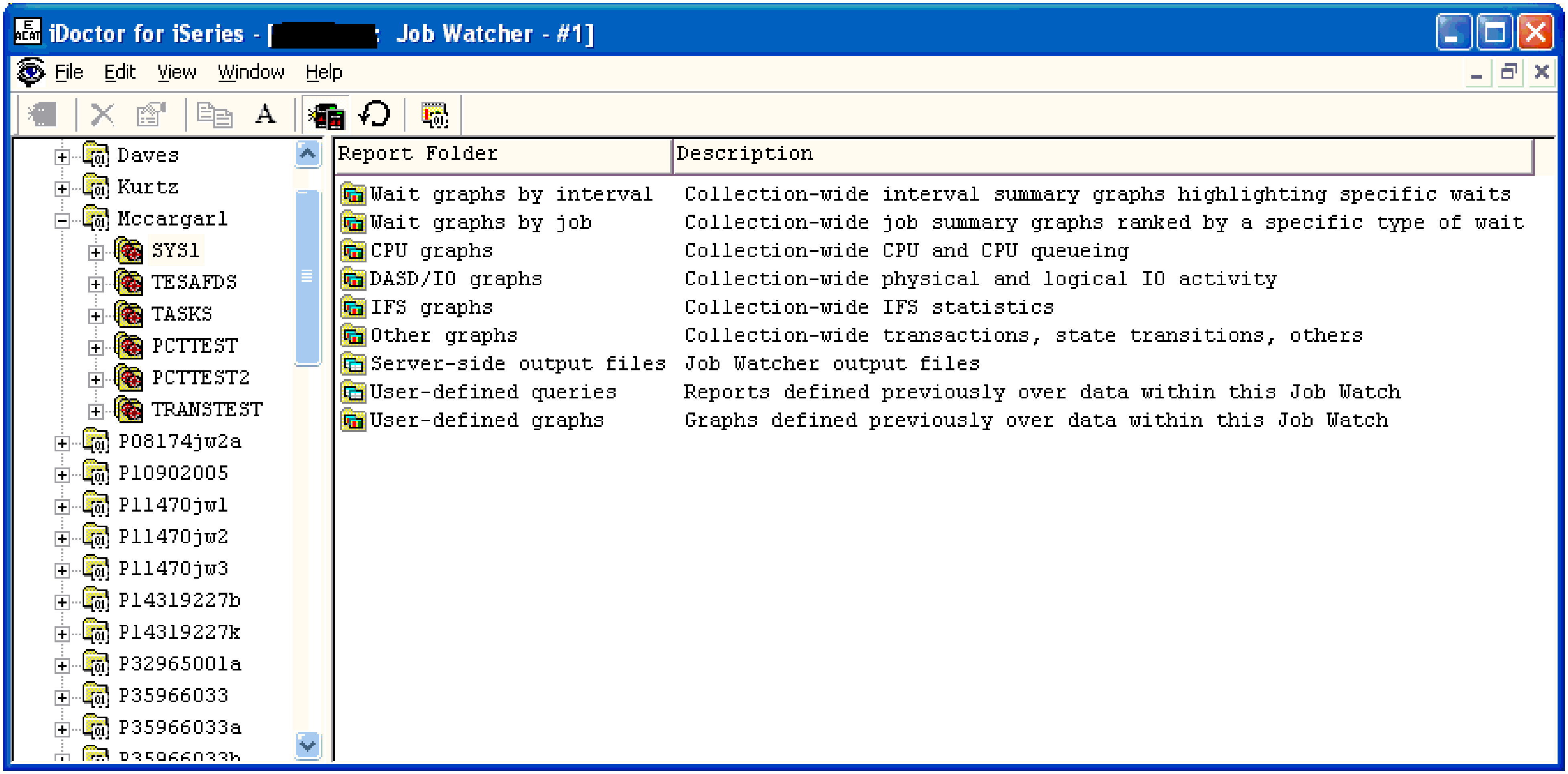
Menu	Description
Display Stack	Shows the interval details window and brings focus to the call stack tab. The call stack may not always be available if it was indicated at collection time to not collect the call stack. In this case the call stack window will not show a call stack but other information about the job's performance during this interval will be available.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in the interval details window. This window will also appear by left-clicking on any bar in the graph.

Table of Contents | Previous | Next

2.7 System-wide Job Watch summary reports

Moving down the tree within each system-wide Job Watch are several report folders. These folders contain many different ways to look at the Job Watcher data. Right-clicking on a job watch provides options to view graphs over many of the same graph category folders listed underneath a system-wide job watch. These popup menu options allow the user to open graphs without needing to expand the graph folders.

An example of the contents of a system-wide Job Watch is shown below:



Report Folder	Description
Wait graphs by interval	Collection-wide interval summary graphs highlighting specific waits
Wait graphs by job	Collection-wide job summary graphs ranked by a specific type of wait
CPU graphs	Collection-wide CPU and CPU queueing
DASD/IO graphs	Collection-wide physical and logical IO activity
IFS graphs	Collection-wide IFS statistics
Other graphs	Collection-wide transactions, state transitions, others
Server-side output files	Job Watcher output files
User-defined queries	Reports defined previously over data within this Job Watch
User-defined graphs	Graphs defined previously over data within this Job Watch

The following table describes the contents of each report folder:

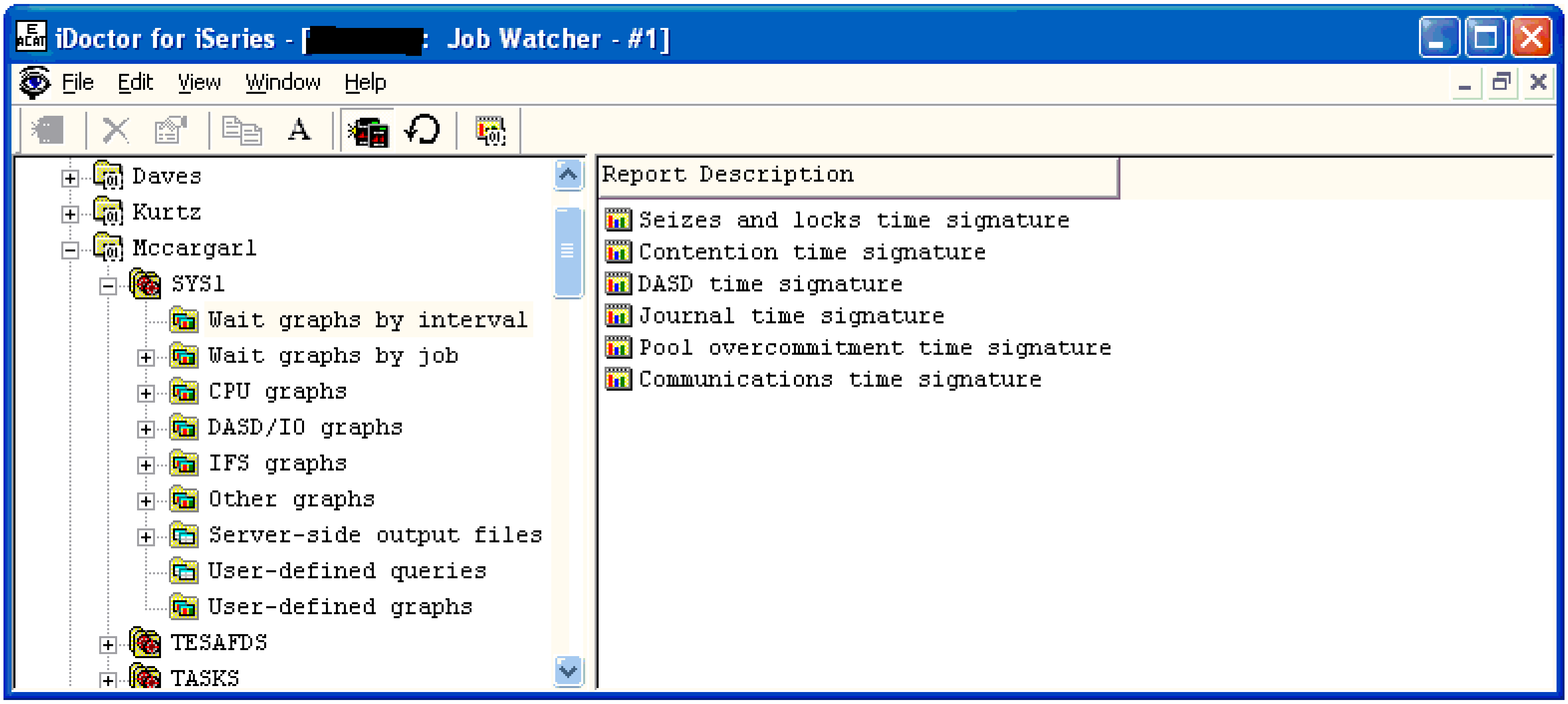
Folder	Description
Wait graphs by interval	Contains system-wide wait summary graphs showing a bar per interval. Each bar is a summary of a particular set of wait times over all jobs for an interval in the job watch. If unsure of where to investigate first, these graphs and the wait graphs by job are a good place to start.
Wait graphs by job	These graphs are wait summary graphs showing a bar per job. The jobs are sorted by the type of wait the graph is showing with the job experience the most time spent in the indicated wait at the top of the graph.
CPU graphs	This folder contains system-wide summary graphs illustrating CPU and CPU queueing usage by interval.
DASD/IO graphs	This folder contains system-wide summary graphs showing IO operations and disk activity by interval.
IFS graphs	This folder contains system-wide summary graphs showing IFS activity by interval.
Java graphs	This folder contains system-wide summary graphs showing Java heap sizes by interval.
Other graphs	This folder contains system-wide summary graphs showing other types of information such as state transitions and transactions. Each bar in the graphs represent an interval of time.
Server-side output files	A complete list of each system-wide Job Watcher output file produced for the collection. These files are queryable through the query definition interface.
User-defined queries	Contains table views over the Job Watcher output files previously defined on the current system. These queries are stored in file QAIDRSQL04 in the current library and library QUSRSYS for system scoped queries. This folder only contains queries for data defined over system-wide Job Watcher output files.
User-defined graphs	Contains graph views over the Job Watcher output files that were previously defined on the current system. These graphs are stored in file QAIDRGPH08 in the current library and library QUSRSYS for system scoped graphs. This folder only contains graphs for data defined over system-wide Job Watcher output files.

Table of Contents | Previous | Next

2.7.1 Wait graphs by interval - summarized

This folder contains a list of graphs which contain summary data over the system-wide job watch relating to waits. These graphs display a bar per interval showing the amount of time spent (across all jobs) in various types of potentially bad waits. High numbers in these graphs do not necessarily indicate a performance problem. A comparison should be made using these graphs from a time when the system is running well, with a time where the system is running poorly in order to determine if there is a problem.

An example of the contents of the wait graphs by interval folder for a system-wide job watch is:



The following table illustrates the menu options available by right-clicking on a graph in the list.

Menu	Description
Open graph(s)	Opens the selected graph(s) into a Data Viewer. If a Data Viewer has already been opened submenus will appear underneath this menu in order to give the option of opening the graph(s) into an already open Data Viewer.

Table of Contents Previous Next

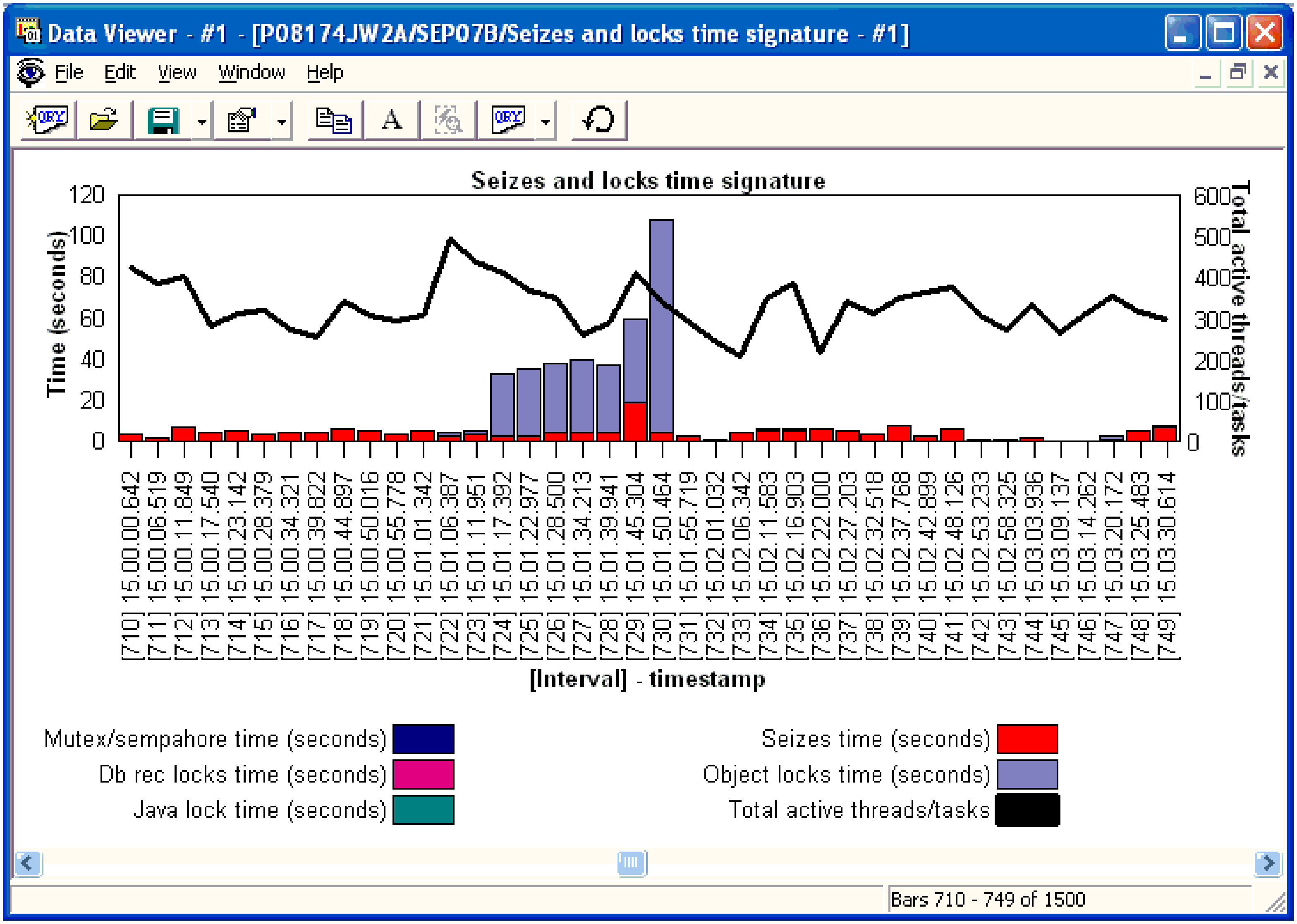
2.7.1.1 Seizes and locks time signature

Description: This graph shows a **summary** of the time all jobs spent in seizes and locks type wait states for each interval within the job watch. The graph makes use of the 32 run/wait buckets used in Job Watcher.

This graph is a good starting point because it provides a high-level overview of when the system experienced a high amount of seizes or object locks. Other types of locks such as java lock time is also shown on this graph.

Graph Type: summarized system-wide (stacked vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Each color represents the total amount of time spent across all threads in a particular type of wait. All times are provided in seconds.
Second Y-Axis: The secondary Y-Axis displays the total number of active threads/tasks on the system for each interval.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	Description

Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.
CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.
DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.
Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.
Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.
Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

Table of Contents | Previous | Next

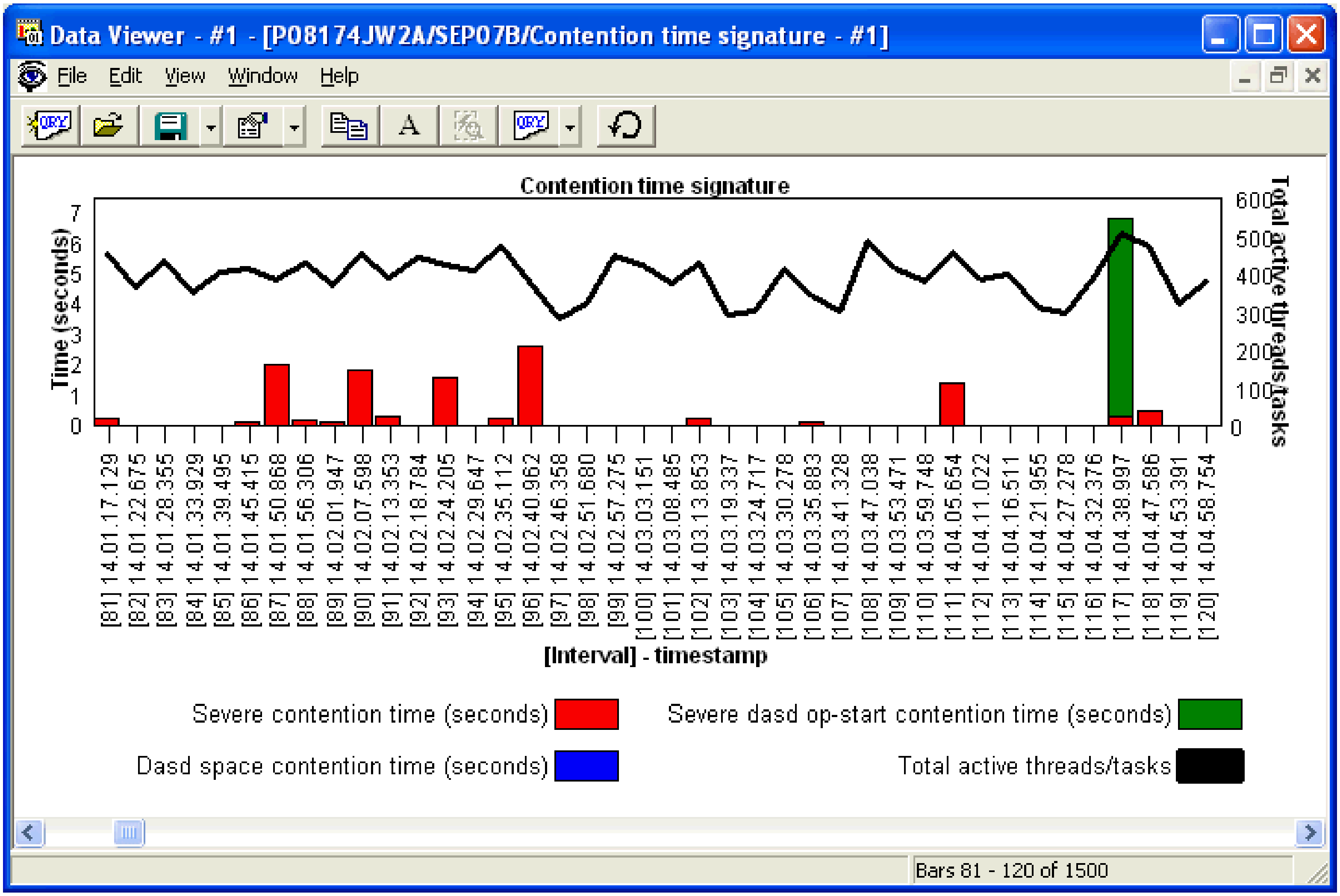
2.7.1.2 Contention time signature

Description: This graph shows a **summary** of severe (or abnormal) contention times for each interval within the job watch. Severe contention time indicates jobs have been unable to perform work and have made several attempts to do work but failed.

If these numbers are high the system is probably running very slowly.

Graph Type: summarized system-wide (stacked vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Each color represents the total amount of time spent across all threads in a particular type of wait. All times are provided in seconds.
Second Y-Axis: The secondary Y-Axis displays the total number of active threads/tasks on the system for each interval.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.
CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.

DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.
Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.
Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.
Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

Table of Contents | Previous | Next

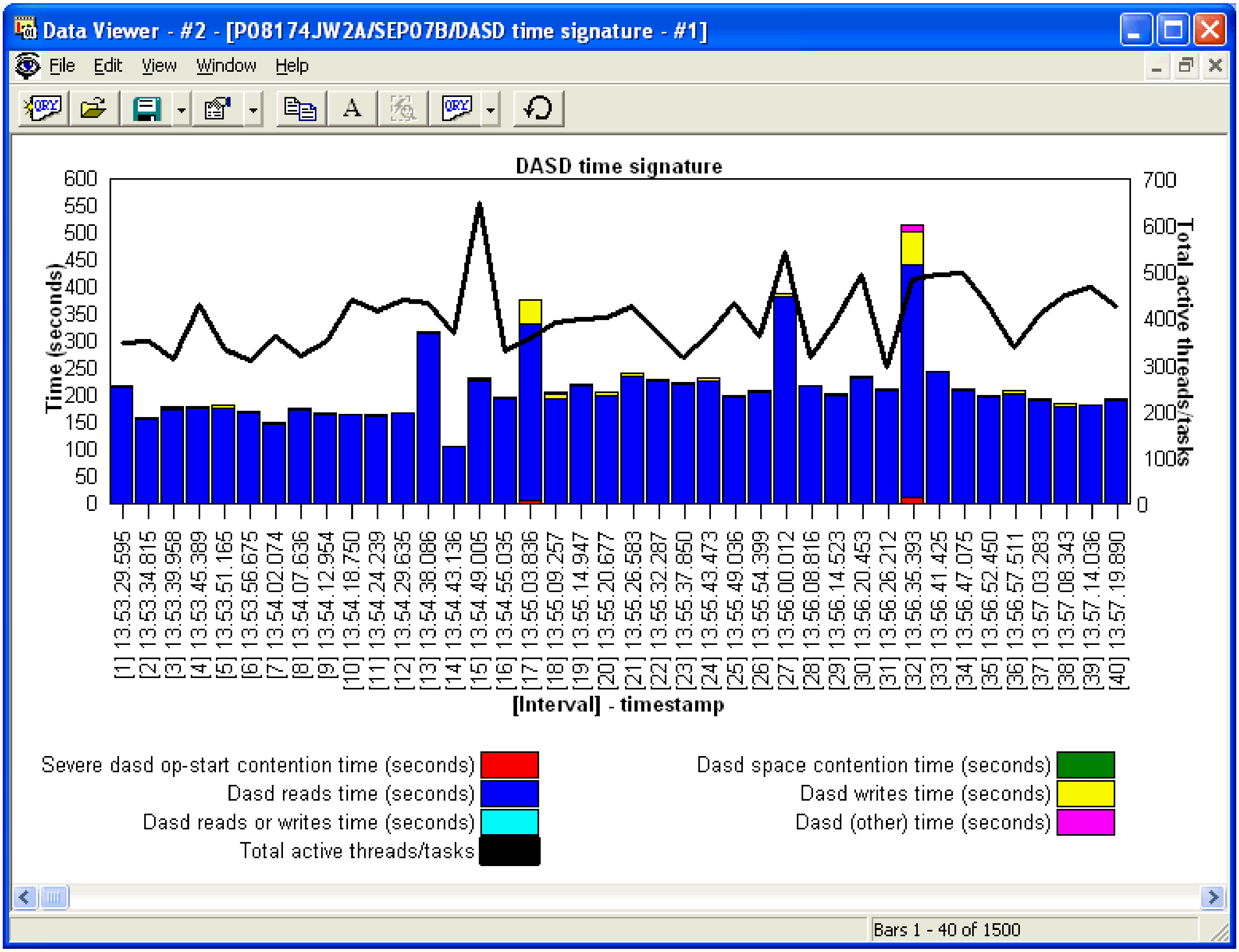
2.7.1.3 DASD time signature

Description: This graph shows a **summary** of the time all jobs spent in DASD waits of various types for each interval within the job watch. The graph makes use of the 32 run/wait buckets used in Job Watcher.

This graph can be used to see a quick summary of when disk use was relatively high or low within the system-wide job watch and the types of operations made.

Graph Type: summarized system-wide (stacked vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Each color represents the total amount of time spent across all threads in a particular type of wait. All times are provided in seconds.
Second Y-Axis: The secondary Y-Axis displays the total number of active threads/tasks on the system for each interval.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	Description

Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.
CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.
DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.
Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.
Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.
Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

Table of Contents | Previous | Next

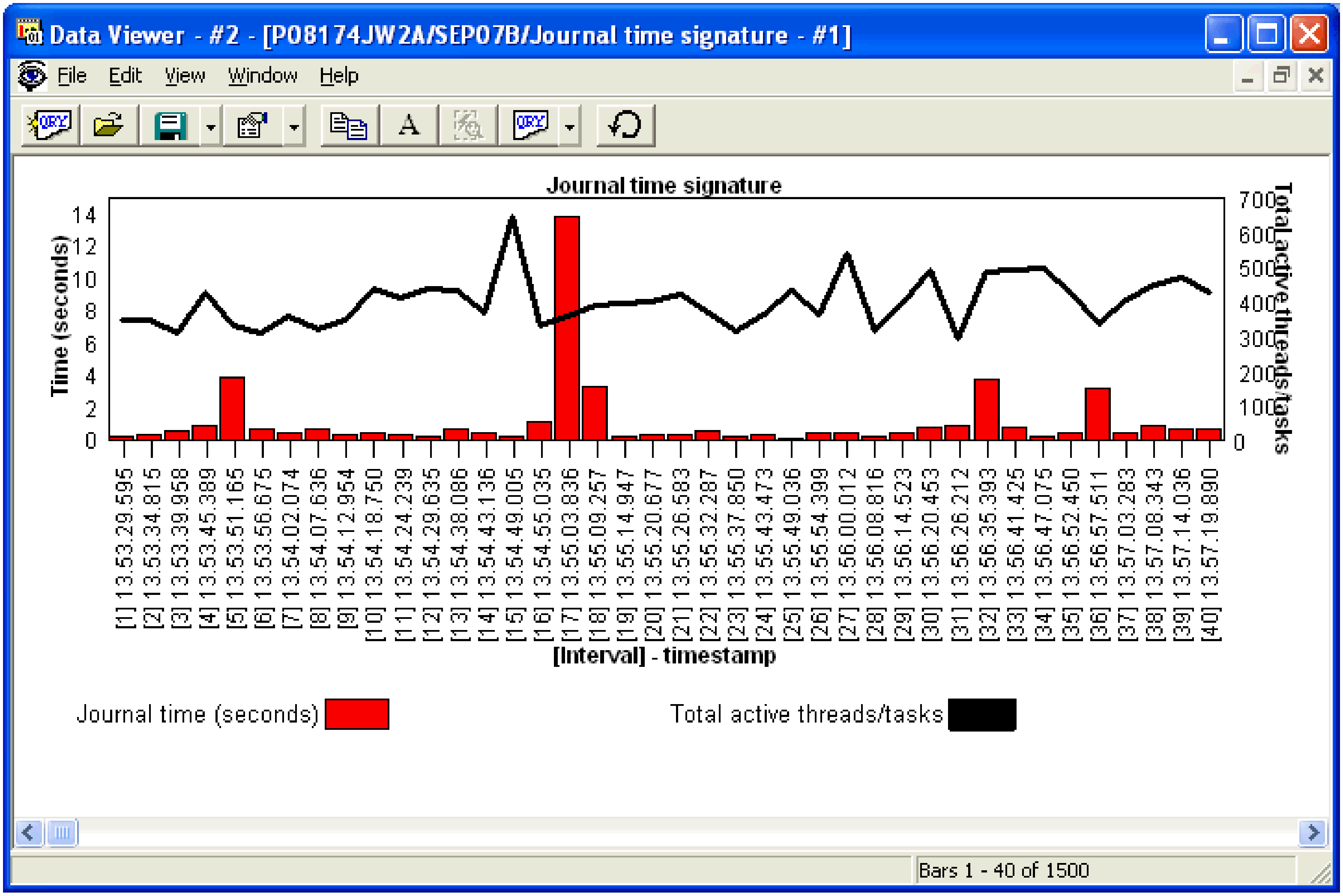
2.7.1.4 Journal time signature

Description: This graph shows a **summary** of the time all jobs spent in journal operations for each interval within the job watch. The graph makes use of the 32 run/wait buckets used in Job Watcher.

This graph can be used to quickly see when journalling was used by the system during the job watch.

Graph Type: summarized job-specific (stacked vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Red bars represents the total amount of time spent in journaling. All times are provided in seconds.
Second Y-Axis: The secondary Y-Axis displays the total number of active threads/tasks on the system for each interval.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.
CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.

DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.
Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.
Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.
Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

Table of Contents | Previous | Next

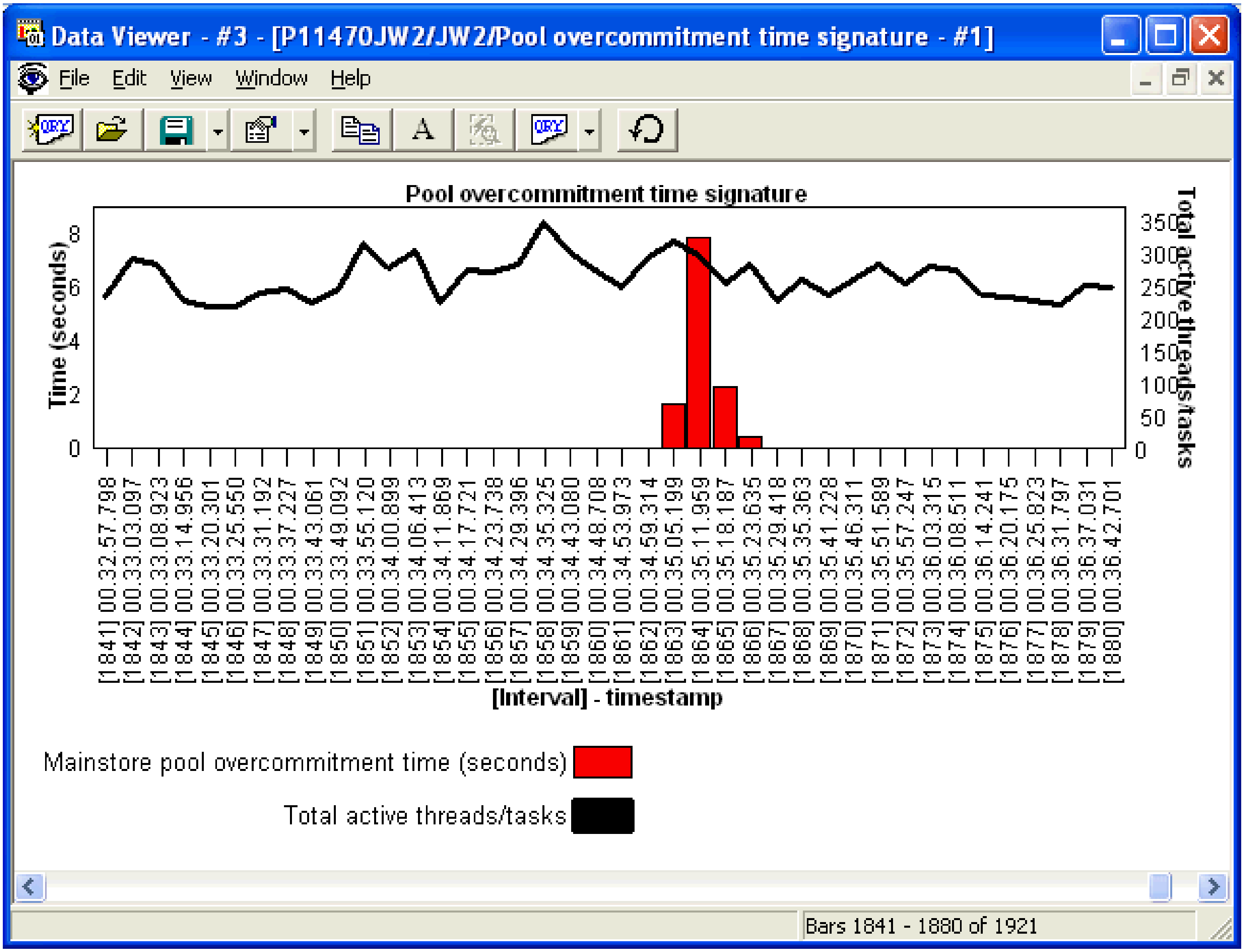
2.7.1.5 Pool overcommitment time signature

Description: This graph shows a **summary** of the time all jobs spent in pool overcommitment wait for each interval within the job watch. The graph makes use of the 32 run/wait buckets used in Job Watcher.

If these numbers are present on a graph it means that during the intervals specified the maintstore pool used by the job is full. Regular operations like DASD reads or page faults are being delayed in order to locate "free" mainstorage pool space to hold the new data.

Graph Type: summarized system-wide (stacked vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Each color represents the total amount of time spent across all threads in mainstorage pool overcommitment time. All times are provided in seconds.
Second Y-Axis: The secondary Y-Axis displays the total number of active threads/tasks on the system for each interval.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	**Description**
Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.
CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.
DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.
Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.
Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.
Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

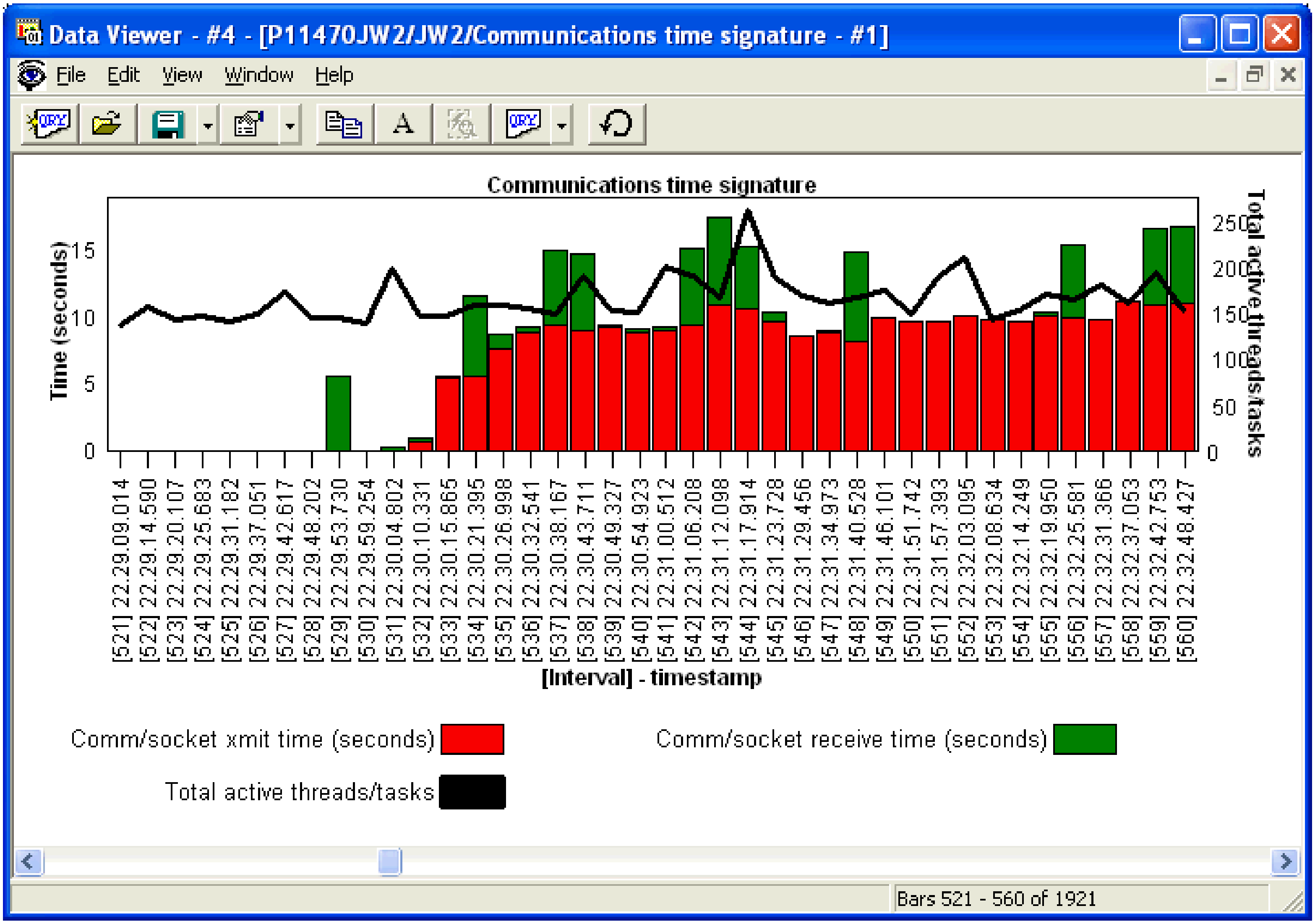
2.7.1.6 Communications time signature

Description: This graph shows a **summary** of the time all jobs spent in communications time for each interval within the job watch. The graph makes use of the 32 run/wait buckets used in Job Watcher.

This graph shows for all jobs the total time spent sending and receiving data over a connection during each interval.

Graph Type: summarized system-wide (stacked vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Each color represents the total amount of time spent across all threads in either comm/socket transmit time or comm/socket receive time. All times are provided in seconds.
Second Y-Axis: The secondary Y-Axis displays the total number of active threads/tasks on the system for each interval.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.

CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.
DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.
Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.
Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.
Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

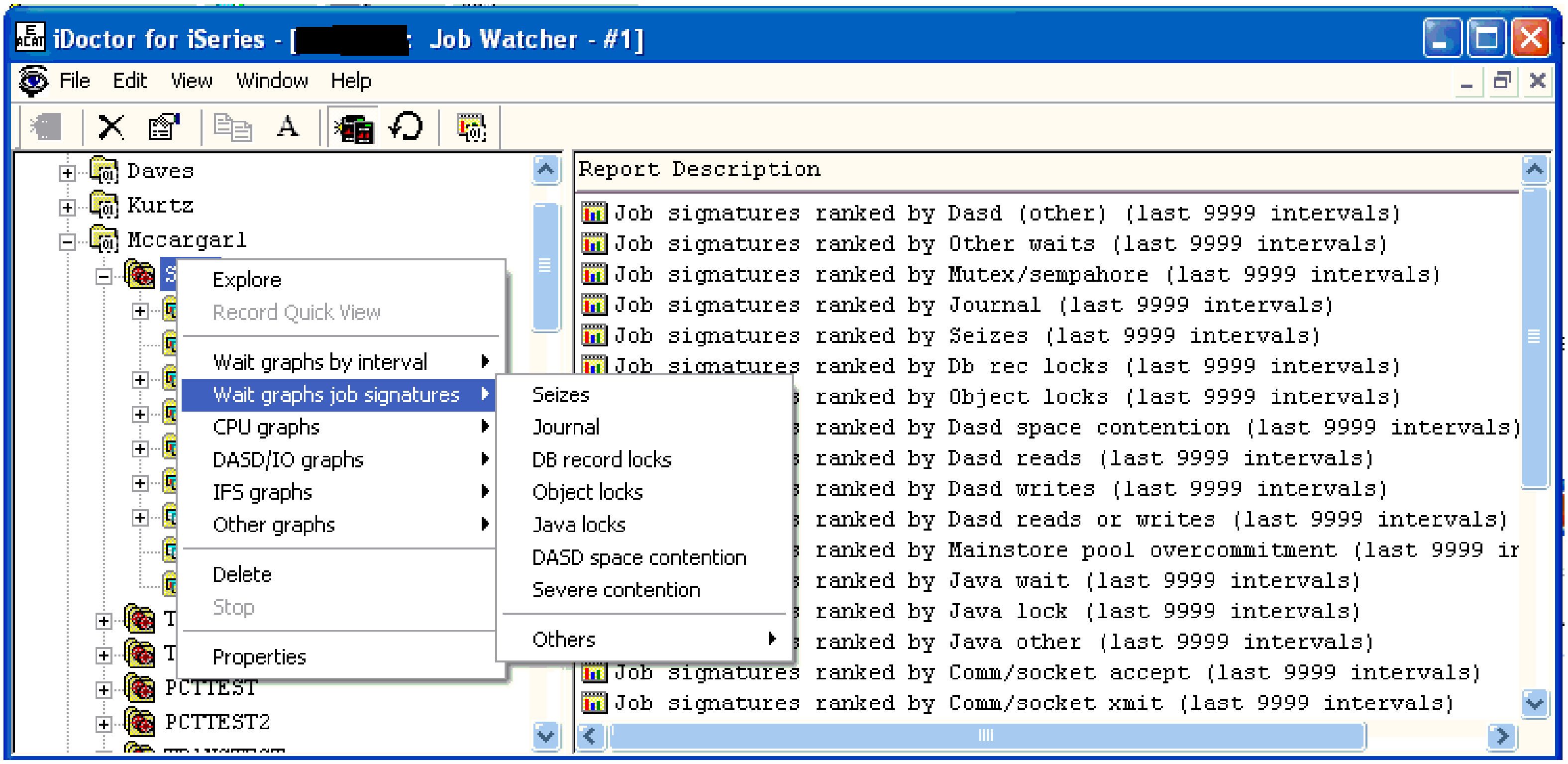
Table of Contents | Previous | Next

2.7.2 Wait graphs by job - summarized

This folder contains a list of graphs which contain job summary data over the system-wide job watch relating to waits. These graphs display a bar per job showing a run/wait time signature sorted by the type of wait indicated in the graph name.

The wait bucket descriptions are shown in the folder listed by bucket number (from 4 - 32), but if the user right-clicks on the job watch and selects the wait graphs by job folder they will see a more manageable list with the buckets that are more commonly associated with a problem listed first (like seizes). The screen capture below shows this expanded wait graphs by job menu along with the complete list of buckets in the list portion of the view.

An example of the contents of the wait graphs by job folder for a system-wide job watch is:



iDoctor for iSeries - [[redacted] : Job Watcher - #1]

Report Description

Job signatures ranked by Dasd (other) (last 9999 intervals)
Job signatures ranked by Other waits (last 9999 intervals)
Job signatures ranked by Mutex/sempahore (last 9999 intervals)
Job signatures ranked by Journal (last 9999 intervals)
Job signatures ranked by Seizes (last 9999 intervals)
Job signatures ranked by Db rec locks (last 9999 intervals)
s ranked by Object locks (last 9999 intervals)
s ranked by Dasd space contention (last 9999 intervals)
s ranked by Dasd reads (last 9999 intervals)
s ranked by Dasd writes (last 9999 intervals)
s ranked by Dasd reads or writes (last 9999 intervals)
s ranked by Mainstore pool overcommitment (last 9999 ir
s ranked by Java wait (last 9999 intervals)
s ranked by Java lock (last 9999 intervals)
s ranked by Java other (last 9999 intervals)
Job signatures ranked by Comm/socket accept (last 9999 intervals)
Job signatures ranked by Comm/socket xmit (last 9999 intervals)

The following table illustrates the menu options available by right-clicking on a graph in the list.

Menu	Description
Open graph(s)	Opens the selected graph(s) into a Data Viewer. If a Data Viewer has already been opened submenus will appear underneath this menu in order to give the option of opening the graph(s) into an already open Data Viewer.

Table of Contents | Previous | Next

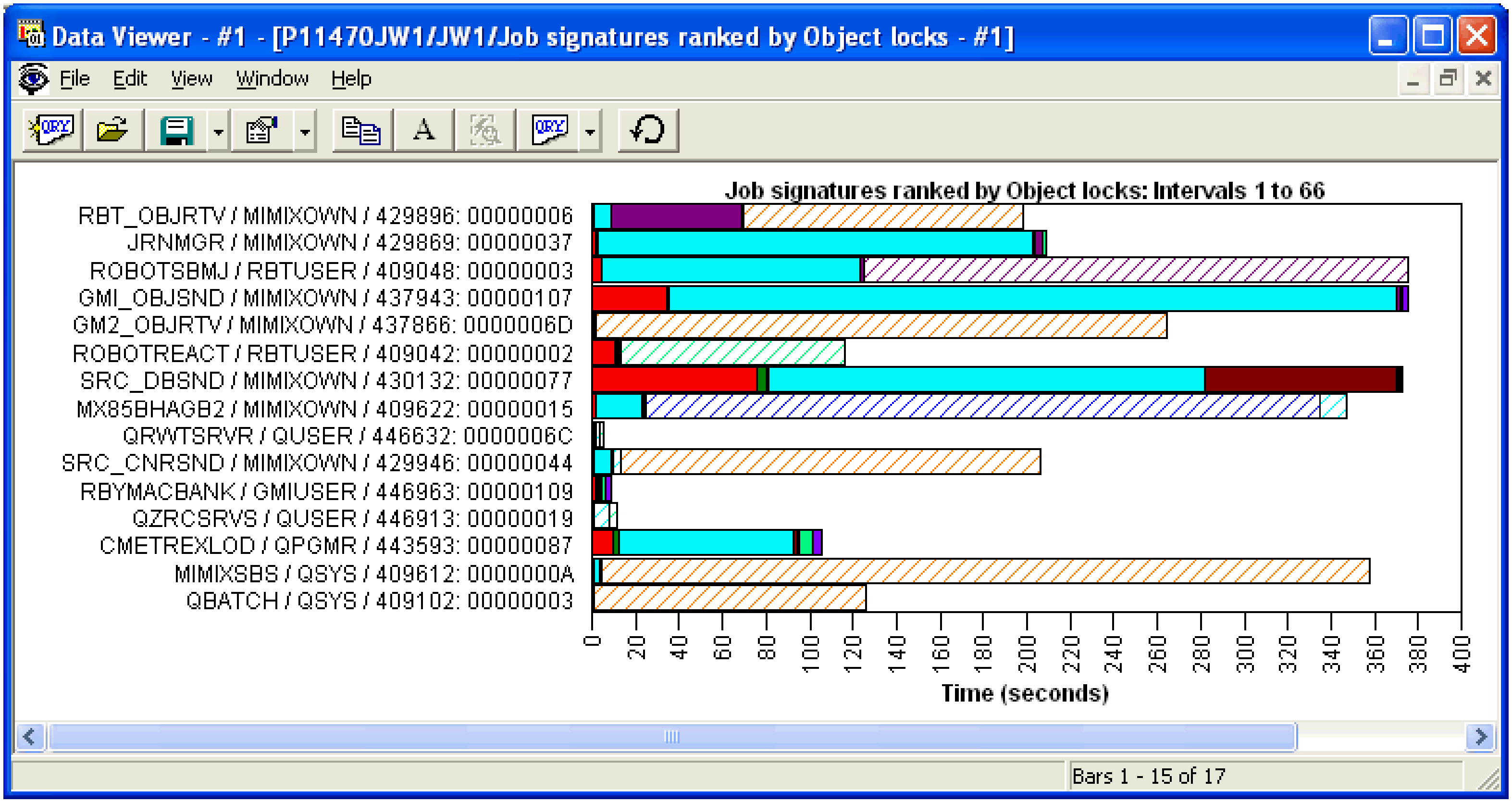
2.7.2.1 Job signatures ranked by <wait type>

Description: This graph shows a **summary** of all jobs on the system ranked by the total time spent in a certain type of wait (like a sezie condition). Each bar in the graph represents a job and the colors in the bar represent various types of wait time the job spent during its existance in the system-wide job watch. The graph makes use of the 32 run/wait buckets used in Job Watcher.

This graph is a good starting point because it provides a high-level overview of which jobs where performing the desired wait type. The total length of the bar indicates the total time the job was collected by the job watch over all intervals. Some jobs may have only have ran for a short while, and other jobs that ran for a long time but were idle during this time will have relatively short bar lengths in the graph (because their data during the idle periods is not collected by Job Watcher).

Graph Type: summarized system-wide (stacked vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Each color represents a type of wait. Place the mouse pointer over a bar to see a description of the wait type. All times are provided in seconds.

Example:


Data Viewer - #1 - [P11470JW1/JW1/Job signatures ranked by Object locks - #1]
File
Edit
View
Window
Help
QRY
A
QRY
Job signatures ranked by Object locks: Intervals 1 to 66
RBT_OBJRTV / MIMIXOWN / 429896: 00000006
JRNMGR / MIMIXOWN / 429869: 00000037
ROBOTSBMJ / RBTUSER / 409048: 00000003
GMI_OBJSND / MIMIXOWN / 437943: 00000107
GM2_OBJRTV / MIMIXOWN / 437866: 0000006D
ROBOTREACT / RBTUSER / 409042: 00000002
SRC_DBSND / MIMIXOWN / 430132: 00000077
MX85BHAGB2 / MIMIXOWN / 409622: 00000015
QRWTSRVR / QUSER / 446632: 0000006C
SRC_CNRSND / MIMIXOWN / 429946: 00000044
RBYMACBANK / GMIUSER / 446963: 00000109
QZRCSRVS / QUSER / 446913: 00000019
CMETREXLOD / QPGMR / 443593: 00000087
MIMIXSBS / QSYS / 409612: 0000000A
QBATCH / QSYS / 409102: 00000003
0
20
40
60
80
100
120
140
160
180
200
220
240
260
280
300
320
340
360
380
400
Time (seconds)
Bars 1 - 15 of 17

Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.
CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.
DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.
Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.

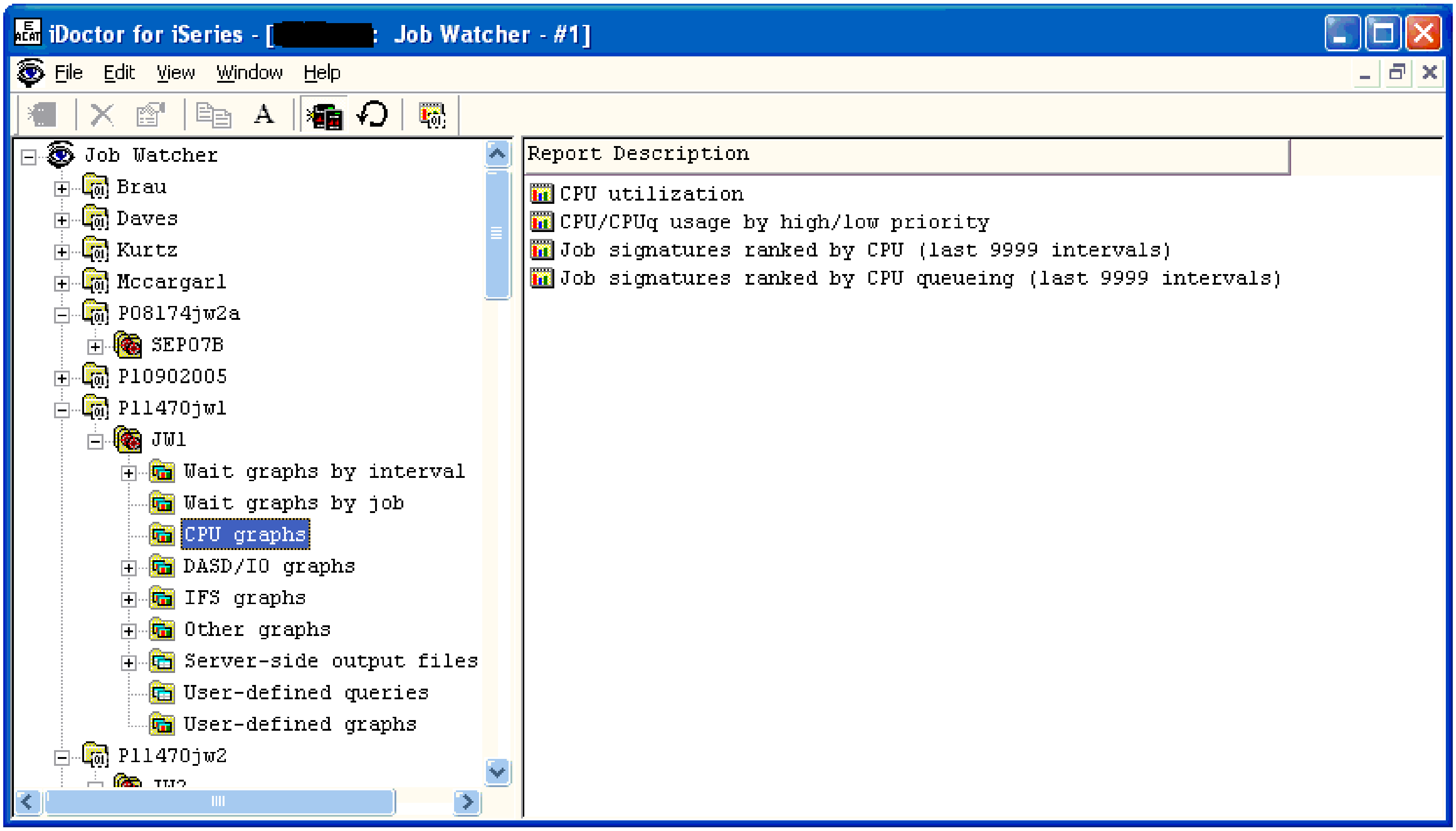
Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.
Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

Table of Contents | Previous | Next

2.7.3 CPU graphs - summarized

This folder contains a list of graphs which contain summary data over the system-wide job watch showing CPU and CPU queueing times. Graphs for CPU utilization and job summaries ranked by those using the most CPU are also available in this folder.

An example of the contents of the CPU graphs folder for a system-wide job watch is:



iDoctor for iSeries - [[redacted]: Job Watcher - #1]

- Job Watcher
  - Brau
  - Daves
  - Kurtz
  - Mccargarl
  - P08174jw2a
    - SEP07B
  - P10902005
  - P11470jw1
    - JW1
      - Wait graphs by interval
      - Wait graphs by job
      - CPU graphs
      - DASD/IO graphs
      - IFS graphs
      - Other graphs
      - Server-side output files
      - User-defined queries
      - User-defined graphs
  - P11470jw2
    - JW2

Report Description
CPU utilization
CPU/CPUq usage by high/low priority
Job signatures ranked by CPU (last 9999 intervals)
Job signatures ranked by CPU queueing (last 9999 intervals)

The following table illustrates the menu options available by right-clicking on a graph in the list.

Menu	Description
Open graph(s)	Opens the selected graph(s) into a Data Viewer. If a Data Viewer has already been opened submenus will appear underneath this menu in order to give the option of opening the graph(s) into an already open Data Viewer.

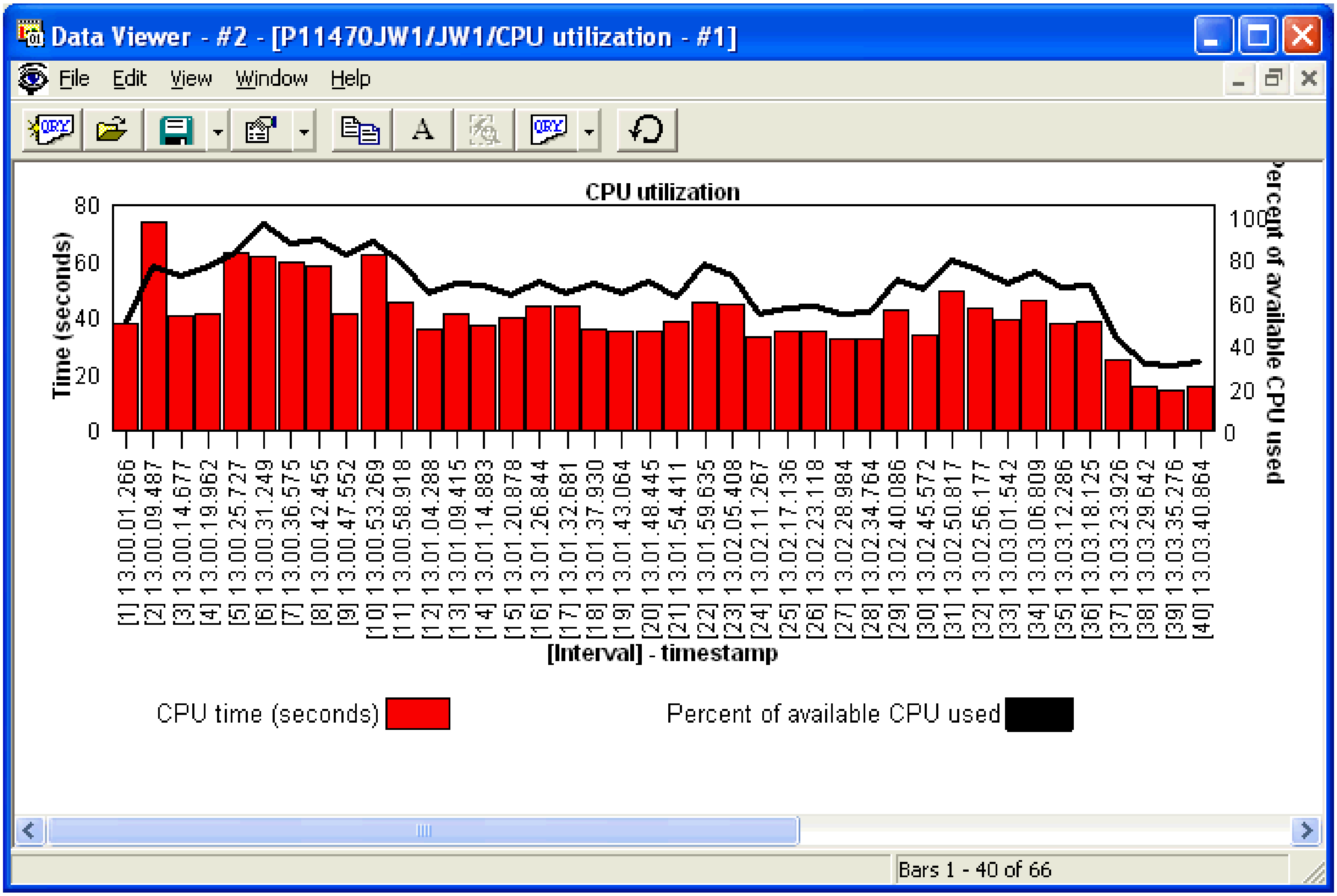
Table of Contents | Previous | Next

2.7.3.1 CPU utilization

Description: This graph shows a **summary** of the CPU time and CPU utilization for each interval in the system-wide job watch.. The bars in the graph show CPU time and the line indicates CPU utilization. CPU utilization shows a percentage of the total available CPU that was used during each interval.

Graph Type: summarized system-wide (vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: The bars show CPU time. All times are provided in seconds.
Second Y-Axis: The line shows CPU utilization as a percentage of total available CPU used each interval.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.
CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.

DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.
Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.
Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.
Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

Table of Contents Previous Next

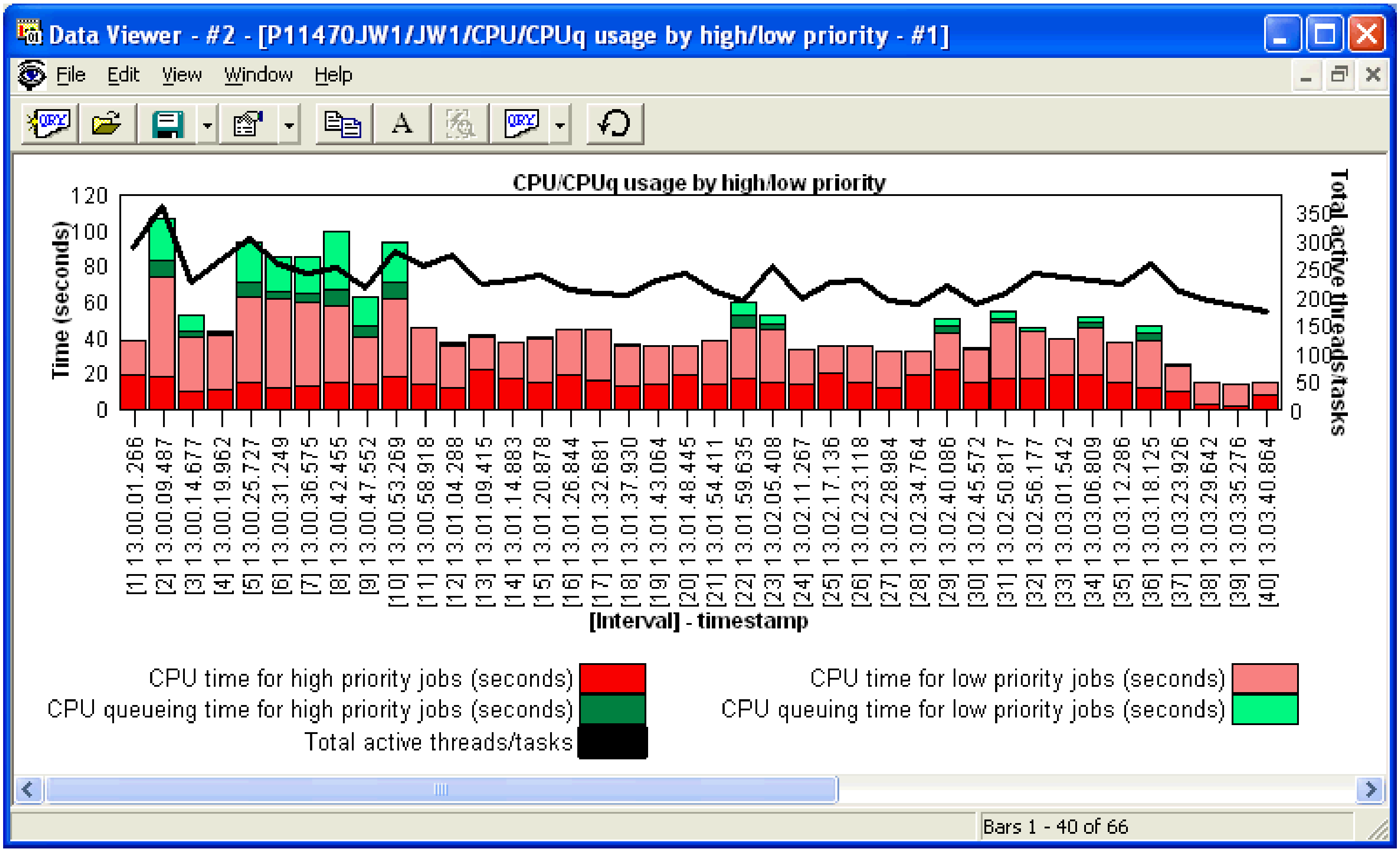
2.7.3.2 CPU/CPUq usage by high/low priority

Description: This graph shows a **summary** of the CPU and CPU queueing times used by high priority (<= 29) and low priority (> 29) jobs during the system-wide job watch. The red bars show CPU times and the green bars show CPU queueing times.

The priority values are configurable on the field selection page of the query definition interface by modifying the SQL expression for fields CPUHIGH, CPULOW, CPUQHIGH and CPUQLOW once the graph has been opened. The priority field in the data is a LIC priority which is 140 higher than XPF priority. For this reason 169 instead of 29 is used to make the comparison in the SQL expressions in the query used behind the graph.

Graph Type: summarized system-wide (stacked vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: The bars show CPU and CPU queueing time. All times are provided in seconds.
Second Y-Axis: The secondary Y-Axis displays the total number of active threads/tasks on the system for each interval.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.
CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.
DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.

Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.
Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.
Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

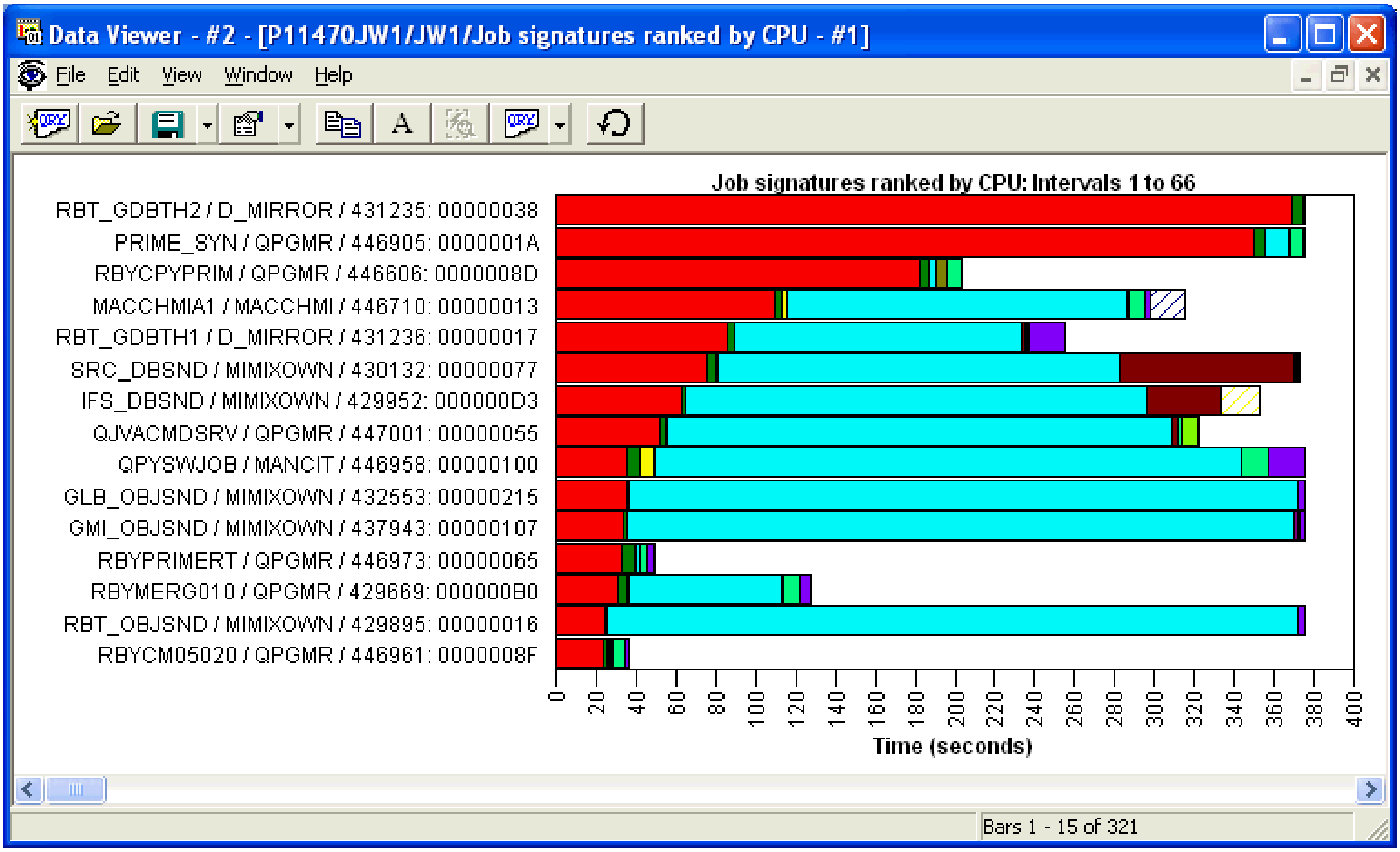
Table of Contents Previous Next

2.7.3.3 Job signatures ranked by CPU

Description: This graph shows a **summary** of the jobs in the system-wide job watch that used the most CPU. Each bar in the graph shows a job's run/wait time signature

Graph Type: summarized system-wide (vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Red in the bars indicate CPU time. Other colors show times for other types of waits. Place the mouse pointer over a bar to see a description of the wait type. All times are provided in seconds.

Example:


Data Viewer - #2 - [P11470JW1/JW1/Job signatures ranked by CPU - #1]
File Edit View Window Help
QRY
A
QRY
Job signatures ranked by CPU: Intervals 1 to 66
RBT_GDBTH2 / D_MIRROR / 431235: 00000038
PRIME_SYN / QPGMR / 446905: 0000001A
RBYCPYPRIM / QPGMR / 446606: 0000008D
MACCHMIA1 / MACCHMI / 446710: 00000013
RBT_GDBTH1 / D_MIRROR / 431236: 00000017
SRC_DBSND / MIMIXOWN / 430132: 00000077
IFS_DBSND / MIMIXOWN / 429952: 000000D3
QJVACMDSRV / QPGMR / 447001: 00000055
QPYSWJOB / MANCIT / 446958: 00000100
GLB_OBJSND / MIMIXOWN / 432553: 00000215
GMI_OBJSND / MIMIXOWN / 437943: 00000107
RBYPRIMERT / QPGMR / 446973: 00000065
RBYMERG010 / QPGMR / 429669: 000000B0
RBT_OBJSND / MIMIXOWN / 429895: 00000016
RBYCM05020 / QPGMR / 446961: 0000008F
0
20
40
60
80
100
120
140
160
180
200
220
240
260
280
300
320
340
360
380
400
Time (seconds)
Bars 1 - 15 of 321

Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.
CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.
DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.
Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.
Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.

Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

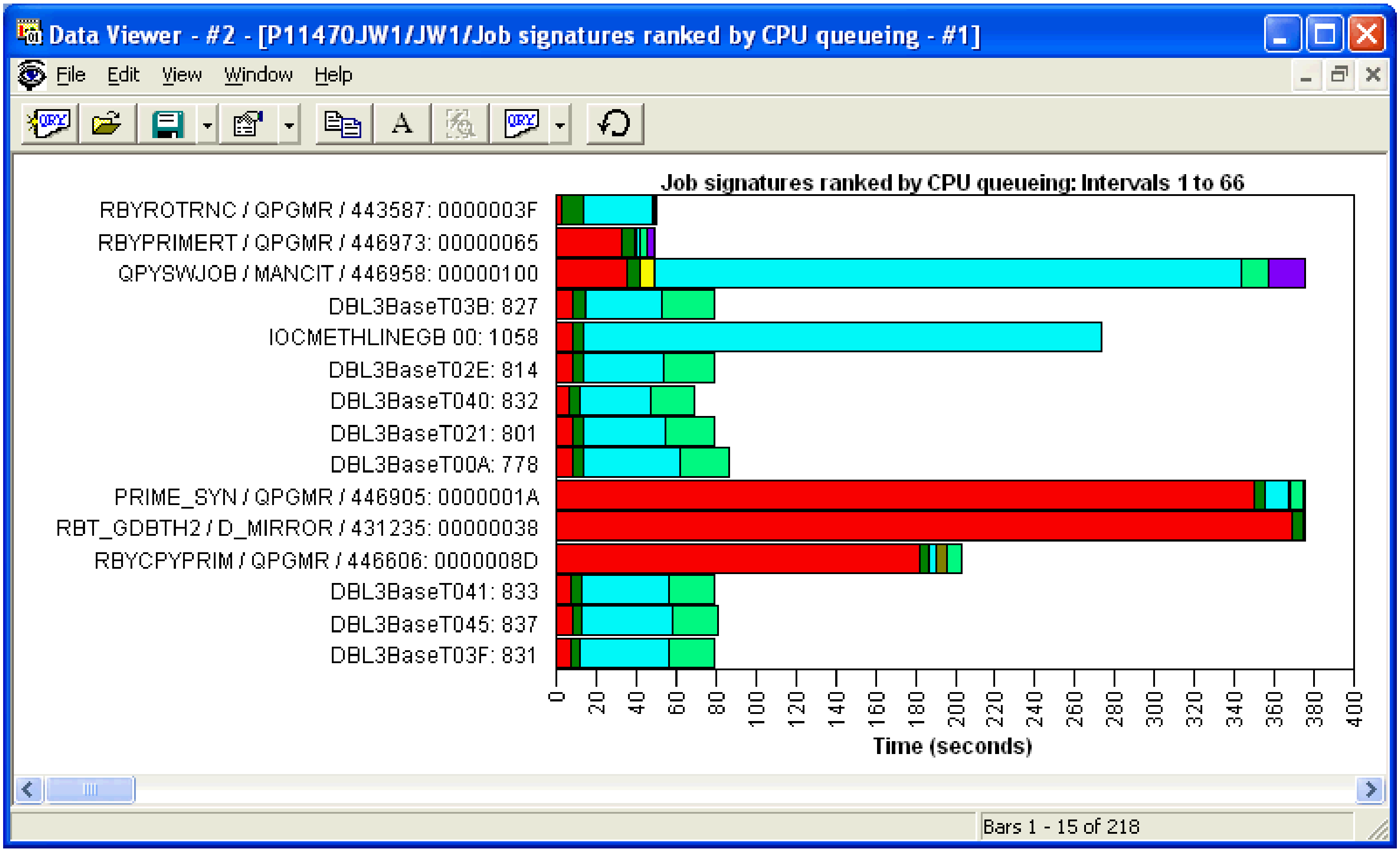
Table of Contents Previous Next

2.7.3.4 Job signatures ranked by CPU queueing

Description: This graph shows a **summary** of the jobs in the system-wide job watch that used the most CPU queueing. Each bar in the graph shows a job's run/wait time signature

Graph Type: summarized system-wide (vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Green in the bars indicate CPU queueing time. Other colors show times for other types of waits. Place the mouse pointer over a bar to see a description of the wait type. All times are provided in seconds.

Example:


Data Viewer - #2 - [P11470JW1/JW1/Job signatures ranked by CPU queueing - #1]
File Edit View Window Help
Job signatures ranked by CPU queueing: Intervals 1 to 66
RBYROTRNC / QPGMR / 443587: 0000003F
RBYPRIMERT / QPGMR / 446973: 00000065
QPYSWJOB / MANCIT / 446958: 00000100
DBL3BaseT03B: 827
IOCMETHLINEGB 00: 1058
DBL3BaseT02E: 814
DBL3BaseT040: 832
DBL3BaseT021: 801
DBL3BaseT00A: 778
PRIME_SYN / QPGMR / 446905: 0000001A
RBT_GDBTH2 / D_MIRROR / 431235: 00000038
RBYCPYPRIM / QPGMR / 446606: 0000008D
DBL3BaseT041: 833
DBL3BaseT045: 837
DBL3BaseT03F: 831
0
20
40
60
80
100
120
140
160
180
200
220
240
260
280
300
320
340
360
380
400
Time (seconds)
Bars 1 - 15 of 218

Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.
CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.
DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.
Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.
Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.
Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.

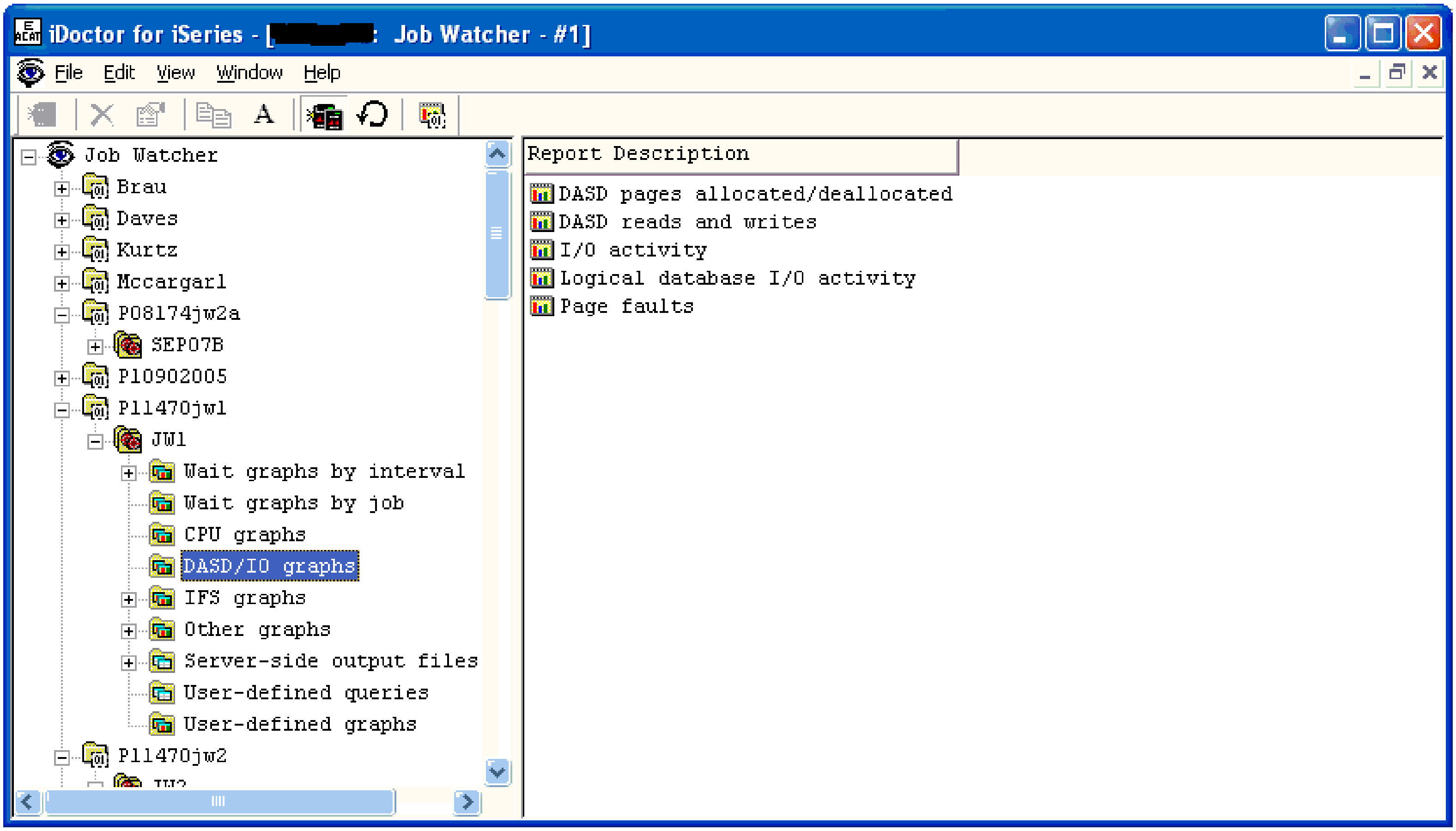
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

Table of Contents Previous Next

2.7.4 DASD/IO graphs - summarized

This folder contains a list of graphs which contain summary data over the system-wide job watch relating to DASD/IO usage. These graphs display a bar per interval showing a type of physical IO or disk activity.

An example of the contents of the DASD/IO graphs folder for a system-wide job watch is:



The following table illustrates the menu options available by right-clicking on a graph in the list.

Menu	Description
Open graph(s)	Opens the selected graph(s) into a Data Viewer. If a Data Viewer has already been opened submenus will appear underneath this menu in order to give the option of opening the graph(s) into an already open Data Viewer.

Table of Contents | Previous | Next

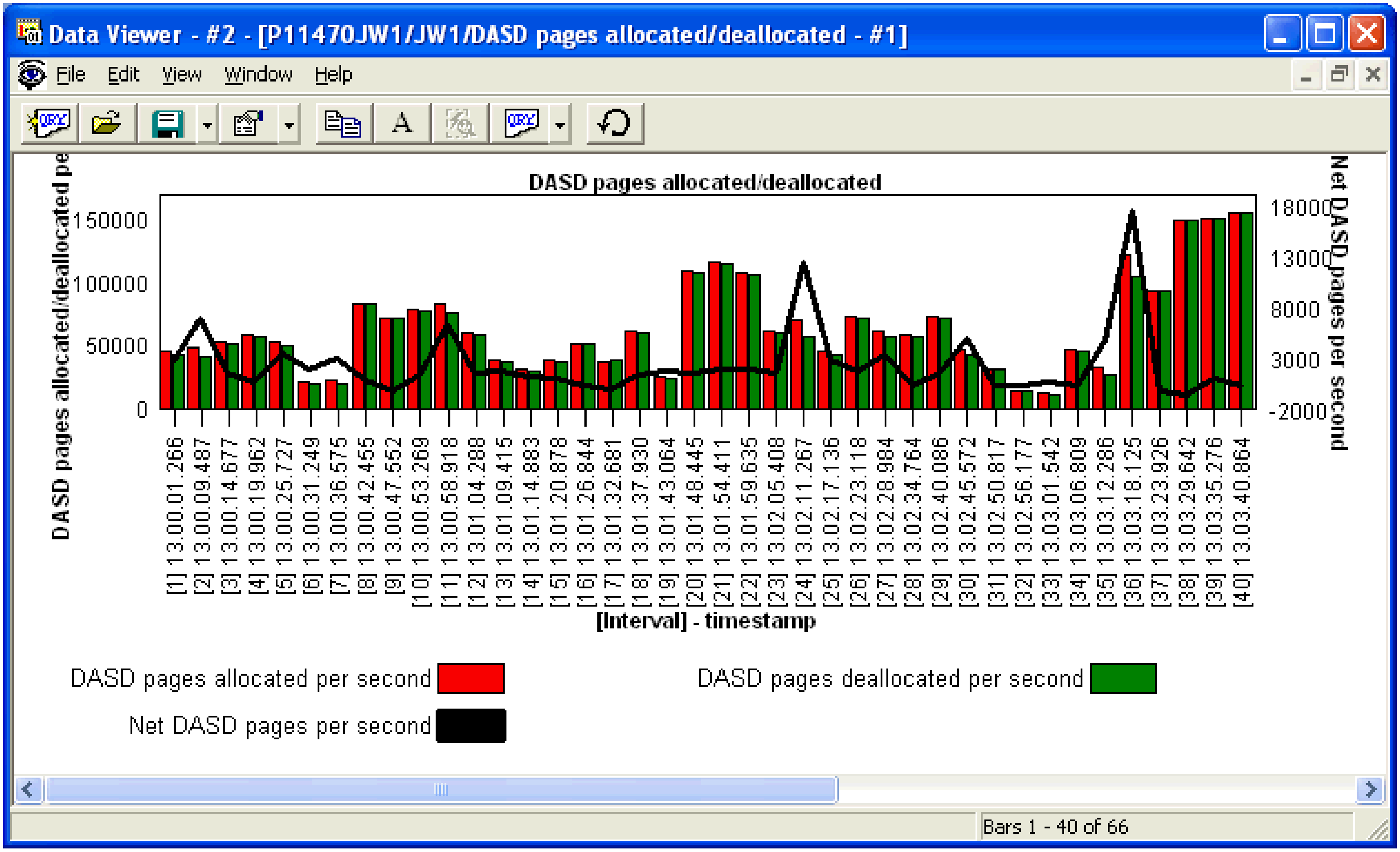
2.7.4.1 DASD pages allocated/deallocated

Description: This graph shows a **summary** of the rates of DASD pages allocated/deallocated each interval in the system-wide job watch.. The bars in the graph show rates of pages allocated and deallocated.

This graph can be useful to see when high DASD space allocations or deallocations were made on the system.

Graph Type: summarized system-wide (vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: The bars show the rate of 4K DASD pages allocated and deallocated. Red bars show allocations and green bars show deallocations. All rates are 4k DASD pages per second.
Second Y-Axis: The line shows the net rate of pages allocated/deallocated each interval.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.
CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.
DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.
Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.

Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.
Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

Table of Contents | Previous | Next

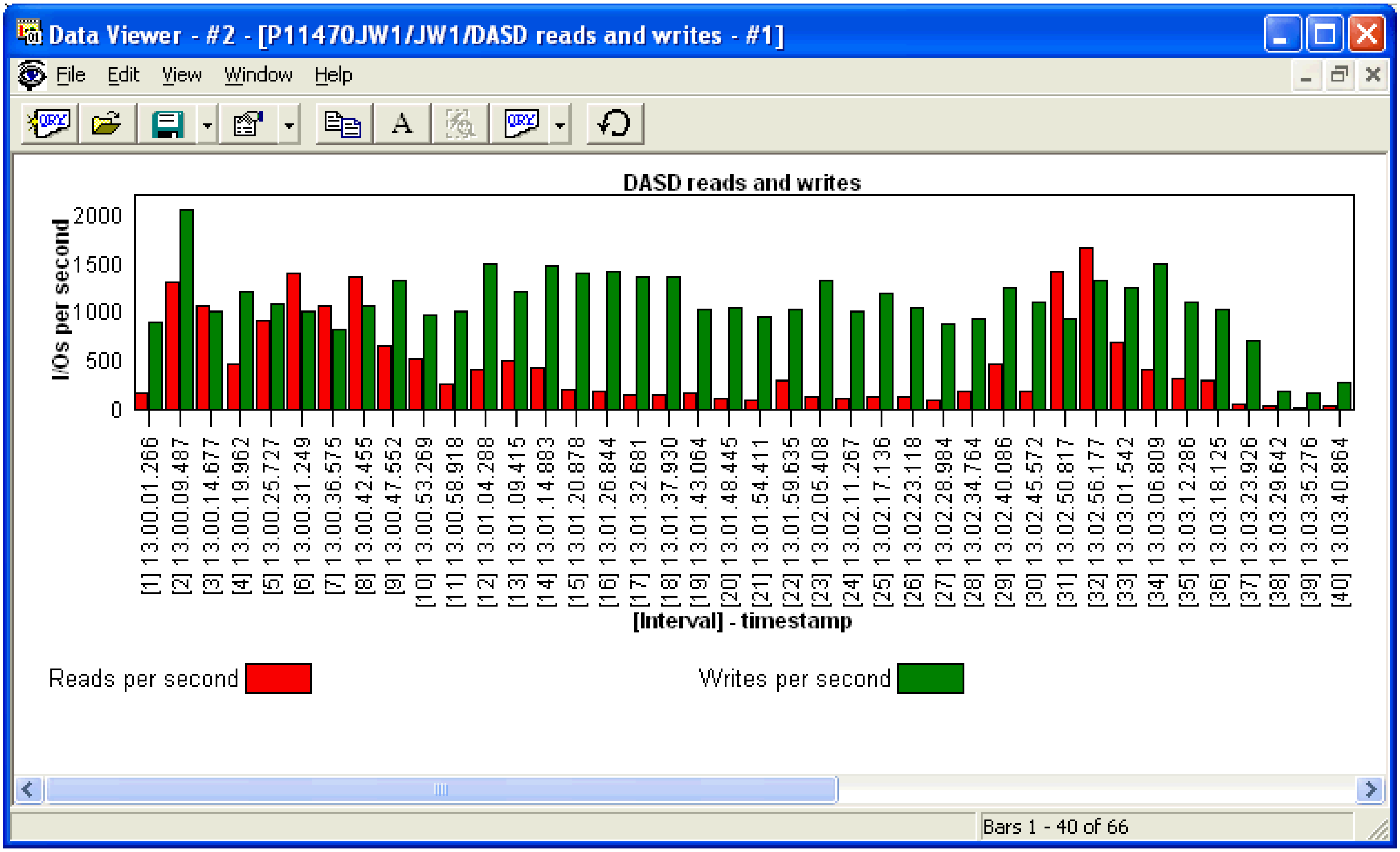
2.7.4.2 DASD reads and writes

Description: This graph shows a **summary** of the rates of DASD reads and writes that occurred each interval in the system-wide job watch. The red bars show reads per second and the green bars show writes per second.

This graph can be useful to see when high disk reads and writes were made on the system.

Graph Type: summarized system-wide (vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: The bars show rates of disk reads and writes per second.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.
CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.
DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.
Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.
Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.

Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

Table of Contents | Previous | Next

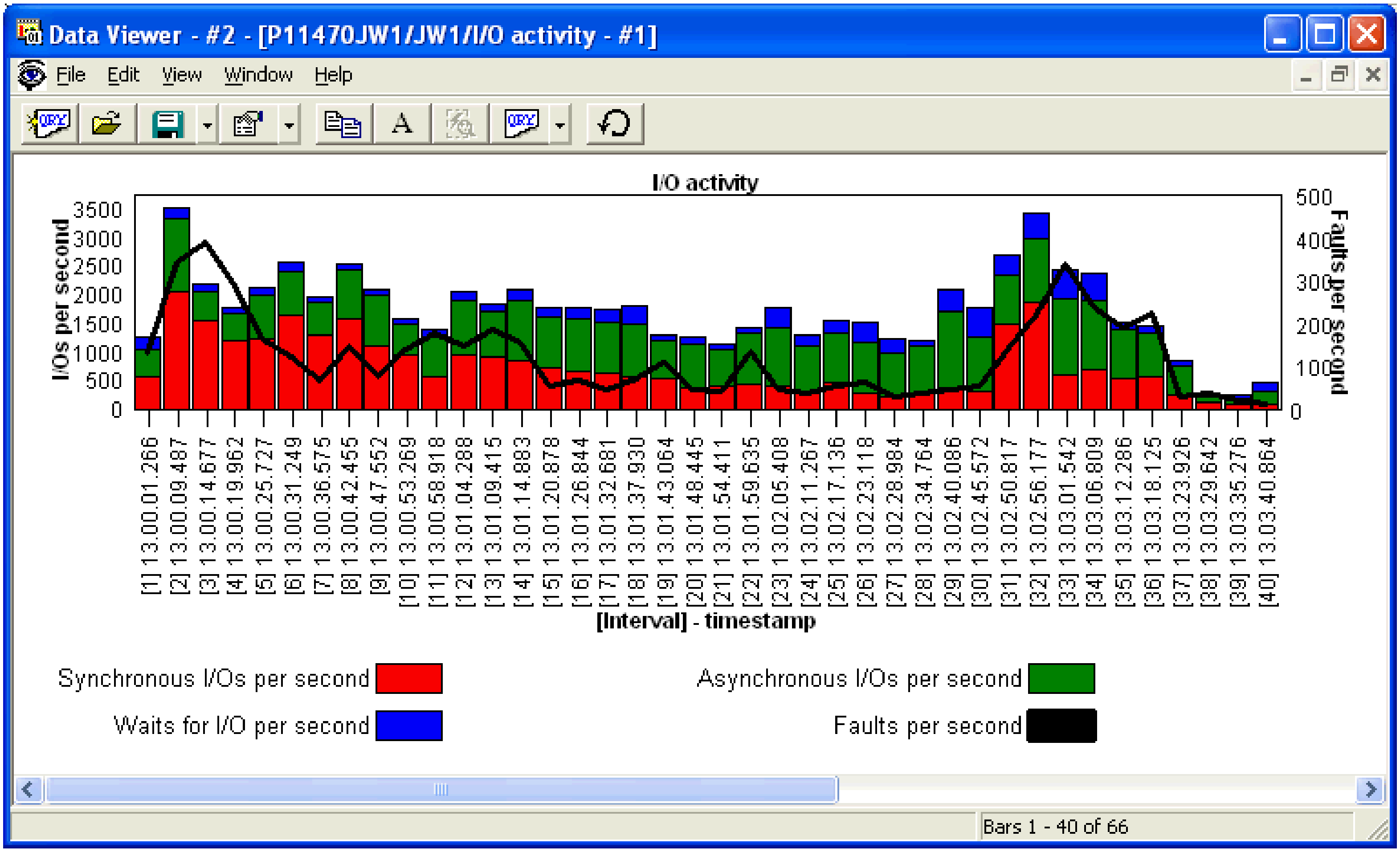
2.7.4.3 I/O activity

Description: This graph shows a **summary** of I/O activity for each interval in the system-wide job watch.. The bars in the graph show rates of synchronous IO, asynchronous IO and waits for IO.

This graph can be useful for comparing synchronous vs asynchronous IO on a system-wide basis.

Graph Type: summarized system-wide (stacked vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: The bars show various types of IO activity in I/Os per second.
Second Y-Axis: The line shows the page faults per second for each interval.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.
CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.
DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.
Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.

Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.
Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

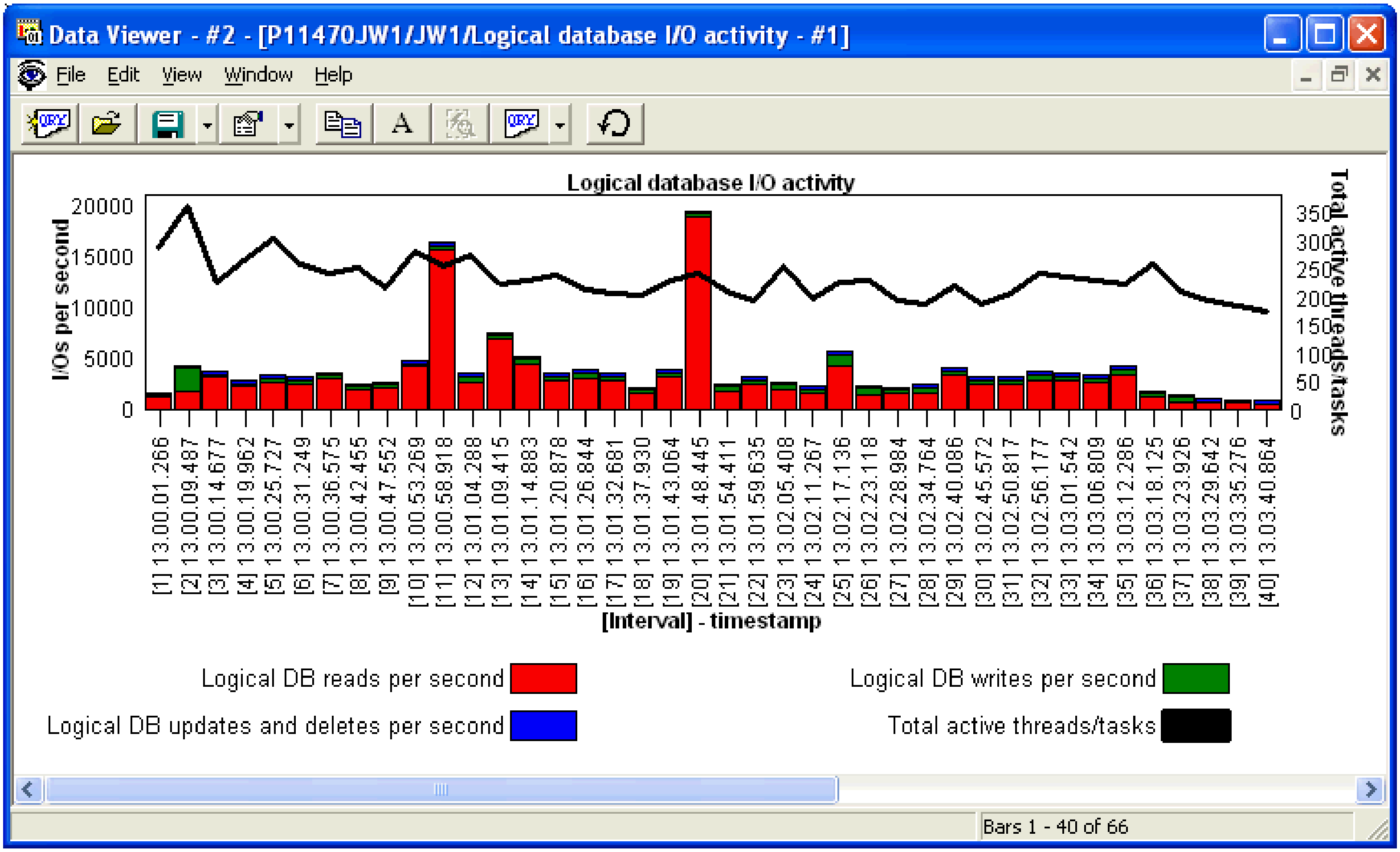
Table of Contents | Previous | Next

2.7.4.4 Logical database I/O activity

Description: This graph shows a **summary** of the rates of logical database I/O including reads, writes, updates and deletes for the system-wide job watch. The bars in the graph show rates of LDIO activity per second.

Graph Type: summarized system-wide (stacked vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: The bars show the LDIO activity per second.
Second Y-Axis: The secondary Y-Axis displays the total number of active threads/tasks on the system for each interval.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.
CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.
DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.
Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.
Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.

Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

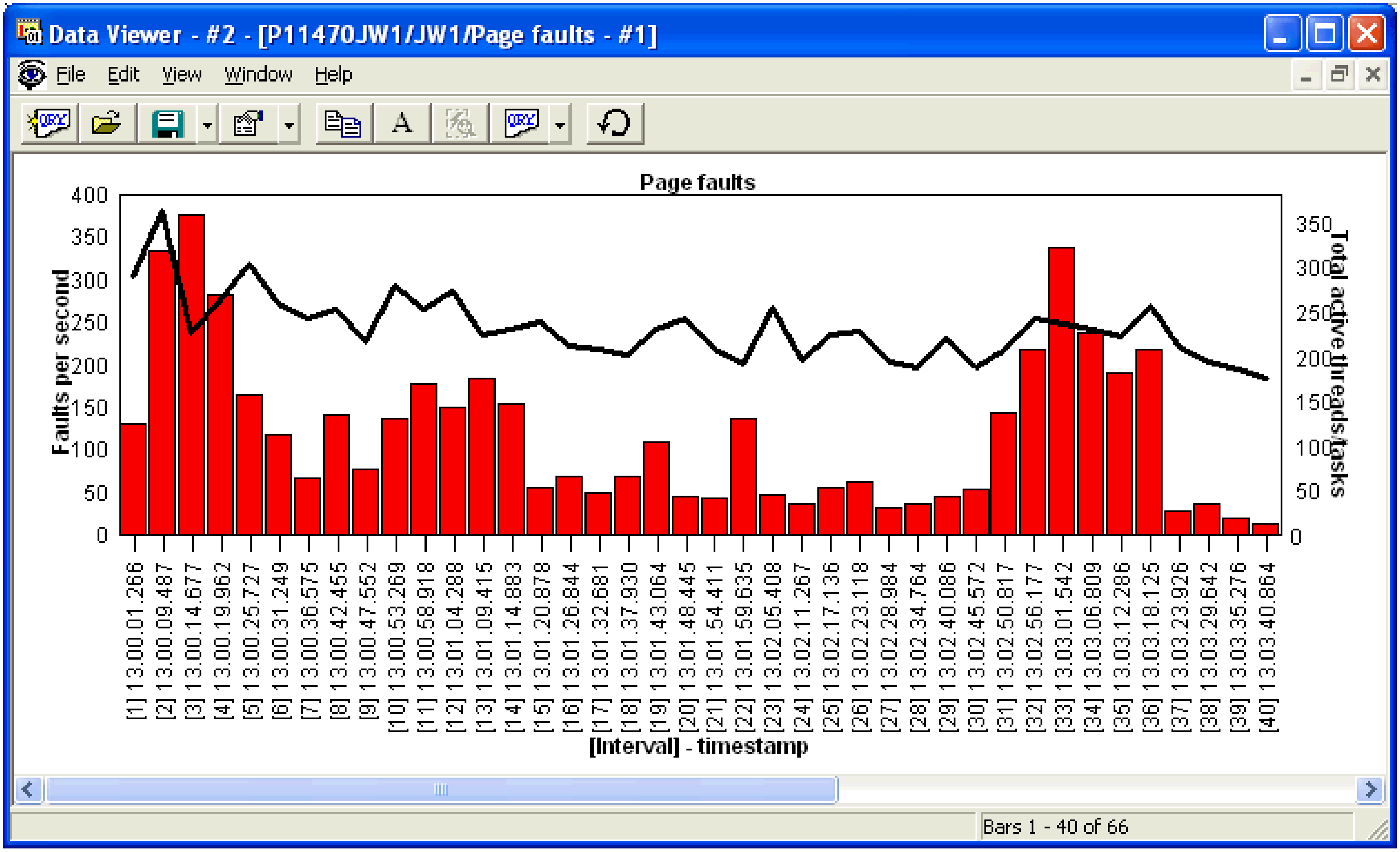
Table of Contents | Previous | Next

2.7.4.5 Page faults

Description: This graph shows a **summary** of the page faults per second for each interval in the system-wide job watch..

Graph Type: summarized system-wide (vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: The bars show the rates of page faulting per second.
Second Y-Axis: The secondary Y-Axis displays the total number of active threads/tasks on the system for each interval.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.
CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.
DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.
Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.
Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.
Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.

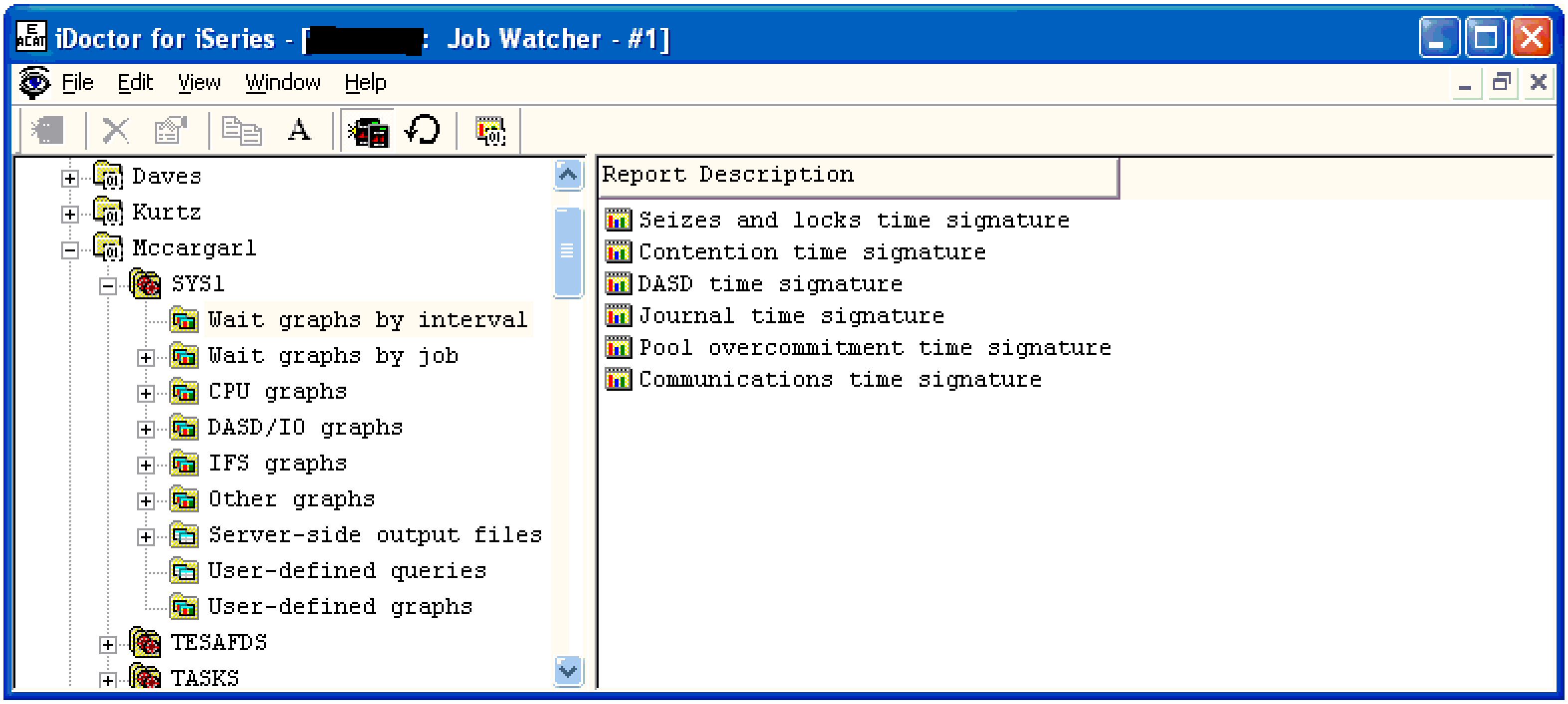
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

Table of Contents | Previous | Next

2.7.5 IFS graphs - summarized

This folder contains a list of graphs which contain summary data over the system-wide job watch relating to waits. These graphs display a bar per interval showing the amount of time spent (across all jobs) in various types of potentially bad waits. High numbers in these graphs do not necessarily indicate a performance problem. A comparison should be made using these graphs from a time when the system is running well, with a time where the system is running poorly in order to determine if there is a problem.

An example of the contents of the wait graphs by interval folder for a system-wide job watch is:



Report Description
Seizes and locks time signature
Contention time signature
DASD time signature
Journal time signature
Pool overcommitment time signature
Communications time signature

The following table illustrates the menu options available by right-clicking on a graph in the list.

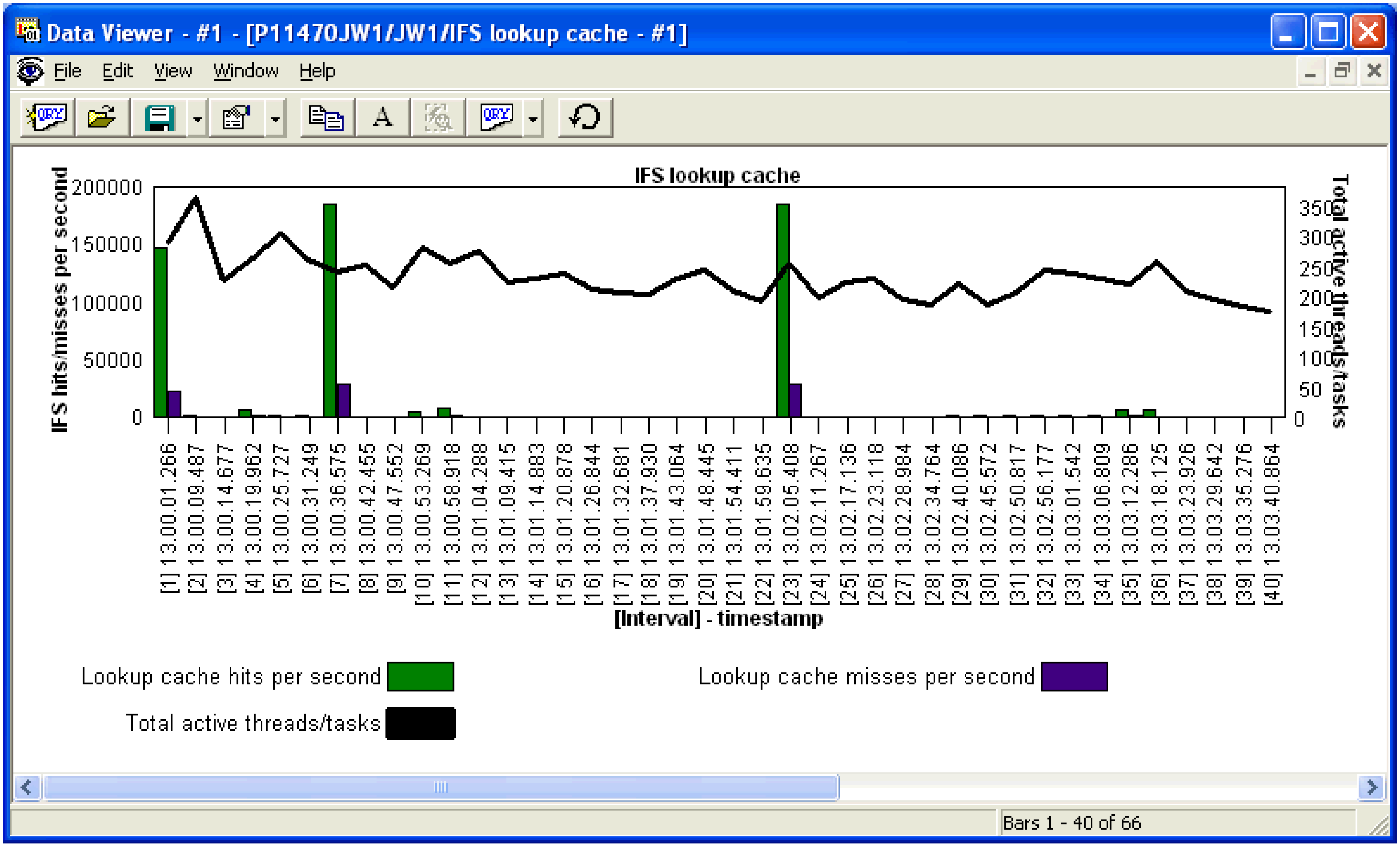
Menu	Description
Open graph(s)	Opens the selected graph(s) into a Data Viewer. If a Data Viewer has already been opened submenus will appear underneath this menu in order to give the option of opening the graph(s) into an already open Data Viewer.

2.7.5.1 IFS lookup cache

Description: This graph shows a **summary** of the rates of IFS lookup cache hits and misses for each interval in the system-wide job watch..

Graph Type: summarized system-wide (vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: The green bars show the rates of lookup cache hits and the blue bars show the rates of lookup cache misses.
Second Y-Axis: The secondary Y-Axis displays the total number of active threads/tasks on the system for each interval.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.
CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.
DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.
Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.
Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.
Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.

Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

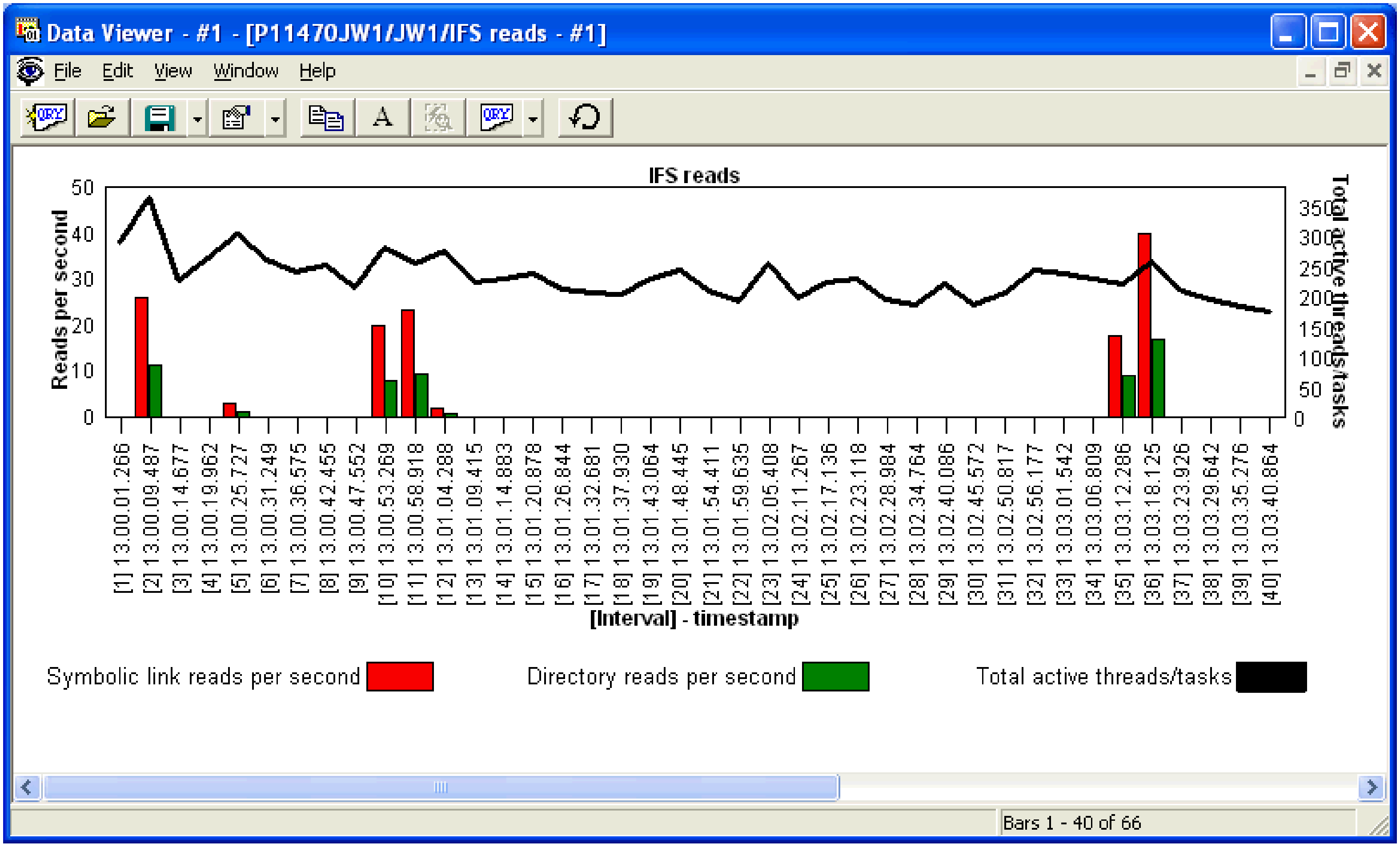
Table of Contents | Previous | Next

2.7.5.2 IFS reads

Description: This graph shows a **summary** of the rates of IFS reads for each interval in the system-wide job watch..

Graph Type: summarized system-wide (vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: The red bars show IFS symbolic link reads per second. The green bars show IFS directory reads per second.
Second Y-Axis: The secondary Y-Axis displays the total number of active threads/tasks on the system for each interval.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.
CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.
DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.
Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.
Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.
Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.

Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

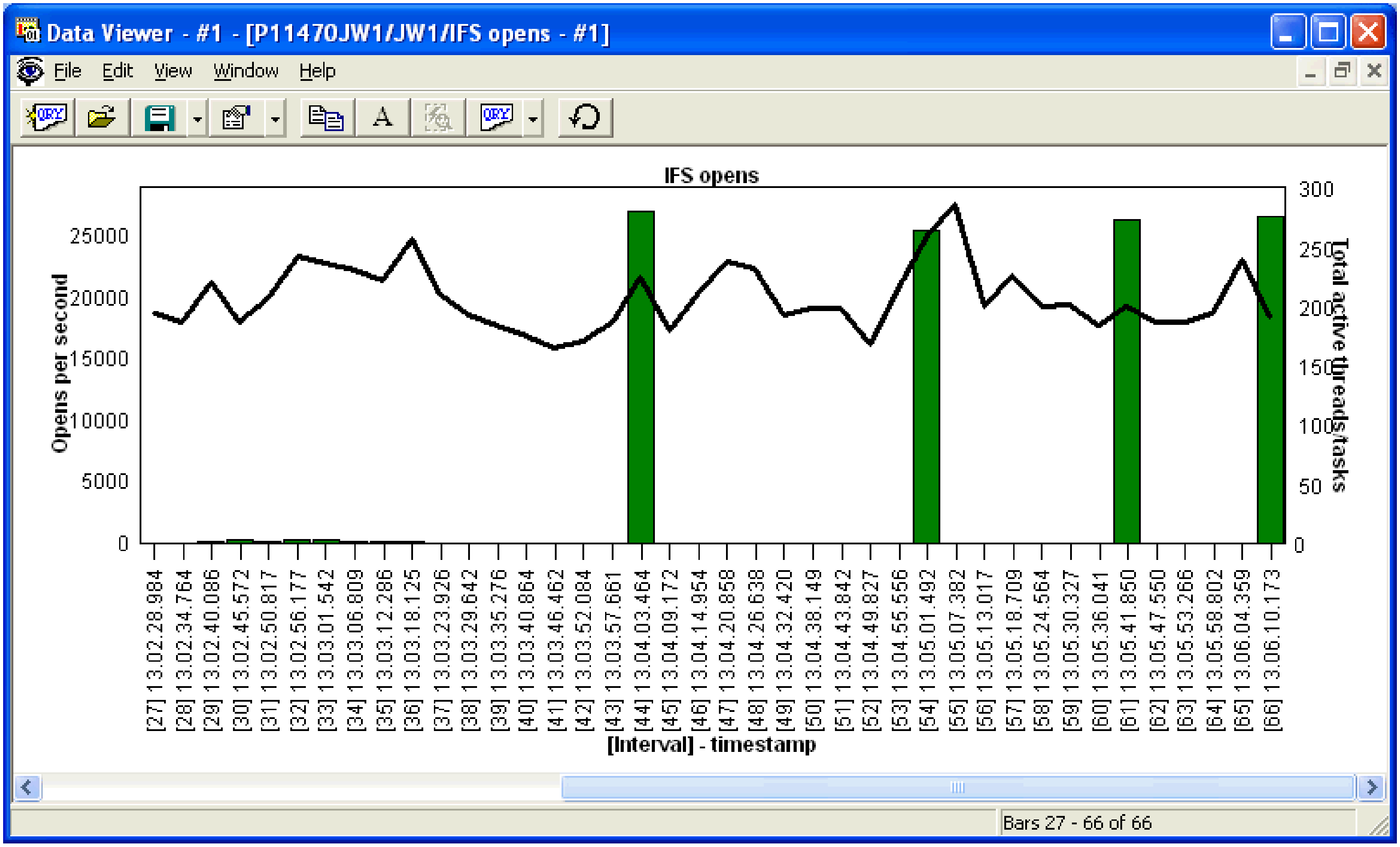
Table of Contents Previous Next

2.7.5.3 IFS opens

Description: This graph shows a **summary** of the rates of IFS file opens during each interval in the system-wide job watch..

Graph Type: summarized system-wide (vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: The green bars show IFS file opens per second.
Second Y-Axis: The secondary Y-Axis displays the total number of active threads/tasks on the system for each interval.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.
CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.
DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.
Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.
Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.
Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.

Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

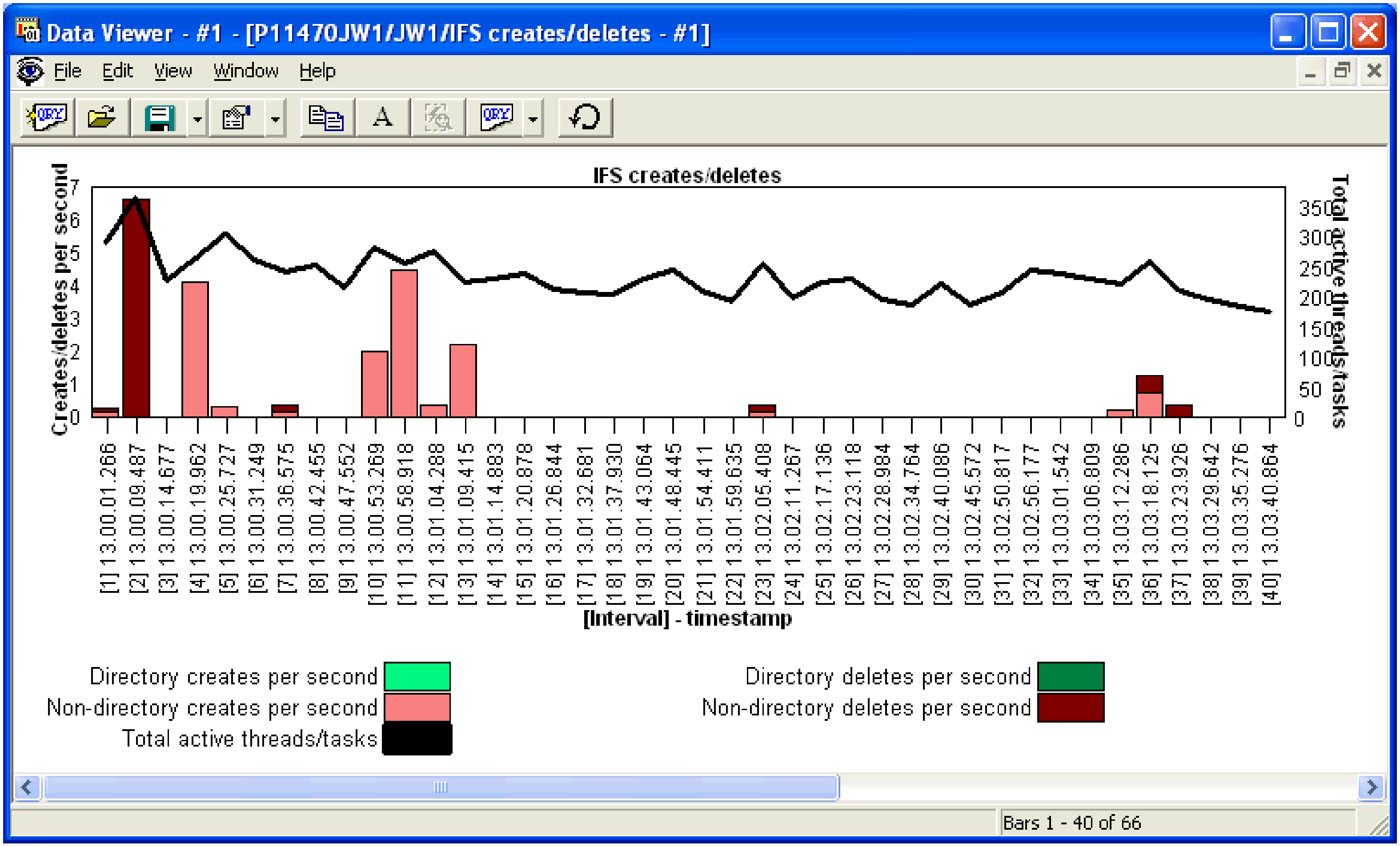
Table of Contents | Previous | Next

2.7.5.4 IFS creates/deletes

Description: This graph shows a **summary** of the rates of IFS directory and non-directory creates and deletes for each interval in the system-wide job watch..

Graph Type: summarized system-wide (vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Contains the rates of IFS directory and non-directory creates and deletes per second. Green colors represent directory operations and red colors represent non-directory operations.
Second Y-Axis: The secondary Y-Axis displays the total number of active threads/tasks on the system for each interval.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.
CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.
DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.
Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.
Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.
Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.

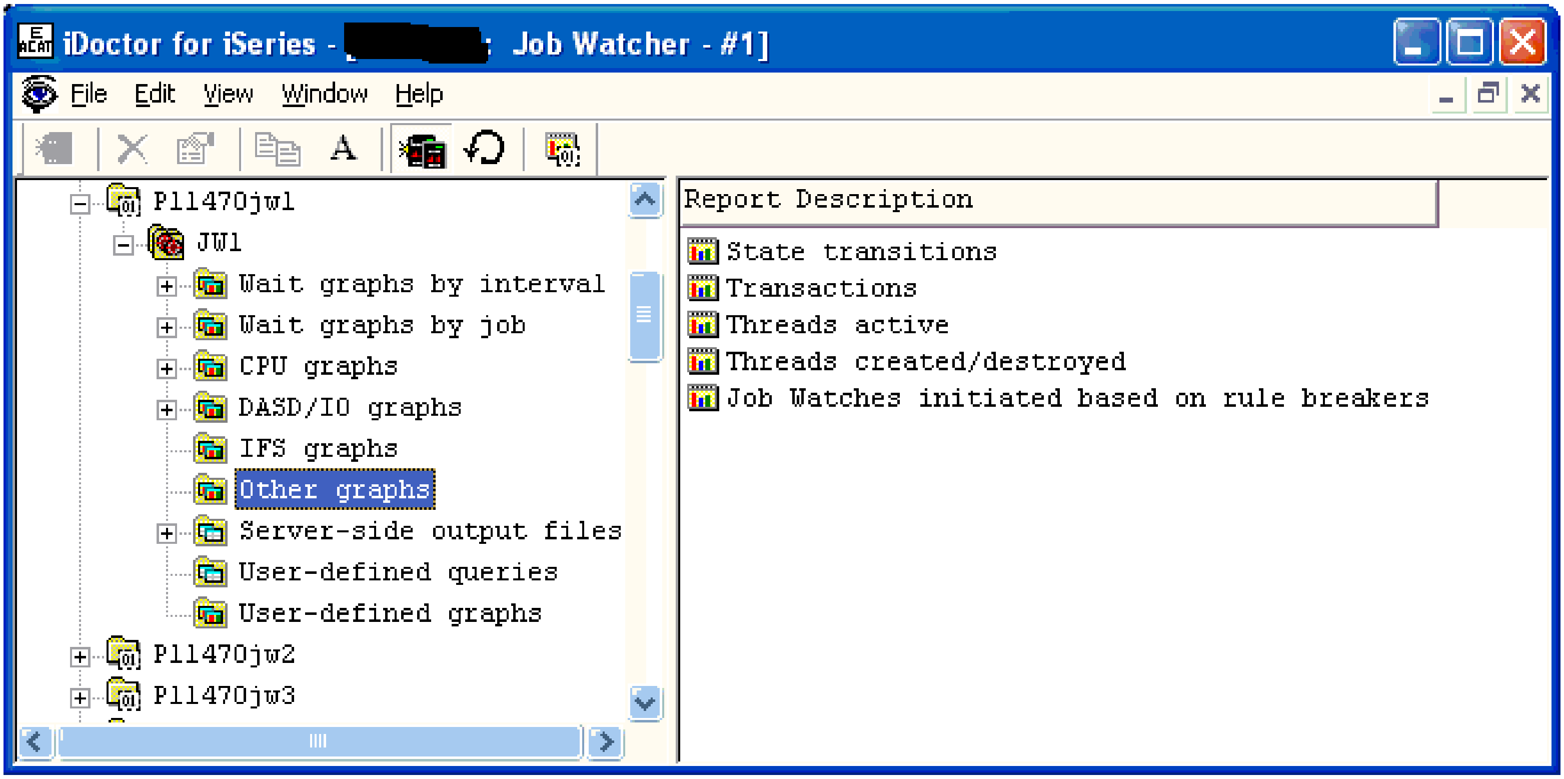
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

Table of Contents Previous Next

2.7.6 Other graphs - summarized

This folder contains a list of graphs which contain performance data not covered in the other graph categories. These graphs display a bar (or set of bars) per interval. The types of data available in these graphs include state transitions and transactions.

An example of the contents of the other graphs folder for a system-wide job watch is:



The following table illustrates the menu options available by right-clicking on a graph in the list.

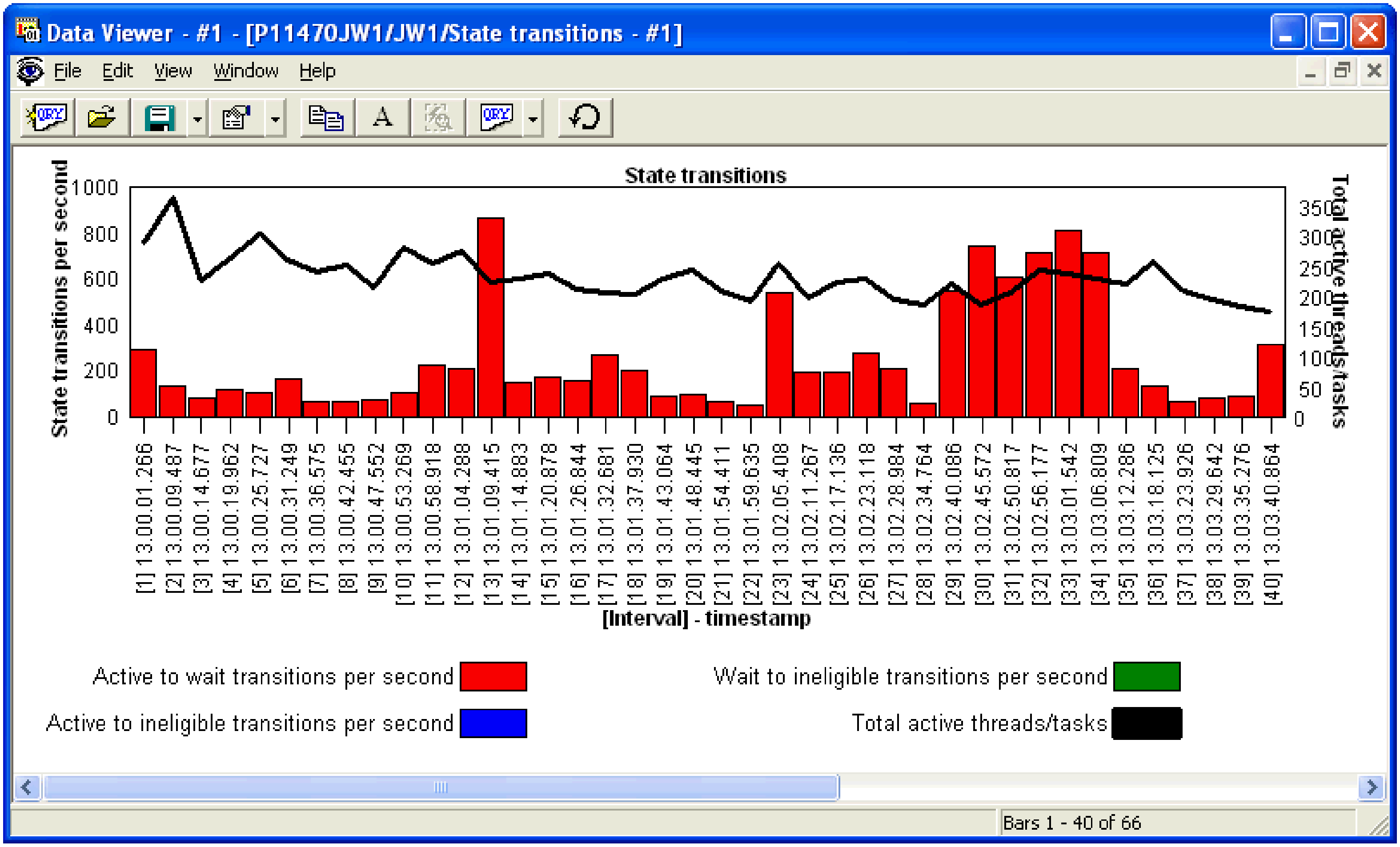
Menu	Description
Open graph(s)	Opens the selected graph(s) into a Data Viewer. If a Data Viewer has already been opened submenus will appear underneath this menu in order to give the option of opening the graph(s) into an already open Data Viewer.

2.7.6.1 State transitions

Description: This graph shows a **summary** of the rates of job state transitions per second for each interval in the system-wide job watch..

Graph Type: summarized system-wide (vertical bar)
X-Axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Each bar shows the rates of the following types of job state transitions: active to wait, wait to ineligible, and active to ineligible.
Second Y-Axis: The secondary Y-Axis displays the total number of active threads/tasks on the system for each interval.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.
CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.
DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.
Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.
Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.
Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.

Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

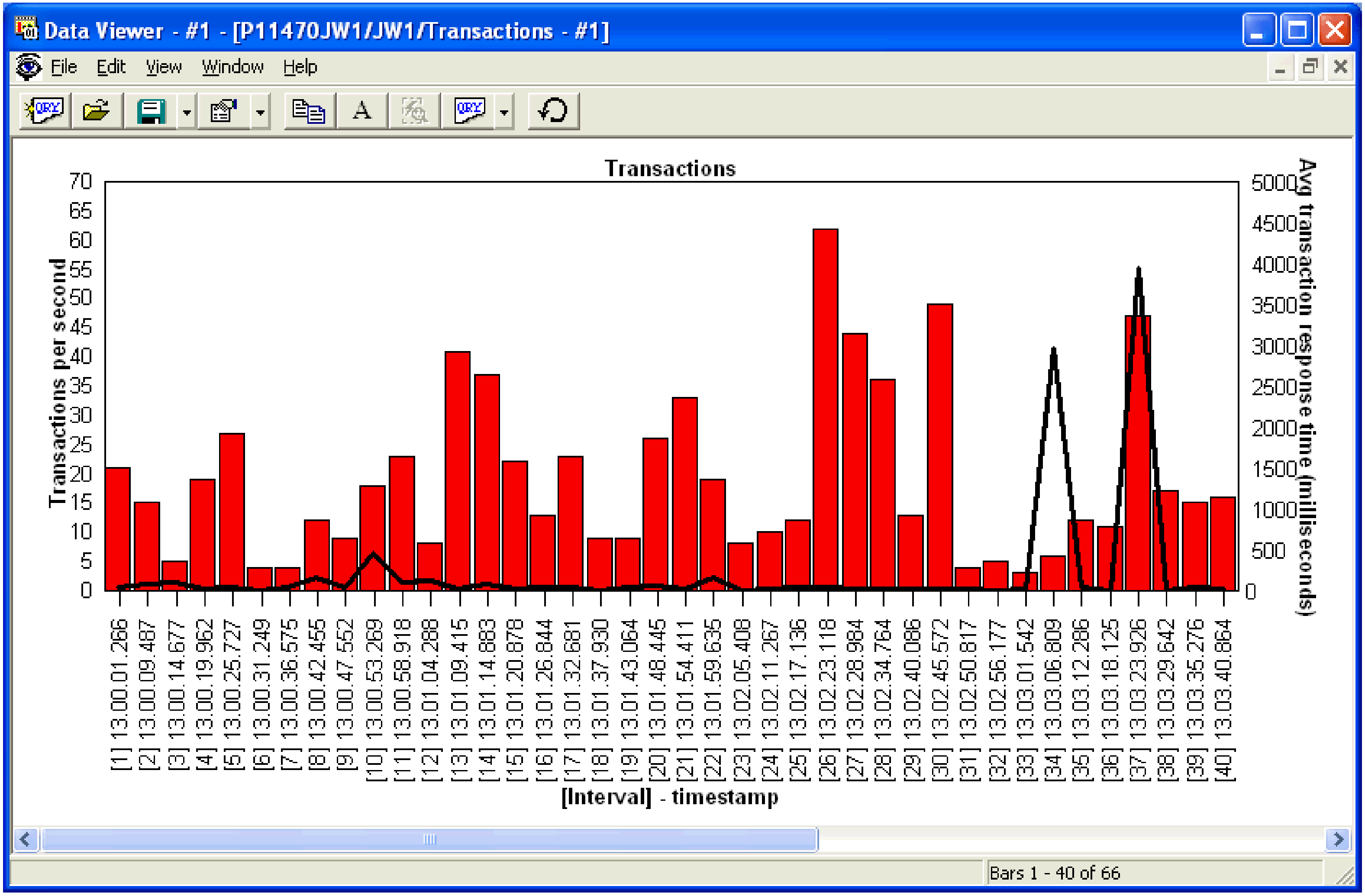
Table of Contents | Previous | Next

2.7.6.2 Transactions

Description: This graph shows a **summary** of the transaction rates per interval and the average transaction response time per interval in the system-wide job watch..

Graph Type: summarized system-wide (vertical bar)
X-Axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Each bar shows the transaction per second during each interval.
Second Y-Axis: The line shows the system-wide transaction average response time (in milliseconds).

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.
CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.
DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.
Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.
Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.
Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.

Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

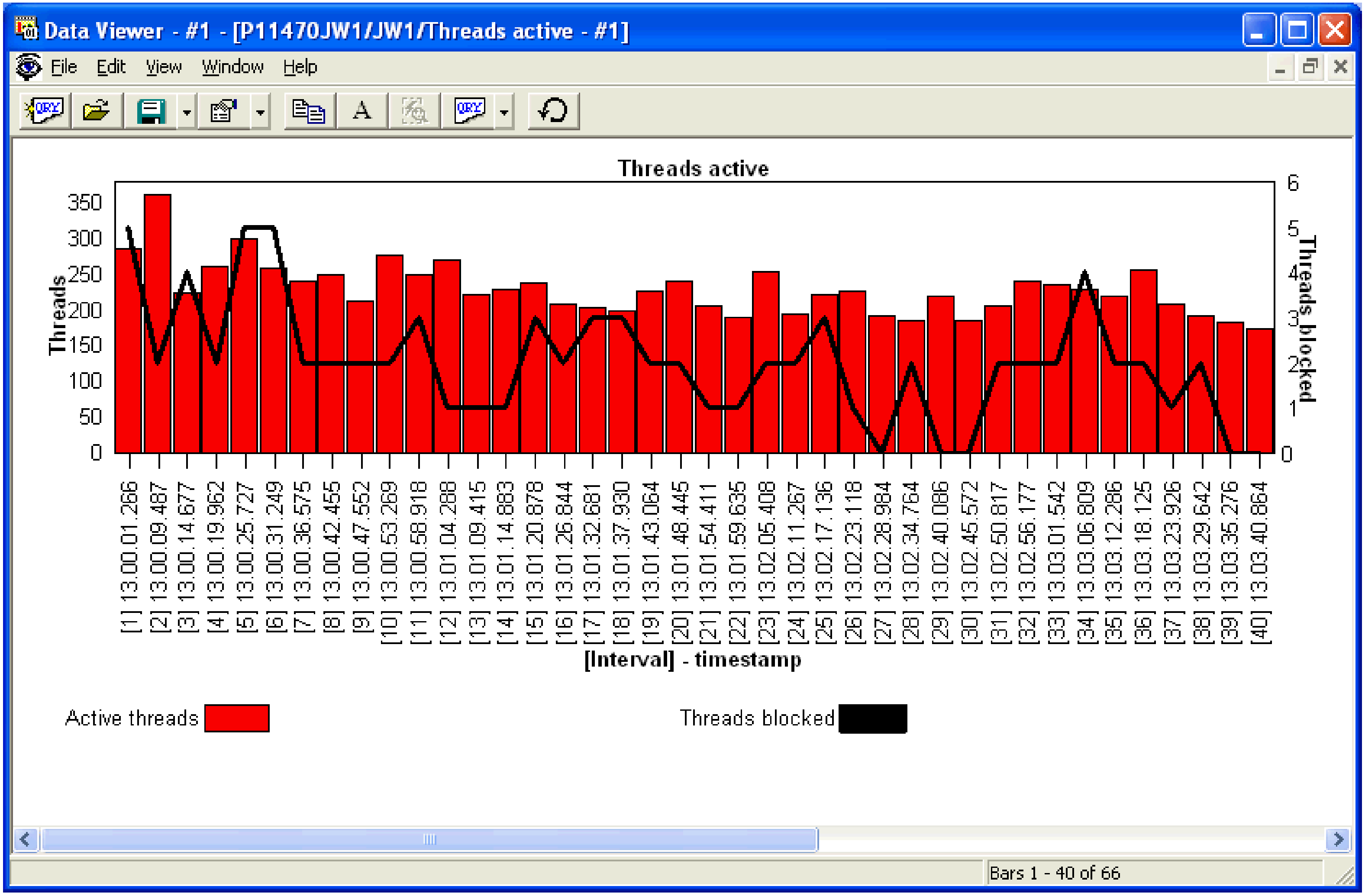
Table of Contents | Previous | Next

2.7.6.3 Threads active

Description: This graph shows a **count** of the total threads active each interval in the system-wide job watch..

Graph Type: summarized system-wide (vertical bar)
X-Axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Each bar shows the total threads active during each interval in the system-wide job watch.
Second Y-Axis: The secondary Y-Axis displays the total number of threads in a "bad wait" each interval. These waits are defined by the system-wide Job Watcher collection engine at V5R2 and include things like seizes and locks.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.
CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.
DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.
Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.
Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.
Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.

Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

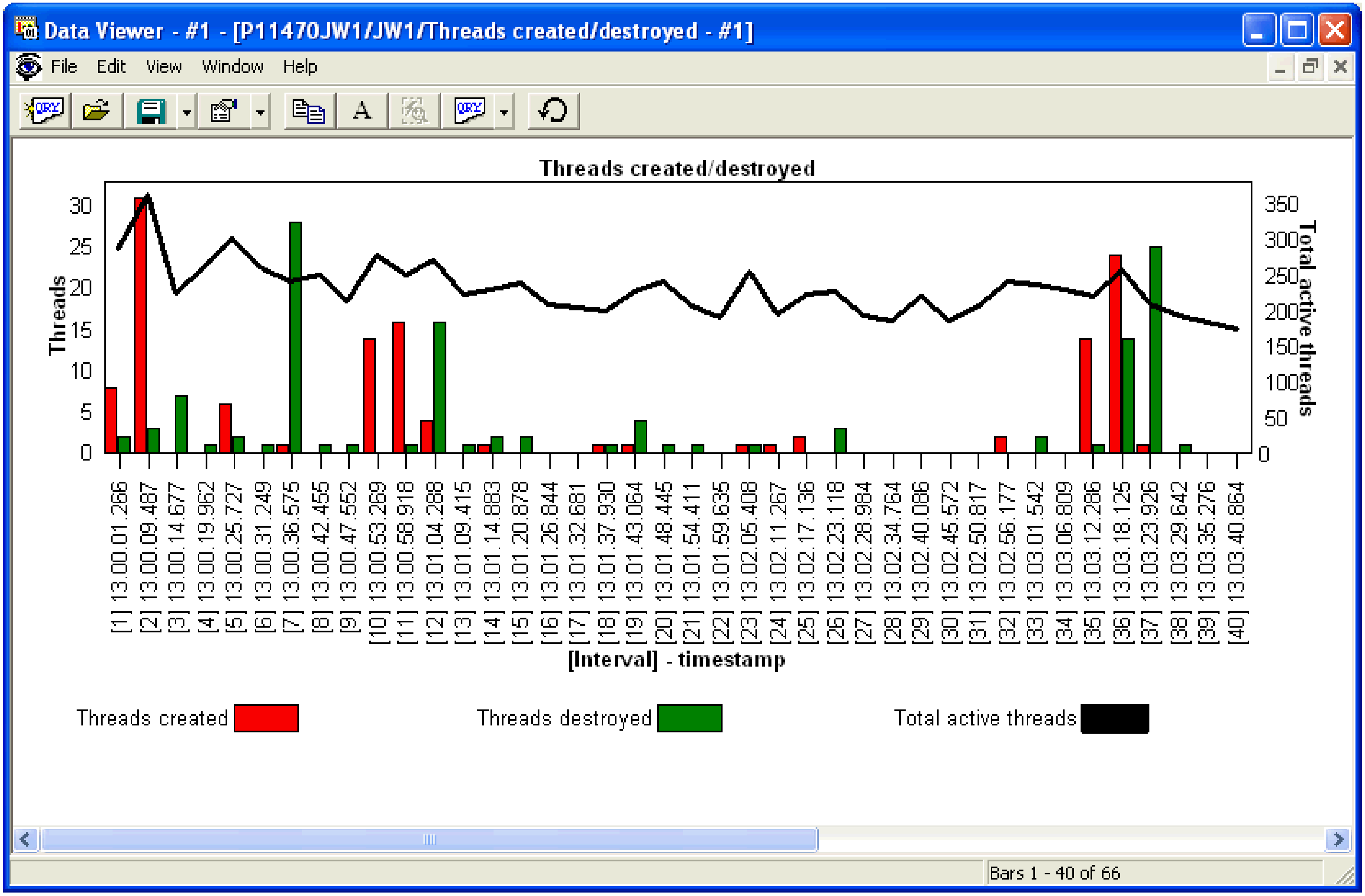
Table of Contents | Previous | Next

2.7.6.4 Threads created/destroyed

Description: This graph shows a **summary** of the total threads created and destroyed each interval in the system-wide job watch..

Graph Type: summarized system-wide (vertical bar)
X-Axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Each bar shows the total threads/tasks created or destroyed each interval in the job watch. The red bars show threads created and the green bars show threads destroyed.
Second Y-Axis: The secondary Y-Axis displays the total number of active threads/tasks on the system for each interval.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.
CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.
DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.
Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.
Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.
Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.

Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

Table of Contents Previous Next

2.7.6.5 Job watches initiated by rule breakers

Description: This graph shows a **summary** of the total job-specific job watches created by a rule definition defined within the current system-wide job watch..

Graph Type: summarized system-wide (vertical bar)
X-Axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Each bar shows the total job-specific job watches started because the rule definition's conditions were satisfied each interval in the job watch.

Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Wait graphs	Displays one of the detailed wait graphs for the bar/job in a job-specific job watch.
CPU graphs	Displays one of the detailed CPU graphs for the bar/job in a job-specific job watch.
DASD/IO graphs	Displays one of the detailed DASD/IO graphs for the bar/job in a job-specific job watch.
IFS graphs	Displays one of the detailed IFS graphs for the bar/job in a job-specific job watch.
Java graphs	Displays one of the detailed Java graphs for the bar/job in a job-specific job watch.
Other graphs	Displays one of the other detailed graphs for the bar/job in a job-specific job watch. This category includes transactions, state transitions and numeric data type exceptions.
Detail reports	Displays one of the table view reports for the bar/job in a job-specific job watch.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window. This window will also appear by left-clicking on any bar in the graph.

Table of Contents Previous Next

2.8 System-wide Job Watch detail reports

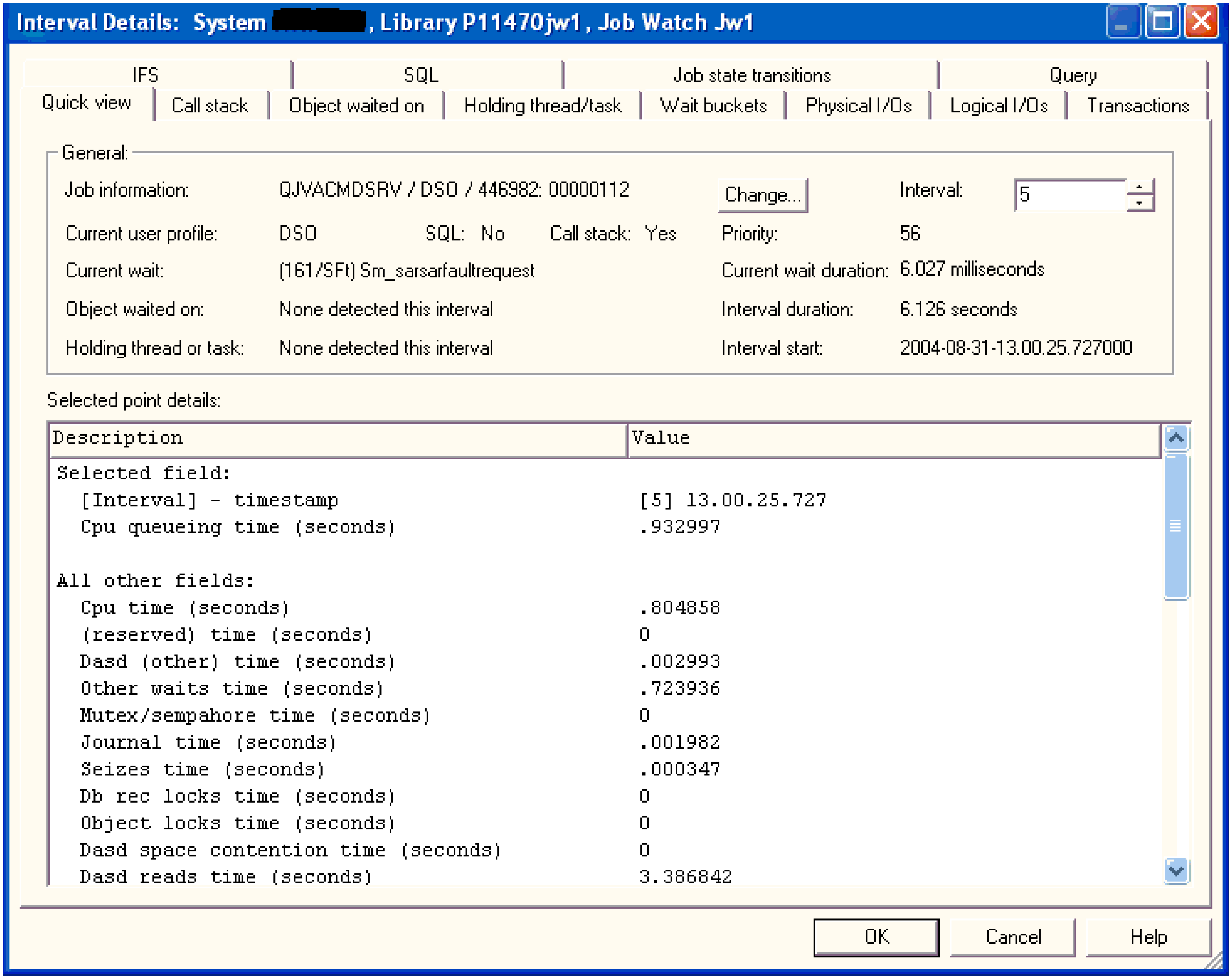
This section covers the detailed property pages, graphs and tables available in a system-wide job watch. These reports are available from by selecting any job/thread/task in a table view or graph and right-clicking on the record or bar representing the job. For example, right-clicking on a bar (represent a job) in a summarized graph provides many different detailed reports showing information about just the specific job selected. In most cases each bar in a detail graph represents some value in the job's data reported every interval.

Table of Contents | Previous | Next

2.8.1 Interval Properties

The interval properties window displays information about a job in a system-wide job watch for a specific interval. These properties contain several different panels which vary depending on the data available. Some panels are not shown if optional data has not been collected.

Getting to the interval details panel can be done by clicking on a bar in one of the system-wide summary graphs job run/wait time signatures or by double-clicking on a record in a report/table in a system-wide Job Watcher output file which contains both a taskcount field and an interval field in the record. These two fields identify the job/thread/task and interval to display in the interval property pages.



Interval Details: System [redacted], Library P11470jw1, Job Watch Jw1

General:

Job information: QJVACMDSRV / DSO / 446982: 00000112

Interval: 5

Current user profile: DSO SQL: No Call stack: Yes

Priority: 56

Current wait: (161/SFt) Sm_sarsarfaultrequest

Current wait duration: 6.027 milliseconds

Object waited on: None detected this interval

Interval duration: 6.126 seconds

Holding thread or task: None detected this interval

Interval start: 2004-08-31-13.00.25.727000

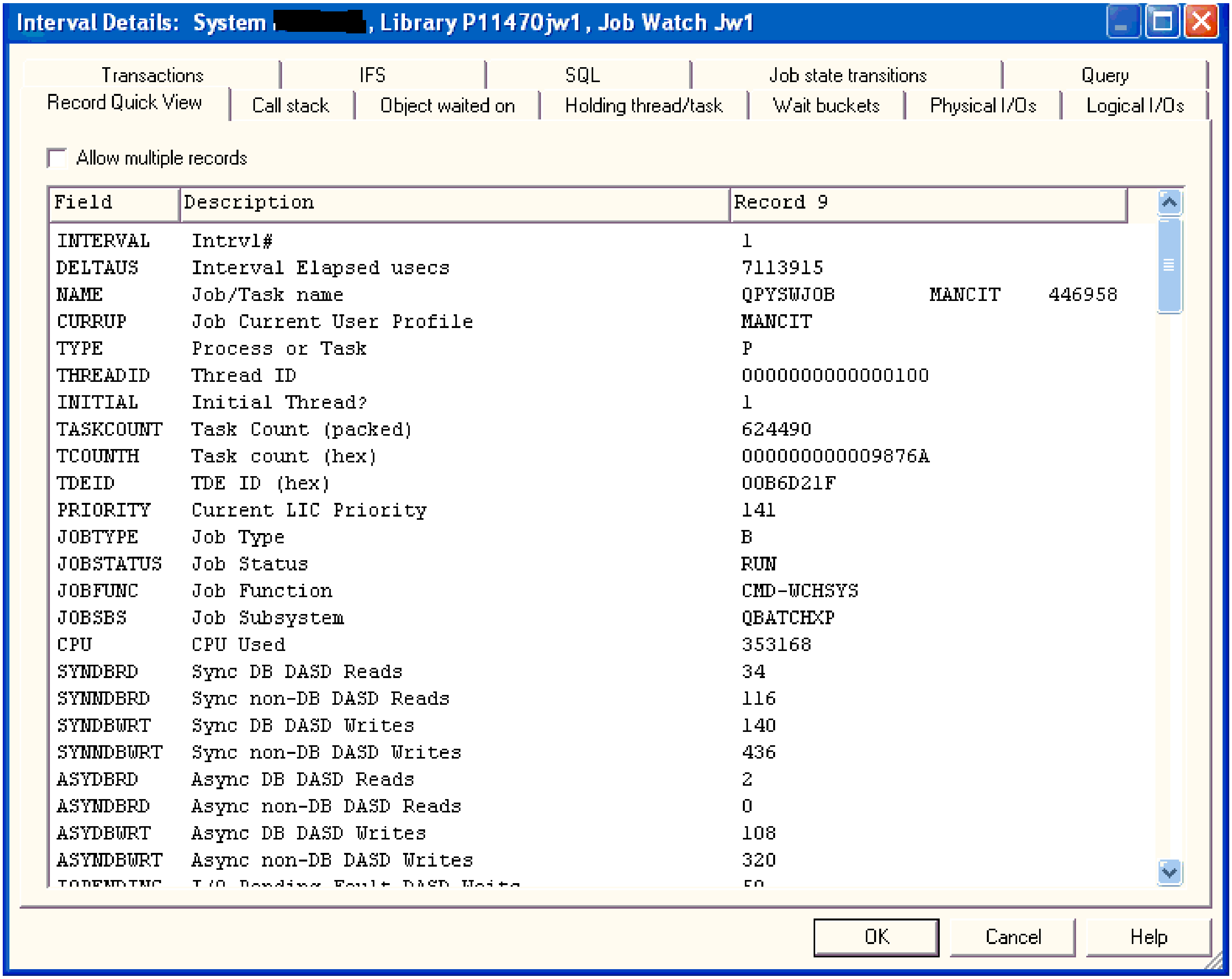
Selected point details:

```
Description                          Value
 Selected field:
   [Interval] - timestamp             [5] 13.00.25.727
   Cpu queueing time (seconds)        .932997

 All other fields:
   Cpu time (seconds)                 .804858
   (reserved) time (seconds)          0
   Dasd (other) time (seconds)        .002993
   Other waits time (seconds)         .723936
   Mutex/sempahore time (seconds)     0
   Journal time (seconds)             .001982
   Seizes time (seconds)              .000347
   Db rec locks time (seconds)        0
   Object locks time (seconds)        0
   Dasd space contention time (seconds)  0
   Dasd reads time (seconds)          3.386842
```

Table of Contents | Previous | Next

2.8.1.1 Record Quick View

This page of the interface is found by double-clicking on one of the Job Watcher output files from a table view. The window displays a record vertically from a table view. The other tabs in the view display additional information about the currently selected job and record . These additional tabs are only shown if a valid taskcount field and interval field is present in the file and SQL query. The taskcount field identifies the job/thread/task within the system-wide job watch.



□ Allow multiple records

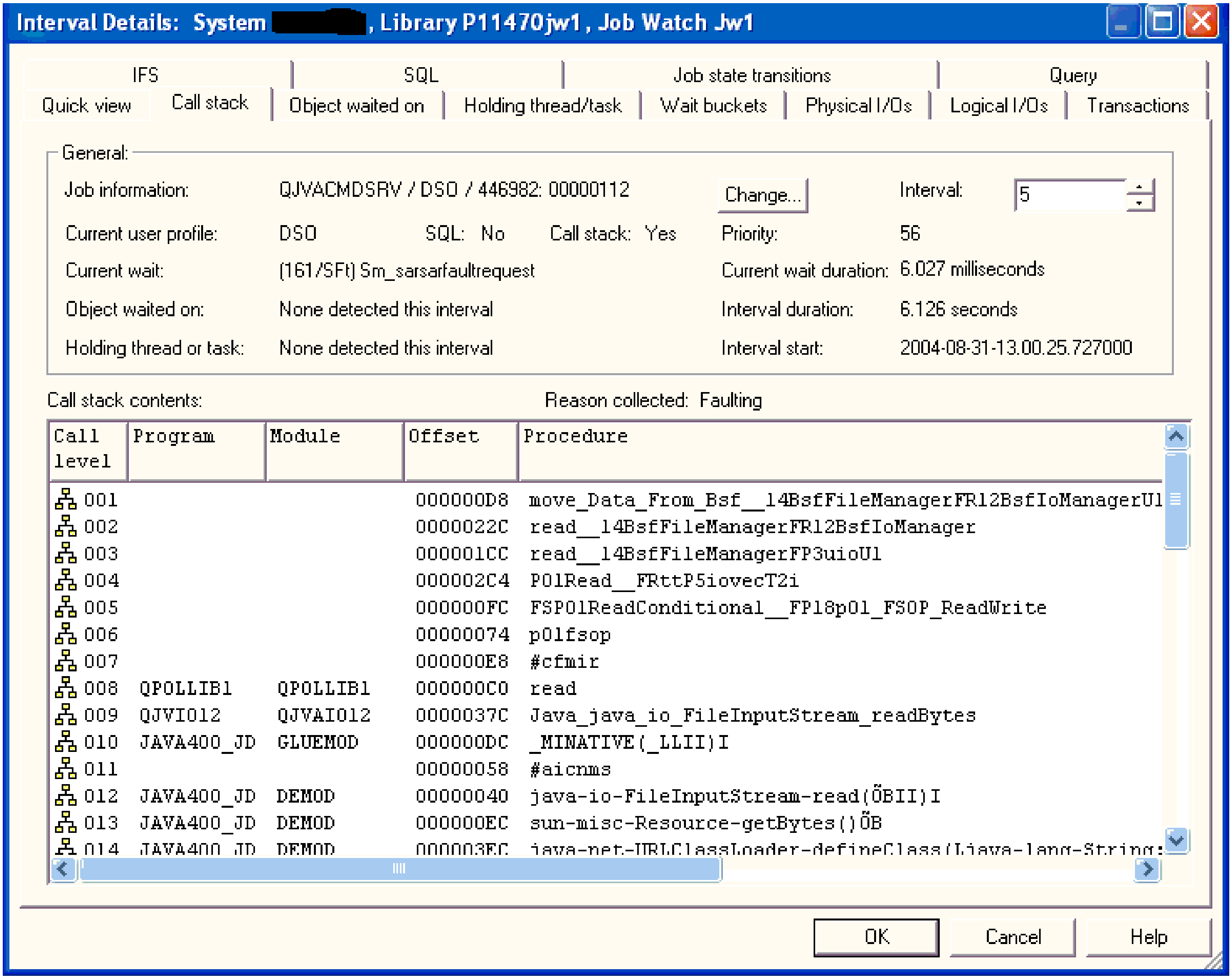
Field	Description	Record 9
INTERVAL	Intrvl#	1
DELTAUS	Interval Elapsed usecs	7113915
NAME	Job/Task name	QPYSWJOB MANCIT 446958
CURRUP	Job Current User Profile	MANCIT
TYPE	Process or Task	P
THREADID	Thread ID	0000000000000100
INITIAL	Initial Thread?	1
TASKCOUNT	Task Count (packed)	624490
TCOUNTH	Task count (hex)	000000000009876A
TDEID	TDE ID (hex)	00B6D21F
PRIORITY	Current LIC Priority	141
JOBTYPE	Job Type	B
JOBSTATUS	Job Status	RUN
JOBFUNC	Job Function	CMD-WCHSYS
JOBSBS	Job Subsystem	QBATCHXP
CPU	CPU Used	353168
SYNDBRD	Sync DB DASD Reads	34
SYNNDBRD	Sync non-DB DASD Reads	116
SYNDBWRT	Sync DB DASD Writes	140
SYNNDBWRT	Sync non-DB DASD Writes	436
ASYDBRD	Async DB DASD Reads	2
ASYNDBRD	Async non-DB DASD Reads	0
ASYDBWRT	Async DB DASD Writes	108
ASYNDBWRT	Async non-DB DASD Writes	320
IOPENDING	I/O Pending Fault DASD Waits	50

Table of Contents Previous Next

2.8.1.2 Call stack

This page shows the call stack (if available) for a job in a system-wide job watch at the moment the snapshot for the interval was taken. The call stack can be up to 50 levels deep, and is shown "bottom up". The call stack includes LIC information below the MI. Call stacks for system-wide job watches are not collected every interval for every job. They are only collected for certain situations depending on the specified parameters for call stacks used in the Start Job Watch Wizard.

If a call stack has been collected a description of the reason the call stack was gathered is shown above the stack on this window. In the example below, page faulting that was occurring when the snapshot was taken caused this call stack to be collected.



General:

Job information: QJVACMDSRV / DSO / 446982: 00000112

Interval: 5

Current user profile: DSO SQL: No Call stack: Yes

Priority: 56

Current wait: (161/SFt) Sm_sarsarfaultrequest

Current wait duration: 6.027 milliseconds

Object waited on: None detected this interval

Interval duration: 6.126 seconds

Holding thread or task: None detected this interval

Interval start: 2004-08-31-13.00.25.727000

Call stack contents: Reason collected: Faulting

Call level	Program	Module	Offset	Procedure
001			000000D8	move_Data_From_Bsf__14BsfFileManagerFR12BsfIoManagerUl
002			0000022C	read__14BsfFileManagerFR12BsfIoManager
003			000001CC	read__14BsfFileManagerFP3uioUl
004			000002C4	P01Read__FRttP5iovecT2i
005			000000FC	FSP01ReadConditional__FP18p01_FSOP_ReadWrite
006			00000074	p01fsop
007			000000E8	#cfmir
008	QPOLLIB1	QPOLLIB1	000000C0	read
009	QJVI012	QJVAI012	0000037C	Java_java_io_FileInputStream_readBytes
010	JAVA400_JD	GLUEMOD	000000DC	_MINATIVE(_LLII)I
011			00000058	#aicnms
012	JAVA400_JD	DEMOD	00000040	java-io-FileInputStream-read(ÕBII)I
013	JAVA400_JD	DEMOD	000000EC	sun-misc-Resource-getBytes()ÕB
014	JAVA400_JD	DEMOD	000003EC	java-net-URLClassLoader-defineClass(Ljava-lang-String;

This window contains the following fields:

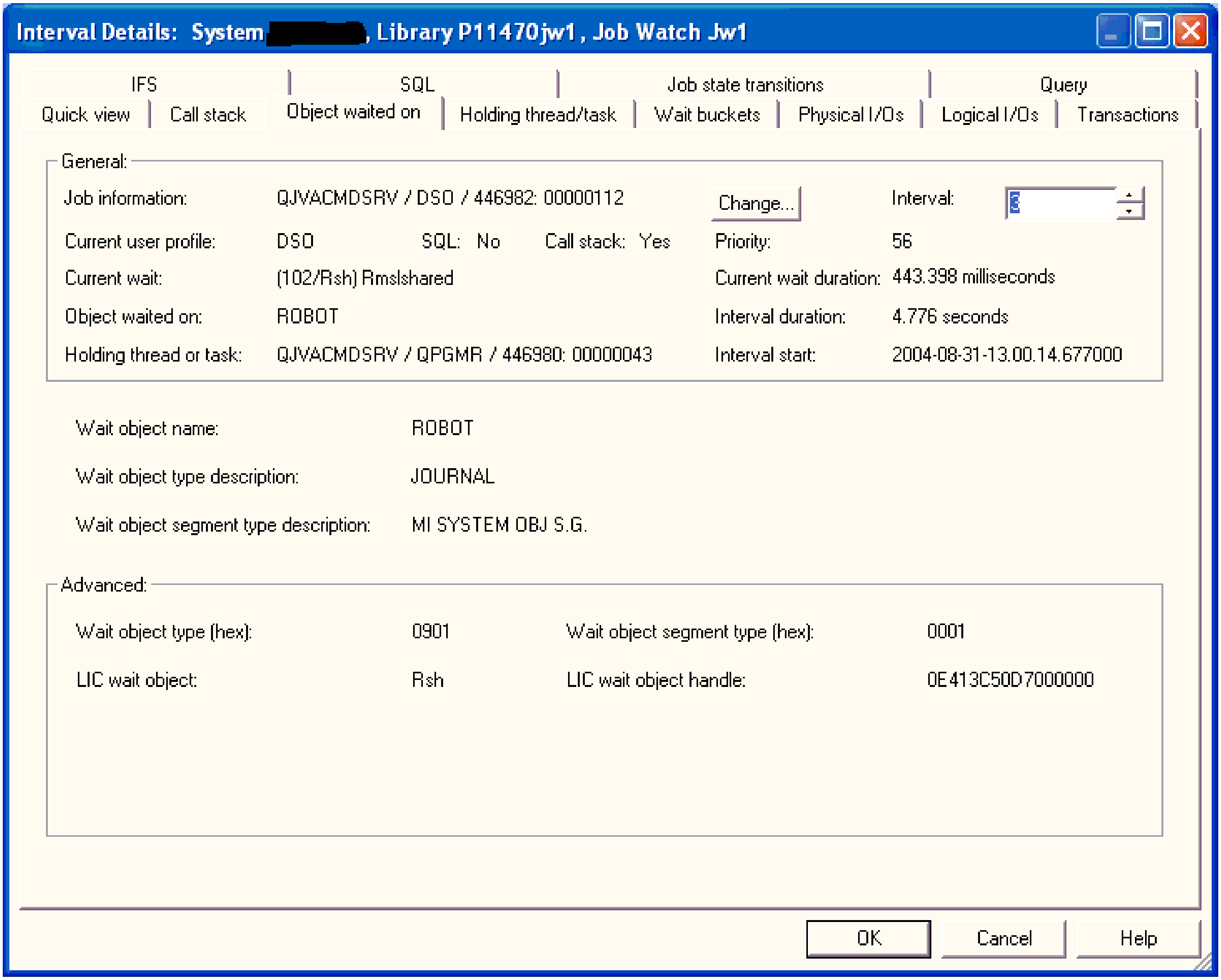
Field name	Description

Job information	The fully qualified job/thread id or task information this interface applies to. The Change... button allows the user to change the current job/thread/task to another. Changing the selected job will update the interface information on this panel but the information on the record quick view panel will still show the data for the original job.
Interval	Displays the selected interval the data shown applies to. The up and down buttons allow the user to change the selected interval. If the interval specified does not exist for the selected job no data will be shown.
Current user profile	The user profile currently active when the job was running at the interval specified.
Priority	The priority of the job for the specified interval.
Current wait	The current wait information contains information about the wait point (numeric identifier and 3 character wait code) and the description of the wait point from file QPYRTJWAT2. This field indicates the type of wait (or CPU) that was occurring when the snapshot to gather information about the job was taken. The wait point shown belongs to a wait bucket. Information about the current bucket is available on the "Wait buckets" panel in this interface.
Current wait duration	The total time the job has been in the current wait type. Everytime the wait type changes or CPU is used, this value is reset at 0. The current wait duration could be days for an idle job doing nothing and this would not indicate any problem. However if the current wait was a seize lock condition and the current wait duration was 30 seconds, this could flag a potential problem. For more information on understanding wait analysis, see the white paper available on the home page of the iDoctor for iSeries Web site.
Object waited on	This field contains information about the object the job/thread/task waited on for the current interval (if any) as well as the type of object being waited on. If an object is specified on this field, additional information about the object is available on the "Object waited on" panel in this interface
Interval duration	The time that elapsed when collecting the selected interval.
Holding thread or task	This field contains information about the job/thread/task that was holding the selected job during the selected interval and preventing it from doing work. If a holding thread or task is specified within this field, additional information about the holder is available on the "Holding thread/task" panel in this interface
Interval start	The date/time the current interval began for the specified job.
Call stack	This contains program, module and procedure names within the current job's call stack for the specified interval. The call stack is gathered at the end of the interval. It is a snapshot of what programs/procedures were running in the job at this particular moment in time. The call stack can contain up to 50 levels of information.

Table of Contents | Previous | Next

2.8.1.3 Object waited on

This page of the interval details shows information about the object the current job was waiting on at the moment the snapshot for the specified interval was taken.



General:

Job information:	QJVACMDSRV / DSO / 446982: 00000112			Interval:	3
Current user profile:	DSO	SQL: No	Call stack: Yes	Priority:	56
Current wait:	(102/Rsh) Rmslshared			Current wait duration:	443.398 milliseconds
Object waited on:	ROBOT			Interval duration:	4.776 seconds
Holding thread or task:	QJVACMDSRV / QPGMR / 446980: 00000043			Interval start:	2004-08-31-13.00.14.677000

Wait object name:	ROBOT
Wait object type description:	JOURNAL
Wait object segment type description:	MI SYSTEM OBJ S.G.

Advanced:

Wait object type (hex):	0901	Wait object segment type (hex):	0001
LIC wait object:	Rsh	LIC wait object handle:	0E413C50D7000000

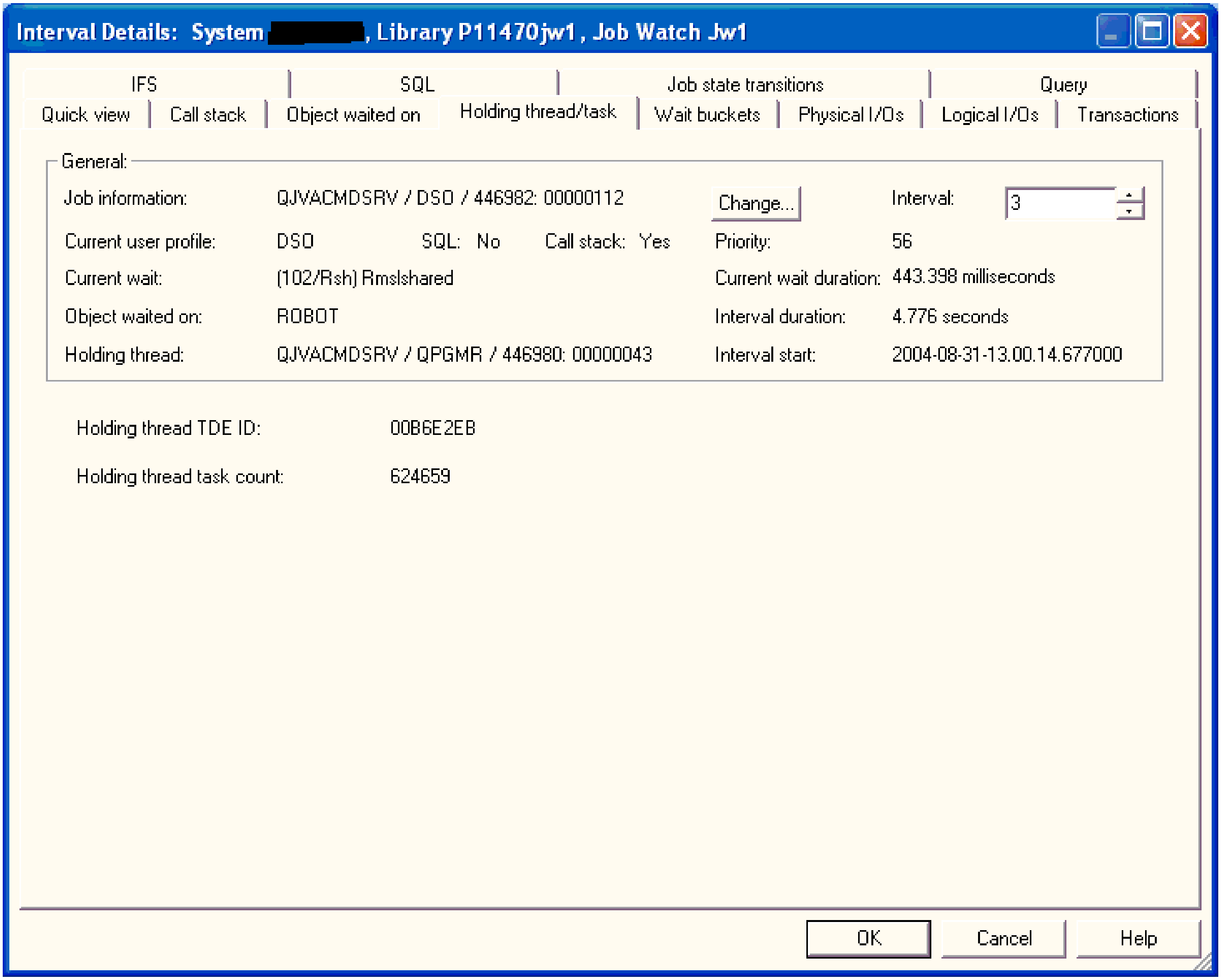
This window contains the following fields:

Field name	Description
Job information	The fully qualified job/thread id or task information this interface applies to. The Change... button allows the user to change the current job/thread/task to another. Changing the selected job will update the interface information on this panel but the information on the record quick view panel will still show the data for the original job.

Interval	Displays the selected interval the data shown applies to. The up and down buttons allow the user to change the selected interval. If the interval specified does not exist for the selected job no data will be shown.
Current user profile	The user profile currently active when the job was running at the interval specified.
Priority	The priority of the job for the specified interval.
Current wait	The current wait information contains information about the wait point (numeric identifier and 3 character wait code) and the description of the wait point from file QPYRTJWAT2. This field indicates the type of wait (or CPU) that was occurring when the snapshot to gather information about the job was taken. The wait point shown belongs to a wait bucket. Information about the current bucket is available on the "Wait buckets" panel in this interface.
Current wait duration	The total time the job has been in the current wait type. Everytime the wait type changes or CPU is used, this value is reset at 0. The current wait duration could be days for an idle job doing nothing and this would not indicate any problem. However if the current wait was a seize lock condition and the current wait duration was 30 seconds, this could flag a potential problem. For more information on understanding wait analysis, see the white paper available on the home page of the iDoctor for iSeries Web site.
Object waited on	This field contains information about the object the job/thread/task waited on for the current interval (if any) as well as the type of object being waited on. If an object is specified on this field, additional information about the object is available on the "Object waited on" panel in this interface
Interval duration	The time that elapsed when collecting the selected interval.
Holding thread or task	This field contains information about the job/thread/task that was holding the selected job during the selected interval and preventing it from doing work. If a holding thread or task is specified within this field, additional information about the holder is available on the "Holding thread/task" panel in this interface
Interval start	The date/time the current interval began for the specified job.

2.8.1.4 Holding thread/task

This page shows information about the job holding the current job.



Interval Details: System [redacted], Library P11470jw1, Job Watch Jw1

Job information: QJVACMDSRV / DSO / 446982: 00000112

Interval: 3

Current user profile: DSO

SQL: No

Call stack: Yes

Priority: 56

Current wait: (102/Rsh) Rmslshared

Current wait duration: 443.398 milliseconds

Object waited on: ROBOT

Interval duration: 4.776 seconds

Holding thread: QJVACMDSRV / QPGMR / 446980: 00000043

Interval start: 2004-08-31-13.00.14.677000

Holding thread TDE ID: 00B6E2EB

Holding thread task count: 624659

This window contains the following fields:

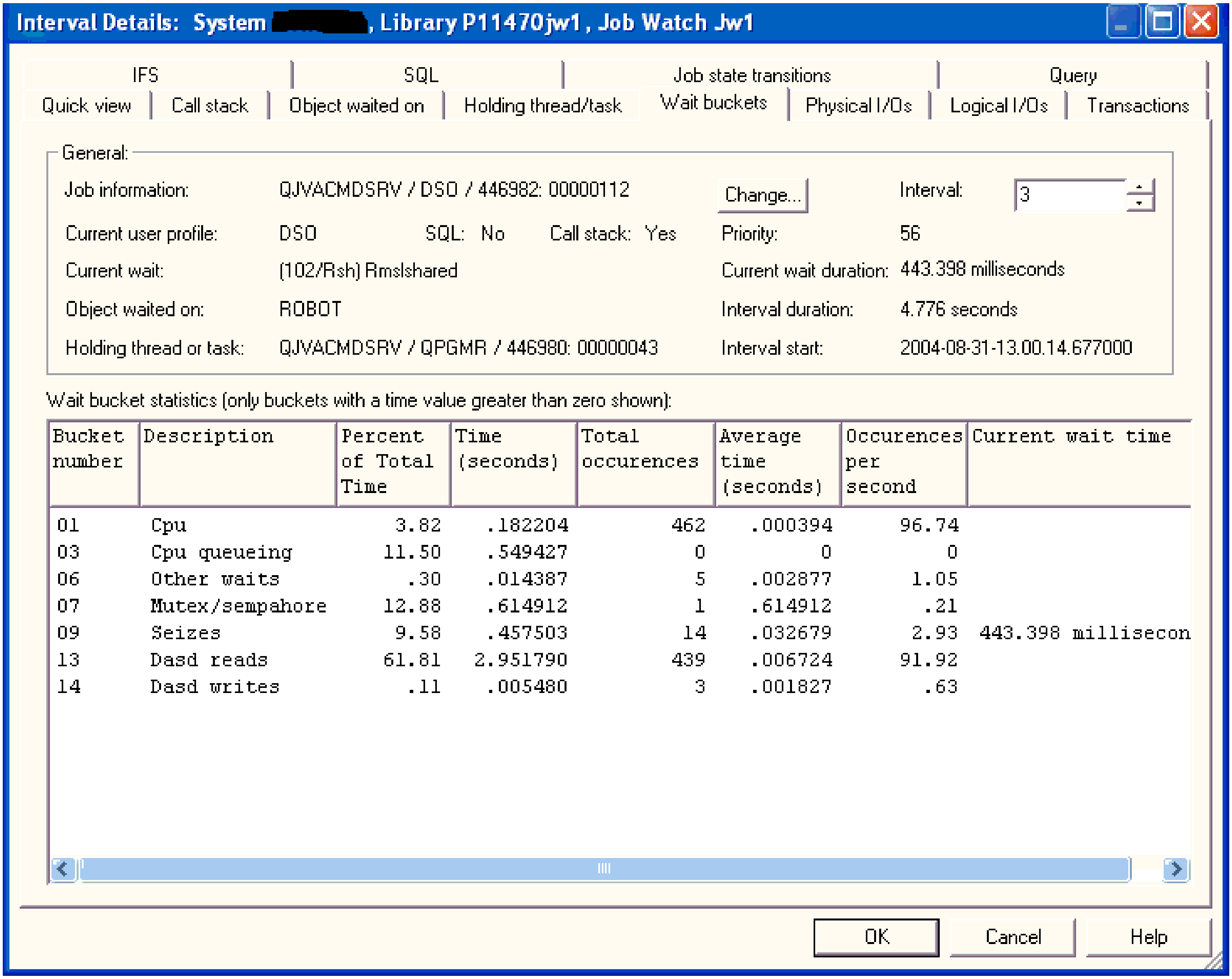
Field name	Description
Job information	The fully qualified job/thread id or task information this interface applies to. The Change... button allows the user to change the current job/thread/task to another. Changing the selected job will update the interface information on this panel but the information on the record quick view panel will still show the data for the original job.
Interval	Displays the selected interval the data shown applies to. The up and down buttons allow the user to change the selected interval. If the interval specified does not exist for the selected job no data will be shown.

Current user profile	The user profile currently active when the job was running at the interval specified.
Priority	The priority of the job for the specified interval.
Current wait	The current wait information contains information about the wait point (numeric identifier and 3 character wait code) and the description of the wait point from file QPYRTJWAT2. This field indicates the type of wait (or CPU) that was occurring when the snapshot to gather information about the job was taken. The wait point shown belongs to a wait bucket. Information about the current bucket is available on the "Wait buckets" panel in this interface.
Current wait duration	The total time the job has been in the current wait type. Everytime the wait type changes or CPU is used, this value is reset at 0. The current wait duration could be days for an idle job doing nothing and this would not indicate any problem. However if the current wait was a seize lock condition and the current wait duration was 30 seconds, this could flag a potential problem. For more information on understanding wait analysis, see the white paper available on the home page of the iDoctor for iSeries Web site.
Object waited on	This field contains information about the object the job/thread/task waited on for the current interval (if any) as well as the type of object being waited on. If an object is specified on this field, additional information about the object is available on the "Object waited on" panel in this interface
Interval duration	The time that elapsed when collecting the selected interval.
Holding thread or task	This field contains information about the job/thread/task that was holding the selected job during the selected interval and preventing it from doing work. If a holding thread or task is specified within this field, additional information about the holder is available on the "Holding thread/task" panel in this interface
Interval start	The date/time the current interval began for the specified job.
Holding thread TDE ID	TDE (task dispatching element) ID of the holding thread or task. This field is displayed as an 8 character hex string.
Holding thread task count	A number uniquely identifing the holding thread.

Table of Contents | Previous | Next

2.8.1.5 Wait buckets

This page shows the waits that occurred during the specified interval for the job indicated. The wait buckets are shown along with the percentage of time spent in each bucket. The final column in the list shows which bucket the current wait (listed in the 3rd line of the general section) belongs to. In the example below the wait point Rmslshared (a type of seize condition) is shown to belong to bucket 9 (seizes).



Interval Details: System [redacted], Library P11470jw1, Job Watch Jw1

General:

Job information: QJVACMDSRV / DSO / 446982: 00000112

Interval: 3

Current user profile: DSO SQL: No Call stack: Yes

Priority: 56

Current wait: (102/Rsh) Rmslshared

Current wait duration: 443.398 milliseconds

Object waited on: ROBOT

Interval duration: 4.776 seconds

Holding thread or task: QJVACMDSRV / QPGMR / 446980: 00000043

Interval start: 2004-08-31-13.00.14.677000

Wait bucket statistics (only buckets with a time value greater than zero shown):

Bucket number	Description	Percent of Total Time	Time (seconds)	Total occurences	Average time (seconds)	Occurences per second	Current wait time
01	Cpu	3.82	.182204	462	.000394	96.74	
03	Cpu queueing	11.50	.549427	0	0	0	
06	Other waits	.30	.014387	5	.002877	1.05	
07	Mutex/sempahore	12.88	.614912	1	.614912	.21	
09	Seizes	9.58	.457503	14	.032679	2.93	443.398 millisecon
13	Dasd reads	61.81	2.951790	439	.006724	91.92	
14	Dasd writes	.11	.005480	3	.001827	.63	

This window contains the following fields:

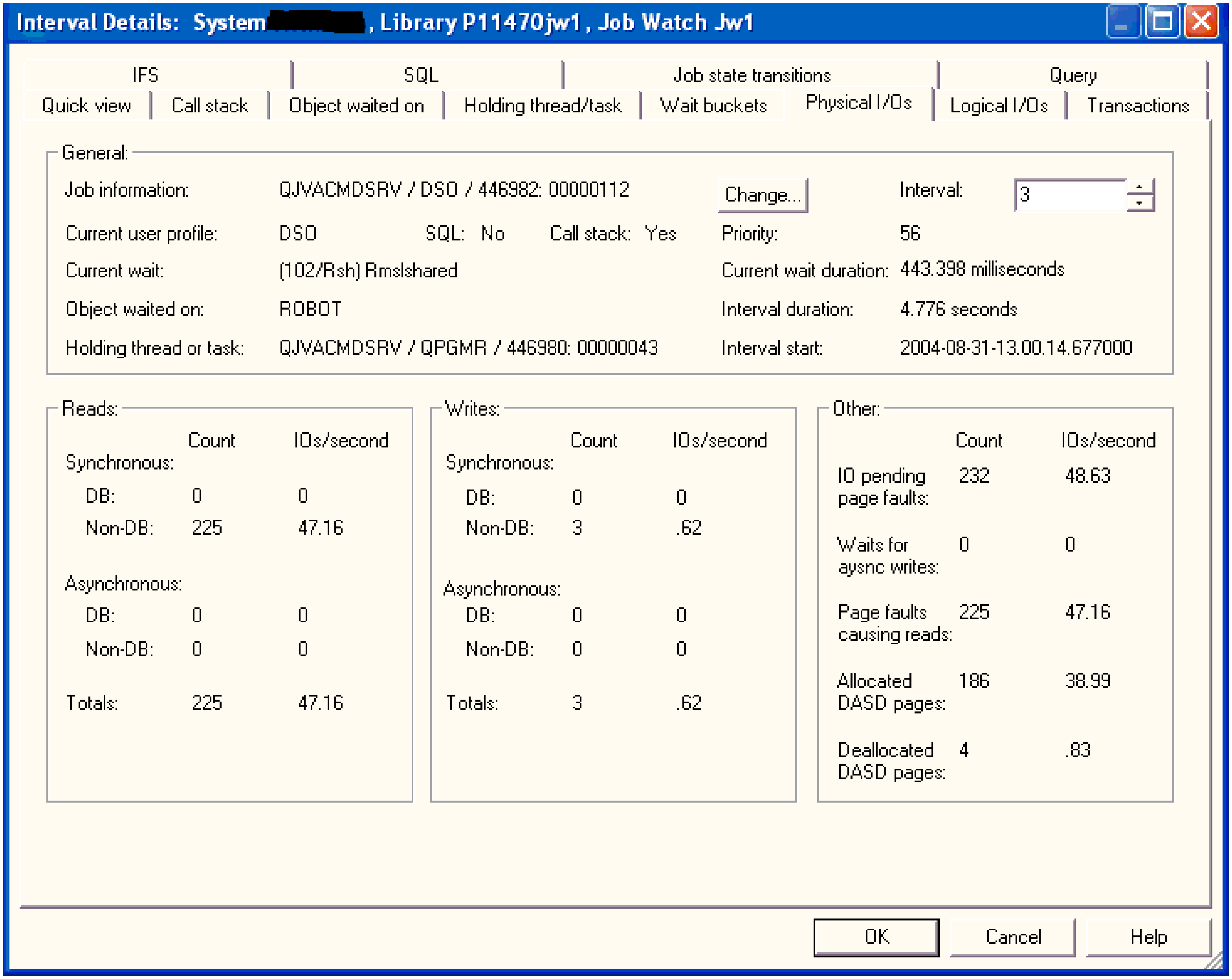
Field name	Description
Job information	The fully qualified job/thread id or task information this interface applies to. The Change... button allows the user to change the current job/thread/task to another. Changing the selected job will update the interface information on this panel but the information on the record quick view panel will still show the data for the original job.

Interval	Displays the selected interval the data shown applies to. The up and down buttons allow the user to change the selected interval. If the interval specified does not exist for the selected job no data will be shown.
Current user profile	The user profile currently active when the job was running at the interval specified.
Priority	The priority of the job for the specified interval.
Current wait	The current wait information contains information about the wait point (numeric identifier and 3 character wait code) and the description of the wait point from file QPYRTJWAT2. This field indicates the type of wait (or CPU) that was occurring when the snapshot to gather information about the job was taken. The wait point shown belongs to a wait bucket. Information about the current bucket is available on the "Wait buckets" panel in this interface.
Current wait duration	The total time the job has been in the current wait type. Everytime the wait type changes or CPU is used, this value is reset at 0. The current wait duration could be days for an idle job doing nothing and this would not indicate any problem. However if the current wait was a seize lock condition and the current wait duration was 30 seconds, this could flag a potential problem. For more information on understanding wait analysis, see the white paper available on the home page of the iDoctor for iSeries Web site.
Object waited on	This field contains information about the object the job/thread/task waited on for the current interval (if any) as well as the type of object being waited on. If an object is specified on this field, additional information about the object is available on the "Object waited on" panel in this interface
Interval duration	The time that elapsed when collecting the selected interval.
Holding thread or task	This field contains information about the job/thread/task that was holding the selected job during the selected interval and preventing it from doing work. If a holding thread or task is specified within this field, additional information about the holder is available on the "Holding thread/task" panel in this interface
Interval start	The date/time the current interval began for the specified job.
Wait bucket statistics	Contains the time spent in each type of wait as well as the number of occurrences of each wait bucket during the specified interval. The occurrence count goes up everytime a transition is made from one type of wait to another.

Table of Contents | Previous | Next

2.8.1.6 Physical I/Os

This window contains many details about the various types of physical I/Os that occurred for the specified job and interval. Statistics for reads and writes as well as page faults and disk space allocations/deallocations are provided on this panel.



Interval Details: System [redacted], Library P11470jw1, Job Watch Jw1

General:

Job information:	QJVACMDSRV / DSO / 446982: 00000112	Interval:	3
Current user profile:	DSO SQL: No Call stack: Yes	Priority:	56
Current wait:	(102/Rsh) Rmslshared	Current wait duration:	443.398 milliseconds
Object waited on:	ROBOT	Interval duration:	4.776 seconds
Holding thread or task:	QJVACMDSRV / QPGMR / 446980: 00000043	Interval start:	2004-08-31-13.00.14.677000

Reads:

	Count	IOs/second
Synchronous:		
DB:	0	0
Non-DB:	225	47.16
Asynchronous:		
DB:	0	0
Non-DB:	0	0
Totals:	225	47.16

Writes:

	Count	IOs/second
Synchronous:		
DB:	0	0
Non-DB:	3	.62
Asynchronous:		
DB:	0	0
Non-DB:	0	0
Totals:	3	.62

Other:

	Count	IOs/second
IO pending page faults:	232	48.63
Waits for aysnc writes:	0	0
Page faults causing reads:	225	47.16
Allocated DASD pages:	186	38.99
Deallocated DASD pages:	4	.83

This window contains the following fields:

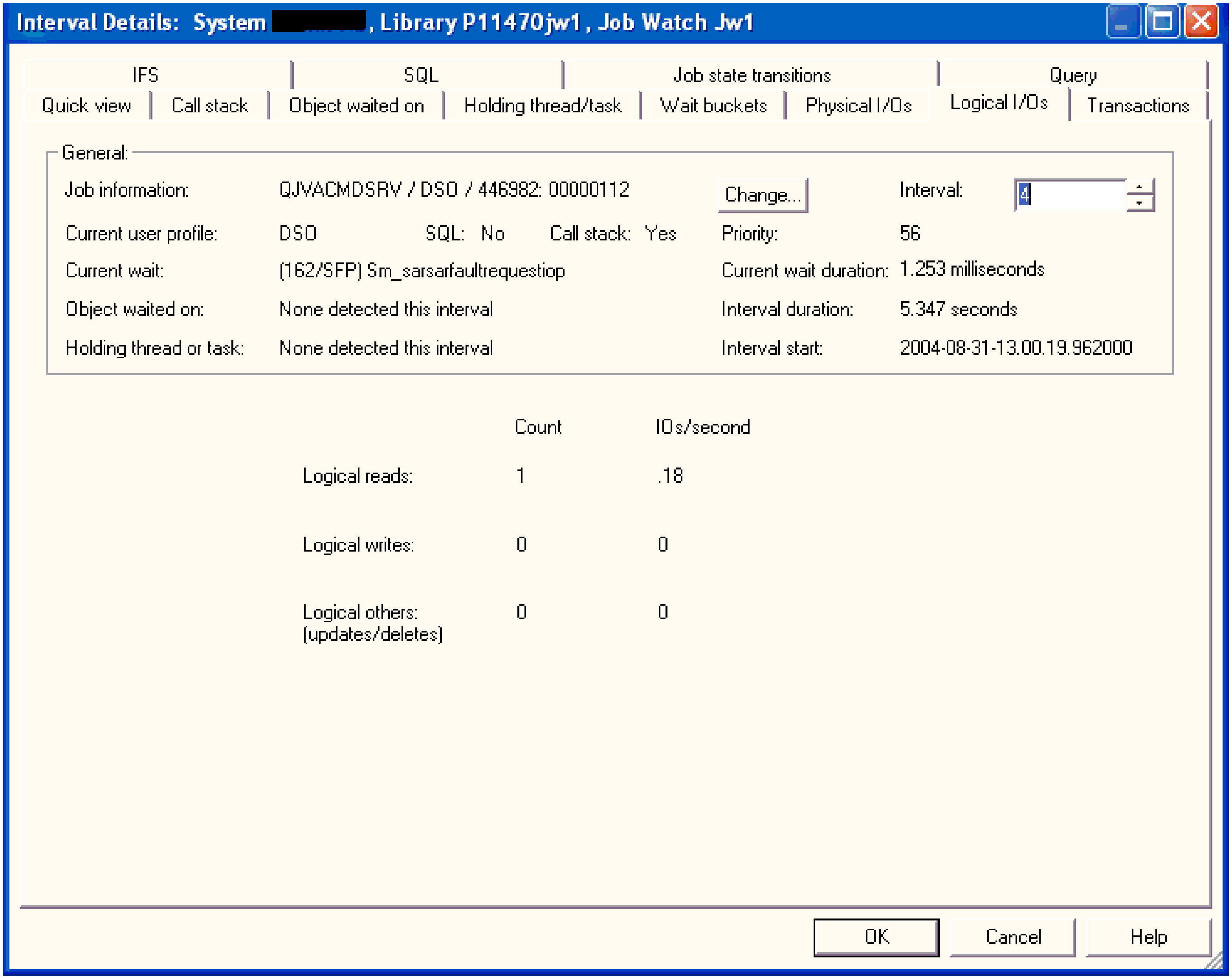
Field name	Description
Job information	The fully qualified job/thread id or task information this interface applies to. The Change... button allows the user to change the current job/thread/task to another. Changing the selected job will update the interface information on this panel but the information on the record quick view panel will still show the data for the original job.
Interval	Displays the selected interval the data shown applies to. The up and down buttons allow the user to change the selected interval. If the interval specified does not exist for the selected job no data will be shown.

Current user profile	The user profile currently active when the job was running at the interval specified.
Priority	The priority of the job for the specified interval.
Current wait	The current wait information contains information about the wait point (numeric identifier and 3 character wait code) and the description of the wait point from file QPYRTJWAT2. This field indicates the type of wait (or CPU) that was occurring when the snapshot to gather information about the job was taken. The wait point shown belongs to a wait bucket. Information about the current bucket is available on the "Wait buckets" panel in this interface.
Current wait duration	The total time the job has been in the current wait type. Everytime the wait type changes or CPU is used, this value is reset at 0. The current wait duration could be days for an idle job doing nothing and this would not indicate any problem. However if the current wait was a seize lock condition and the current wait duration was 30 seconds, this could flag a potential problem. For more information on understanding wait analysis, see the white paper available on the home page of the iDoctor for iSeries Web site.
Object waited on	This field contains information about the object the job/thread/task waited on for the current interval (if any) as well as the type of object being waited on. If an object is specified on this field, additional information about the object is available on the "Object waited on" panel in this interface
Interval duration	The time that elapsed when collecting the selected interval.
Holding thread or task	This field contains information about the job/thread/task that was holding the selected job during the selected interval and preventing it from doing work. If a holding thread or task is specified within this field, additional information about the holder is available on the "Holding thread/task" panel in this interface
Interval start	The date/time the current interval began for the specified job.

Table of Contents | Previous | Next

2.8.1.7 Logical I/Os

This page contains statistics for logical database I/Os occurred for the specified job and interval. The LDIO statistics includes rates and totals of reads, writes and updates/deletes.



Interval Details: System [redacted], Library P11470jw1, Job Watch Jw1

General:

Job information: QJVACMDSRV / DSO / 446982: 00000112

Interval: 4

Current user profile: DSO SQL: No Call stack: Yes

Priority: 56

Current wait: (162/SFP) Sm_sarsarfaultrequestiop

Current wait duration: 1.253 milliseconds

Object waited on: None detected this interval

Interval duration: 5.347 seconds

Holding thread or task: None detected this interval

Interval start: 2004-08-31-13.00.19.962000

	Count	IOs/second
Logical reads:	1	.18
Logical writes:	0	0
Logical others: (updates/deletes)	0	0

This window contains the following fields:

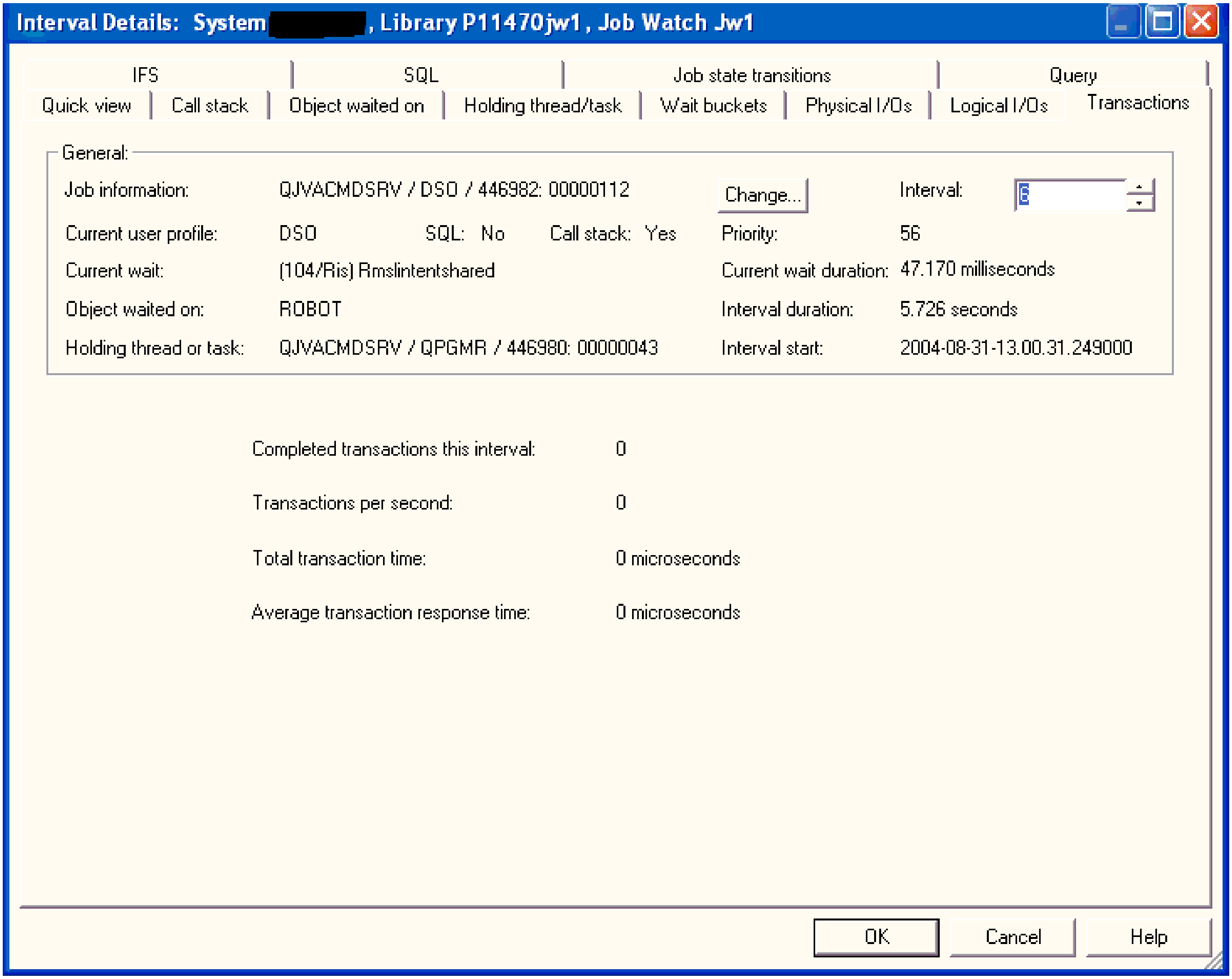
Field name	Description
Job information	The fully qualified job/thread id or task information this interface applies to. The Change... button allows the user to change the current job/thread/task to another. Changing the selected job will update the interface information on this panel but the information on the record quick view panel will still show the data for the original job.
Interval	Displays the selected interval the data shown applies to. The up and down buttons allow the user to change the selected interval. If the interval specified does not exist for the selected job no data will be shown.

Current user profile	The user profile currently active when the job was running at the interval specified.
Priority	The priority of the job for the specified interval.
Current wait	The current wait information contains information about the wait point (numeric identifier and 3 character wait code) and the description of the wait point from file QPYRTJWAT2. This field indicates the type of wait (or CPU) that was occurring when the snapshot to gather information about the job was taken. The wait point shown belongs to a wait bucket. Information about the current bucket is available on the "Wait buckets" panel in this interface.
Current wait duration	The total time the job has been in the current wait type. Everytime the wait type changes or CPU is used, this value is reset at 0. The current wait duration could be days for an idle job doing nothing and this would not indicate any problem. However if the current wait was a seize lock condition and the current wait duration was 30 seconds, this could flag a potential problem. For more information on understanding wait analysis, see the white paper available on the home page of the iDoctor for iSeries Web site.
Object waited on	This field contains information about the object the job/thread/task waited on for the current interval (if any) as well as the type of object being waited on. If an object is specified on this field, additional information about the object is available on the "Object waited on" panel in this interface
Interval duration	The time that elapsed when collecting the selected interval.
Holding thread or task	This field contains information about the job/thread/task that was holding the selected job during the selected interval and preventing it from doing work. If a holding thread or task is specified within this field, additional information about the holder is available on the "Holding thread/task" panel in this interface
Interval start	The date/time the current interval began for the specified job.

Table of Contents Previous Next

2.8.1.8 Transactions

This page of the interval details window provides statistics on the transactions that ocurred during the specified job and interval. The total number of completed transactions, transaction rate, and transaction times are provided on this window.


Interval Details: System [redacted], Library P11470jw1, Job Watch Jw1

General:

Job information: QJVACMDSRV / DSO / 446982: 00000112

Interval: 8

Current user profile: DSO SQL: No Call stack: Yes

Priority: 56

Current wait: (104/Ris) Rmslintentshared

Current wait duration: 47.170 milliseconds

Object waited on: ROBOT

Interval duration: 5.726 seconds

Holding thread or task: QJVACMDSRV / QPGMR / 446980: 00000043

Interval start: 2004-08-31-13.00.31.249000

Completed transactions this interval: 0

Transactions per second: 0

Total transaction time: 0 microseconds

Average transaction response time: 0 microseconds

This window contains the following fields:

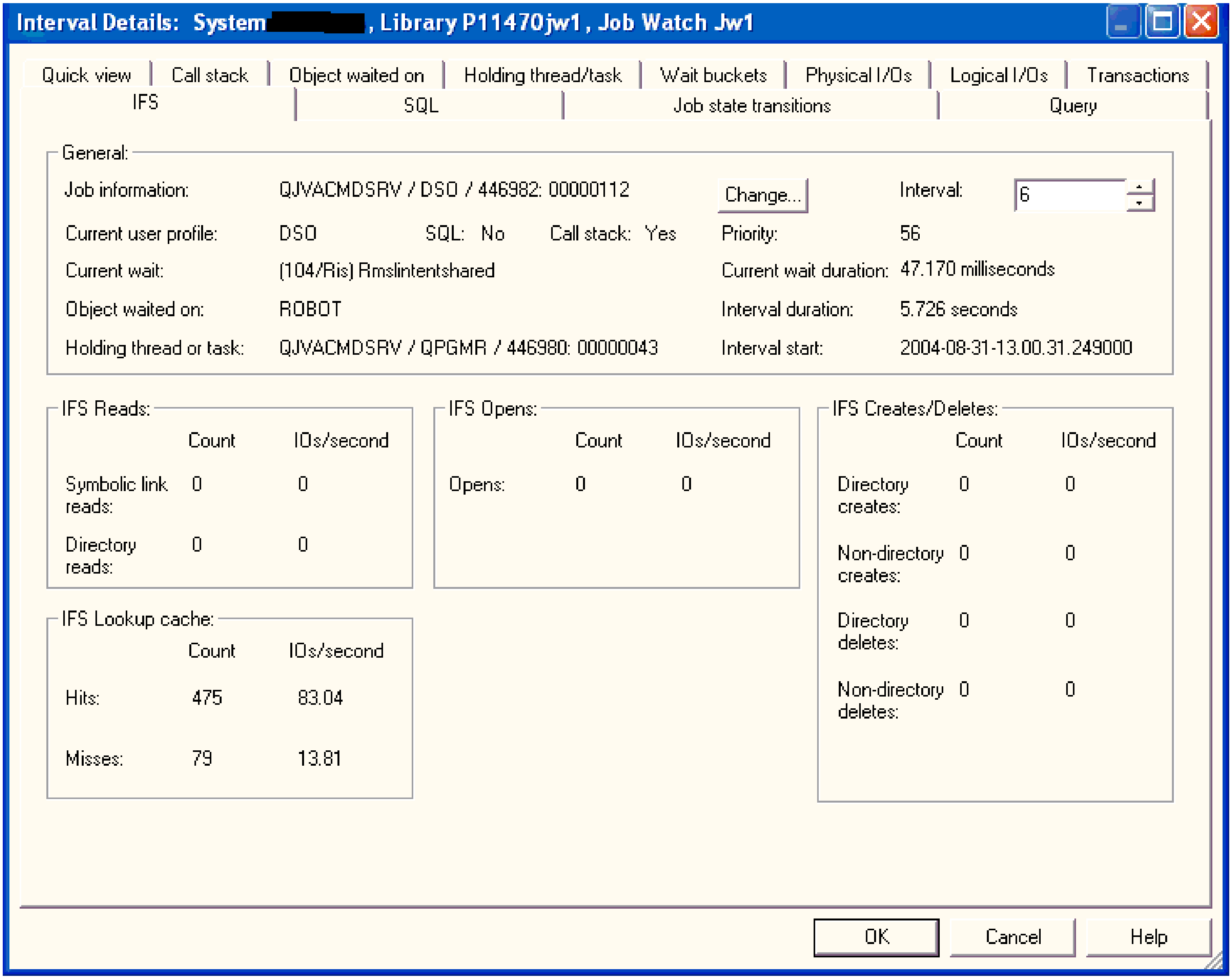
Field name	Description
Job information	The fully qualified job/thread id or task information this interface applies to. The Change... button allows the user to change the current job/thread/task to another. Changing the selected job will update the interface information on this panel but the information on the record quick view panel will still show the data for the original job.
Interval	Displays the selected interval the data shown applies to. The up and down buttons allow the user to change the selected interval. If the interval specified does not exist for the selected job no data will be shown.

Current user profile	The user profile currently active when the job was running at the interval specified.
Priority	The priority of the job for the specified interval.
Current wait	The current wait information contains information about the wait point (numeric identifier and 3 character wait code) and the description of the wait point from file QPYRTJWAT2. This field indicates the type of wait (or CPU) that was occurring when the snapshot to gather information about the job was taken. The wait point shown belongs to a wait bucket. Information about the current bucket is available on the "Wait buckets" panel in this interface.
Current wait duration	The total time the job has been in the current wait type. Everytime the wait type changes or CPU is used, this value is reset at 0. The current wait duration could be days for an idle job doing nothing and this would not indicate any problem. However if the current wait was a seize lock condition and the current wait duration was 30 seconds, this could flag a potential problem. For more information on understanding wait analysis, see the white paper available on the home page of the iDoctor for iSeries Web site.
Object waited on	This field contains information about the object the job/thread/task waited on for the current interval (if any) as well as the type of object being waited on. If an object is specified on this field, additional information about the object is available on the "Object waited on" panel in this interface
Interval duration	The time that elapsed when collecting the selected interval.
Holding thread or task	This field contains information about the job/thread/task that was holding the selected job during the selected interval and preventing it from doing work. If a holding thread or task is specified within this field, additional information about the holder is available on the "Holding thread/task" panel in this interface
Interval start	The date/time the current interval began for the specified job.

Table of Contents Previous Next

2.8.1.9 IFS

This page of the interval details window provides statistics on IFS activity that occurred during the specified job and interval. IFS reads, opens, creates/deletes and lookup cache hits and misses are provided on this window.



Interval Details: System [redacted], Library P11470jw1, Job Watch Jw1

General:

Job information: QJVACMDSRV / DSO / 446982: 00000112

Interval: 6

Current user profile: DSO SQL: No Call stack: Yes

Priority: 56

Current wait: (104/Ris) Rmslintentshared

Current wait duration: 47.170 milliseconds

Object waited on: ROBOT

Interval duration: 5.726 seconds

Holding thread or task: QJVACMDSRV / QPGMR / 446980: 00000043

Interval start: 2004-08-31-13.00.31.249000

IFS Reads:

	Count	IOs/second
Symbolic link reads:	0	0
Directory reads:	0	0

IFS Opens:

	Count	IOs/second
Opens:	0	0

IFS Creates/Deletes:

	Count	IOs/second
Directory creates:	0	0
Non-directory creates:	0	0
Directory deletes:	0	0
Non-directory deletes:	0	0

IFS Lookup cache:

	Count	IOs/second
Hits:	475	83.04
Misses:	79	13.81

This window contains the following fields:

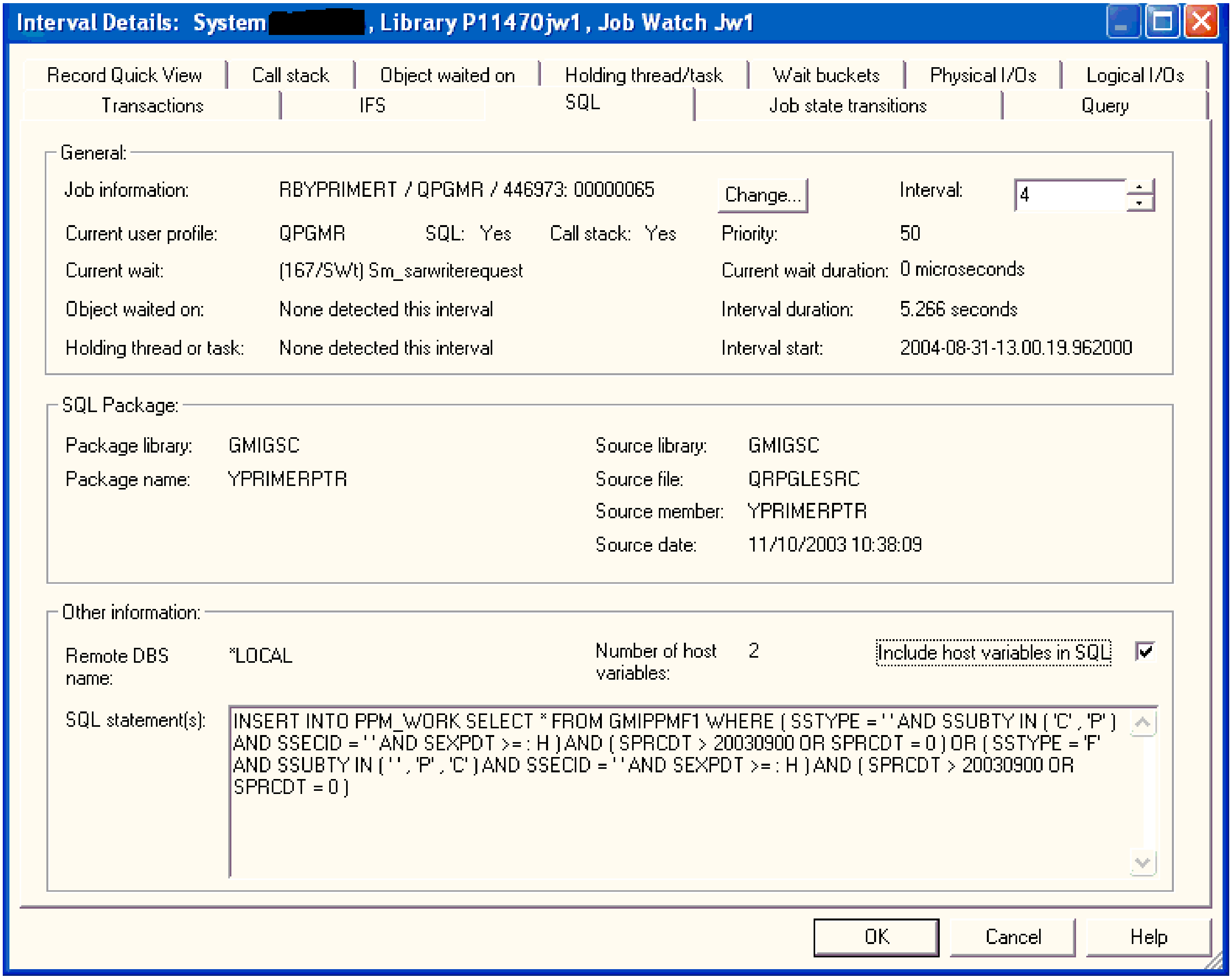
Field name	Description
Job information	The fully qualified job/thread id or task information this interface applies to. The Change... button allows the user to change the current job/thread/task to another. Changing the selected job will update the interface information on this panel but the information on the record quick view panel will still show the data for the original job.
Interval	Displays the selected interval the data shown applies to. The up and down buttons allow the user to change the selected interval. If the interval specified does not exist for the selected job no data will be shown.

Current user profile	The user profile currently active when the job was running at the interval specified.
Priority	The priority of the job for the specified interval.
Current wait	The current wait information contains information about the wait point (numeric identifier and 3 character wait code) and the description of the wait point from file QPYRTJWAT2. This field indicates the type of wait (or CPU) that was occurring when the snapshot to gather information about the job was taken. The wait point shown belongs to a wait bucket. Information about the current bucket is available on the "Wait buckets" panel in this interface.
Current wait duration	The total time the job has been in the current wait type. Everytime the wait type changes or CPU is used, this value is reset at 0. The current wait duration could be days for an idle job doing nothing and this would not indicate any problem. However if the current wait was a seize lock condition and the current wait duration was 30 seconds, this could flag a potential problem. For more information on understanding wait analysis, see the white paper available on the home page of the iDoctor for iSeries Web site.
Object waited on	This field contains information about the object the job/thread/task waited on for the current interval (if any) as well as the type of object being waited on. If an object is specified on this field, additional information about the object is available on the "Object waited on" panel in this interface
Interval duration	The time that elapsed when collecting the selected interval.
Holding thread or task	This field contains information about the job/thread/task that was holding the selected job during the selected interval and preventing it from doing work. If a holding thread or task is specified within this field, additional information about the holder is available on the "Holding thread/task" panel in this interface
Interval start	The date/time the current interval began for the specified job.

Table of Contents | Previous | Next

2.8.1.10 SQL

This page of the interval details window provides the SQL statement running at the time the snapshot was taken for the job in the system-wide job watch.



Interval Details: System [redacted], Library P11470jw1, Job Watch Jw1

General:

Job information: RBYPRIMERT / QPGMR / 446973: 00000065

Interval: 4

Current user profile: QPGMR SQL: Yes Call stack: Yes

Priority: 50

Current wait: (167/SWt) Sm_sarwriterequest

Current wait duration: 0 microseconds

Object waited on: None detected this interval

Interval duration: 5.266 seconds

Holding thread or task: None detected this interval

Interval start: 2004-08-31-13.00.19.962000

SQL Package:

Package library: GMIGSC

Package name: YPRIMERPTR

Source library: GMIGSC

Source file: QRPGLESRC

Source member: YPRIMERPTR

Source date: 11/10/2003 10:38:09

Other information:

Remote DBS name: *LOCAL

Number of host variables: 2

Include host variables in SQL ☑

SQL statement(s):

```
INSERT INTO PPM_WORK SELECT * FROM GMIPPMF1 WHERE ( SSTYPE = ' ' AND SSUBTY IN ( 'C' , 'P' )
AND SSECID = ' ' AND SEXPDT >= : H ) AND ( SPRCDT > 20030900 OR SPRCDT = 0 ) OR ( SSTYPE = 'F'
AND SSUBTY IN ( ' ' , 'P' , 'C' ) AND SSECID = ' ' AND SEXPDT >= : H ) AND ( SPRCDT > 20030900 OR
SPRCDT = 0 )
```

This window contains the following fields:

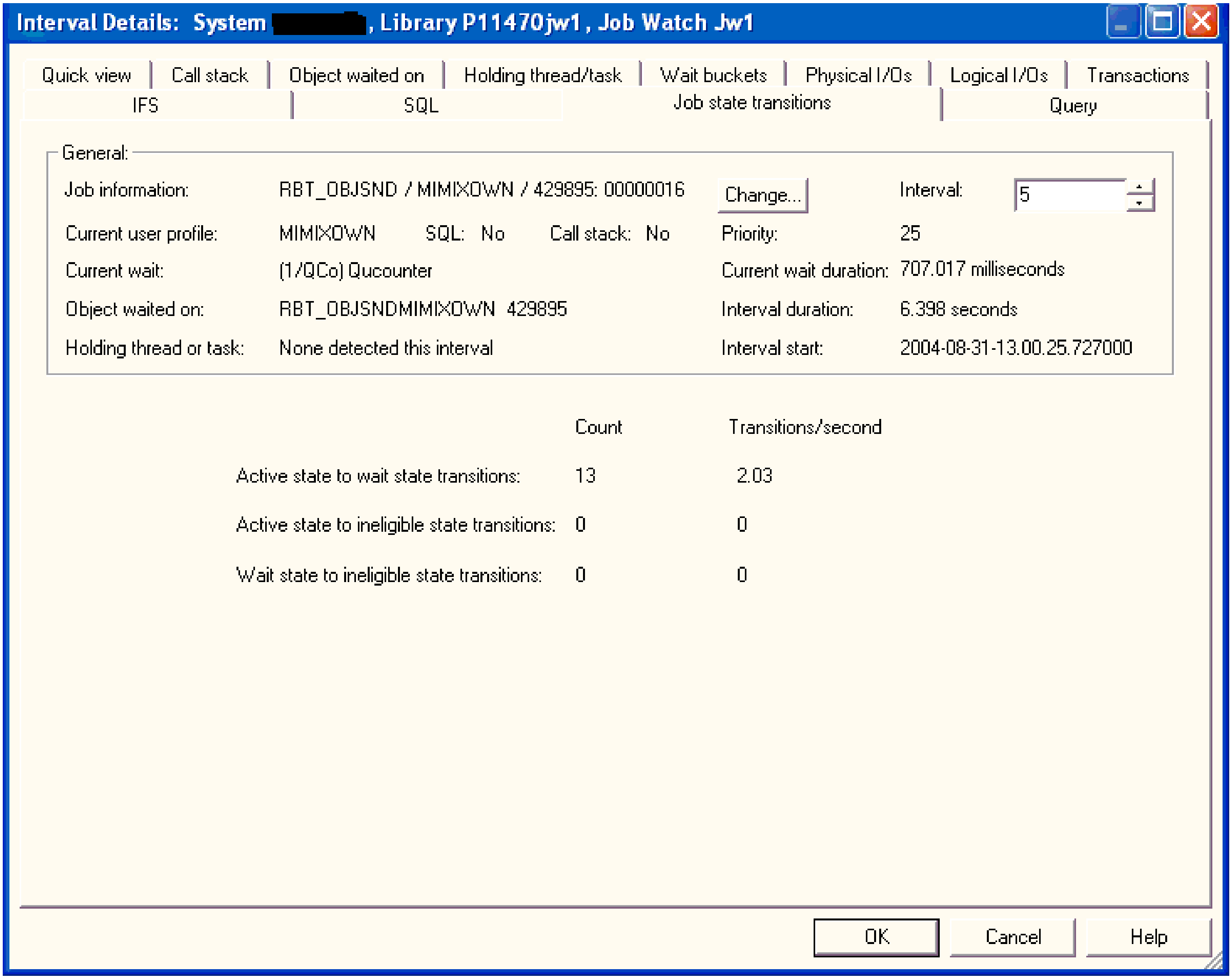
Field name	Description
Job information	The fully qualified job/thread id or task information this interface applies to. The Change... button allows the user to change the current job/thread/task to another. Changing the selected job will update the interface information on this panel but the information on the record quick view panel will still show the data for the original job.
Interval	Displays the selected interval the data shown applies to. The up and down buttons allow the user to change the selected interval. If the interval specified does not exist for the selected job no data will be shown.

Current user profile	The user profile currently active when the job was running at the interval specified.
Priority	The priority of the job for the specified interval.
Current wait	The current wait information contains information about the wait point (numeric identifier and 3 character wait code) and the description of the wait point from file QPYRTJWAT2. This field indicates the type of wait (or CPU) that was occurring when the snapshot to gather information about the job was taken. The wait point shown belongs to a wait bucket. Information about the current bucket is available on the "Wait buckets" panel in this interface.
Current wait duration	The total time the job has been in the current wait type. Everytime the wait type changes or CPU is used, this value is reset at 0. The current wait duration could be days for an idle job doing nothing and this would not indicate any problem. However if the current wait was a seize lock condition and the current wait duration was 30 seconds, this could flag a potential problem. For more information on understanding wait analysis, see the white paper available on the home page of the iDoctor for iSeries Web site.
Object waited on	This field contains information about the object the job/thread/task waited on for the current interval (if any) as well as the type of object being waited on. If an object is specified on this field, additional information about the object is available on the "Object waited on" panel in this interface
Interval duration	The time that elapsed when collecting the selected interval.
Holding thread or task	This field contains information about the job/thread/task that was holding the selected job during the selected interval and preventing it from doing work. If a holding thread or task is specified within this field, additional information about the holder is available on the "Holding thread/task" panel in this interface
Interval start	The date/time the current interval began for the specified job.

Table of Contents | Previous | Next

2.8.1.11 Job state transitions

This page of the interval details window provides statistics on the job state transitions that ocurred during the specified job and interval.



Interval Details: System [redacted], Library P11470jw1, Job Watch Jw1

Job information: RBT_OBJSND / MIMIXOWN / 429895: 00000016

Interval: 5

Current user profile: MIMIXOWN SQL: No Call stack: No

Priority: 25

Current wait: (1/QCo) Qucounter

Current wait duration: 707.017 milliseconds

Object waited on: RBT_OBJSNDMIMIXOWN 429895

Interval duration: 6.398 seconds

Holding thread or task: None detected this interval

Interval start: 2004-08-31-13.00.25.727000

	Count	Transitions/second
Active state to wait state transitions:	13	2.03
Active state to ineligible state transitions:	0	0
Wait state to ineligible state transitions:	0	0

This window contains the following fields:

Field name	Description
Job information	The fully qualified job/thread id or task information this interface applies to. The Change... button allows the user to change the current job/thread/task to another. Changing the selected job will update the interface information on this panel but the information on the record quick view panel will still show the data for the original job.
Interval	Displays the selected interval the data shown applies to. The up and down buttons allow the user to change the selected interval. If the interval specified does not exist for the selected job no data will be shown.

Current user profile	The user profile currently active when the job was running at the interval specified.
Priority	The priority of the job for the specified interval.
Current wait	The current wait information contains information about the wait point (numeric identifier and 3 character wait code) and the description of the wait point from file QPYRTJWAT2. This field indicates the type of wait (or CPU) that was occurring when the snapshot to gather information about the job was taken. The wait point shown belongs to a wait bucket. Information about the current bucket is available on the "Wait buckets" panel in this interface.
Current wait duration	The total time the job has been in the current wait type. Everytime the wait type changes or CPU is used, this value is reset at 0. The current wait duration could be days for an idle job doing nothing and this would not indicate any problem. However if the current wait was a seize lock condition and the current wait duration was 30 seconds, this could flag a potential problem. For more information on understanding wait analysis, see the white paper available on the home page of the iDoctor for iSeries Web site.
Object waited on	This field contains information about the object the job/thread/task waited on for the current interval (if any) as well as the type of object being waited on. If an object is specified on this field, additional information about the object is available on the "Object waited on" panel in this interface
Interval duration	The time that elapsed when collecting the selected interval.
Holding thread or task	This field contains information about the job/thread/task that was holding the selected job during the selected interval and preventing it from doing work. If a holding thread or task is specified within this field, additional information about the holder is available on the "Holding thread/task" panel in this interface
Interval start	The date/time the current interval began for the specified job.

Table of Contents Previous Next

2.8.2 Wait graphs - detailed

The detailed wait graphs provide information about waiting that occurred in a job within a system-wide Job Watch.

Table of Contents Previous Next

2.8.2.1 Run/wait time signature

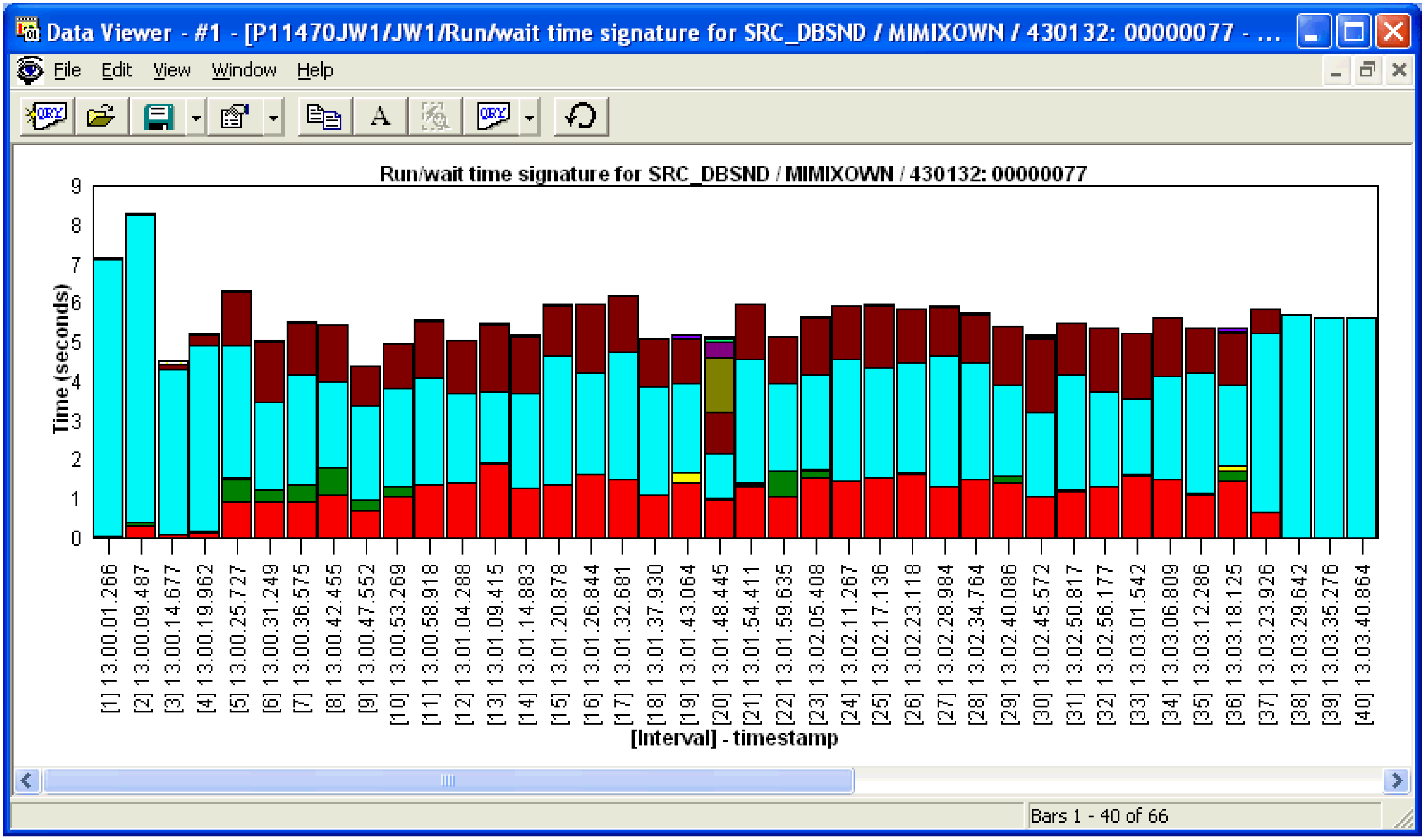
Description: This graph shows a **detailed** look at the time a job spent in various types of waits throughout its existence during the job watch. The graph makes use of the 32 run/wait buckets. To determine which type of wait a color represents, place the mouse pointer of a color/bar of interest and a description of the wait as well as other statistics about the interval/wait will be displayed. Due to screen space limitations and the high number of colors the legend is not shown by default but it is available by using the Show Legend option on the Graph Definition -> General page.

This graph can help pinpoint within a specific job where the job started to perform work and how efficiently it was performing that work. If a interval in the graph suddenly becomes suspicious (high seize time for example) clicking on the bar will provide other valuable information through the interval details window.

The total height of the stacked bar (which includes all waits for that interval and CPU) indicates the elapsed interval time.

Graph Type: detailed system-wide (stacked vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Each color represents the amount of time the job/thread/task spent in one of the 32 different wait buckets within an interval. Time is listed in seconds.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Display Stack	Shows the interval details window and brings focus to the call stack tab. The call stack may not always be available if it was indicated at collection time to not collect the call stack. In this case the call stack window will not show a call stack but other information about the job's performance during this interval will be available.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.

Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in the interval details window. This window will also appear by left-clicking on any bar in the graph.

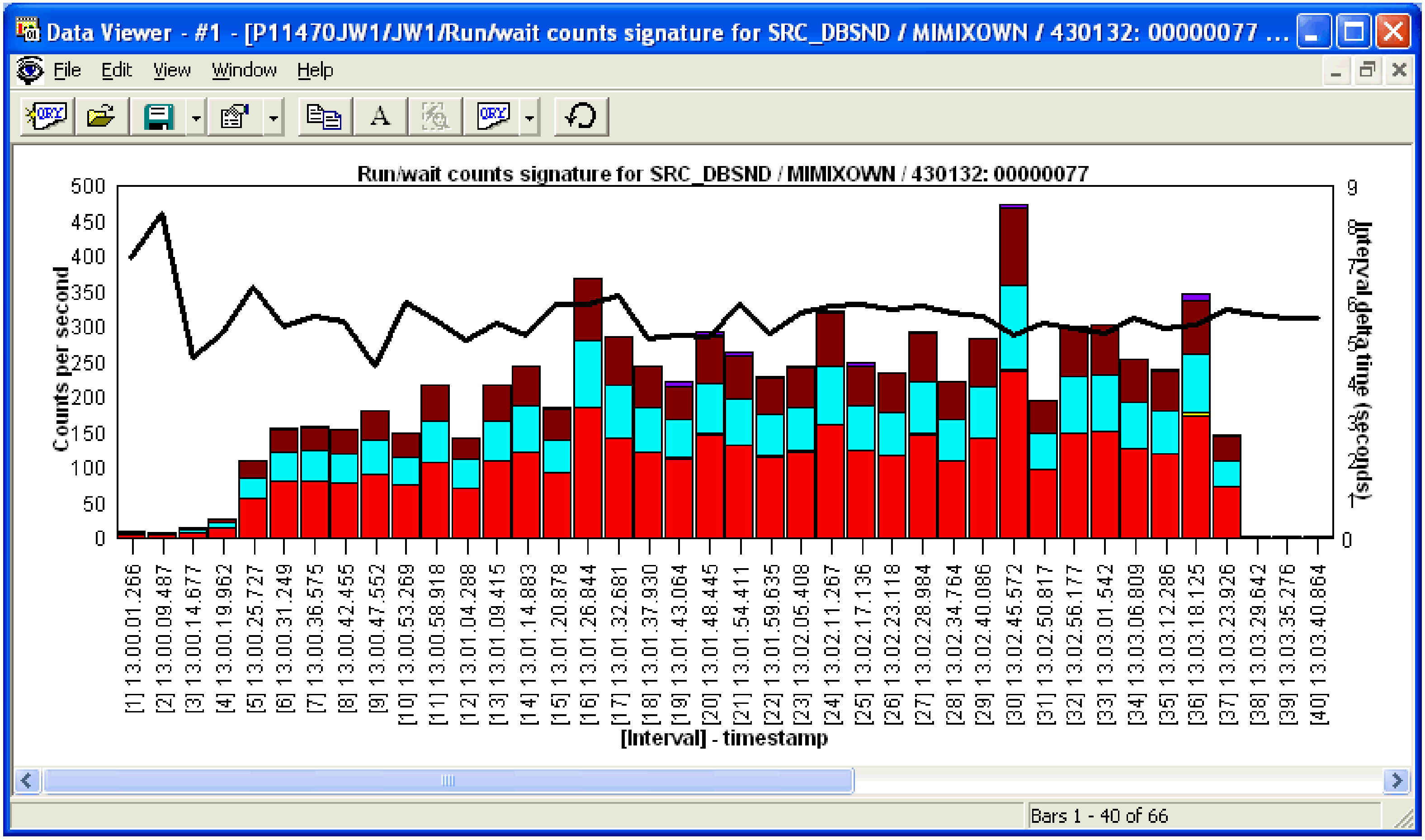
2.8.2.2 Run/wait counts signature

Description: This graph shows a **detailed** look at the transitions from one wait type to another throughout a job's existence during the job watch. The graph makes use of the counts provided by the 32 run/wait buckets. To determine which type of wait a color represents, place the mouse pointer of a color/bar of interest and a description of the wait as well as other statistics about the interval/wait will be displayed. Due to screen space limitations and the high number of colors the legend is not shown by default but it is available by using the Show Legend option on the Graph Definition -> General page.

This graph can help pinpoint within a specific job where things happened where a job was idle vs not idle. If a job is stuck in the way wait for an entire interval the size of the bar will be very small (1 count per interval time). On the other hand if the job is performing work the counts will be noticeably high.

Graph Type: detailed system-wide (stacked vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Each color represents the amount of time the job/thread/task spent in one of the 32 different wait buckets within an interval. Time is listed in seconds.
Second Y-Axis: This line represents the elapsed time for each interval in seconds. The interval time could vary depending on the overall performance of the system which can cause delays in the Job Watcher engine. For example, these delays may cause an interval of 1 second to actually take 1.5 seconds to complete.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Display Stack	Shows the interval details window and brings focus to the call stack tab. The call stack may not always be available if it was indicated at collection time to not collect the call stack. In this case the call stack window will not show a call stack but other information about the job's performance during this interval will be available.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.

Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in the interval details window. This window will also appear by left-clicking on any bar in the graph.

Table of Contents | Previous | Next

2.8.2.3 Objected waited on

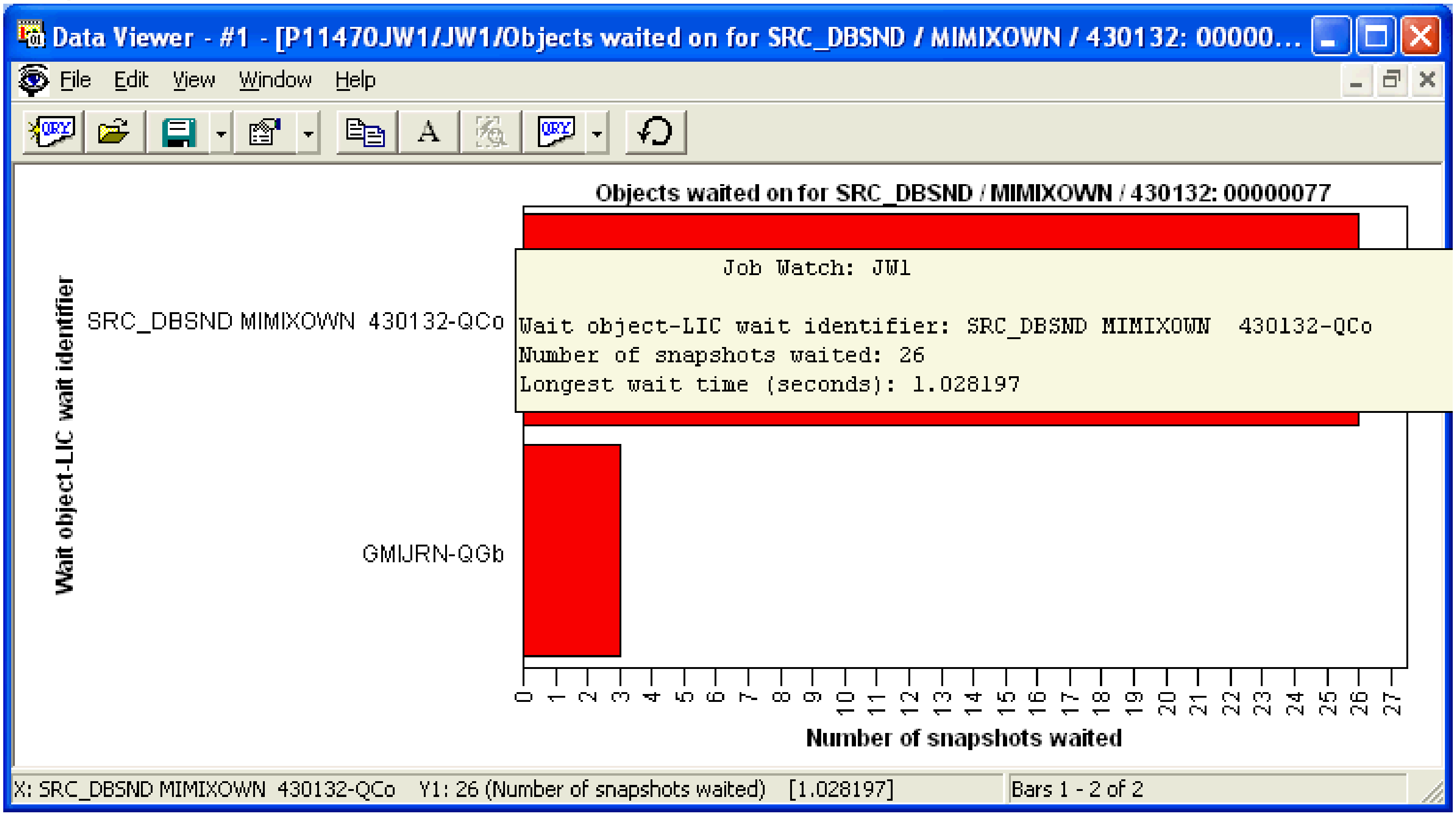
Description: This graph shows the list of objects a job has waited on throughout its existence during the job watch. The graph shows the number of snapshots taken where the job was waiting to use an object, the name/type information for each object and the type of wait.

The objects waited on does not necessarily indicate a problem, depending on the type of work the job is performing as well as what is normal for the application. However in other cases especially situations where a program is waiting to use a file, the objected waited on feature can help identify locking conflicts.

Placing the mouse over a bar will also show the longest chunk of time spent waiting on the object during the collection.

Graph Type: detailed system-wide (horizontal bar)
X-axis: The name of the object being waited on, and the 3 character LIC wait identifier showing the type of wait that occurred. 1 bar is created per object being waited on for the job.
Y-Axis: The number of snapshots taken where the object waited on was the same.

Example:


Data Viewer - #1 - [P11470JW1/JW1/Objects waited on for SRC_DBSND / MIMIXOWN / 430132: 00000...
File
Edit
View
Window
Help
Objects waited on for SRC_DBSND / MIMIXOWN / 430132: 00000077
Wait object-LIC wait identifier
SRC_DBSND MIMIXOWN 430132-QCo
GMIJRN-QGb
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
Number of snapshots waited

Job Watch: JW1

Wait object-LIC wait identifier: SRC_DBSND MIMIXOWN 430132-QCo
Number of snapshots waited: 26
Longest wait time (seconds): 1.028197

X: SRC_DBSND MIMIXOWN 430132-QCo Y1: 26 (Number of snapshots waited) [1.028197] Bars 1 - 2 of 2

Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Display Stack	Not available for this graph.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.

Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window.

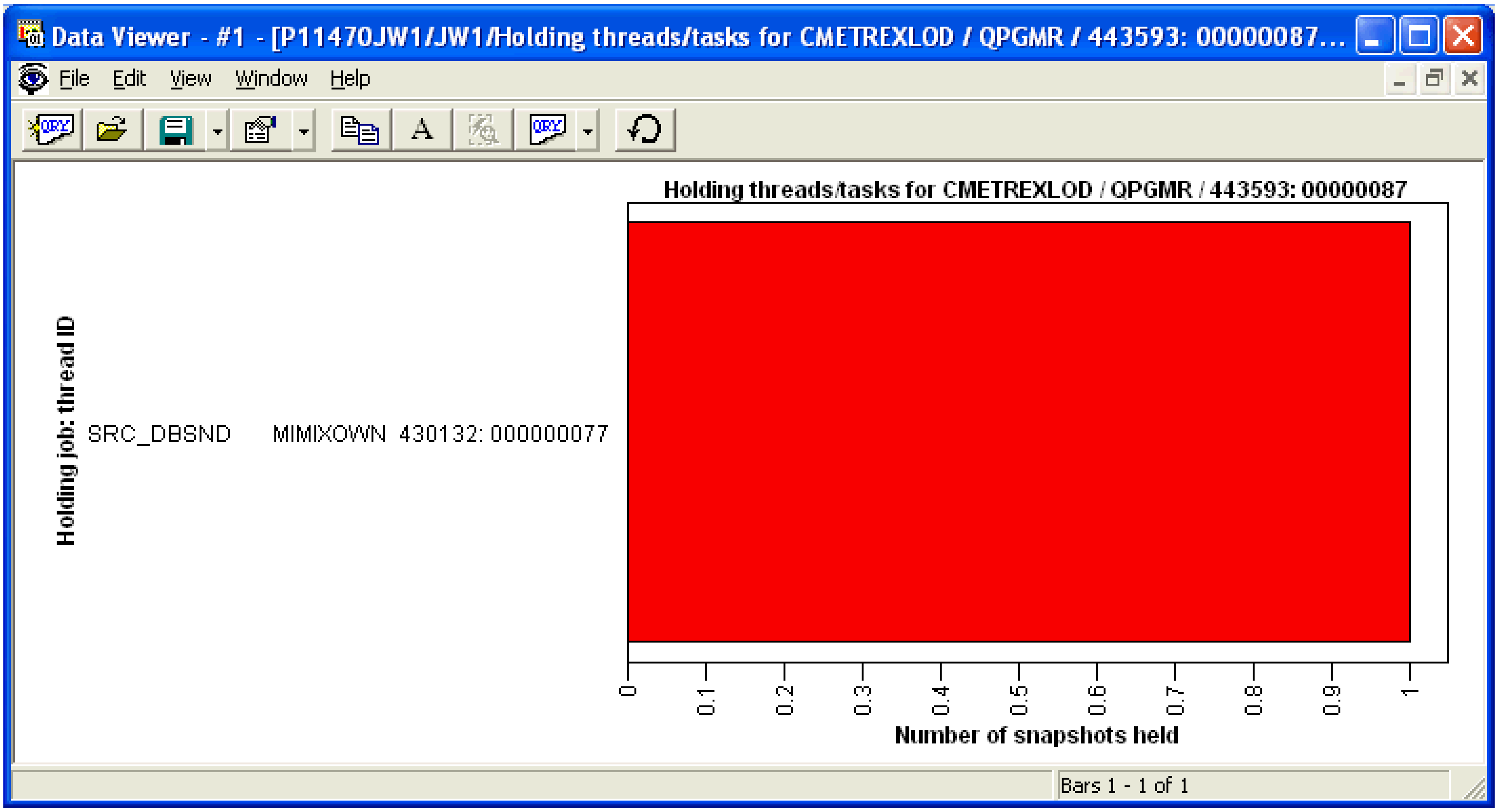
Table of Contents Previous Next

2.8.2.4 Holding threads/tasks

Description: This graph displays a list of jobs/threads/tasks that were holding a specific job in the job watch.

Graph Type: detailed system-wide (horizontal bar)
X-axis: Each bar represents a job that was holding the job listed in the title of the graph (in the example below the holder is SRC_DBSND and the waiter job is CMETREXLOD)
Y-Axis: The number of snapshots where the holder was detected. This does NOT mean the job was held for this many intervals consecutively. This value is a count of how many times the holder was detected for the waiting job.

Example:


Data Viewer - #1 - [P11470JW1/JW1/Holding threads/tasks for CMETREXLOD / QPGMR / 443593: 00000087...
File Edit View Window Help
Holding threads/tasks for CMETREXLOD / QPGMR / 443593: 00000087
Holding job: thread ID
SRC_DBSND MIMIXOWN 430132: 000000077
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Number of snapshots held
Bars 1 - 1 of 1

Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Display Stack	Not available for this graph.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in a properties window.

Table of Contents Previous Next

2.8.3 CPU graphs - detailed

The detailed CPU graphs provide statistics for the CPU and CPU queueing that occurred for a job in a system-wide Job Watch.

Table of Contents Previous Next

2.8.3.1 CPU/CPUq time signature

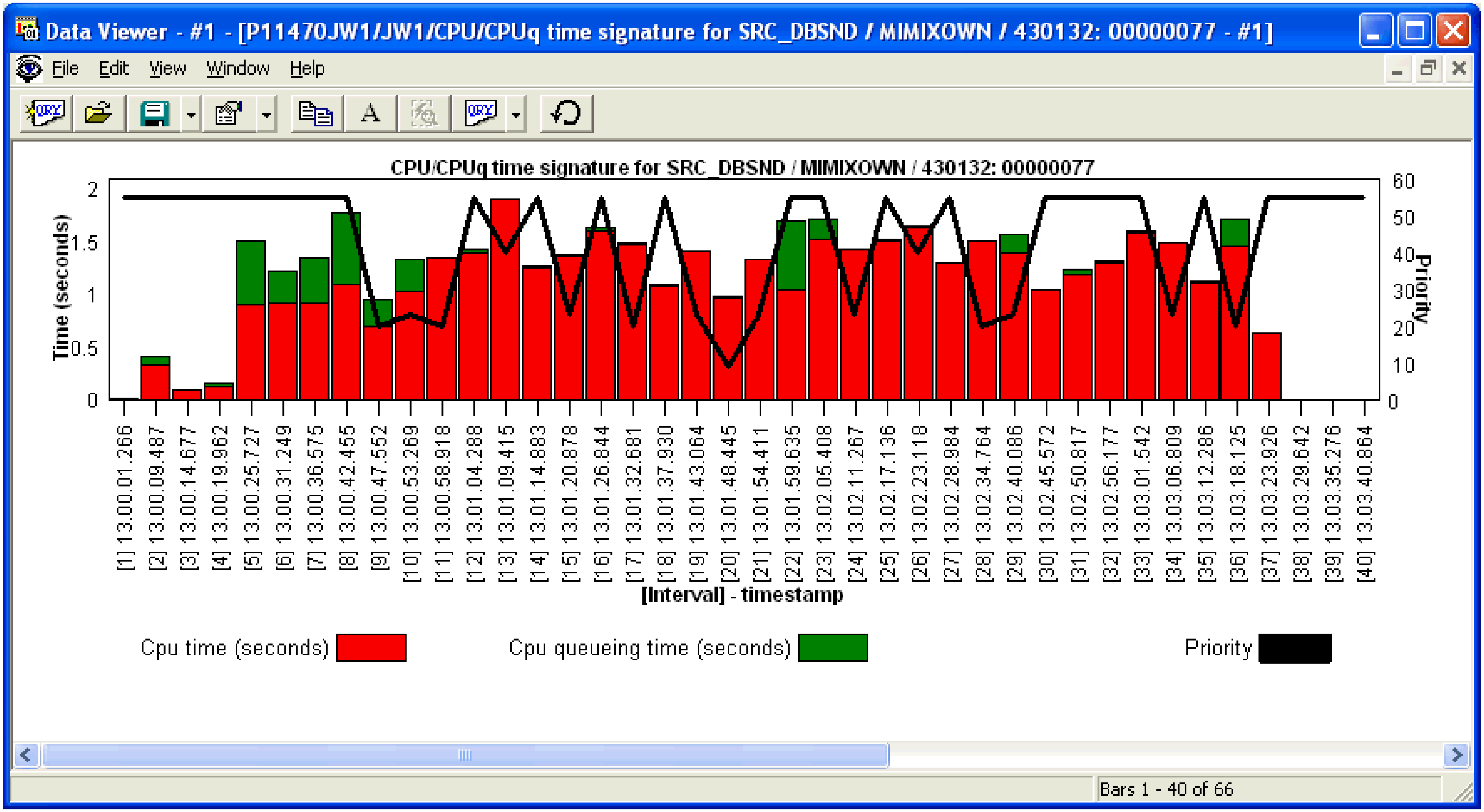
Description: This graph shows a **detailed** look at the time the job spent using CPU and waiting for CPU throughout its existence during the job watch. Red colors indicate CPU use, and green colors represent CPU queueing time.

This graph can help locate when high CPU, or CPU queueing is occurring for a specific job.

Graph Type: detailed system-wide (stacked vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Each color represents the amount of time the job/thread/task spent using CPU or CPU queueing. All times are listed in seconds.
Second Y-Axis: The line shows the job's priority during the collection.

Example:

2.8.3.1 CPU/CPUq time signature



Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Display Stack	Shows the interval details window and brings focus to the call stack tab. The call stack may not always be available if it was indicated at collection time to not collect the call stack. In this case the call stack window will not show a call stack but other information about the job's performance during this interval will be available.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.

Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in the interval details window. This window will also appear by left-clicking on any bar in the graph.

Table of Contents | Previous | Next

2.8.4 DASD/IO graphs - detailed

The detailed DASD/IO graphs provide statistics for disk space allocation/deallocations, as well all physical and logical IOs that occurred for a job in a system-wide Job Watch.

Table of Contents | Previous | Next

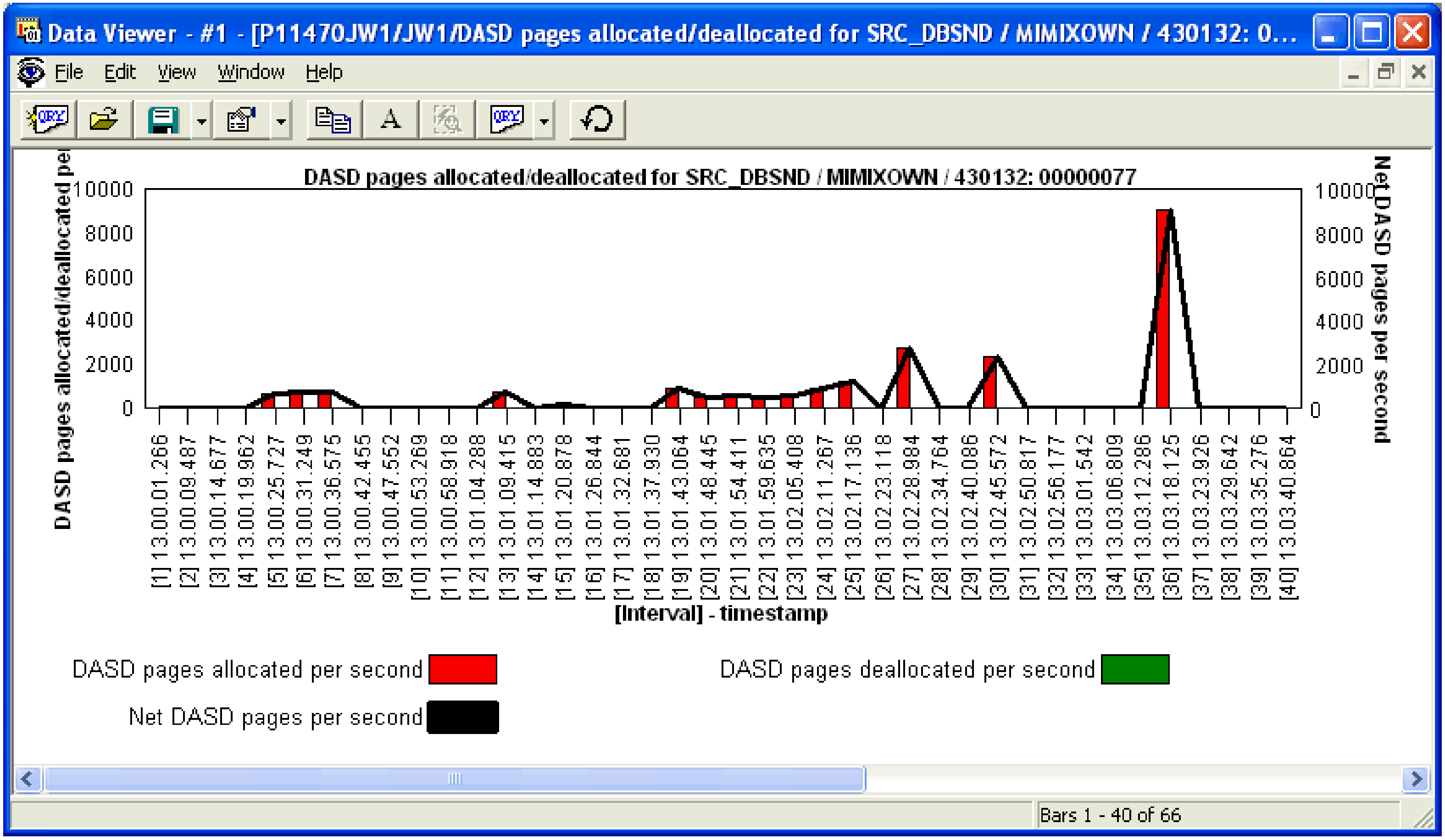
2.8.4.1 DASD pages allocated/deallocated

Description: This graph shows a **detailed** look at the amount of disk space allocated and deallocated by a job throughout its existence during the job watch. Red colors indicate space allocations and green colors indicate space deallocations. Allocations and deallocations listed are the number of 4K dasd pages allocated/deallocated per second. A "DASD page" is a 4k (4096 bytes) block of data.

This graph can help locate where high CPU, or CPU queueing is occurring for a specific job.

Graph Type: detailed system-wide (vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Contains the rate of DASD pages allocations and deallocations for each interval.
Second Y-Axis: This line represents the rate of allocations minus the rate of deallocations per interval (the net dasd pages per second)

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Display Stack	Shows the interval details window and brings focus to the call stack tab. The call stack may not always be available if it was indicated at collection time to not collect the call stack. In this case the call stack window will not show a call stack but other information about the job's performance during this interval will be available.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.

Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in the interval details window. This window will also appear by left-clicking on any bar in the graph.

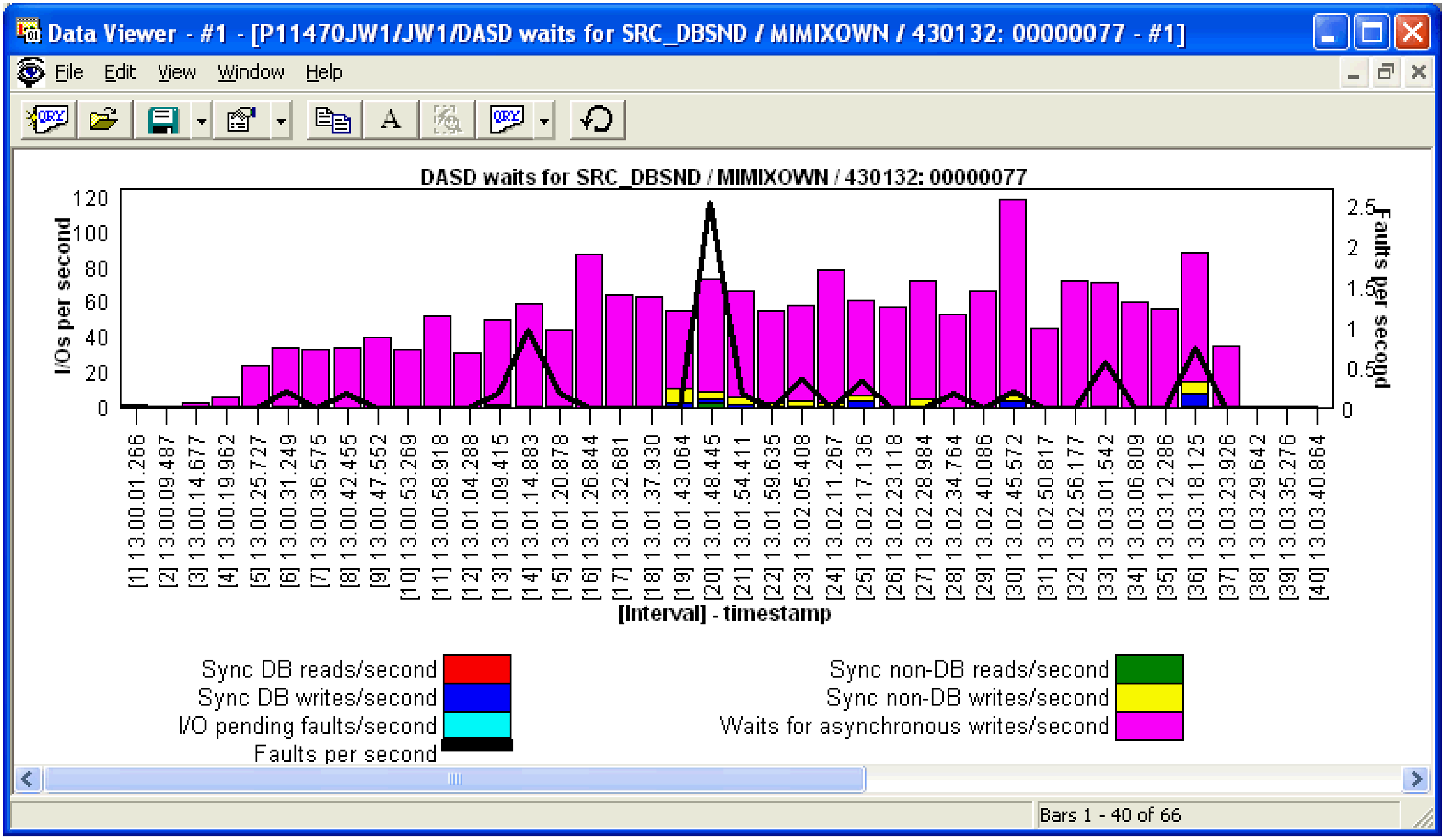
Table of Contents Previous Next

2.8.4.2 DASD waits

Description: This graph shows a **detailed** look at the disk IO operations and page faulting occuring by a job throughout its existence during the job watch. Each color in the graph represents a different type of IO operation.

Graph Type: detailed system-wide (stacked vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Contains rates of various types of disk operations for a specific job.
Second Y-Axis: This line shows the faults per second over time.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Display Stack	Shows the interval details window and brings focus to the call stack tab. The call stack may not always be available if it was indicated at collection time to not collect the call stack. In this case the call stack window will not show a call stack but other information about the job's performance during this interval will be available.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.

Properties	Displays details about the selected bar in the interval details window. This window will also appear by left-clicking on any bar in the graph.

Table of Contents Previous Next

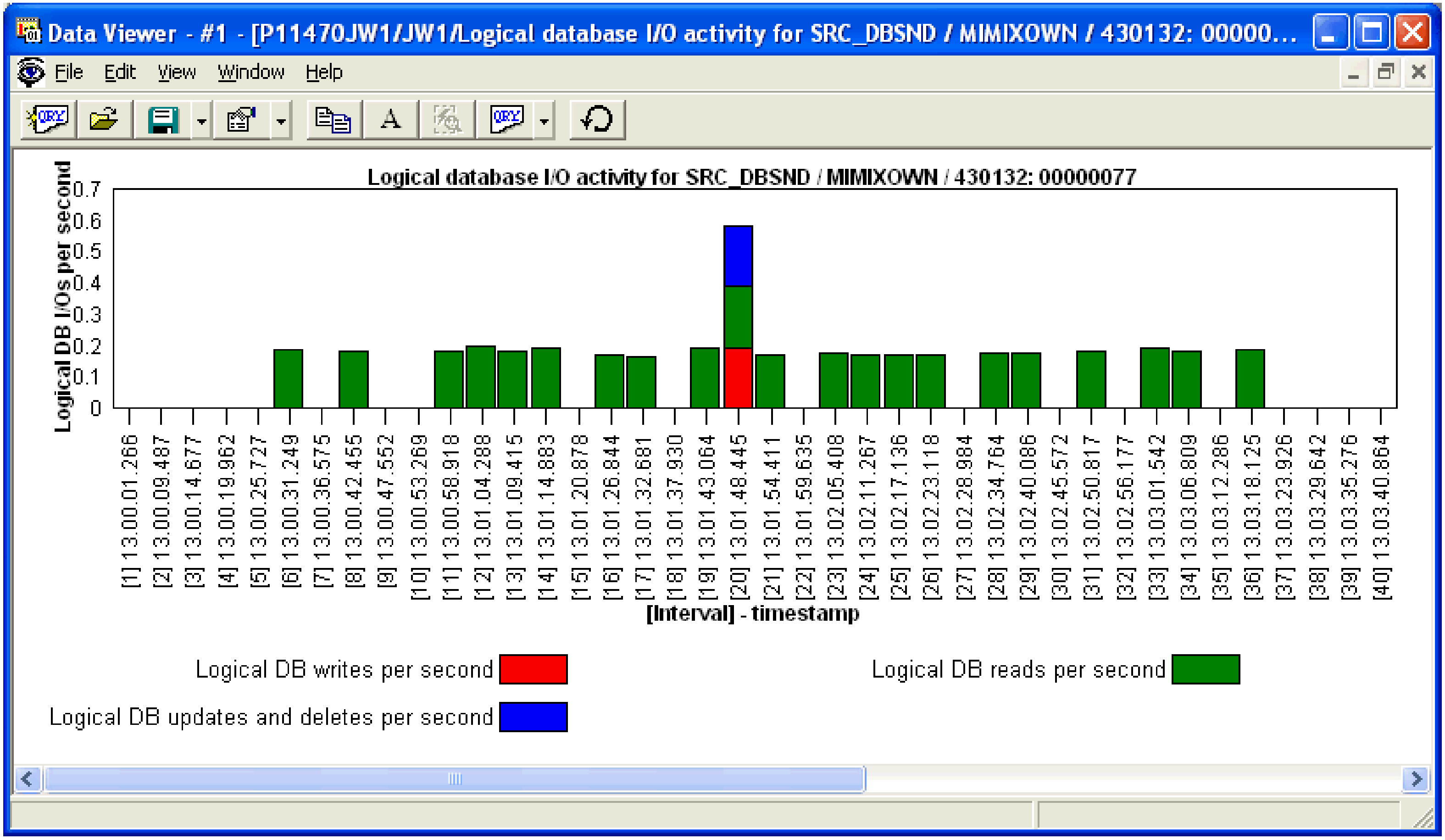
2.8.4.4 Logical database I/O activity

Description: This graph shows a **detailed** look at the rates of logical database I/O activity including writes, reads and updates/deletes for a job in the system-wide job watch.

This graph may be used to locate where LDIO activity is occurring within a specific job.

Graph Type: detailed system-wide (vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Contains the rate of logical database I/Os for writes, reads and updates/deletes.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Display Stack	Shows the interval details window and brings focus to the call stack tab. The call stack may not always be available if it was indicated at collection time to not collect the call stack. In this case the call stack window will not show a call stack but other information about the job's performance during this interval will be available.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.

Properties	Displays details about the selected bar in the interval details window. This window will also appear by left-clicking on any bar in the graph.

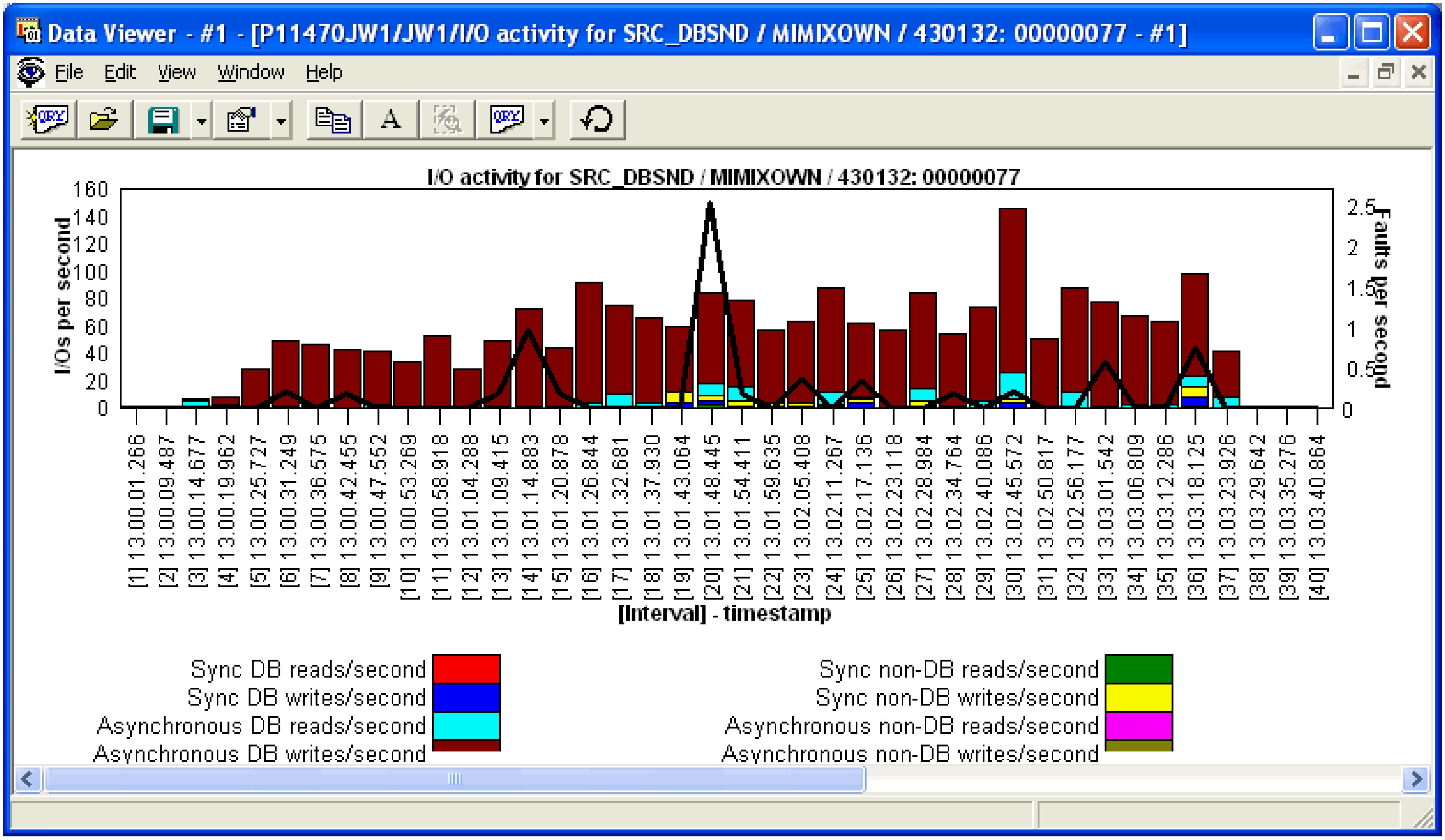
Table of Contents Previous Next

2.8.4.3 I/O Activity

Description: This graph shows a **detailed** look at the physical I/Os that were made by a job throughout its existence during the job watch. The graph provides rates of database and non-database reads and writes. Page fault rates are listed on the secondary Y-axis.

Graph Type: detailed system-wide (stacked vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Contains the rate of async and synchronous DB and non-DB reads and writes.
Second Y-Axis: The line represents the page fault rate for the job over time.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Display Stack	Shows the interval details window and brings focus to the call stack tab. The call stack may not always be available if it was indicated at collection time to not collect the call stack. In this case the call stack window will not show a call stack but other information about the job's performance during this interval will be available.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.

Properties	Displays details about the selected bar in the interval details window. This window will also appear by left-clicking on any bar in the graph.

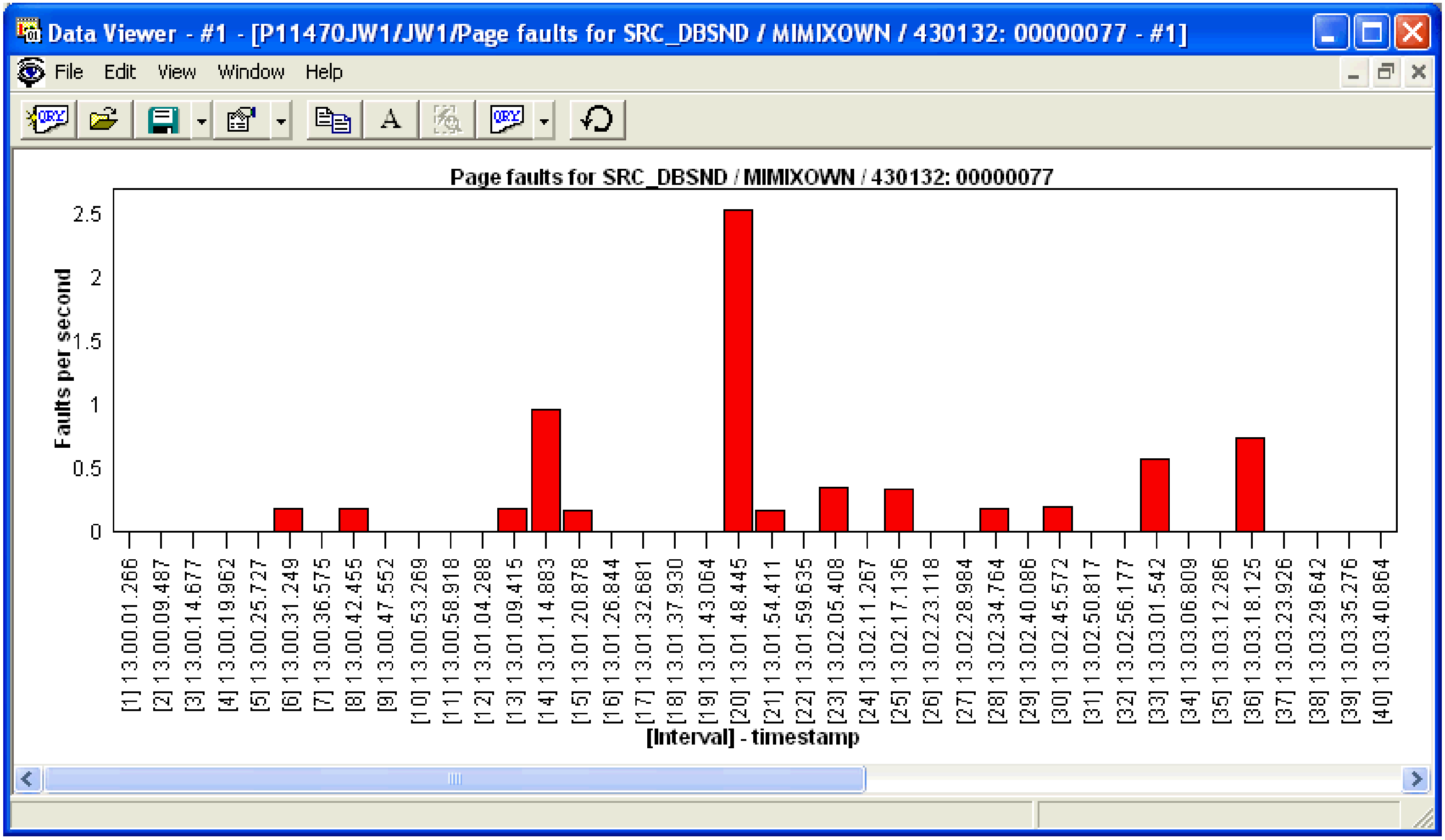
Table of Contents Previous Next

2.8.4.5 Page faults

Description: This graph shows a **detailed** look at the page fault rate per second for a job in a system-wide job watch.

Graph Type: detailed system-wide (vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Each bar displays the page fault rate for the job each interval.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Display Stack	Shows the interval details window and brings focus to the call stack tab. The call stack may not always be available if it was indicated at collection time to not collect the call stack. In this case the call stack window will not show a call stack but other information about the job's performance during this interval will be available.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in the interval details window. This window will also appear by left-clicking on any bar in the graph.

Table of Contents | Previous | Next

2.8.5 IFS graphs - detailed

The detailed IFS graphs provide IFS statistics for a job in a job-specific Job Watch.

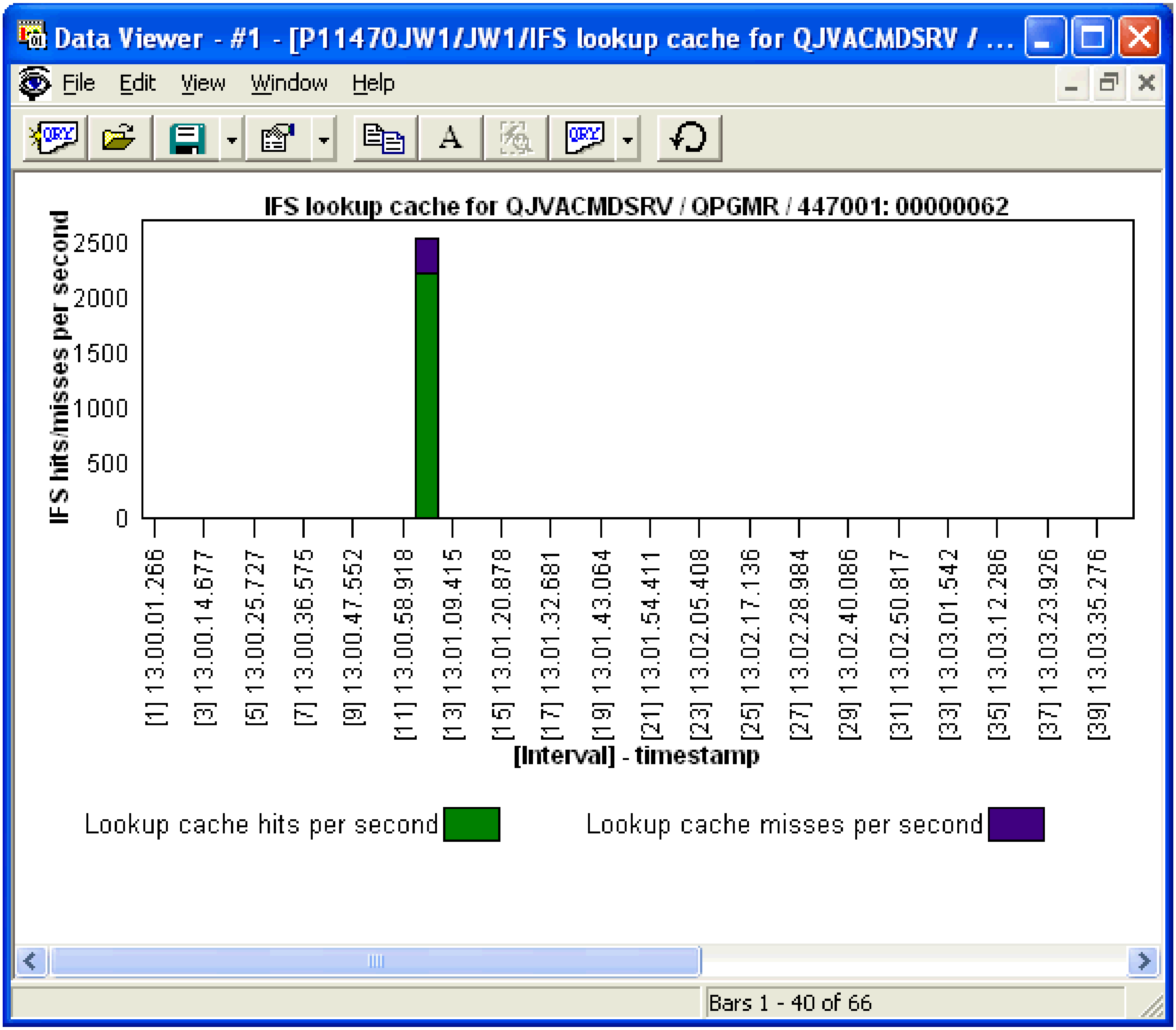
Table of Contents Previous Next

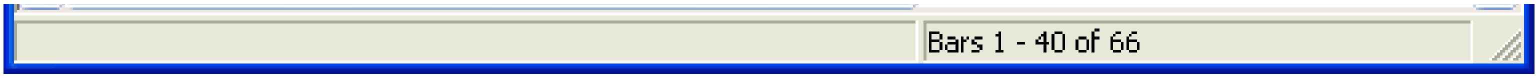
2.8.5.1 IFS lookup cache

Description: This graph shows a **detailed** look at the rate of hits/misses for the IFS lookup cache by a job throughout its existence during the job watch. Lookup cache hit rate is indicated in green and misses are shown in blue.

Graph Type: detailed system-wide (stacked vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Contains the rate of lookup cache hits and misses.

Example:





Bars 1 - 40 of 66

Right-clicking on a bar in this graph provides the following menu options:

Menu	**Description**
Display Stack	Shows the interval details window and brings focus to the call stack tab. The call stack may not always be available if it was indicated at collection time to not collect the call stack. In this case the call stack window will not show a call stack but other information about the job's performance during this interval will be available.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in the interval details window. This window will also appear by left-clicking on any bar in the graph.

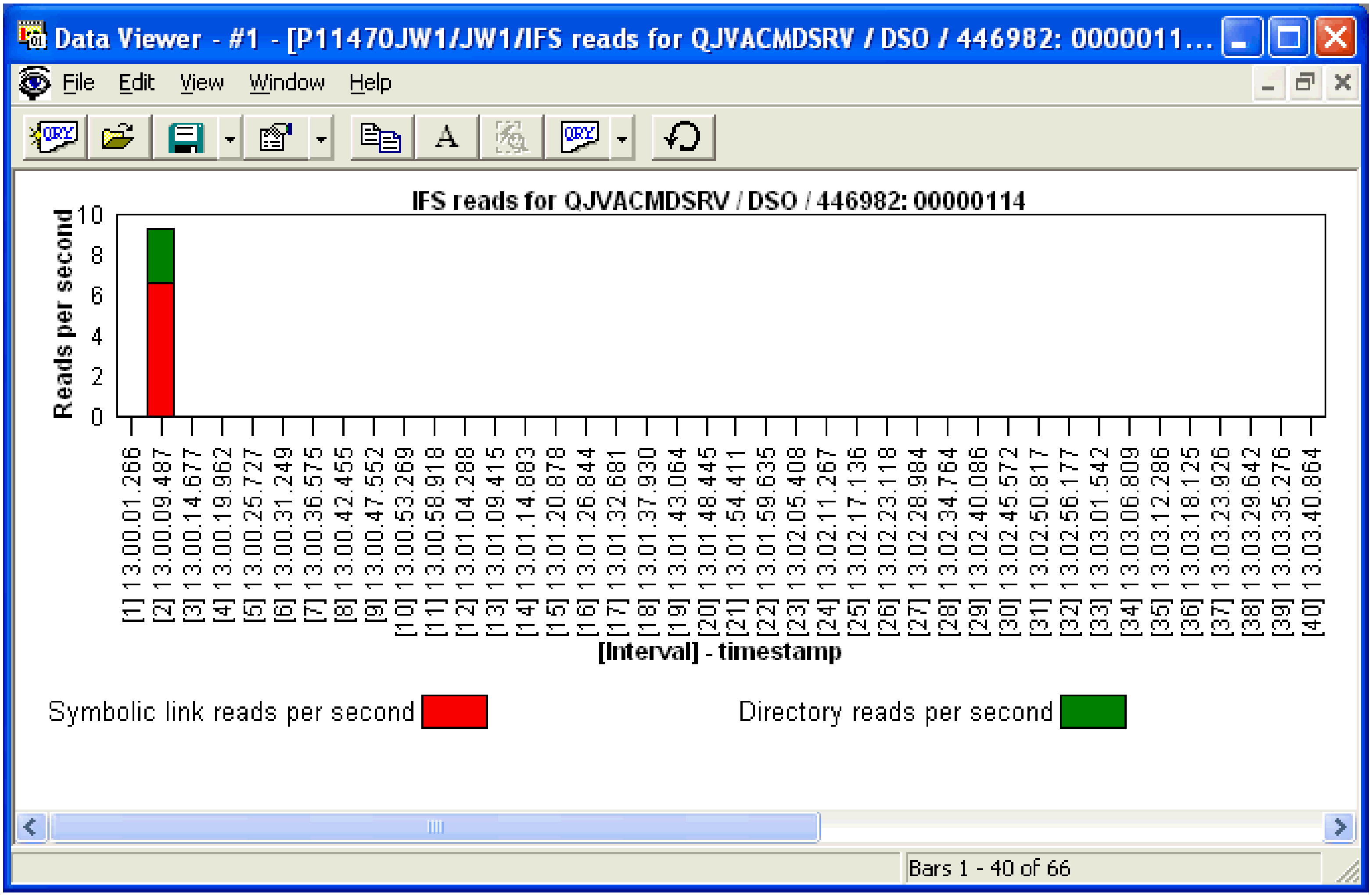
Table of Contents | Previous | Next

2.8.5.2 IFS reads

Description: This graph shows a **detailed** look at the rate of IFS directory reads and symbolic link reads for a job throughout its existence during the job watch.

Graph Type: detailed system-wide (stacked vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Contains the rate of symbolic link reads per second in red and directory reads per second in green.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Display Stack	Shows the interval details window and brings focus to the call stack tab. The call stack may not always be available if it was indicated at collection time to not collect the call stack. In this case the call stack window will not show a call stack but other information about the job's performance during this interval will be available.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.

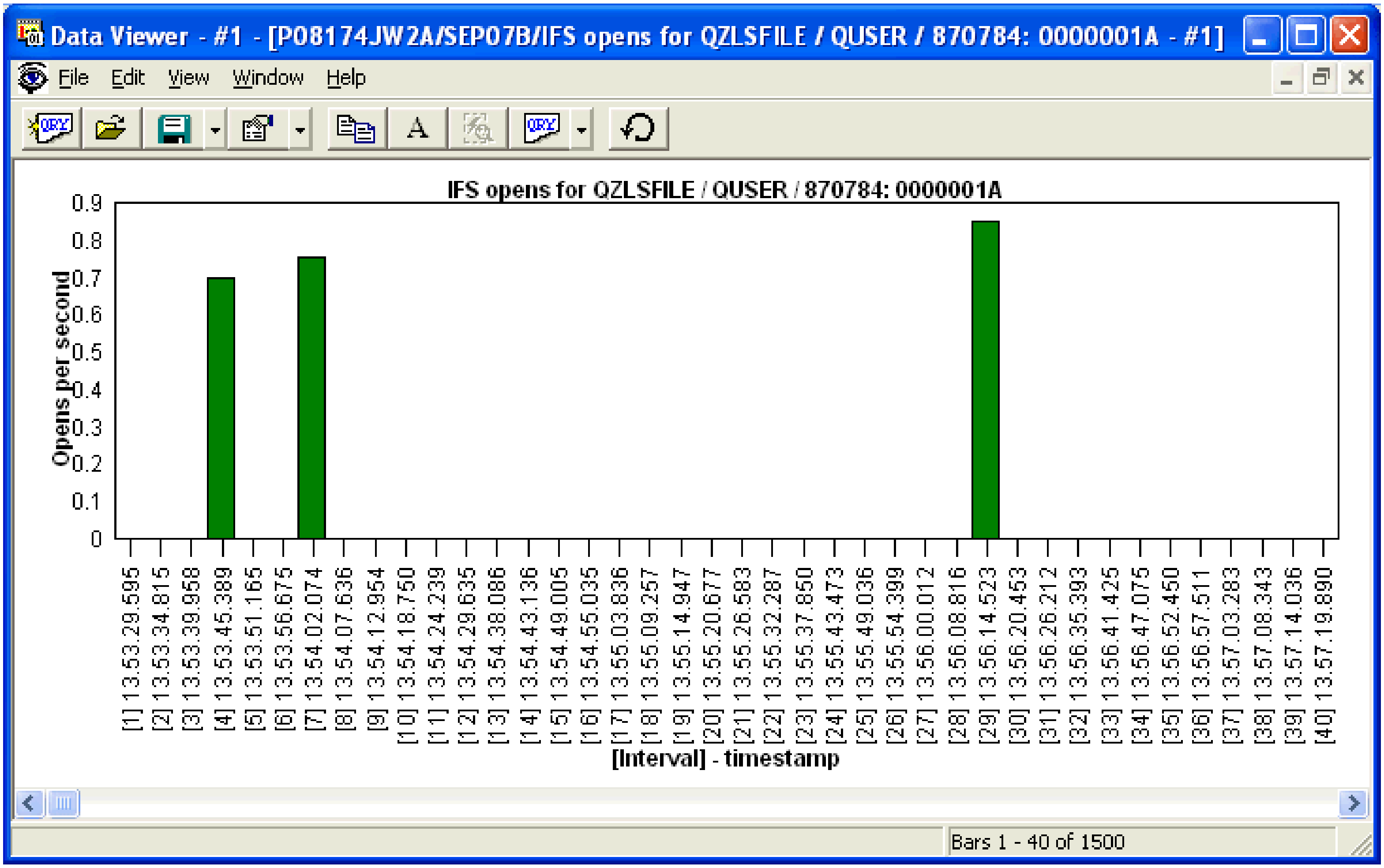
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in the interval details window. This window will also appear by left-clicking on any bar in the graph.

2.8.5.3 IFS opens

Description: This graph shows a **detailed** look at the rate of IFS file opens by a job throughout its existence during the job watch.

Graph Type: detailed system-wide (vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Contains the number of IFS file opens occuring for each interval per second.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	Description
Display Stack	Shows the interval details window and brings focus to the call stack tab. The call stack may not always be available if it was indicated at collection time to not collect the call stack. In this case the call stack window will not show a call stack but other information about the job's performance during this interval will be available.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.

Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in the interval details window. This window will also appear by left-clicking on any bar in the graph.

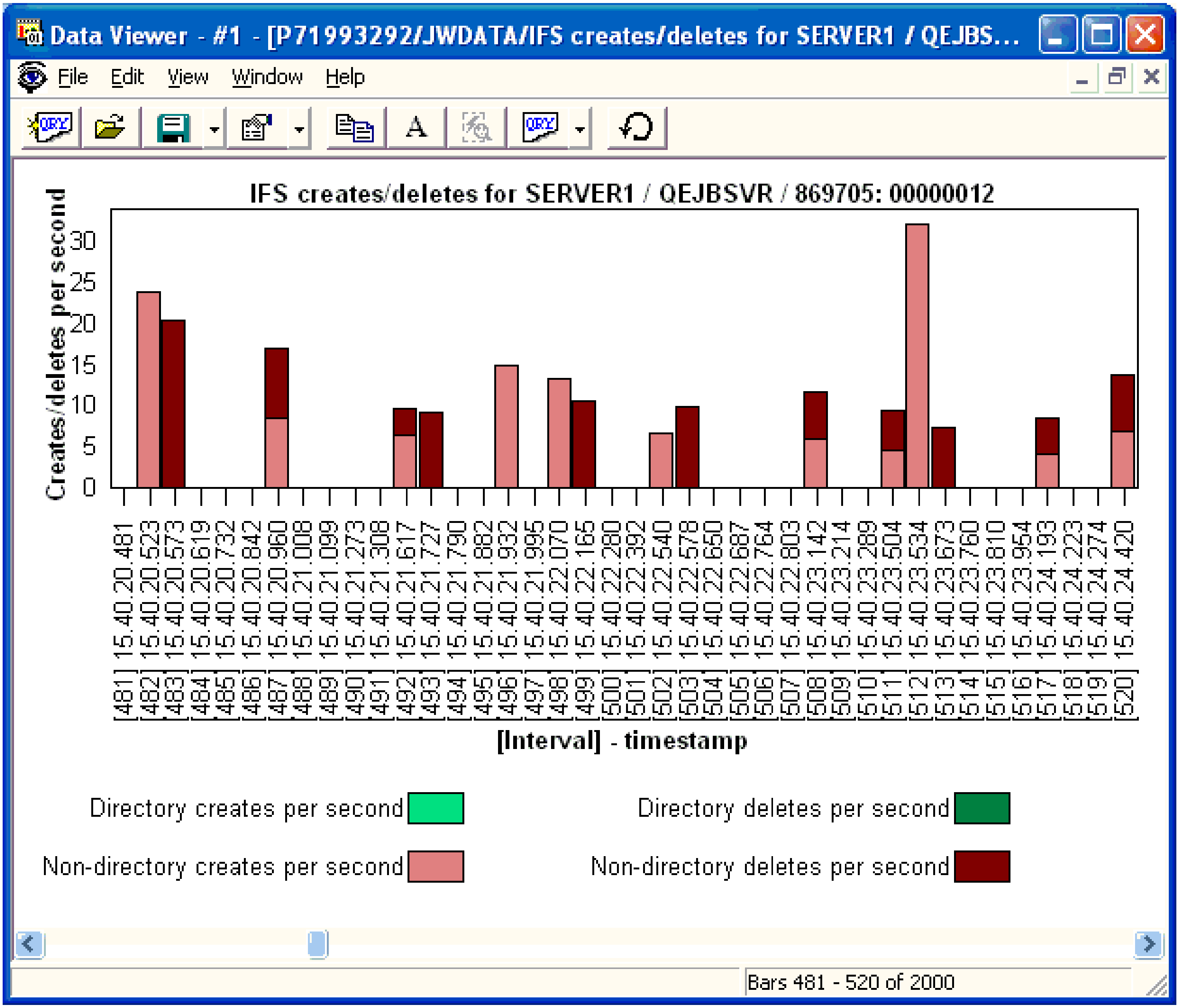
Table of Contents | Previous | Next

2.8.5.4 IFS creates/deletes

Description: This graph shows a **detailed** look at the rate of hits/misses for the IFS creates and deletes performed by a job throughout its existence during the job watch. The creates and deletes are categorized into two types: directory and non-directory.

Graph Type: detailed system-wide (stacked vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Contains the rates of IFS directory and non-directory creates and deletes per second. Green colors represent directory operations and red colors represent non-directory operations.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	**Description**
Display Stack	Shows the interval details window and brings focus to the call stack tab. The call stack may not always be available if it was indicated at collection time to not collect the call stack. In this case the call stack window will not show a call stack but other information about the job's performance during this interval will be available.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in the interval details window. This window will also appear by left-clicking on any bar in the graph.

Table of Contents | Previous | Next

2.8.7 Other graphs - detailed

The detailed "other" graphs provide graphs that were not covered in the other categories for a job in a job-specific Job Watch.

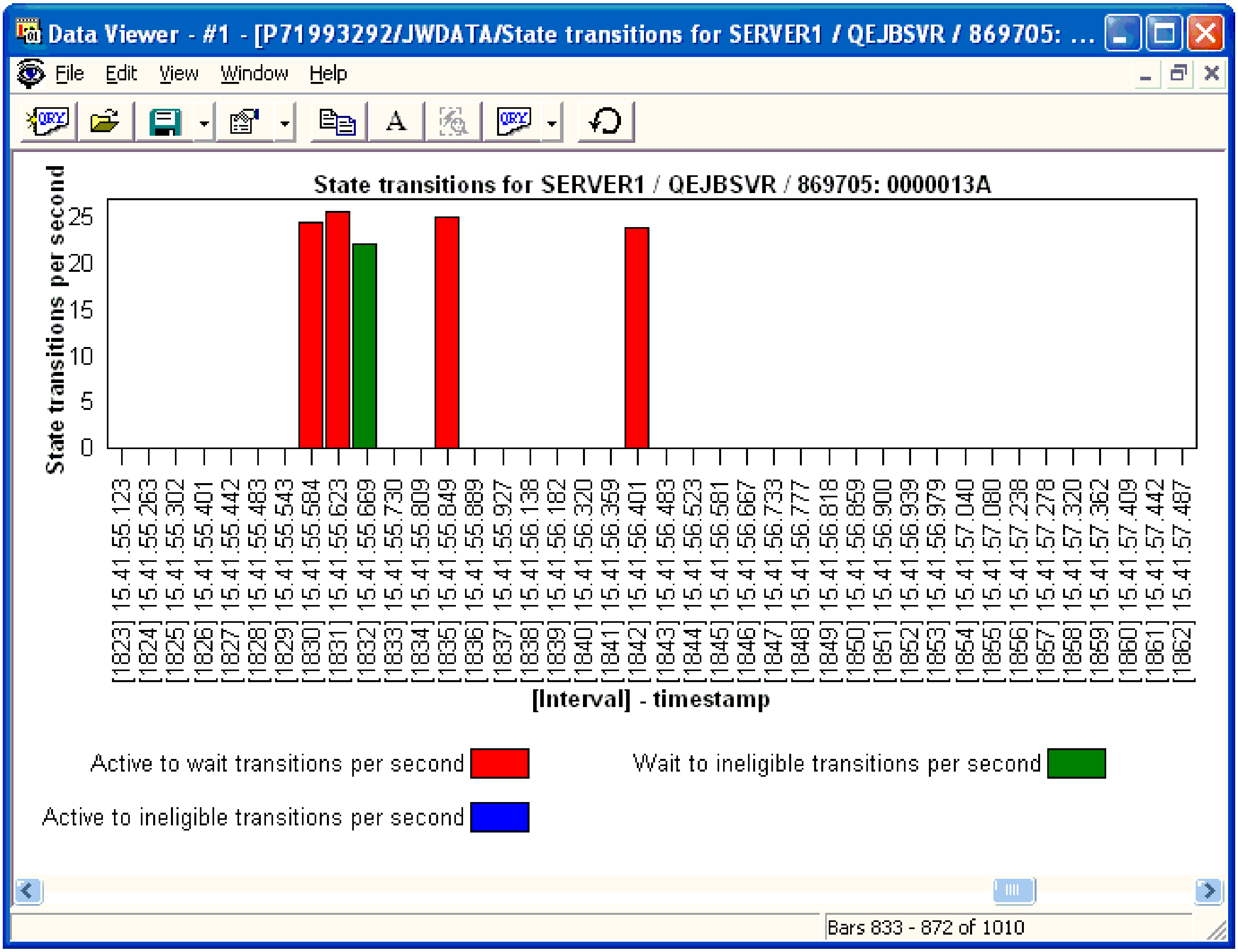
Table of Contents | Previous | Next

2.8.7.1 State transitions

Description: This graph shows a **detailed** look at the job state transitions occurring for a job in the job watch.

Graph Type: detailed system-wide (stacked vertical bar)
X-axis: Interval number and time of day. The time of day is listed as hh:mm:ss:xxx
Y-Axis: Each bar shows the rates of the following types of job state transitions: active to wait, wait to ineligible, and active to ineligible.

Example:



Right-clicking on a bar in this graph provides the following menu options:

Menu	Description

Display Stack	Shows the interval details window and brings focus to the call stack tab. The call stack may not always be available if it was indicated at collection time to not collect the call stack. In this case the call stack window will not show a call stack but other information about the job's performance during this interval will be available.
Copy	Makes a copy of the graph view to the clipboard as a bitmap image. This image can be pasted into applications that accept images from the clipboard.
Save As...	Provides the option to create a .JPG image from the current graph view.
Preferences...	Displays the Preferences interface.
Graph definition...	Displays an interface allowing customization of the graph definition used to build the graph.
Query definition...	Displays an interface allowing customization of the query definition used to build the graph.
Properties	Displays details about the selected bar in the interval details window. This window will also appear by left-clicking on any bar in the graph.

Table of Contents | Previous | Next

Chapter 4 PEX Analyzer

This chapter provides an overview of the interfaces within the iDoctor for iSeries PEX Analyzer component.

Table of Contents | Previous | Next

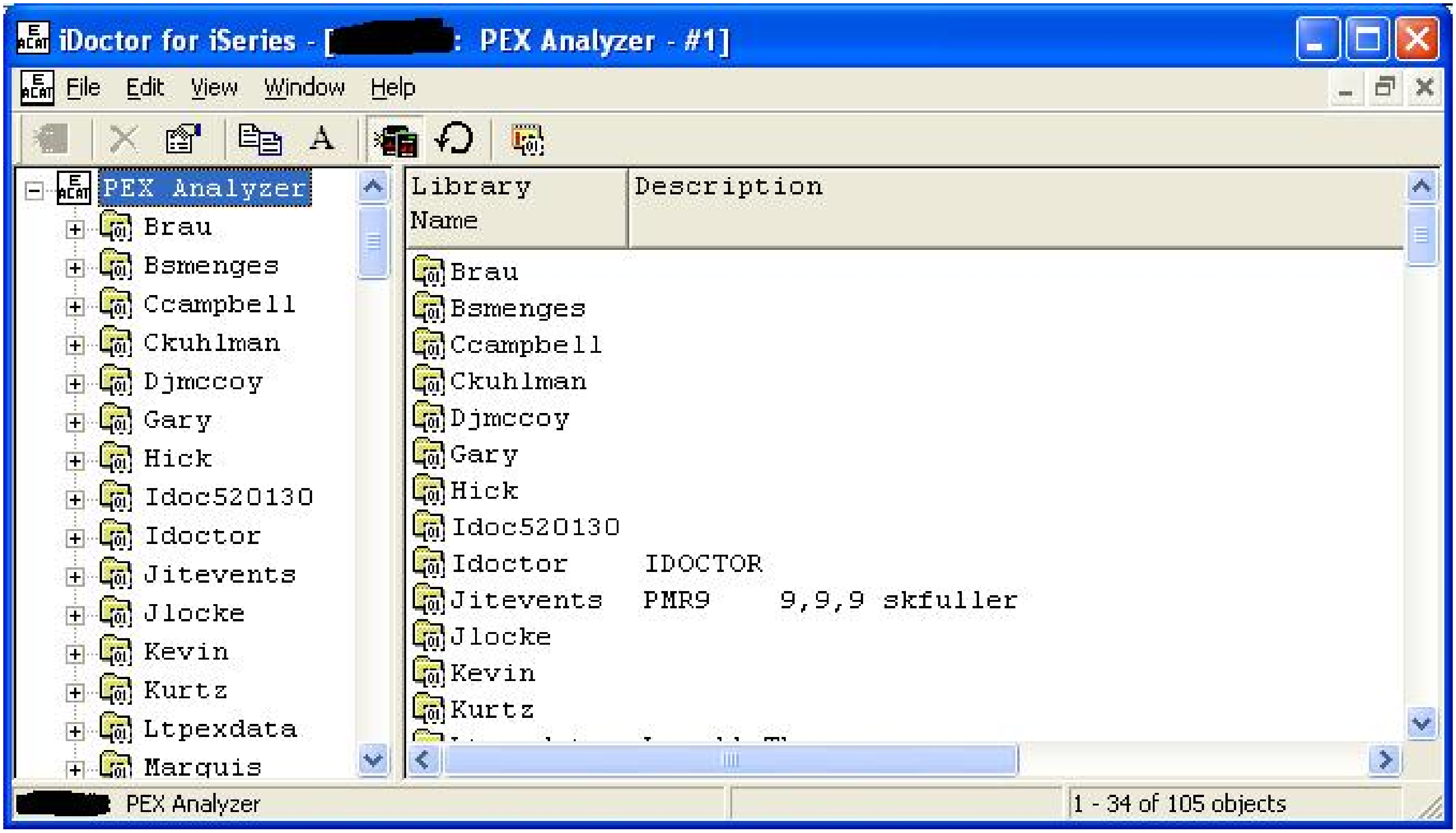
4.1 PEX Analyzer Basics

The PEX Analyzer tool provides a number of interfaces that allows a user to work with PEX collections, and also analyze them.

Starting PEX Analyzer

PEX Analyzer is a component of the iDoctor for iSeries suite of tools. iDoctor for iSeries can be started using the Start menu: Start->Programs->iDoctor for iSeries. Once the iDoctor for iSeries application appears, the PEX Analyzer component is started from the Connection List View by right-clicking on a system name and choosing the PEX Analyzer menu.

After starting PEX Analyzer you will see a window similar to the following:



iDoctor for iSeries - [[redacted]: PEX Analyzer - #1]

Library Name	Description
Brau	
Bsmenges	
Ccampbell	
Ckuhlman	
Djmccoy	
Gary	
Hick	
Idoc520130	
Idoctor	IDOCTOR
Jitevents	PMR9 9,9,9 skfuller
Jlocke	
Kevin	
Kurtz	

[redacted] PEX Analyzer | 1 - 34 of 105 objects

The 'PEX Analyzer' folder contains a list of library folders, each representing a library on the iSeries system that contains PEX database files (collections). The list displays each library's name and description. To be more specific a library is displayed in this list if the QAYPERUNI file was found in the library or if the library contains one or more collections currently in progress that have not produced their output files yet.

PEX Analyzer Objects

There are four main types of objects within the tree/list views of PEX Analyzer in the following order: **Libraries**, **collections**, **analyses**, and **reports**. Each of these will covered in more detail in the next sections.

PEX Analyzer Menu Options

The following menu options are available by right-clicking on the 'PEX Analyzer' folder in the tree/list view above.

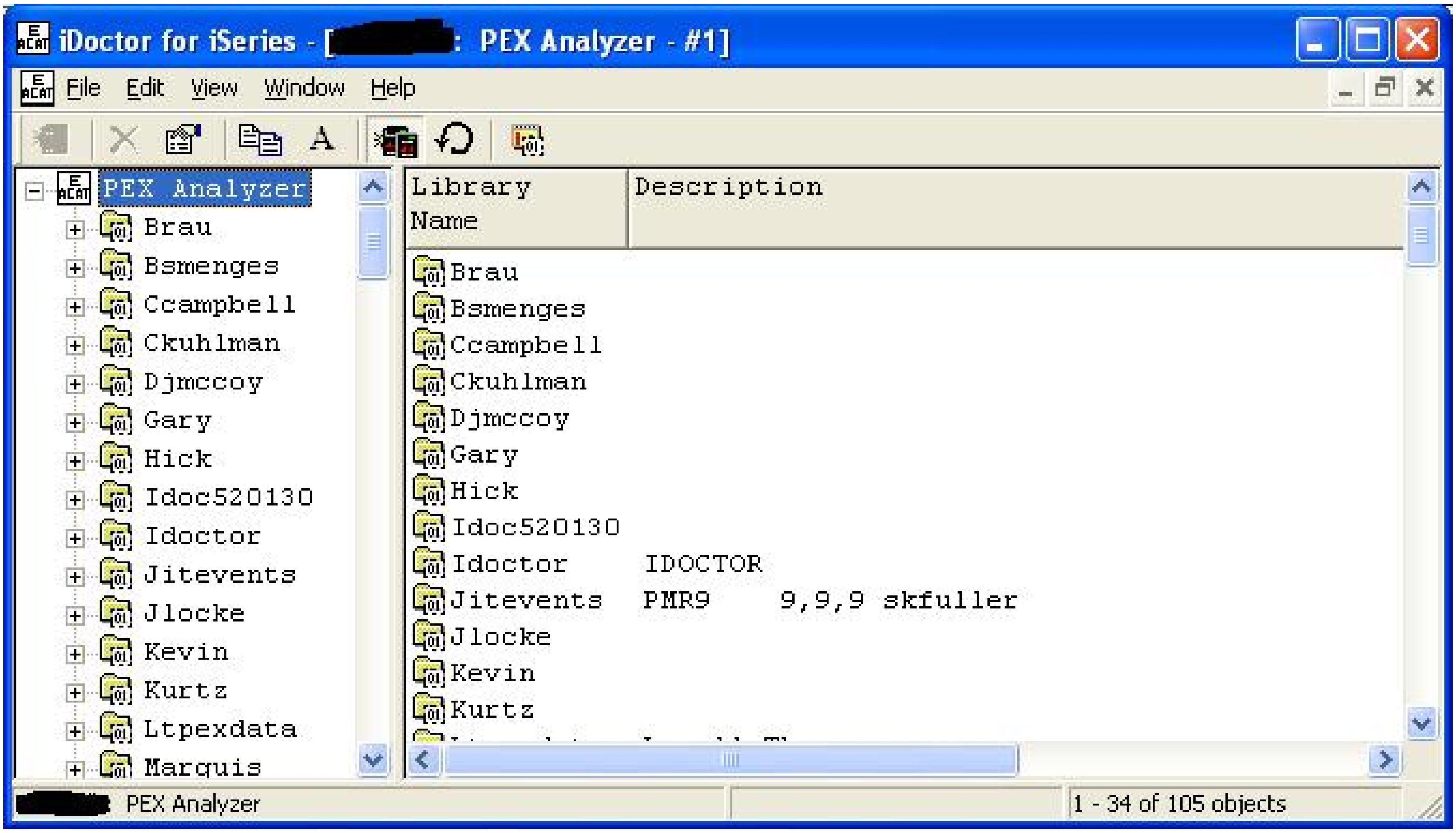
Menu Item	Description
Explore	Displays the contents of the PEX Analyzer folder (a list of libraries on the system containing PEX data) in the right pane of the tree/list window.
Create PEX Definition...	Displays the PEX Definition Wizard which lets you create a PEX definition on your system. This is an interface over the ADDPEXDFN command.
Create PEX Filter...	Displays the PEX Filter Wizard which allows the user to define a PEX filter on the current system. This is an interface over the ADDPEXFTR command.
Create PEX Collection...	Displays the PEX Collection Wizard which lets you create a PEX collection on the system. PEX collections contain detailed performance information and can be analyzed via PEX Analyzer.
Work with PEX Definitions	Displays the PEX Definitions View which is a list of all the PEX definitions on your system. You may change a PEX definition, delete, or create PEX collections using an existing PEX definitions using this view.
Work with PEX Filters	Displays the PEX Filters View which is a list of all the PEX filters on your system. You may create, change or delete PEX filters from this view.
Manage incompatible PEX data	Displays a wizard that lets you detect and optionally delete any libraries that contain PEX collections or analyses that are incompatible with your current version.
Migrate query and graph definitions	This will convert any old user-defined query or graph definitions that exist on the system to the current format.
Open New Data Viewer	Opens a new Data Viewer window. This window is used to display tables and graphs on the system. You can open PEX Analyzer reports into this window or you can also open any other type of physical file and view as a graph or table.
Properties	Use this menu to display the PEX Analyzer property pages. The PEX property pages contain version information and display the job queue and subsystem settings to use when running PEX analysis jobs.

Table of Contents | Previous | Next

4.2 Libraries

The 'PEX Analyzer' folder contains a list of library folders, each representing a library on the iSeries system that contains PEX database files. The list displays each library's name and description.

By clicking on a library in the tree you will see its contents (a list of collections). Double-clicking a library in the list portion of this view will also display the collections within it.



PEX Analyzer
- Brau
- Bsmenges
- Ccampbell
- Ckuhlman
- Djmccoy
- Gary
- Hick
- Idoc520130
- Idoctor
- Jitevents
- Jlocke
- Kevin
- Kurtz
- Ltpexdata
- Marquis

Library Name	Description
Brau	
Bsmenges	
Ccampbell	
Ckuhlman	
Djmccoy	
Gary	
Hick	
Idoc520130	
Idoctor	IDOCTOR
Jitevents	PMR9 9,9,9 skfuller
Jlocke	
Kevin	
Kurtz	

[redacted] PEX Analyzer | 1 - 34 of 105 objects

Table of Contents | Previous | Next

4.2.1 Menu Options

A library folder in PEX Analyzer has a basic set of menu options that are available within all components as well as some options specific to PEX Analyzer.

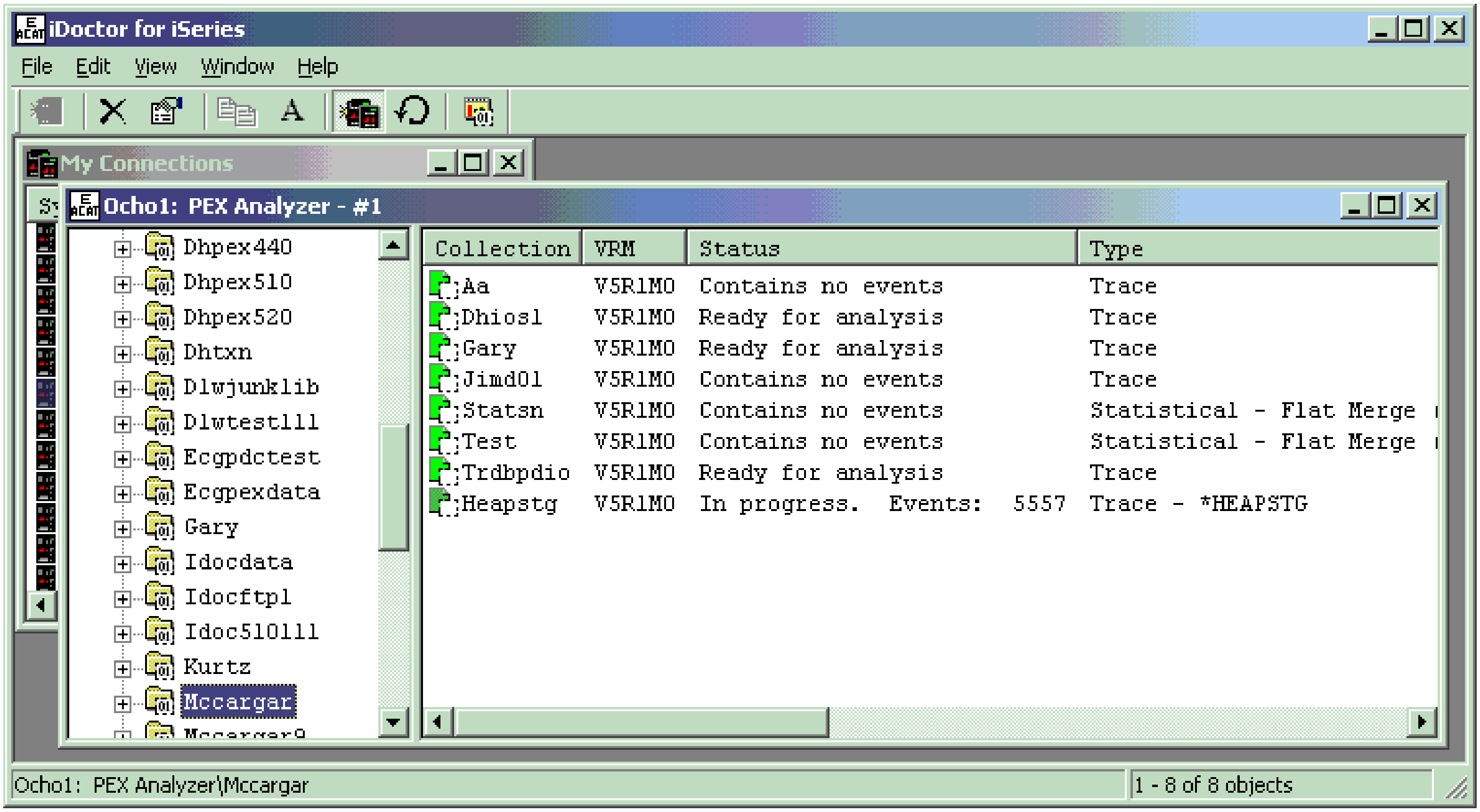
The following are the options for a library folder that are only available in PEX Analyzer.

Menu	**Description**
Create PEX Collection...	Displays the PEX Collection Wizard which lets you create a PEX collection on the system. PEX collections contain detailed performance information and can be analyzed via PEX Analyzer.
Delete all PEX Analyses...	Use this option to delete all PEX analyses in the current library.
Delete all PEX Collections and Analyses...	Use this option to delete all PEX collections and analyses in the current library.

Table of Contents | Previous | Next

4.3 PEX Collections

Moving down the tree within each library folder are one or more PEX collections. PEX collections that are in the process of being created will be shown in this list as well as the collections that have already been created and are ready for analysis.



Collection	VRM	Status	Type
Aa	V5R1M0	Contains no events	Trace
Dhiosl	V5R1M0	Ready for analysis	Trace
Gary	V5R1M0	Ready for analysis	Trace
Jimd01	V5R1M0	Contains no events	Trace
Statsn	V5R1M0	Contains no events	Statistical - Flat Merge
Test	V5R1M0	Contains no events	Statistical - Flat Merge
Trdbpdio	V5R1M0	Ready for analysis	Trace
Heapstg	V5R1M0	In progress. Events: 5557	Trace - *HEAPSTG

Collection Status

Each collection has a status field indicating if and how a collection may be used. The status is reflected by the type of icon shown for each collection and is summarized in the table below.

Collection Icon	What it means
Green	Collection may be analyzed (using the Analysis Wizard) or deleted. The properties of the collection are viewable.
Dark Green	Collection is either in the process of being created or failed to be created. The status field indicates the status of the job used for creating the collection. The status field for in progress collections will show the number of events that have been collected so far.
Green and Red	Collection is read-only. The collection properties and any already created analyses may be viewed. The collection cannot be deleted.
Red	The collection is not useable. This usually indicates that the PEX data VRM is not supported by the tool. It could also indicate that the PEX collection did not complete successfully or is damaged in some way (missing files).

Table of Contents | Previous | Next

4.3.1 Menu Options

PEX Collections have a different set of menu options depending on if the collection is in progress or if it has finished collecting and is ready for analysis.

Collections that are in progress have the following set of menu options:

Menu	**Description**
End Collection -> Create DB files	Stop the collection prematurely and immediately begin creating DB files. Use this after your test has completed and you don't need to collect data any longer.
End Collection -> Create single object (for transport to another system)	Use this option to generate a single file for the collection (a *MGTCOL object). These objects can be sent to another system using the Transfer to... option for a completed collection.
End Collection -> Delete	Stops collecting immediately and destroys the data that has been collected so far.
End Collection -> Stop	Ends collection and begins dumping the data.
Restart	Destroys the data that has been collected so far and then restarts the collection using the same settings.
Resume	Resumes a suspended collection. Use this option to start a collection that was created in 'Standby mode'.
Properties	Displays the basic collection properties like the name and type. From collection properties you can view the job log of the IDOCCOL job running the PEX collection in progress.

Collections that are complete and ready for analysis have these menu options:

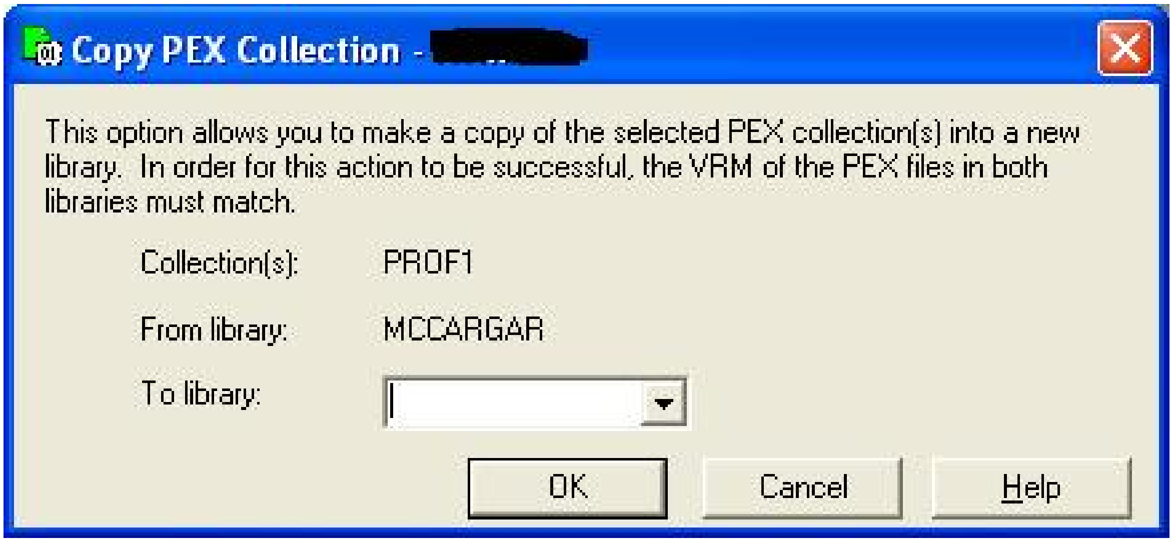
Menu	**Description**
Explore	Displays the analyses within the PEX collection in the list portion of the PEX Analyzer component view.
Record Quick View	Displays the fields pertaining to the PEX collection vertically. This can help reduce the amount of scrolling required to see all the fields.
Select fields...	Displays an interface that lets you modify which fields are shown for collections. You can optionally include fields such as number of events or number of jobs to get additional information about a list of collections without needing to go into 'Properties'.

Copy...	Copy one or more collections to a new library.
Delete...	Delete one or more collections in the current library.
Transfer to...	FTP the selected collection(s) to a remote system.
Properties	Displays the detailed collection properties.

Table of Contents | Previous | Next

4.3.2 Copying

PEX collections can be easily copied to a new library by using the Copy... menu found by right-clicking on a collection. You can select more than one collection at a time to copy multiple collections to a new library in one step.



Copy PEX Collection - [redacted]

This option allows you to make a copy of the selected PEX collection(s) into a new library. In order for this action to be successful, the VRM of the PEX files in both libraries must match.

Collection(s): PROF1

From library: MCCARGAR

To library:

Field	Description
Collections	List of collections in the 'from library' to be copied to the 'to library'
From library	Name of the library the PEX collections will be copied from.
To library	The name of the library that will receive the PEX collections. By clicking the down arrow you can choose from a list of all libraries on the system. You may also specify a new library that does not exist and it will be created.

Table of Contents | Previous | Next

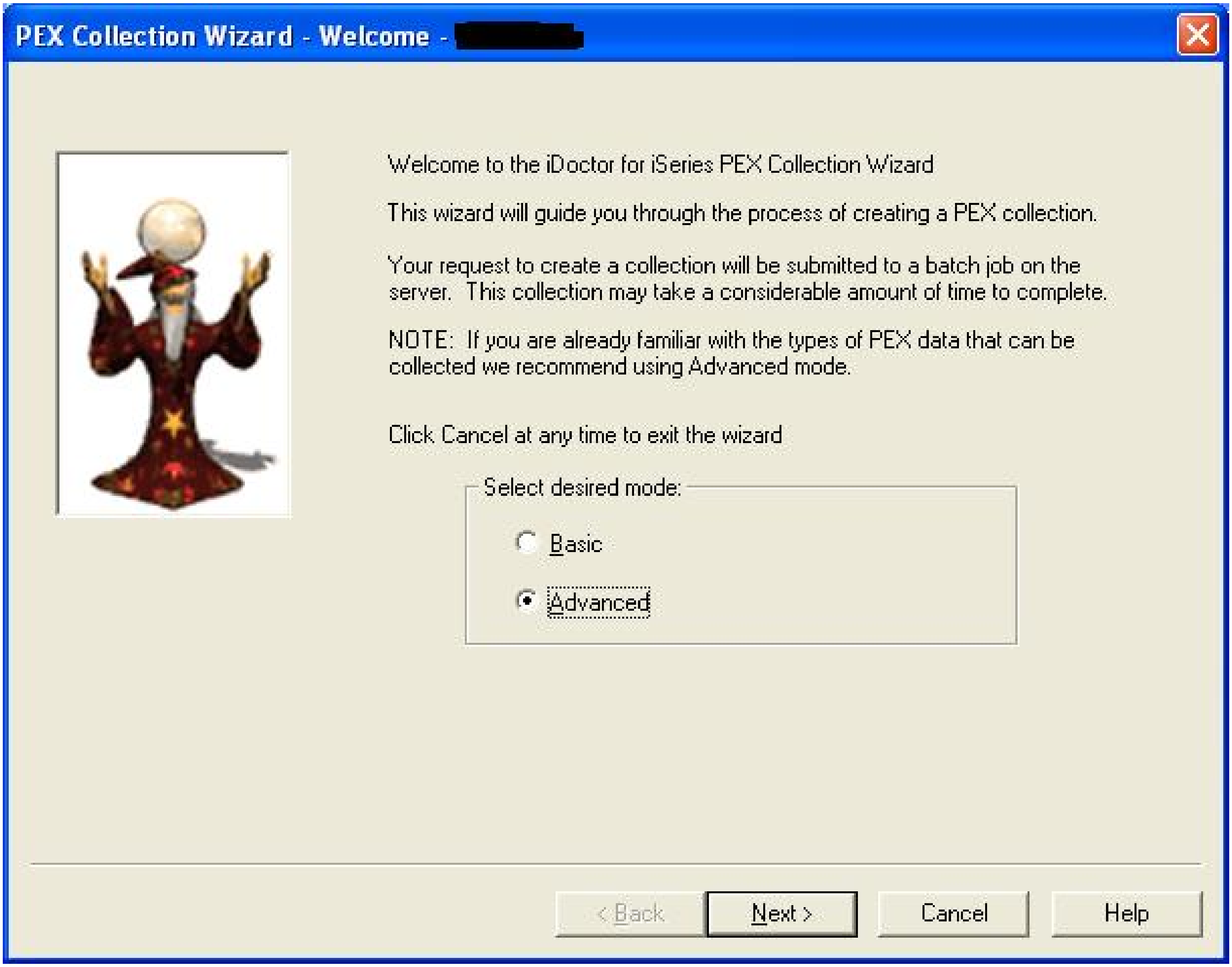
4.3.3 Creating - The PEX Collection Wizard

Within iDoctor for iSeries there are two ways to create a new PEX collection. You can either use the PEX Collection Wizard in the client or you can use the STRIDOCCOL green screen command. This section covers the PEX Collection Wizard in the GUI.

PEX Collections are created using a PEX definition. Definitions can be created using the green screen ADDPEXDFN command or via the PEX Definition Wizard, also available in the GUI. iDoctor ships several commonly used PEX definitions called 'iDoctor-supplied' PEX definitions. There are several different iDoctor-supplied definitions which cover the most basic problem types. PEX definitions are used to define the specific types of events to capture in your iSeries application or across the entire system and store them in database files when creating a PEX collection.

You can access the PEX Collection Wizard using the Create PEX Collection popup menu when right-clicking on either the PEX Analyzer icon or a library icon.

PEX Collections are created from a batch job on the server (job name IDOCCOL). Depending on the type of data collected, the number of events collected, and the size of ths system, the collection could take anywhere from 30 seconds to hours or even days to complete. For this reason it is important to keep the total time of collection (the Duration parameter) as small as needed. Collecting only the events that are needed only for the jobs needed is also of critical importance to improve problem resolution time.



PEX Collection Wizard - Welcome - [redacted]

Welcome to the iDoctor for iSeries PEX Collection Wizard

This wizard will guide you through the process of creating a PEX collection.

Your request to create a collection will be submitted to a batch job on the server. This collection may take a considerable amount of time to complete.

NOTE: If you are already familiar with the types of PEX data that can be collected we recommend using Advanced mode.

Click Cancel at any time to exit the wizard

Select desired mode:

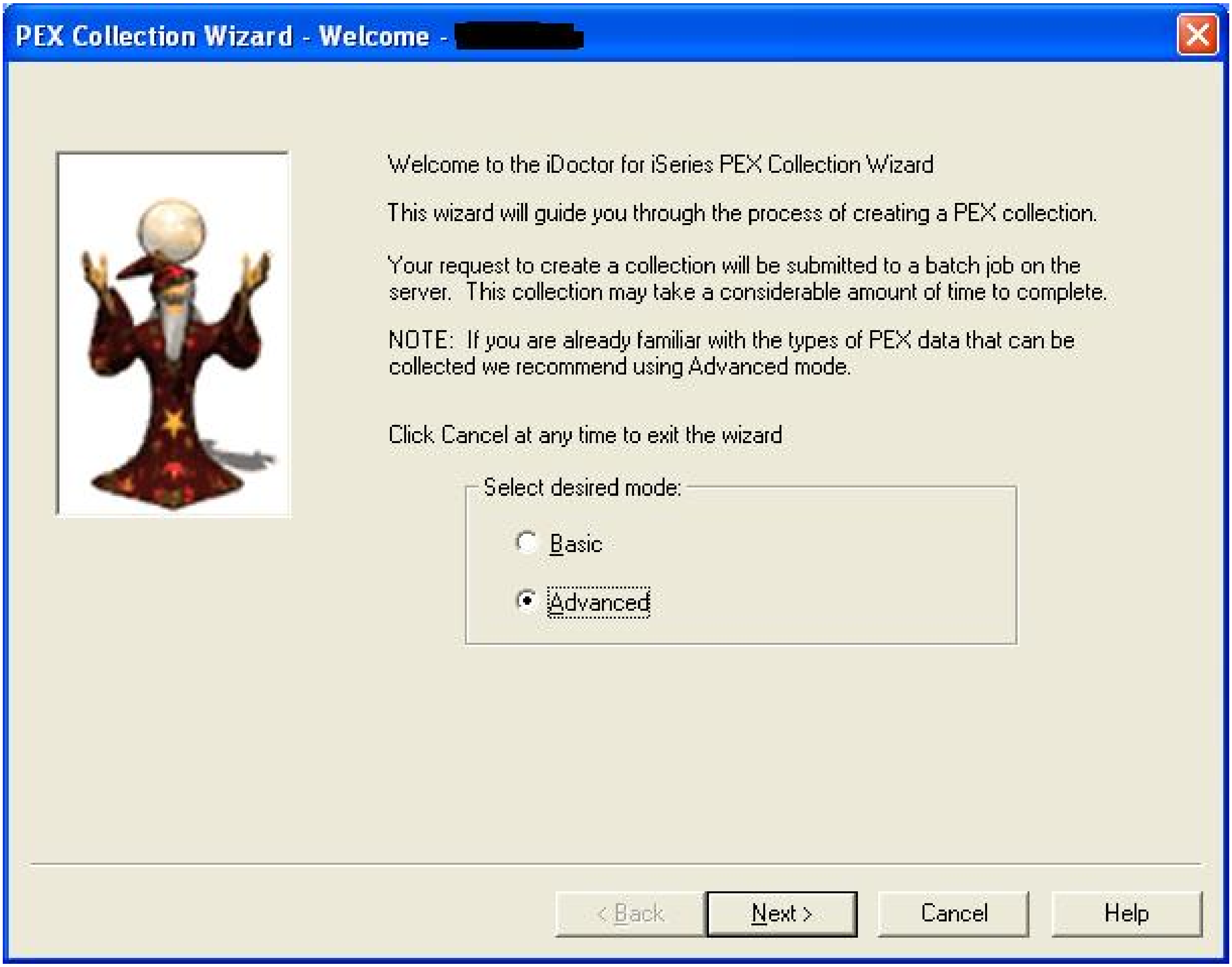
○ Basic

● Advanced

Table of Contents Previous Next

4.3.3.1 Welcome

The Welcome page in the PEX Collection Wizard introduces the user to the wizard and offers information about what the wizard will do. From here a user can decide which mode to run the wizard in: basic or advanced. Basic mode will follow with a series of questions designed to help a user determine what type of iDoctor-supplied definition best fits the type of performance problem they are having. Advanced mode skips the questions and goes right to the Options page.



PEX Collection Wizard - Welcome - [redacted]

Welcome to the iDoctor for iSeries PEX Collection Wizard

This wizard will guide you through the process of creating a PEX collection.

Your request to create a collection will be submitted to a batch job on the server. This collection may take a considerable amount of time to complete.

NOTE: If you are already familiar with the types of PEX data that can be collected we recommend using Advanced mode.

Click Cancel at any time to exit the wizard

Select desired mode:

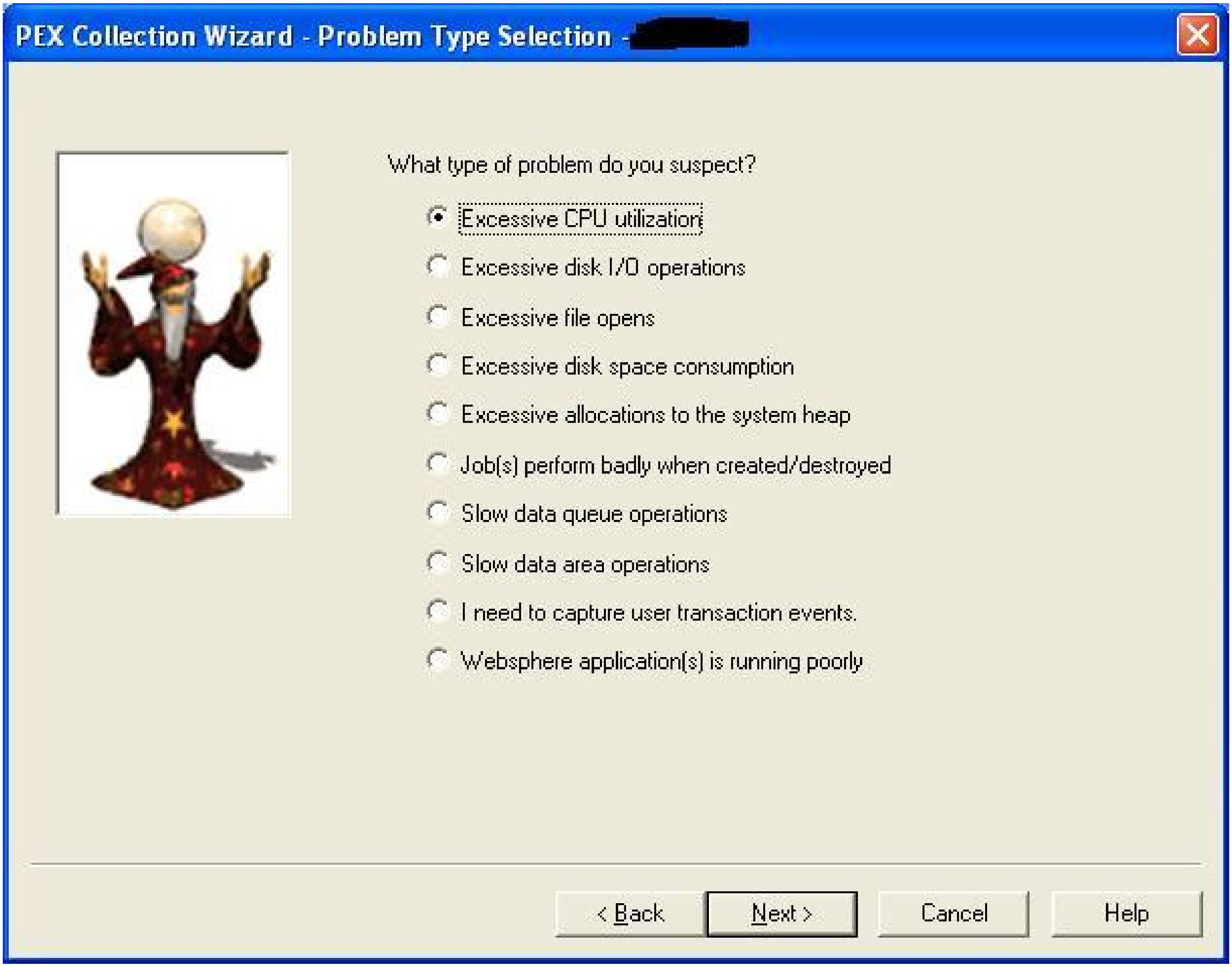
○ Basic

● Advanced

4.3.3.2 Problem Type Question Pages

The Problem Type Question Pages present a series of questions designed to help a user more easily determine the type of iDoctor-supplied definition to use. These question pages are only shown when running the Wizard in Basic mode. An expert would typically use Advanced mode and skip these questions. Each set of responses on these pages leads to a single iDoctor-supplied definition. This definition will be automatically selected on the Collection Options Page once this iDoctor-supplied definition type is determined.

The questions will follow a flow from general categories of problems like the page shown below down to very specific questions that when answered will determine exactly which iDoctor-supplied definition best fits the situation.



PEX Collection Wizard - Problem Type Selection - [redacted]

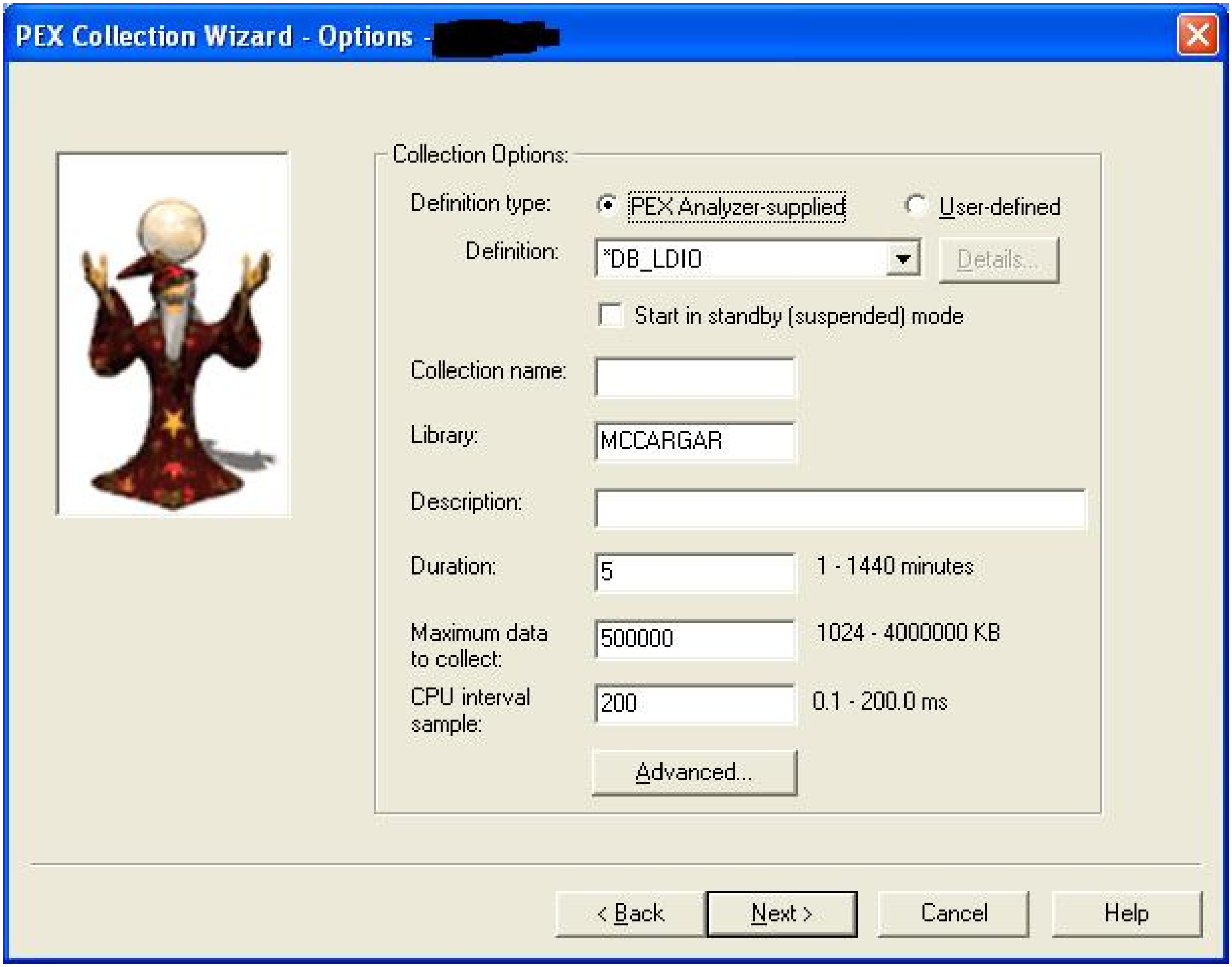
What type of problem do you suspect?

- ● Excessive CPU utilization
- ○ Excessive disk I/O operations
- ○ Excessive file opens
- ○ Excessive disk space consumption
- ○ Excessive allocations to the system heap
- ○ Job(s) perform badly when created/destroyed
- ○ Slow data queue operations
- ○ Slow data area operations
- ○ I need to capture user transaction events.
- ○ Websphere application(s) is running poorly

Table of Contents | Previous | Next

4.3.3.3 Options

The Options Page allows the user to specify the most basic pieces of information about a collection like the PEX definition to use when creating the collection, the name of the collection, library to store the collection in, and more.



PEX Collection Wizard - Options - [redacted]

Collection Options:

Definition type: ● PEX Analyzer-supplied ○ User-defined

Definition: *DB_LDIO

□ Start in standby (suspended) mode

Collection name:

Library: MCCARGAR

Description:

Duration: 5 1 - 1440 minutes

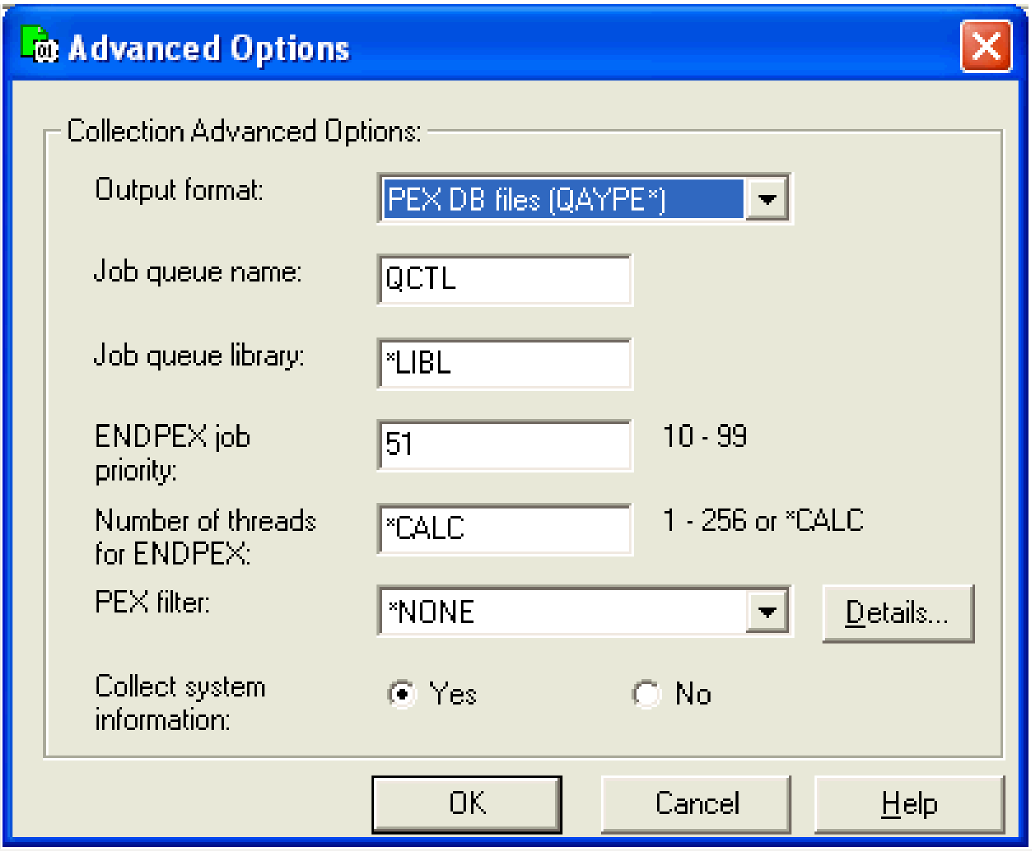
Maximum data to collect: 500000 1024 - 4000000 KB

CPU interval sample: 200 0.1 - 200.0 ms

The following table provide more information about each of the criteria available on this page:

Field	Description

Definition Type	This indicates if the PEX definition will be PEX Analyzer-supplied or user-defined. You can define your own PEX definition using the PEX Definition Wizard or via the ADDPEXDFN command. If you select the user-defined option you can click the Details.. button to quickly see all the information about the PEX definition in a format similar to that provided via PRTPEXRPT command. When using a user-defined PEX definition the rest of the selection pages in the Wizard like Job selection and Task selection are skipped. This is because any Job or Task criteria will come from the PEX definition itself.
Definition	The name of the PEX Analyzer-supplied or user-defined PEX definition. You must select a value from the list. In Basic mode this field will be preselected based on the answers to the problem type questions.
Details button	Displays the properties for the selected user-defined definition.
Start in standby mode option	Select this box to create the collection but to have it be initially in suspended mode. This option is useful if you need to start the collection at a more exact time (right after a test program is called perhaps) because resuming a suspended collection is much faster than starting a new one
Collection name	The name of the PEX collection. The collection name matches the member name created in each of the PEX files stored in the library.
Library name	The name of the library to create the PEX collection in.
Duration (minutes)	The total amount of time to spend collecting data. This value is listed in minutes and must have a value from 1 to 1440. Certain definition types like task switch can generate many million events (records) in a relatively short amount of time. Make sure this value is not too large to avoid ending up with much more data than desired.
Maximum data to collect	The maximum amount of disk space this collection should use in kilobytes. The default value is 500,000. When using a user-defined PEX definition this parameter is ignored because it is provided within the PEX definition.
CPU interval sample	Specifies the size of the interval which CPU samples are taken of the program. A low interval will cause a high number of samples to be taken, and will also cause higher overhead. A low interval will also provide relatively more data. This parameter will be not available if it does not apply to the selected PEX Analyzer-supplied definition.
Advanced button	Displays the interface below for working with some of the options that are less frequently used. Information on these options is provided with the help text for the STRIDOCCOL command.



Advanced Options

Collection Advanced Options:

Output format: PEX DB files (QAYPE*)

Job queue name: QCTL

Job queue library: *LIBL

ENDPEX job priority: 51 10 - 99

Number of threads for ENDPEX: *CALC 1 - 256 or *CALC

PEX filter: *NONE

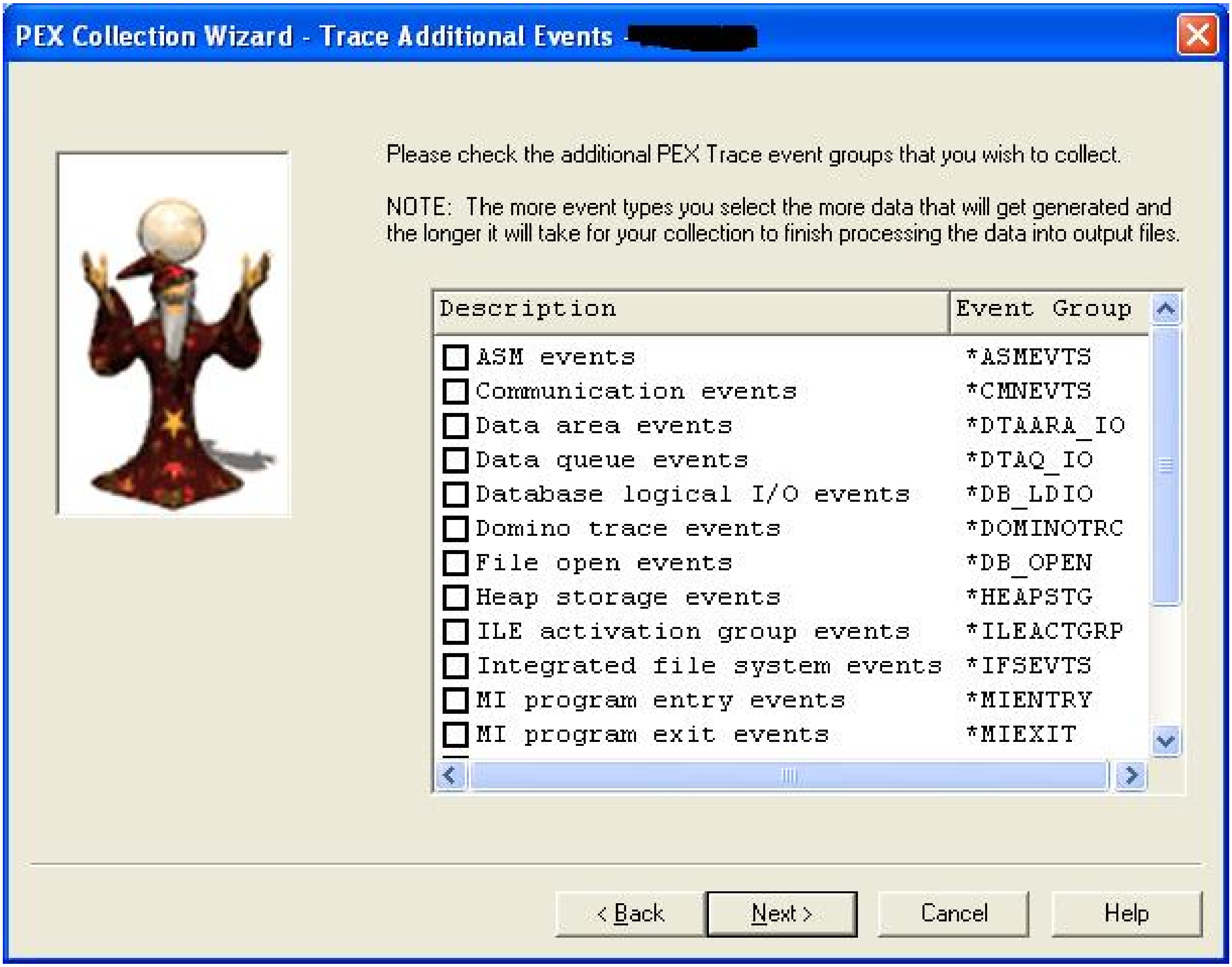
Collect system information: ● Yes ○ No

Table of Contents | Previous | Next

4.3.3.4 Trace Additional Events

This page is only shown when the PEX Analyzer-supplied definition is one that generates a PEX Trace collection. You will not see this page when creating a collection using a user-defined PEX definition.

The list contains additional event groups that can be added to the collection. The more event types you select the more data that will get generated and the longer it will take for your collection to finish processing the data into the PEX database files.



PEX Collection Wizard - Trace Additional Events - [redacted]

Please check the additional PEX Trace event groups that you wish to collect.

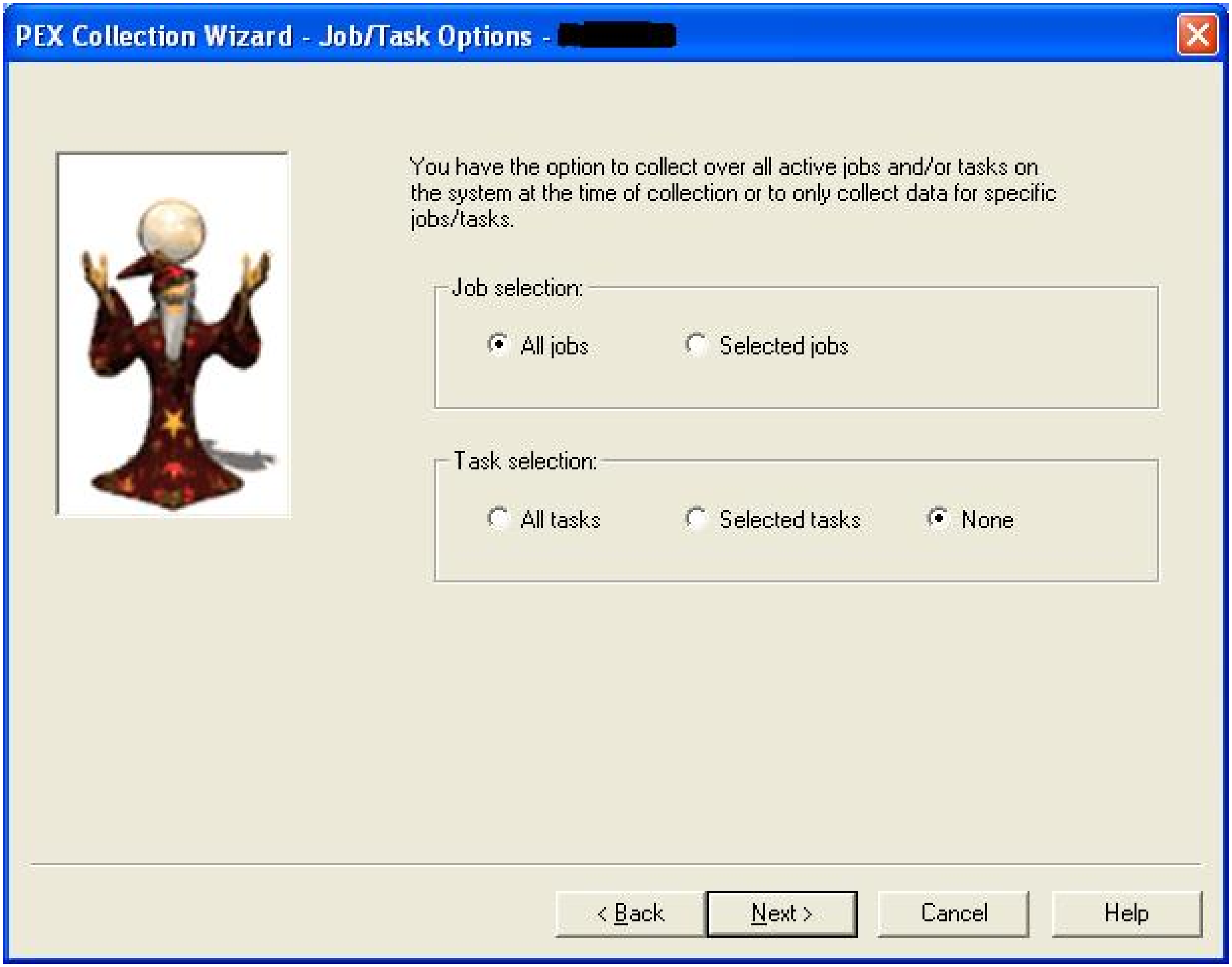
NOTE: The more event types you select the more data that will get generated and the longer it will take for your collection to finish processing the data into output files.

Description	Event Group
□ ASM events	*ASMEVTS
□ Communication events	*CMNEVTS
□ Data area events	*DTAARA_IO
□ Data queue events	*DTAQ_IO
□ Database logical I/O events	*DB_LDIO
□ Domino trace events	*DOMINOTRC
□ File open events	*DB_OPEN
□ Heap storage events	*HEAPSTG
□ ILE activation group events	*ILEACTGRP
□ Integrated file system events	*IFSEVTS
□ MI program entry events	*MIENTRY
□ MI program exit events	*MIEXIT

Table of Contents | Previous | Next

4.3.3.5 Job/Task Options

On this page you may decide if you would like to select specific jobs or tasks to include in the PEX collection. In some cases selecting all jobs is not allowed based on the type of collection being created. Selecting to include tasks in the collection is always optional.



PEX Collection Wizard - Job/Task Options - [redacted]

You have the option to collect over all active jobs and/or tasks on the system at the time of collection or to only collect data for specific jobs/tasks.

Job selection:

● All jobs ○ Selected jobs

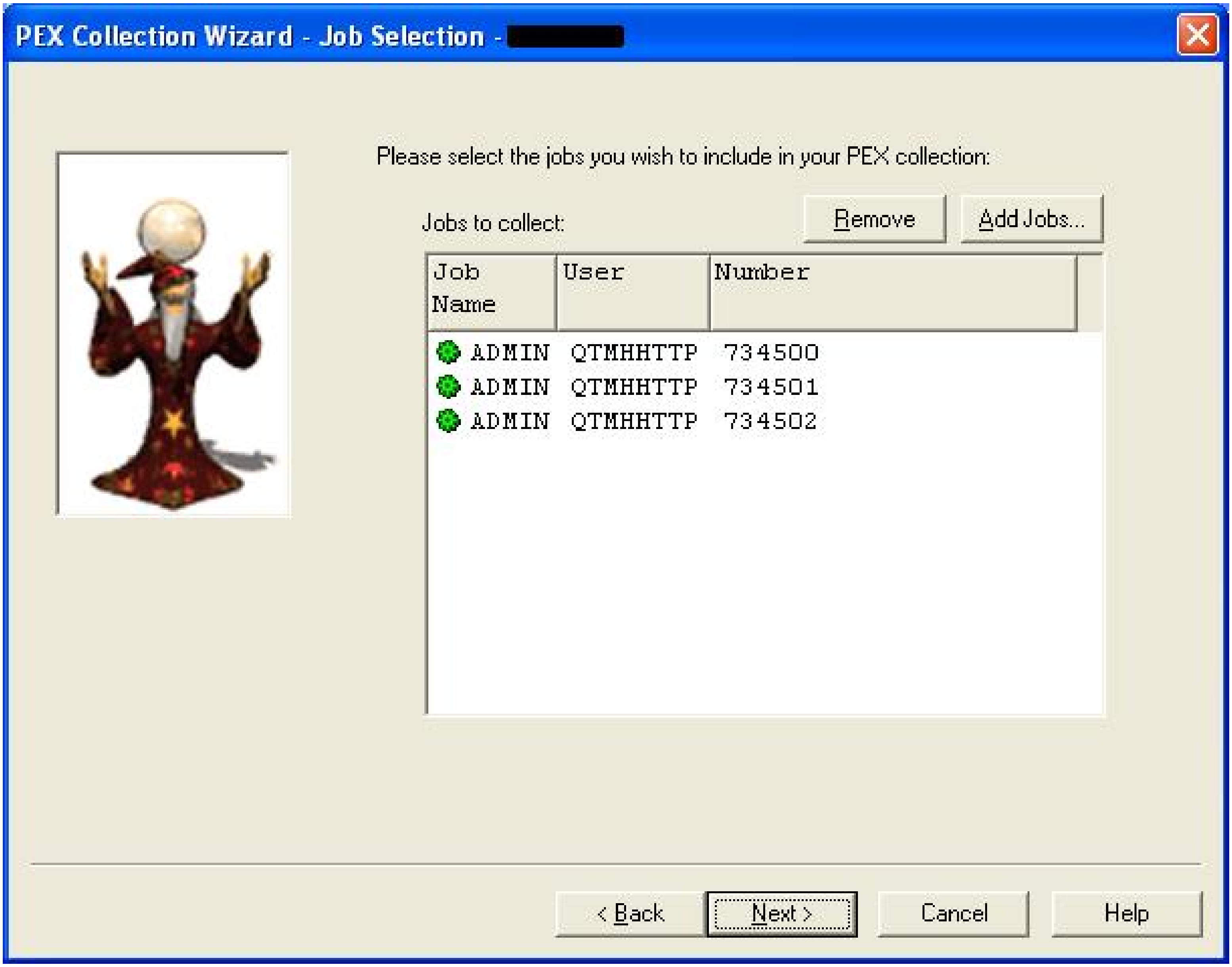
Task selection:

○ All tasks ○ Selected tasks ● None

Table of Contents | Previous | Next

4.3.3.6 Job Selection

The job selection page displays a list of selected job information to be captured in the PEX collection. There are also two buttons on this page used to add or remove jobs from the list.



PEX Collection Wizard - Job Selection - [redacted]

Please select the jobs you wish to include in your PEX collection:

Jobs to collect:

Job Name	User	Number
ADMIN	QTMHHTTP	734500
ADMIN	QTMHHTTP	734501
ADMIN	QTMHHTTP	734502

The table below summarizes the different elements on this page:

Field	Description
Jobs list	A list of jobs to collect information about in the PEX collection.

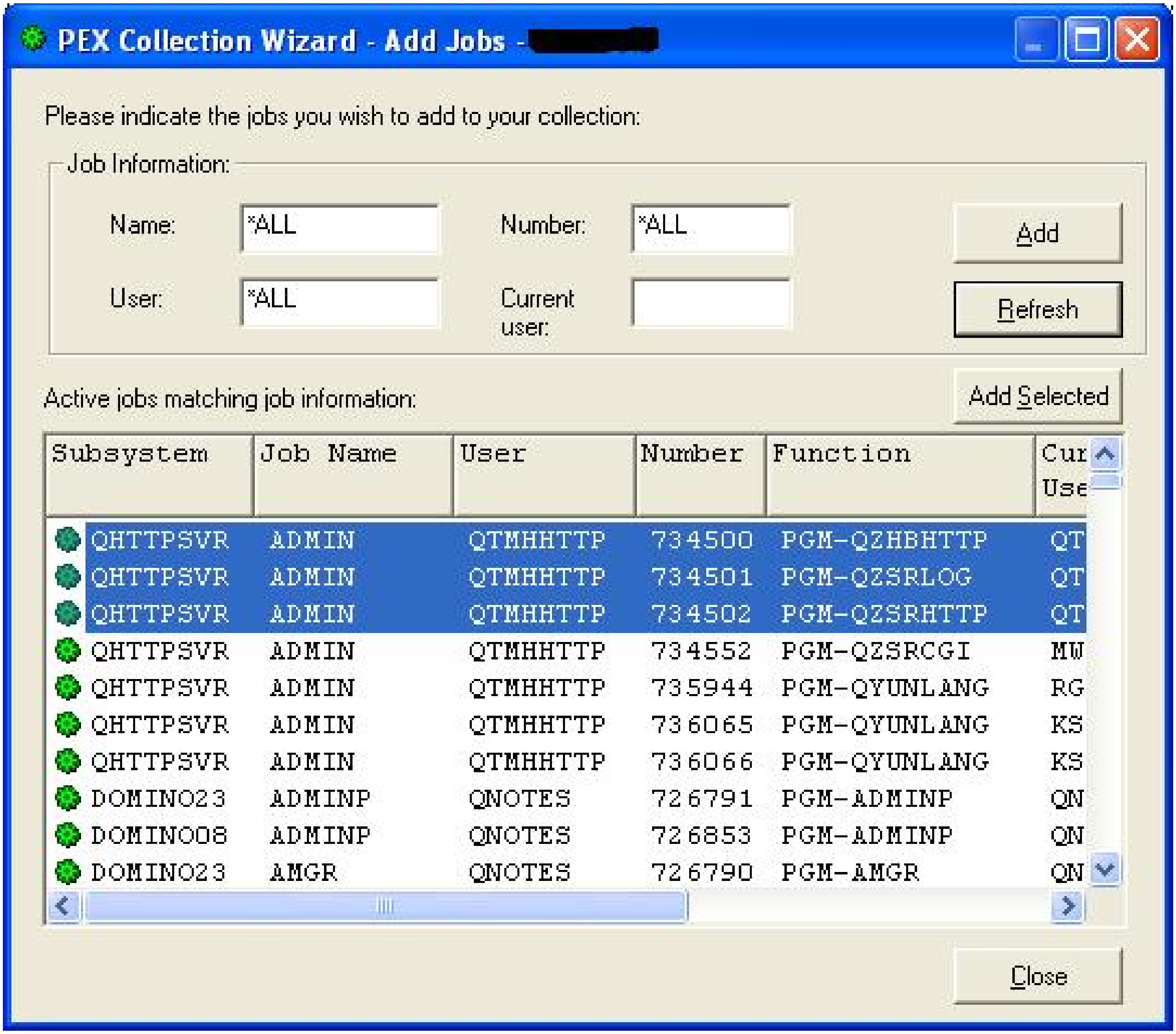
Remove button	This button removes the selected jobs from the list.
Add Jobs button	Use this button to open the Add Jobs Window (discussed in the next section). This window is used to select and add additional jobs to the list.

Table of Contents | Previous | Next

4.3.3.7 Add Jobs Window

The add jobs window allows a user to add jobs to the Job Selection page in the wizard. Job information can be of two types: generic job name/generic job user/generic job number -or- job name/job user/job number.

The "Job Information" portion of the window includes text fields used to define a generic job to add to the Job Selection Page or to use as a filter when refreshing the list of jobs shown in the window. The Add button will add the current generic job to the Job Selection page and the Add Selected button will add the selected jobs from the active jobs list to the Job Selection page.



PEX Collection Wizard - Add Jobs - [redacted]

Please indicate the jobs you wish to add to your collection:

Job Information:

Name: *ALL

Number: *ALL

User: *ALL

Current user:

Active jobs matching job information:

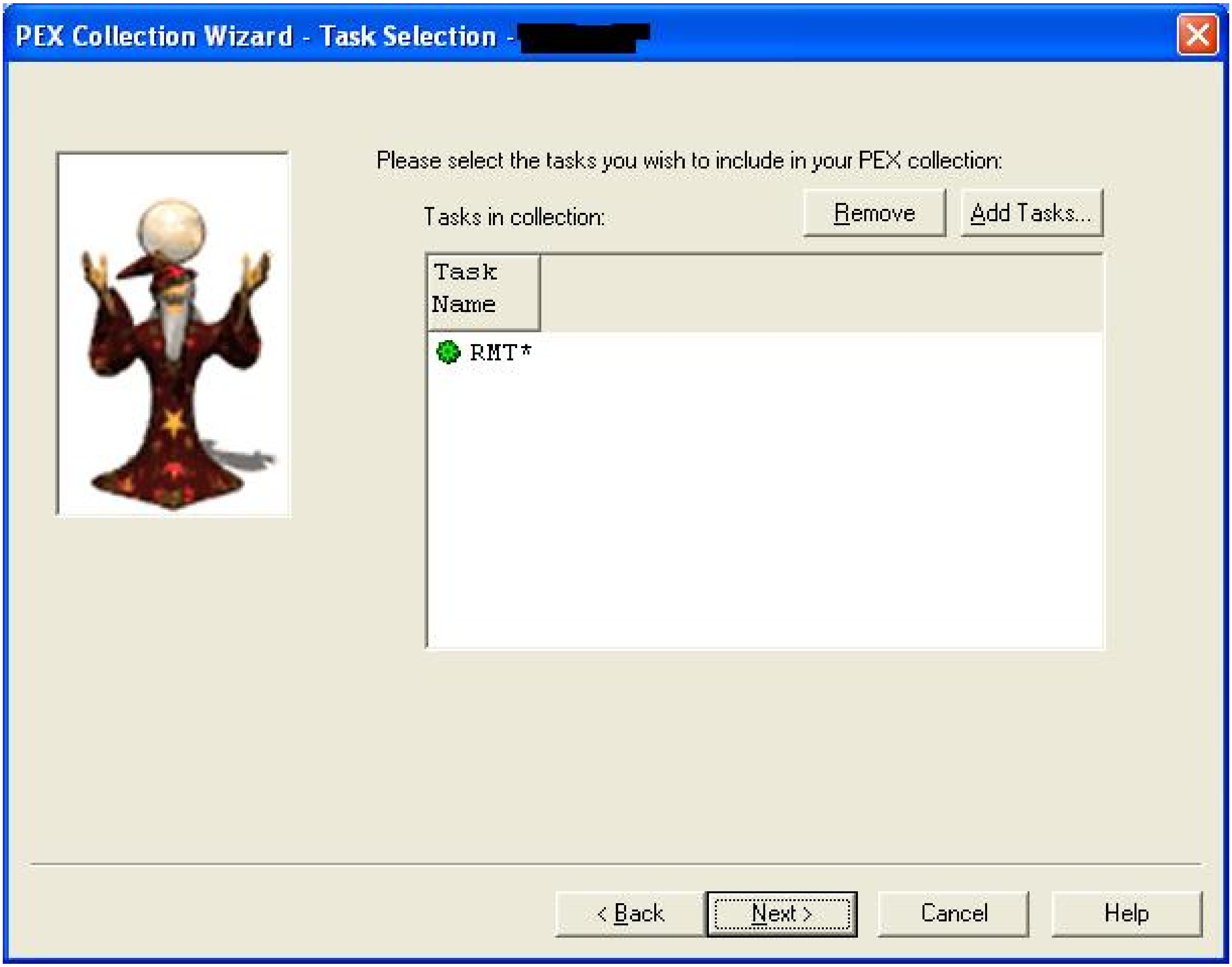
Subsystem	Job Name	User	Number	Function	Cur Use
QHTTPSVR	ADMIN	QTMHHTTP	734500	PGM-QZHBHTTP	QT
QHTTPSVR	ADMIN	QTMHHTTP	734501	PGM-QZSRLOG	QT
QHTTPSVR	ADMIN	QTMHHTTP	734502	PGM-QZSRHTTP	QT
QHTTPSVR	ADMIN	QTMHHTTP	734552	PGM-QZSRCGI	MW
QHTTPSVR	ADMIN	QTMHHTTP	735944	PGM-QYUNLANG	RG
QHTTPSVR	ADMIN	QTMHHTTP	736065	PGM-QYUNLANG	KS
QHTTPSVR	ADMIN	QTMHHTTP	736066	PGM-QYUNLANG	KS
DOMINO23	ADMINP	QNOTES	726791	PGM-ADMINP	QN
DOMINO08	ADMINP	QNOTES	726853	PGM-ADMINP	QN
DOMINO23	AMGR	QNOTES	726790	PGM-AMGR	QN

The table below summarizes the different elements on this page:

Field	**Description**
Name	A text field for entering a generic or specific job name. When specifying a generic name use a * at the end of the job name.
User	A text field for entering a generic or specific job user. When specifying a generic name use a * at the end of the job user name.
Number	A text field for entering a specific job number or *ALL.
Add button	This button will add the current job name/user/number values in the text fields to the Job Selection page. This can be used to add a generic job name/user/number value such as QZ*/FRED/*ALL This value indicates all job names starting with QZ, for job user FRED.
Refresh button	This button is used to refresh the active jobs list based on the current values specified in the name, user and number text fields.
Add Selected button	Use this button to add the selected active jobs to the Job Selection Page.
Active jobs matching job information list	This list shows all active jobs on the system matching the current Job information specified. When this window is first open the list will show all active jobs on the system.

4.3.3.8 Task Selection

The task selection page displays a list of selected tasks (or generic task names) to be captured in the PEX collection. There are also two buttons on this page used to add or remove entries from the list.



PEX Collection Wizard - Task Selection - [redacted]

Please select the tasks you wish to include in your PEX collection:

Tasks in collection:

The table below summarizes the different elements on this page:

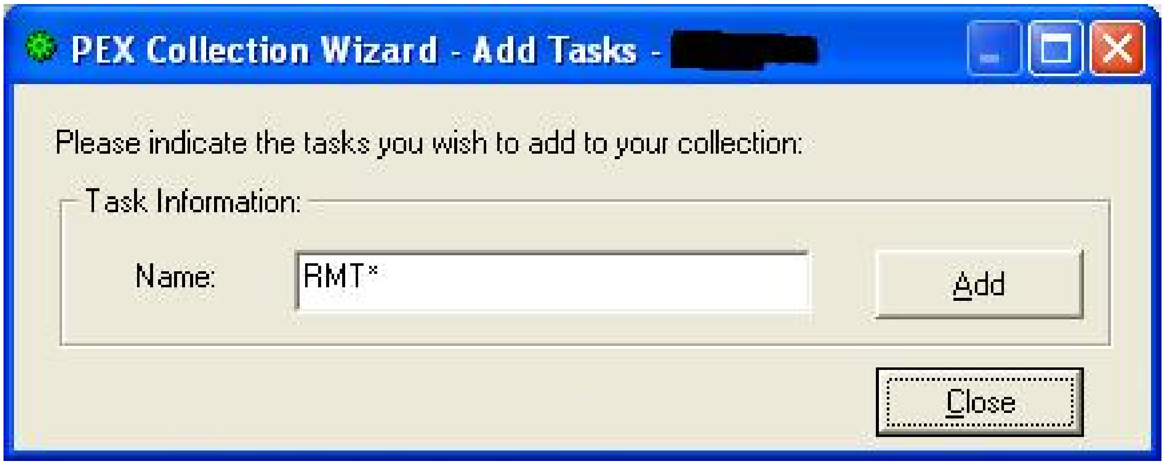
Field	Description
Tasks list	A list of tasks (specific task names or generic task names) that will be captured in the PEX collection.

Remove button	This button removes the selected entries from the list.
Add Tasks button	Use this button to open the Add Tasks Window (discussed in the next section). This window is used to add task information to the task list.

Table of Contents Previous Next

4.3.3.9 Add Tasks Window

The add tasks window allows a user to add tasks to the Task Selection page in the wizard. The task name can either be *ALL, *NONE, a generic task name like Q*, or a specific task name. Change the task name field and click the add button for each task that you would like to capture in your PEX collection.



PEX Collection Wizard - Add Tasks - [redacted]

Please indicate the tasks you wish to add to your collection:

Task Information:

Name: RMT*

The table below summarizes the different elements on this page:

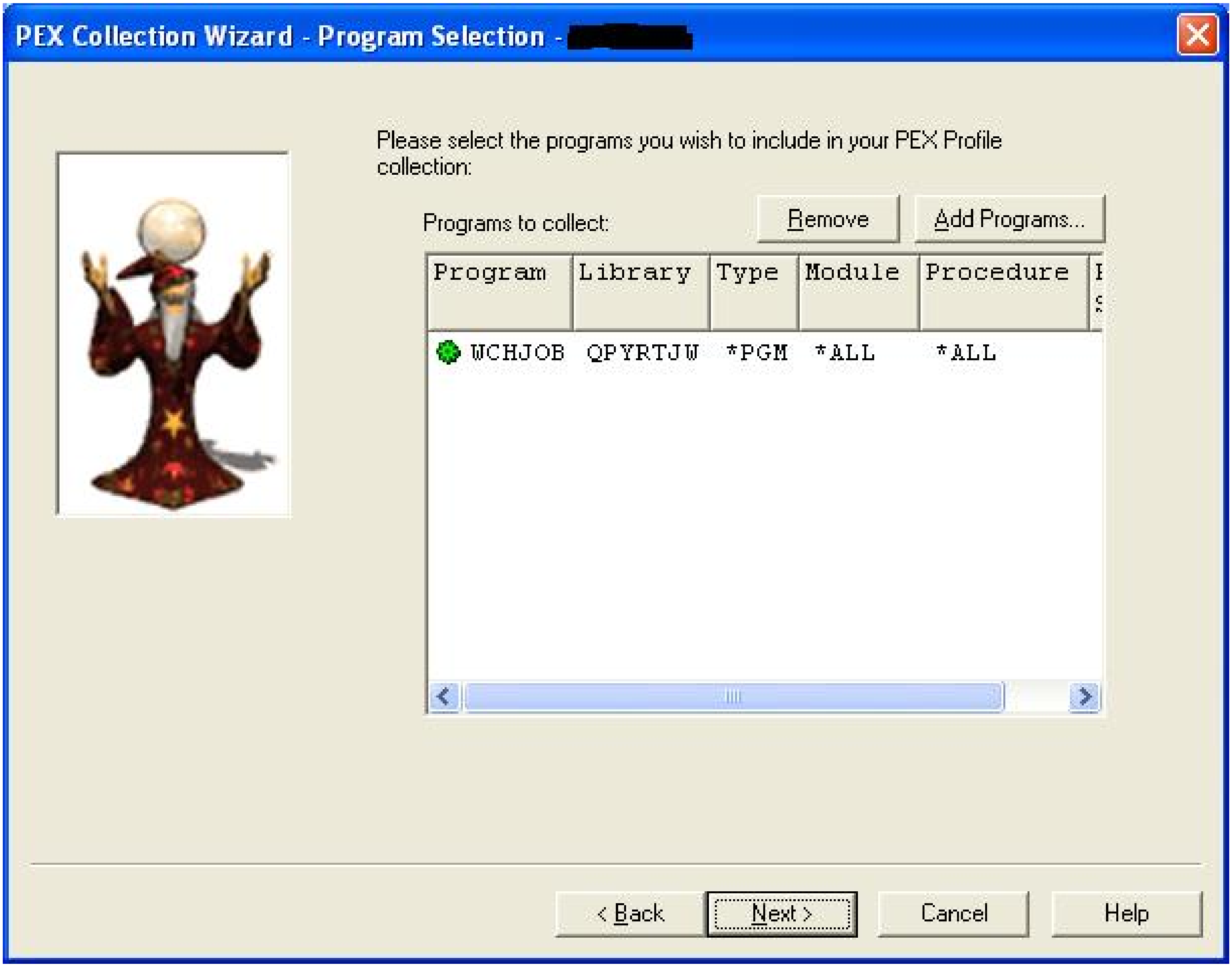
Field	Description
Name	A text field for entering a generic or specific task name. This value can also be *ALL or *NONE. When specifying a generic name use a * at the end of the task name.
Add button	This button will add the current task information to the Task Selection page.
Close button	Closes this window

Table of Contents | Previous | Next

4.3.3.10 Program Selection

The program selection page allows the user to select up to 16 program/module/procedure entries when creating a PEX Profile collection. This window is only displayed when using the PEX Analyzer-supplied definition *PROFILE.

This page displays a list of selected program information to be captured in the PEX collection. Note that in order to get any useful information from this definition type you must select programs that have the enable for profiling flag turned on. There are also two buttons on this page used to add or remove programs from the list.



PEX Collection Wizard - Program Selection - [redacted]

Please select the programs you wish to include in your PEX Profile collection:

Programs to collect:

Program	Library	Type	Module	Procedure
WCHJOB	QPYRTJW	*PGM	*ALL	*ALL

The table below summarizes the different elements on this page:

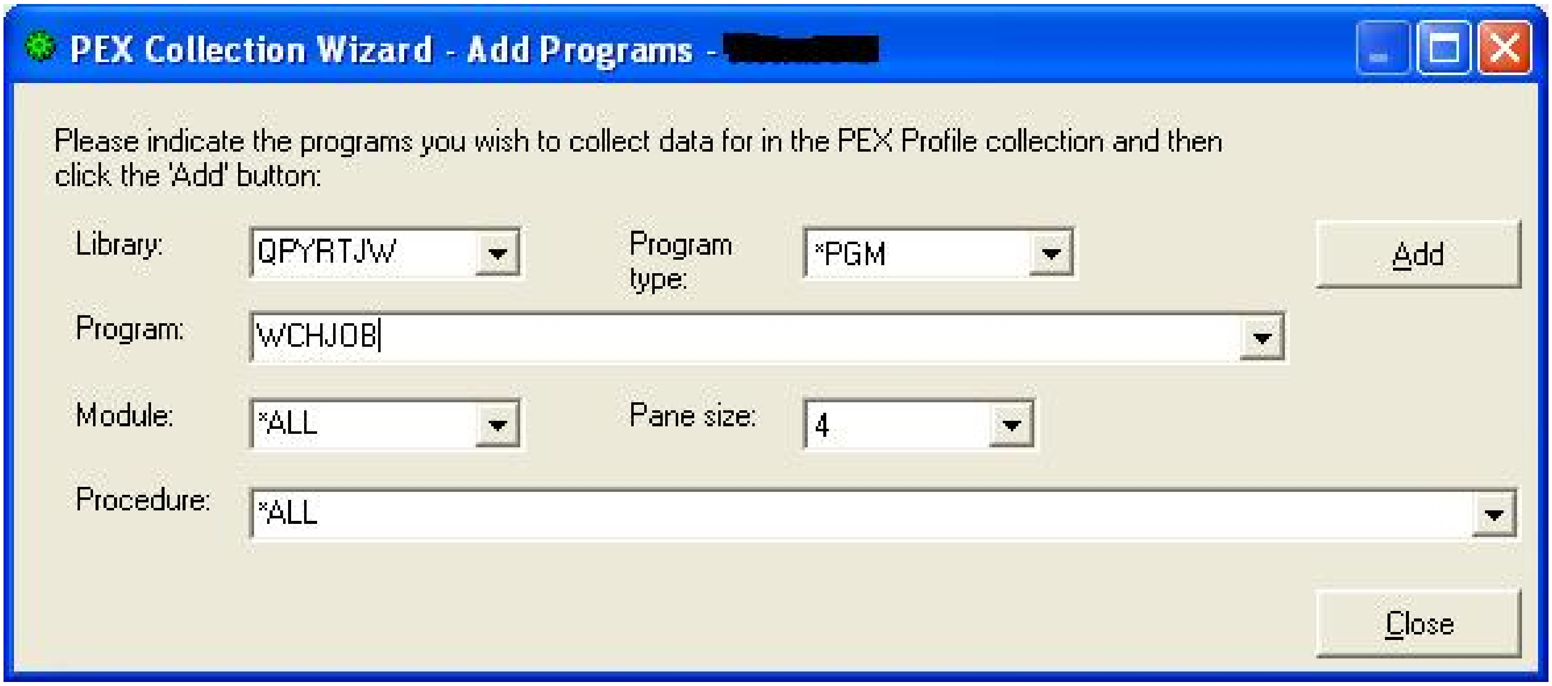
Field	Description
Programs list	A list of program information that will be captured in the PEX Profile collection.
Remove button	This button removes the selected program information from the Programs list.
Add Programs button	Use this button to open the Add Programs Window (discussed in the next section). This window is used to add program information to the Programs list.

Table of Contents Previous Next

4.3.3.11 Add Programs Window

The add programs window allows a user to browse any programs/service programs on the system using generic program and library names for the purpose of adding them to a PEX Profile collection. After finding the programs you want to add to the collection click the Add button to add the selected program/module/procedure to the list.

The enable profiling flag must be turned on in the program and module you select in order to add the program/module/procedure information to the Program Selection Page of the Wizard.



PEX Collection Wizard - Add Programs - [redacted]

Please indicate the programs you wish to collect data for in the PEX Profile collection and then click the 'Add' button:

Library: QPYRTJW

Program type: *PGM

Program: WCHJOB

Module: *ALL

Pane size: 4

Procedure: *ALL

The table below summarizes the different elements on this page:

Field	Description
Library	A dropdown list for entering or selecting the library name.
Program	A dropdown list for entering or selecting the program name.
Type	This drop down lists contains the values *PGM and *SRVPGM. This offers the user the choice of viewing programs or service program objects.
Pane size	The pane size is the number of consecutive program instruction addresses assigned to each counter. The smaller the pane size, the more fine-grained the program profile information will be.
Add button	This button will add the selected program, module and procedure to the Program Selection Page. The program added must be enabled for profiling.

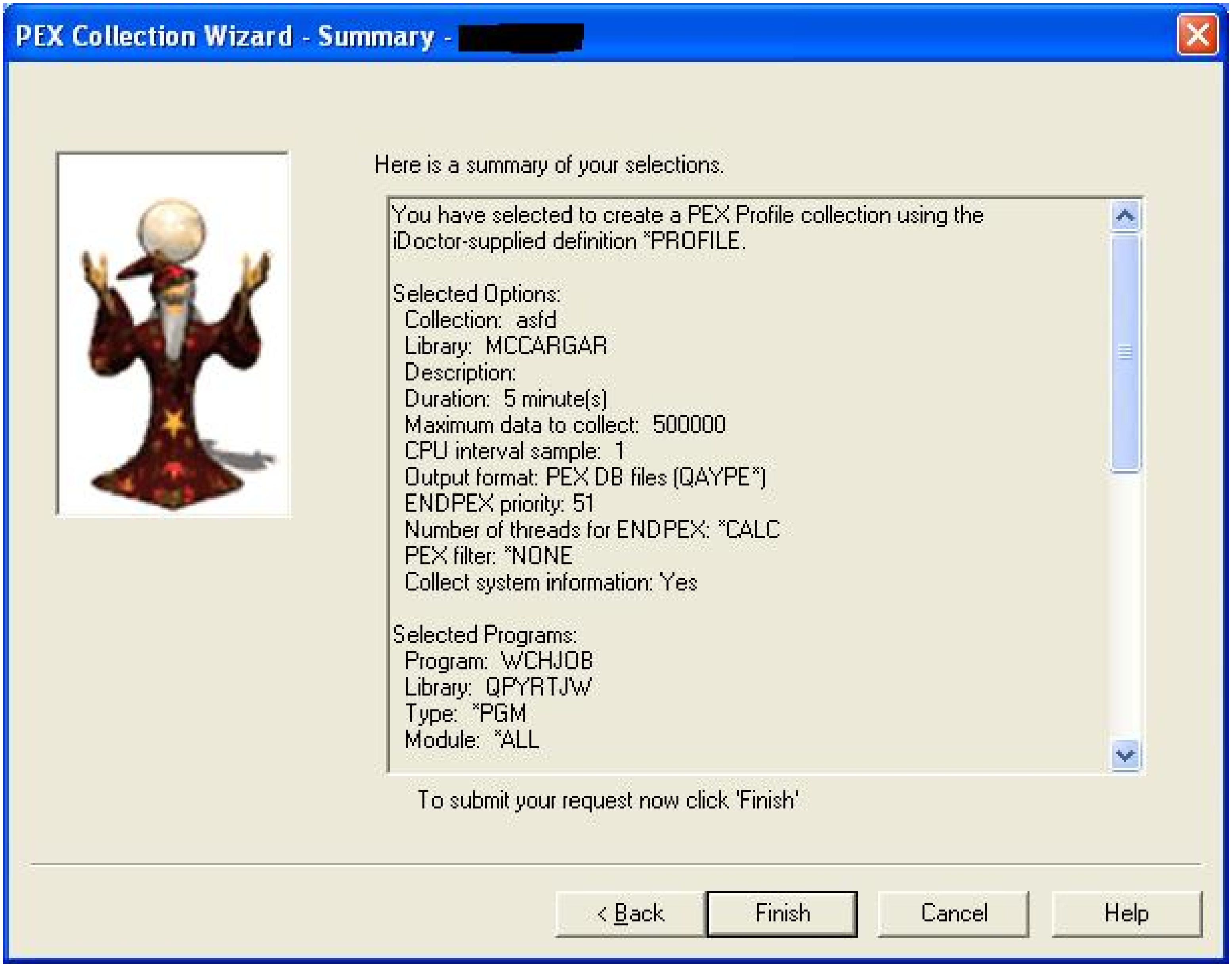
Program List	A dropdown list of programs found within the selcted library. Changing your selected program in the list will update the module dropdown list showing the module information found within the program.
Module List	A dropdown list of modules found within the selected program in the Program List. Changing the selected module in the list will refresh the procedure list showing the procedure information found within the seleted program.
Procedure List	A dropdown list of procedures found within the selected module and program.
Close button	Close the Add Programs window.

4.3.3.12 Summary

The final page of the PEX Collection Wizard presents a summarization of all of the input provided in the wizard. It lists all of the details about the selected jobs/tasks/programs as well as information from the Options page like collection name and duration.

To submit the job to create the PEX Collection click on the Finish button. After submitting your collection go to the library that the collection is going to be created in to see the status of the PEX collection in progress.

At the bottom of this window is the command string that will be issued by this wizard. This command string can be copied to a green screen window and ran there if necessary. In this way the GUI can be used to construct the STRIDOCCOL command string for another system if the GUI is not available on that system.



PEX Collection Wizard - Summary - [redacted]

Here is a summary of your selections.

You have selected to create a PEX Profile collection using the iDoctor-supplied definition *PROFILE.

Selected Options:
 Collection: asfd
 Library: MCCARGAR
 Description:
 Duration: 5 minute(s)
 Maximum data to collect: 500000
 CPU interval sample: 1
 Output format: PEX DB files (QAYPE*)
 ENDPEX priority: 51
 Number of threads for ENDPEX: *CALC
 PEX filter: *NONE
 Collect system information: Yes

Selected Programs:
 Program: WCHJOB
 Library: QPYRTJW
 Type: *PGM
 Module: *ALL

To submit your request now click 'Finish'

Table of Contents | Previous | Next

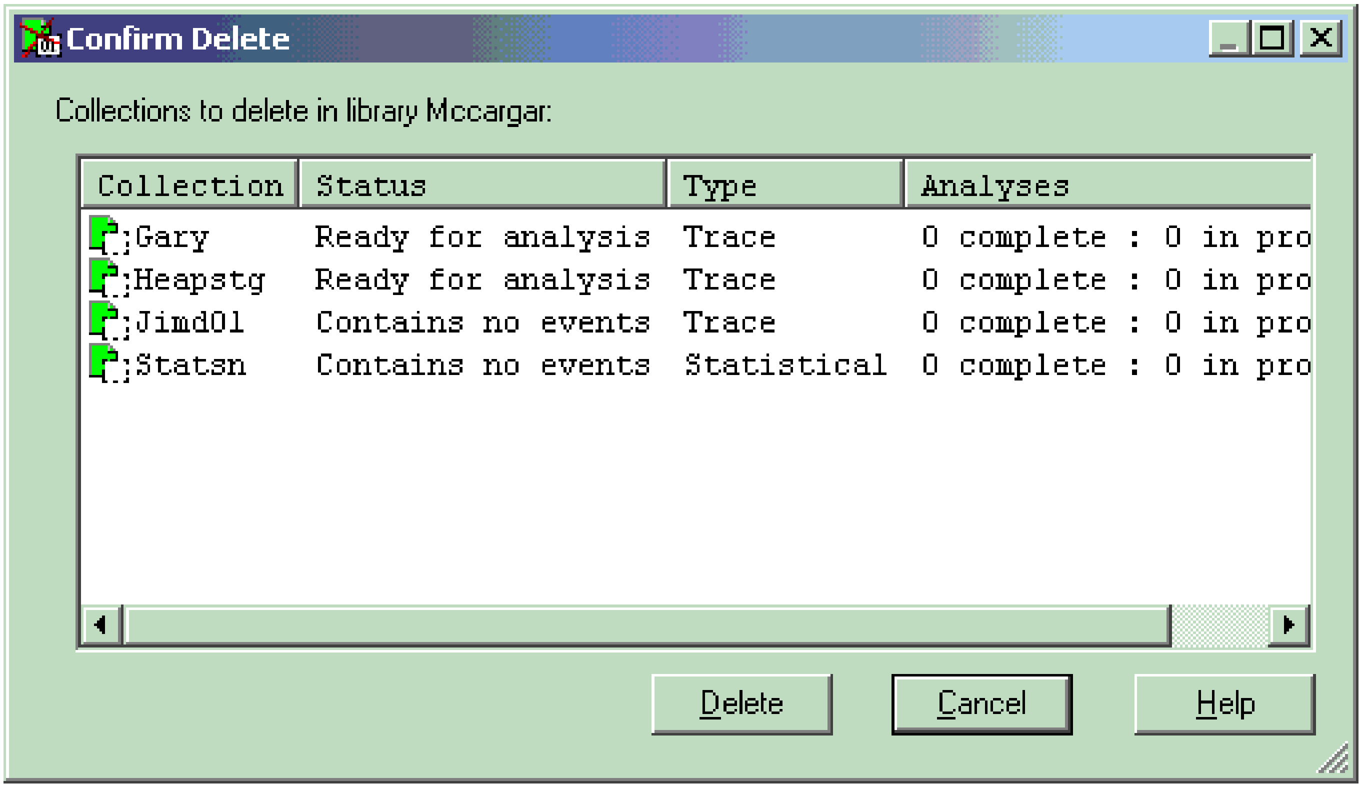
4.3.4 Defining

The events to capture in a PEX collection are defined using a PEX definition. To learn more visit the section on PEX Definitions.

Table of Contents Previous Next

4.3.5 Deleting

A user may choose to delete one or more collections using the Delete... menu found by right-clicking on a collection. If you have trouble deleting a collection there may be someone else using it (possibly you if you have some of its data open in the Data Viewer). Ensure that all table or graph views over the collections you want to delete have been closed down before trying to delete them.



Collections to delete in library Mccargar:

Collection	Status	Type	Analyses
Gary	Ready for analysis	Trace	0 complete : 0 in pro
Heapstg	Ready for analysis	Trace	0 complete : 0 in pro
Jimd01	Contains no events	Trace	0 complete : 0 in pro
Statsn	Contains no events	Statistical	0 complete : 0 in pro

[An example of the Delete Collection Dialog]

Table of Contents Previous Next

4.3.6 Generating analyses

A PEX collection may be analyzed by right-clicking on it and choosing the Analyze Data... menu. The types of analyses available depends on the type of data has been collected. Visit the documentation on PEX Analyses for more information.

Table of Contents | Previous | Next

4.3.7 Property Pages

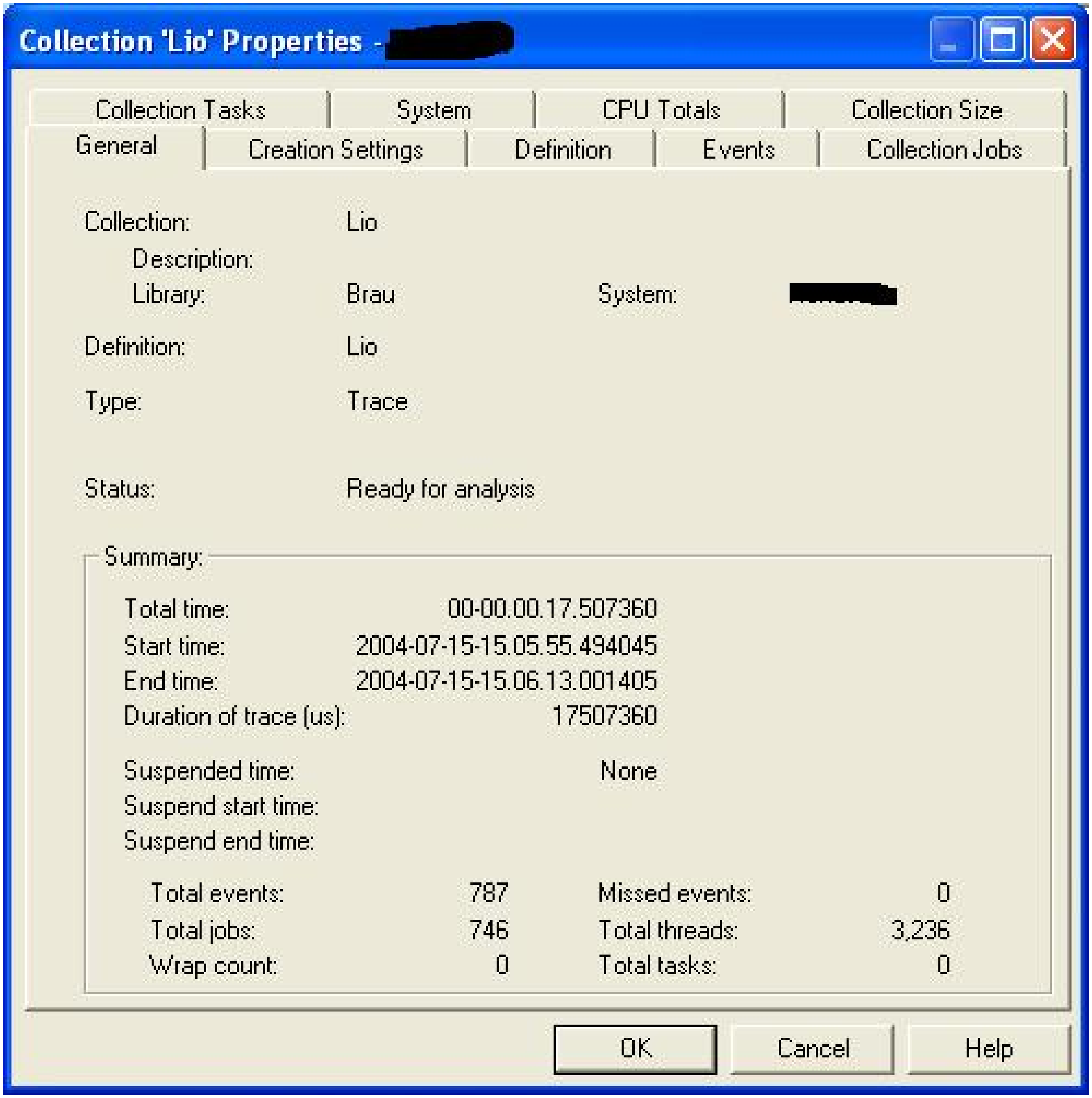
A PEX collection contains a huge amount of performance information which is summarized within the PEX collection property pages. A user may invoke the property pages by right-clicking on the desired collection and choosing Properties.

There are three primary types of collections: Trace, Profile and Statistical. The property pages available to the user varies slightly based on the type of collection used. Each property page will be discussed next along with any differences that exist based on the collection type involved.

Table of Contents Previous Next

4.3.7.1 General

The General property page provides a summary of the most basic information about a collection. This information includes the name of the collection, the system it was created on, and basic totals for the entire collection such as the number of events gathered and the number of jobs and tasks collected.



Collection 'Lio' Properties - [redacted]

Collection: Lio
Description:
Library: Brau
System: [redacted]
Definition: Lio
Type: Trace
Status: Ready for analysis

Summary:
Total time: 00-00.00.17.507360
Start time: 2004-07-15-15.05.55.494045
End time: 2004-07-15-15.06.13.001405
Duration of trace (us): 17507360
Suspended time: None
Suspend start time:
Suspend end time:

Total events:	787	Missed events:	0
Total jobs:	746	Total threads:	3,236
Wrap count:	0	Total tasks:	0

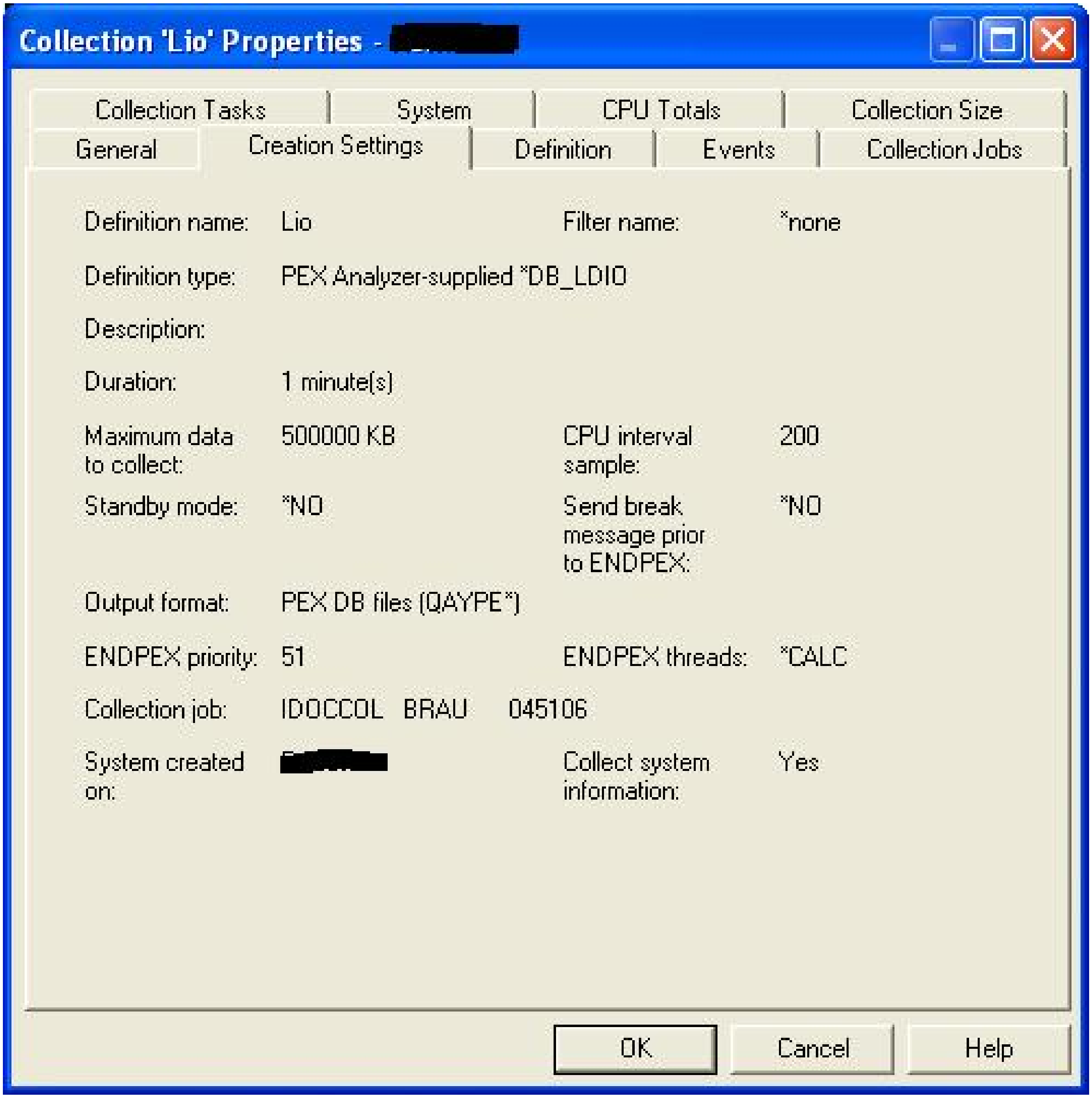
The following information is displayed on the General property page:

Field Name	**Field Description**
Collection name	PEX collection name. The collection name is a unique database file member name in the specified library.
Description	Collection description. A user-defined description given to a collection when it is created.
Library	Library containing the collection.
System	The current system name.
Definition	The name of the PEX definition that was used to create the collection.
Type	Trace, statistical, and profile are the three primary types of PEX data. There are two subtypes of statistical - hierarchical and flat.
Status	Indicates whether the collection is useable by PEX Analyzer or indicates the status of the job creating the collection if the collection is currently in the process of being created.
Show Job Log button	This button is used to display the job log of the job used for creating the PEX collection. If the job log is not found on the system the button will not be visible.
Total time	Total elapsed time the collection was gathering data.
Start time	Date and time the collection was created.
End time	Date and time the collection stopped gathering data.
Duration of trace (us):	The total time of the trace in microseconds.
Suspended time (us)	Total number of microseconds the collection was suspended during data gathering. If this field is 0 the collection was not suspended.
Suspend start time	Date and time the collection was suspended. If this field is blank the collection was not suspended.
Suspend end time	Date and time the collection was resumed. If this field is blank the collection was not suspended.
Total events	Number of individual events that occurred within the course of the collection.
Total jobs	Number of unique jobs captured in the collection.
Missed events	Number of events missed. If this is greater than zero your collection data may be damaged.
Total threads	Number of unique threads captured in the collection.
Total tasks	Number of unique tasks captured in the collection.
Wrap count	Indicates if the collection data was wrapped. In some cases the analysis output is unreliable if the collection data was wrapped.

Table of Contents | Previous | Next

4.3.7.2 Creation settings

The Creation settings property page provides information about the parameters used to create the collection. This window is only available if the STRIDOCCOL command (or the GUI) was used to create the collection.



Collection 'Lio' Properties - [redacted]

Definition name:	Lio	Filter name:	*none
Definition type:	PEX Analyzer-supplied *DB_LDIO		
Description:			
Duration:	1 minute(s)		
Maximum data to collect:	500000 KB	CPU interval sample:	200
Standby mode:	*NO	Send break message prior to ENDPEX:	*NO
Output format:	PEX DB files (QAYPE*)		
ENDPEX priority:	51	ENDPEX threads:	*CALC
Collection job:	IDOCCOL BRAU 045106		
System created on:	[redacted]	Collect system information:	Yes

The following information is displayed on this property page:

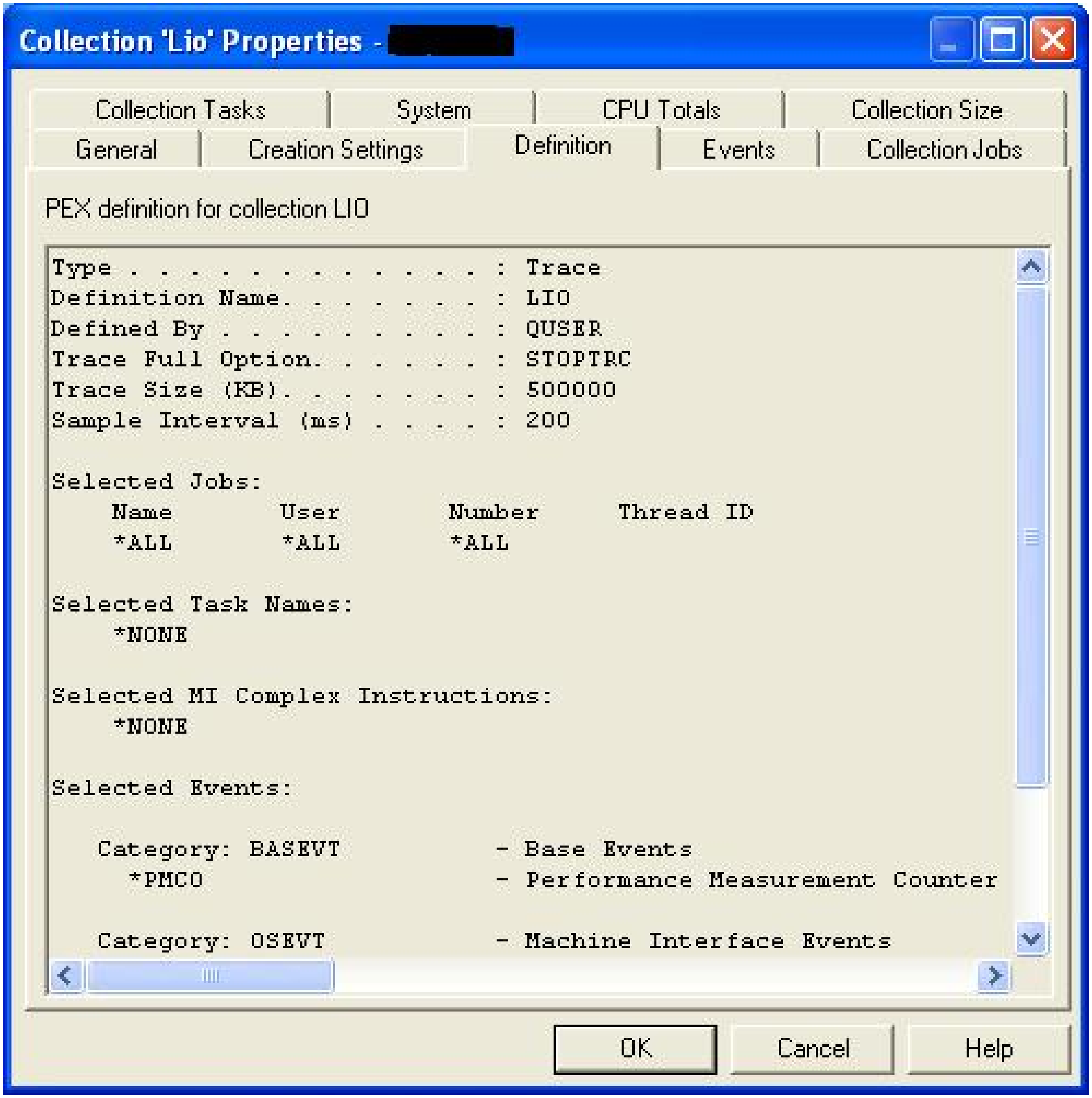
Field Name	**Field Description**
Definition name	The name of the PEX definition that was used to create the PEX collection.
Filter	The name of the PEX filter if any that was used during creation of the collection.
Definition type	The type of definition. This is either a PEX Analyzer supplied type (a PEX definition that came with PEX Analyzer) or a user-defined type (a user created PEX definition).
Description	A description for the collection.
Duration	The user specified time to run the collection. This may not reflect the actual total time if the collection was ended early.
Maximum data to collect	Indicates how much data should be collected in the PEX collection (in MB) before the data is wrapped.
CPU interval sample	The value, in milliseconds, is used to cause a process to wake up at this inteval value, and indicate what pgm is executing and the percentage of CPU used between this wakeup and the previous wakeup within this job. By setting this to a low value, 1-10ms, you can profile the entire system, if all jobs are within the collection, and determine what pgm's are using the CPU.
Standy mode	This value indicates if the user indicated the collection to be started in a suspended state.
Send break message prior to ENDPEX	Whether this option was enabled at the time of collection. See STRIDOCCOL command help for more details.
Output format	Indicates if the files were specified to be written to the QAYPE* PEX files or to a single file for transporting to another system.
ENDPEX priority	This value controls the run priority of the job/threads created for use by the ENDPEX processing of the collected data.
ENDPEX threads	The job priority to use when ENDPEX is running.
Collection job	The name of the job that created the collection.
System created on	The name of the system the collection was created on
Collect system information	Whether or not WRKSYSSTS and WRKDSKSTS is collected along with the PEX data.

Table of Contents | Previous | Next

4.3.7.3 Definition

The Definition settings property page provides information about the PEX definition that was used to create the collection. This information is provided in format nearly identical to that given by PRTPEXRPT or PRTPEXDFN commands.

This panel is only available at releases V5R2 and higher.

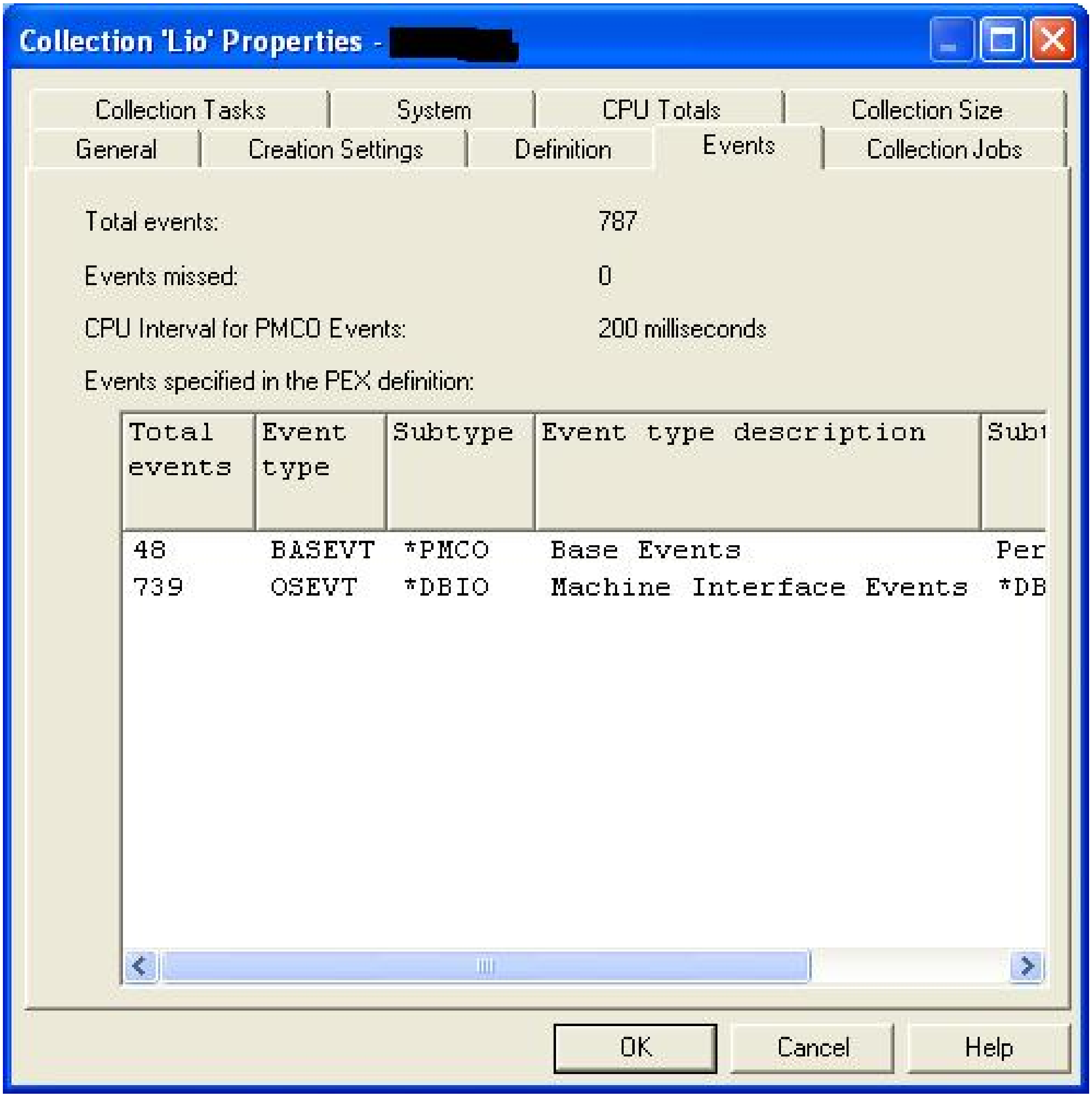


PEX definition for collection LIO

```
Type . . . . . . . . . . . . . : Trace
Definition Name. . . . . . . . : LIO
Defined By . . . . . . . . . . : QUSER
Trace Full Option. . . . . . . : STOPTRC
Trace Size (KB). . . . . . . . : 500000
Sample Interval (ms) . . . . . : 200

Selected Jobs:
    Name        User        Number      Thread ID
    *ALL        *ALL        *ALL

Selected Task Names:
    *NONE

Selected MI Complex Instructions:
    *NONE

Selected Events:

   Category: BASEVT          - Base Events
     *PMCO                   - Performance Measurement Counter

   Category: OSEVT           - Machine Interface Events
```

4.3.7.4 Events

The Events property page provides information about the events that were captured within the collection.



Total events: 787

Events missed: 0

CPU Interval for PMCO Events: 200 milliseconds

Events specified in the PEX definition:

Total events	Event type	Subtype	Event type description	Sub
48	BASEVT	*PMCO	Base Events	Per
739	OSEVT	*DBIO	Machine Interface Events	*DB

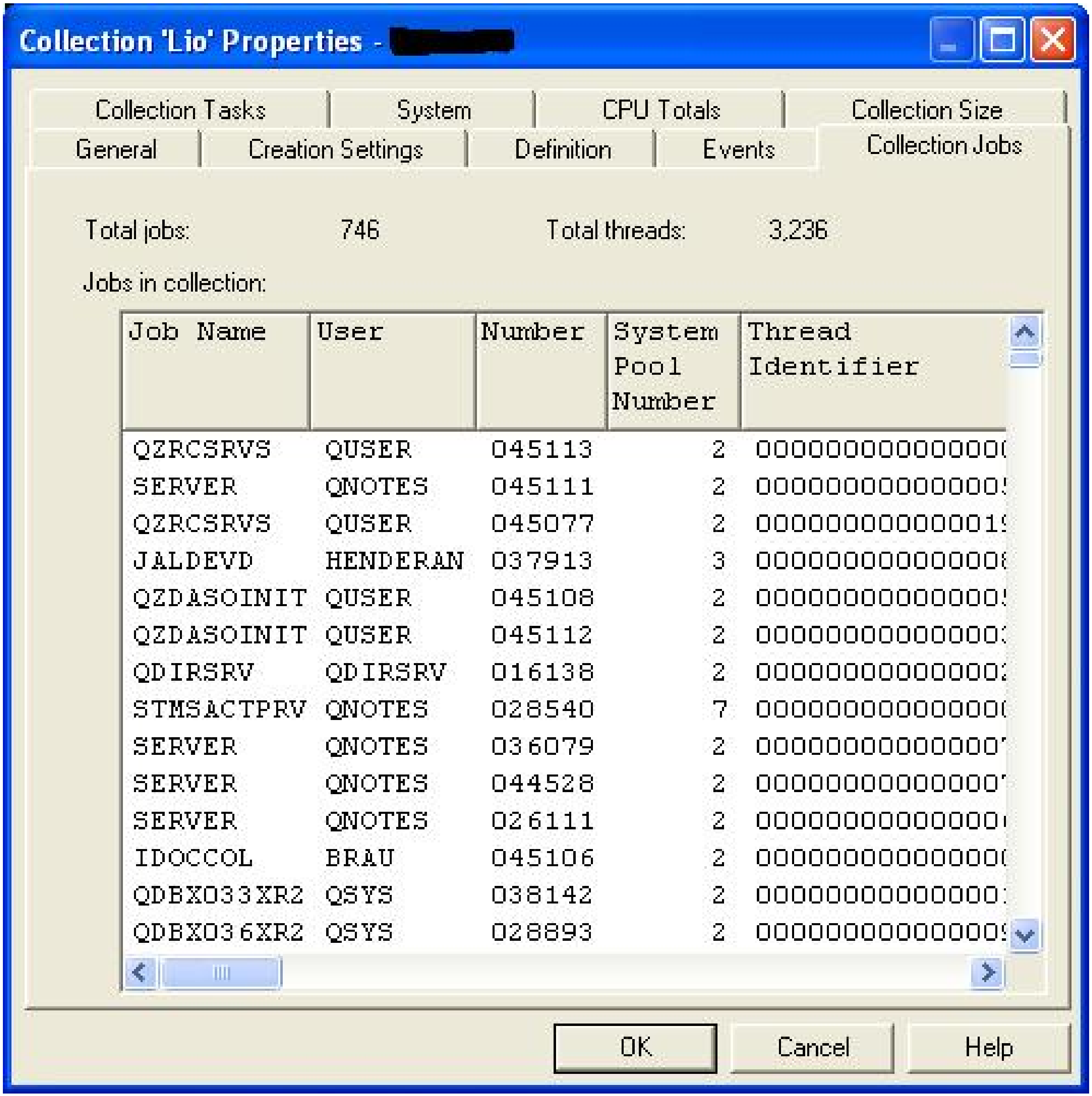
The following information is displayed on the Events property page:

Field Name	**Field Description**
Total events	The total number of instances of requested events that exist within the collection.
Events missed	The total number of events not captured within the collection.
CPU Interval for PMCO Events	A value used to determine the frequency of Performance Measurement Counter Overflow (PMCO) events. A low value generates more PMCO events than a high value.
Events specified in the PEX definition list	A list of events requested to be included in the collection data. An event is an event type/subtype combination. This combination of event type and subtype is referred to as a PEX trace event.

Table of Contents | Previous | Next

4.3.7.5 Collection Jobs

The Collection Jobs property page lists every job and thread that was captured in the PEX collection.



Total jobs: 746

Total threads: 3,236

Jobs in collection:

Job Name	User	Number	System Pool Number	Thread Identifier
QZRCSRVS	QUSER	045113	2	000000000000000
SERVER	QNOTES	045111	2	000000000000005
QZRCSRVS	QUSER	045077	2	000000000000019
JALDEVD	HENDERAN	037913	3	000000000000008
QZDASOINIT	QUSER	045108	2	000000000000005
QZDASOINIT	QUSER	045112	2	000000000000003
QDIRSRV	QDIRSRV	016138	2	000000000000002
STMSACTPRV	QNOTES	028540	7	000000000000000
SERVER	QNOTES	036079	2	000000000000007
SERVER	QNOTES	044528	2	000000000000007
SERVER	QNOTES	026111	2	000000000000006
IDOCCOL	BRAU	045106	2	000000000000000
QDBX033XR2	QSYS	038142	2	000000000000001
QDBX036XR2	QSYS	028893	2	000000000000009

[An example of a collection's Collection Jobs Page]

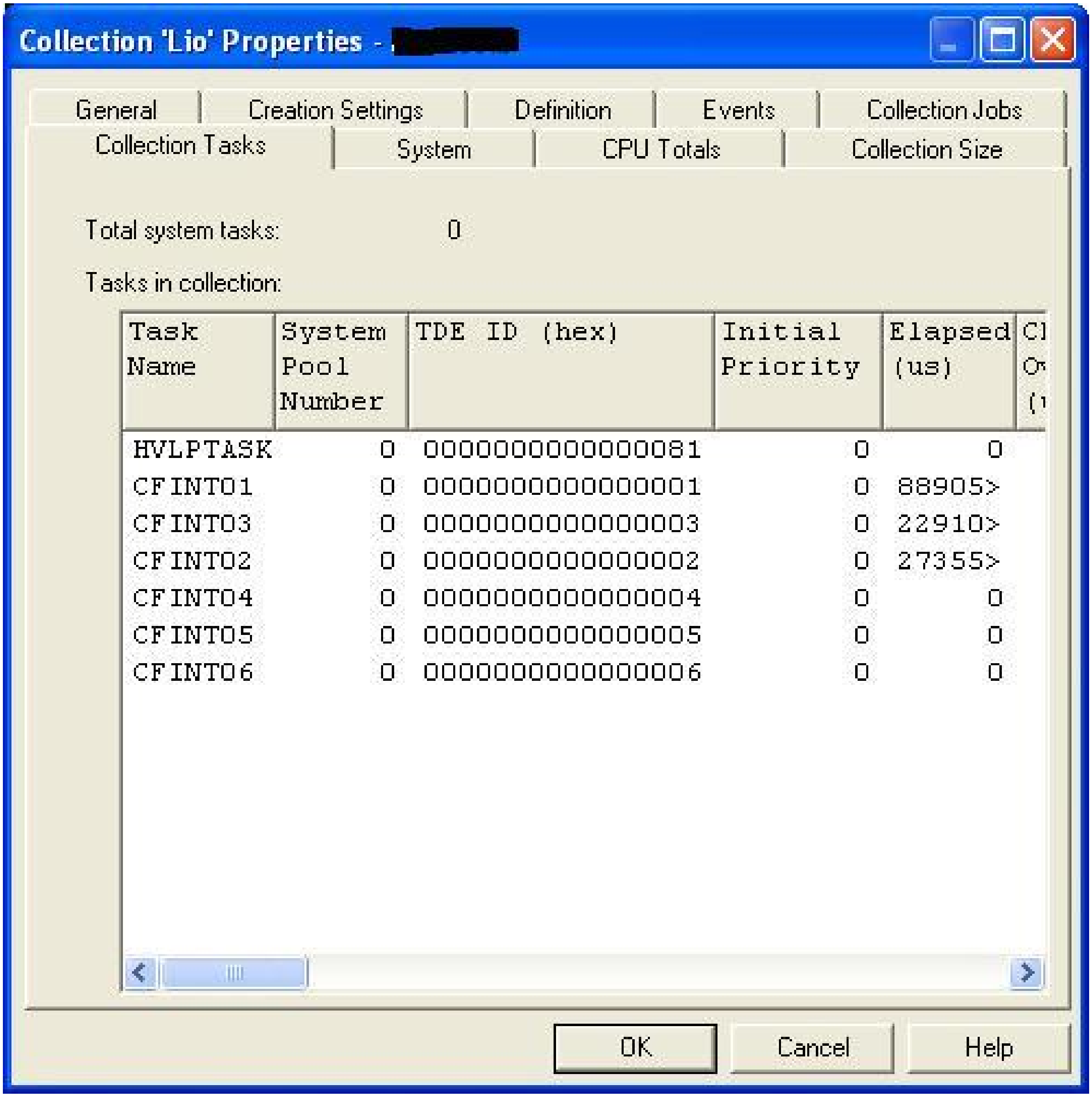
The following information is displayed on the Collection Jobs property page:

Field Name	**Field Description**
Total jobs	The total number of jobs in the collection.
Total threads	The total number of threads in the collection.
Jobs in collection list	A list of all jobs included in the collection. Due to the potential for huge amounts of data, a menu is available by right-clicking on the list providing additional features. The popup menu offers copy and paste to the clipboard, customizable font, and search capabilities.

Table of Contents Previous Next

4.3.7.6 Collection Tasks

The Collection Tasks property page lists every task that was captured in the PEX collection. This page also lists the total number of system tasks captured.



Total system tasks: 0

Tasks in collection:

Task Name	System Pool Number	TDE ID (hex)	Initial Priority	Elapsed (us)	Cl O (
HVLPTASK	0	0000000000000081	0	0	
CFINT01	0	0000000000000001	0	88905>	
CFINT03	0	0000000000000003	0	22910>	
CFINT02	0	0000000000000002	0	27355>	
CFINT04	0	0000000000000004	0	0	
CFINT05	0	0000000000000005	0	0	
CFINT06	0	0000000000000006	0	0	

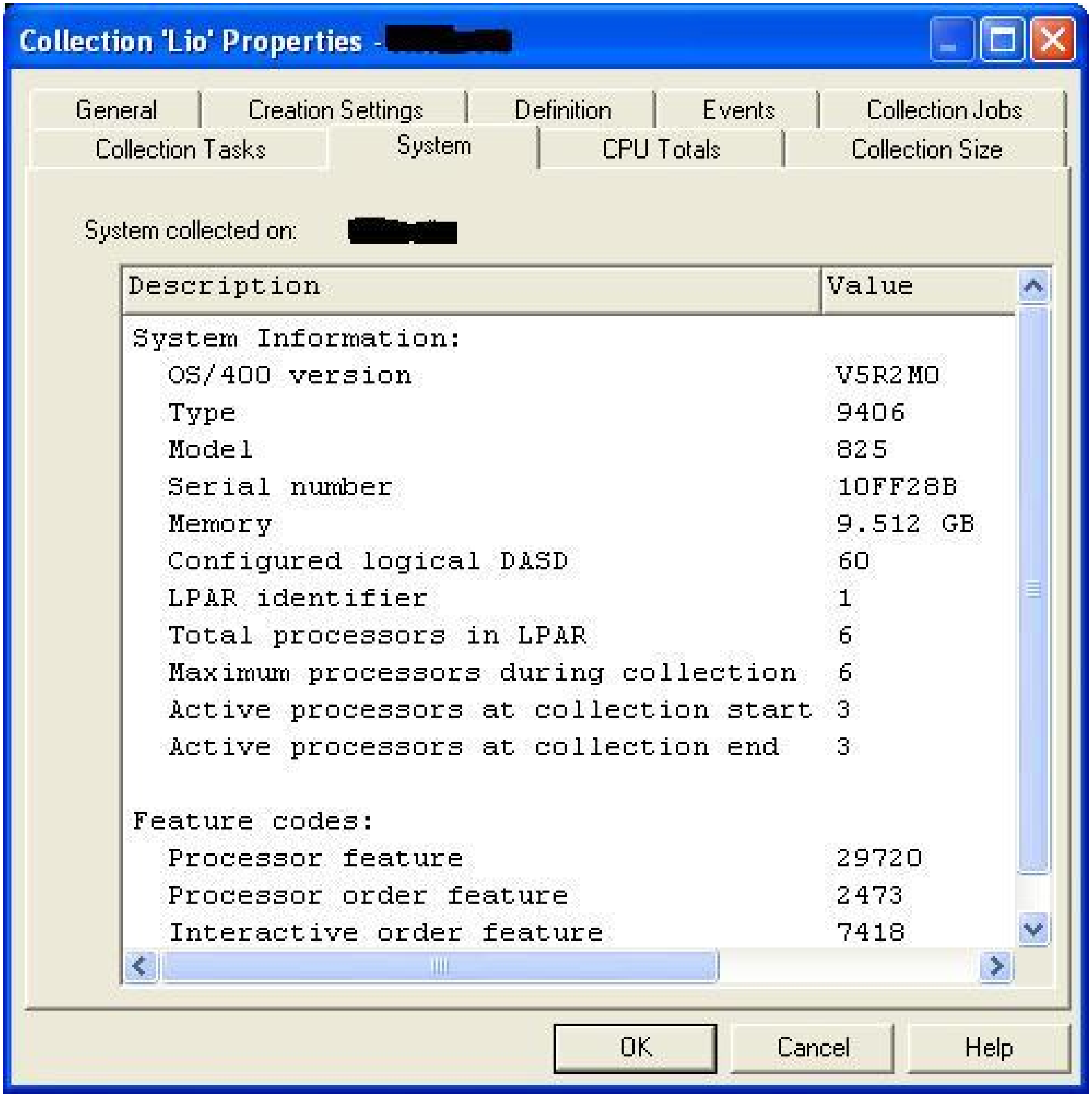
The following information is displayed on the Collection Tasks property page:

Field Name	Field Description
Total system tasks	The total number of system tasks included in the collection. Tasks named CFINT01...CFINTnn where nn equals the number of processors on the system are not included in this total.
Requested system tasks list	A list of all tasks included in the collection. You will always get tasks of name CFINT01...CFINTnn where nn equals number of processors on the system. Due to the potential for huge amounts of data, a menu is available by right-clicking on the list providing additional features. The popup menu offers copy and paste to the clipboard, customizable font, and search capabilities.

Table of Contents Previous Next

4.3.7.7 System Page

The System property page provides basic information about the system the collection was created on.



Collection 'Lio' Properties - [redacted]

System collected on: [redacted]

```
Description                                   Value
System Information:
   OS/400 version                             V5R2M0
   Type                                       9406
   Model                                      825
   Serial number                              10FF28B
   Memory                                     9.512 GB
   Configured logical DASD                    60
   LPAR identifier                            1
   Total processors in LPAR                   6
   Maximum processors during collection       6
   Active processors at collection start      3
   Active processors at collection end        3

Feature codes:
   Processor feature                          29720
   Processor order feature                    2473
   Interactive order feature                  7418
```

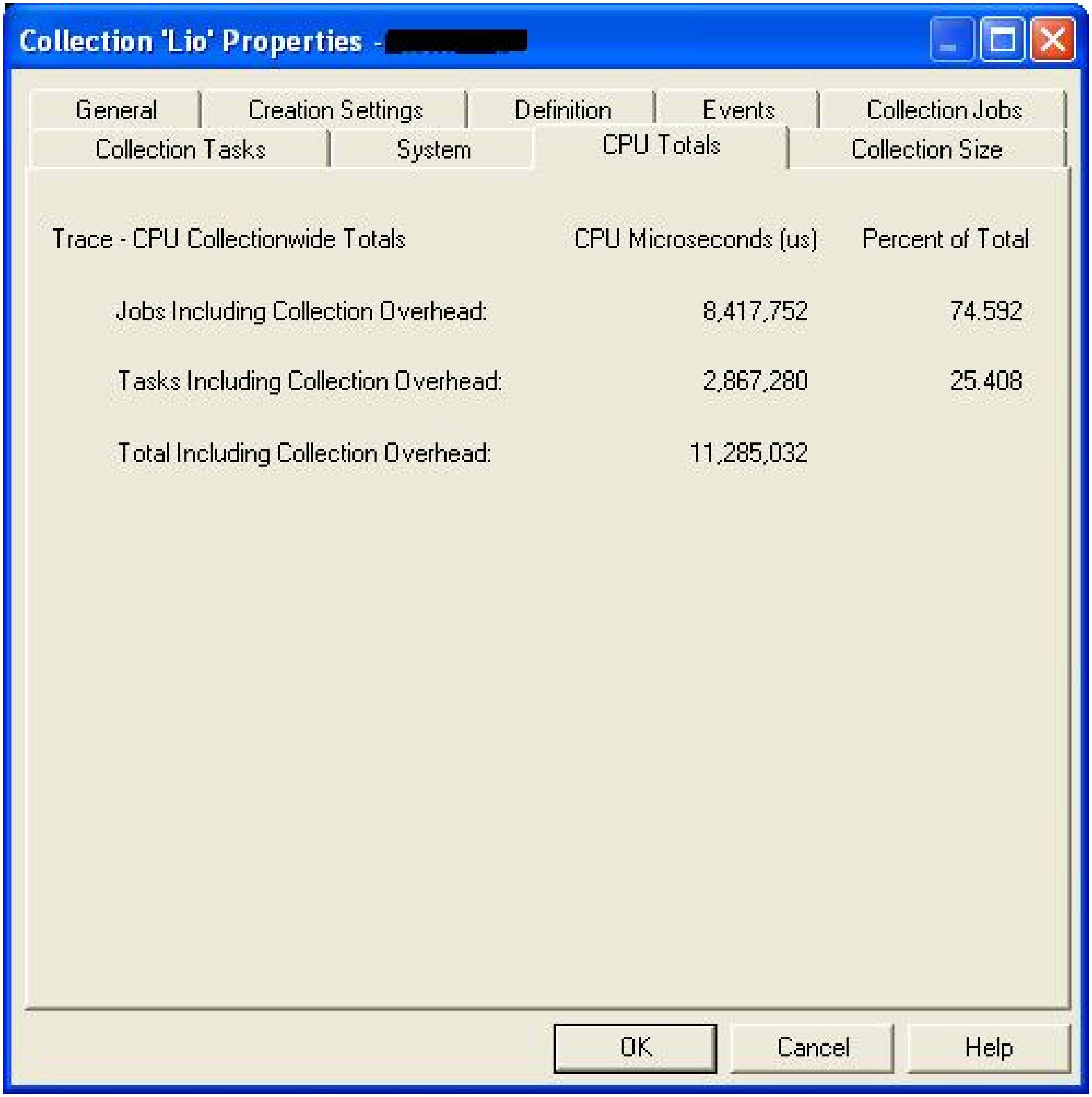
The following information is displayed on the System property page:

Field Name	Field Description
System collected on	iSeries system the data was collected on.
OS/400 Version	Version/Release/Mod Level of OS/400.
Type	iSeries system type
Model	iSeries system model
Serial Number	Unique system serial number identifier for the iSeries.
Memory	Total system memory.
Configured logical DASD	Total number of operational DASD.
Total processors in LPAR	Total number of CPU processors.
LPAR Identifier	Logical Partition identifier. A value of 0 indicates that this is not an LPAR system.

Table of Contents | Previous | Next

4.3.7.8 Trace CPU Totals

The Trace CPU Totals property page provides information about summarized CPU for a Trace collection.



Collection 'Lio' Properties - [redacted]

Trace - CPU Collectionwide Totals	CPU Microseconds (us)	Percent of Total
Jobs Including Collection Overhead:	8,417,752	74.592
Tasks Including Collection Overhead:	2,867,280	25.408
Total Including Collection Overhead:	11,285,032	

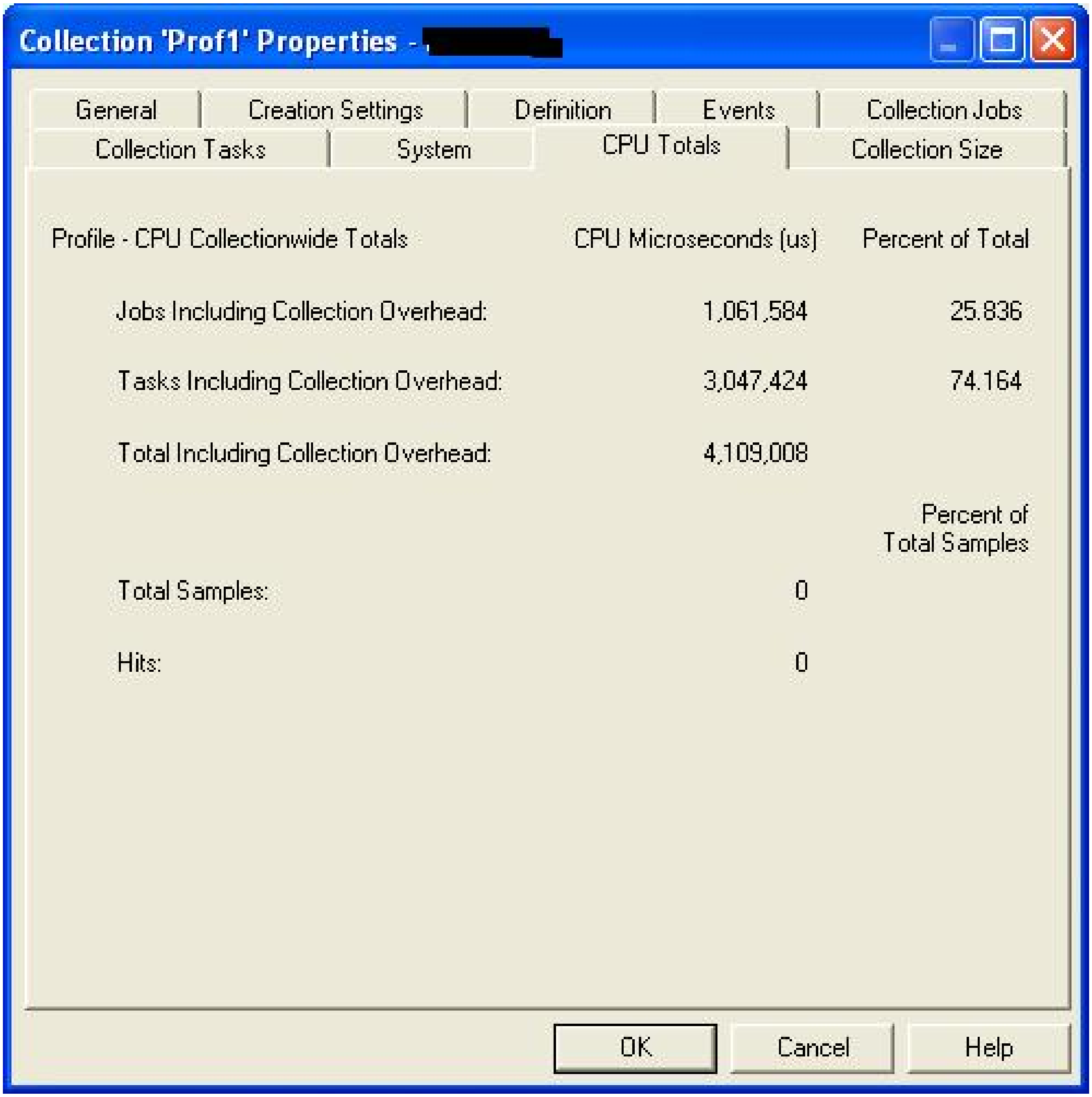
The following information is displayed on the Trace CPU Totals property page:

Field Name	Field Description
Jobs including collection overhead (us)	Total number of CPU microseconds including pex collection overhead for all jobs in the collection.
Jobs including collection overhead (% of total)	Percentage of Total CPU used by jobs in the collection.
Tasks including collection overhead (us)	Total number of CPU microseconds including pex collection overhead for all tasks in the collection.
Tasks including collection overhead (% of total)	Percentage of Total CPU used by tasks in the collection.
Total including collection overhead	The total CPU microseconds for the collection including collection overhead.

Table of Contents Previous Next

4.3.7.9 Profile CPU Totals

The Profile CPU Totals property page provides information about summarized CPU for a Profile collection.



Collection 'Prof1' Properties - [redacted]

Profile - CPU Collectionwide Totals	CPU Microseconds (us)	Percent of Total
Jobs Including Collection Overhead:	1,061,584	25.836
Tasks Including Collection Overhead:	3,047,424	74.164
Total Including Collection Overhead:	4,109,008	
		Percent of Total Samples
Total Samples:	0	
Hits:	0	

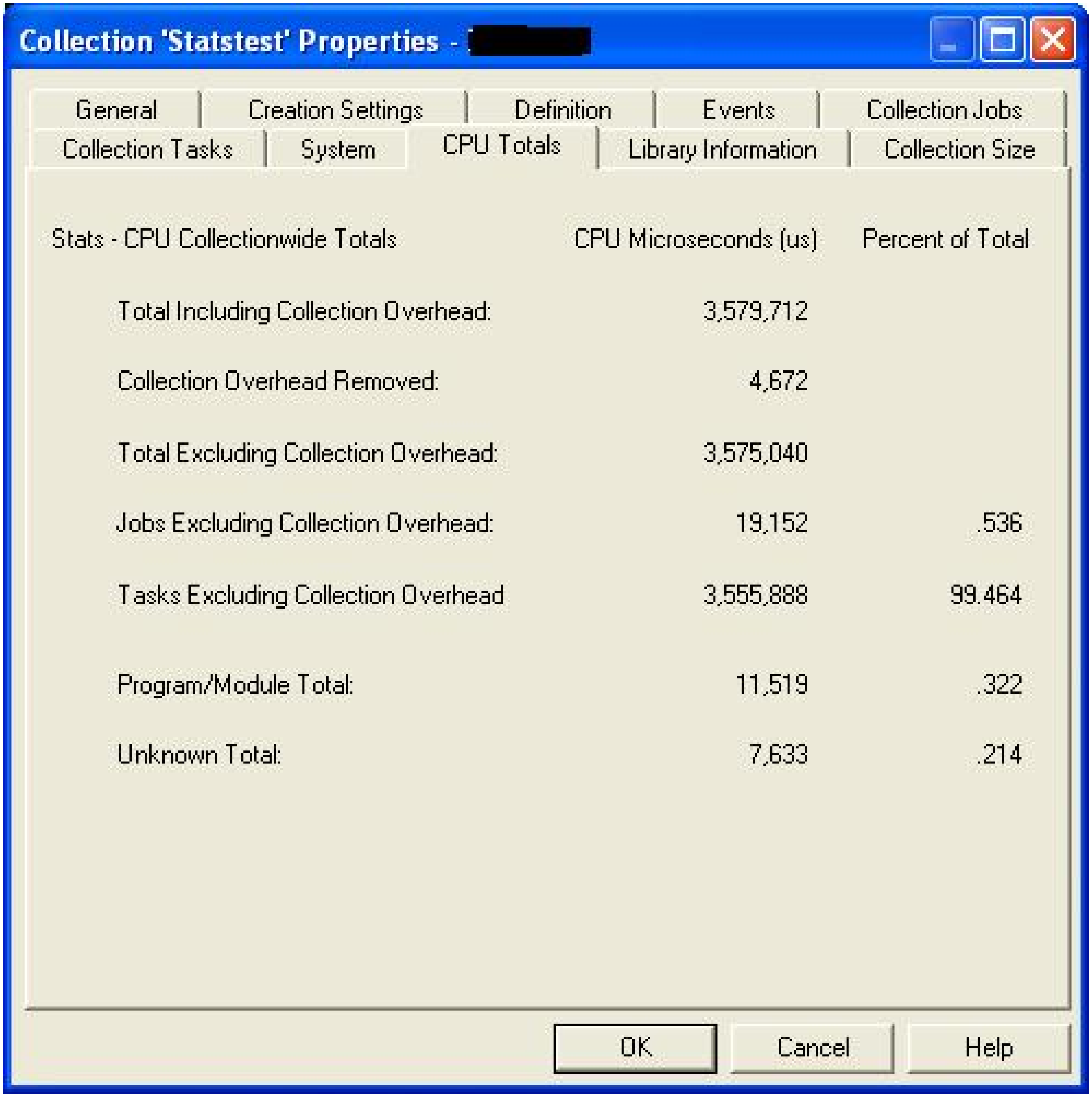
The following information is displayed on the Profile CPU Totals property page:

Field Name	**Field Description**
Jobs including collection overhead (us)	Total number of CPU microseconds including pex collection overhead for all jobs in the collection.
Jobs including collection overhead (% of total)	Percentage of Total CPU used by jobs in the collection.
Tasks including collection overhead (us)	Total number of CPU microseconds including pex collection overhead for all tasks in the collection.
Tasks including collection overhead (% of total)	Percentage of Total CPU used by tasks in the collection.
Total including collection overhead	The total CPU microseconds for the collection including collection overhead.
Total samples	Total sample count.
Hits	Hit count from the total samples.
Hits (% of total samples)	Percentage of hit CPU microseconds used over the total samples in the collection.

Table of Contents Previous Next

4.3.7.10 Statistical CPU Totals

The Stats CPU Totals property page provides information about summarized CPU for a Statistical collection.



Collection 'Statstest' Properties - [redacted]

Stats - CPU Collectionwide Totals	CPU Microseconds (us)	Percent of Total
Total Including Collection Overhead:	3,579,712	
Collection Overhead Removed:	4,672	
Total Excluding Collection Overhead:	3,575,040	
Jobs Excluding Collection Overhead:	19,152	.536
Tasks Excluding Collection Overhead	3,555,888	99.464
Program/Module Total:	11,519	.322
Unknown Total:	7,633	.214

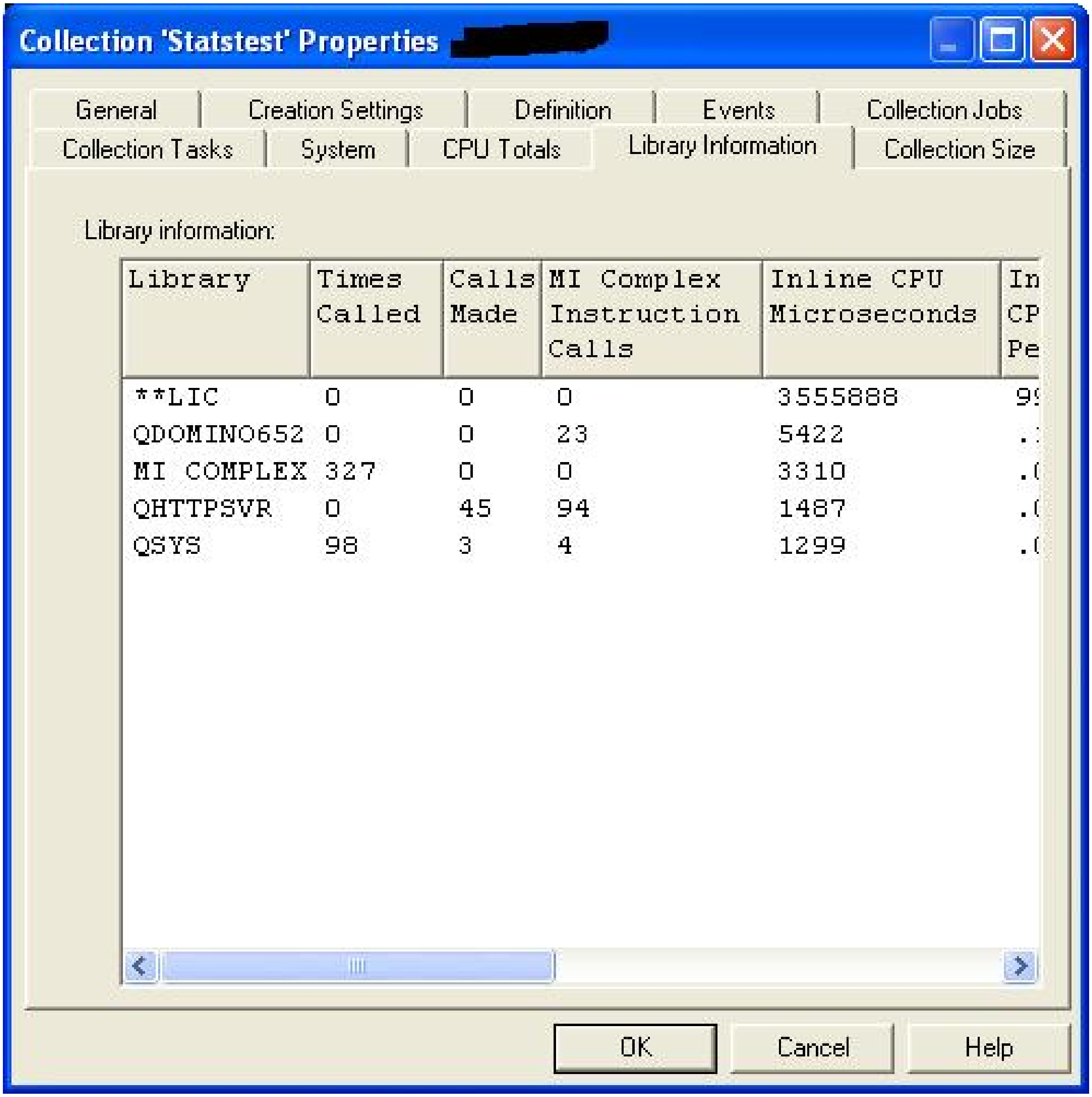
The following information is displayed on the Stats CPU Totals property page:

Field Name	**Field Description**
Total including collection overhead (us)	The total CPU microseconds for the collection including collection overhead.
Collection overhead removed (us)	The total removed collection overhead in CPU microseconds.
Total excluding collection overhead (us)	The total CPU microseconds for the collection excluding collection overhead.
Jobs excluding collection overhead (us)	Total number of CPU microseconds excluding PEX collection overhead for all jobs in the collection.
Jobs excluding collection overhead (% of total)	Percentage of Total CPU excluding collection overhead used by jobs in the collection.
Tasks excluding collection overhead (us)	Total number of CPU microseconds excluding PEX collection overhead for all tasks in the collection.
Tasks excluding collection overhead (% of total)	Percentage of Total CPU excluding collection overhead used by tasks in the collection.
Program/Module (us)	Total number of CPU microseconds used by programs and modules.
Program/Module (% of total)	Percentage of CPU microseconds devoted to program/module activity of the total CPU.
Unknown (us)	Total number of CPU microseconds which cannot be attributed to specific programs or modules.
Unknown (% of total)	Percentage of CPU microseconds devoted to unknown activity of the total CPU.

Table of Contents | Previous | Next

4.3.7.11 Library Information

The Library Information property page provides information about the libraries in the PEX collection. This page is only shown for collections of type Statistical.



Library information:

Library	Times Called	Calls Made	MI Complex Instruction Calls	Inline CPU Microseconds	In CP Pe
**LIC	0	0	0	3555888	9
QDOMINO652	0	0	23	5422	.
MI COMPLEX	327	0	0	3310	.
QHTTPSVR	0	45	94	1487	.
QSYS	98	3	4	1299	.

The following information is displayed on the Library Information property page:

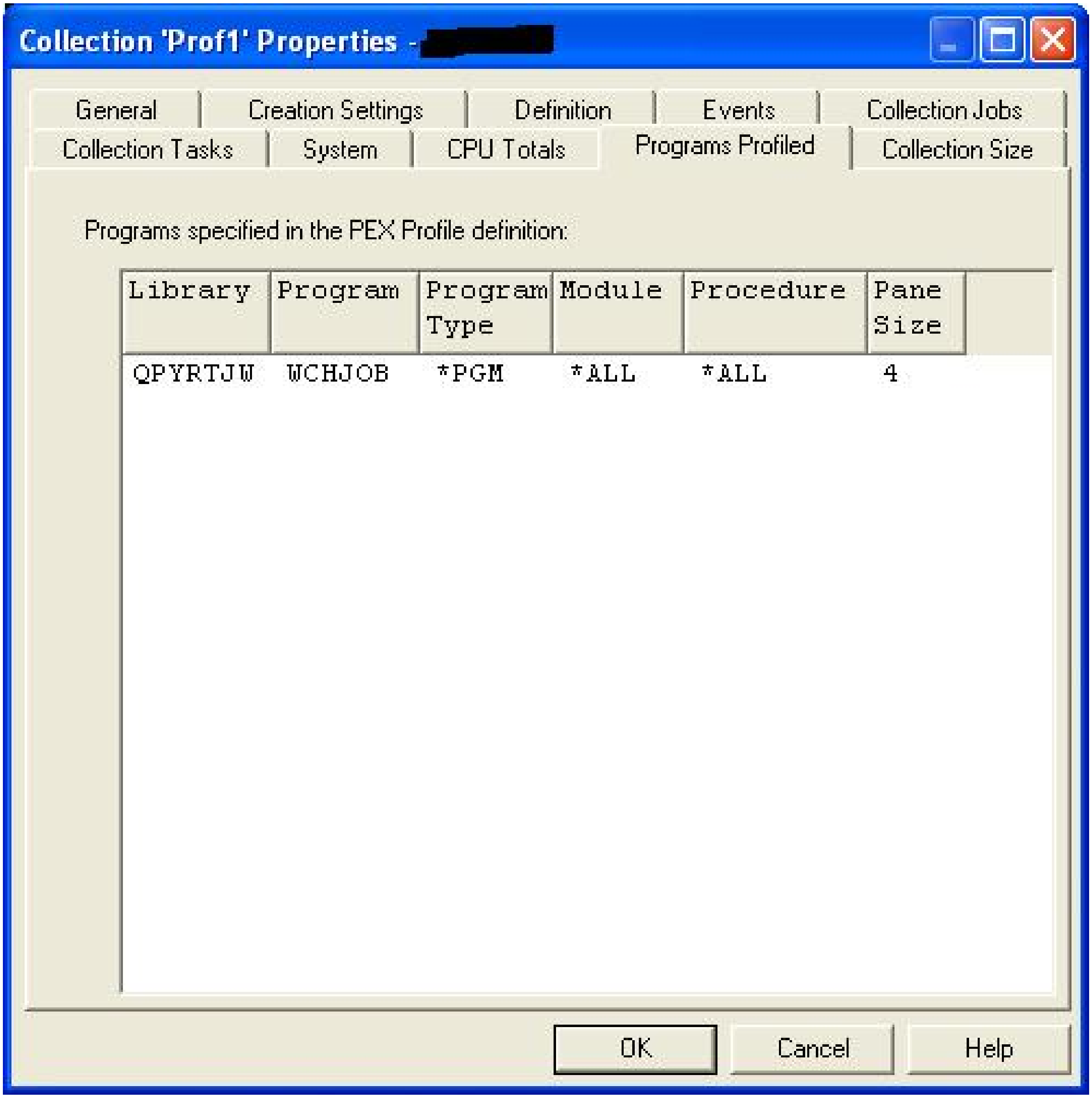
Field Name	Field Description
Library Information List	List of libraries with additional statistics as they pertain to each library in the list.

Table of Contents | Previous | Next

4.3.7.12 Programs Profiled

The Programs Profiled property page provides information about the programs included in the PEX definition for a PEX Profile collection.

This panel is only shown at release V5R1. The same information is provided on the Definition panel at release V5R2.



Programs specified in the PEX Profile definition:

Library	Program	Program Type	Module	Procedure	Pane Size
QPYRTJW	WCHJOB	*PGM	*ALL	*ALL	4

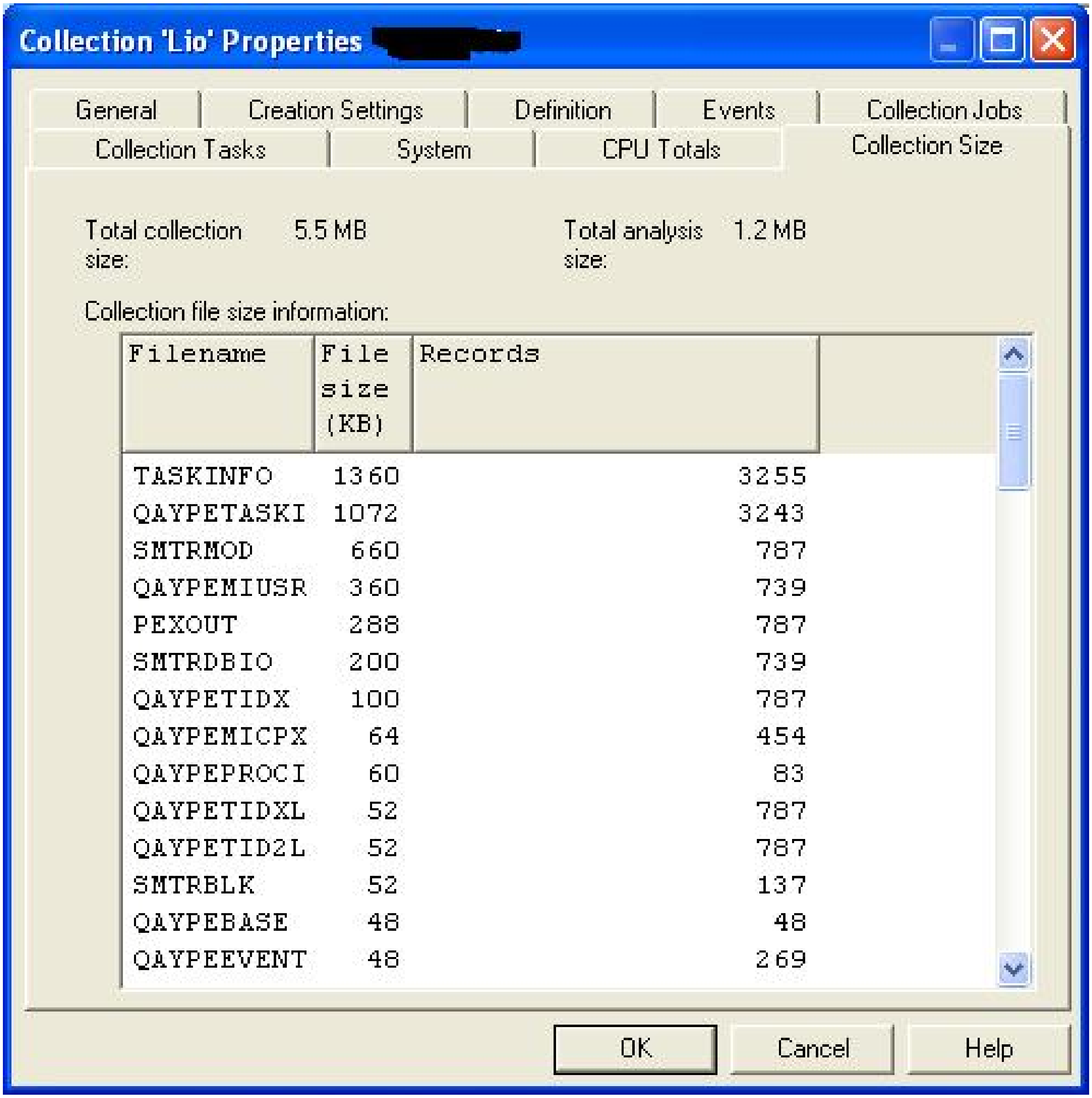
Name	Description
Programs Profiled List	List of programs and/or service programs from the PEX definition used for this collection.

Table of Contents | Previous | Next

4.3.7.13 Collection Size

This panel displays information about the disk space consumed by the current PEX collection. The file size and total records of each PEX Analyzer file associated with this collection is displayed in a list. Totals for the collection and all analyses for this collection are also displayed above the list.

This panel is only available at releases V5R2 and higher.



Total collection size: 5.5 MB

Total analysis size: 1.2 MB

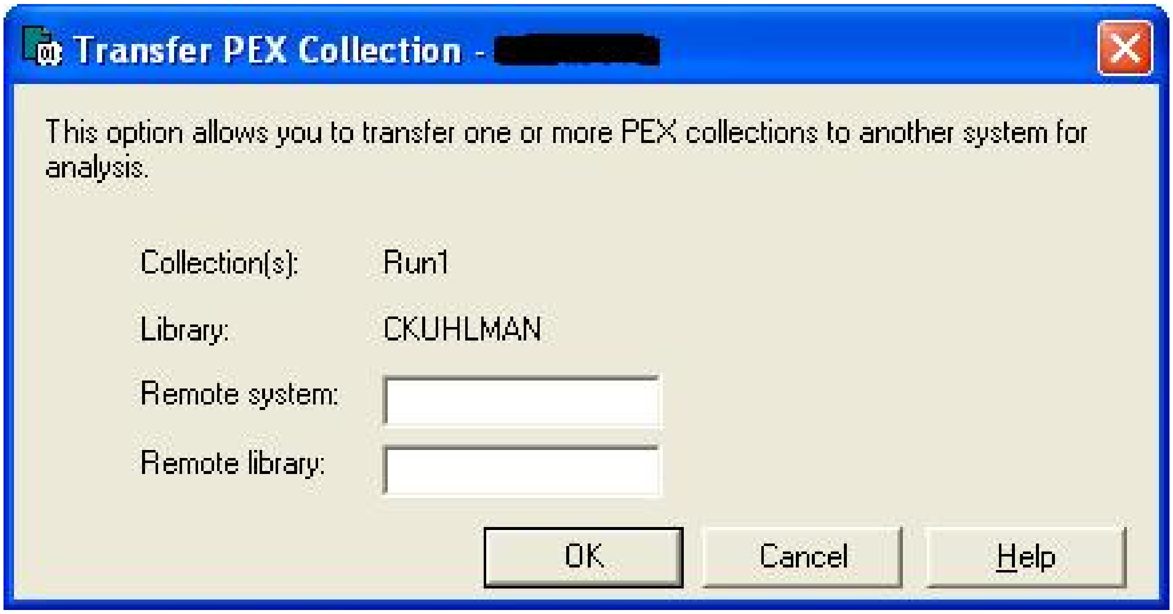
Collection file size information:

Filename	File size (KB)	Records
TASKINFO	1360	3255
QAYPETASKI	1072	3243
SMTRMOD	660	787
QAYPEMIUSR	360	739
PEXOUT	288	787
SMTRDBIO	200	739
QAYPETIDX	100	787
QAYPEMICPX	64	454
QAYPEPROCI	60	83
QAYPETIDXL	52	787
QAYPETID2L	52	787
SMTRBLK	52	137
QAYPEBASE	48	48
QAYPEEVENT	48	269

4.3.8 Transferring to another system

A user may move one or more PEX collections to another system using PEX Analyzer. Access this feature by selecting one or more collections, right-clicking and choosing the 'Transfer to...' menu.

Moving PEX collections can be very time saving because it eliminates several manual steps that would normally be required. Transferring a collection is accomplished using a series of remote command to save the PEX data to a save file, FTP it to the remote system and the restore the PEX data to the remote library specified.



Transfer PEX Collection - [redacted]

This option allows you to transfer one or more PEX collections to another system for analysis.

Collection(s): Run1

Library: CKUHLMAN

Remote system:

Remote library:

The options available on this page are summarized in the following table:

Field	Description
Collection(s)	Lists the collection(s) that are to be transferred from the current system.
Library	The library on the current system the collection(s) will be transferred from.
Remote system	The remote system name or IP address to send the collection(s) to.
Remote library	The library on the remote system to send the collection(s) to.

Table of Contents | Previous | Next

4.4 PEX Definitions

A **PEX definition** is a member in a specific system database (QUSRSYS/QAPEXDFN), and it controls most of the aspects of making a PEX collection except the following:

- When to begin making the collection
- When to end the collection
- The library where collection data is stored

A PEX definition also controls:

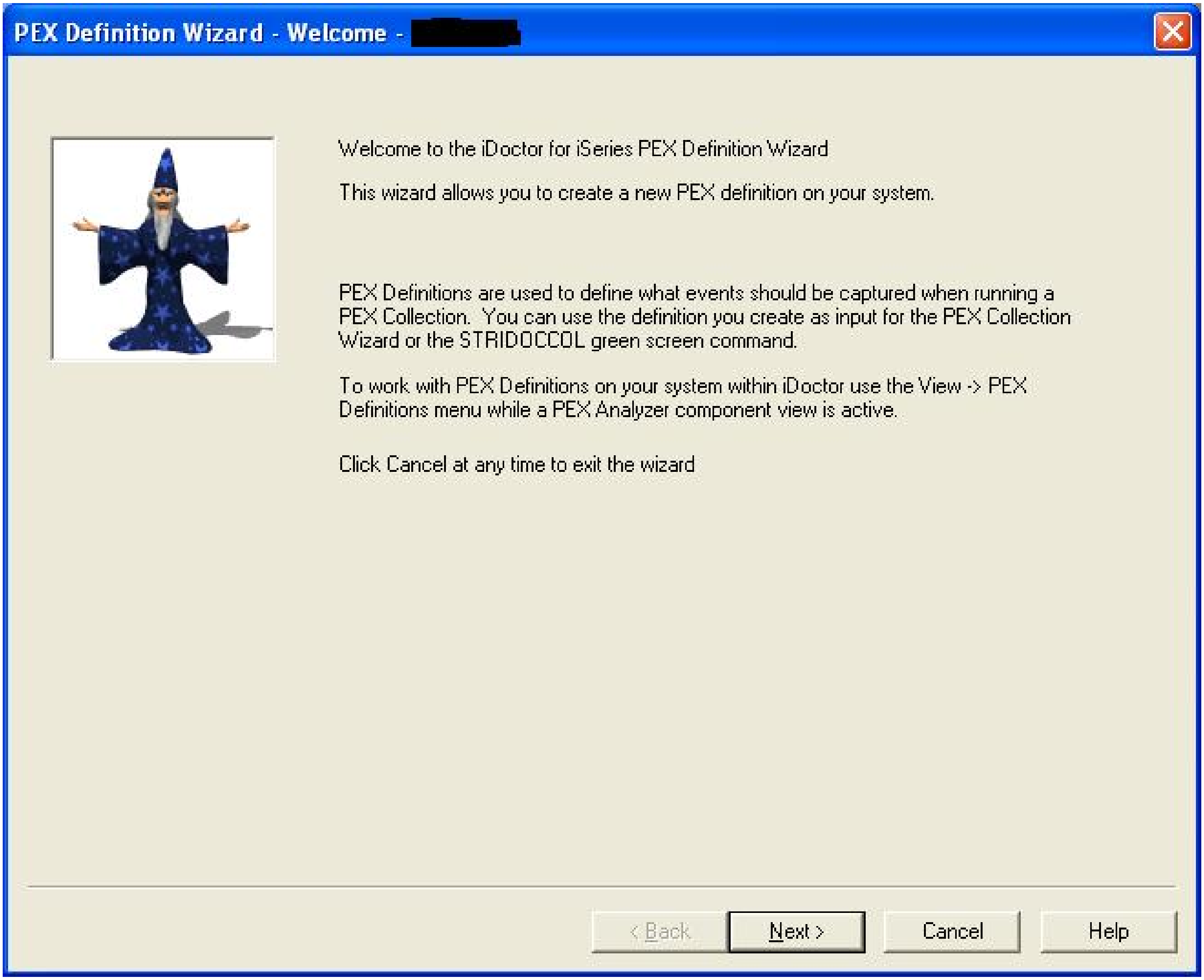
- Which subset of the hundreds of event types are to be activated
- Granularity of CPU sampling
- Maximum amount of data to be collected
- Subset of jobs or system tasks (or all jobs and all tasks) that are to be traced

To create or change a PEX definition, use the PEX Definition Wizard discussed within this section.

4.4.1 Creating or Changing - The PEX Definition Wizard

Within the iDoctor for iSeries client you can create new PEX definitions or easily change an existing PEX definition using the PEX Definition Wizard. You can access this wizard via the PEX Definitions view.

The PEX Definition Wizard is a full-featured interface over the ADDPEXDFN command.



PEX Definition Wizard - Welcome - [redacted]

Welcome to the iDoctor for iSeries PEX Definition Wizard

This wizard allows you to create a new PEX definition on your system.

PEX Definitions are used to define what events should be captured when running a PEX Collection. You can use the definition you create as input for the PEX Collection Wizard or the STRIDOCCOL green screen command.

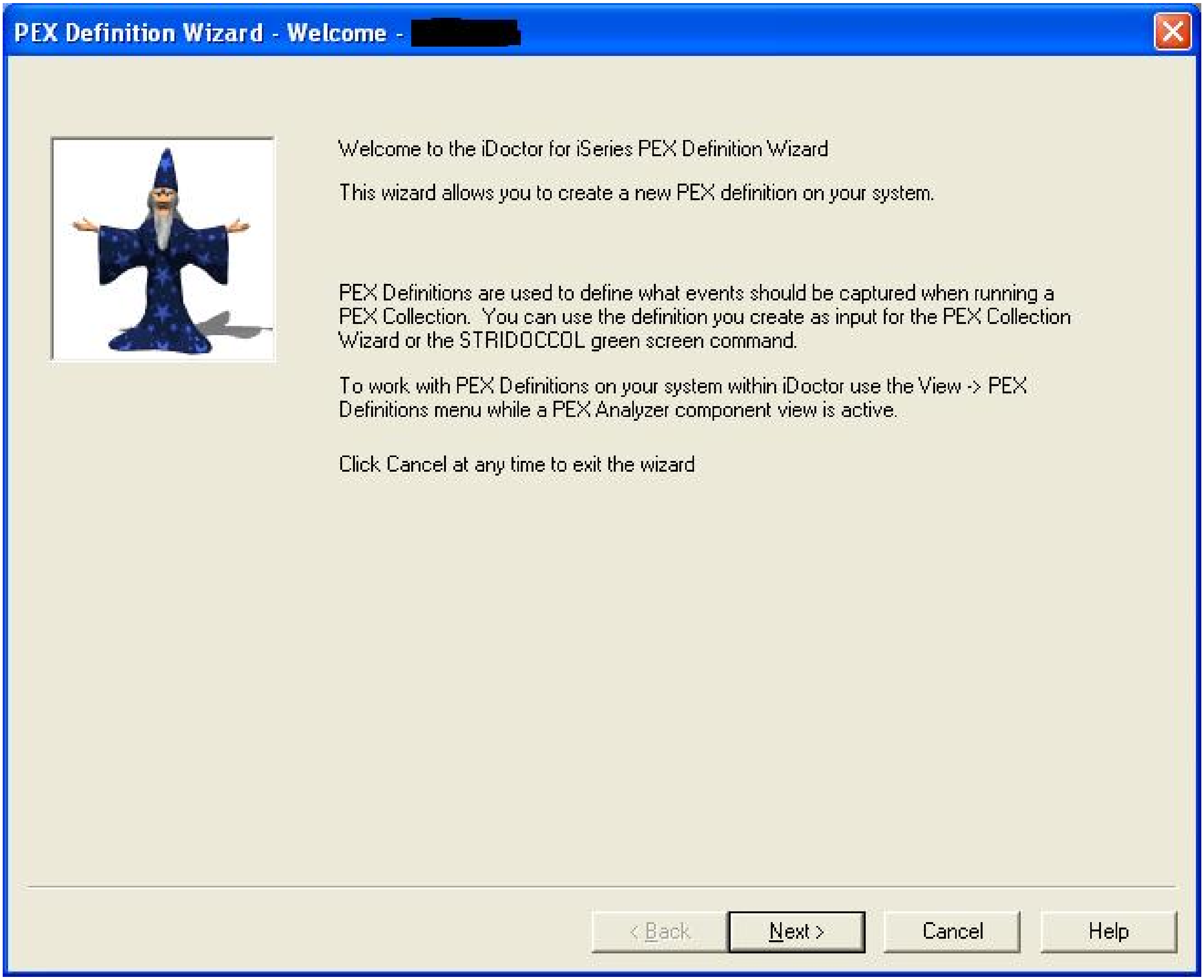
To work with PEX Definitions on your system within iDoctor use the View -> PEX Definitions menu while a PEX Analyzer component view is active.

Click Cancel at any time to exit the wizard

Table of Contents | Previous | Next

4.4.1.1 Welcome

The Welcome page in the PEX Definition Wizard introduces the user to the wizard and offers information about what the wizard will do. When changing an existing PEX definition all of the current values will be prefilled into the appropriate places within the PEX Definition interface.



PEX Definition Wizard - Welcome - [redacted]

Welcome to the iDoctor for iSeries PEX Definition Wizard

This wizard allows you to create a new PEX definition on your system.

PEX Definitions are used to define what events should be captured when running a PEX Collection. You can use the definition you create as input for the PEX Collection Wizard or the STRIDOCCOL green screen command.

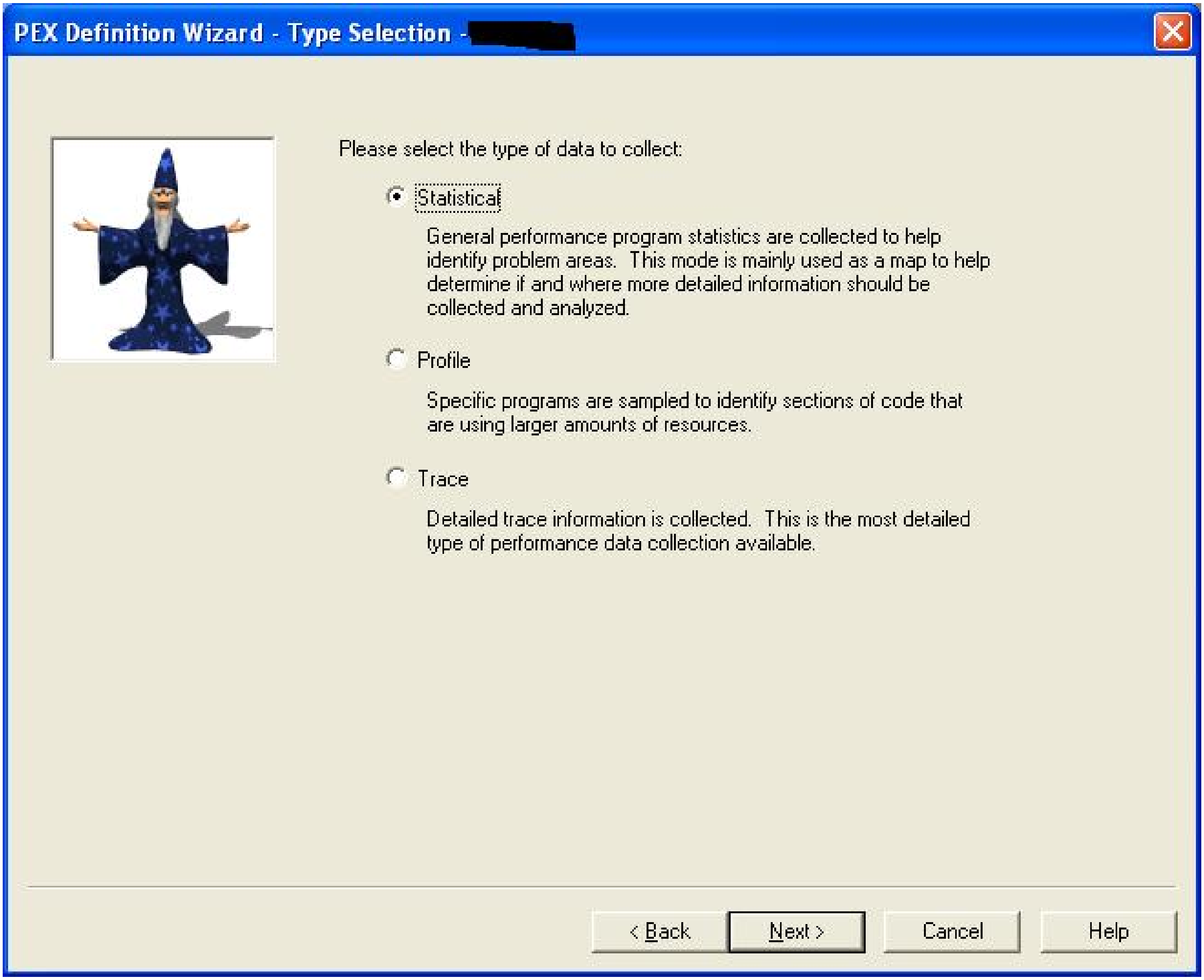
To work with PEX Definitions on your system within iDoctor use the View -> PEX Definitions menu while a PEX Analyzer component view is active.

Click Cancel at any time to exit the wizard

Table of Contents | Previous | Next

4.4.1.2 Type Selection

The Type Selection page in the PEX Definition Wizard lets a user decide the most important characteristic about the definition: its type. The value choosen (Statistical, Profile, or Trace), influences what's seen on the following pages.



PEX Definition Wizard - Type Selection - [redacted]

Please select the type of data to collect:

● Statistical

General performance program statistics are collected to help identify problem areas. This mode is mainly used as a map to help determine if and where more detailed information should be collected and analyzed.

○ Profile

Specific programs are sampled to identify sections of code that are using larger amounts of resources.

○ Trace

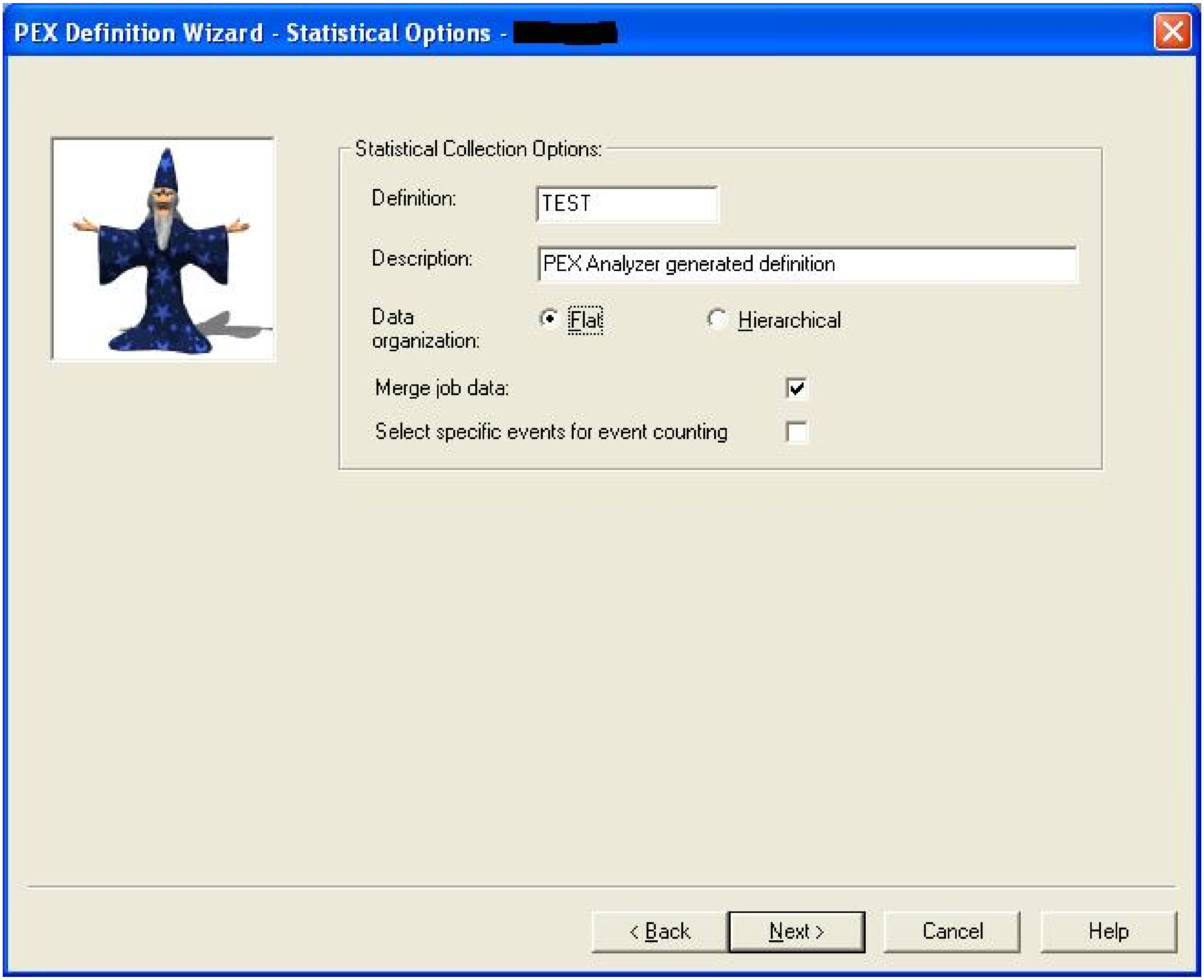
Detailed trace information is collected. This is the most detailed type of performance data collection available.

Table of Contents | Previous | Next

4.4.1.3 Options

The Options page in the PEX Definition Wizard lets the user decide the most basic parameters for the definition. This page is different depending on the type selection parameter specified on the previous page.

An example of the Statistical Options Page:



PEX Definition Wizard - Statistical Options - [redacted]

Statistical Collection Options:

Definition: TEST

Description: PEX Analyzer generated definition

Data organization: ● Flat ○ Hierarchical

Merge job data: ☑

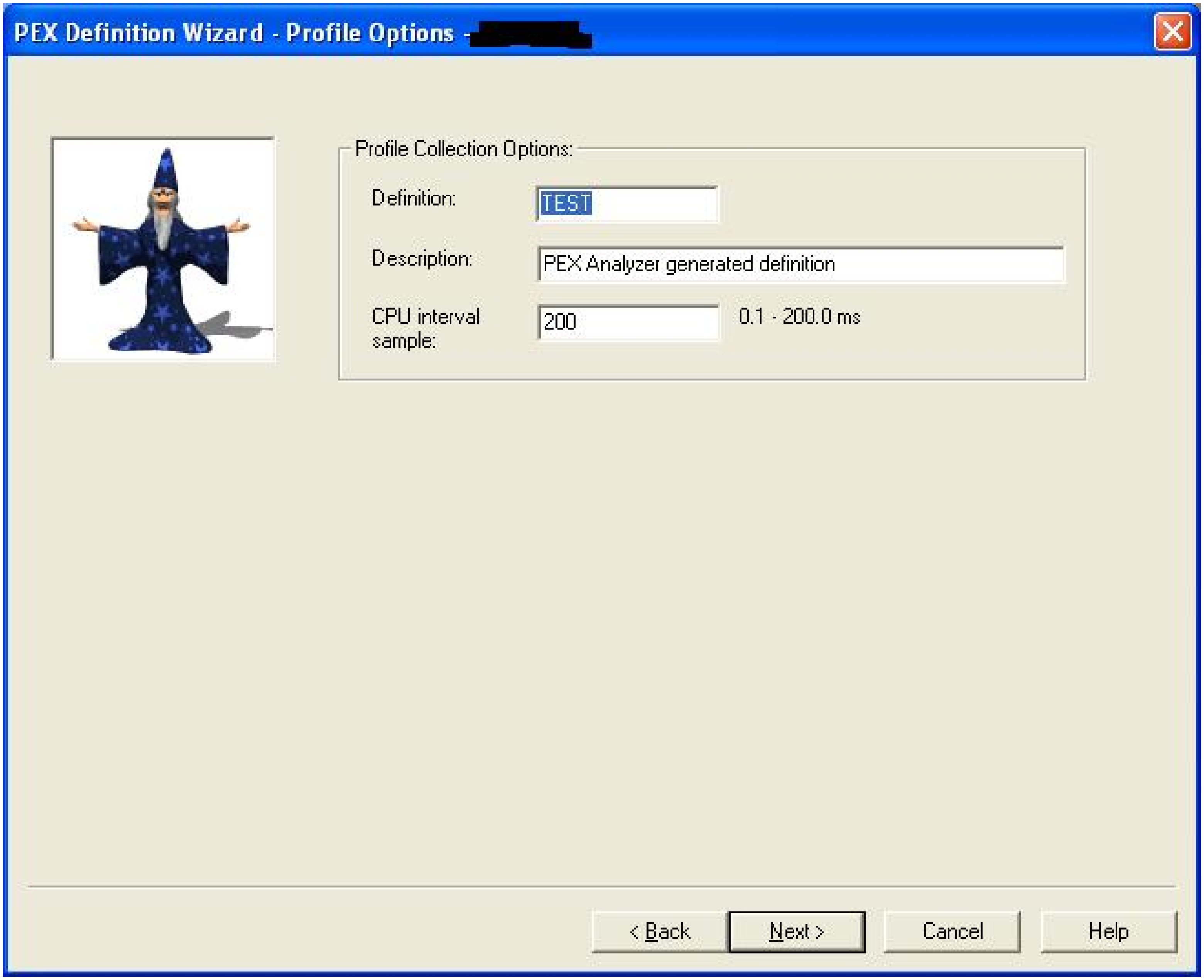
Select specific events for event counting ☐

This table defines the options available on this page:

Field	Description

Definition	Name of the PEX definition. This value matches the member name of the definition stored in the QUSRSYS/QAPEXDFN file.
Description	Specifies the text that briefly describes the PEX definition. The possible values are: *BLANK Text is not specified. 'description' Specify no more than 50 characters of text**.**
Data Organization	Specifies how the data is organized. The possible values are: Flat The report will not present data with a parent-child call flow relationship. Hierarchical The report will present data showing a parent-child call flow relationship.
Merge Job Data	Specifies if data from different jobs should be merged together or not. This parameter is only available if the Data Organization parameter value is Flat.
Select specific events for counting	At release V5R2 only, statistical definitions allow you to define event counters which tell you within the Statistical data the total number of events that occurred within each counter bucket. This checkbox will enable this option.

An example of the Profile Options Page:


PEX Definition Wizard - Profile Options - [redacted]
Profile Collection Options:
< Back
Next >
Cancel
Help

Definition: TEST

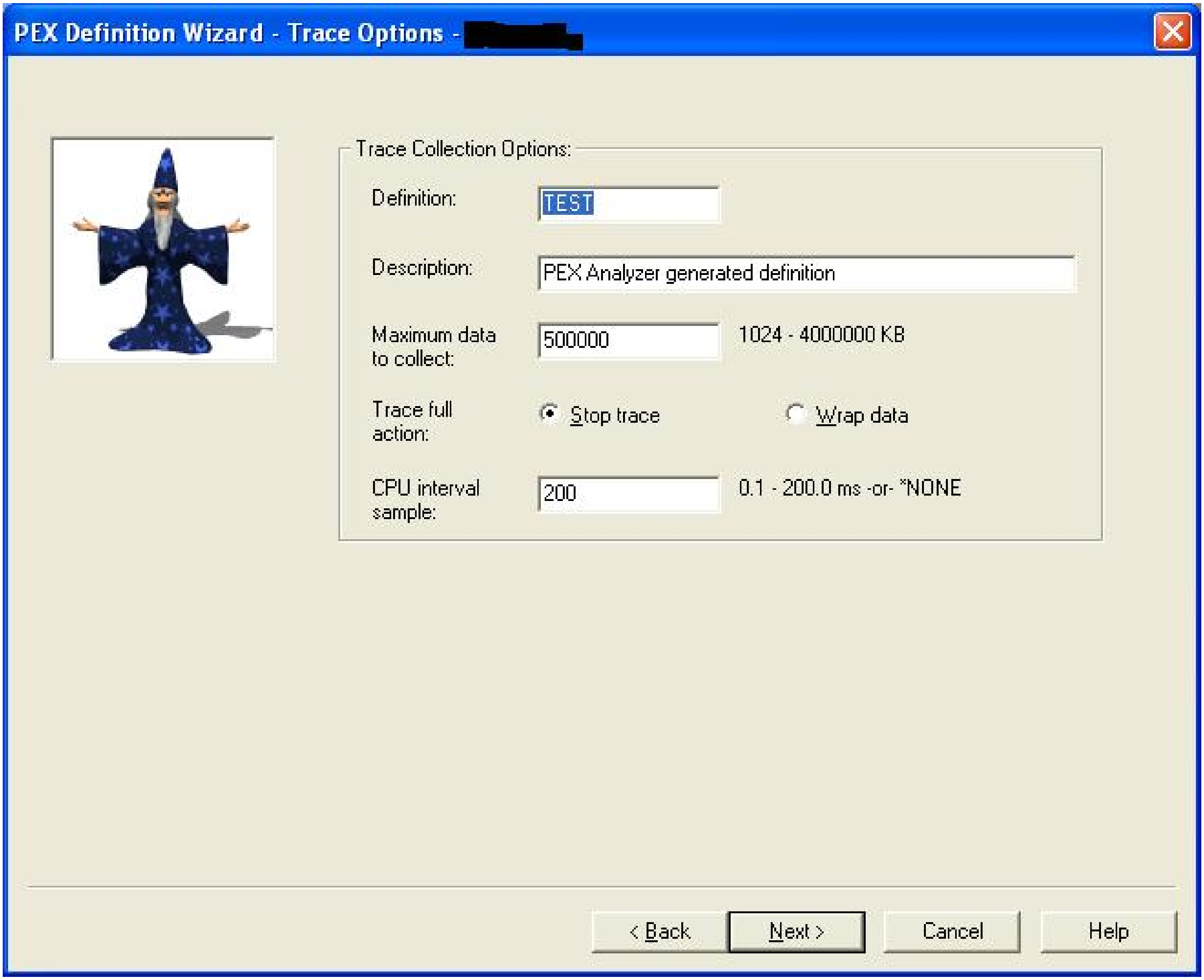
Description: PEX Analyzer generated definition

CPU interval sample: 200 0.1 - 200.0 ms

This table defines the options available on this page:

Field	**Description**
Definition	Name of the PEX definition. This value matches the member name of the definition stored in the QUSRSYS/QAPEXDFN file.
Description	Specifies the text that briefly describes the PEX definition. The possible values are: *BLANK Text is not specified. 'description' Specify no more than 50 characters of text**.**
CPU Interval Sample	Specifies the size of the interval which CPU samples are taken of the program. A low interval will cause a high number of samples to be taken, and will also cause higher overhead. A low interval will also provide relatively more data. This parameter will be not available if it does not apply to the selected iDoctor-supplied definition.

An example of the Trace Options Page is shown below:



PEX Definition Wizard - Trace Options - [redacted]

Trace Collection Options:

Definition: TEST

Description: PEX Analyzer generated definition

Maximum data to collect: 500000 1024 - 4000000 KB

Trace full action: ● Stop trace ○ Wrap data

CPU interval sample: 200 0.1 - 200.0 ms -or- *NONE

This table defines the options available on this page:

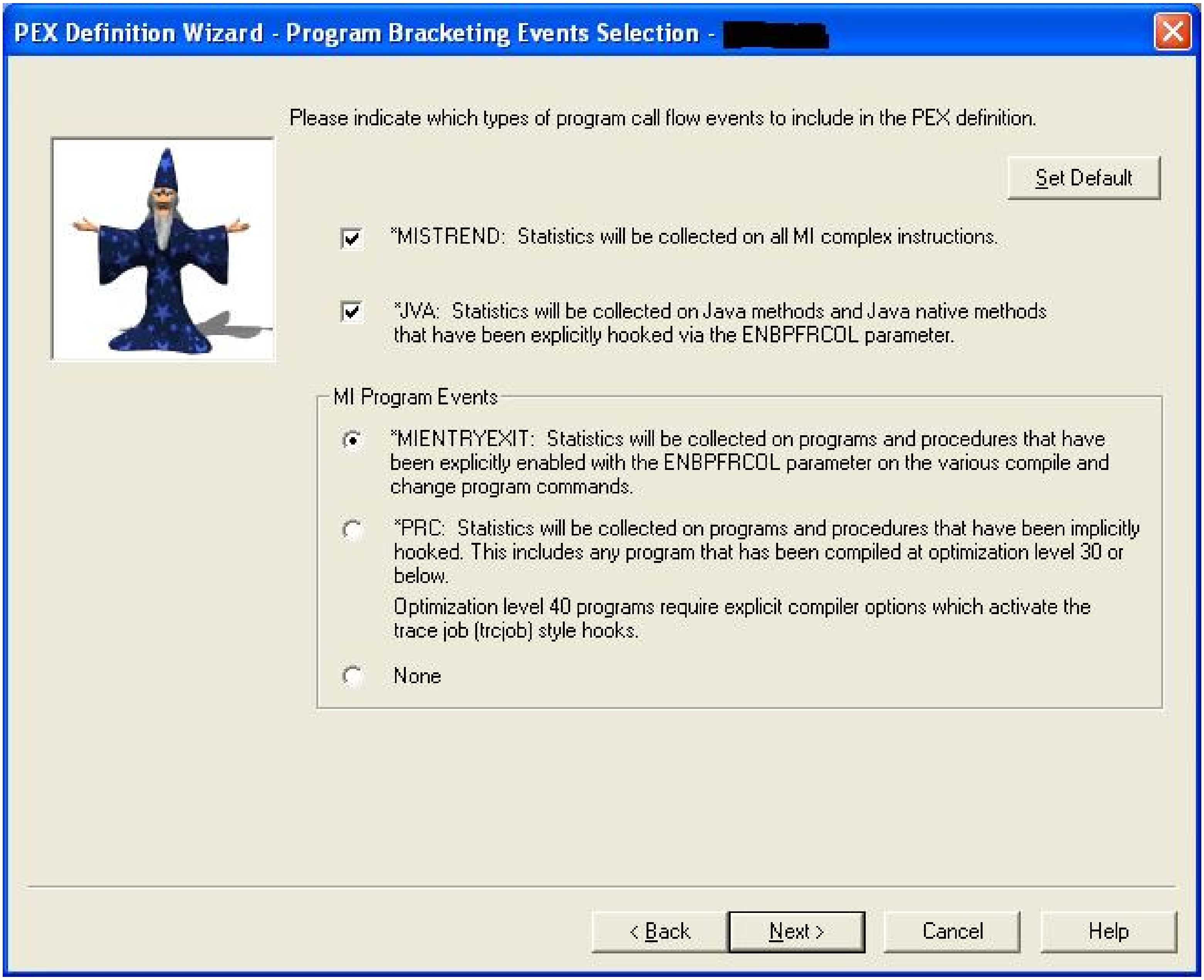
Field	Description
Definition	Name of the PEX definition. This value matches the member name of the definition stored in the QUSRSYS/QAPEXDFN file.
Description	Specifies the text that briefly describes the PEX definition. The possible values are: *BLANK Text is not specified. 'description' Specify no more than 50 characters of text**.**

Data Organization	Specifies how the data is organized. The possible values are: Flat The performance explorer tool will not collect data for a parent-child relationship. Hierarchical The performance explorer tool will collect data for a parent-child relationship.
Maximum data to collect	The maximum amount of disk space this collection should use in kilobytes. The default value is 500,000. When using a user-defined PEX definition this parameter is ignored because it is provided within the PEX definition.
CPU interval sample	Specifies the size of the interval which CPU samples are taken of the program. A low interval will cause a high number of samples to be taken, and will also cause higher overhead. A low interval will also provide relatively more data. This parameter will be grayed out if it does not apply to the selected iDoctor-supplied definition.
Trace full action	The action to take if the maximum data to collect value is reached. The choices are to either suspend/stop the collection or wrap the data. If the data is wrapped the oldest trace records will be overwritten with the newest ones.

Table of Contents | Previous | Next

4.4.1.4 Program Bracketing Events Selection

This page lets you decide which program call flow events are included in the type Statistical definition.



PEX Definition Wizard - Program Bracketing Events Selection - [redacted]

Please indicate which types of program call flow events to include in the PEX definition.

☑ *MISTREND: Statistics will be collected on all MI complex instructions.

☑ *JVA: Statistics will be collected on Java methods and Java native methods that have been explicitly hooked via the ENBPFRCOL parameter.

MI Program Events

● *MIENTRYEXIT: Statistics will be collected on programs and procedures that have been explicitly enabled with the ENBPFRCOL parameter on the various compile and change program commands.

○ *PRC: Statistics will be collected on programs and procedures that have been implicitly hooked. This includes any program that has been compiled at optimization level 30 or below.
Optimization level 40 programs require explicit compiler options which activate the trace job (trcjob) style hooks.

○ None

The following table summarizes the possible program bracketing event groups:

Event Group	Description
*MISTREND	Statistics are to be collected on all machine instructions.

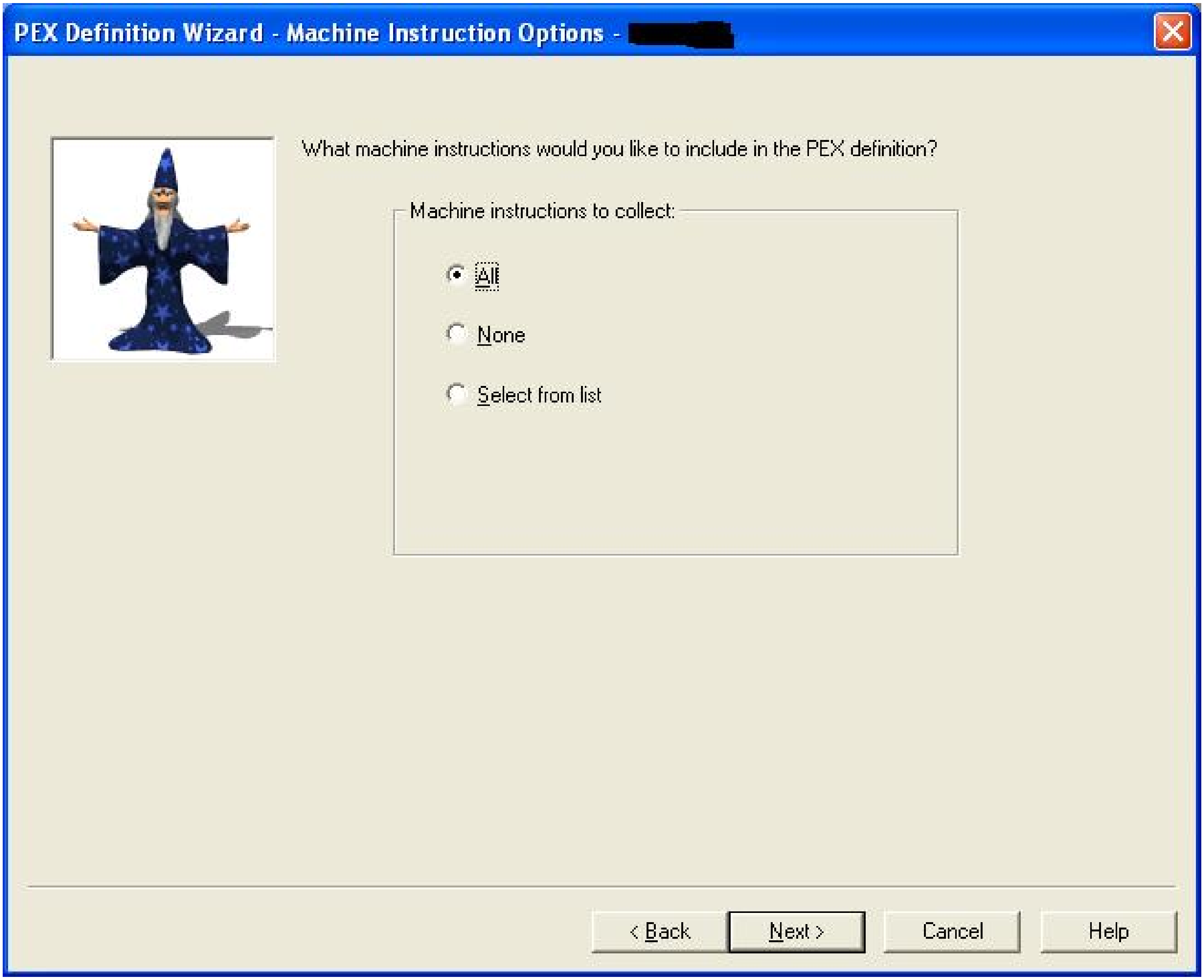
*JVA	Statistics are to be collected on Java methods and Java native methods that have been explicitly hooked via the ENBPFRCOL parameter (or its equivalents) on the Java and JIT compile commands.
*MIENTRYEXIT	Statistics are to be collected on programs and procedures that have been explicitly hooked via the ENBPFRCOL parameter on the various compile and change job commands.
*PRC	Statistics are to be collected on programs and procedures that have been implicitly hooked. This includes any program that has been compiled at optimization level 30 or below. Optimization level 40 programs require explicit compiler options which activate the trace job (trcjob) style hooks. *MIENTRYEXIT and *PRC are mutually exclusive.

Table of Contents | Previous | Next

4.4.1.5 Machine Instruction Option

This page is used for Trace based PEX Definitions and is used to select whether or MI instructions should be included in the PEX definition.

Using the **Select from list** option will display a window to select specific MI instructions from the list of all available MI instructions for the current version and release of OS/400.



PEX Definition Wizard - Machine Instruction Options - [redacted]

What machine instructions would you like to include in the PEX definition?

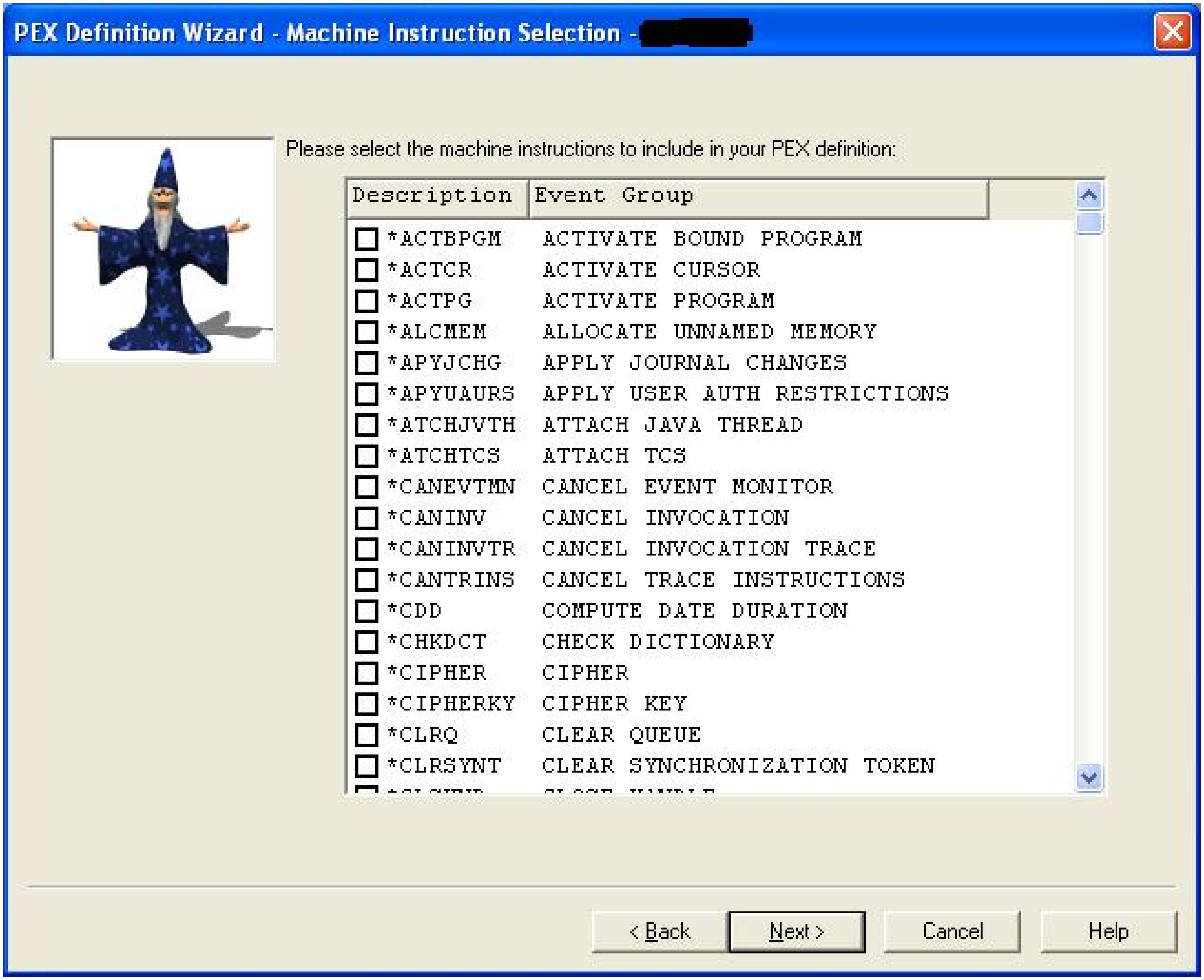
Machine instructions to collect:

● All

○ None

○ Select from list

The machine instruction selection page is shown below:



PEX Definition Wizard - Machine Instruction Selection - [redacted]

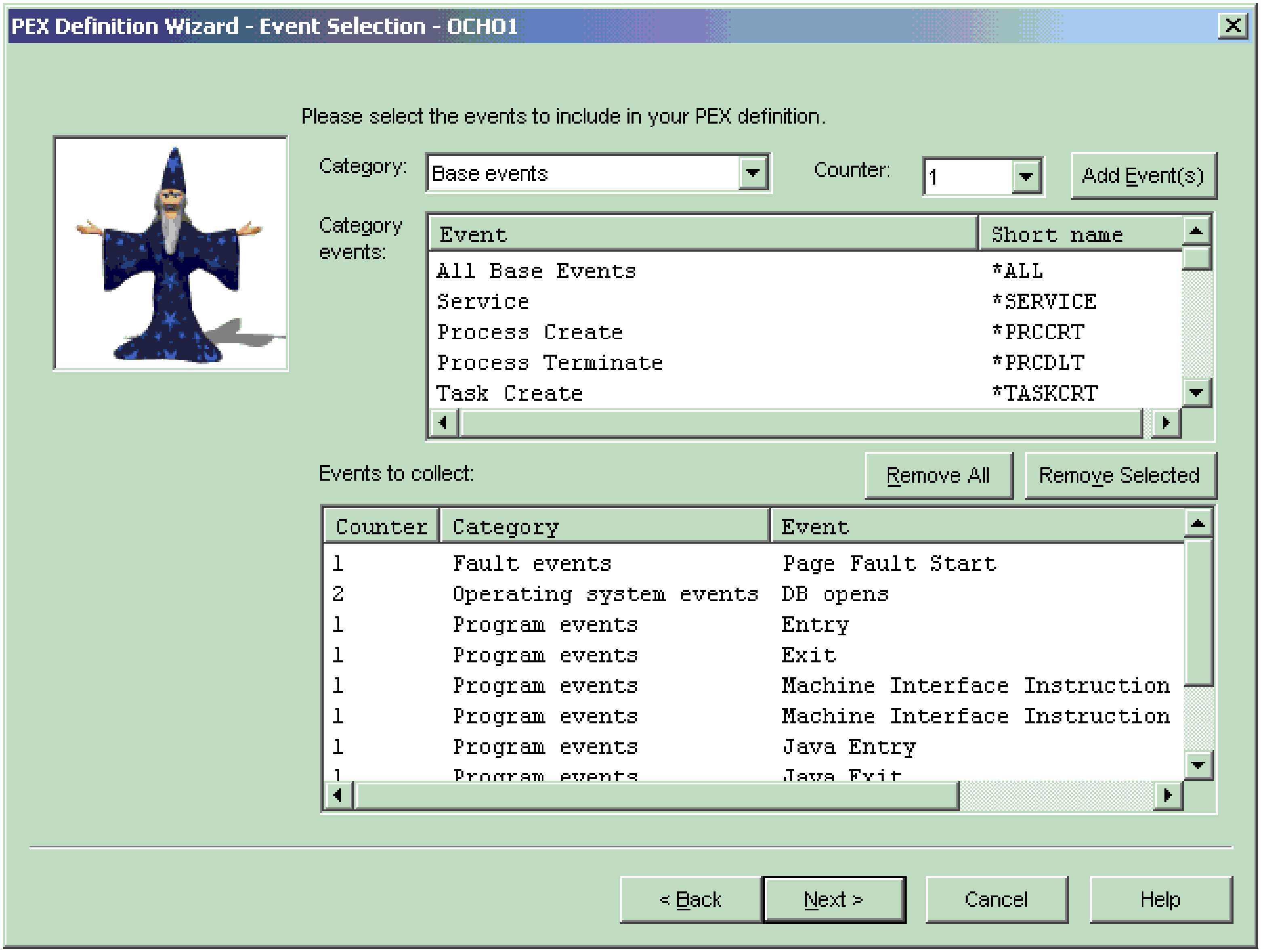
Please select the machine instructions to include in your PEX definition:

Description	Event Group
□ *ACTBPGM	ACTIVATE BOUND PROGRAM
□ *ACTCR	ACTIVATE CURSOR
□ *ACTPG	ACTIVATE PROGRAM
□ *ALCMEM	ALLOCATE UNNAMED MEMORY
□ *APYJCHG	APPLY JOURNAL CHANGES
□ *APYUAURS	APPLY USER AUTH RESTRICTIONS
□ *ATCHJVTH	ATTACH JAVA THREAD
□ *ATCHTCS	ATTACH TCS
□ *CANEVTMN	CANCEL EVENT MONITOR
□ *CANINV	CANCEL INVOCATION
□ *CANINVTR	CANCEL INVOCATION TRACE
□ *CANTRINS	CANCEL TRACE INSTRUCTIONS
□ *CDD	COMPUTE DATE DURATION
□ *CHKDCT	CHECK DICTIONARY
□ *CIPHER	CIPHER
□ *CIPHERKY	CIPHER KEY
□ *CLRQ	CLEAR QUEUE
□ *CLRSYNT	CLEAR SYNCHRONIZATION TOKEN

Table of Contents | Previous | Next

4.4.1.6 Event Selection

This page lets you decide which specifc events to include in the definition. This window is available when defining a Statistical or Trace definition. When defining a Statistical definition there is a counter field which lets you assign events to specific counter buckets. The purpose of the counter bucket is to provide the total occurrences of all the events specified in each bucket.



PEX Definition Wizard - Event Selection - OCH01

Please select the events to include in your PEX definition.

Category: Base events

Counter: 1

Category events:

Event	Short name
All Base Events	*ALL
Service	*SERVICE
Process Create	*PRCCRT
Process Terminate	*PRCDLT
Task Create	*TASKCRT

Events to collect:

Counter	Category	Event
1	Fault events	Page Fault Start
2	Operating system events	DB opens
1	Program events	Entry
1	Program events	Exit
1	Program events	Machine Interface Instruction
1	Program events	Machine Interface Instruction
1	Program events	Java Entry
1	Program events	Java Exit

The following table summarizes the inputs available on this window:

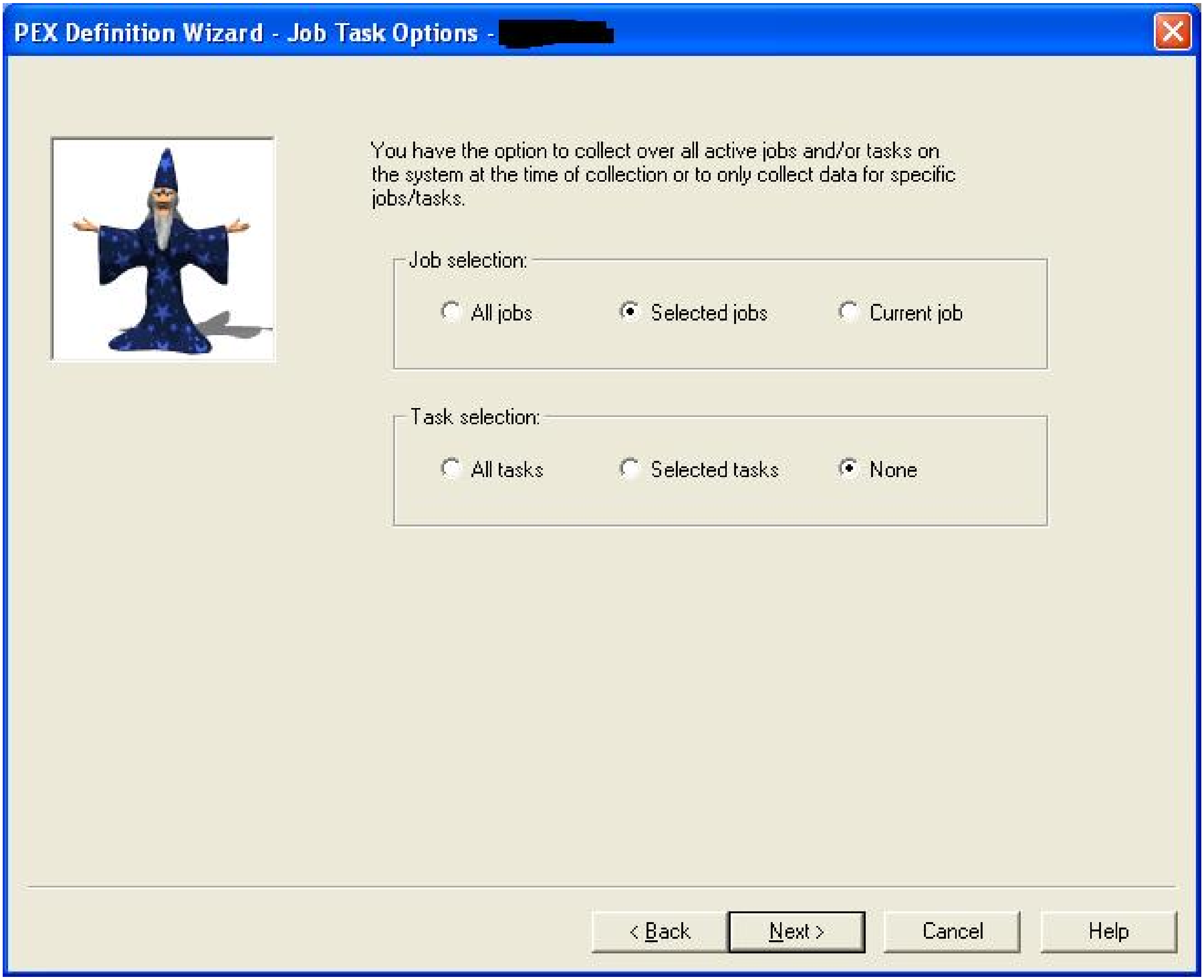
Field	Description
Category	This drop down lists contains the possible categories of events. Change the value and the list of events will be update to show the events within the selected category.

Counter	This drop down lists contains counter buckets to select from. The selected counter is used when adding the selected category events to the Events to collect list. This field is only visible when defining a statistical collection.
Category Events list	Displays the list of events to select from within the selected category.
Add Events button	Adds the selected events from the category events list to the events to collect list.
Events to collect list	Displays the current events to collect for this PEX definition. This list shows each specific category and event name to be collected.
Remove All button	Clears the Events to collect list.
Remove Selected button	Removes the selected events from the Events to collect list.

Table of Contents | Previous | Next

4.4.1.7 Job/Task Options

On this page you may decide if you would like to select specific jobs or tasks to include in the PEX definition. Selecting specific jobs and tasks is optional, but is necessary when you only want to collect data for the job(s) or task(s) you are interested in.



PEX Definition Wizard - Job Task Options - [redacted]

You have the option to collect over all active jobs and/or tasks on the system at the time of collection or to only collect data for specific jobs/tasks.

Job selection:

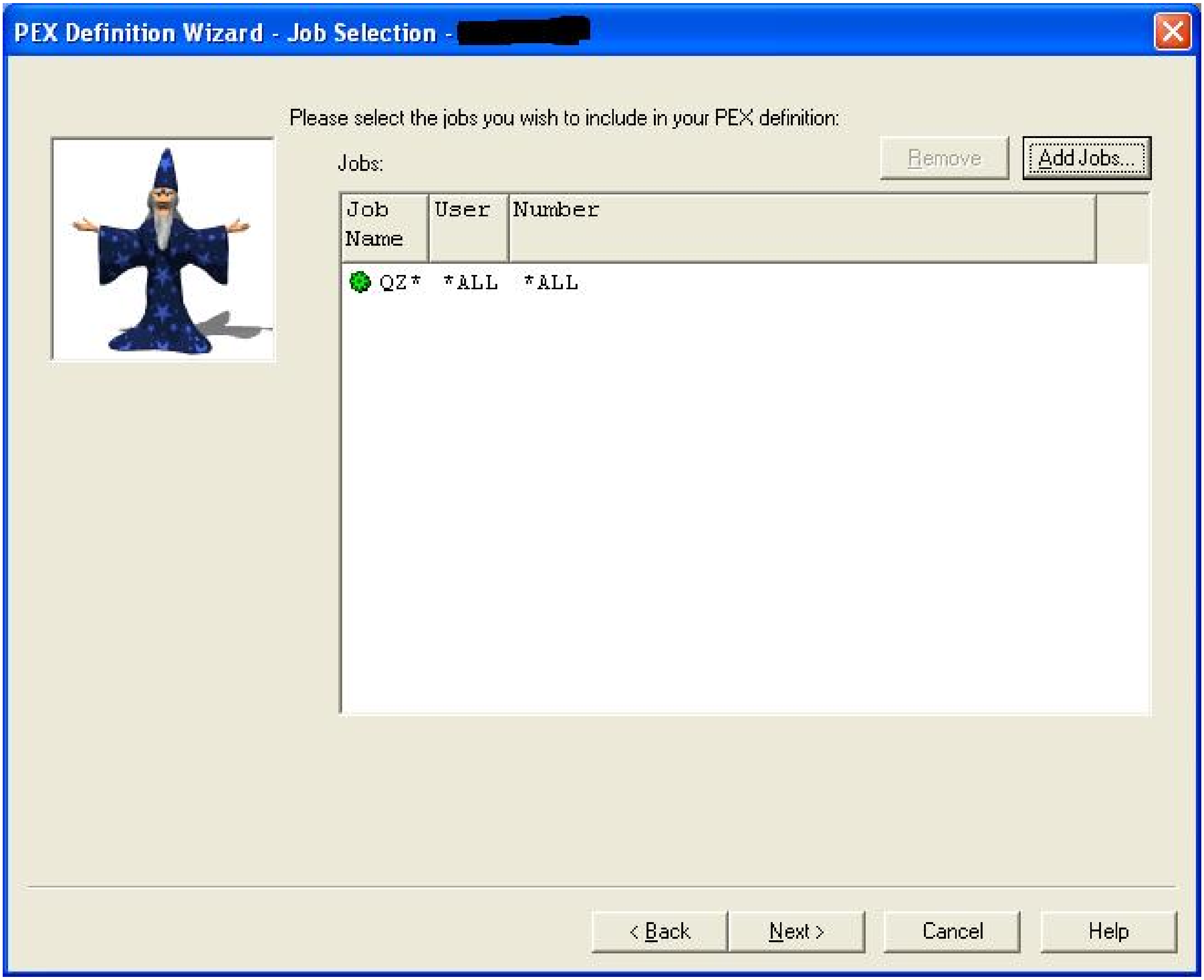
○ All jobs ● Selected jobs ○ Current job

Task selection:

○ All tasks ○ Selected tasks ● None

4.4.1.8 Job Selection

The job selection page displays a list of selected job information to use in the PEX definition. There are also two buttons on this page used to add or remove jobs from the list.



PEX Definition Wizard - Job Selection - [redacted]

Please select the jobs you wish to include in your PEX definition:

Jobs:

Job Name	User	Number
QZ*	*ALL	*ALL

The table below summarizes the different elements on this page:

Field	Description
Jobs list	A list of jobs to collect information about in the PEX definition.

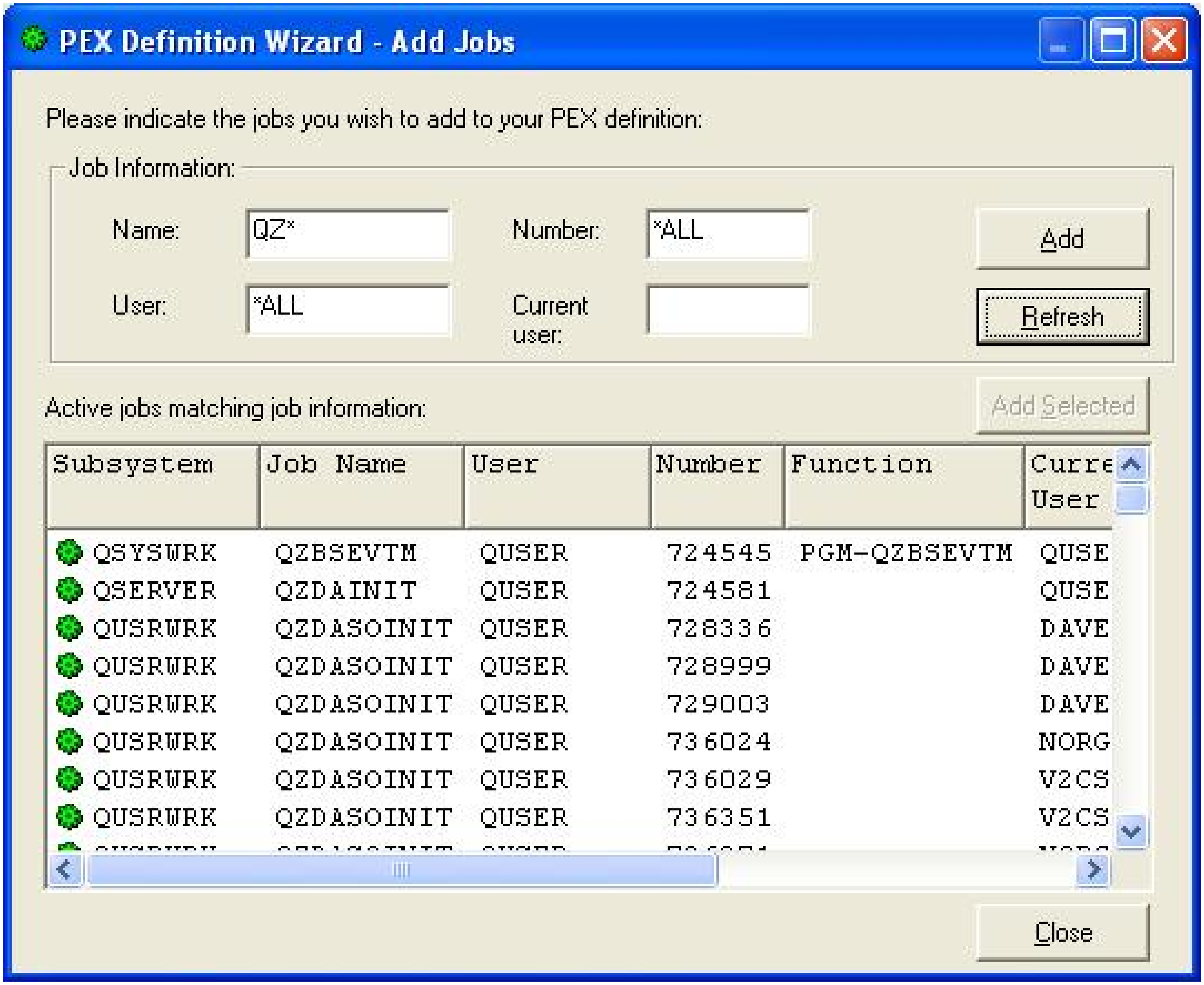
Remove button	This button removes the selected jobs from the list.
Add Jobs button	Use this button to open the Add Jobs Window (discussed in the next section). This window is used to select and add additional jobs to the list.

Table of Contents Previous Next

4.4.1.9 Add Jobs Window

The add jobs window allows a user to add jobs to the Job Selection page in the wizard. Job information can be of two types: generic job name/generic job user/generic job number -or- job name/job user/job number.

The "Job Information" portion of the window includes text fields used to define a generic job to add to the Job Selection Page or to use as a filter when refreshing the list of jobs shown in the window. The Add button will add the current generic job to the Job Selection page and the Add Selected button will add the selected jobs from the active jobs list to the Job Selection page.



PEX Definition Wizard - Add Jobs

Please indicate the jobs you wish to add to your PEX definition:

Job Information:

Name: QZ*

Number: *ALL

User: *ALL

Current user:

Active jobs matching job information:

Subsystem	Job Name	User	Number	Function	Curre User
QSYSWRK	QZBSEVTM	QUSER	724545	PGM-QZBSEVTM	QUSE
QSERVER	QZDAINIT	QUSER	724581		QUSE
QUSRWRK	QZDASOINIT	QUSER	728336		DAVE
QUSRWRK	QZDASOINIT	QUSER	728999		DAVE
QUSRWRK	QZDASOINIT	QUSER	729003		DAVE
QUSRWRK	QZDASOINIT	QUSER	736024		NORG
QUSRWRK	QZDASOINIT	QUSER	736029		V2CS
QUSRWRK	QZDASOINIT	QUSER	736351		V2CS

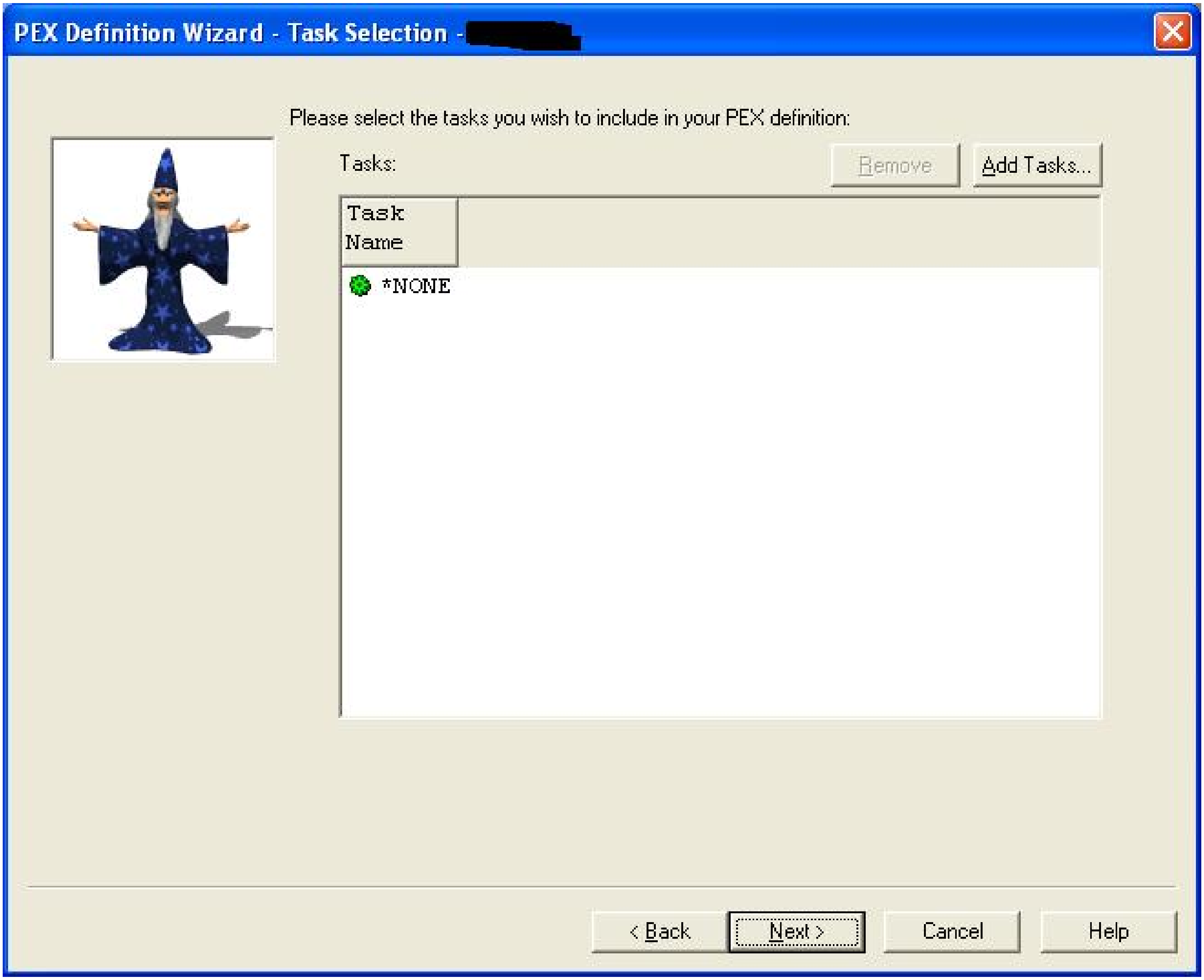
The table below summarizes the different elements on this page:

Field	**Description**
Name	A text field for entering a generic or specific job name. When specifying a generic name use a * at the end of the job name.
User	A text field for entering a generic or specific job user. When specifying a generic name use a * at the end of the job user name.
Number	A text field for entering a specific job number or *ALL.
Current user	This field allows the list of jobs to be filtered by the current user profile value for each job. Unlike name, user and number fields this field does not apply when clicking the **Add** button. It is only used to filter the list of jobs on this window.
Add button	This button will add the current job name/user/number values in the text fields to the Job Selection page. This can be used to add a generic job name/user/number value such as QZ*/FRED/*ALL This value indicates all job names starting with QZ, for job user FRED.
Refresh button	This button is used to refresh the active jobs list based on the current values specified in the name, user, number and current user fields.
Add Selected button	Use this button to add the selected jobs to the Job Selection Page.
Active jobs matching job information list	This list shows all active jobs on the system matching the current Job information specified. When this window is first opened this list will not be shown until the refresh button is pressed.

Table of Contents | Previous | Next

4.4.1.10 Task Selection

The task selection page displays a list of selected tasks to include in the PEX definition. There are also two buttons on this page used to add or remove tasks from the list.



PEX Definition Wizard - Task Selection - [redacted]

Please select the tasks you wish to include in your PEX definition:

Tasks:

Task Name
*NONE

The table below summarizes the different elements on this page:

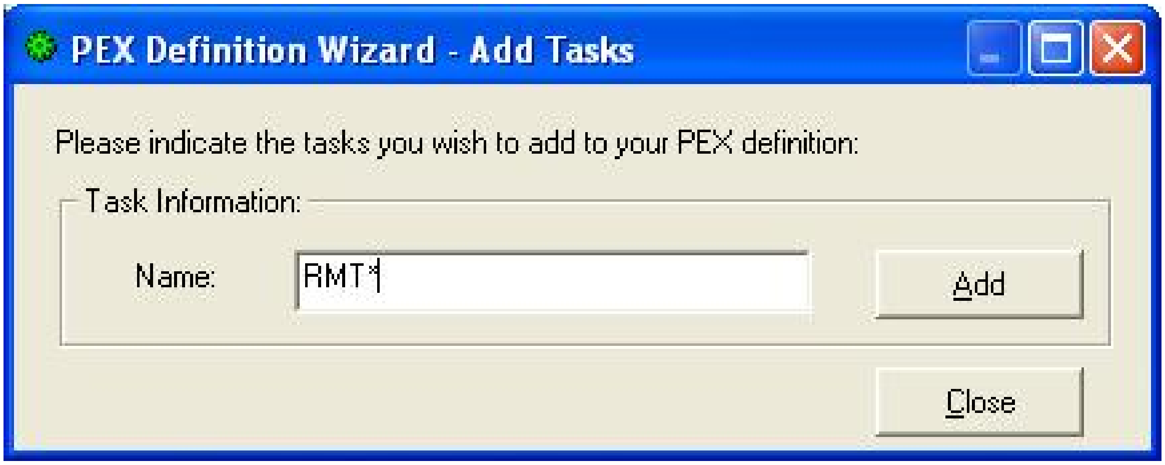
Field	Description
Tasks list	A list of tasks to include in the PEX definition.

Remove button	This button removes the selected tasks from the list.
Add Tasks button	Use this button to open the Add Tasks Window (discussed in the next section). This window is used to add task information to the task list.

Table of Contents Previous Next

4.4.1.11 Add Tasks Window

The add tasks window allows a user to add tasks to the Task Selection page in the wizard. The task name can either be *ALL, *NONE, a generic task name like Q*, or a specific task name. Change the task name field and click the add button for each task that you would like to include in your PEX definition



Please indicate the tasks you wish to add to your PEX definition:

Name: RMT*

The table below summarizes the different elements on this page:

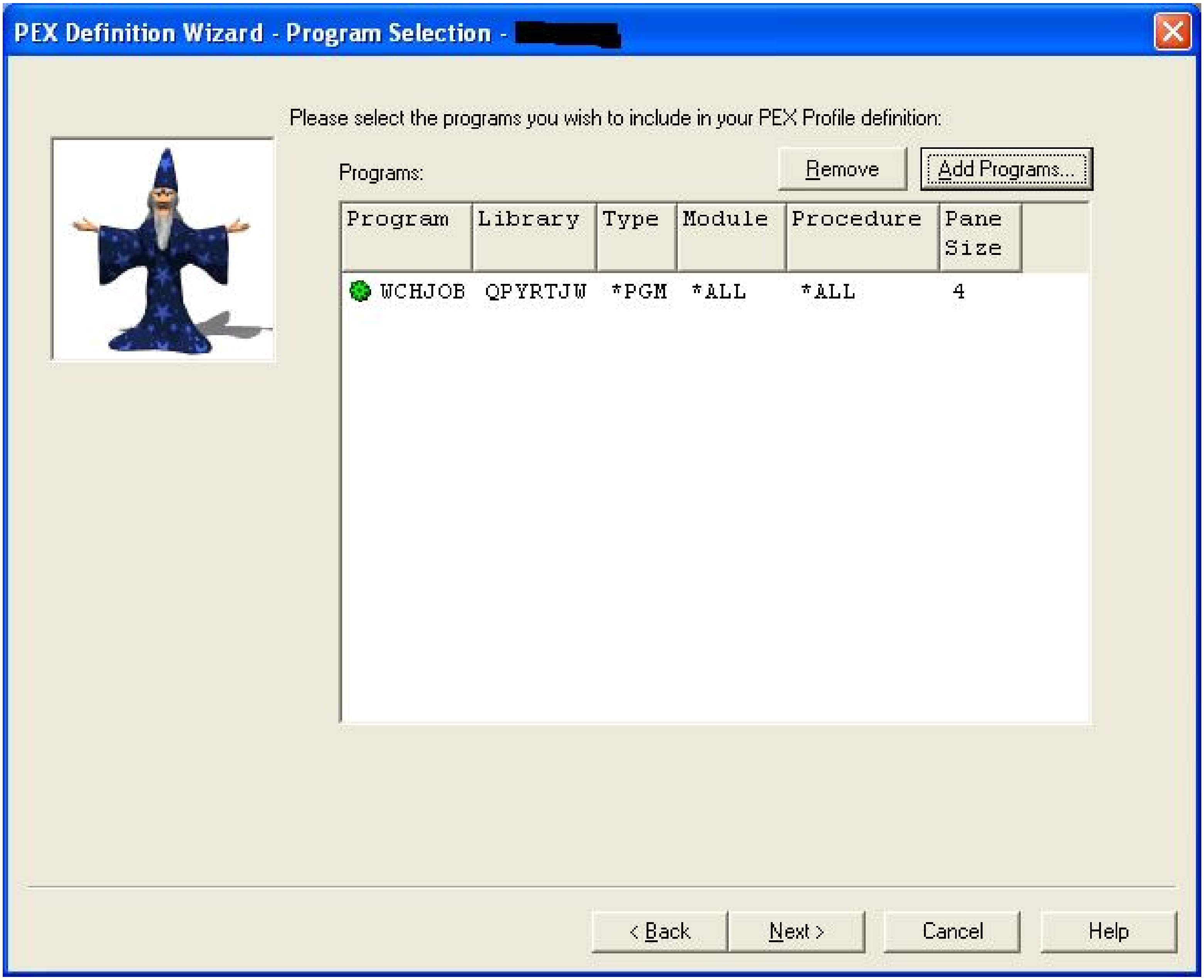
Field	Description
Name	A text field for entering a generic or specific task name. This value can also be *ALL or *NONE. When specifying a generic name use a * at the end of the task name.
Add button	This button will add the current task information to the Task Selection page.
Close button	Closes this window

Table of Contents Previous Next

4.4.1.12 Program Selection

The program selection page allows the user to select up to 16 program/module/procedure entries when creating a PEX Profile definition.

This page displays a list of selected program information to be captured in the PEX definition. Note that in order to get any useful information from this definition type you must select programs that have the enable for profiling flag turned on. There are also two buttons on this page used to add or remove programs from the list.



PEX Definition Wizard - Program Selection - [redacted]

Please select the programs you wish to include in your PEX Profile definition:

Programs:

Program	Library	Type	Module	Procedure	Pane Size
WCHJOB	QPYRTJW	*PGM	*ALL	*ALL	4

The table below summarizes the different elements on this page:

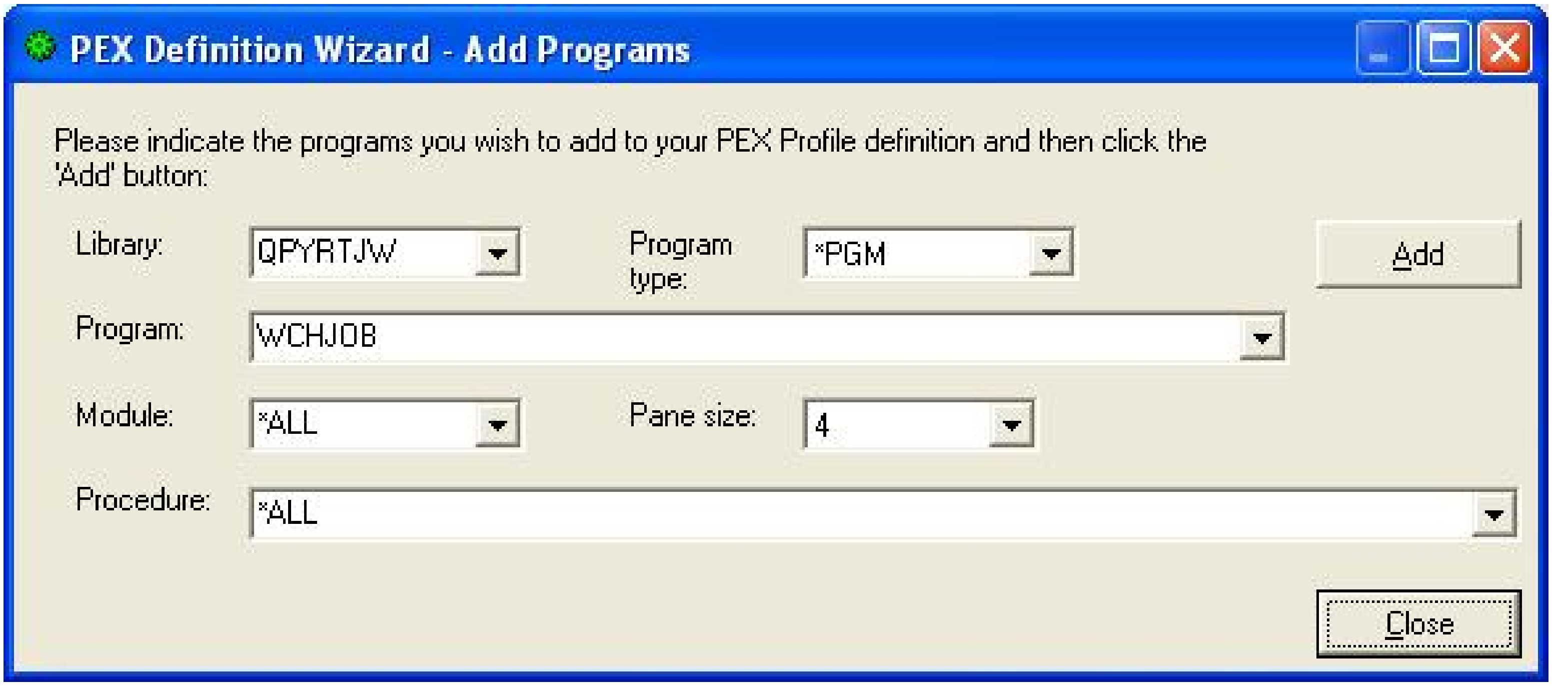
Field	Description
Programs list	A list of program information that will be included in the PEX definition.
Remove button	This button removes the selected program information from the Programs list.
Add Programs button	Use this button to open the Add Programs Window (discussed in the next section). This window is used to select and add additional program information to the Programs list.

Table of Contents Previous Next

4.4.1.13 Add Programs Window

The add programs window allows a user to browse any programs/service programs on the system using generic program and library names for the purpose of adding them to a PEX Profile definition. After finding the programs you want to add to the definition click the Add button to add the selected program/module/procedure to the list.

The enable profiling flag must be turned on in the program and module you select in order to add the program/module/procedure information to the Program Selection Page of the Wizard.



PEX Definition Wizard - Add Programs

Please indicate the programs you wish to add to your PEX Profile definition and then click the 'Add' button:

Library: QPYRTJW

Program type: *PGM

Program: WCHJOB

Module: *ALL

Pane size: 4

Procedure: *ALL

The table below summarizes the different elements on this page:

Field	Description
Library	A dropdown list for entering or selecting the library name.
Program	A dropdown list for entering or selecting the program name.
Type	This drop down lists contains the values *PGM and *SRVPGM. This offers the user the choice of viewing programs or service program objects.
Pane size	The pane size is the number of consecutive program instruction addresses assigned to each counter. The smaller the pane size, the more fine-grained the program profile information will be.
Add button	This button will add the selected program, module and procedure to the Program Selection Page. The program added must be enabled for profiling.

Program List	A dropdown list of programs found within the selcted library. Changing your selected program in the list will update the module dropdown list showing the module information found within the program.
Module List	A dropdown list of modules found within the selected program in the Program List. Changing the selected module in the list will refresh the procedure list showing the procedure information found within the seleted program.
Procedure List	A dropdown list of procedures found within the selected module and program.
Close button	Close the Add Programs window.

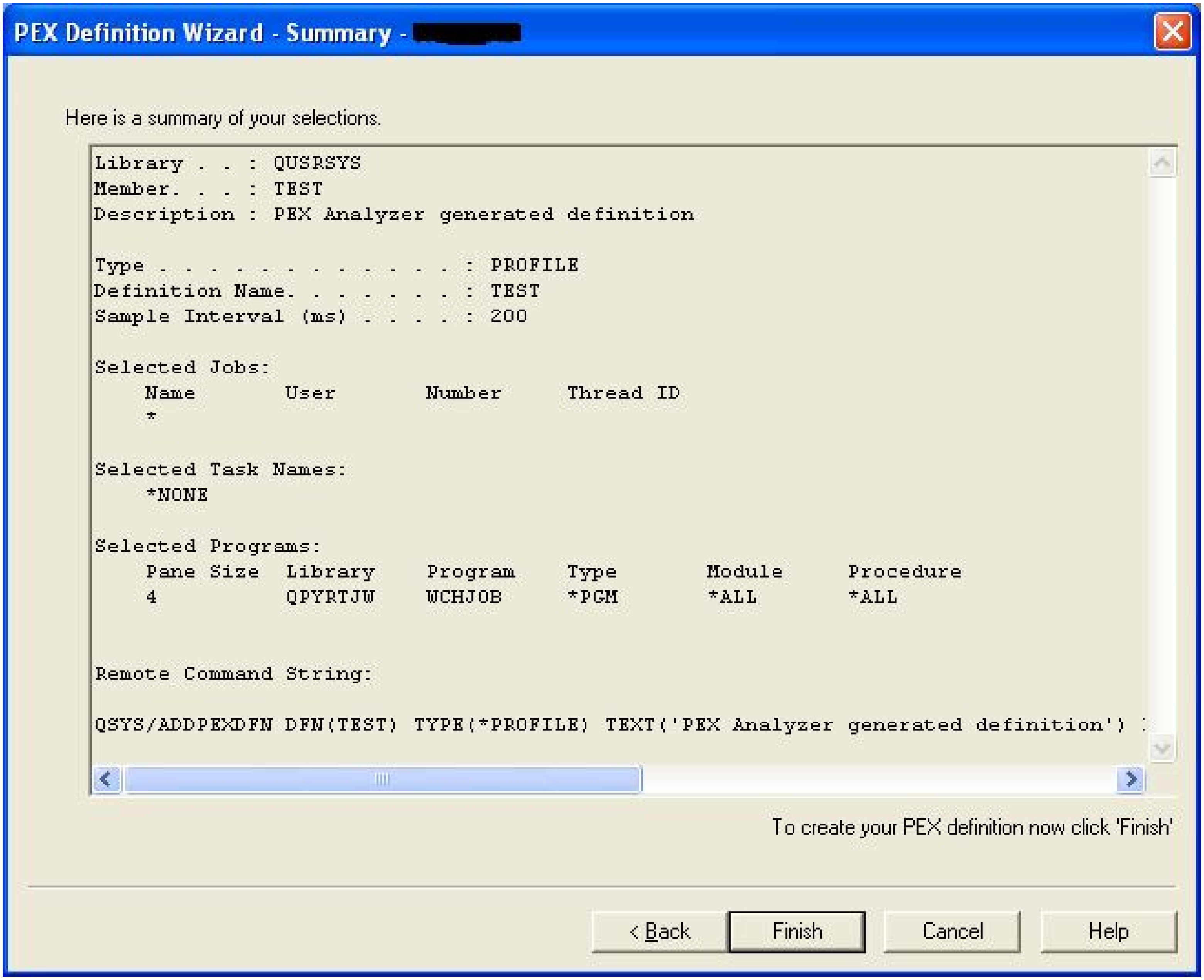
Table of Contents   Previous   Next

4.4.1.14 Summary

The final page of the PEX Definition Wizard presents a summarization of all of the input provided in the wizard. It lists all of the details about the type of PEX definition to create or change, as well as the selected jobs or tasks, and the events to include

To create the PEX definition as defined click on the Finish button. After creating your definition you can use the PEX Collection Wizard to create a PEX collection using your new PEX definition.

When changing a PEX definition the existing definition is first removed and then readded using the ADDPEXDFN command. The CHGPEXDFN command is not used by this interface.


PEX Definition Wizard - Summary - [redacted]

Here is a summary of your selections.

```
Library . . : QUSRSYS
Member. . . : TEST
Description : PEX Analyzer generated definition

Type . . . . . . . . . . . . : PROFILE
Definition Name. . . . . . . : TEST
Sample Interval (ms) . . . . : 200

Selected Jobs:
    Name        User        Number      Thread ID
    *

Selected Task Names:
    *NONE

Selected Programs:
    Pane Size   Library     Program     Type        Module      Procedure
    4           QPYRTJW     WCHJOB      *PGM        *ALL        *ALL


Remote Command String:

QSYS/ADDPEXDFN DFN(TEST) TYPE(*PROFILE) TEXT('PEX Analyzer generated definition')
```

To create your PEX definition now click 'Finish'

Table of Contents | Previous | Next

4.4.2 Deleting

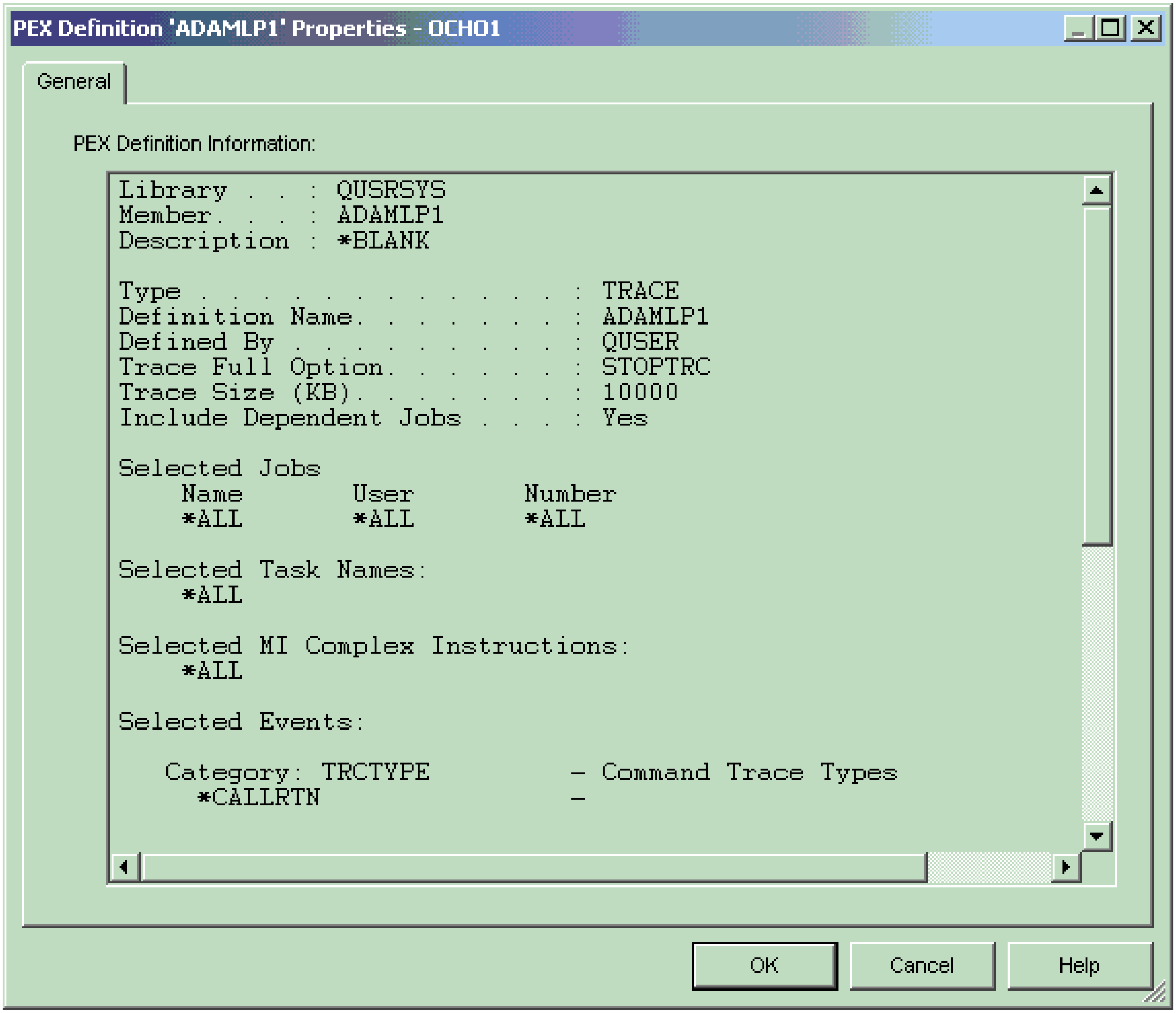
PEX Definitions can be deleted from within the PEX Definitions View. Right-click on one or more PEX definitions and choose the 'Delete...' menu to get rid of any unwanted definitions.

Table of Contents Previous Next

4.4.3 Properties

PEX Definition properties can be accessed from within the PEX definitions view. Double-click on any PEX definition to view its properties or right-click and choose the properties menu after selecting a PEX definition.

The properties shows all of the information about the PEX definition consist with that provided via the PRTPEXRPT command but conveniently via the GUI. You can also access a definitions properties within the PEX Collection Wizard by clicking the 'Details...' button within the Options page of the wizard.



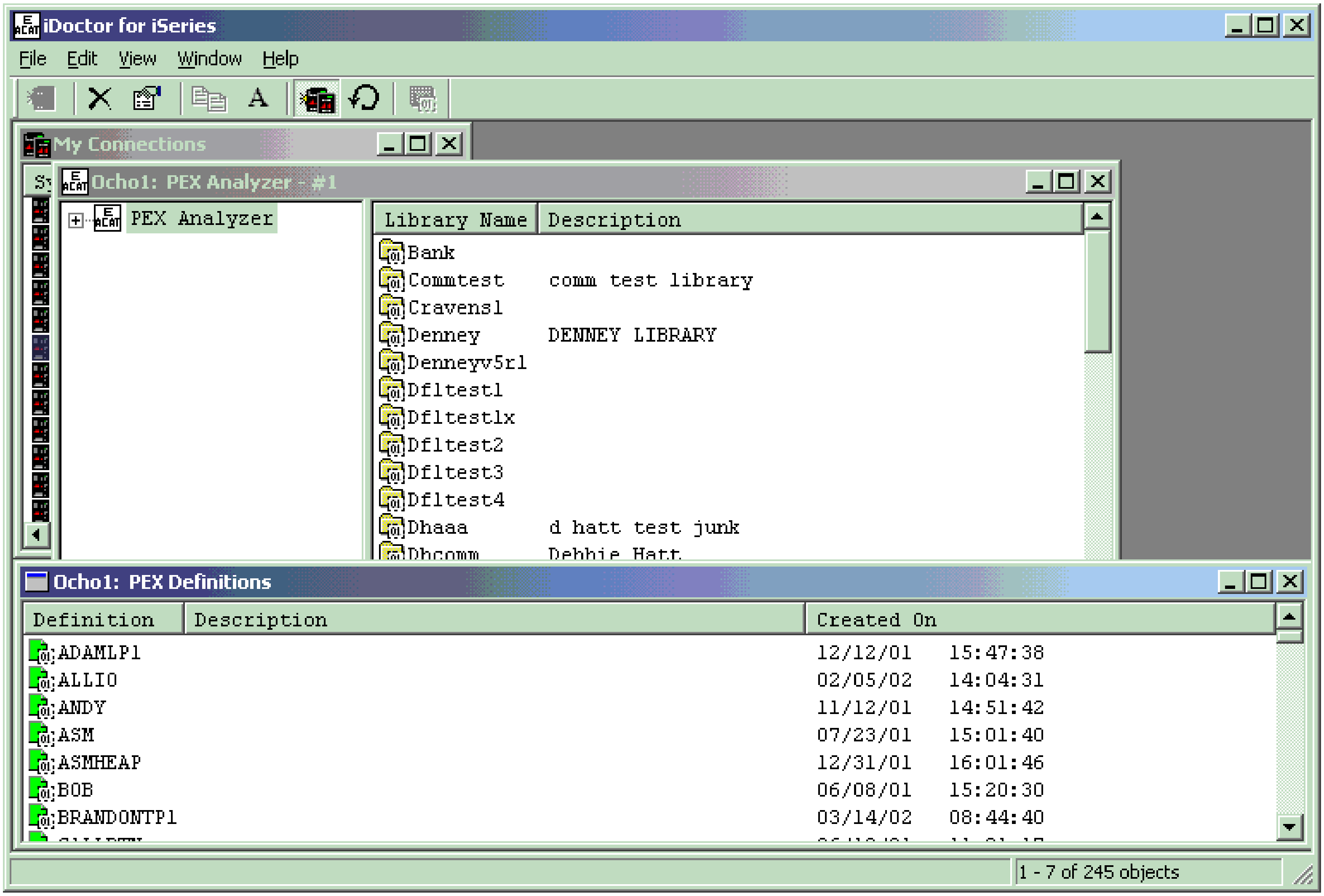
PEX Definition Information:

```
Library . . : QUSRSYS
Member. . . : ADAMLP1
Description : *BLANK

Type . . . . . . . . . . . . . : TRACE
Definition Name. . . . . . . . : ADAMLP1
Defined By . . . . . . . . . . : QUSER
Trace Full Option. . . . . . . : STOPTRC
Trace Size (KB). . . . . . . . : 10000
Include Dependent Jobs . . . . : Yes

Selected Jobs
    Name        User        Number
    *ALL        *ALL        *ALL

Selected Task Names:
    *ALL

Selected MI Complex Instructions:
    *ALL

Selected Events:

   Category: TRCTYPE          - Command Trace Types
     *CALLRTN                 -
```

Table of Contents Previous Next

4.4.4 Working With

PEX Definitions can be viewed and manipulated from the PEX Definitions View. Access this view by using the 'View -> PEX Definitions' menu whenever a PEX Analyzer component view is active and has the current focus. The view contains a list of all PEX definitions defined on the system for the associated PEX Analyzer view.



Library Name	Description
Bank	
Commtest	comm test library
Cravens1	
Denney	DENNEY LIBRARY
Denneyv5r1	
Dfltest1	
Dfltest1x	
Dfltest2	
Dfltest3	
Dfltest4	
Dhaaa	d hatt test junk
Dhcomm	Debbie Hatt

Definition	Description	Created On	
ADAMLP1		12/12/01	15:47:38
ALLIO		02/05/02	14:04:31
ANDY		11/12/01	14:51:42
ASM		07/23/01	15:01:40
ASMHEAP		12/31/01	16:01:46
BOB		06/08/01	15:20:30
BRANDONTP1		03/14/02	08:44:40

Initially the list is sorted by name, by you can sort by time last changed if desired. Click one of the column headers to modify the sort sequence.

Menu Options

There are several options available for PEX definitions summarized in the table below:

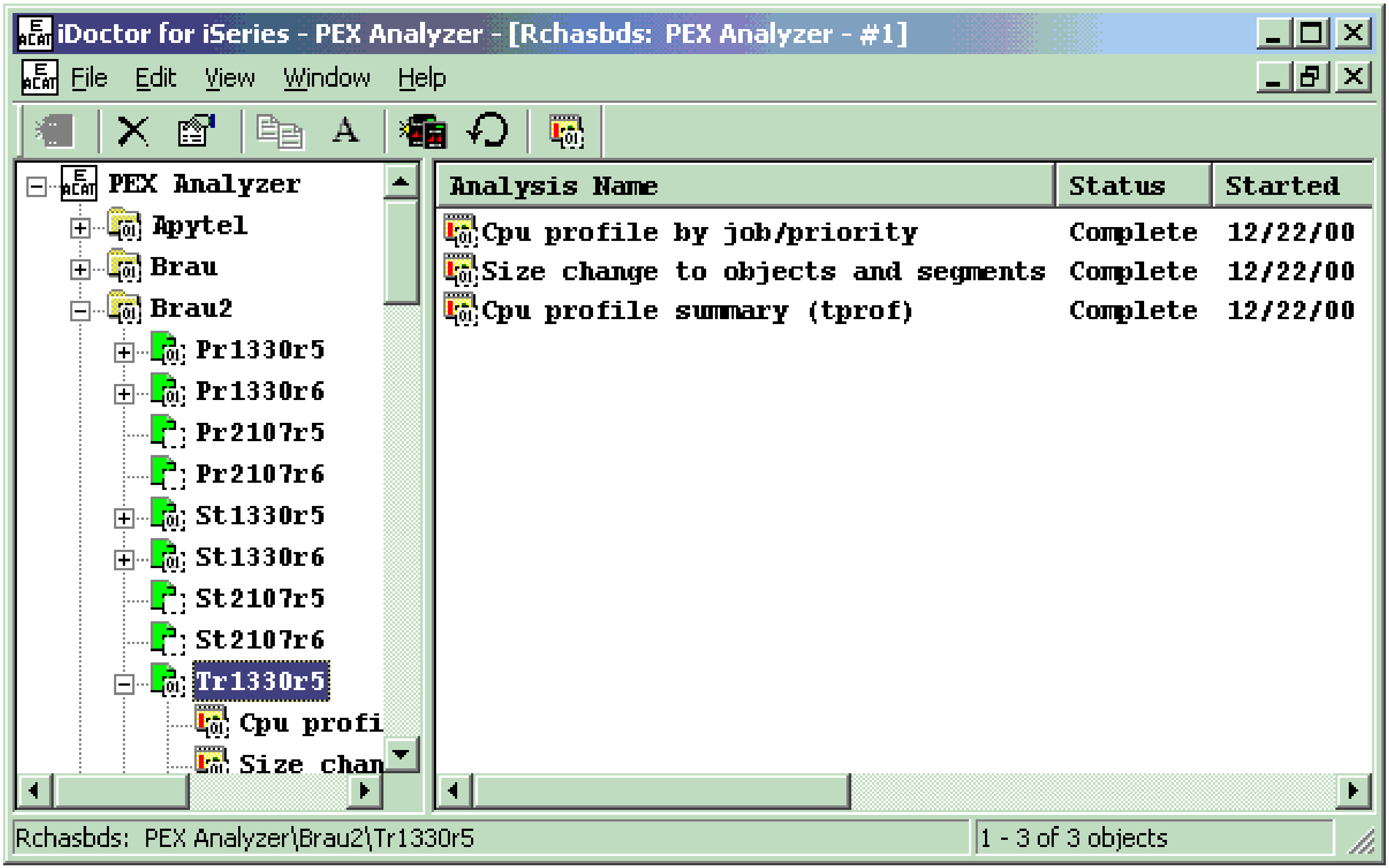
Menu	Description
Create PEX Definition...	Create a new PEX definition using the PEX Definition Wizard.
Change PEX Definition...	Change the PEX definition settings using the PEX Definition Wizard.
Create PEX Collection...	Lets you create a PEX collection within the PEX Collection Wizard using the selected PEX definition.
Delete...	Deletes the selected PEX definitions

Properties	Shows the PEX definition's properties.

Table of Contents Previous Next

4.5 Analyses

Every collection may contain zero or more analyses. Analyses may or may not be useable by the tool depending on the analysis status. The list of analyses displays to the user the analysis type, status, start time, and any subsetting criteria that was applied on the collection data when the analysis was created.



Analysis Name	Status	Started
Cpu profile by job/priority	Complete	12/22/00
Size change to objects and segments	Complete	12/22/00
Cpu profile summary (tprof)	Complete	12/22/00

Rchasbds: PEX Analyzer\Brau2\Tr1330r5

1 - 3 of 3 objects

[Displaying the list of analyses within library 'Brau2', collection 'Tr1330r5']

Table of Contents | Previous | Next

4.5.1 Menu Options

PEX Analyses have the following menu options available:

Menu	Description
Explore	Displays the contents of the analysis (one or more reports) in the right pane of the tree/list window.
Record Quick View	Displays the fields pertaining to the PEX analysis vertically.
Select fields...	Displays an interface that lets you modify which fields are shown for analyses.
Delete...	Deletes an analysis. Select multiple analyses in order to delete more than one analysis at a time.
Properties	Use this menu to display the propery pages for an analysis. The property pages contain general information about the analysis type and status as well as the subsetting information that was applied to the collection when the analysis was created. Intervalized analyses will also have a property page displaying the interval size and start and end time of the analysis.

Table of Contents | Previous | Next

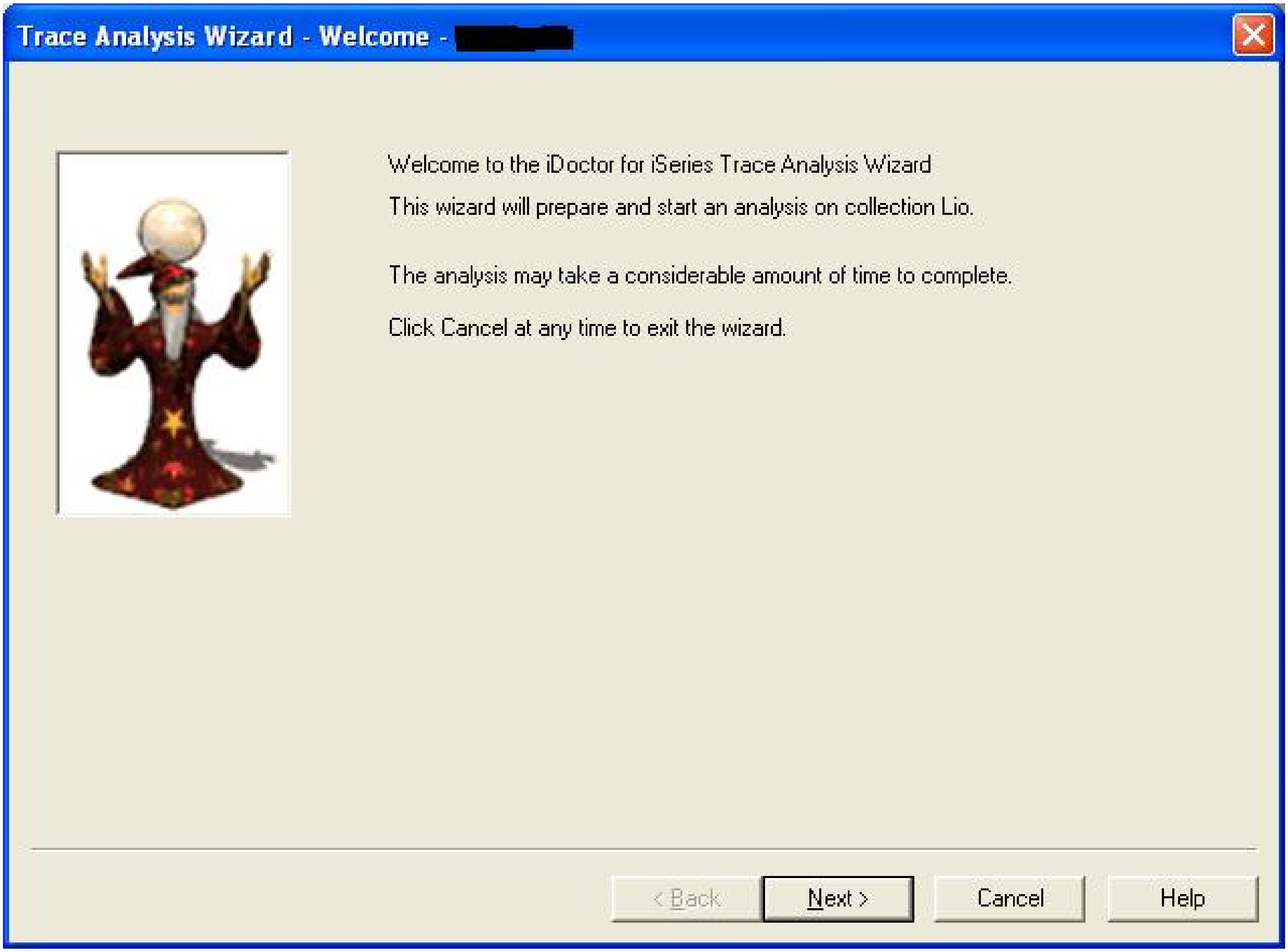
4.5.2 Creating - The Analysis Wizard

The Analysis Wizard offers the iDoctor for iSeries user a convenient way of subsetting a collection and formatting collection data in a way that best suits their particular needs. Selecting the Analyze Data menu for a selected collection invokes the Analysis Wizard. A collection must have a status of 'Ready for analysis' in order for the wizard to be available. The Analysis Wizard gathers information such as what type of analysis to create and how the collection data should be subsetted. This information is used to execute a server-side program call which will create the analysis so it may be viewed in iDoctor.

Depending on the type of collection and what types of analysis and subsetting are possible on the PEX collection the content of the Analysis Wizard will vary. The next sections will list the common and specialized screens within the wizard and provide a discussion of their basic use.

4.5.2.1 Welcome Page

The Welcome Page simply informs the user that the wizard allows them to perform analyses over their PEX collection. The content of this page will differ slightly depending on the type of the PEX collection being analyzed.



Welcome to the iDoctor for iSeries Trace Analysis Wizard

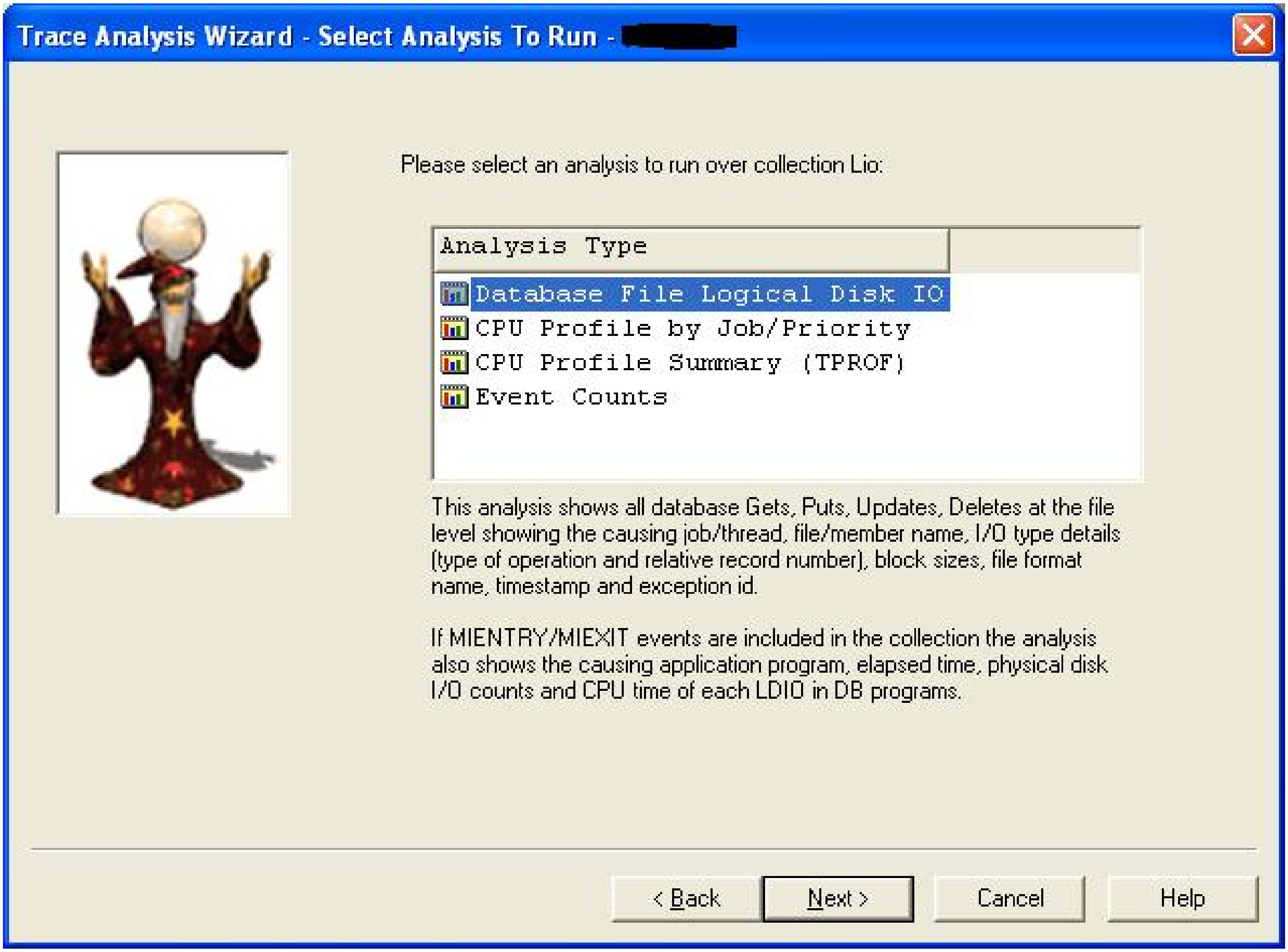
This wizard will prepare and start an analysis on collection Lio.

The analysis may take a considerable amount of time to complete.

Click Cancel at any time to exit the wizard.

4.5.2.2 Select Analysis To Run

The Select Analysis To Run Page is used to select the desired analysis to run on the current collection. The available analysis types are determined when the PEX collection was created on the iSeries system. To help make your selection a detailed description of the currently selected analysis is displayed underneath the list of available analysis types.



Trace Analysis Wizard - Select Analysis To Run - [redacted]

Please select an analysis to run over collection Lio:

Analysis Type
Database File Logical Disk IO
CPU Profile by Job/Priority
CPU Profile Summary (TPROF)
Event Counts

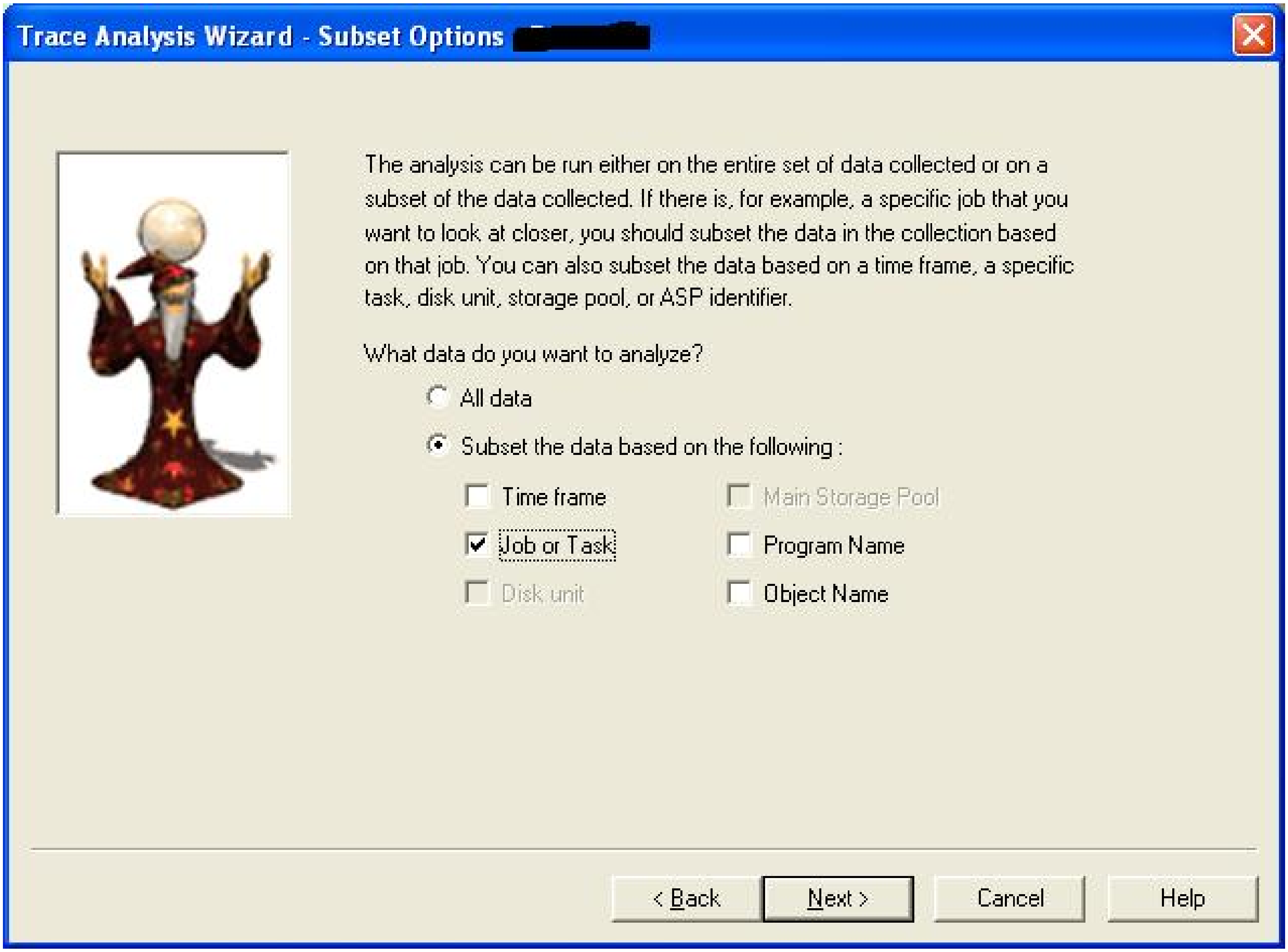
This analysis shows all database Gets, Puts, Updates, Deletes at the file level showing the causing job/thread, file/member name, I/O type details (type of operation and relative record number), block sizes, file format name, timestamp and exception id.

If MIENTRY/MIEXIT events are included in the collection the analysis also shows the causing application program, elapsed time, physical disk I/O counts and CPU time of each LDIO in DB programs.

Table of Contents | Previous | Next

4.5.2.3 Trace Subset Options

The Trace Subset Options Page presents the possible ways the PEX Trace collection may be analyzed. Note: The determination for which options should be enabled is made when the PEX Trace collection is created on the iSeries system. Selecting one or more subset options on this page will cause their corresponding screens to be displayed next in the Analysis Wizard.



Trace Analysis Wizard - Subset Options [redacted]

The analysis can be run either on the entire set of data collected or on a subset of the data collected. If there is, for example, a specific job that you want to look at closer, you should subset the data in the collection based on that job. You can also subset the data based on a time frame, a specific task, disk unit, storage pool, or ASP identifier.

What data do you want to analyze?

○ All data

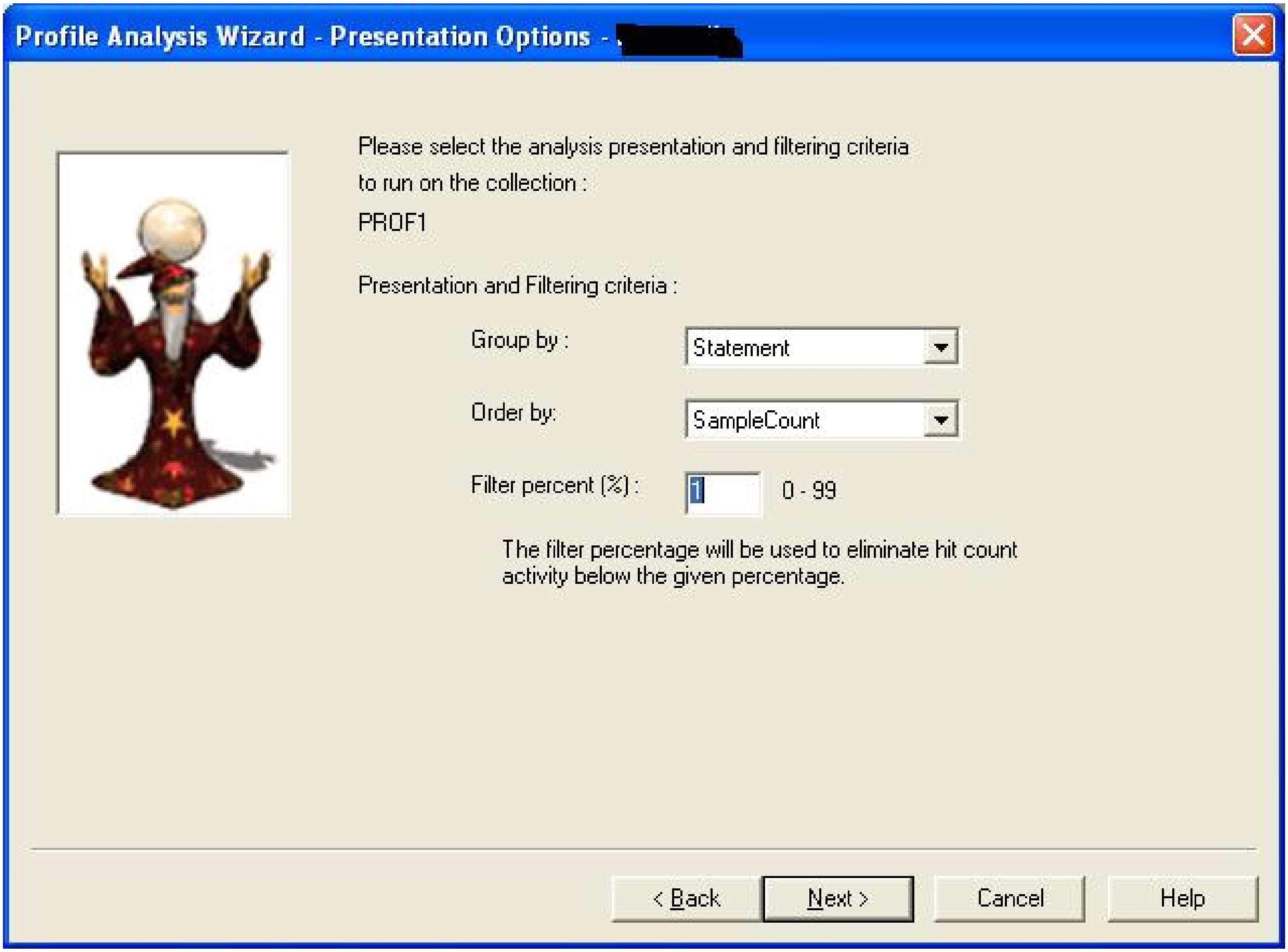
● Subset the data based on the following :

☐ Time frame
☐ Main Storage Pool
☑ Job or Task
☐ Program Name
☐ Disk unit
☐ Object Name

Table of Contents | Previous | Next

4.5.2.4 Profile Presentation Options

The Profile Presentation Options Page allows the user to group and sort the PEX Profile collection data to suit their preferences. The filter percentage field is used to remove any hit count activity below the given percentage.



Profile Analysis Wizard - Presentation Options - [redacted]

Please select the analysis presentation and filtering criteria to run on the collection :

PROF1

Presentation and Filtering criteria :

Group by : Statement

Order by: SampleCount

Filter percent (%) : 1 0 - 99

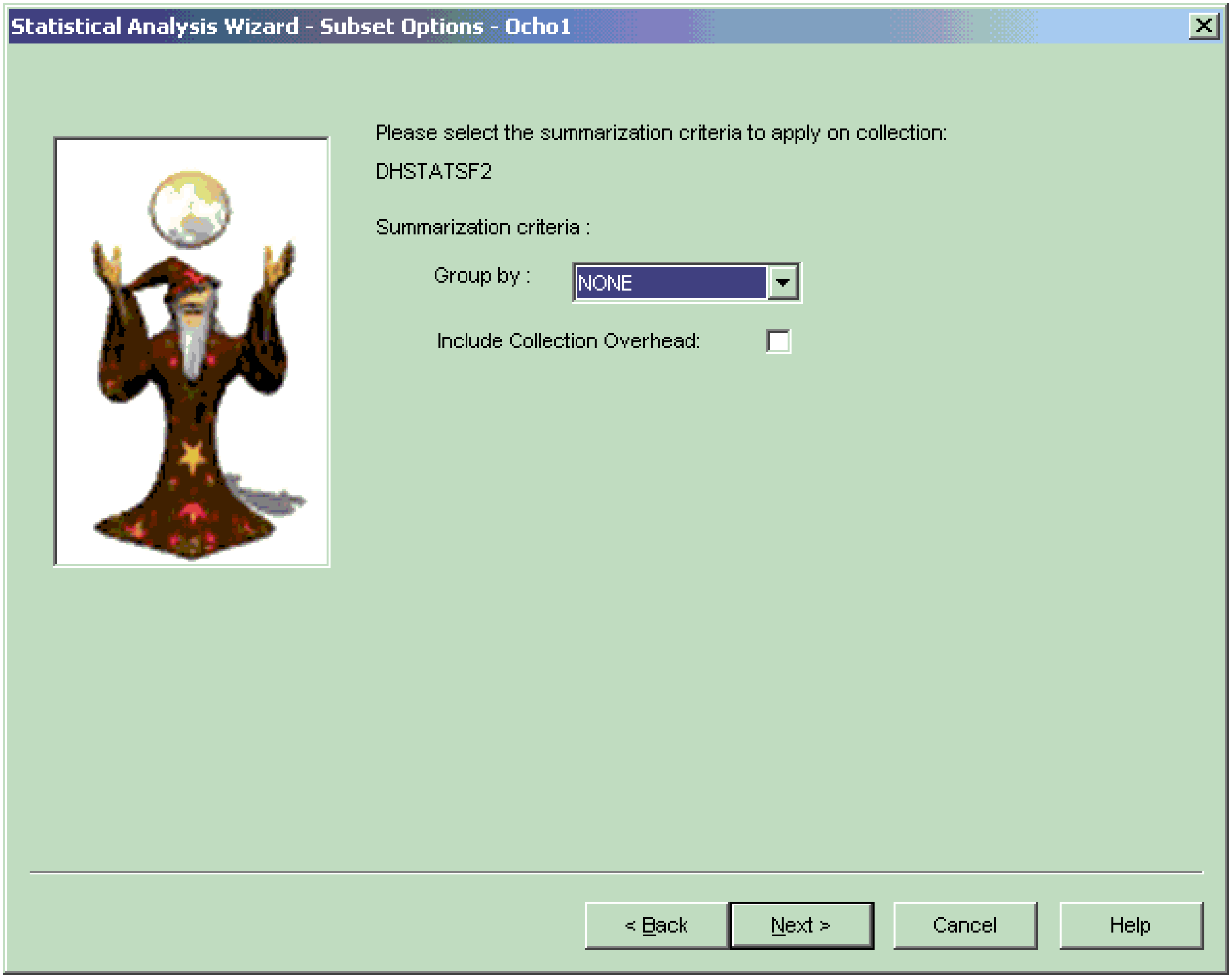
The filter percentage will be used to eliminate hit count activity below the given percentage.

Table of Contents | Previous | Next

4.5.2.5 Statistical Subset Options

The Statistical Subset Options Page allows the user to summarize the PEX Stats collection data to suit their preferences. A "group by" option is used to categorize the analysis data by either module name or program name. An option to include collection overhead is also available. In most cases collection overhead should not be included.

This page is not shown for Statistical Hierarchical collections at release V5R2 or higher.



Statistical Analysis Wizard - Subset Options - Ocho1

Please select the summarization criteria to apply on collection:

DHSTATSF2

Summarization criteria :

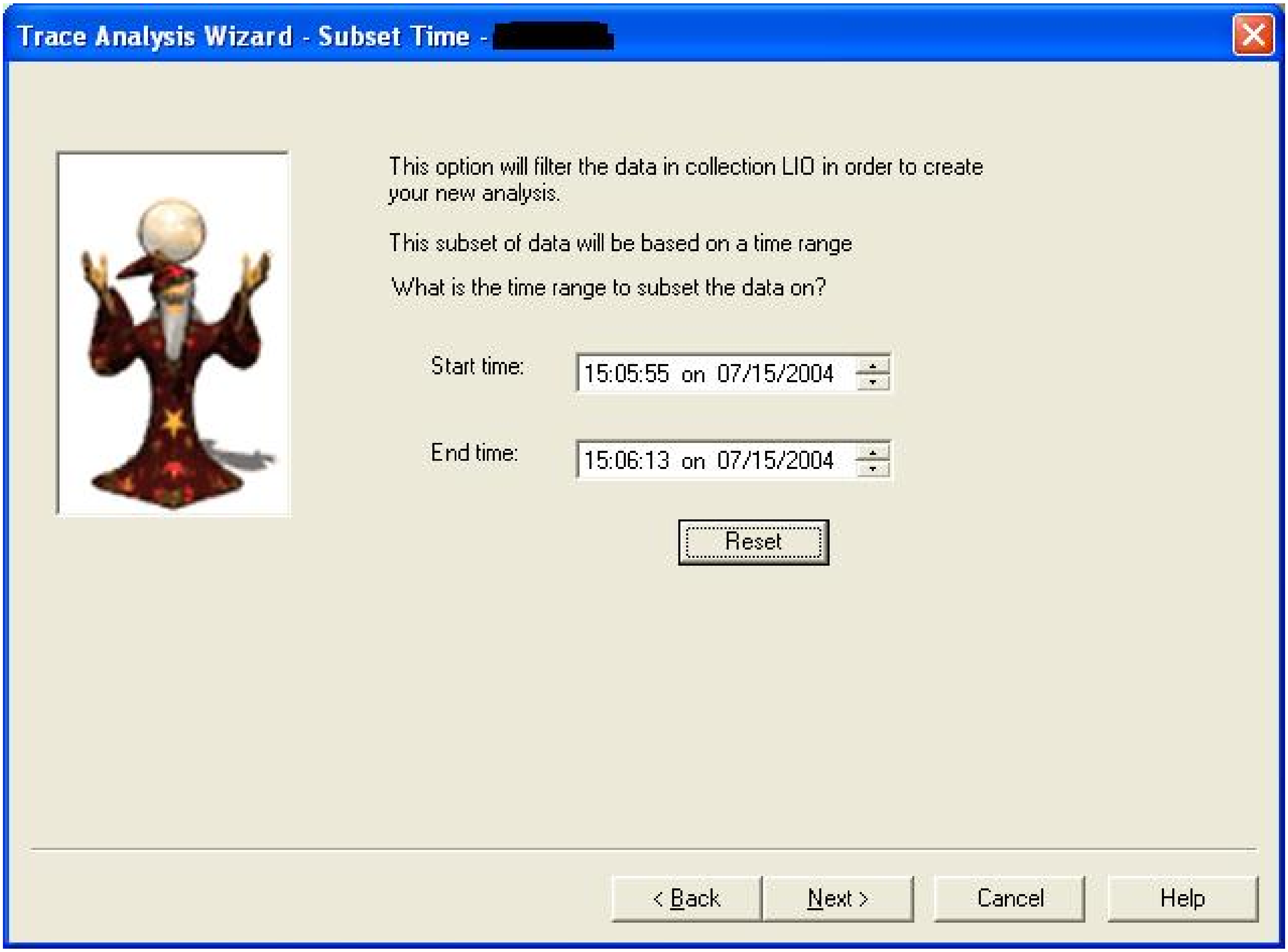
Group by : NONE

Include Collection Overhead: □

Table of Contents | Previous | Next

4.5.2.6 Subset Time

The Subset Time Page allows the user to subset a PEX collection by time in order to look at a specific block of time in the collection. When this page first loads the collection start time and collection end time are shown. Modify the time values to be a subset of the start/end time range in order to subset the PEX collection by time. The Reset button may be used to restore the initial start time and end time values. This type of subsetting is only available for PEX Trace collections.



Trace Analysis Wizard - Subset Time - [redacted]

This option will filter the data in collection LIO in order to create your new analysis.

This subset of data will be based on a time range

What is the time range to subset the data on?

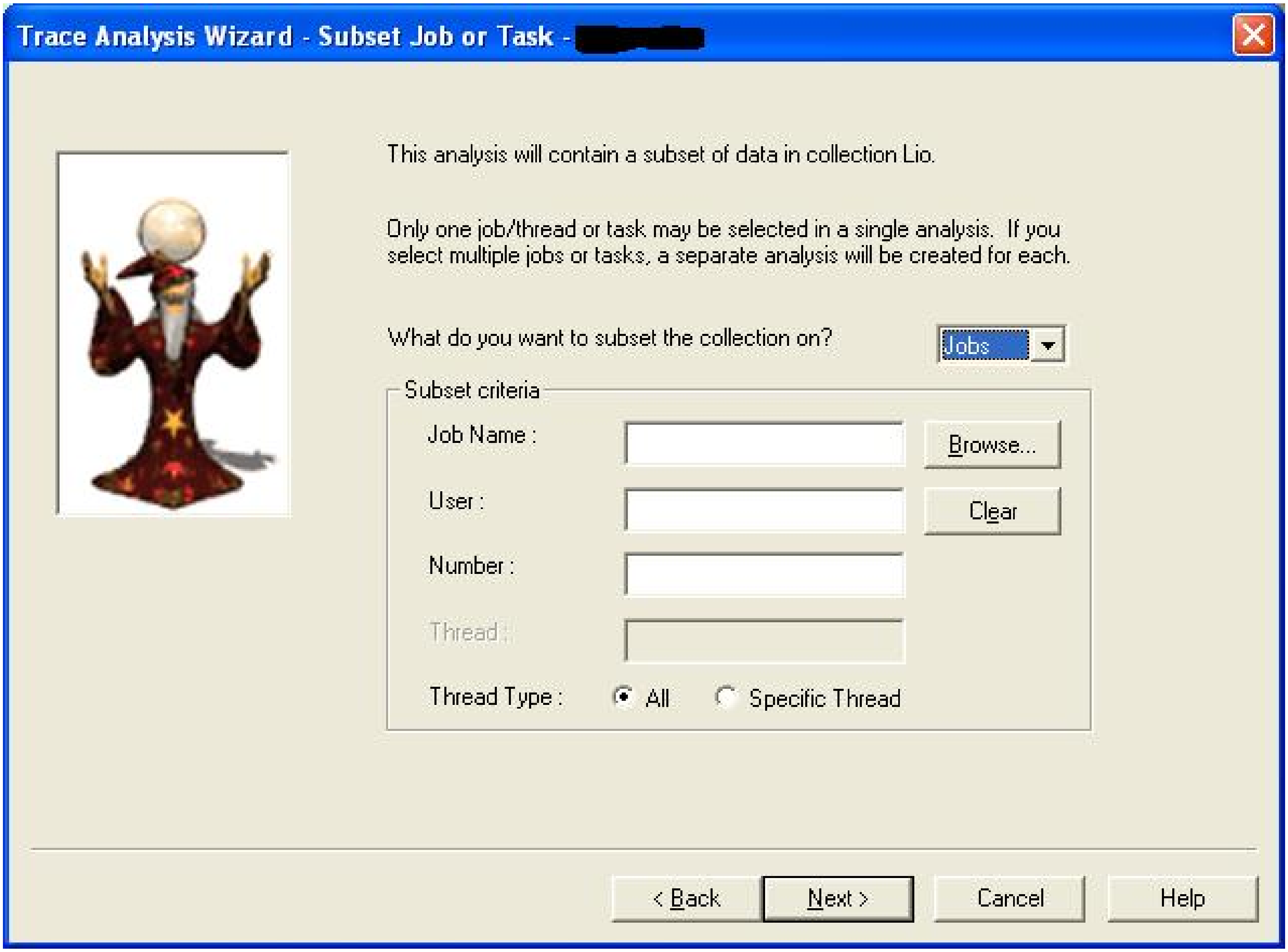
Start time: 15:05:55 on 07/15/2004

End time: 15:06:13 on 07/15/2004

Table of Contents Previous Next

4.5.2.7 Subset Job or Task

The Subset Job or Task Page allows the user to subset a PEX collection by a specific job or task within the PEX collection. Task subsetting is only available on PEX Trace collections. Job subsetting is available on Trace, Statistical Hierarchial, and Statistical Flat Merge Jobs *NO collections.



Trace Analysis Wizard - Subset Job or Task - [redacted]

This analysis will contain a subset of data in collection Lio.

Only one job/thread or task may be selected in a single analysis. If you select multiple jobs or tasks, a separate analysis will be created for each.

What do you want to subset the collection on? Jobs

Subset criteria

Job Name :

User :

Number :

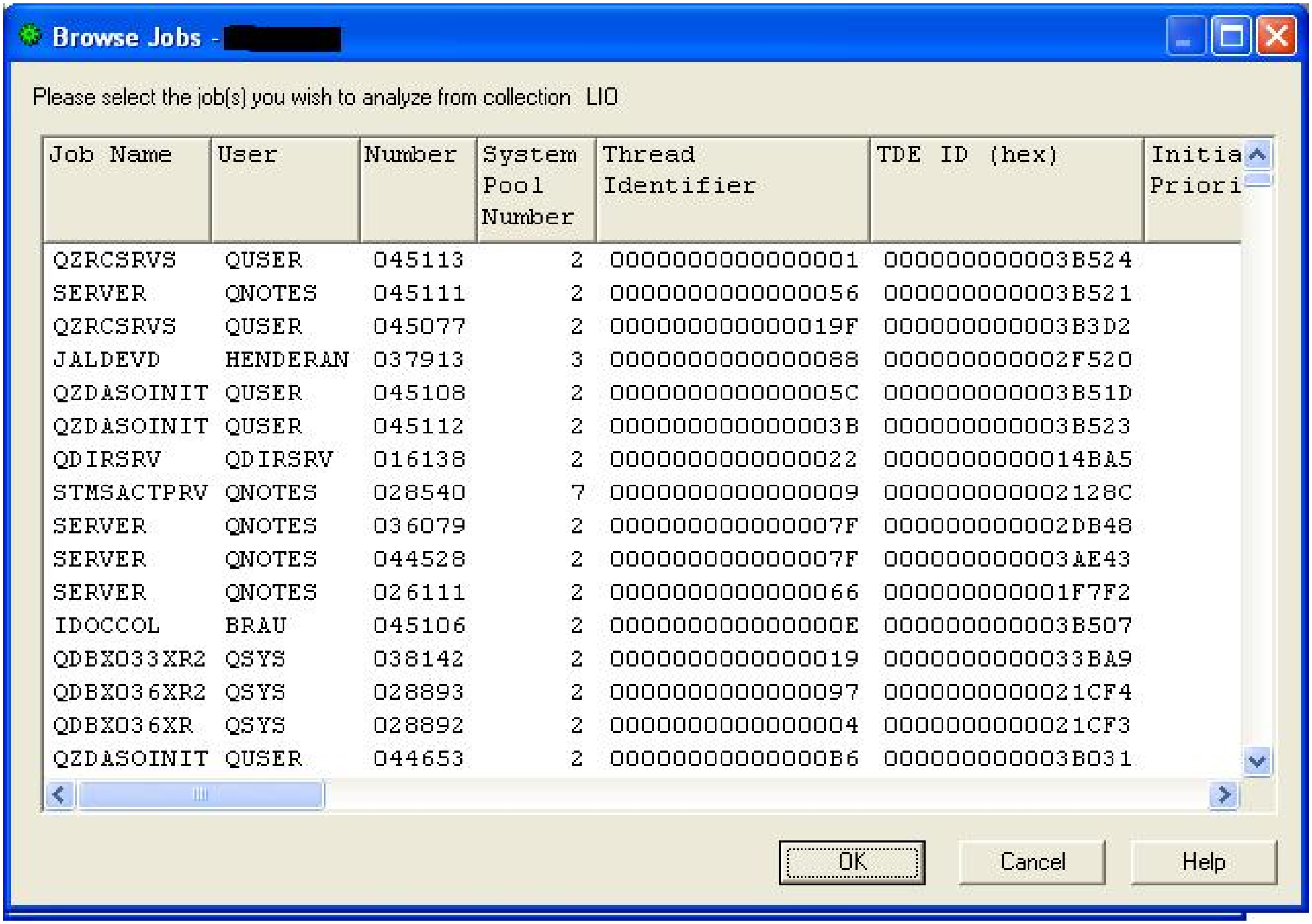
Thread :

Thread Type : ● All ○ Specific Thread

Table of Contents | Previous | Next

4.5.2.7.1 Browse Jobs Dialog

The Browse Jobs Dialog allows a user to select the job they would like to subset the colleciton over. This window is displayed when the user has chosen to subset a collection by job and they click the Browse... button from the Subset Job or Task page in the Analysis Wizard. An option to Find a job in the list is available by right-clicking on the list.



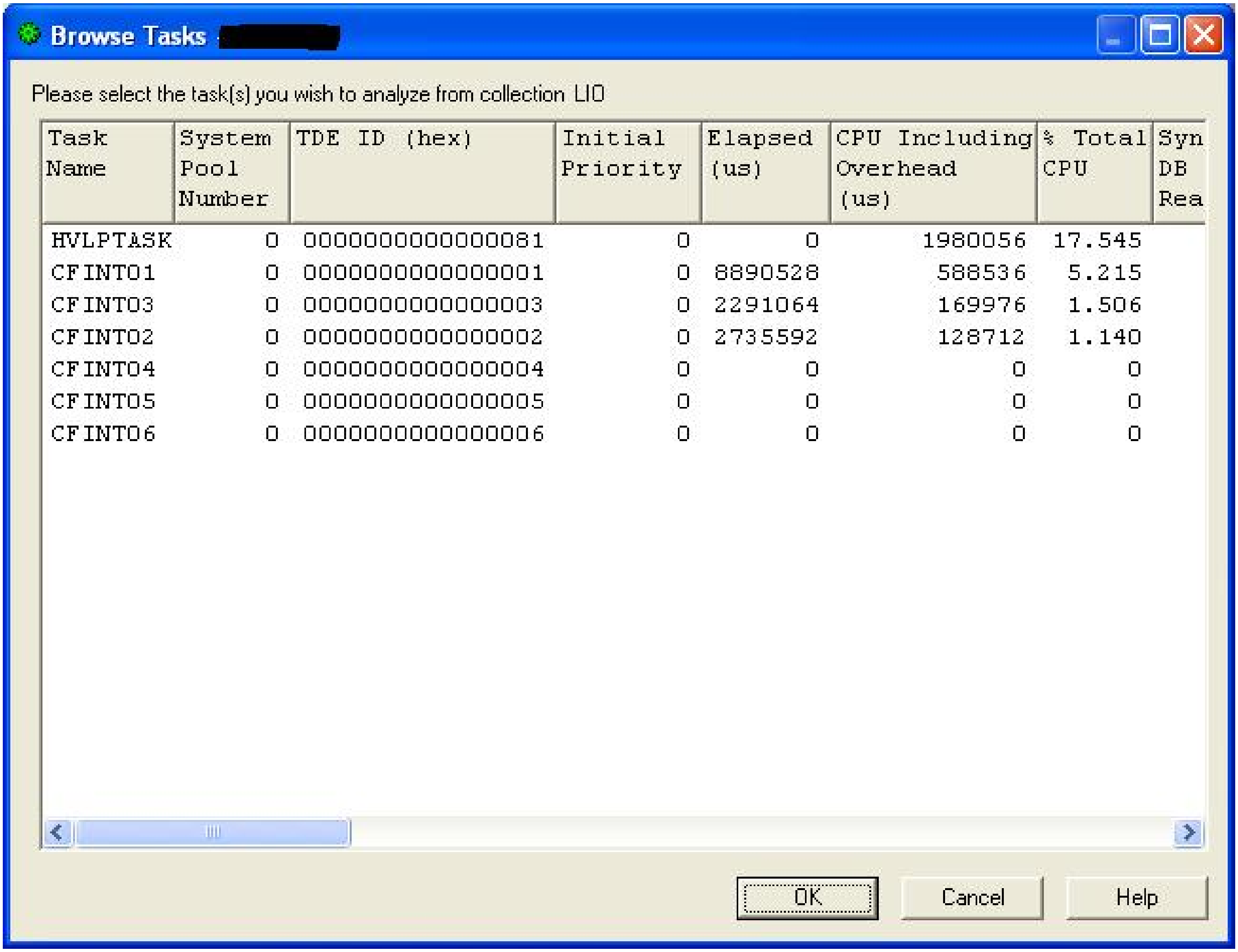
Browse Jobs - [redacted]

Please select the job(s) you wish to analyze from collection LIO

Job Name	User	Number	System Pool Number	Thread Identifier	TDE ID (hex)	Initia Priori
QZRCSRVS	QUSER	045113	2	0000000000000001	000000000003B524	
SERVER	QNOTES	045111	2	0000000000000056	000000000003B521	
QZRCSRVS	QUSER	045077	2	000000000000019F	000000000003B3D2	
JALDEVD	HENDERAN	037913	3	0000000000000088	000000000002F520	
QZDASOINIT	QUSER	045108	2	000000000000005C	000000000003B51D	
QZDASOINIT	QUSER	045112	2	000000000000003B	000000000003B523	
QDIRSRV	QDIRSRV	016138	2	0000000000000022	0000000000014BA5	
STMSACTPRV	QNOTES	028540	7	0000000000000009	000000000002128C	
SERVER	QNOTES	036079	2	000000000000007F	000000000002DB48	
SERVER	QNOTES	044528	2	000000000000007F	000000000003AE43	
SERVER	QNOTES	026111	2	0000000000000066	000000000001F7F2	
IDOCCOL	BRAU	045106	2	000000000000000E	000000000003B507	
QDBX033XR2	QSYS	038142	2	0000000000000019	0000000000033BA9	
QDBX036XR2	QSYS	028893	2	0000000000000097	0000000000021CF4	
QDBX036XR	QSYS	028892	2	0000000000000004	0000000000021CF3	
QZDASOINIT	QUSER	044653	2	00000000000000B6	000000000003B031	

4.5.2.7.2 Browse Tasks Dialog

The Browse Tasks Dialog allows a user to select the task they would like to subset the collection over. This window is displayed when the user has chosen to subset a collection by task and they click the Browse... button from the Subset Job or Task page in the Analysis Wizard. An option to Find a job in the list is available by right-clicking on the list.



Browse Tasks - [redacted]

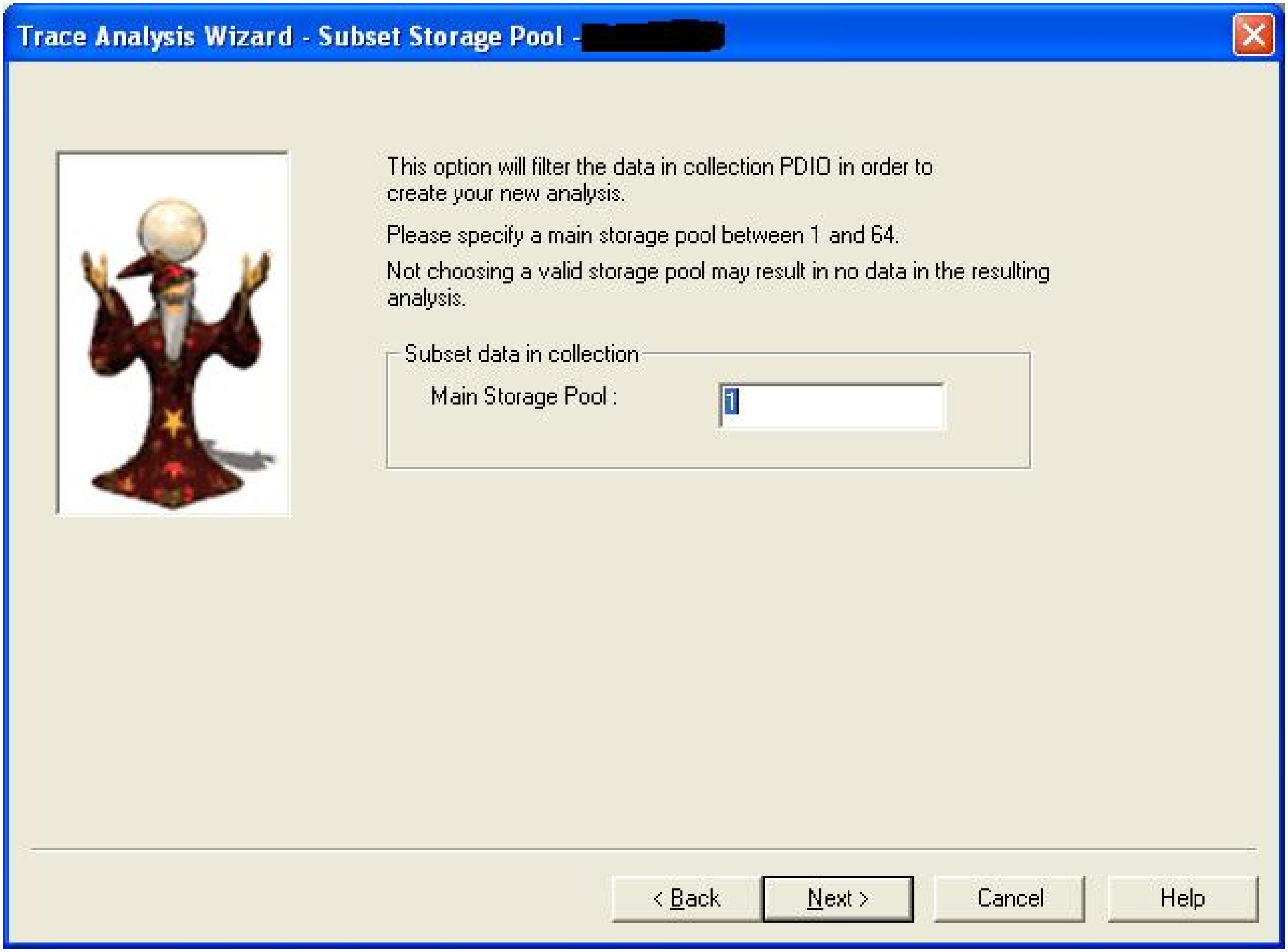
Please select the task(s) you wish to analyze from collection LIO

Task Name	System Pool Number	TDE ID (hex)	Initial Priority	Elapsed (us)	CPU Including Overhead (us)	% Total CPU	Syn DB Rea
HVLPTASK	0	0000000000000081	0	0	1980056	17.545	
CFINT01	0	0000000000000001	0	8890528	588536	5.215	
CFINT03	0	0000000000000003	0	2291064	169976	1.506	
CFINT02	0	0000000000000002	0	2735592	128712	1.140	
CFINT04	0	0000000000000004	0	0	0	0	
CFINT05	0	0000000000000005	0	0	0	0	
CFINT06	0	0000000000000006	0	0	0	0	

Table of Contents | Previous | Next

4.5.2.8 Subset Storage Pool Page

The Subset Storage Pool Page allows a user to subset by main storage pool. If an invalid storage pool is entered no data will be available in the resulting analysis. This type of subsetting is only available for PEX Trace collections.



Trace Analysis Wizard - Subset Storage Pool - [redacted]

This option will filter the data in collection PDIO in order to create your new analysis.

Please specify a main storage pool between 1 and 64.

Not choosing a valid storage pool may result in no data in the resulting analysis.

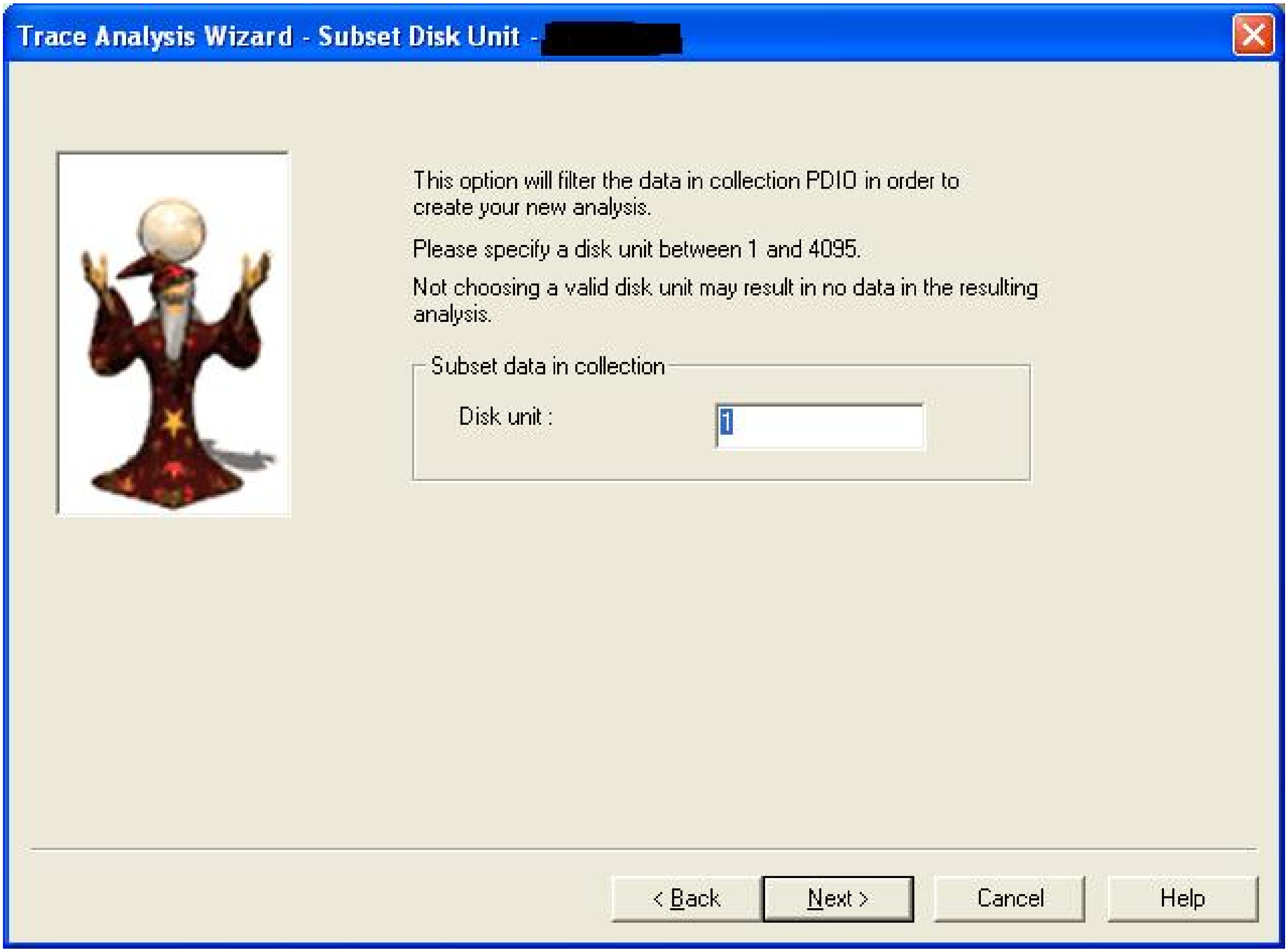
Subset data in collection

Main Storage Pool : 1

Table of Contents Previous Next

4.5.2.9 Subset Disk Unit Page

The Subset Disk Unit Page allows the collection data to subsetted by a specific disk unit. If an invalid disk unit is entered no data will be available in the resulting analysis. This type of subsetting is only available for PEX Trace collections.



Trace Analysis Wizard - Subset Disk Unit - [redacted]

This option will filter the data in collection PDIO in order to create your new analysis.

Please specify a disk unit between 1 and 4095.

Not choosing a valid disk unit may result in no data in the resulting analysis.

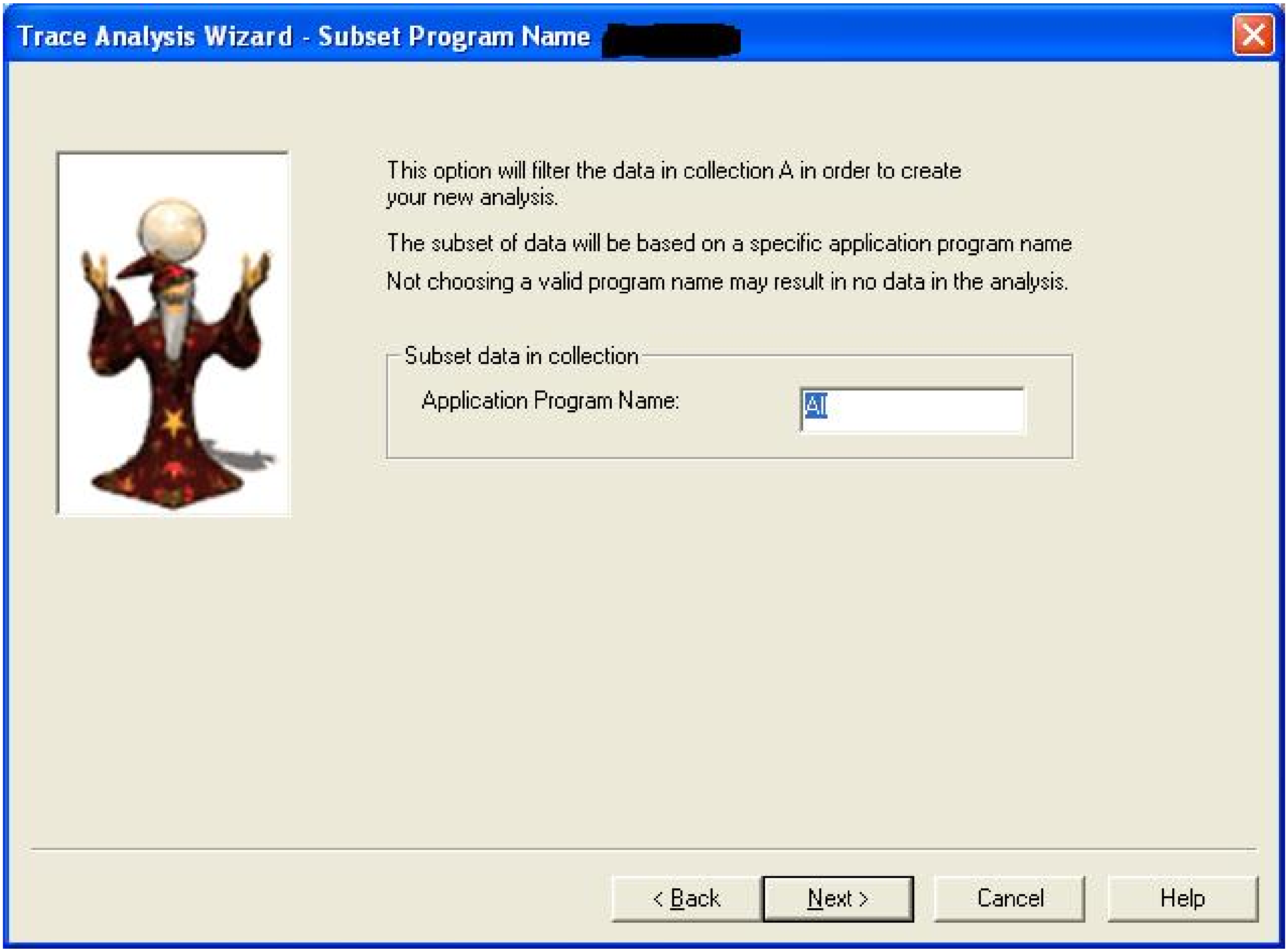
Subset data in collection

Disk unit : 1

Table of Contents Previous Next

4.5.3.10 Subset Program Page

The Subset Program Page allows a user to subset on a specific application program name. Choosing All or leaving the field blank will include all programs in the resulting analysis. This type of subsetting is only available for PEX Trace collections.



Trace Analysis Wizard - Subset Program Name [redacted]

This option will filter the data in collection A in order to create your new analysis.

The subset of data will be based on a specific application program name

Not choosing a valid program name may result in no data in the analysis.

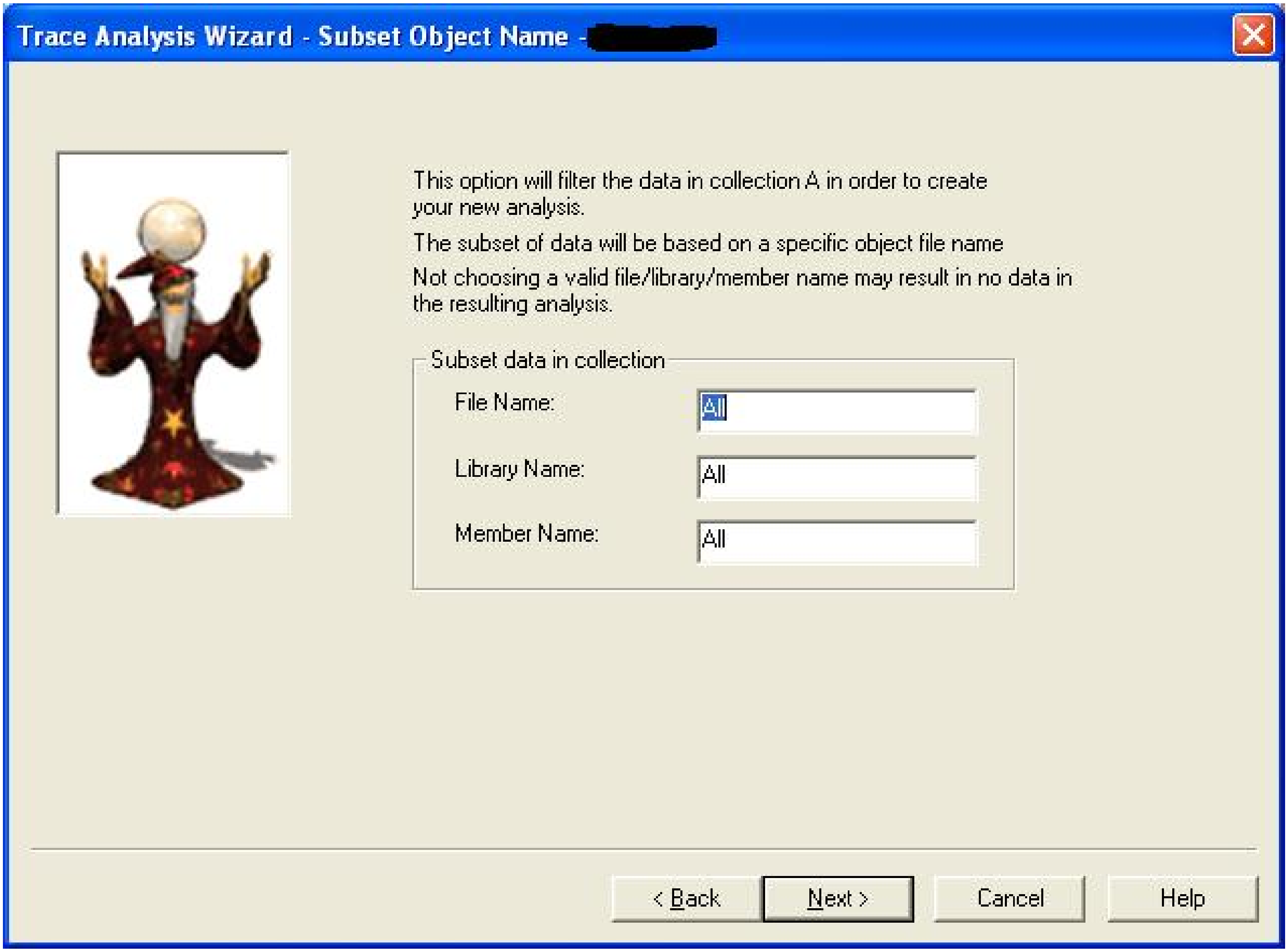
Subset data in collection

Application Program Name: All

Table of Contents Previous Next

4.5.3.11 Subset Object Page

The Subset Object Page allows a user to subset on a specific object filename. If an invalid library, file, member combination is entered no data will be available in the resulting analysis. This type of subsetting is only available for PEX Trace collections.



Trace Analysis Wizard - Subset Object Name - [redacted]

This option will filter the data in collection A in order to create your new analysis.

The subset of data will be based on a specific object file name

Not choosing a valid file/library/member name may result in no data in the resulting analysis.

Subset data in collection

File Name: All

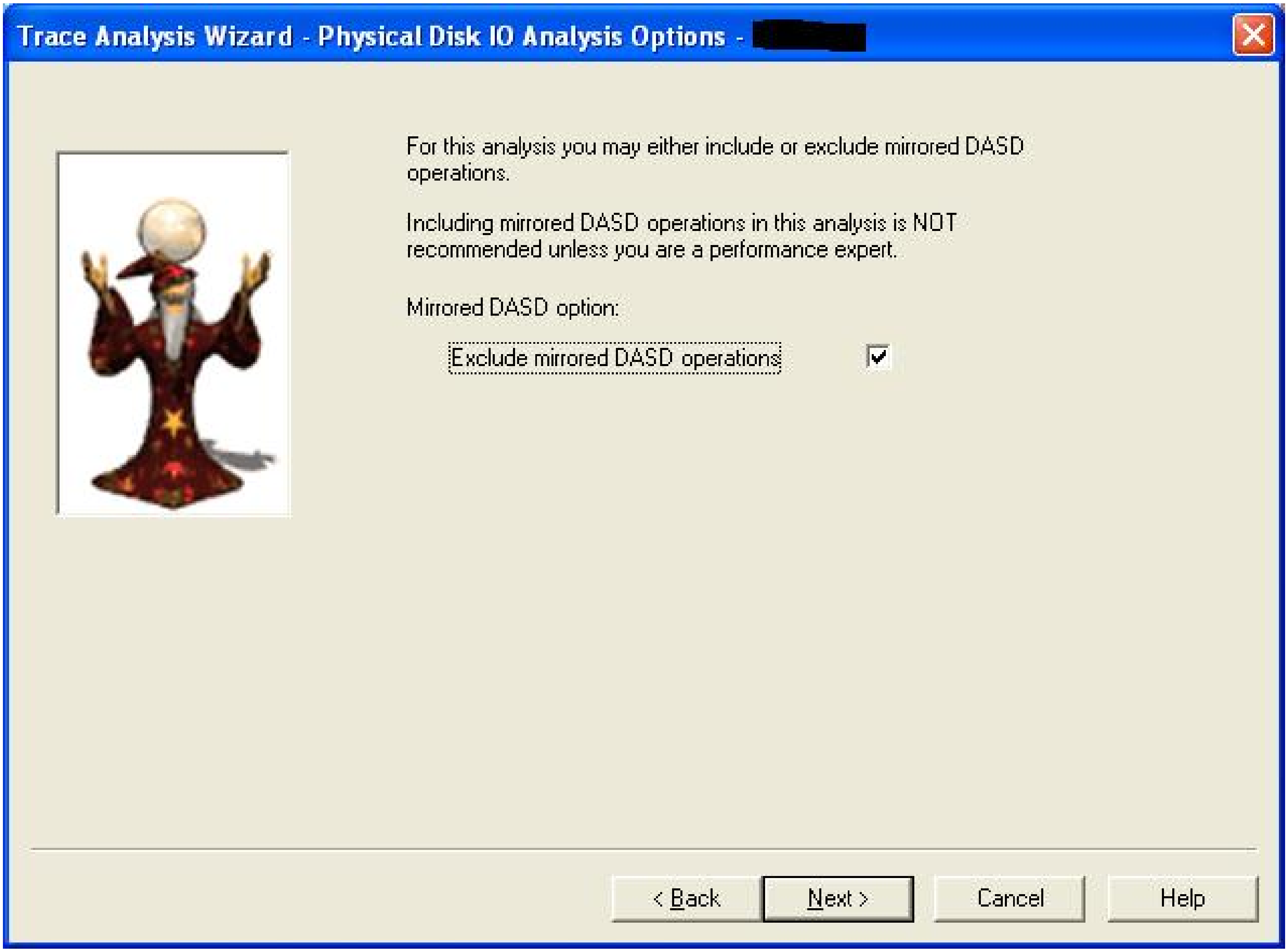
Library Name: All

Member Name: All

4.5.3.12 Physical Disk IO Analysis Options

This page provides the options that only apply to the Physical Disk IO analysis. Mirrored DASD operations may be included or excluded from the analysis data by selecting the box.

The default and recommended option is to exclude mirrored DASD operations.



Trace Analysis Wizard - Physical Disk IO Analysis Options - [redacted]

For this analysis you may either include or exclude mirrored DASD operations.

Including mirrored DASD operations in this analysis is NOT recommended unless you are a performance expert.

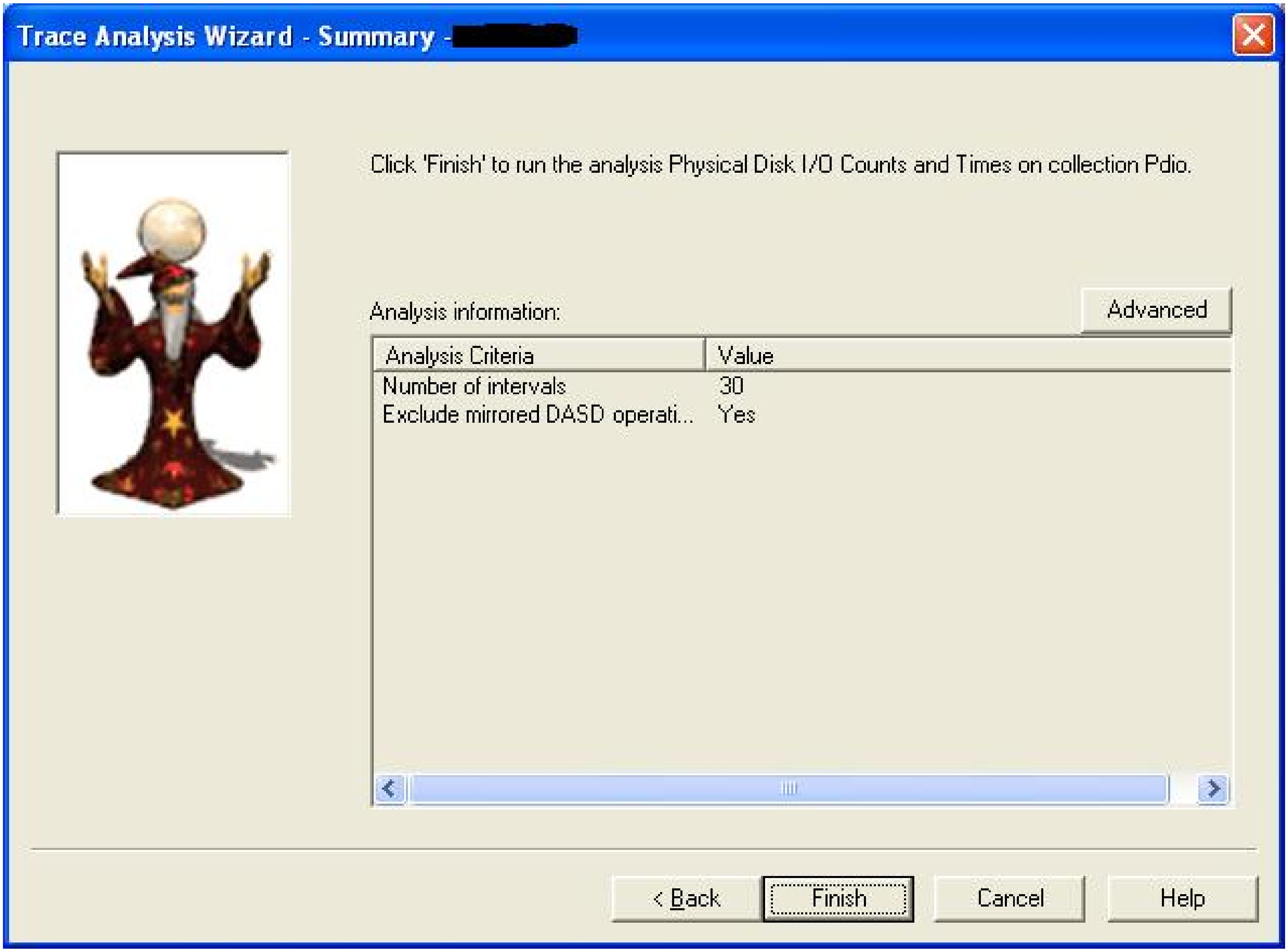
Mirrored DASD option:

Exclude mirrored DASD operations ☑

Table of Contents | Previous | Next

4.5.3.13 Summary Page

This final page of the Wizard lists all of the selection criteria identifying how the analysis will be created. Clicking Finish will submit the analysis request to the iSeries system. If the current collection is a PEX Trace collection the Advanced button is available for setting the optional value for the number of intervals the data will be subsetted into.



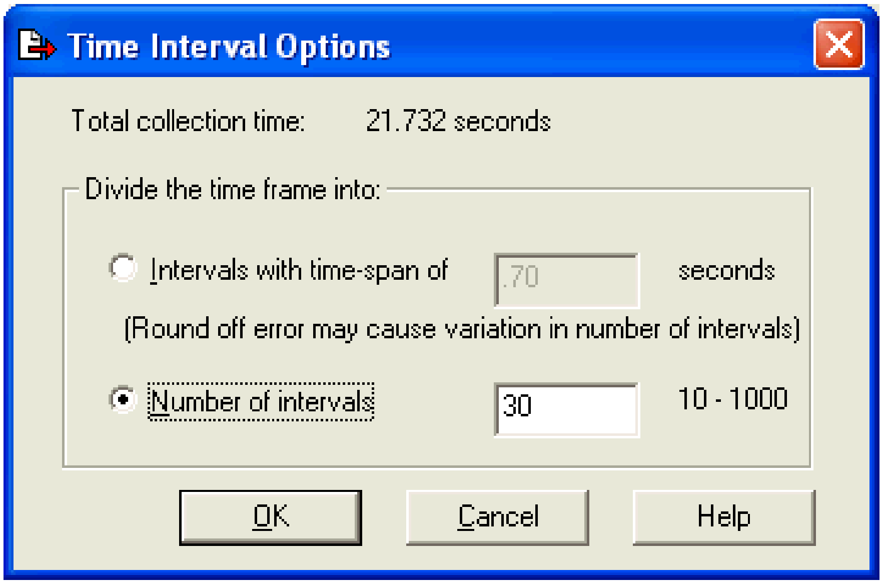
Trace Analysis Wizard - Summary - [redacted]

Click 'Finish' to run the analysis Physical Disk I/O Counts and Times on collection Pdio.

Analysis information:

Analysis Criteria	Value
Number of intervals	30
Exclude mirrored DASD operati...	Yes

The Advanced button will display the following window for specifying how the trace data should be split up for this analysis. Intervals can be created based on a single user specified number of intervals or by specifying a desired interval length in seconds.



Total collection time: 21.732 seconds

Divide the time frame into:

○ Intervals with time-span of .70 seconds

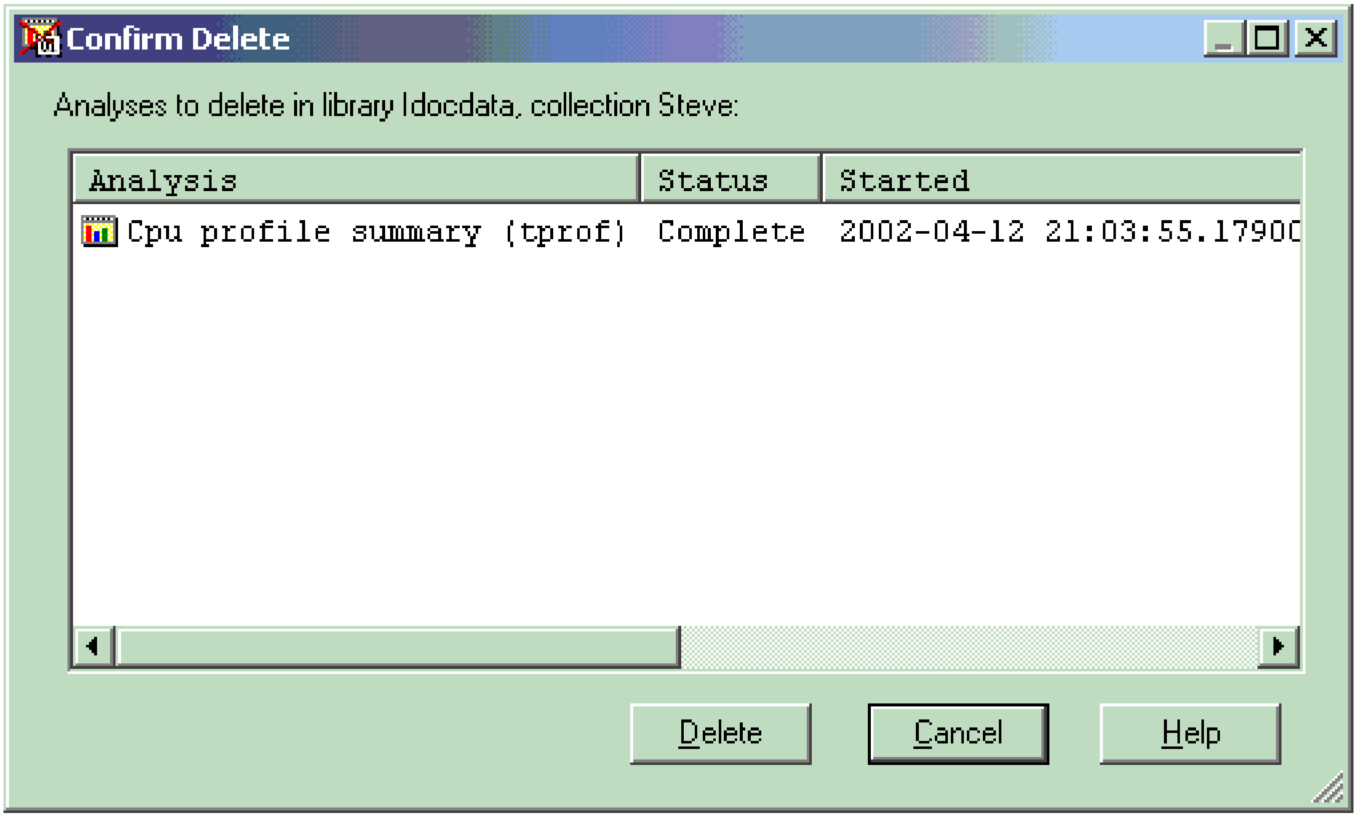
(Round off error may cause variation in number of intervals)

● Number of intervals 30 10 - 1000

Table of Contents Previous Next

4.5.3 Deleting

A user may choose to delete one or more analyses. Select the analyses you would like to delete and choose the 'Delete...' menu. If you have trouble deleting an analysis there may be someone else using it (possibly yourself). Ensure that all table or graph views over the analysis to be deleted have been closed down.



Analyses to delete in library Idocdata, collection Steve:

Analysis	Status	Started
Cpu profile summary (tprof)	Complete	2002-04-12 21:03:55.17900

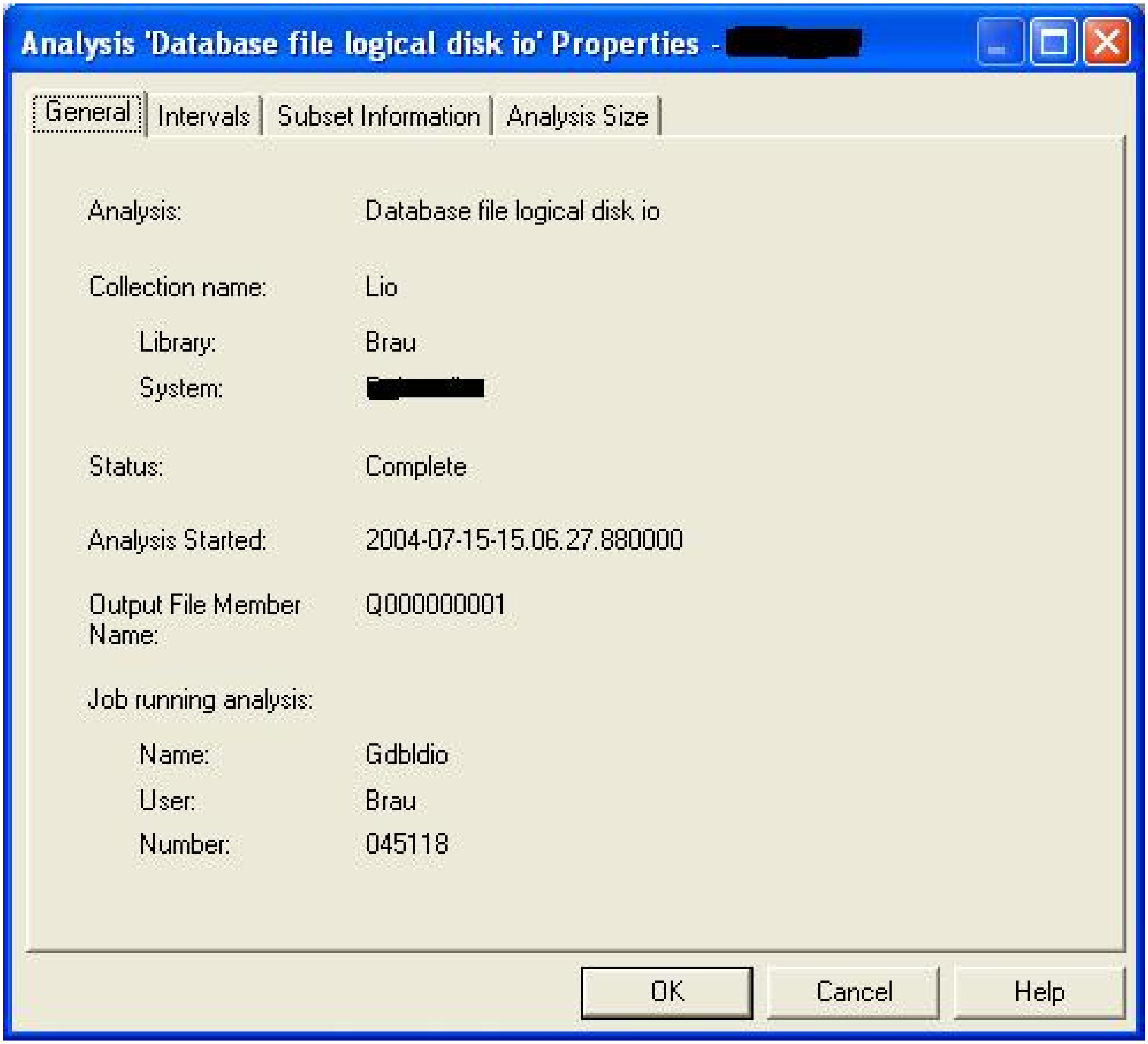
Table of Contents | Previous | Next

4.5.4 Properties

An analysis is a subset of performance data within a collection. There are several property pages for analyses which are defined in this section. A user may invoke the property pages by right-clicking on the desired analysis and choosing Properties.

4.5.4.1 General

The General property page provides a summary of the most basic information about an analysis.



Analysis 'Database file logical disk io' Properties - [redacted]

Analysis: Database file logical disk io

Collection name: Lio

Library: Brau

System: [redacted]

Status: Complete

Analysis Started: 2004-07-15-15.06.27.880000

Output File Member Name: Q000000001

Job running analysis:

Name: Gdbldio

User: Brau

Number: 045118

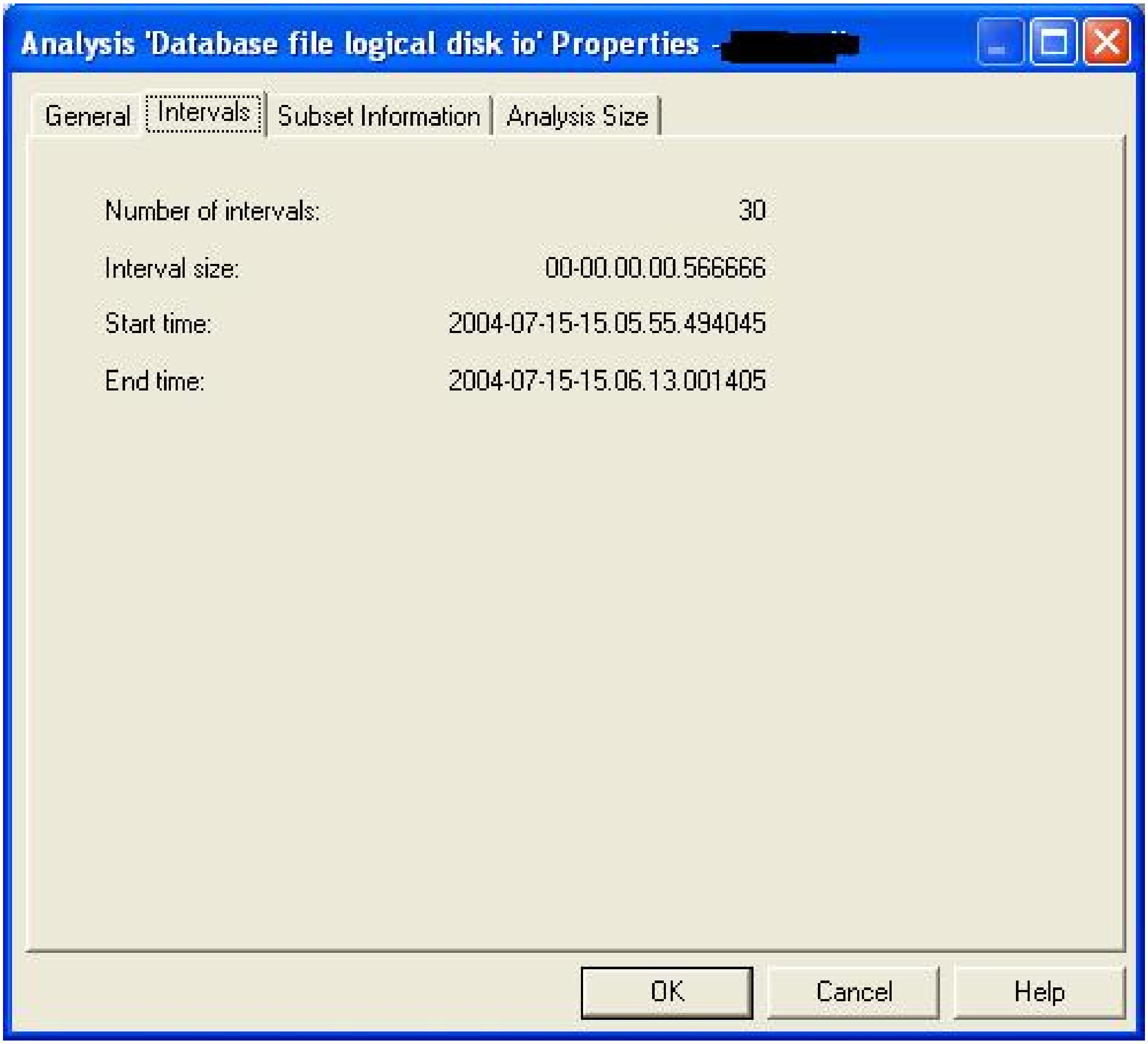
The following information is displayed on the General property page:

Field Name	Field Description

Analysis	The name of the PEX analysis. An analysis is a consolidation of data in a more useable format to help identify a specific problem type. An analysis can contain from 1 to n different reports. Some of these reports are graphable but all of the reports are viewable as table.
Collection name	PEX collection name.
Library	The library containing the collection and this analysis.
System	The system that the collection was initially created on.
Status	Indicates whether the analysis is useable by iDoctor for iSeries. If the status is not 'Complete' then the analysis cannot be viewed in the data viewer.
Analysis Started	Date/time the analysis was started.
Output Database/File Member Name	This is the member name assigned to all reports for this analysis. Each analysis creates one or more G_* files in the collection library.
Show Job Log Button	If the analysis job is currently active, this button will be enabled. Click this button to display the active job log of the analysis creation job being executed on the server.
Job running analysis	The job name/user/number identifying the job currently processing (or that has finished processing) the analysis. If an error occurs in the analysis creation remote program call, you may look at the job log using the job information listed for error reporting purposes.

4.5.4.2 Intervals

The Intervals property page provides information about the total size of an interval in an analysis, the start and ending time of the analysis and information about the time the collection was suspended. If the collection was suspended the resulting analysis will have a gap in the data where information was not collected. The total suspended time is also provided on this page.



Analysis 'Database file logical disk io' Properties - [redacted]

Number of intervals:	30
Interval size:	00-00.00.00.566666
Start time:	2004-07-15-15.05.55.494045
End time:	2004-07-15-15.06.13.001405

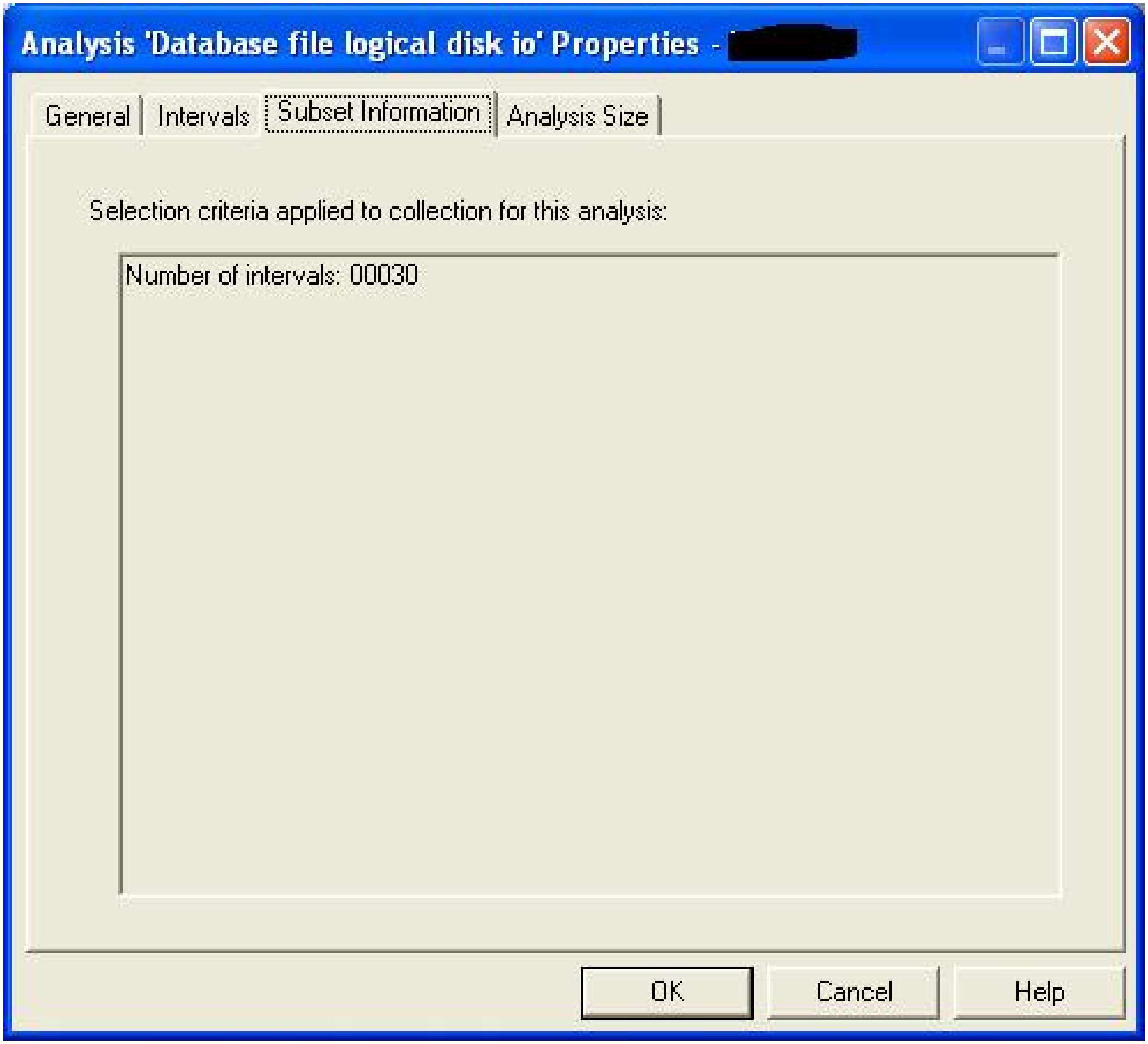
The following fields are available on this page:

Field Name	Field Description
Number of intervals	The total number of time intervals the analysis data has been broken up into.
Inteval size	The size of each time interval.
Start time	The start time for the data in the current analysis. This is either the subsetted start time or the collection start time.
End time	The end time for the data in the current analysis. This is either the subsetted end time or the collection end time.
Suspended time size	The total number of seconds or microseconds where the collecting of data did not occur.
Suspend start time	The time that the collection stopped collecting data.

Table of Contents | Previous | Next

4.5.4.3 Subset Information

The Subset Information property page provides detailed information about how an analysis was defined.



Analysis 'Database file logical disk io' Properties - [redacted]

Selection criteria applied to collection for this analysis:

Number of intervals: 00030

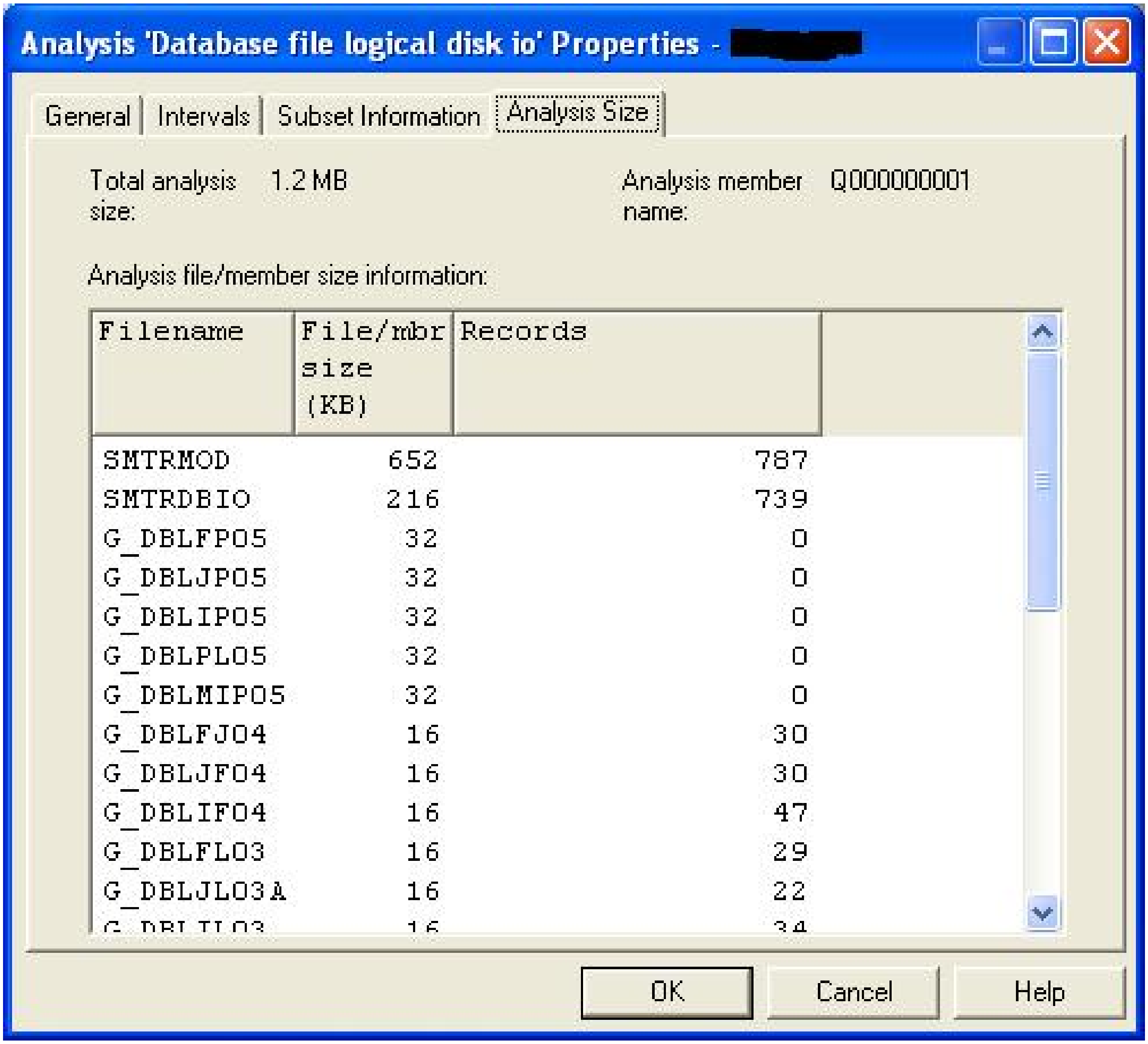
The following information is displayed on the Subset Information property page:

Field Name	Field Description
List of selection criteria	This is a list of all selection criteria that took place for the analysis. It represents how the data in the collection was broken down to produce the analysis data.

Table of Contents | Previous | Next

4.5.4.4 Analysis Size

The Analysis Size property page provides a breakdown of the size of the current analysis in total and for each file that makes up the analysis.



Analysis 'Database file logical disk io' Properties - [redacted]

Total analysis size: 1.2 MB

Analysis member name: Q000000001

Analysis file/member size information:

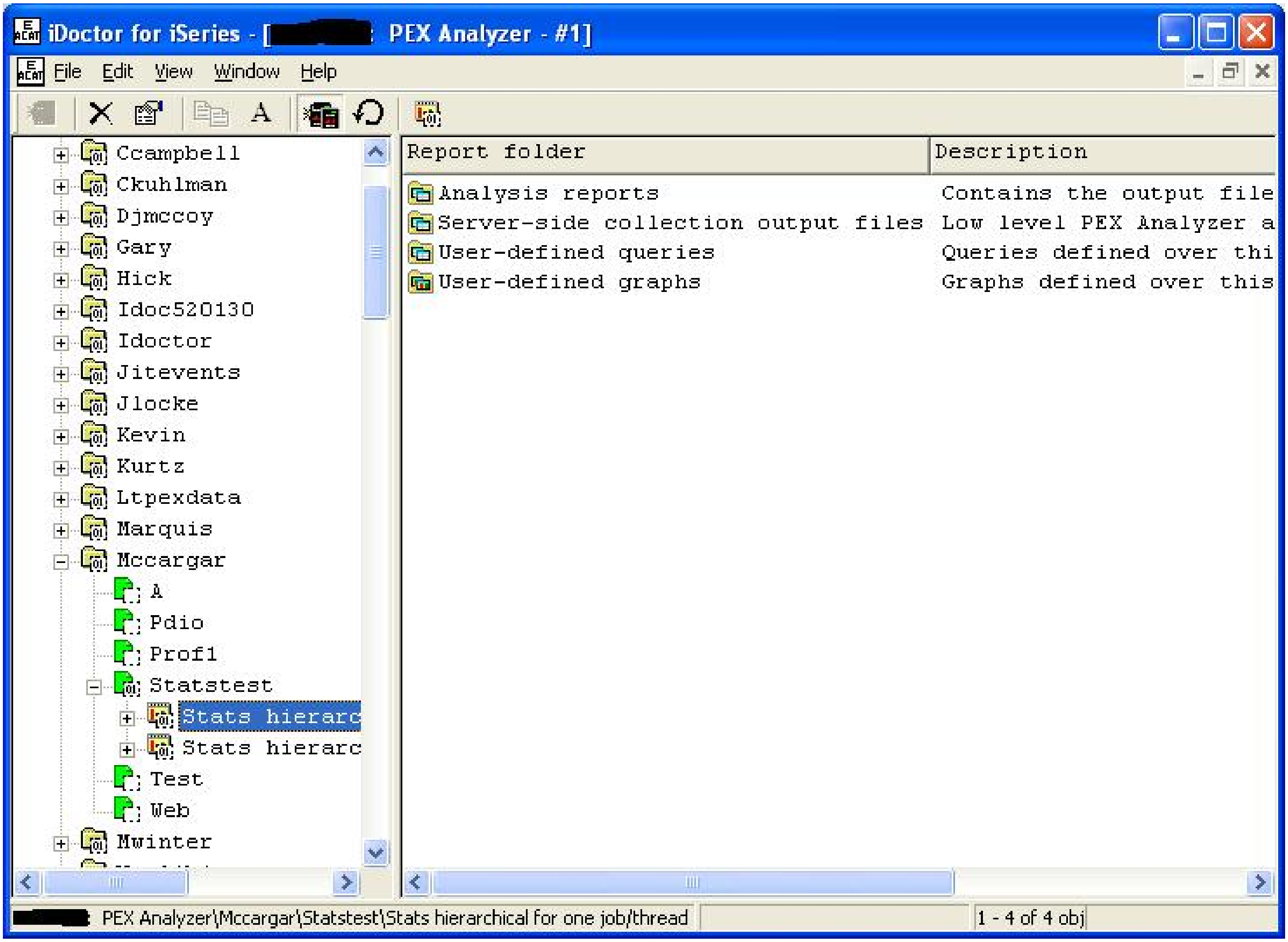
Filename	File/mbr size (KB)	Records
SMTRMOD	652	787
SMTRDBIO	216	739
G_DBLFP05	32	0
G_DBLJP05	32	0
G_DBLIP05	32	0
G_DBLPL05	32	0
G_DBLMIP05	32	0
G_DBLFJ04	16	30
G_DBLJF04	16	30
G_DBLIF04	16	47
G_DBLFL03	16	29
G_DBLJL03A	16	22
G_DBLIL03	16	34

Table of Contents | Previous | Next

4.6 Report Folders

Every PEX Analyzer analysis contains the following report folders:

Analysis reports - Provides a list of the output files generated by this analysis.
Server-side collection ouptut files - A list of the PEX collection files that are associated with this analysis. These are the PEX files that the analysis reports data was generated from.
User-defined queries/User-defined graphs - User created queries and graphs created over one or more of the files in the other folders.



Report folder	Description
Analysis reports	Contains the output file
Server-side collection output files	Low level PEX Analyzer a
User-defined queries	Queries defined over thi
User-defined graphs	Graphs defined over this

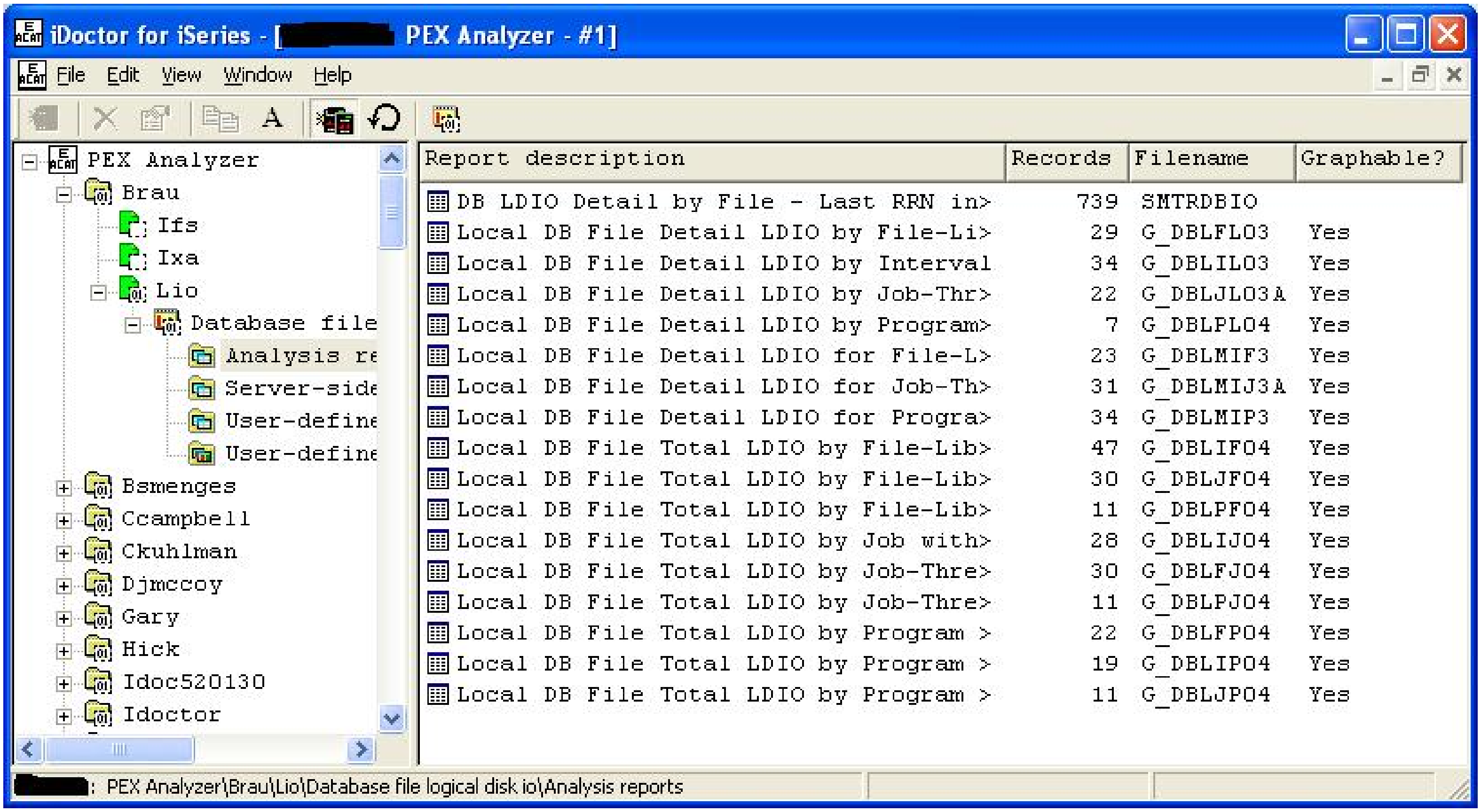
[redacted] PEX Analyzer\Mccargar\Statstest\Stats hierarchical for one job/thread

1 - 4 of 4 obj

4.7 Analysis Reports

The analysis reports folder contains the list of output files generated by the PEX Analysis. Each of these files may be opened in the Data Viewer as a table view and some of these files may be opened as a graph in the Data Viewer depending on the graphable flag shown in the list of reports.

An example of the contents of this folder is the following:



Report description	Records	Filename	Graphable?
DB LDIO Detail by File - Last RRN in>	739	SMTRDBIO	
Local DB File Detail LDIO by File-Li>	29	G_DBLFL03	Yes
Local DB File Detail LDIO by Interval	34	G_DBLIL03	Yes
Local DB File Detail LDIO by Job-Thr>	22	G_DBLJL03A	Yes
Local DB File Detail LDIO by Program>	7	G_DBLPL04	Yes
Local DB File Detail LDIO for File-L>	23	G_DBLMIF3	Yes
Local DB File Detail LDIO for Job-Th>	31	G_DBLMIJ3A	Yes
Local DB File Detail LDIO for Progra>	34	G_DBLMIP3	Yes
Local DB File Total LDIO by File-Lib>	47	G_DBLIF04	Yes
Local DB File Total LDIO by File-Lib>	30	G_DBLJF04	Yes
Local DB File Total LDIO by File-Lib>	11	G_DBLPF04	Yes
Local DB File Total LDIO by Job with>	28	G_DBLIJ04	Yes
Local DB File Total LDIO by Job-Thre>	30	G_DBLFJ04	Yes
Local DB File Total LDIO by Job-Thre>	11	G_DBLPJ04	Yes
Local DB File Total LDIO by Program >	22	G_DBLFP04	Yes
Local DB File Total LDIO by Program >	19	G_DBLIP04	Yes
Local DB File Total LDIO by Program >	11	G_DBLJP04	Yes

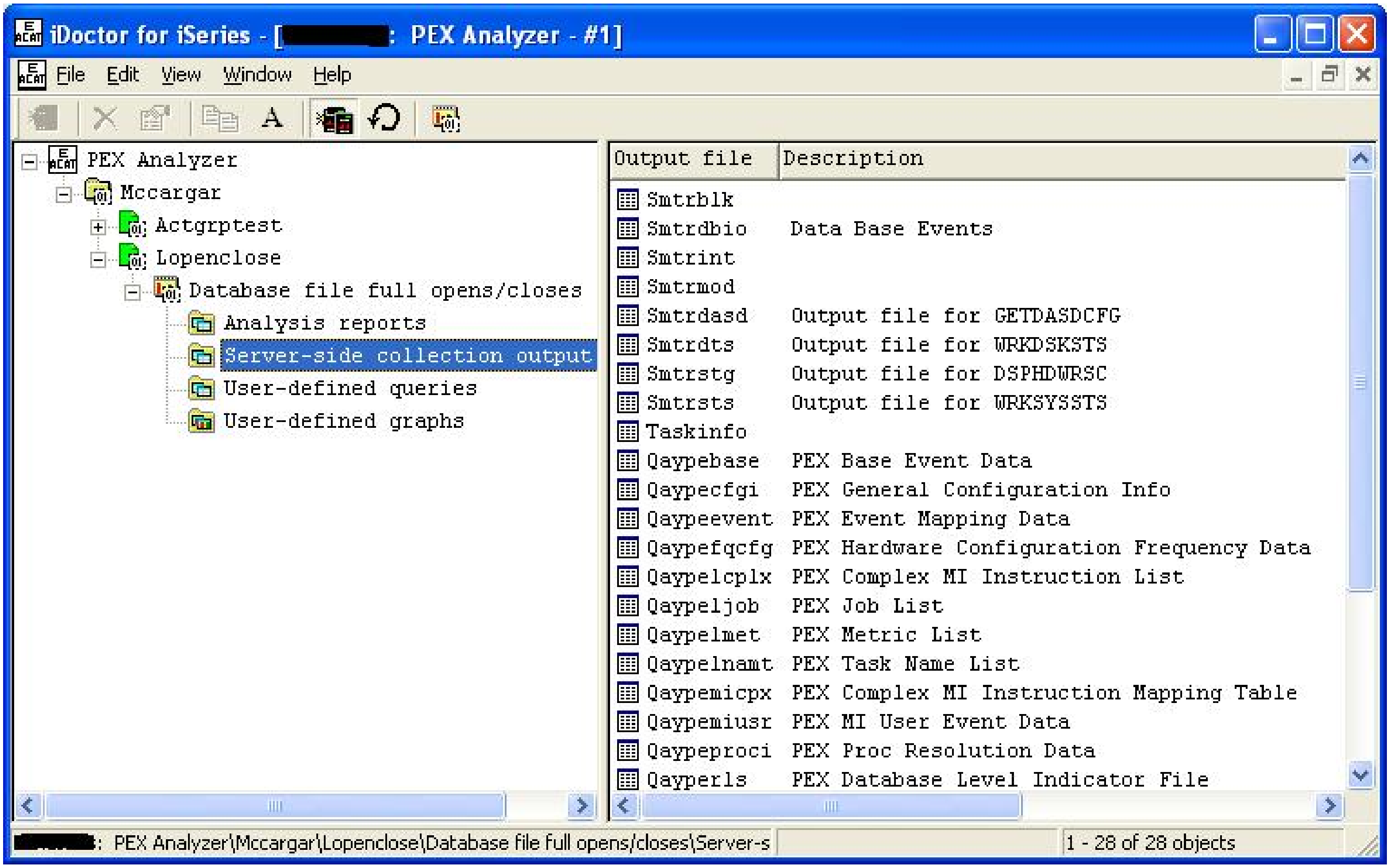
[redacted]: PEX Analyzer\Brau\Lio\Database file logical disk io\Analysis reports

Table of Contents | Previous | Next

4.8 Server-side collection output files

The analysis reports folder contains the list of output files generated by the PEX Analysis. Each of these files may be opened in the Data Viewer as a table view and some of these files may be opened as a graph in the Data Viewer depending on the graphable flag shown in the list of reports.

An example of the contents of this folder is the following:



Output file	Description
Smtrblk	
Smtrdbio	Data Base Events
Smtrint	
Smtrmod	
Smtrdasd	Output file for GETDASDCFG
Smtrdts	Output file for WRKDSKSTS
Smtrstg	Output file for DSPHDWRSC
Smtrsts	Output file for WRKSYSSTS
Taskinfo	
Qaypebase	PEX Base Event Data
Qaypecfgi	PEX General Configuration Info
Qaypeevent	PEX Event Mapping Data
Qaypefqcfg	PEX Hardware Configuration Frequency Data
Qaypelcplx	PEX Complex MI Instruction List
Qaypeljob	PEX Job List
Qaypelmet	PEX Metric List
Qaypelnamt	PEX Task Name List
Qaypemicpx	PEX Complex MI Instruction Mapping Table
Qaypemiusr	PEX MI User Event Data
Qaypeproci	PEX Proc Resolution Data
Qayperls	PEX Database Level Indicator File

[redacted]: PEX Analyzer\Mccargar\Lopenclose\Database file full opens/closes\Server-s

1 - 28 of 28 objects

Table of Contents | Previous | Next

Chapter 5 PEX Analyzer analysis reports

This chapter provides details about the analysis reports provided with PEX Analyzer.

Table of Contents Previous Next

5.1 Database file logical disk IO reports

This analysis shows all database Gets, Puts, Updates, Deletes at the file level showing the causing job/thread, file/member name, I/O type details (type of operation and relative record number), block sizes, file format name, timestamp and exception id.

If MIENTRY/MIEXIT events are included in the collection the analysis also shows the causing application program, elapsed time, physical disk I/O counts and CPU time of each LDIO in DB programs.

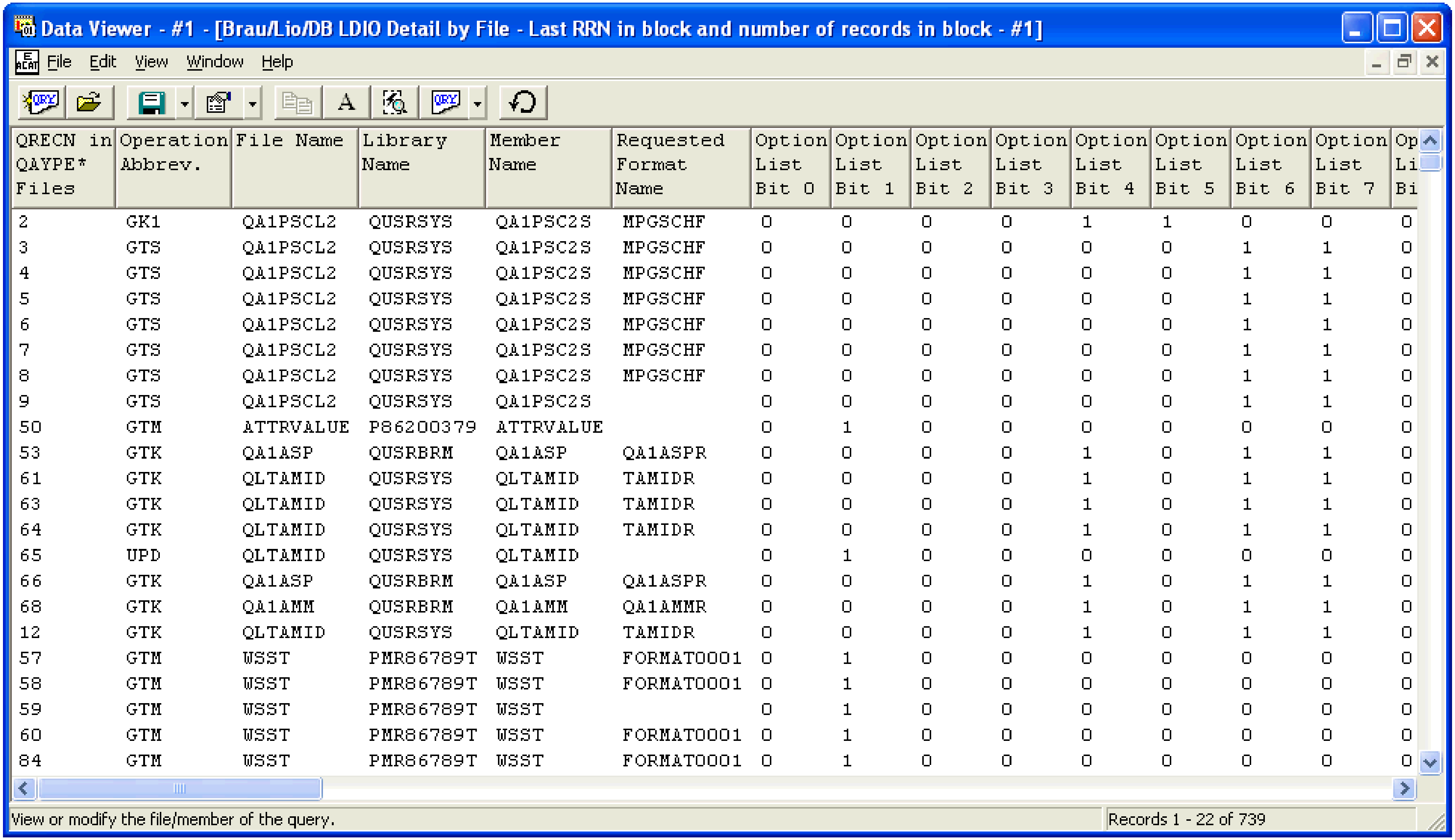
This section describes each of the reports available in this analysis.

Table of Contents | Previous | Next

5.1.1 DB LDIO Detail by File - Last RRN in block and number of records in block

Description: This report displays the LDIO related events captured in detail for each library/file/member occurring within the collection.

Example:


Data Viewer - #1 - [Brau/Lio/DB LDIO Detail by File - Last RRN in block and number of records in block - #1]
File
Edit
View
Window
Help

QRECN in QAYPE* Files	Operation Abbrev.	File Name	Library Name	Member Name	Requested Format Name	Option List Bit 0	Option List Bit 1	Option List Bit 2	Option List Bit 3	Option List Bit 4	Option List Bit 5	Option List Bit 6	Option List Bit 7	Op Li Bi
2	GK1	QA1PSCL2	QUSRSYS	QA1PSC2S	MPGSCHF	0	0	0	0	1	1	0	0	0
3	GTS	QA1PSCL2	QUSRSYS	QA1PSC2S	MPGSCHF	0	0	0	0	0	0	1	1	0
4	GTS	QA1PSCL2	QUSRSYS	QA1PSC2S	MPGSCHF	0	0	0	0	0	0	1	1	0
5	GTS	QA1PSCL2	QUSRSYS	QA1PSC2S	MPGSCHF	0	0	0	0	0	0	1	1	0
6	GTS	QA1PSCL2	QUSRSYS	QA1PSC2S	MPGSCHF	0	0	0	0	0	0	1	1	0
7	GTS	QA1PSCL2	QUSRSYS	QA1PSC2S	MPGSCHF	0	0	0	0	0	0	1	1	0
8	GTS	QA1PSCL2	QUSRSYS	QA1PSC2S	MPGSCHF	0	0	0	0	0	0	1	1	0
9	GTS	QA1PSCL2	QUSRSYS	QA1PSC2S		0	0	0	0	0	0	1	1	0
50	GTM	ATTRVALUE	P86200379	ATTRVALUE		0	1	0	0	0	0	0	0	0
53	GTK	QA1ASP	QUSRBRM	QA1ASP	QA1ASPR	0	0	0	0	1	0	1	1	0
61	GTK	QLTAMID	QUSRSYS	QLTAMID	TAMIDR	0	0	0	0	1	0	1	1	0
63	GTK	QLTAMID	QUSRSYS	QLTAMID	TAMIDR	0	0	0	0	1	0	1	1	0
64	GTK	QLTAMID	QUSRSYS	QLTAMID	TAMIDR	0	0	0	0	1	0	1	1	0
65	UPD	QLTAMID	QUSRSYS	QLTAMID		0	1	0	0	0	0	0	0	0
66	GTK	QA1ASP	QUSRBRM	QA1ASP	QA1ASPR	0	0	0	0	1	0	1	1	0
68	GTK	QA1AMM	QUSRBRM	QA1AMM	QA1AMMR	0	0	0	0	1	0	1	1	0
12	GTK	QLTAMID	QUSRSYS	QLTAMID	TAMIDR	0	0	0	0	1	0	1	1	0
57	GTM	WSST	PMR86789T	WSST	FORMAT0001	0	1	0	0	0	0	0	0	0
58	GTM	WSST	PMR86789T	WSST	FORMAT0001	0	1	0	0	0	0	0	0	0
59	GTM	WSST	PMR86789T	WSST		0	1	0	0	0	0	0	0	0
60	GTM	WSST	PMR86789T	WSST	FORMAT0001	0	1	0	0	0	0	0	0	0
84	GTM	WSST	PMR86789T	WSST	FORMAT0001	0	1	0	0	0	0	0	0	0

View or modify the file/member of the query.

Records 1 - 22 of 739

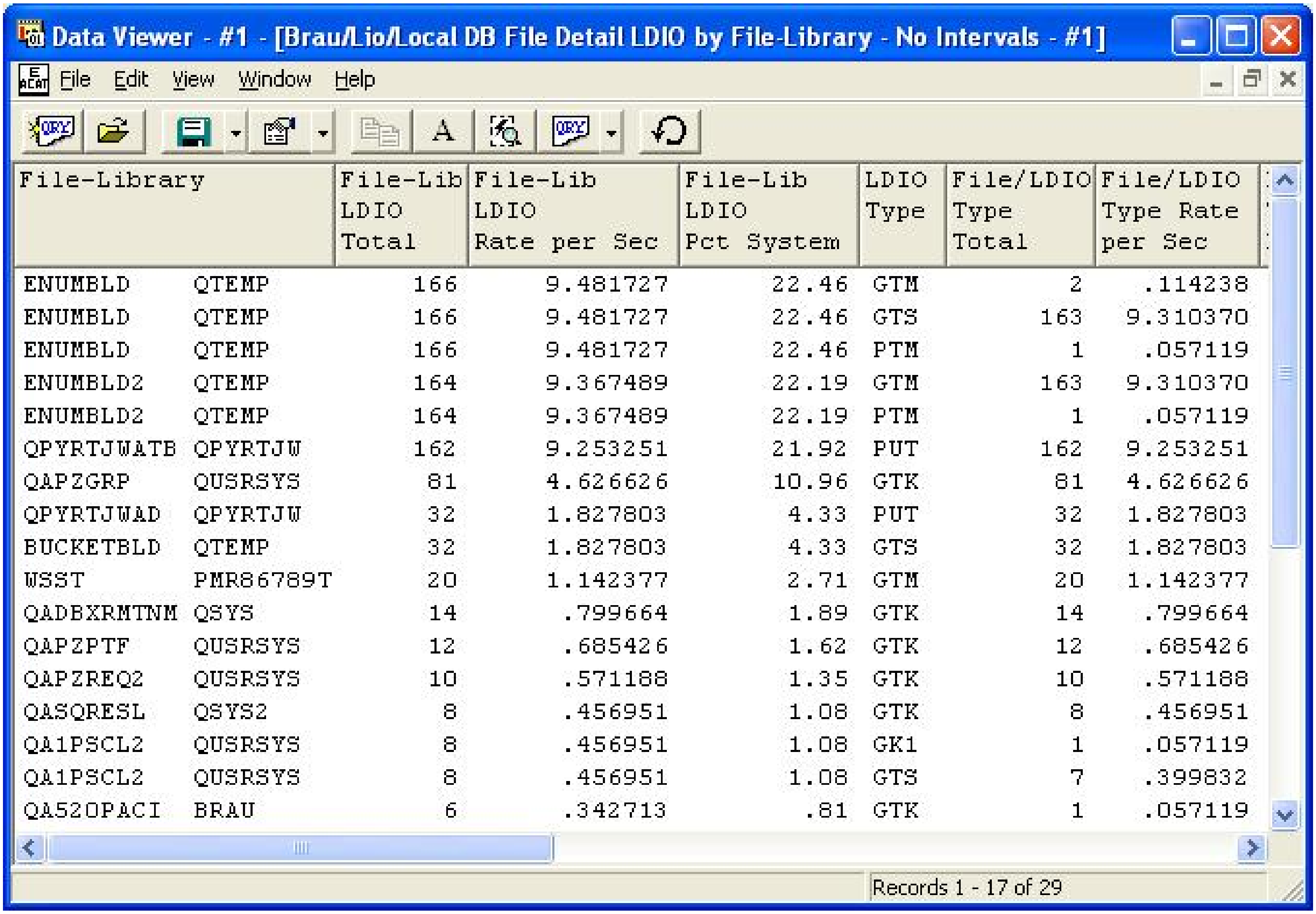
Table of Contents Previous Next

5.1.2 DB File Detail LDIO by File-Library - No Intervals

Description: This report shows the files with the most "logical disk I/O" (LDIO) activity during the collection. For each of these files, it also shows the type of LDIO and what percentage of the total collection's LDIO each file represents.

Each color in a bar graph represents a type of LDIO. Clicking one of these colors brings up additional details about the LDIO, including the full file name and the amount of physical disk I/O (PDIO) that occurred on behalf of this LDIO.

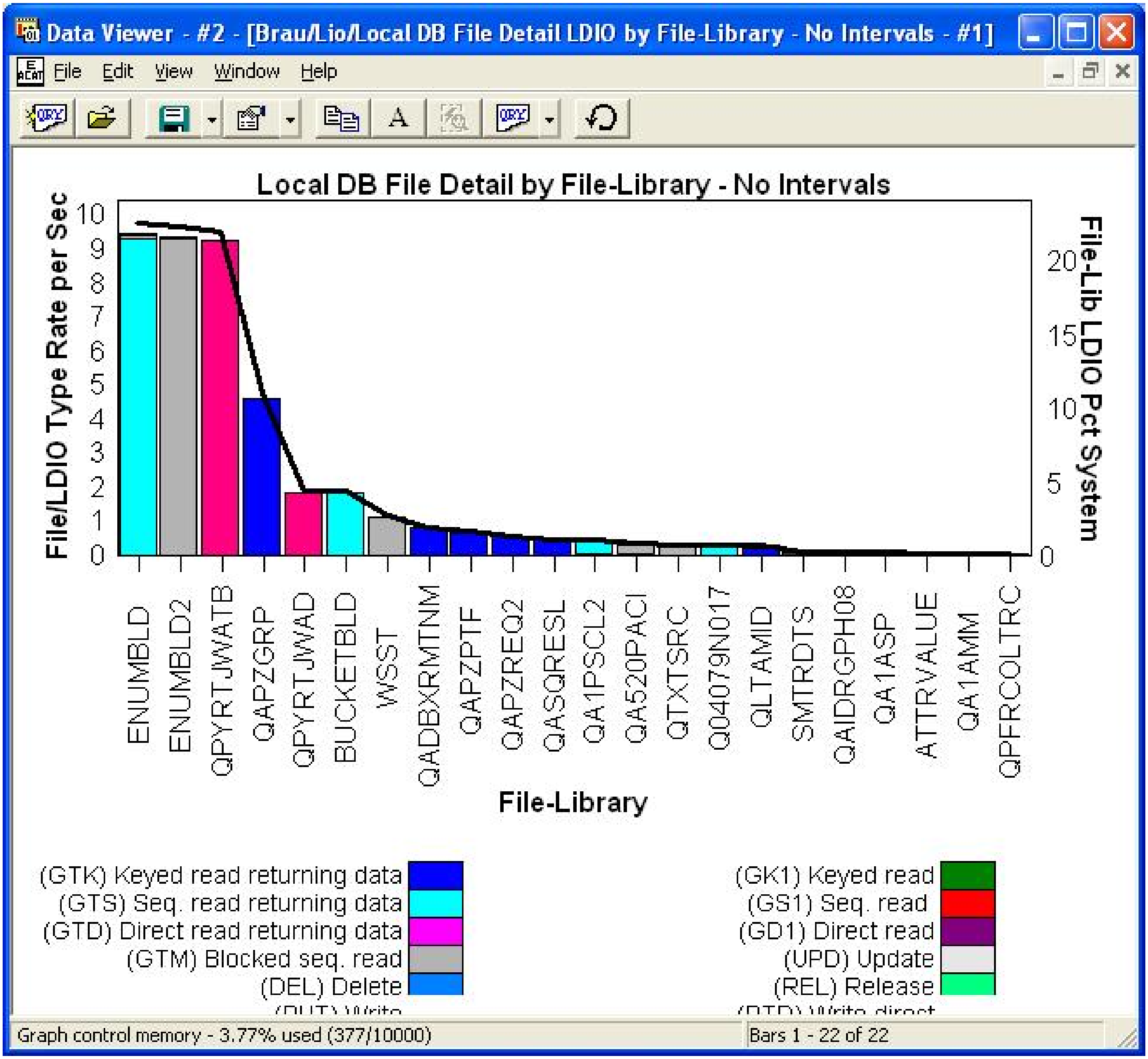
Example:


Data Viewer - #1 - [Brau/Lio/Local DB File Detail LDIO by File-Library - No Intervals - #1]
File
Edit
View
Window
Help
QRY
A
QRY
Records 1 - 17 of 29

File-Library		File-Lib LDIO Total	File-Lib LDIO Rate per Sec	File-Lib LDIO Pct System	LDIO Type	File/LDIO Type Total	File/LDIO Type Rate per Sec
ENUMBLD	QTEMP	166	9.481727	22.46	GTM	2	.114238
ENUMBLD	QTEMP	166	9.481727	22.46	GTS	163	9.310370
ENUMBLD	QTEMP	166	9.481727	22.46	PTM	1	.057119
ENUMBLD2	QTEMP	164	9.367489	22.19	GTM	163	9.310370
ENUMBLD2	QTEMP	164	9.367489	22.19	PTM	1	.057119
QPYRTJWATB	QPYRTJW	162	9.253251	21.92	PUT	162	9.253251
QAPZGRP	QUSRSYS	81	4.626626	10.96	GTK	81	4.626626
QPYRTJWAD	QPYRTJW	32	1.827803	4.33	PUT	32	1.827803
BUCKETBLD	QTEMP	32	1.827803	4.33	GTS	32	1.827803
WSST	PMR86789T	20	1.142377	2.71	GTM	20	1.142377
QADBXRMTNM	QSYS	14	.799664	1.89	GTK	14	.799664
QAPZPTF	QUSRSYS	12	.685426	1.62	GTK	12	.685426
QAPZREQ2	QUSRSYS	10	.571188	1.35	GTK	10	.571188
QASQRESL	QSYS2	8	.456951	1.08	GTK	8	.456951
QA1PSCL2	QUSRSYS	8	.456951	1.08	GK1	1	.057119
QA1PSCL2	QUSRSYS	8	.456951	1.08	GTS	7	.399832
QA52OPACI	BRAU	6	.342713	.81	GTK	1	.057119

Graph Type: stacked vertical bar
X-axis: File and library name. The entire field is shown by placing the mouse over a bar.
Y-Axis: This value is the number of LDIO ops occurring of each type per second for each file. Each color represents a different type of LDIO.
Second Y-Axis: This line shows how much each file/library's LDIO ops contributed to the total system LDIOs.

Example:



Data Viewer - #2 - [Brau/Lio/Local DB File Detail LDIO by File-Library - No Intervals - #1]

Graph control memory - 3.77% used (377/10000)

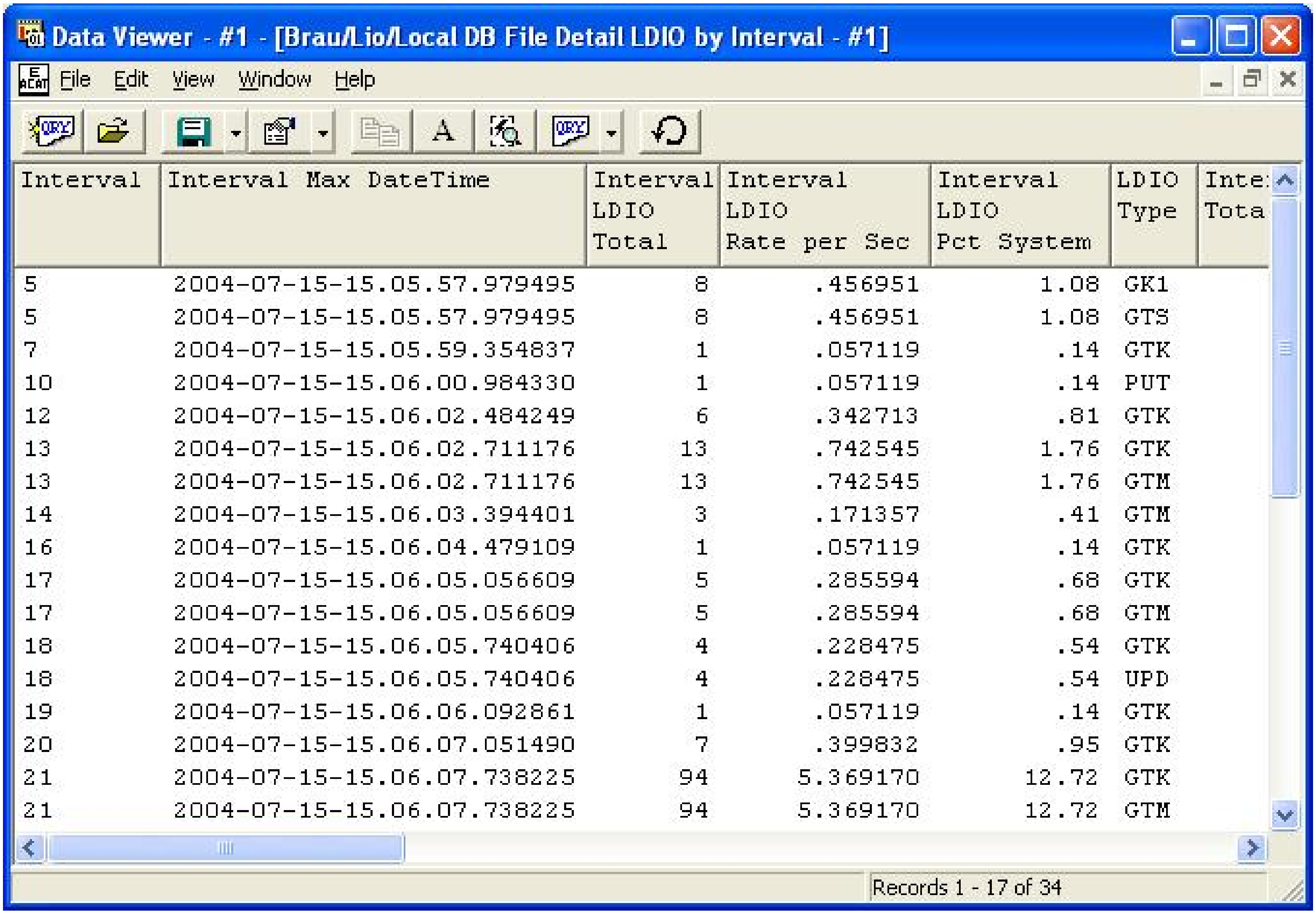
Bars 1 - 22 of 22

Table of Contents Previous Next

5.1.3 DB File Detail LDIO by Interval

Description: This report shows the amount and type of "logical disk I/O" (LDIO) activity for each time interval in the collection. The percentage each interval's LDIO represents of the collection's total LDIO is also shown. Each color in a bar represents a type of LDIO. Clicking one of these colors brings up additional details about the LDIO for the interval selected.

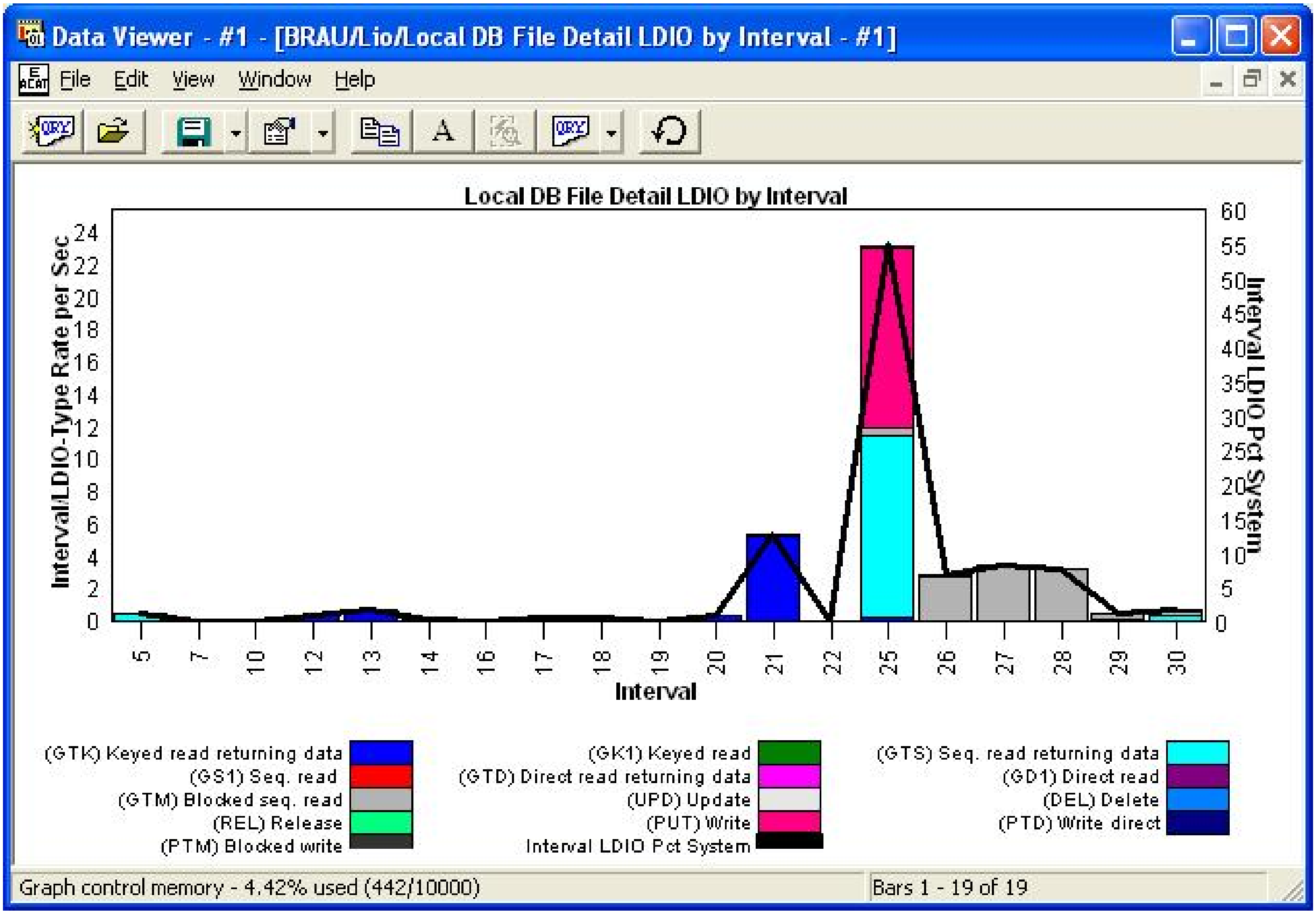
Example:


Data Viewer - #1 - [Brau/Lio/Local DB File Detail LDIO by Interval - #1]
File
Edit
View
Window
Help

Interval	Interval Max DateTime	Interval LDIO Total	Interval LDIO Rate per Sec	Interval LDIO Pct System	LDIO Type	Inte: Tota
5	2004-07-15-15.05.57.979495	8	.456951	1.08	GK1	
5	2004-07-15-15.05.57.979495	8	.456951	1.08	GTS	
7	2004-07-15-15.05.59.354837	1	.057119	.14	GTK	
10	2004-07-15-15.06.00.984330	1	.057119	.14	PUT	
12	2004-07-15-15.06.02.484249	6	.342713	.81	GTK	
13	2004-07-15-15.06.02.711176	13	.742545	1.76	GTK	
13	2004-07-15-15.06.02.711176	13	.742545	1.76	GTM	
14	2004-07-15-15.06.03.394401	3	.171357	.41	GTM	
16	2004-07-15-15.06.04.479109	1	.057119	.14	GTK	
17	2004-07-15-15.06.05.056609	5	.285594	.68	GTK	
17	2004-07-15-15.06.05.056609	5	.285594	.68	GTM	
18	2004-07-15-15.06.05.740406	4	.228475	.54	GTK	
18	2004-07-15-15.06.05.740406	4	.228475	.54	UPD	
19	2004-07-15-15.06.06.092861	1	.057119	.14	GTK	
20	2004-07-15-15.06.07.051490	7	.399832	.95	GTK	
21	2004-07-15-15.06.07.738225	94	5.369170	12.72	GTK	
21	2004-07-15-15.06.07.738225	94	5.369170	12.72	GTM	

Records 1 - 17 of 34

Graph Type: stacked vertical bar
X-axis: Interval number
Y-Axis: This value is the number of LDIO ops occurring of each type per second within each interval. Each color represents a different type of LDIO.
Second Y-Axis: This line shows how much each interval's total LDIOs for the collection contributed to the total system LDIOs.



Graph control memory - 4.42% used (442/10000)

Bars 1 - 19 of 19

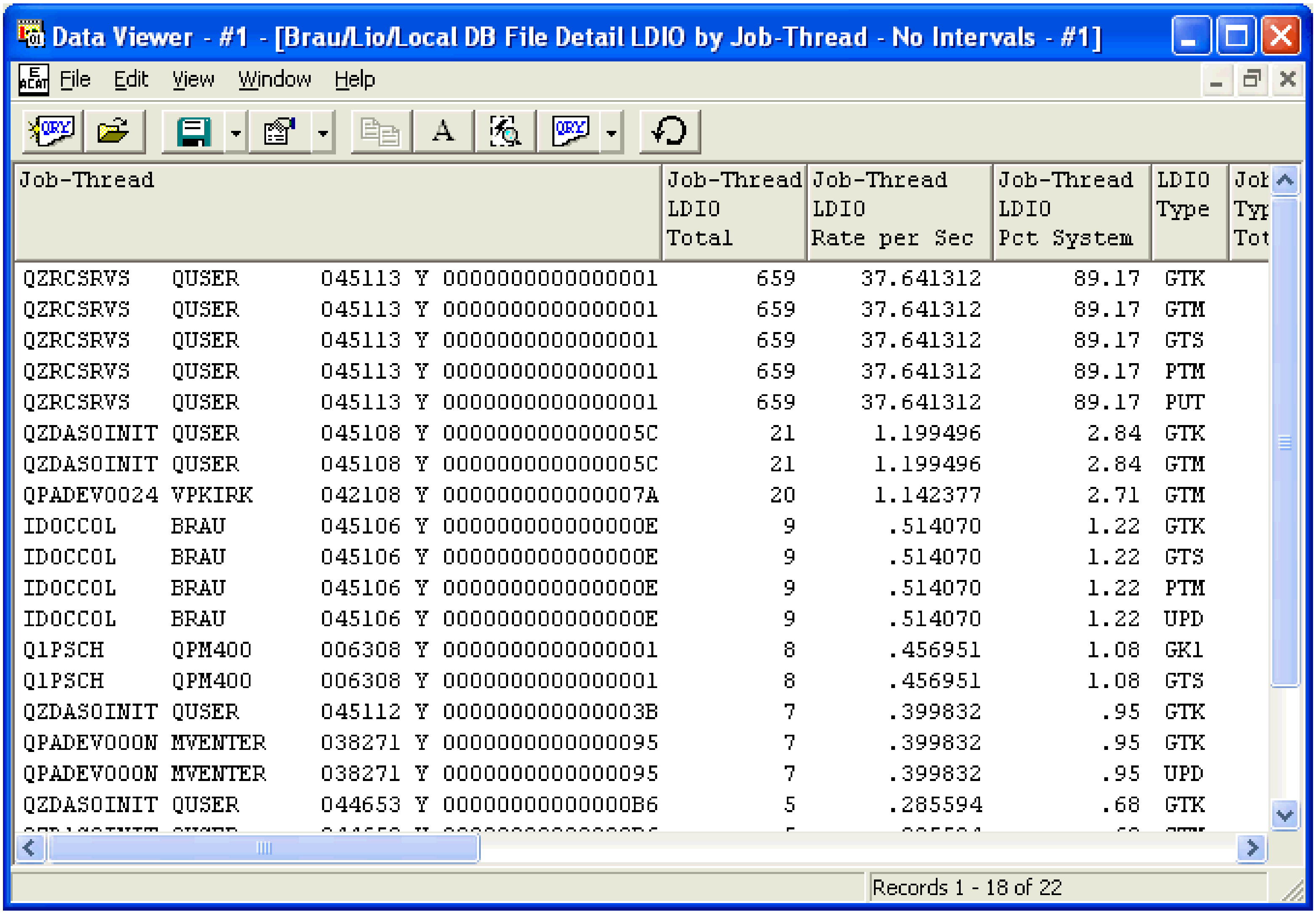
Table of Contents Previous Next

5.1.4 DB File Detail LDIO by Job-thread - No Intervals

Description: This graph shows the jobs or threads with the most "logical disk I/O" (LDIO) activity during the collection.

For each of these jobs or threads, it also shows the type of LDIO and what percentage of the total collection's LDIO each job/thread represents. Each color in a bar represents a type of LDIO. Clicking one of these colors brings up additional details about the LDIO for the job-thread selected.

Example:



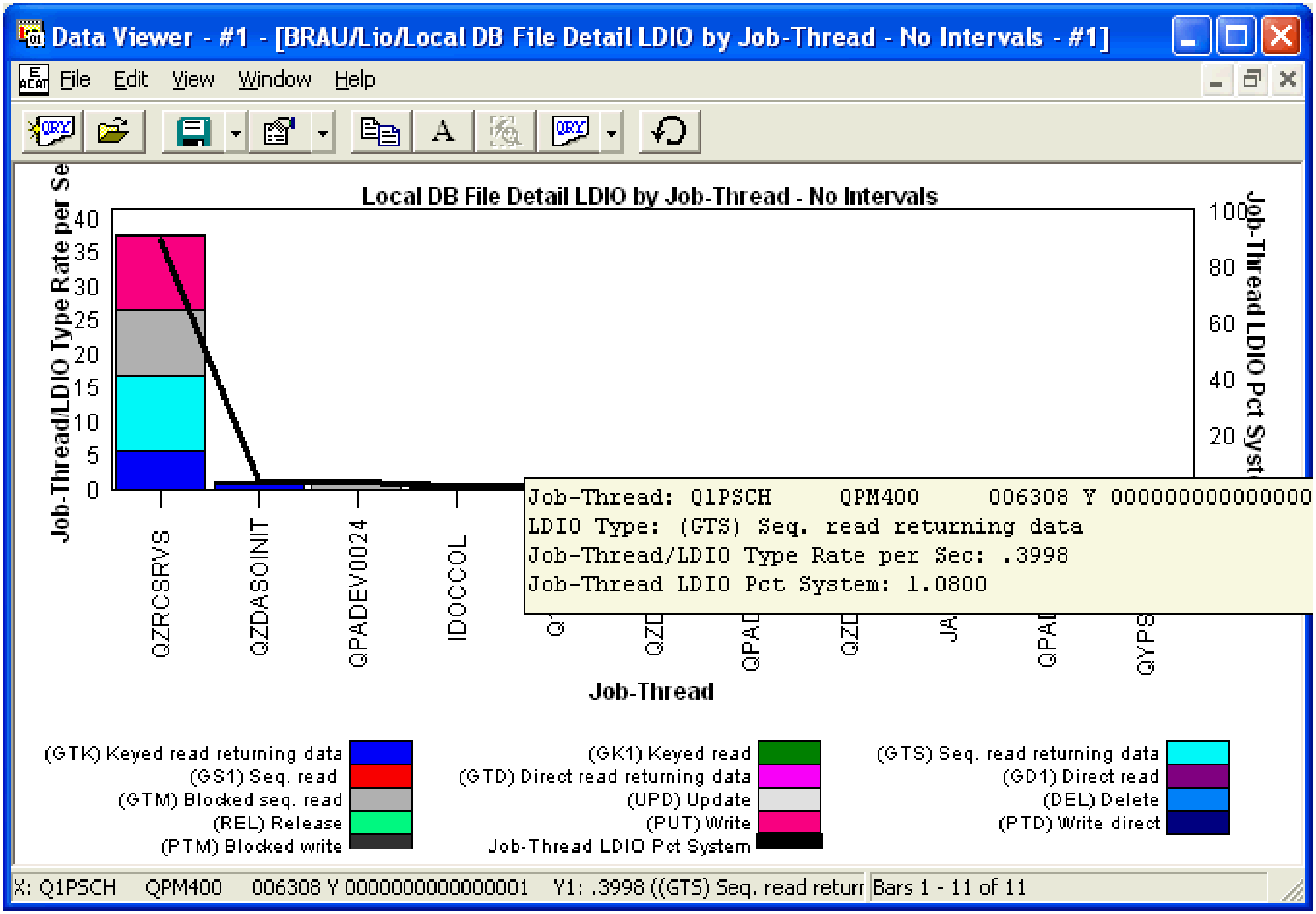
Data Viewer - #1 - [Brau/Lio/Local DB File Detail LDIO by Job-Thread - No Intervals - #1]
File
Edit
View
Window
Help

Job-Thread	Job-Thread LDIO Total	Job-Thread LDIO Rate per Sec	Job-Thread LDIO Pct System	LDIO Type	Jol Typ Tot
QZRCSRVS QUSER 045113 Y 0000000000000001	659	37.641312	89.17	GTK	
QZRCSRVS QUSER 045113 Y 0000000000000001	659	37.641312	89.17	GTM	
QZRCSRVS QUSER 045113 Y 0000000000000001	659	37.641312	89.17	GTS	
QZRCSRVS QUSER 045113 Y 0000000000000001	659	37.641312	89.17	PTM	
QZRCSRVS QUSER 045113 Y 0000000000000001	659	37.641312	89.17	PUT	
QZDASOINIT QUSER 045108 Y 000000000000005C	21	1.199496	2.84	GTK	
QZDASOINIT QUSER 045108 Y 000000000000005C	21	1.199496	2.84	GTM	
QPADEV0024 VPKIRK 042108 Y 000000000000007A	20	1.142377	2.71	GTM	
IDOCCOL BRAU 045106 Y 000000000000000E	9	.514070	1.22	GTK	
IDOCCOL BRAU 045106 Y 000000000000000E	9	.514070	1.22	GTS	
IDOCCOL BRAU 045106 Y 000000000000000E	9	.514070	1.22	PTM	
IDOCCOL BRAU 045106 Y 000000000000000E	9	.514070	1.22	UPD	
Q1PSCH QPM400 006308 Y 0000000000000001	8	.456951	1.08	GK1	
Q1PSCH QPM400 006308 Y 0000000000000001	8	.456951	1.08	GTS	
QZDASOINIT QUSER 045112 Y 000000000000003B	7	.399832	.95	GTK	
QPADEV000N MVENTER 038271 Y 0000000000000095	7	.399832	.95	GTK	
QPADEV000N MVENTER 038271 Y 0000000000000095	7	.399832	.95	UPD	
QZDASOINIT QUSER 044653 Y 00000000000000B6	5	.285594	.68	GTK	

Records 1 - 18 of 22

Graph Type: stacked vertical bar
X-axis: Fully qualified job name and thread id. The entire field is shown by placing the mouse over a bar.
Y-Axis: This value is the number of LDIO ops occurring of each type per second for a job within the collection. Each color represents a different type of LDIO.
Second Y-Axis: This line shows how much each job's total LDIO ops contributed to the total system LDIOs.

Example:



```
Job-Thread: Q1PSCH      QPM400      006308 Y 0000000000000000
LDIO Type: (GTS) Seq. read returning data
Job-Thread/LDIO Type Rate per Sec: .3998
Job-Thread LDIO Pct System: 1.0800
```

X: Q1PSCH QPM400 006308 Y 0000000000000001 Y1: .3998 ((GTS) Seq. read returr | Bars 1 - 11 of 11

Table of Contents Previous Next

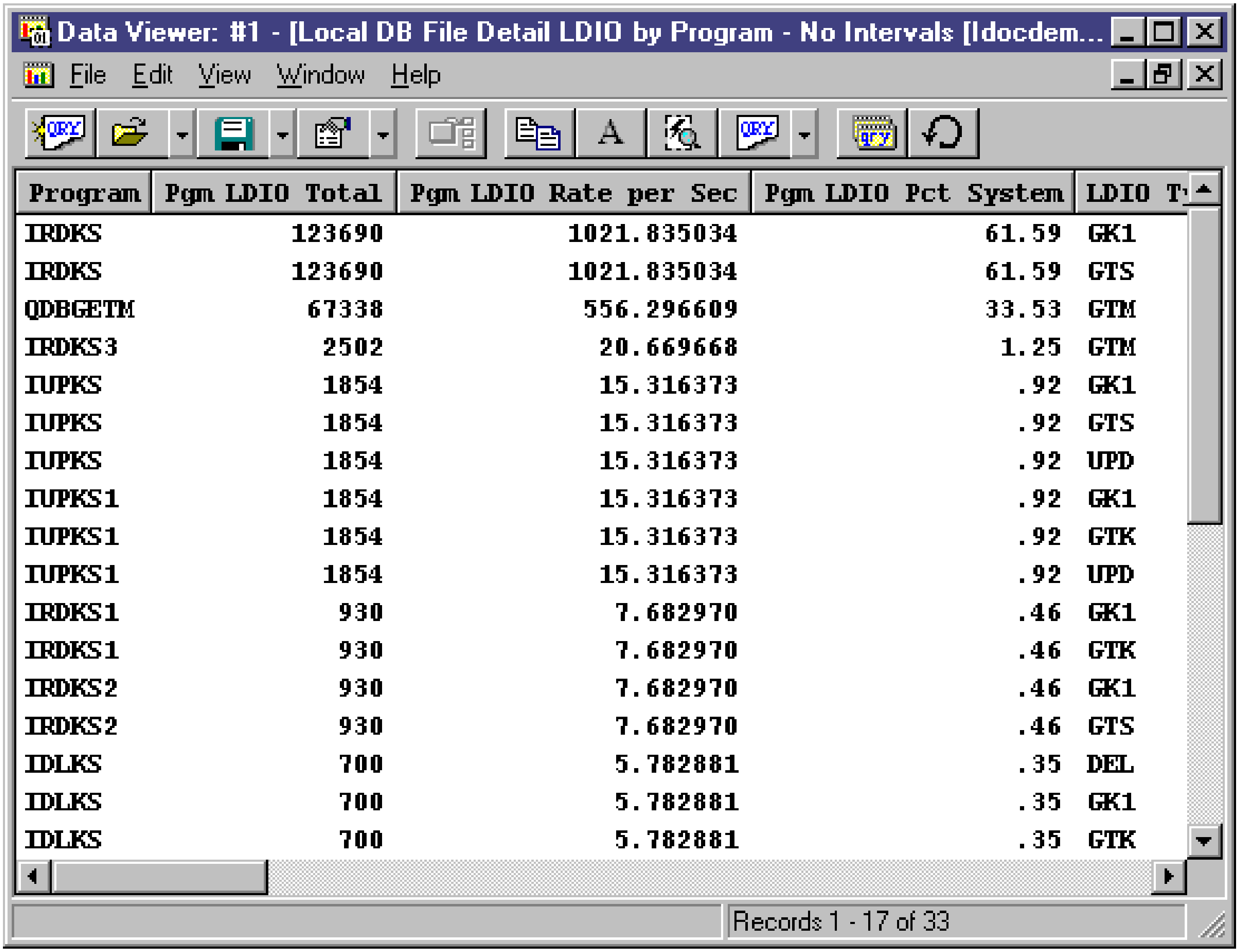
5.1.5 DB File Detail LDIO by Program - No Intervals

Description: This graph shows the programs with the most "logical disk I/O" (LDIO) activity during the collection.

For each of these programs, it also shows the type of LDIO and what percentage of the total collection's LDIO each program represents. Each color in a bar represents a type of LDIO. Clicking one of these colors brings up additional details about the LDIO for the program selected.

Note: In order for program names to be provided, *MIENTRY and *MIEXIT events must be included in the PEX collection.

Example:


Data Viewer: #1 - [Local DB File Detail LDIO by Program - No Intervals [Idocdem...
File Edit View Window Help

Program	Pgm LDIO Total	Pgm LDIO Rate per Sec	Pgm LDIO Pct System	LDIO T
IRDKS	123690	1021.835034	61.59	GK1
IRDKS	123690	1021.835034	61.59	GTS
QDBGETM	67338	556.296609	33.53	GTM
IRDKS3	2502	20.669668	1.25	GTM
IUPKS	1854	15.316373	.92	GK1
IUPKS	1854	15.316373	.92	GTS
IUPKS	1854	15.316373	.92	UPD
IUPKS1	1854	15.316373	.92	GK1
IUPKS1	1854	15.316373	.92	GTK
IUPKS1	1854	15.316373	.92	UPD
IRDKS1	930	7.682970	.46	GK1
IRDKS1	930	7.682970	.46	GTK
IRDKS2	930	7.682970	.46	GK1
IRDKS2	930	7.682970	.46	GTS
IDLKS	700	5.782881	.35	DEL
IDLKS	700	5.782881	.35	GK1
IDLKS	700	5.782881	.35	GTK

Records 1 - 17 of 33

Graph Type: stacked vertical bar
X-axis: This axis shows programs in the collection that performed LDIO operations.
Y-Axis: This value is the number of LDIO ops occurring of each type per second. Each color represents a different type of LDIO.
Second Y-Axis: This line shows how much each program's total LDIO contributed to the total system LDIO.

Example:

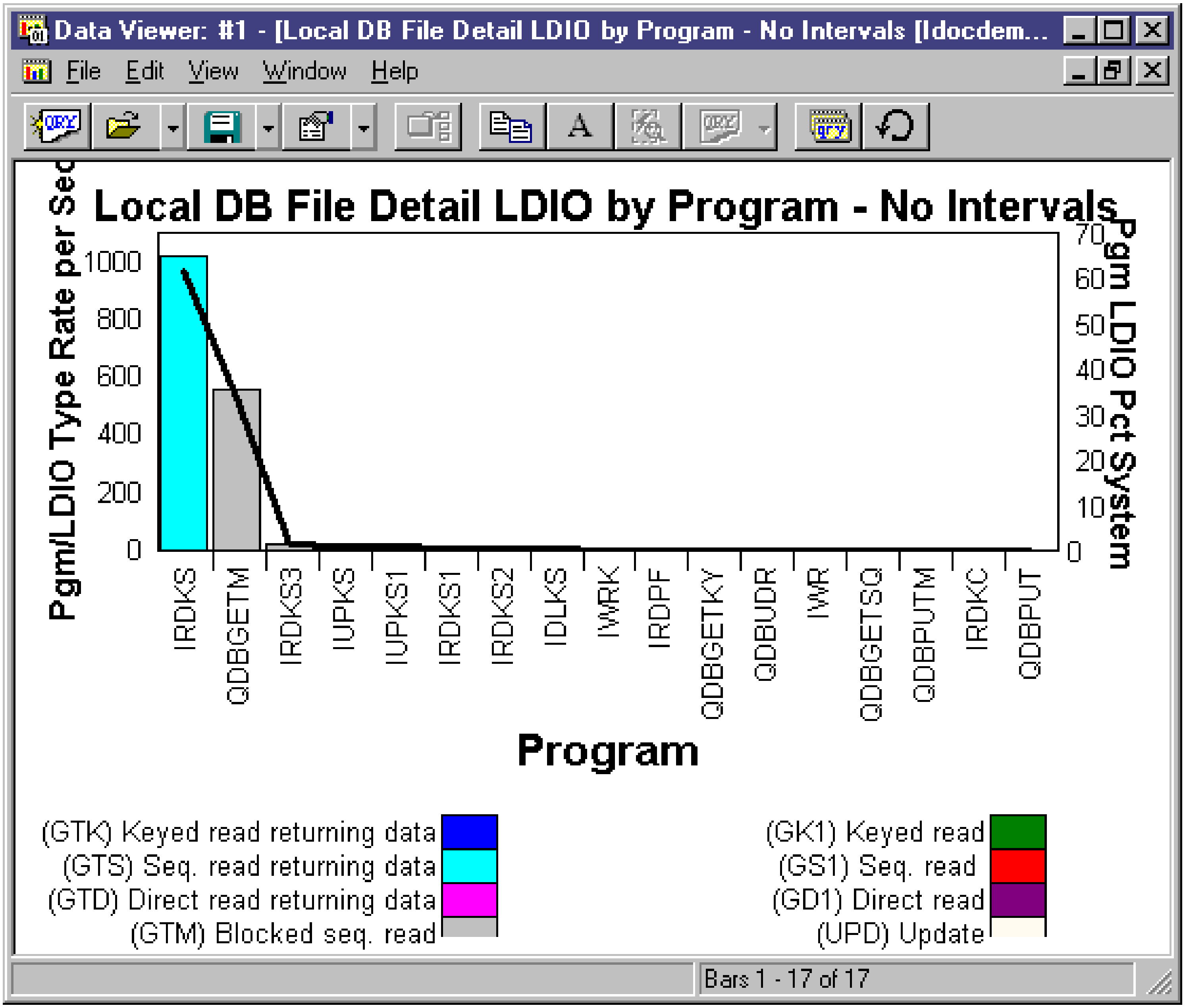
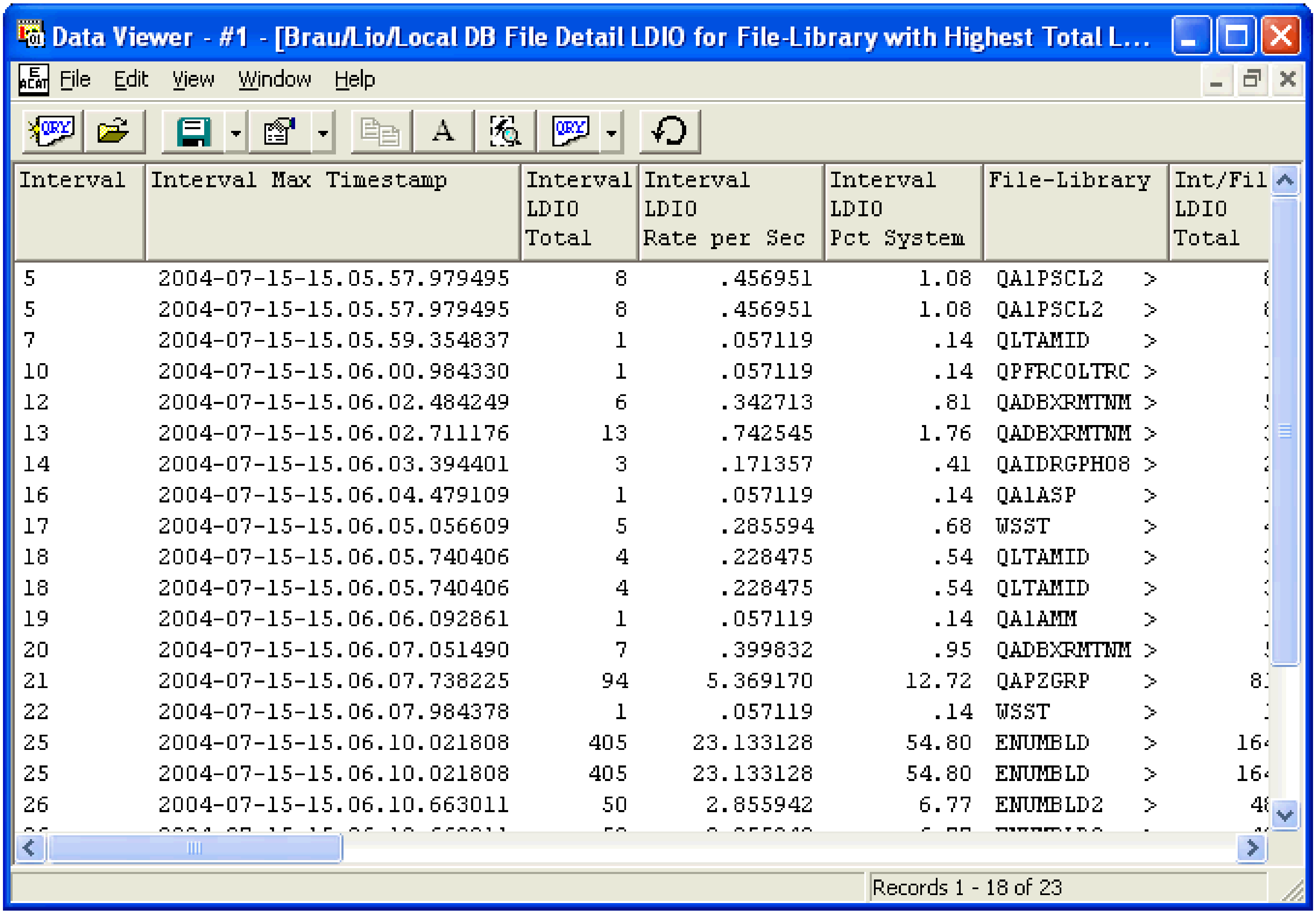

Data Viewer: #1 - [Local DB File Detail LDIO by Program - No Intervals [Idocdem...
File Edit View Window Help
Local DB File Detail LDIO by Program - No Intervals
Pgm/LDIO Type Rate per Sec
1000
800
600
400
200
0
Pgm LDIO Pct System
70
60
50
40
30
20
10
0
IRDKS
QDBGETM
IRDKS3
IUPKS
IUPKS1
IRDKS1
IRDKS2
IDLKS
IWRK
IRDPF
QDBGETKY
QDBUDR
IWR
QDBGETSQ
QDBPUTM
IRDKC
QDBPUT
Program
(GTK) Keyed read returning data
(GTS) Seq. read returning data
(GTD) Direct read returning data
(GTM) Blocked seq. read
(GK1) Keyed read
(GS1) Seq. read
(GD1) Direct read
(UPD) Update
Bars 1 - 17 of 17

Table of Contents | Previous | Next

5.1.6 DB File Detail LDIO for File-Library With Highest Total LDIO per Interval

Description: This report shows the file with the most "logical disk I/O" (LDIO) activity for each time interval in the collection. It also shows the type(s) of LDIO done to these files and the percentage of the collection's total LDIO. Each file's LDIO activity per interval is also shown.

Each color in a bar represents a type of LDIO. Clicking one of these colors brings up additional details about the LDIO for the file/interval selected.


Data Viewer - #1 - [Brau/Lio/Local DB File Detail LDIO for File-Library with Highest Total L...
File Edit View Window Help

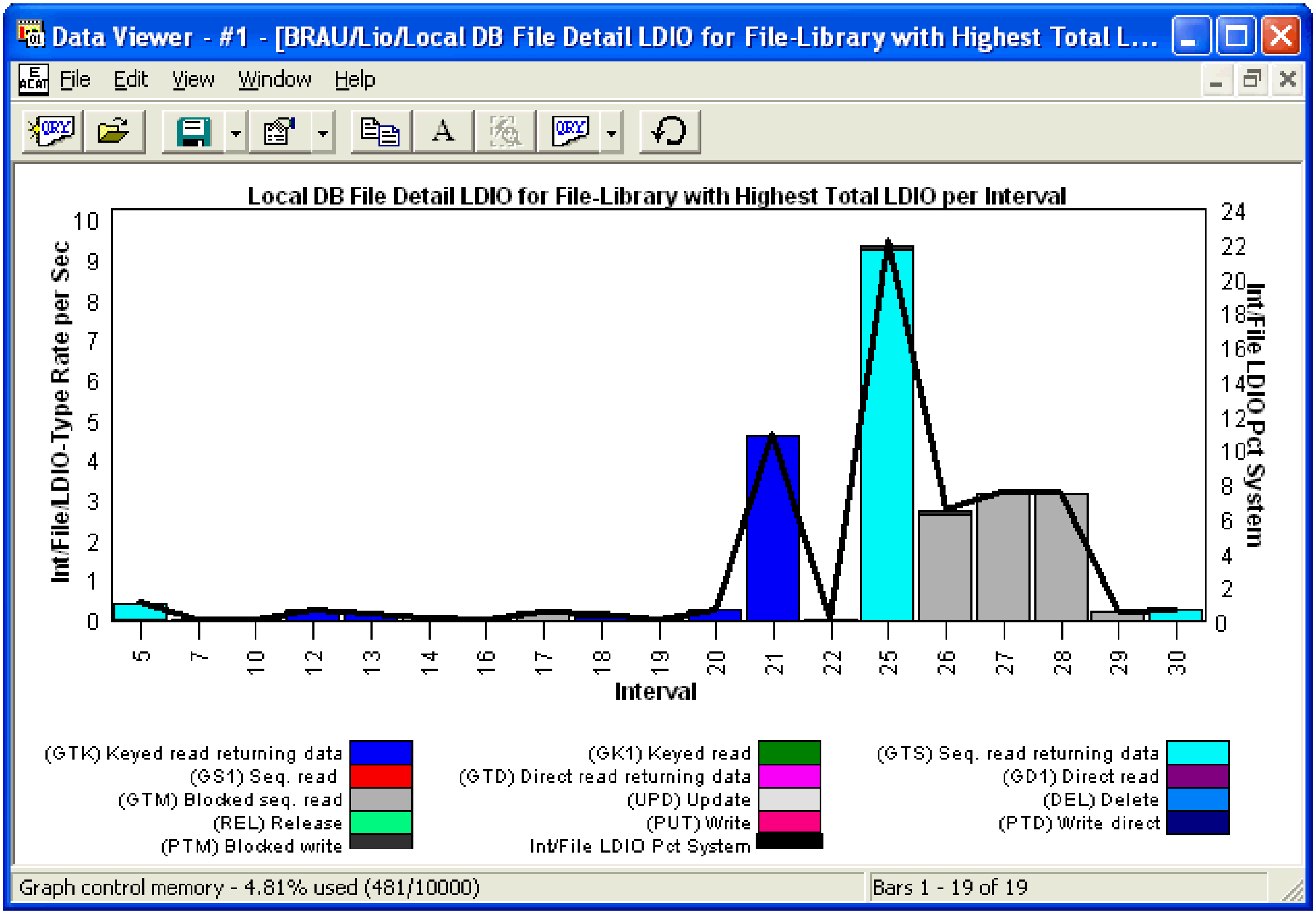
Interval	Interval Max Timestamp	Interval LDIO Total	Interval LDIO Rate per Sec	Interval LDIO Pct System	File-Library	Int/Fil LDIO Total
5	2004-07-15-15.05.57.979495	8	.456951	1.08	QA1PSCL2 >	8
5	2004-07-15-15.05.57.979495	8	.456951	1.08	QA1PSCL2 >	8
7	2004-07-15-15.05.59.354837	1	.057119	.14	QLTAMID >	1
10	2004-07-15-15.06.00.984330	1	.057119	.14	QPFRCOLTRC >	1
12	2004-07-15-15.06.02.484249	6	.342713	.81	QADBXRMTNM >	5
13	2004-07-15-15.06.02.711176	13	.742545	1.76	QADBXRMTNM >	3
14	2004-07-15-15.06.03.394401	3	.171357	.41	QAIDRGPH08 >	2
16	2004-07-15-15.06.04.479109	1	.057119	.14	QA1ASP >	1
17	2004-07-15-15.06.05.056609	5	.285594	.68	WSST >	4
18	2004-07-15-15.06.05.740406	4	.228475	.54	QLTAMID >	3
18	2004-07-15-15.06.05.740406	4	.228475	.54	QLTAMID >	3
19	2004-07-15-15.06.06.092861	1	.057119	.14	QA1AMM >	1
20	2004-07-15-15.06.07.051490	7	.399832	.95	QADBXRMTNM >	5
21	2004-07-15-15.06.07.738225	94	5.369170	12.72	QAPZGRP >	81
22	2004-07-15-15.06.07.984378	1	.057119	.14	WSST >	1
25	2004-07-15-15.06.10.021808	405	23.133128	54.80	ENUMBLD >	164
25	2004-07-15-15.06.10.021808	405	23.133128	54.80	ENUMBLD >	164
26	2004-07-15-15.06.10.663011	50	2.855942	6.77	ENUMBLD2 >	48

Records 1 - 18 of 23

Graph Type: stacked vertical bar
X-axis: Interval number
Y-Axis: This value is the number of LDIO ops occurring of each type per second. Each color represents a different type of LDIO.
Second Y-Axis: This line shows how much each file/library's LDIOs contributed to the total system LDIOs for every interval.

Example:

5.1.6 DB File Detail LDIO for File-Library With Highest Total LDIO per Interval



Graph control memory - 4.81% used (481/10000)

Bars 1 - 19 of 19

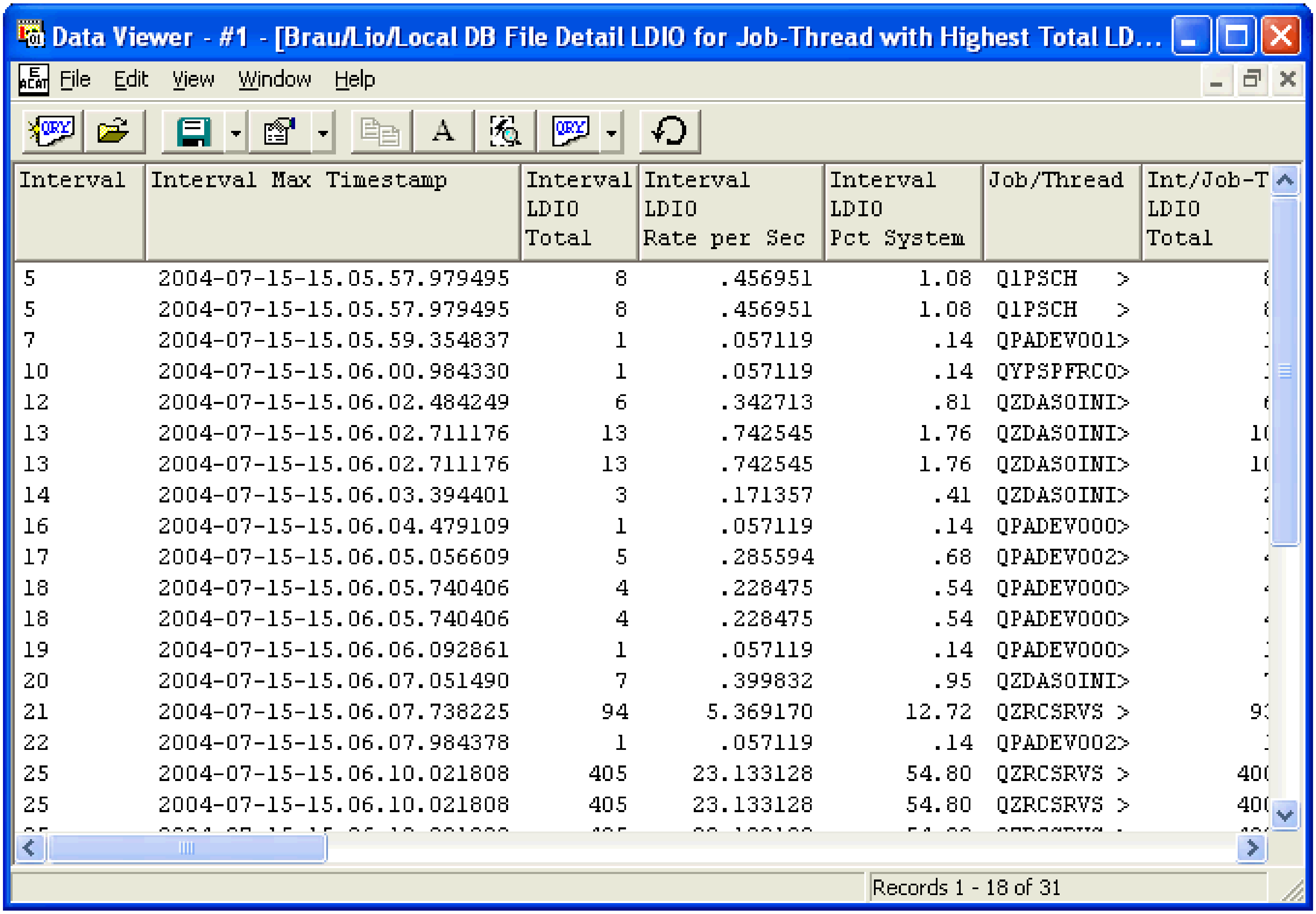
Table of Contents | Previous | Next

5.1.7 DB File Detail LDIO for Job-thread with Highest Total LDIO per Interval

Description: This graph shows the job or thread with the most "logical disk I/O" (LDIO) activity for each time interval in the collection.
It also shows the type(s) of LDIO done by these jobs/threads and the percentage of the collection's total LDIO. Each job/thread's LDIO activity is also shown.

Each color in a bar represents a type of LDIO. Clicking one of these colors brings up additional details about the LDIO for the interval selected.

Example:

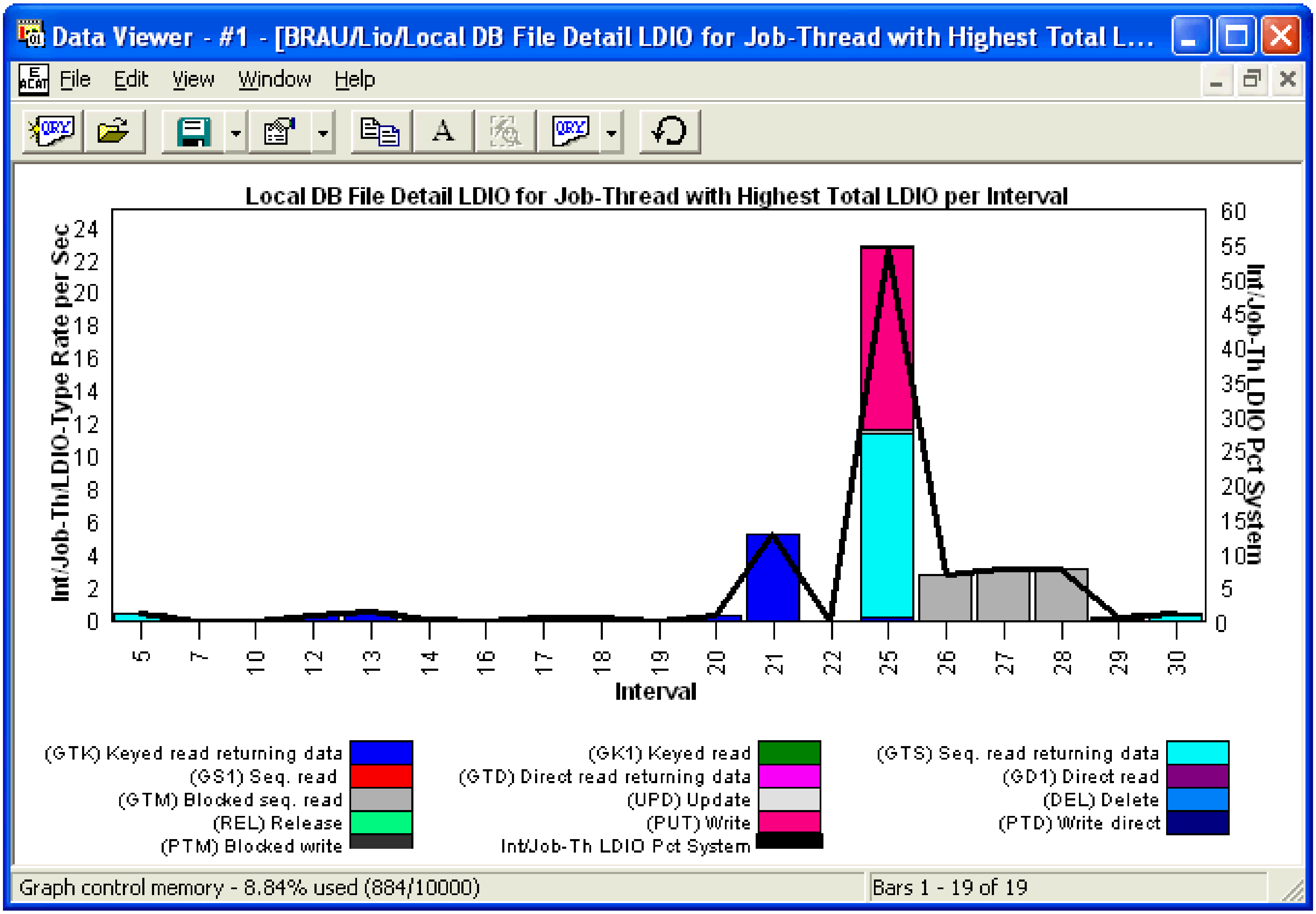


Data Viewer - #1 - [Brau/Lio/Local DB File Detail LDIO for Job-Thread with Highest Total LD...
File Edit View Window Help
Records 1 - 18 of 31

Interval	Interval Max Timestamp	Interval LDIO Total	Interval LDIO Rate per Sec	Interval LDIO Pct System	Job/Thread	Int/Job-T LDIO Total
5	2004-07-15-15.05.57.979495	8	.456951	1.08	Q1PSCH >	8
5	2004-07-15-15.05.57.979495	8	.456951	1.08	Q1PSCH >	8
7	2004-07-15-15.05.59.354837	1	.057119	.14	QPADEV001>	1
10	2004-07-15-15.06.00.984330	1	.057119	.14	QYPSPFRCO>	1
12	2004-07-15-15.06.02.484249	6	.342713	.81	QZDASOINI>	6
13	2004-07-15-15.06.02.711176	13	.742545	1.76	QZDASOINI>	10
13	2004-07-15-15.06.02.711176	13	.742545	1.76	QZDASOINI>	10
14	2004-07-15-15.06.03.394401	3	.171357	.41	QZDASOINI>	2
16	2004-07-15-15.06.04.479109	1	.057119	.14	QPADEV000>	1
17	2004-07-15-15.06.05.056609	5	.285594	.68	QPADEV002>	4
18	2004-07-15-15.06.05.740406	4	.228475	.54	QPADEV000>	4
18	2004-07-15-15.06.05.740406	4	.228475	.54	QPADEV000>	4
19	2004-07-15-15.06.06.092861	1	.057119	.14	QPADEV000>	1
20	2004-07-15-15.06.07.051490	7	.399832	.95	QZDASOINI>	7
21	2004-07-15-15.06.07.738225	94	5.369170	12.72	QZRCSRVS >	9
22	2004-07-15-15.06.07.984378	1	.057119	.14	QPADEV002>	1
25	2004-07-15-15.06.10.021808	405	23.133128	54.80	QZRCSRVS >	40
25	2004-07-15-15.06.10.021808	405	23.133128	54.80	QZRCSRVS >	40

Graph Type: stacked vertical bar
X-axis: Interval number
Y-Axis: This value is the number of LDIO ops occurring of each type per second for the job/thread having the highest total LDIO. Each color represents a different type of LDIO.
Second Y-Axis: This line shows how much each job/thread's LDIOs contributed to the total system LDIOs.

Example:

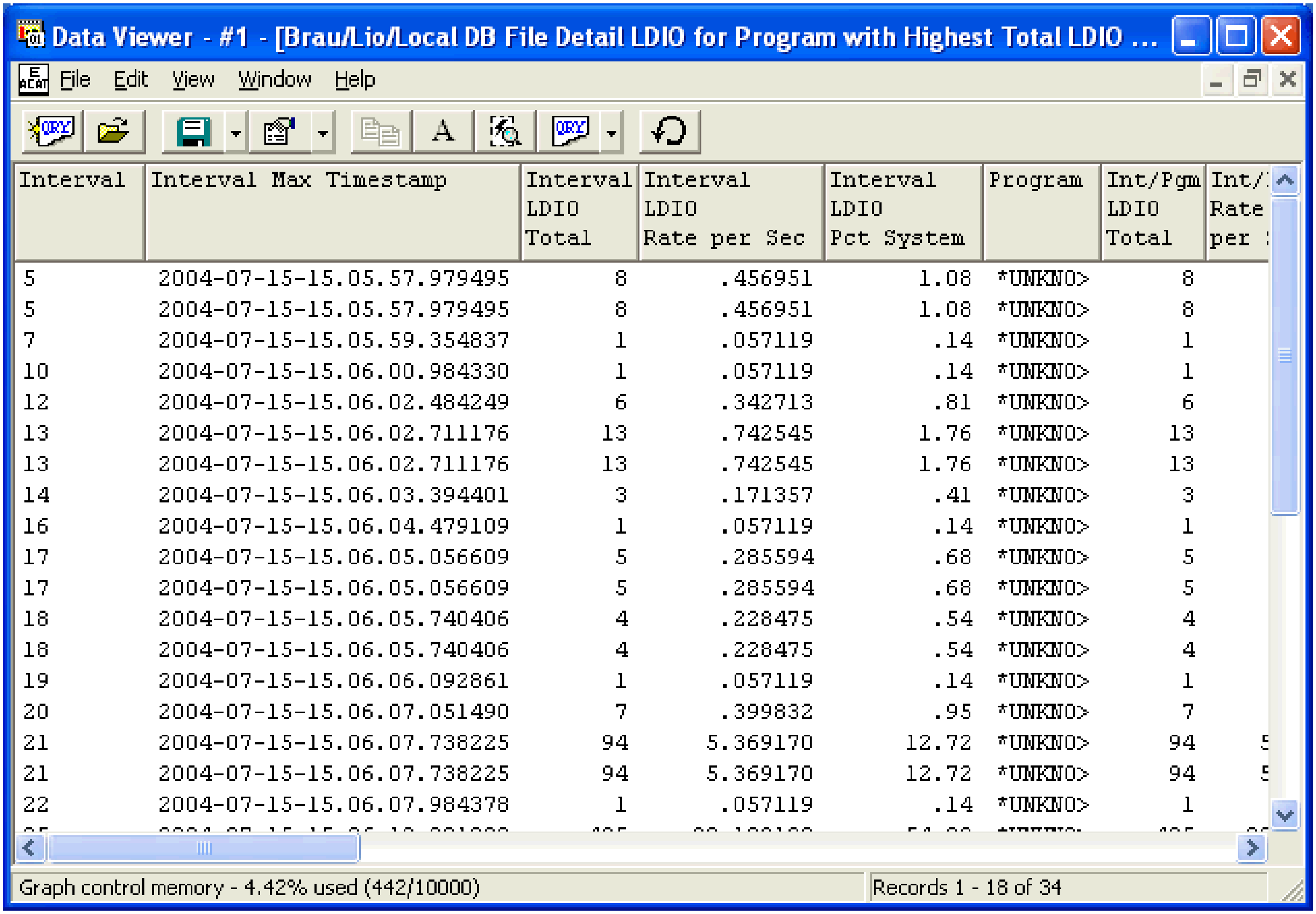

Data Viewer - #1 - [BRAU/Lio/Local DB File Detail LDIO for Job-Thread with Highest Total L...
File Edit View Window Help
Local DB File Detail LDIO for Job-Thread with Highest Total LDIO per Interval
Int/Job-Th LDIO-Type Rate per Sec
0
2
4
6
8
10
12
14
16
18
20
22
24
Int/Job-Th LDIO Pct System
0
5
10
15
20
25
30
35
40
45
50
55
60
5
7
10
12
13
14
16
17
18
19
20
21
22
25
26
27
28
29
30
Interval
(GTK) Keyed read returning data
(GS1) Seq. read
(GTM) Blocked seq. read
(REL) Release
(PTM) Blocked write
(GK1) Keyed read
(GTD) Direct read returning data
(UPD) Update
(PUT) Write
Int/Job-Th LDIO Pct System
(GTS) Seq. read returning data
(GD1) Direct read
(DEL) Delete
(PTD) Write direct
Graph control memory - 8.84% used (884/10000)
Bars 1 - 19 of 19

5.1.8 DB File Detail LDIO for Program with Highest Total LDIO per Interval

Description: This report shows the program with the most "logical disk I/O" (LDIO) activity for each time interval in the collection. It also shows the type(s) of LDIO done by these programs and the percentage of the collection's total LDIO. Each program's LDIO activity is also shown.

Note: In order for program names to be provided, *MIENTRY and *MIEXIT events must be included in the PEX collection.

Example:



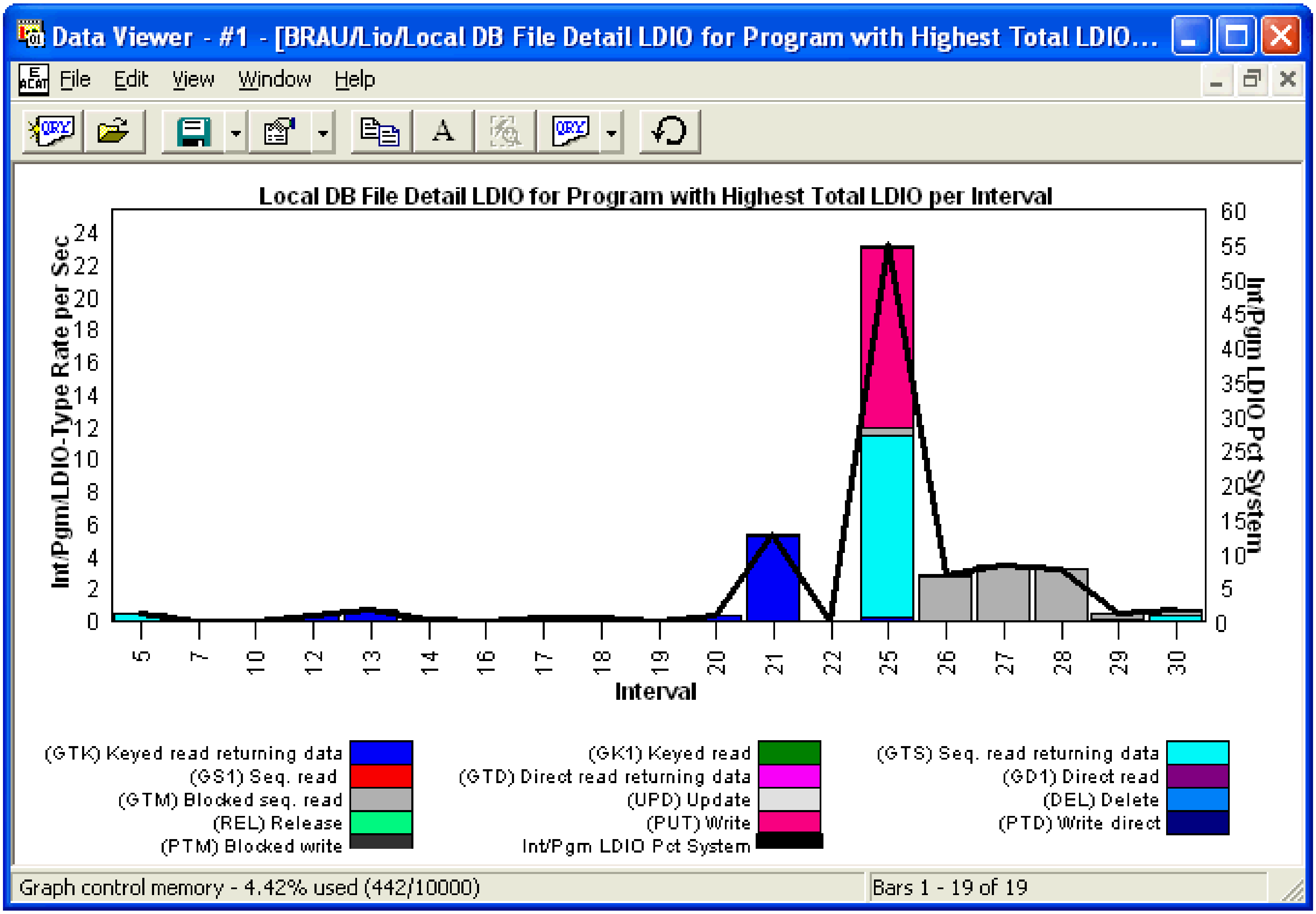
Data Viewer - #1 - [Brau/Lio/Local DB File Detail LDIO for Program with Highest Total LDIO ...
File Edit View Window Help

Interval	Interval Max Timestamp	Interval LDIO Total	Interval LDIO Rate per Sec	Interval LDIO Pct System	Program	Int/Pgm LDIO Total	Int/ Rate per
5	2004-07-15-15.05.57.979495	8	.456951	1.08	*UNKNO>	8	
5	2004-07-15-15.05.57.979495	8	.456951	1.08	*UNKNO>	8	
7	2004-07-15-15.05.59.354837	1	.057119	.14	*UNKNO>	1	
10	2004-07-15-15.06.00.984330	1	.057119	.14	*UNKNO>	1	
12	2004-07-15-15.06.02.484249	6	.342713	.81	*UNKNO>	6	
13	2004-07-15-15.06.02.711176	13	.742545	1.76	*UNKNO>	13	
13	2004-07-15-15.06.02.711176	13	.742545	1.76	*UNKNO>	13	
14	2004-07-15-15.06.03.394401	3	.171357	.41	*UNKNO>	3	
16	2004-07-15-15.06.04.479109	1	.057119	.14	*UNKNO>	1	
17	2004-07-15-15.06.05.056609	5	.285594	.68	*UNKNO>	5	
17	2004-07-15-15.06.05.056609	5	.285594	.68	*UNKNO>	5	
18	2004-07-15-15.06.05.740406	4	.228475	.54	*UNKNO>	4	
18	2004-07-15-15.06.05.740406	4	.228475	.54	*UNKNO>	4	
19	2004-07-15-15.06.06.092861	1	.057119	.14	*UNKNO>	1	
20	2004-07-15-15.06.07.051490	7	.399832	.95	*UNKNO>	7	
21	2004-07-15-15.06.07.738225	94	5.369170	12.72	*UNKNO>	94	5
21	2004-07-15-15.06.07.738225	94	5.369170	12.72	*UNKNO>	94	5
22	2004-07-15-15.06.07.984378	1	.057119	.14	*UNKNO>	1	

Graph control memory - 4.42% used (442/10000)   Records 1 - 18 of 34

Graph Type: stacked vertical bar
X-axis: Interval number
Y-Axis: This value is the number of LDIO ops occurring of each type per second. Each color represents a different type of LDIO.
Second Y-Axis: This line shows how much each program's LDIO ops contributed to the total system LDIOs.

Example:


Data Viewer - #1 - [BRAU/Lio/Local DB File Detail LDIO for Program with Highest Total LDIO...
File Edit View Window Help
Local DB File Detail LDIO for Program with Highest Total LDIO per Interval
Int/Pgm LDIO-Type Rate per Sec
0
2
4
6
8
10
12
14
16
18
20
22
24
Int/Pgm LDIO Pct System
0
5
10
15
20
25
30
35
40
45
50
55
60
5
7
10
12
13
14
16
17
18
19
20
21
22
25
26
27
28
29
30
Interval
(GTK) Keyed read returning data
(GS1) Seq. read
(GTM) Blocked seq. read
(REL) Release
(PTM) Blocked write
(GK1) Keyed read
(GTD) Direct read returning data
(UPD) Update
(PUT) Write
Int/Pgm LDIO Pct System
(GTS) Seq. read returning data
(GD1) Direct read
(DEL) Delete
(PTD) Write direct

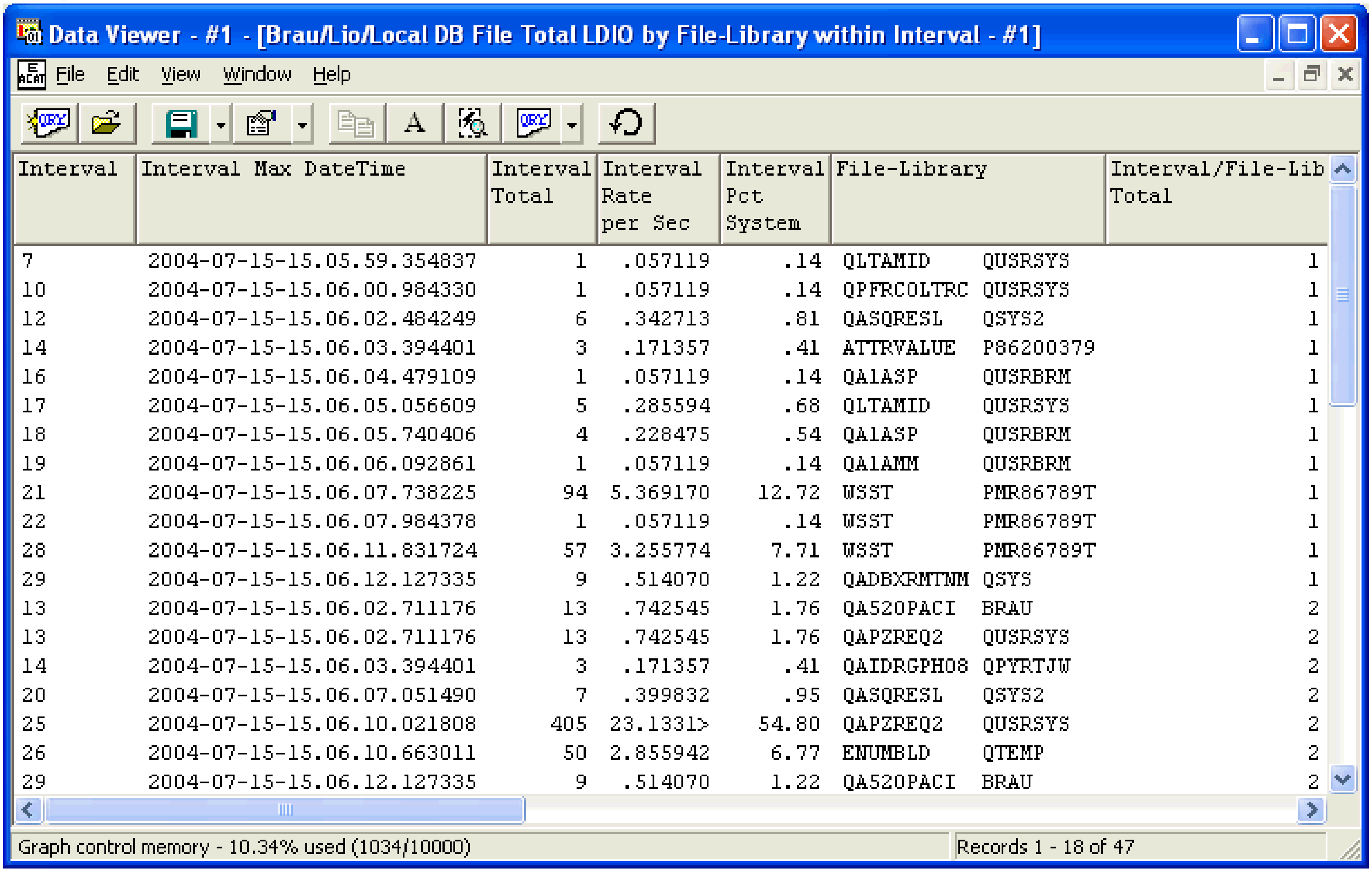
Graph control memory - 4.42% used (442/10000)

Bars 1 - 19 of 19

5.1.9 DB File Detail LDIO by File-library within Interval

Description: This graph shows the "logical disk I/O" (LDIO) activity for every file by time interval. It shows the percentage of the collection's total LDIO each interval/file's LDIO activity represents.

Example:


Data Viewer - #1 - [Brau/Lio/Local DB File Total LDIO by File-Library within Interval - #1]
File
Edit
View
Window
Help

Interval	Interval Max DateTime	Interval Total	Interval Rate per Sec	Interval Pct System	File-Library	Interval/File-Lib Total
7	2004-07-15-15.05.59.354837	1	.057119	.14	QLTAMID QUSRSYS	1
10	2004-07-15-15.06.00.984330	1	.057119	.14	QPFRCOLTRC QUSRSYS	1
12	2004-07-15-15.06.02.484249	6	.342713	.81	QASQRESL QSYS2	1
14	2004-07-15-15.06.03.394401	3	.171357	.41	ATTRVALUE P86200379	1
16	2004-07-15-15.06.04.479109	1	.057119	.14	QA1ASP QUSRBRM	1
17	2004-07-15-15.06.05.056609	5	.285594	.68	QLTAMID QUSRSYS	1
18	2004-07-15-15.06.05.740406	4	.228475	.54	QA1ASP QUSRBRM	1
19	2004-07-15-15.06.06.092861	1	.057119	.14	QA1AMM QUSRBRM	1
21	2004-07-15-15.06.07.738225	94	5.369170	12.72	WSST PMR86789T	1
22	2004-07-15-15.06.07.984378	1	.057119	.14	WSST PMR86789T	1
28	2004-07-15-15.06.11.831724	57	3.255774	7.71	WSST PMR86789T	1
29	2004-07-15-15.06.12.127335	9	.514070	1.22	QADBXRMTNM QSYS	1
13	2004-07-15-15.06.02.711176	13	.742545	1.76	QA520PACI BRAU	2
13	2004-07-15-15.06.02.711176	13	.742545	1.76	QAPZREQ2 QUSRSYS	2
14	2004-07-15-15.06.03.394401	3	.171357	.41	QAIDRGPH08 QPYRTJW	2
20	2004-07-15-15.06.07.051490	7	.399832	.95	QASQRESL QSYS2	2
25	2004-07-15-15.06.10.021808	405	23.1331>	54.80	QAPZREQ2 QUSRSYS	2
26	2004-07-15-15.06.10.663011	50	2.855942	6.77	ENUMBLD QTEMP	2
29	2004-07-15-15.06.12.127335	9	.514070	1.22	QA520PACI BRAU	2

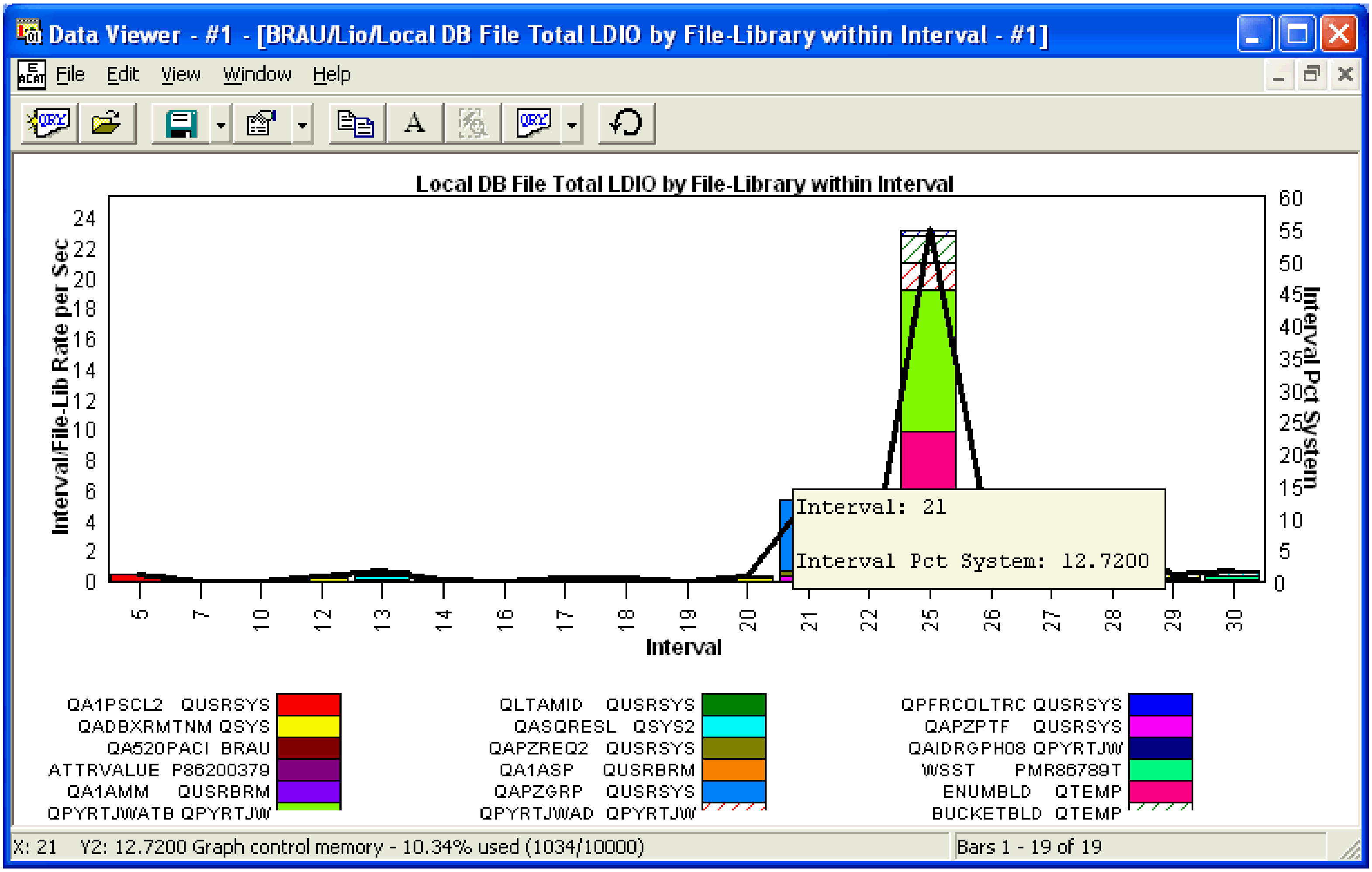
Graph control memory - 10.34% used (1034/10000)

Records 1 - 18 of 47

Graph Type: stacked vertical bar
X-axis: Interval number
Y-Axis: This value is the number of LDIO ops occurring per second for a file within an interval. Each color represents a different file.
Second Y-Axis: This line shows how much the collection's LDIOs contributed to the total system LDIOs for each interval.

Example:

5.1.9 DB File Detail LDIO by File-library within Interval



Interval: 21

Interval Pct System: 12.7200

X: 21 Y2: 12.7200 Graph control memory - 10.34% used (1034/10000)

Bars 1 - 19 of 19

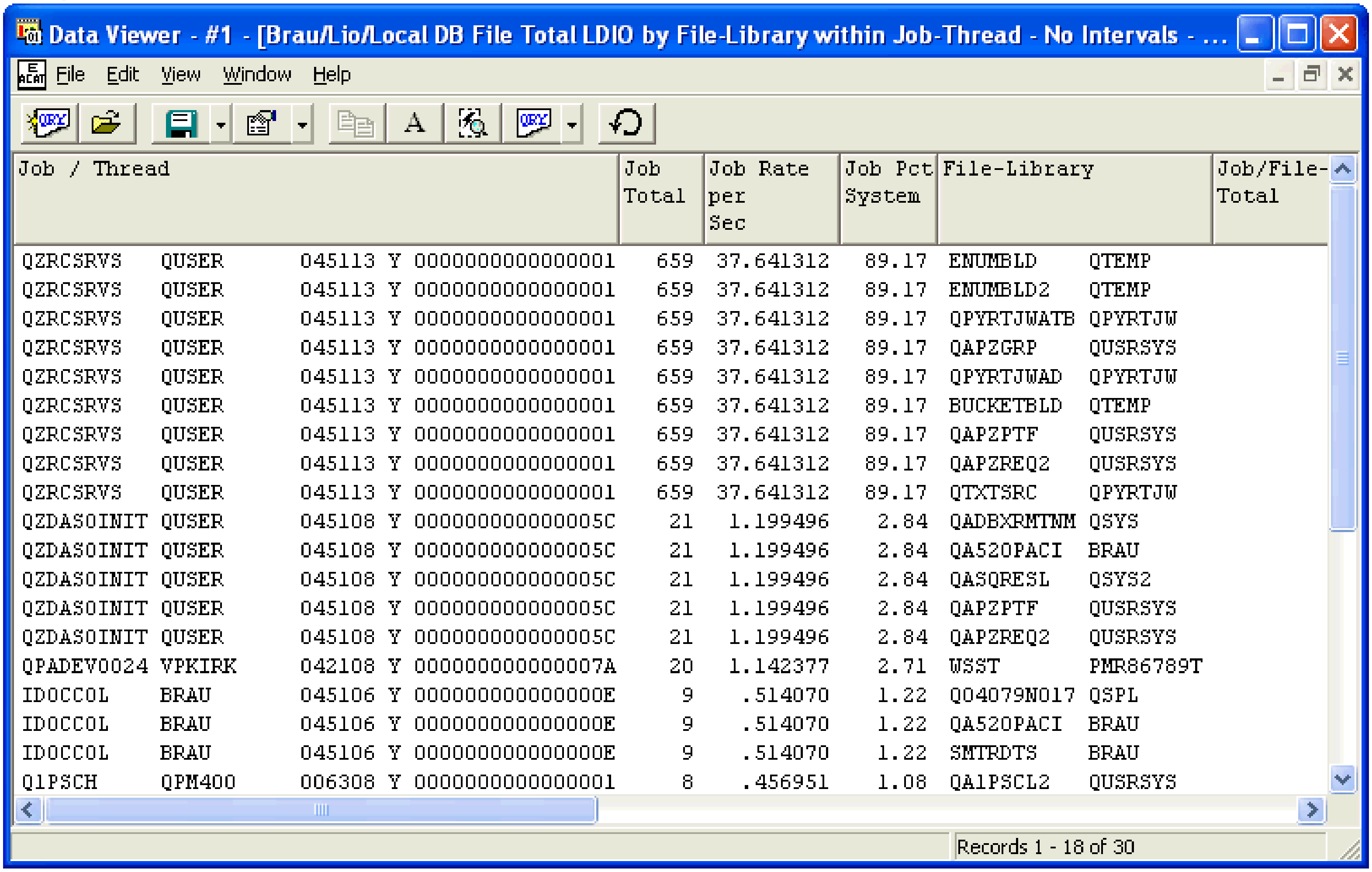
Table of Contents | Previous | Next

5.1.10 DB File Detail LDIO by File-library within Job-thread - No Intervals

Description: This graph shows the job or thread with the most "logical disk I/O" (LDIO) activity for file and time interval in the collection. It also shows the type(s) of LDIO done by these jobs/threads and the percentage of the collection's total LDIO.

Each color in a bar represents a type of LDIO. Clicking one of these colors brings up additional details about the LDIO for the file and job selected.

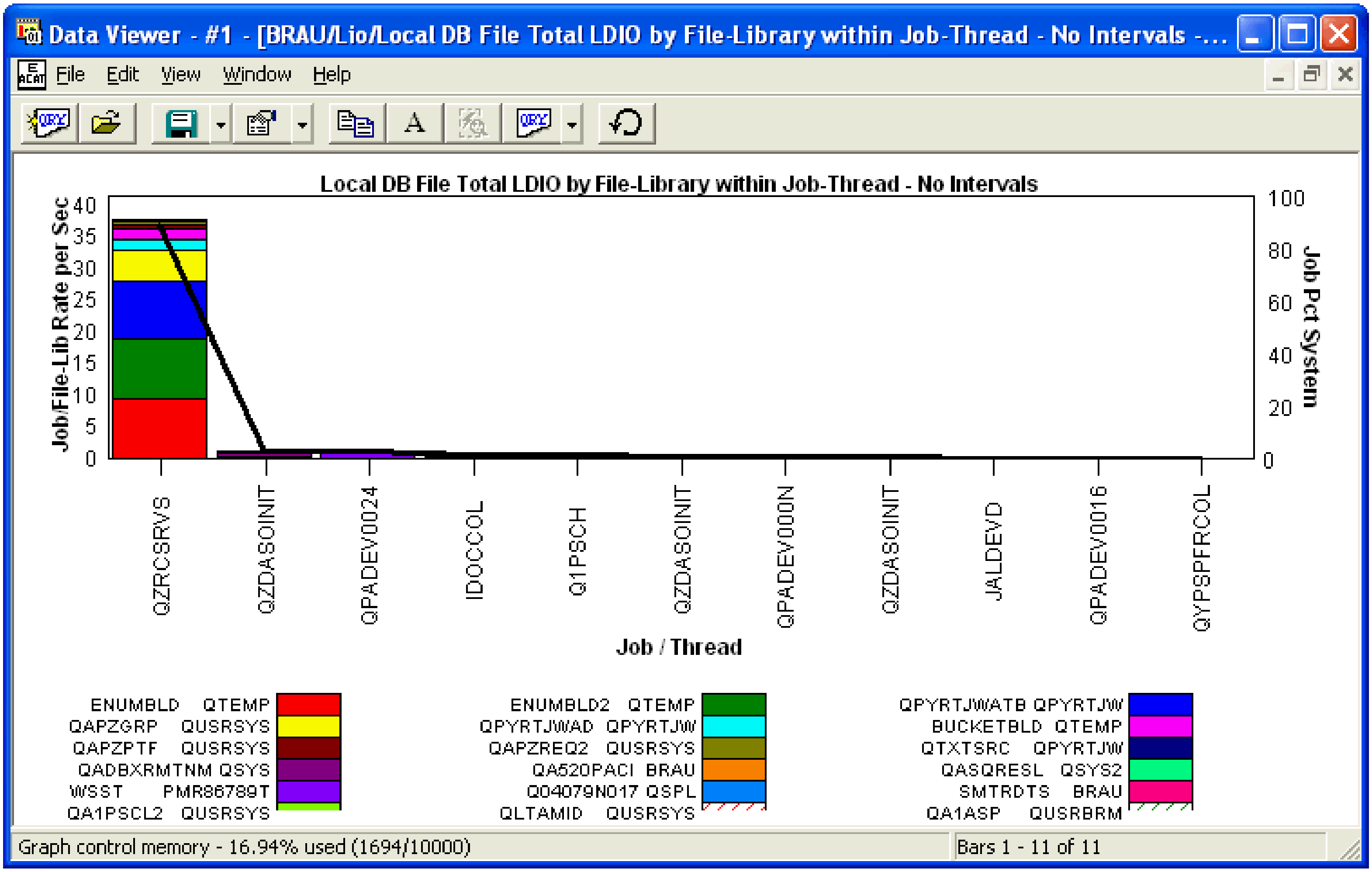
Example:


Data Viewer - #1 - [Brau/Lio/Local DB File Total LDIO by File-Library within Job-Thread - No Intervals - ...
File
Edit
View
Window
Help
Records 1 - 18 of 30

Job / Thread	Job Total	Job Rate per Sec	Job Pct System	File-Library	Job/File-Total
QZRCSRVS QUSER 045113 Y 0000000000000001	659	37.641312	89.17	ENUMBLD QTEMP	
QZRCSRVS QUSER 045113 Y 0000000000000001	659	37.641312	89.17	ENUMBLD2 QTEMP	
QZRCSRVS QUSER 045113 Y 0000000000000001	659	37.641312	89.17	QPYRTJWATB QPYRTJW	
QZRCSRVS QUSER 045113 Y 0000000000000001	659	37.641312	89.17	QAPZGRP QUSRSYS	
QZRCSRVS QUSER 045113 Y 0000000000000001	659	37.641312	89.17	QPYRTJWAD QPYRTJW	
QZRCSRVS QUSER 045113 Y 0000000000000001	659	37.641312	89.17	BUCKETBLD QTEMP	
QZRCSRVS QUSER 045113 Y 0000000000000001	659	37.641312	89.17	QAPZPTF QUSRSYS	
QZRCSRVS QUSER 045113 Y 0000000000000001	659	37.641312	89.17	QAPZREQ2 QUSRSYS	
QZRCSRVS QUSER 045113 Y 0000000000000001	659	37.641312	89.17	QTXTSRC QPYRTJW	
QZDASOINIT QUSER 045108 Y 000000000000005C	21	1.199496	2.84	QADBXRMTNM QSYS	
QZDASOINIT QUSER 045108 Y 000000000000005C	21	1.199496	2.84	QA520PACI BRAU	
QZDASOINIT QUSER 045108 Y 000000000000005C	21	1.199496	2.84	QASQRESL QSYS2	
QZDASOINIT QUSER 045108 Y 000000000000005C	21	1.199496	2.84	QAPZPTF QUSRSYS	
QZDASOINIT QUSER 045108 Y 000000000000005C	21	1.199496	2.84	QAPZREQ2 QUSRSYS	
QPADEV0024 VPKIRK 042108 Y 000000000000007A	20	1.142377	2.71	WSST PMR86789T	
IDOCCOL BRAU 045106 Y 000000000000000E	9	.514070	1.22	Q04079N017 QSPL	
IDOCCOL BRAU 045106 Y 000000000000000E	9	.514070	1.22	QA520PACI BRAU	
IDOCCOL BRAU 045106 Y 000000000000000E	9	.514070	1.22	SMTRDTS BRAU	
Q1PSCH QPM400 006308 Y 0000000000000001	8	.456951	1.08	QA1PSCL2 QUSRSYS	

Graph Type: stacked vertical bar
X-axis: Job/thread contributing LDIOs within the collection. The entire field is shown by placing the mouse over a bar.
Y-Axis: This value is the number of LDIO ops occurring of each type per second for a specific file within a job. Each color represents a different file.
Second Y-Axis: This line shows how much each jobs's total LDIO ops contributed to the total system LDIOs.

Example:



Data Viewer - #1 - [BRAU/Lio/Local DB File Total LDIO by File-Library within Job-Thread - No Intervals -...

Graph control memory - 16.94% used (1694/10000)

Bars 1 - 11 of 11

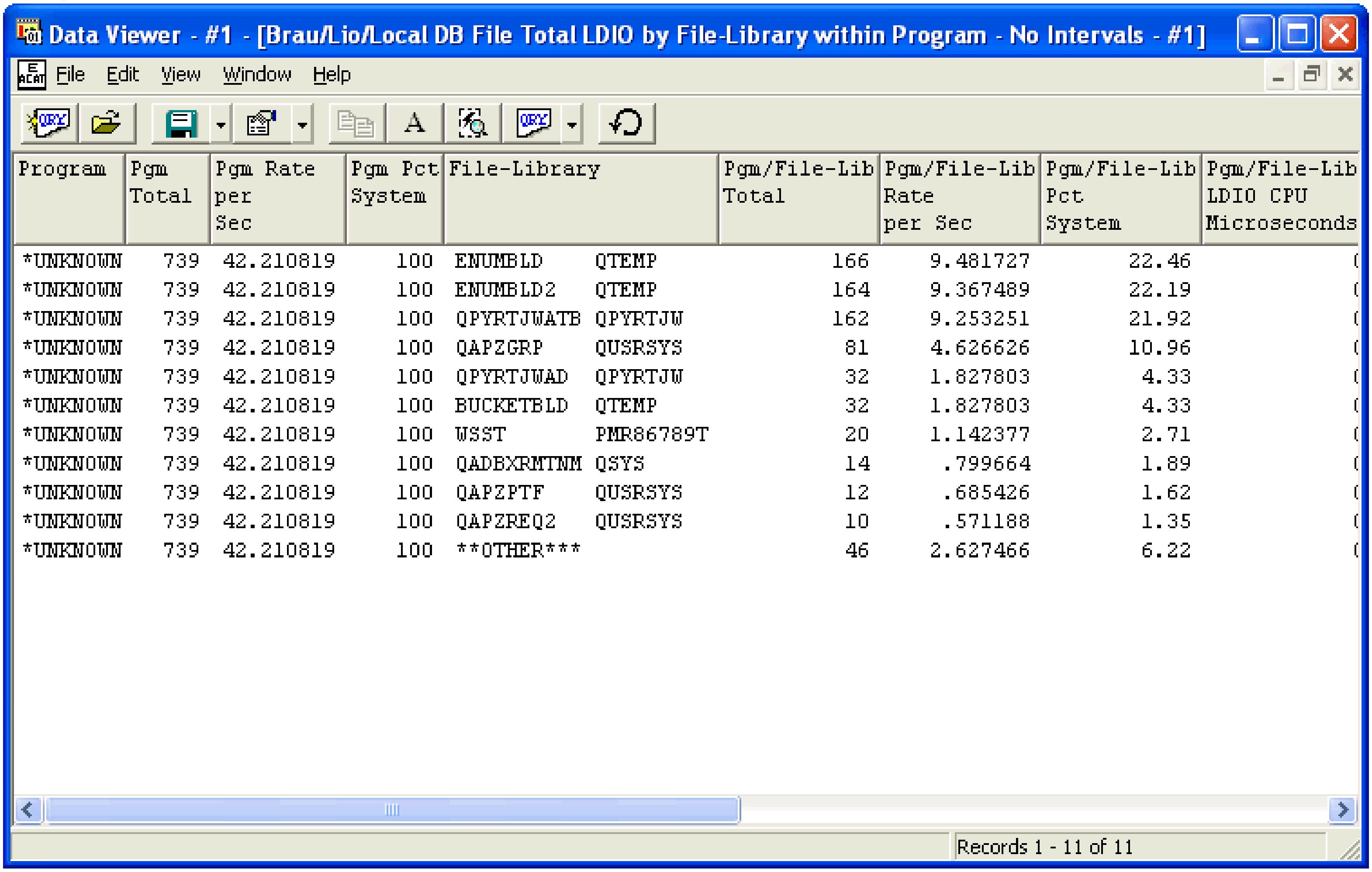
Table of Contents | Previous | Next

5.1.11 DB File Detail LDIO by File-library within Program - No Intervals

Description: This report shows the program LDIO activity for each file within the collection.

Note: In order for program names to be provided, *MIENTRY and *MIEXIT events must be included in the PEX collection.

Example:

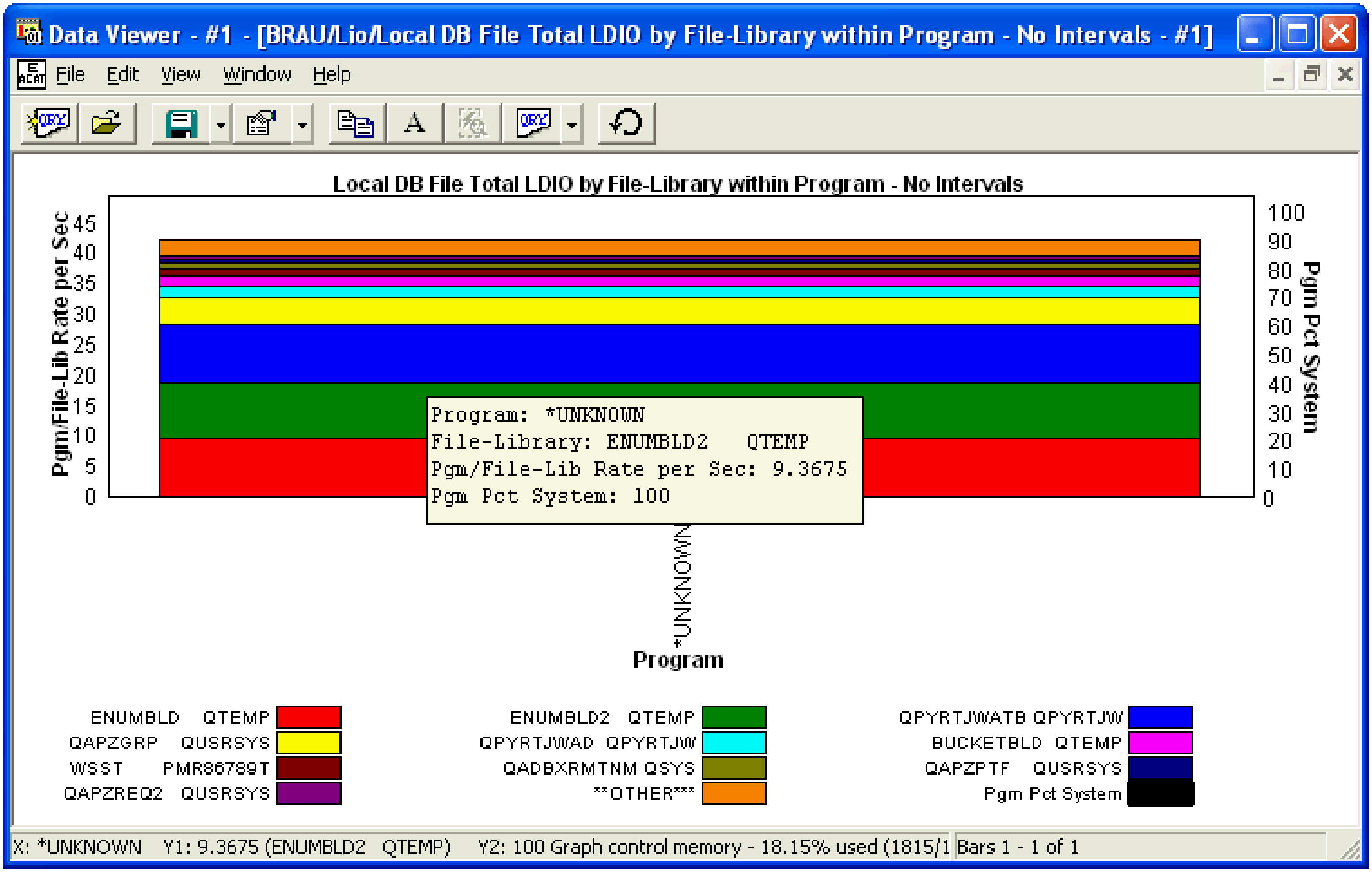

Data Viewer - #1 - [Brau/Lio/Local DB File Total LDIO by File-Library within Program - No Intervals - #1]
File
Edit
View
Window
Help

Program	Pgm Total	Pgm Rate per Sec	Pgm Pct System	File-Library		Pgm/File-Lib Total	Pgm/File-Lib Rate per Sec	Pgm/File-Lib Pct System	Pgm/File-Lib LDIO CPU Microseconds
*UNKNOWN	739	42.210819	100	ENUMBLD	QTEMP	166	9.481727	22.46	(
*UNKNOWN	739	42.210819	100	ENUMBLD2	QTEMP	164	9.367489	22.19	(
*UNKNOWN	739	42.210819	100	QPYRTJWATB	QPYRTJW	162	9.253251	21.92	(
*UNKNOWN	739	42.210819	100	QAPZGRP	QUSRSYS	81	4.626626	10.96	(
*UNKNOWN	739	42.210819	100	QPYRTJWAD	QPYRTJW	32	1.827803	4.33	(
*UNKNOWN	739	42.210819	100	BUCKETBLD	QTEMP	32	1.827803	4.33	(
*UNKNOWN	739	42.210819	100	WSST	PMR86789T	20	1.142377	2.71	(
*UNKNOWN	739	42.210819	100	QADBXRMTNM	QSYS	14	.799664	1.89	(
*UNKNOWN	739	42.210819	100	QAPZPTF	QUSRSYS	12	.685426	1.62	(
*UNKNOWN	739	42.210819	100	QAPZREQ2	QUSRSYS	10	.571188	1.35	(
*UNKNOWN	739	42.210819	100	**OTHER***		46	2.627466	6.22	(

Records 1 - 11 of 11

Graph Type: stacked vertical bar
X-axis: Program name. The entire field is shown by placing the mouse over a bar.
Y-Axis: This value is the number of LDIO ops occurring per second for a file by program. Each color represents a different file
Second Y-Axis: This line shows how much each program's LDIO ops contributed to the total system LDIOs.

Example:



```
Program: *UNKNOWN
File-Library: ENUMBLD2    QTEMP
Pgm/File-Lib Rate per Sec: 9.3675
Pgm Pct System: 100
```

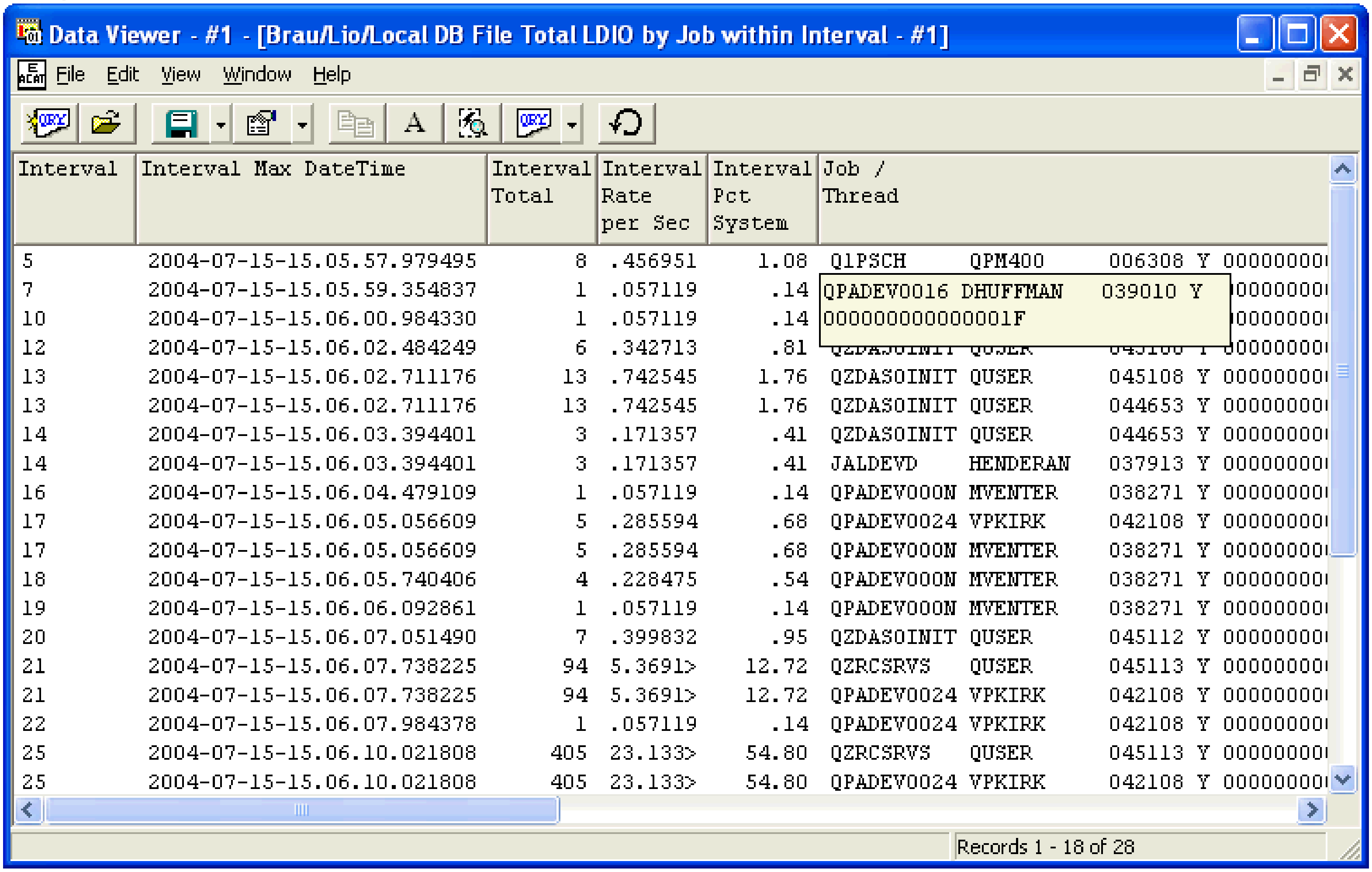
X: *UNKNOWN Y1: 9.3675 (ENUMBLD2 QTEMP) Y2: 100 Graph control memory - 18.15% used (1815/1 Bars 1 - 1 of 1

Table of Contents | Previous | Next

5.1.12 DB File Detail LDIO by Job within Interval

Description: This report shows the LDIO activity for each job within the collection each interval.
Each color in the graph repesents a unique job. The graph can be used to see which jobs were the most dominant as far as the LDIO operation rates throughout the collection.

Example:



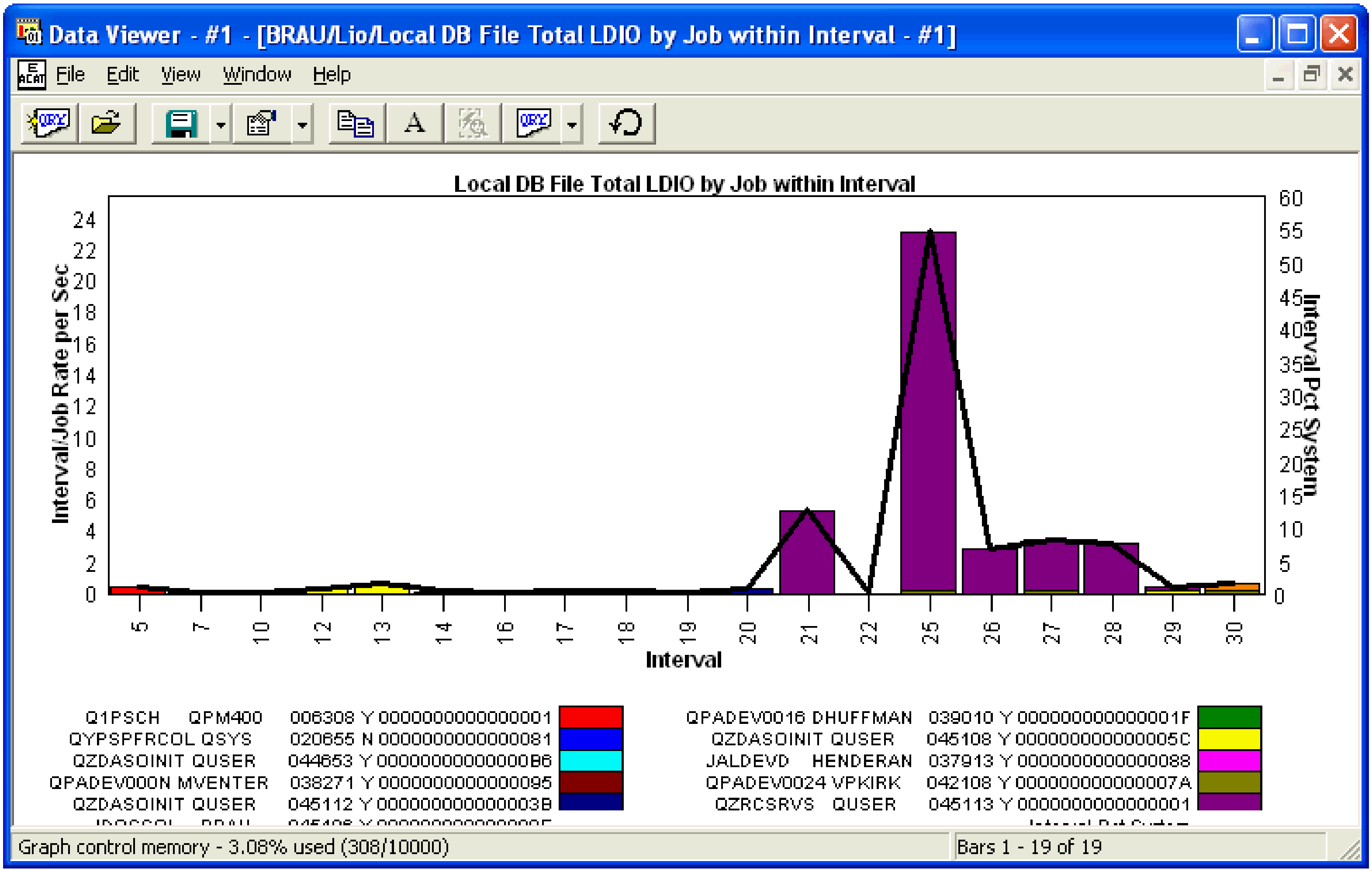
Data Viewer - #1 - [Brau/Lio/Local DB File Total LDIO by Job within Interval - #1]
File
Edit
View
Window
Help
QRY
QRY
A

```
Interval  Interval Max DateTime          Interval Interval Interval Job /
                                         Total    Rate     Pct      Thread
                                                  per Sec  System
5          2004-07-15-15.05.57.979495        8  .456951     1.08  Q1PSCH     QPM400     006308 Y 00000000
7          2004-07-15-15.05.59.354837        1  .057119      .14 QPADEV0016 DHUFFMAN    039010 Y 00000000
10         2004-07-15-15.06.00.984330        1  .057119      .14 000000000000001F                 00000000
12         2004-07-15-15.06.02.484249        6  .342713      .81  QZDASOINIT QUSER      045108 Y 00000000
13         2004-07-15-15.06.02.711176       13  .742545     1.76  QZDASOINIT QUSER      045108 Y 00000000
13         2004-07-15-15.06.02.711176       13  .742545     1.76  QZDASOINIT QUSER      044653 Y 00000000
14         2004-07-15-15.06.03.394401        3  .171357      .41  QZDASOINIT QUSER      044653 Y 00000000
14         2004-07-15-15.06.03.394401        3  .171357      .41  JALDEVD    HENDERAN   037913 Y 00000000
16         2004-07-15-15.06.04.479109        1  .057119      .14  QPADEV000N MVENTER    038271 Y 00000000
17         2004-07-15-15.06.05.056609        5  .285594      .68  QPADEV0024 VPKIRK     042108 Y 00000000
17         2004-07-15-15.06.05.056609        5  .285594      .68  QPADEV000N MVENTER    038271 Y 00000000
18         2004-07-15-15.06.05.740406        4  .228475      .54  QPADEV000N MVENTER    038271 Y 00000000
19         2004-07-15-15.06.06.092861        1  .057119      .14  QPADEV000N MVENTER    038271 Y 00000000
20         2004-07-15-15.06.07.051490        7  .399832      .95  QZDASOINIT QUSER      045112 Y 00000000
21         2004-07-15-15.06.07.738225       94  5.3691>    12.72  QZRCSRVS   QUSER      045113 Y 00000000
21         2004-07-15-15.06.07.738225       94  5.3691>    12.72  QPADEV0024 VPKIRK     042108 Y 00000000
22         2004-07-15-15.06.07.984378        1  .057119      .14  QPADEV0024 VPKIRK     042108 Y 00000000
25         2004-07-15-15.06.10.021808      405  23.133>    54.80  QZRCSRVS   QUSER      045113 Y 00000000
25         2004-07-15-15.06.10.021808      405  23.133>    54.80  QPADEV0024 VPKIRK     042108 Y 00000000
```

Records 1 - 18 of 28

Graph Type: stacked vertical bar
X-axis: Interval number
Y-Axis: This value is the number of LDIO ops occurring per second per job. Each color represents a different job.
Second Y-Axis: This line shows how interval's LDIO ops contributed to the total system LDIOs.

Example:


Data Viewer - #1 - [BRAU/Lio/Local DB File Total LDIO by Job within Interval - #1]
File
Edit
View
Window
Help
Local DB File Total LDIO by Job within Interval
Interval/Job Rate per Sec
0
2
4
6
8
10
12
14
16
18
20
22
24
Interval Pct System
0
5
10
15
20
25
30
35
40
45
50
55
60
Interval
5
7
10
12
13
14
16
17
18
19
20
21
22
25
26
27
28
29
30
Q1PSCH QPM400 006308 Y 0000000000000001
QYPSPFRCOL QSYS 020655 N 0000000000000081
QZDASOINIT QUSER 044653 Y 00000000000000B6
QPADEV000N MVENTER 038271 Y 0000000000000095
QZDASOINIT QUSER 045112 Y 000000000000003B
QPADEV0016 DHUFFMAN 039010 Y 000000000000001F
QZDASOINIT QUSER 045108 Y 000000000000005C
JALDEVD HENDERAN 037913 Y 0000000000000088
QPADEV0024 VPKIRK 042108 Y 000000000000007A
QZRCSRVS QUSER 045113 Y 0000000000000001

Graph control memory - 3.08% used (308/10000)

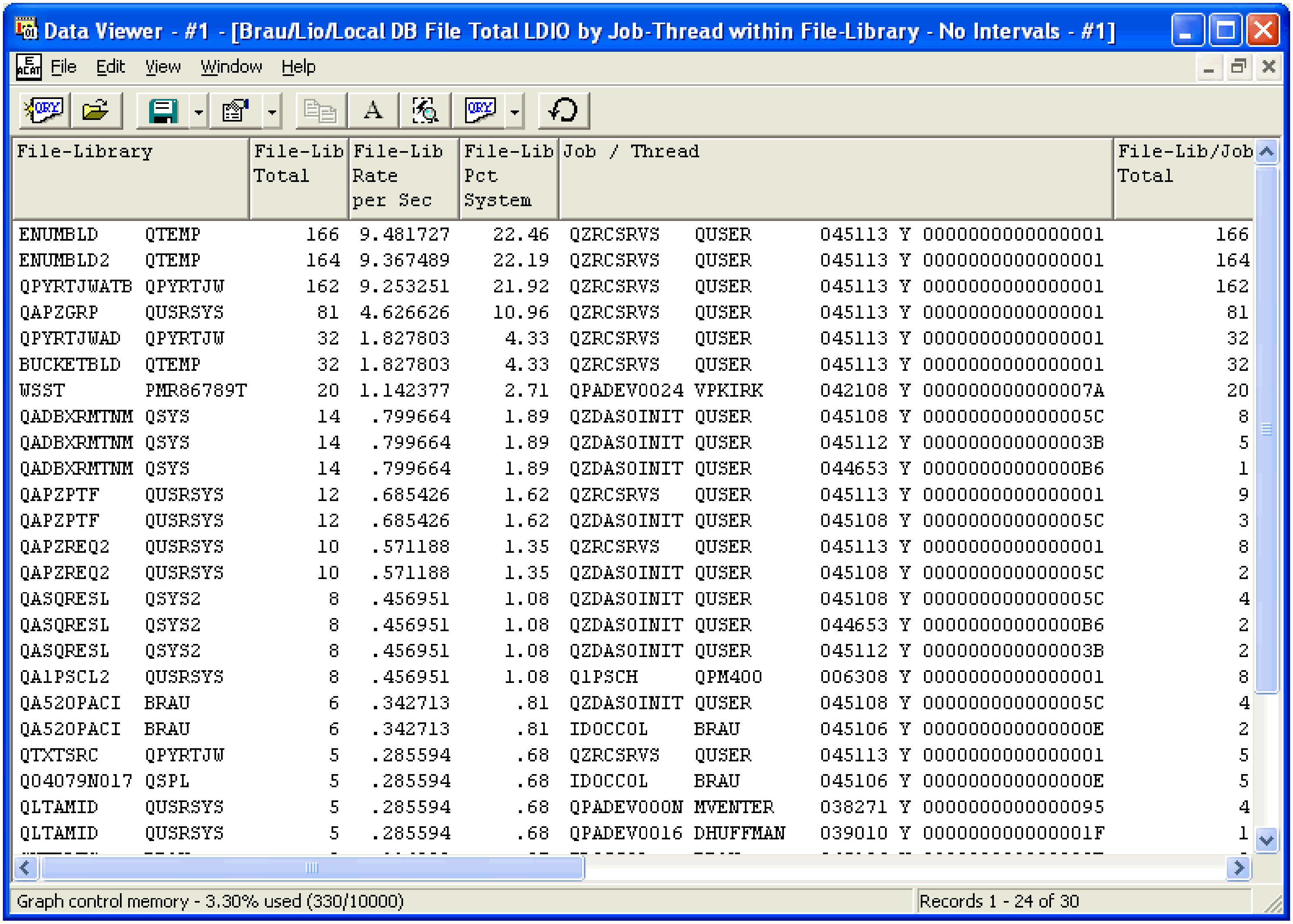
Bars 1 - 19 of 19

Table of Contents | Previous | Next

5.1.13 DB File Detail LDIO by Job-thread within File-library - No intervals

Description: This report shows the LDIO activity for each job by file within the collection.
Each color in the graph repesents a unique job. The X axis in the graph displays file names with the files/jobs receiving the highest rate of LDIO operations first.

Example:



Data Viewer - #1 - [Brau/Lio/Local DB File Total LDIO by Job-Thread within File-Library - No Intervals - #1]
File Edit View Window Help

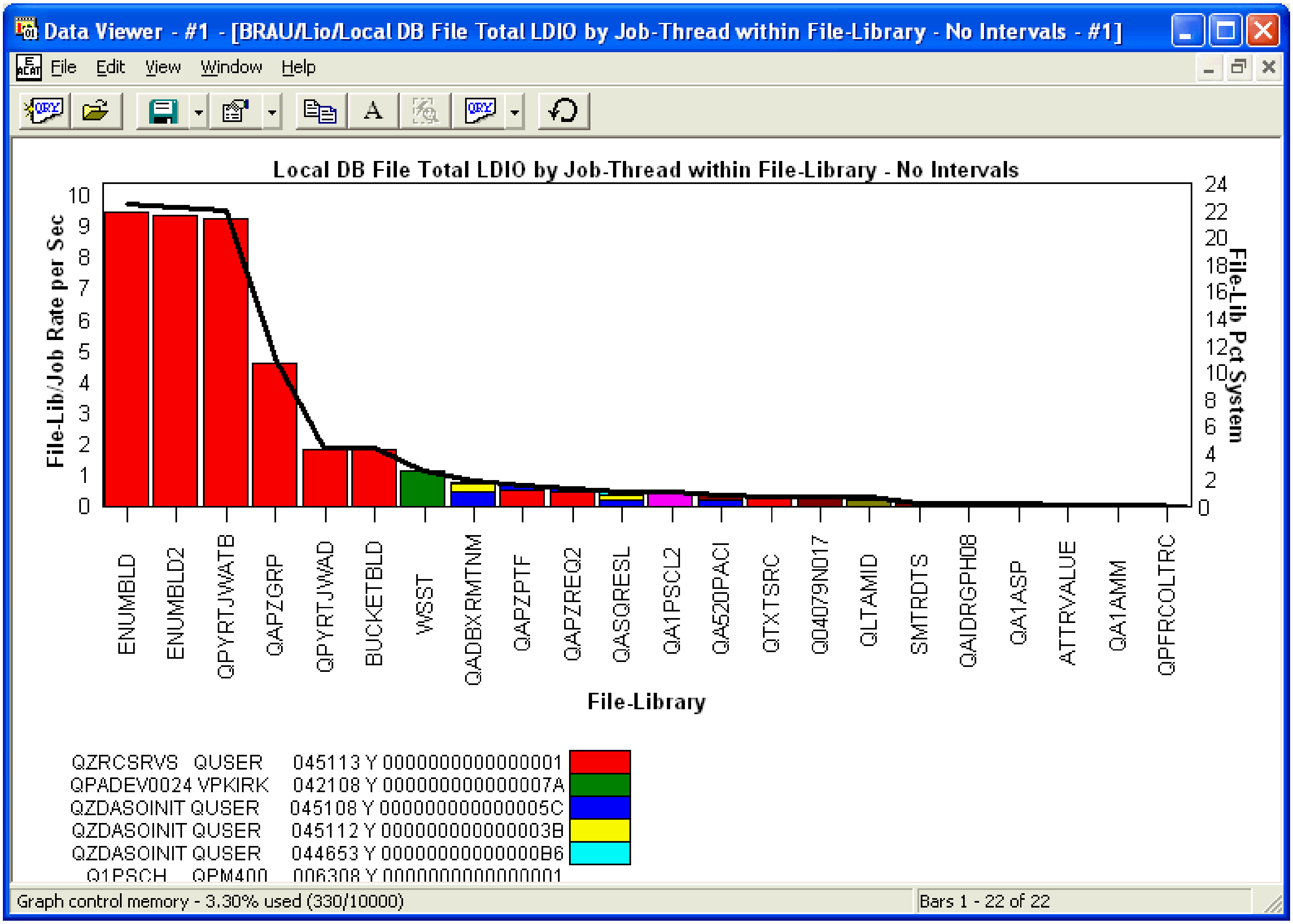
File-Library		File-Lib Total	File-Lib Rate per Sec	File-Lib Pct System	Job / Thread	File-Lib/Job Total
ENUMBLD	QTEMP	166	9.481727	22.46	QZRCSRVS QUSER 045113 Y 0000000000000001	166
ENUMBLD2	QTEMP	164	9.367489	22.19	QZRCSRVS QUSER 045113 Y 0000000000000001	164
QPYRTJWATB	QPYRTJW	162	9.253251	21.92	QZRCSRVS QUSER 045113 Y 0000000000000001	162
QAPZGRP	QUSRSYS	81	4.626626	10.96	QZRCSRVS QUSER 045113 Y 0000000000000001	81
QPYRTJWAD	QPYRTJW	32	1.827803	4.33	QZRCSRVS QUSER 045113 Y 0000000000000001	32
BUCKETBLD	QTEMP	32	1.827803	4.33	QZRCSRVS QUSER 045113 Y 0000000000000001	32
WSST	PMR86789T	20	1.142377	2.71	QPADEV0024 VPKIRK 042108 Y 000000000000007A	20
QADBXRMTNM	QSYS	14	.799664	1.89	QZDASOINIT QUSER 045108 Y 000000000000005C	8
QADBXRMTNM	QSYS	14	.799664	1.89	QZDASOINIT QUSER 045112 Y 000000000000003B	5
QADBXRMTNM	QSYS	14	.799664	1.89	QZDASOINIT QUSER 044653 Y 00000000000000B6	1
QAPZPTF	QUSRSYS	12	.685426	1.62	QZRCSRVS QUSER 045113 Y 0000000000000001	9
QAPZPTF	QUSRSYS	12	.685426	1.62	QZDASOINIT QUSER 045108 Y 000000000000005C	3
QAPZREQ2	QUSRSYS	10	.571188	1.35	QZRCSRVS QUSER 045113 Y 0000000000000001	8
QAPZREQ2	QUSRSYS	10	.571188	1.35	QZDASOINIT QUSER 045108 Y 000000000000005C	2
QASQRESL	QSYS2	8	.456951	1.08	QZDASOINIT QUSER 045108 Y 000000000000005C	4
QASQRESL	QSYS2	8	.456951	1.08	QZDASOINIT QUSER 044653 Y 00000000000000B6	2
QASQRESL	QSYS2	8	.456951	1.08	QZDASOINIT QUSER 045112 Y 000000000000003B	2
QA1PSCL2	QUSRSYS	8	.456951	1.08	Q1PSCH QPM400 006308 Y 0000000000000001	8
QA52OPACI	BRAU	6	.342713	.81	QZDASOINIT QUSER 045108 Y 000000000000005C	4
QA52OPACI	BRAU	6	.342713	.81	IDOCCOL BRAU 045106 Y 000000000000000E	2
QTXTSRC	QPYRTJW	5	.285594	.68	QZRCSRVS QUSER 045113 Y 0000000000000001	5
Q04079N017	QSPL	5	.285594	.68	IDOCCOL BRAU 045106 Y 000000000000000E	5
QLTAMID	QUSRSYS	5	.285594	.68	QPADEV000N MVENTER 038271 Y 0000000000000095	4
QLTAMID	QUSRSYS	5	.285594	.68	QPADEV0016 DHUFFMAN 039010 Y 000000000000001F	1

Graph control memory - 3.30% used (330/10000)   Records 1 - 24 of 30

Graph Type: stacked vertical bar
X-axis: File and library name. The entire field is shown by placing the mouse over a bar.
Y-Axis: This value is the number of LDIO ops occurring per second per file within job. Each color represents a different job.
Second Y-Axis: This line shows how much each file/library's LDIO ops contributed to the total system LDIOs.

Example:

5.1.13 DB File Detail LDIO by Job-thread within File-library - No intervals



QZRCSRVS	QUSER	045113	Y	0000000000000001	🟥
QPADEV0024	VPKIRK	042108	Y	000000000000007A	🟩
QZDASOINIT	QUSER	045108	Y	000000000000005C	🟦
QZDASOINIT	QUSER	045112	Y	000000000000003B	🟨
QZDASOINIT	QUSER	044653	Y	00000000000000B6	■
Q1PSCH	QPM400	006308	Y	0000000000000001	

Graph control memory - 3.30% used (330/10000)

Bars 1 - 22 of 22

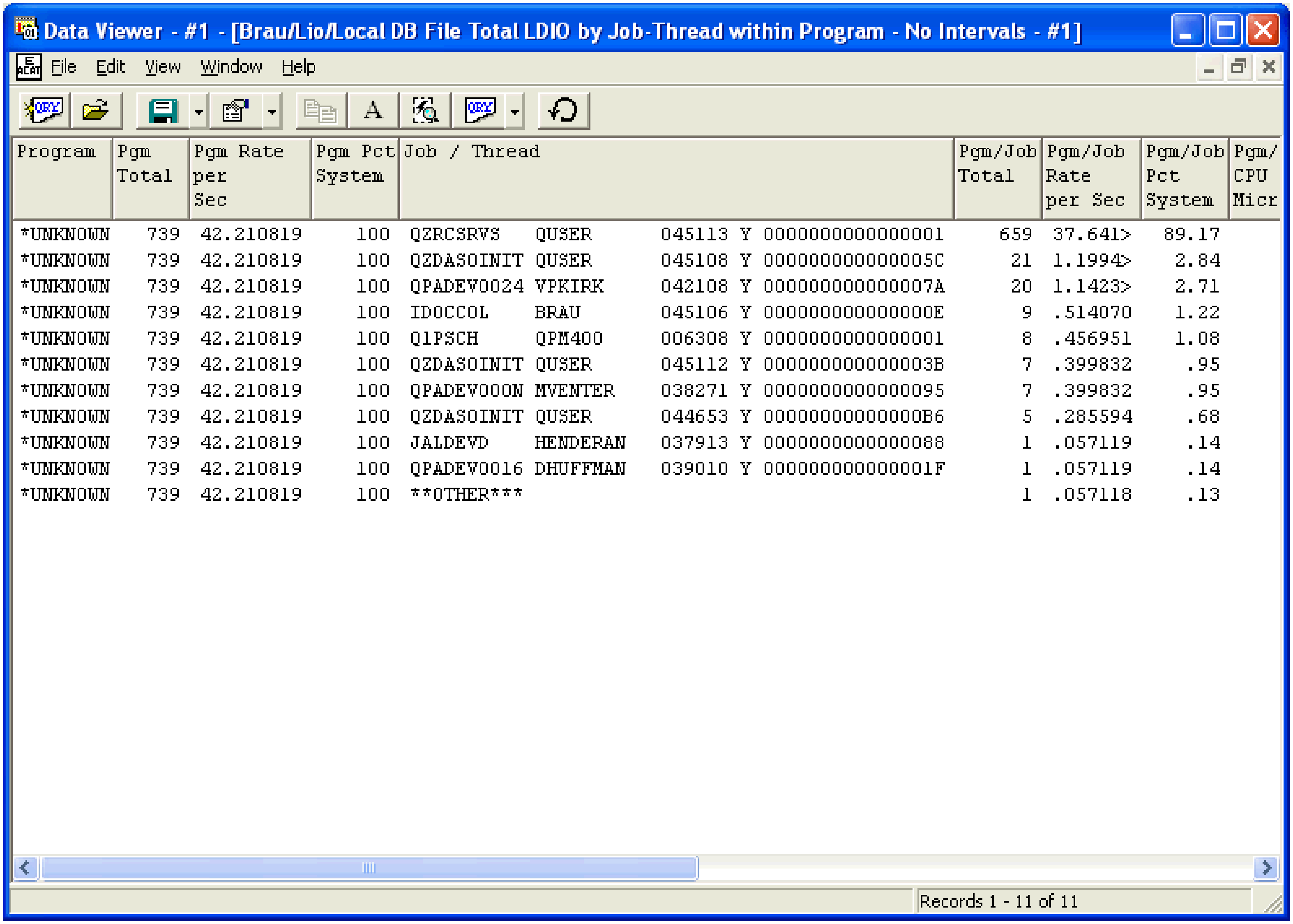
Table of Contents Previous Next

5.1.14 DB File Detail LDIO by Job-thread within program - No Intervals

Description: This report shows the program LDIO activity for each job within the collection.

Note: In order for program names to be provided, *MIENTRY and *MIEXIT events must be included in the PEX collection.

Example:

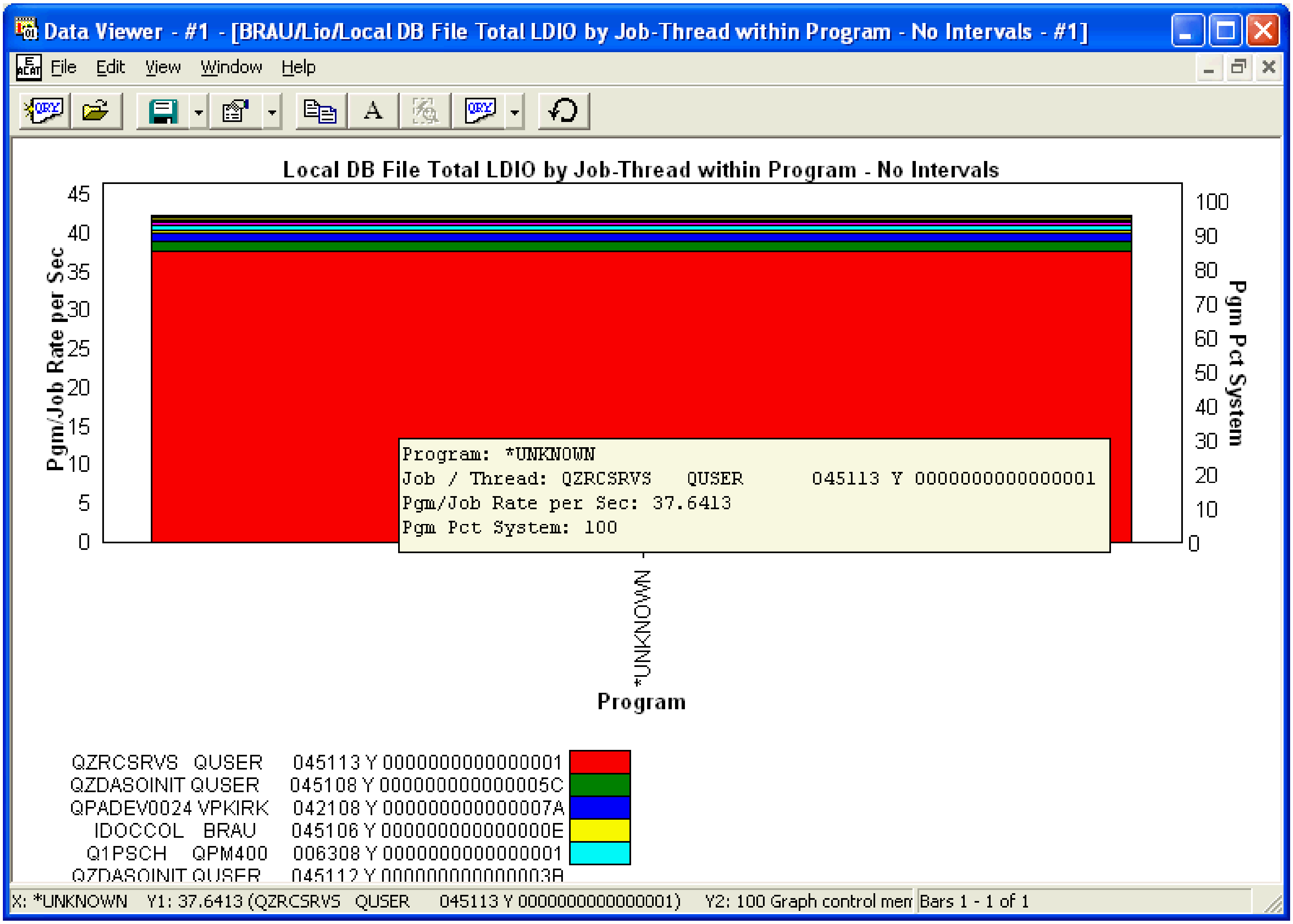

Data Viewer - #1 - [Brau/Lio/Local DB File Total LDIO by Job-Thread within Program - No Intervals - #1]
File
Edit
View
Window
Help
Records 1 - 11 of 11

Program	Pgm Total	Pgm Rate per Sec	Pgm Pct System	Job / Thread	Pgm/Job Total	Pgm/Job Rate per Sec	Pgm/Job Pct System	Pgm/ CPU Micr
*UNKNOWN	739	42.210819	100	QZRCSRVS QUSER 045113 Y 0000000000000001	659	37.641>	89.17	
*UNKNOWN	739	42.210819	100	QZDASOINIT QUSER 045108 Y 000000000000005C	21	1.1994>	2.84	
*UNKNOWN	739	42.210819	100	QPADEV0024 VPKIRK 042108 Y 000000000000007A	20	1.1423>	2.71	
*UNKNOWN	739	42.210819	100	IDOCCOL BRAU 045106 Y 000000000000000E	9	.514070	1.22	
*UNKNOWN	739	42.210819	100	Q1PSCH QPM400 006308 Y 0000000000000001	8	.456951	1.08	
*UNKNOWN	739	42.210819	100	QZDASOINIT QUSER 045112 Y 000000000000003B	7	.399832	.95	
*UNKNOWN	739	42.210819	100	QPADEV000N MVENTER 038271 Y 0000000000000095	7	.399832	.95	
*UNKNOWN	739	42.210819	100	QZDASOINIT QUSER 044653 Y 00000000000000B6	5	.285594	.68	
*UNKNOWN	739	42.210819	100	JALDEVD HENDERAN 037913 Y 0000000000000088	1	.057119	.14	
*UNKNOWN	739	42.210819	100	QPADEV0016 DHUFFMAN 039010 Y 000000000000001F	1	.057119	.14	
*UNKNOWN	739	42.210819	100	**OTHER***	1	.057118	.13	

Graph Type: stacked vertical bar
X-axis: Program name.
Y-Axis: This value is the number of LDIO ops occurring of each type per second by job within program. Each color represents a different type of job.
Second Y-Axis: This line shows how much program's LDIO ops contributed to the total system LDIOs.

Example:

5.1.14 DB File Detail LDIO by Job-thread within program - No Intervals


Data Viewer - #1 - [BRAU/Lio/Local DB File Total LDIO by Job-Thread within Program - No Intervals - #1]
File
Edit
View
Window
Help
Local DB File Total LDIO by Job-Thread within Program - No Intervals
Pgm/Job Rate per Sec
45
40
35
30
25
20
15
10
5
0
Pgm Pct System
100
90
80
70
60
50
40
30
20
10
0
*UNKNOWN
Program

```
Program: *UNKNOWN
Job / Thread: QZRCSRVS   QUSER      045113 Y 0000000000000001
Pgm/Job Rate per Sec: 37.6413
Pgm Pct System: 100
```

```
QZRCSRVS   QUSER    045113 Y 0000000000000001 🟥
QZDASOINIT QUSER    045108 Y 000000000000005C 🟩
QPADEV0024 VPKIRK   042108 Y 000000000000007A 🟦
  IDOCCOL   BRAU    045106 Y 000000000000000E 🟨
  Q1PSCH   QPM400   006308 Y 0000000000000001 🟦
QZDASOINIT QUSER    045112 Y 000000000000003B
```

X: *UNKNOWN   Y1: 37.6413 (QZRCSRVS   QUSER   045113 Y 0000000000000001)   Y2: 100 Graph control men   Bars 1 - 1 of 1

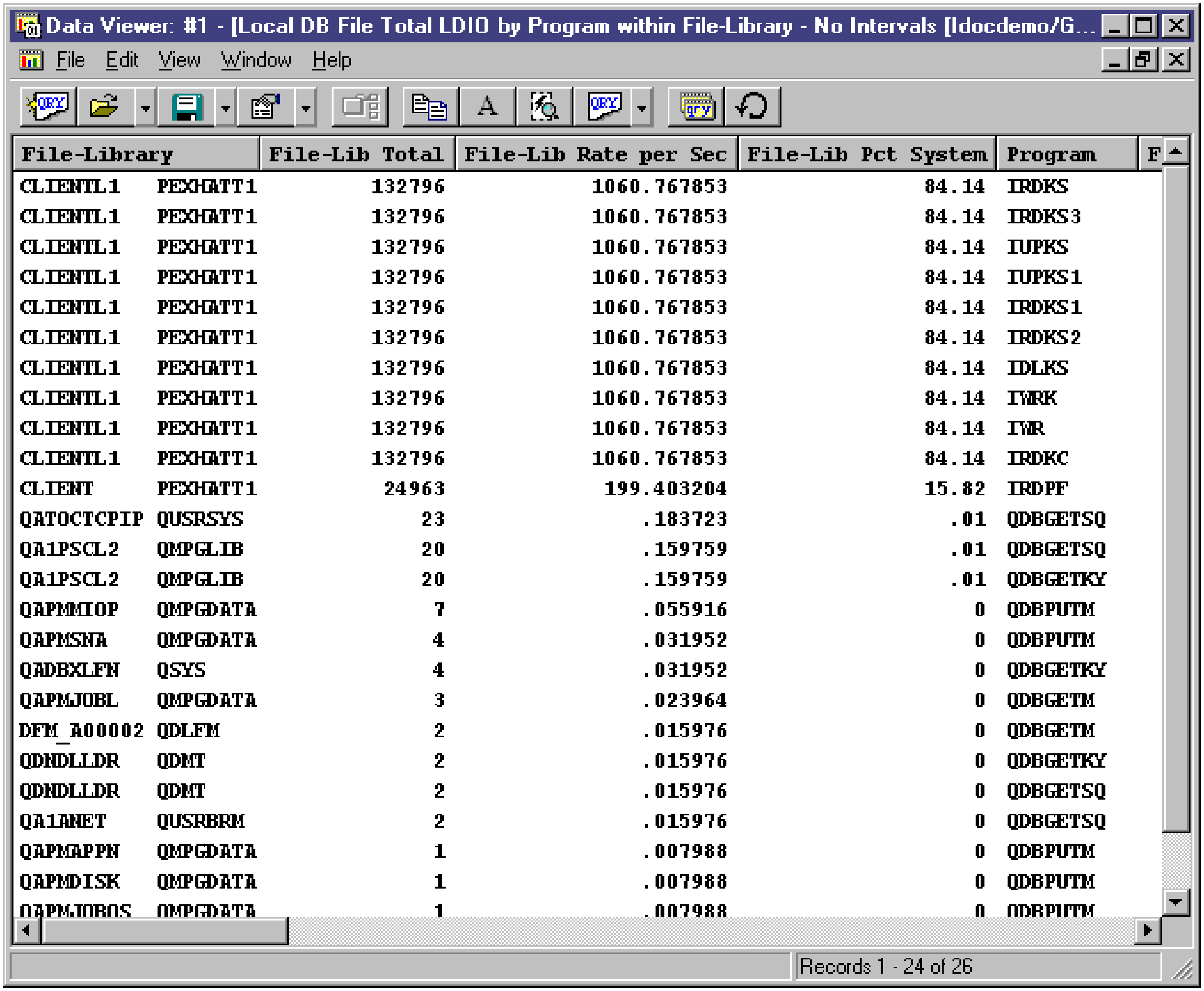
Table of Contents | Previous | Next

5.1.15 DB File Detail LDIO by Program within File-library - No Intervals

Description: This report shows the program LDIO activity for each file within the collection.

Note: In order for program names to be provided, *MIENTRY and *MIEXIT events must be included in the PEX collection.

Example:


Data Viewer: #1 - [Local DB File Total LDIO by Program within File-Library - No Intervals [Idocdemo/G...
File
Edit
View
Window
Help
QRY
QRY
qry

File-Library		File-Lib Total	File-Lib Rate per Sec	File-Lib Pct System	Program	F
CLIENTL1	PEXHATT1	132796	1060.767853	84.14	IRDKS	
CLIENTL1	PEXHATT1	132796	1060.767853	84.14	IRDKS3	
CLIENTL1	PEXHATT1	132796	1060.767853	84.14	IUPKS	
CLIENTL1	PEXHATT1	132796	1060.767853	84.14	IUPKS1	
CLIENTL1	PEXHATT1	132796	1060.767853	84.14	IRDKS1	
CLIENTL1	PEXHATT1	132796	1060.767853	84.14	IRDKS2	
CLIENTL1	PEXHATT1	132796	1060.767853	84.14	IDLKS	
CLIENTL1	PEXHATT1	132796	1060.767853	84.14	IWRK	
CLIENTL1	PEXHATT1	132796	1060.767853	84.14	IWR	
CLIENTL1	PEXHATT1	132796	1060.767853	84.14	IRDKC	
CLIENT	PEXHATT1	24963	199.403204	15.82	IRDPF	
QATOCTCPIP	QUSRSYS	23	.183723	.01	QDBGETSQ	
QA1PSCL2	QMPGLIB	20	.159759	.01	QDBGETSQ	
QA1PSCL2	QMPGLIB	20	.159759	.01	QDBGETKY	
QAPMMIOP	QMPGDATA	7	.055916	0	QDBPUTM	
QAPMSNA	QMPGDATA	4	.031952	0	QDBPUTM	
QADBXLFN	QSYS	4	.031952	0	QDBGETKY	
QAPMJOBL	QMPGDATA	3	.023964	0	QDBGETM	
DFM_A00002	QDLFM	2	.015976	0	QDBGETM	
QDNDLLDR	QDMT	2	.015976	0	QDBGETKY	
QDNDLLDR	QDMT	2	.015976	0	QDBGETSQ	
QA1ANET	QUSRBRM	2	.015976	0	QDBGETSQ	
QAPMAPPN	QMPGDATA	1	.007988	0	QDBPUTM	
QAPMDISK	QMPGDATA	1	.007988	0	QDBPUTM	
QAPMJOBOS	QMPGDATA	1	.007988	0	QDBPUTM	

Records 1 - 24 of 26

Graph Type: stacked vertical bar
X-axis: File and library name. The entire field is shown by placing the mouse over a bar.
Y-Axis: This value is the number of LDIO ops occurring per second for a file within program. Each color represents a different program.

Second Y-Axis: This line shows how much each file/library's LDIO ops contributed to the total system LDIOs.

Example:

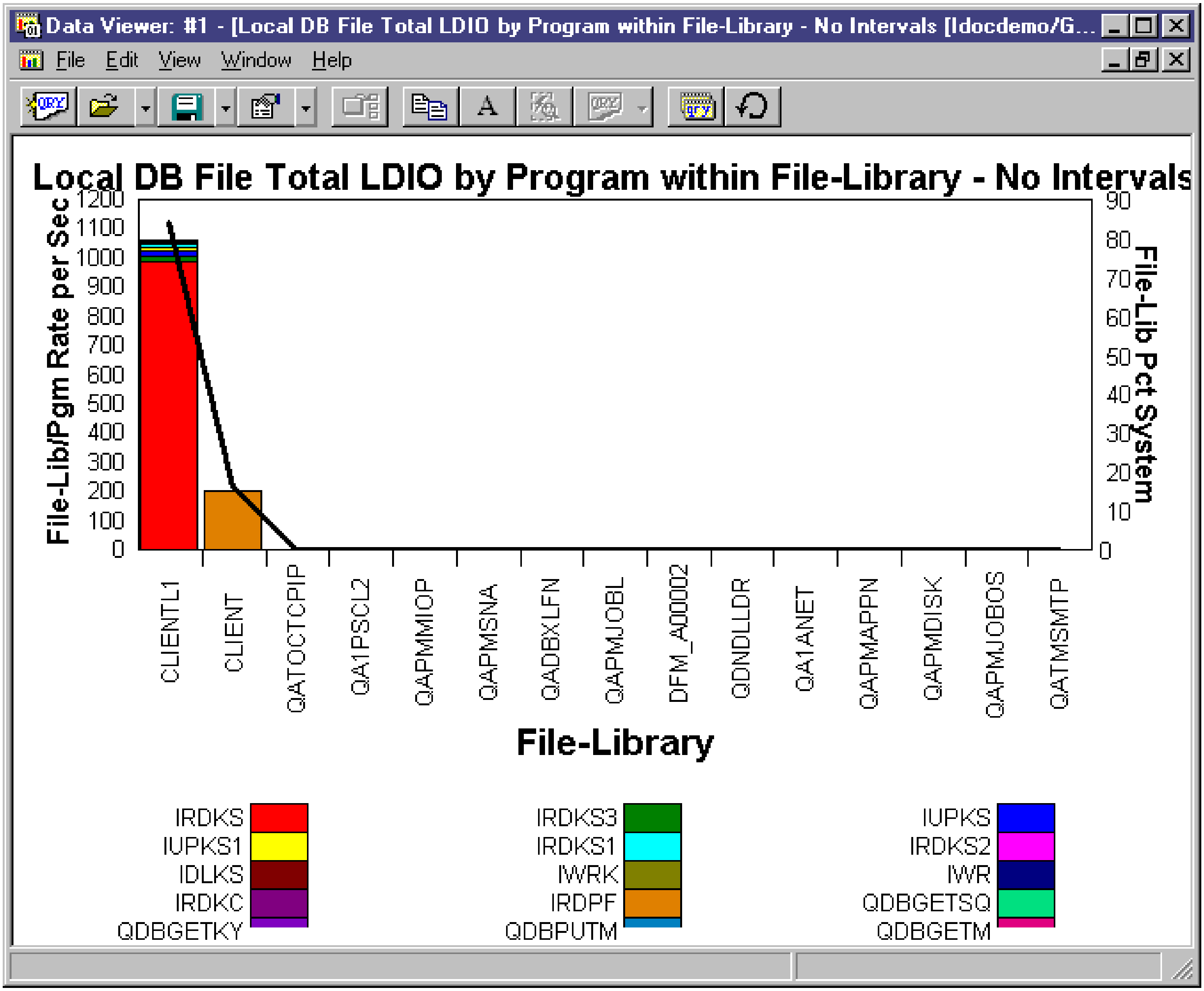


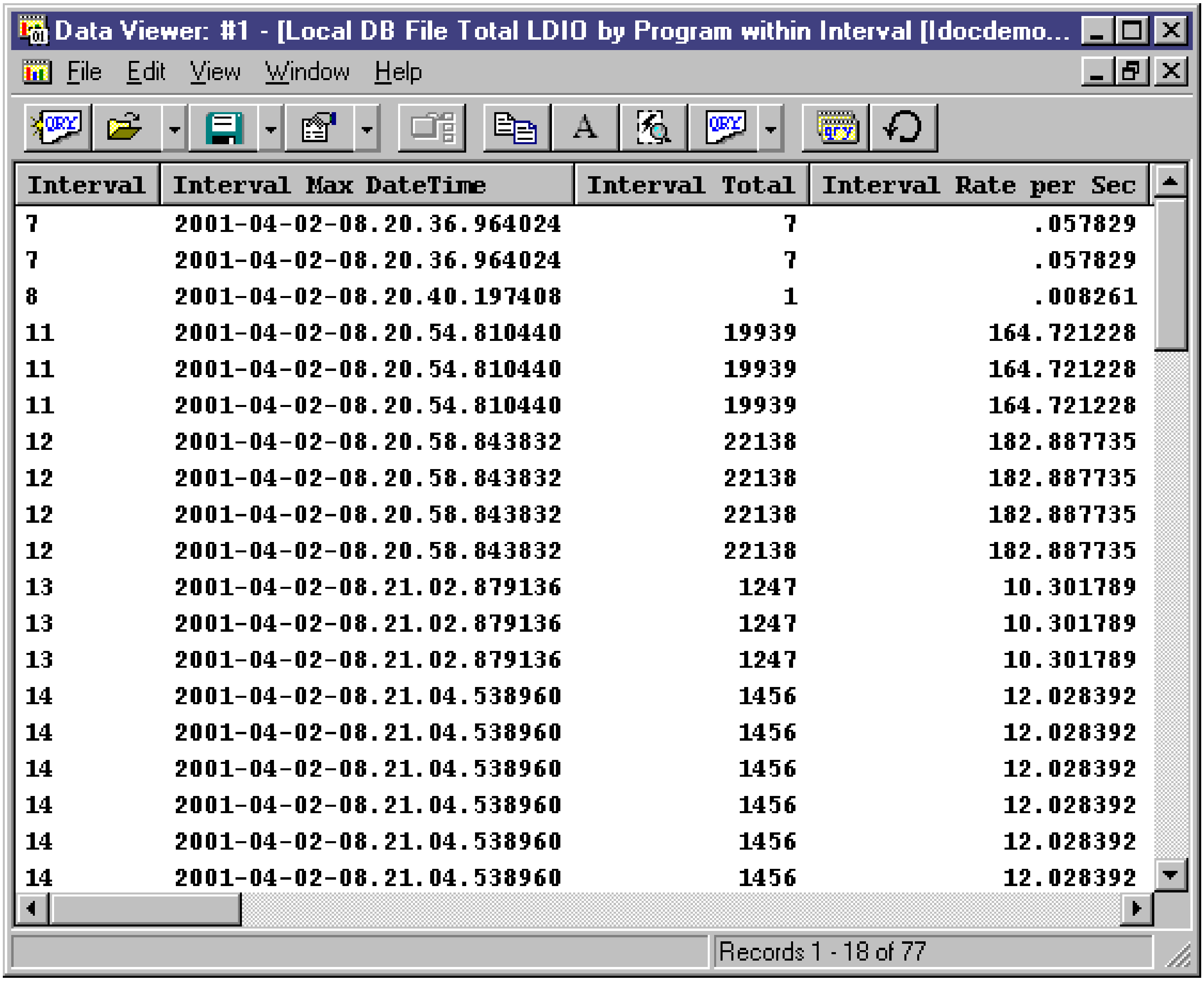
Table of Contents | Previous | Next

5.1.16 DB File Detail LDIO by Program within Interval

Description: This report shows the program LDIO activity for each interval in the collection.

Note: In order for program names to be provided, *MIENTRY and *MIEXIT events must be included in the PEX collection.

Example:


Data Viewer: #1 - [Local DB File Total LDIO by Program within Interval [Idocdemo...
File
Edit
View
Window
Help
QRY
QRY
qry
Records 1 - 18 of 77

Interval	Interval Max DateTime	Interval Total	Interval Rate per Sec
7	2001-04-02-08.20.36.964024	7	.057829
7	2001-04-02-08.20.36.964024	7	.057829
8	2001-04-02-08.20.40.197408	1	.008261
11	2001-04-02-08.20.54.810440	19939	164.721228
11	2001-04-02-08.20.54.810440	19939	164.721228
11	2001-04-02-08.20.54.810440	19939	164.721228
12	2001-04-02-08.20.58.843832	22138	182.887735
12	2001-04-02-08.20.58.843832	22138	182.887735
12	2001-04-02-08.20.58.843832	22138	182.887735
12	2001-04-02-08.20.58.843832	22138	182.887735
13	2001-04-02-08.21.02.879136	1247	10.301789
13	2001-04-02-08.21.02.879136	1247	10.301789
13	2001-04-02-08.21.02.879136	1247	10.301789
14	2001-04-02-08.21.04.538960	1456	12.028392
14	2001-04-02-08.21.04.538960	1456	12.028392
14	2001-04-02-08.21.04.538960	1456	12.028392
14	2001-04-02-08.21.04.538960	1456	12.028392
14	2001-04-02-08.21.04.538960	1456	12.028392
14	2001-04-02-08.21.04.538960	1456	12.028392

Graph Type: stacked vertical bar

X-axis: Interval number
Y-Axis: This value is the number of LDIO ops occurring per second by program within interval. Each color represents a different program
Second Y-Axis: This line shows how much each collection interval's total LDIOs contributed to the total system LDIOs.

Example:

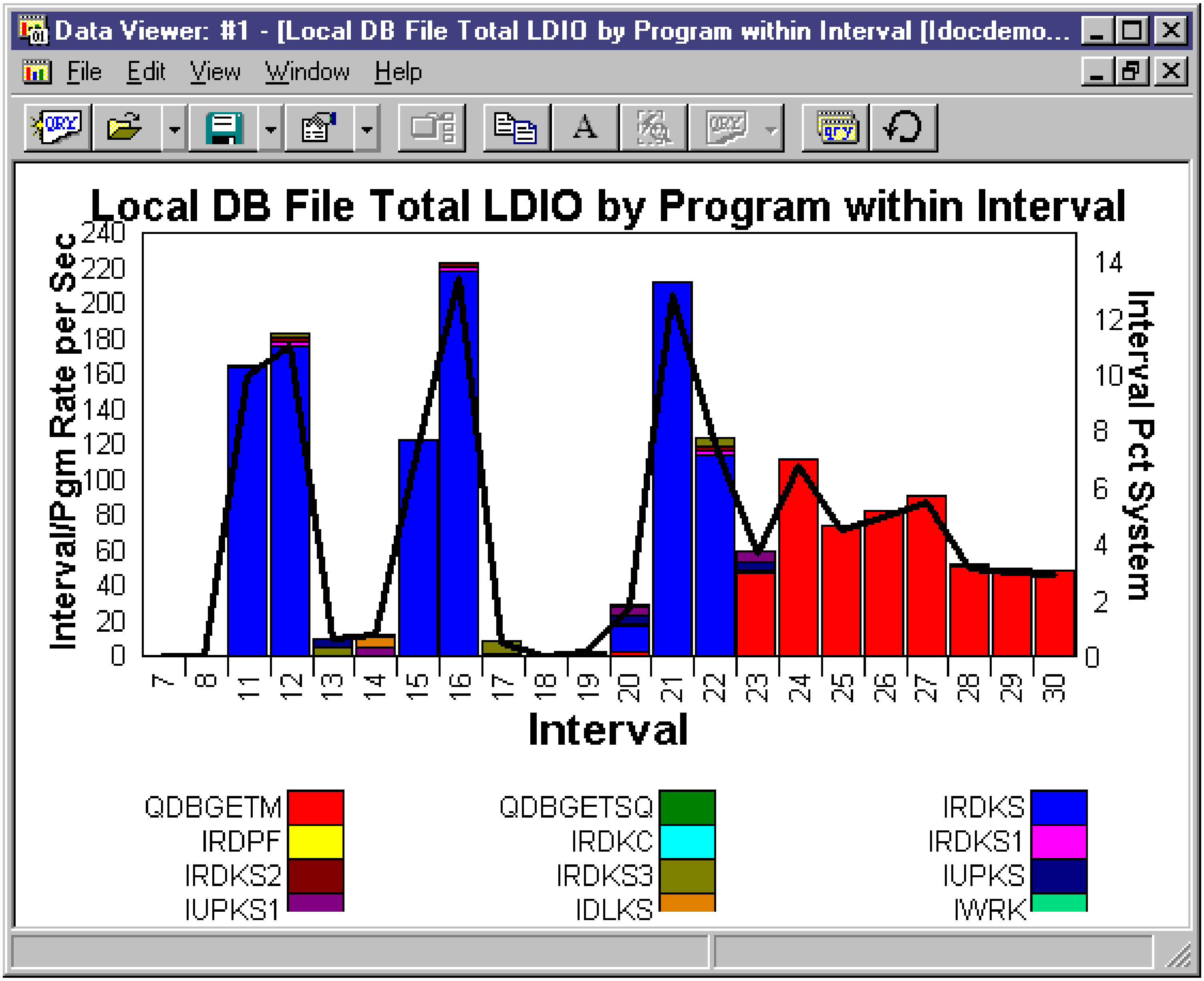


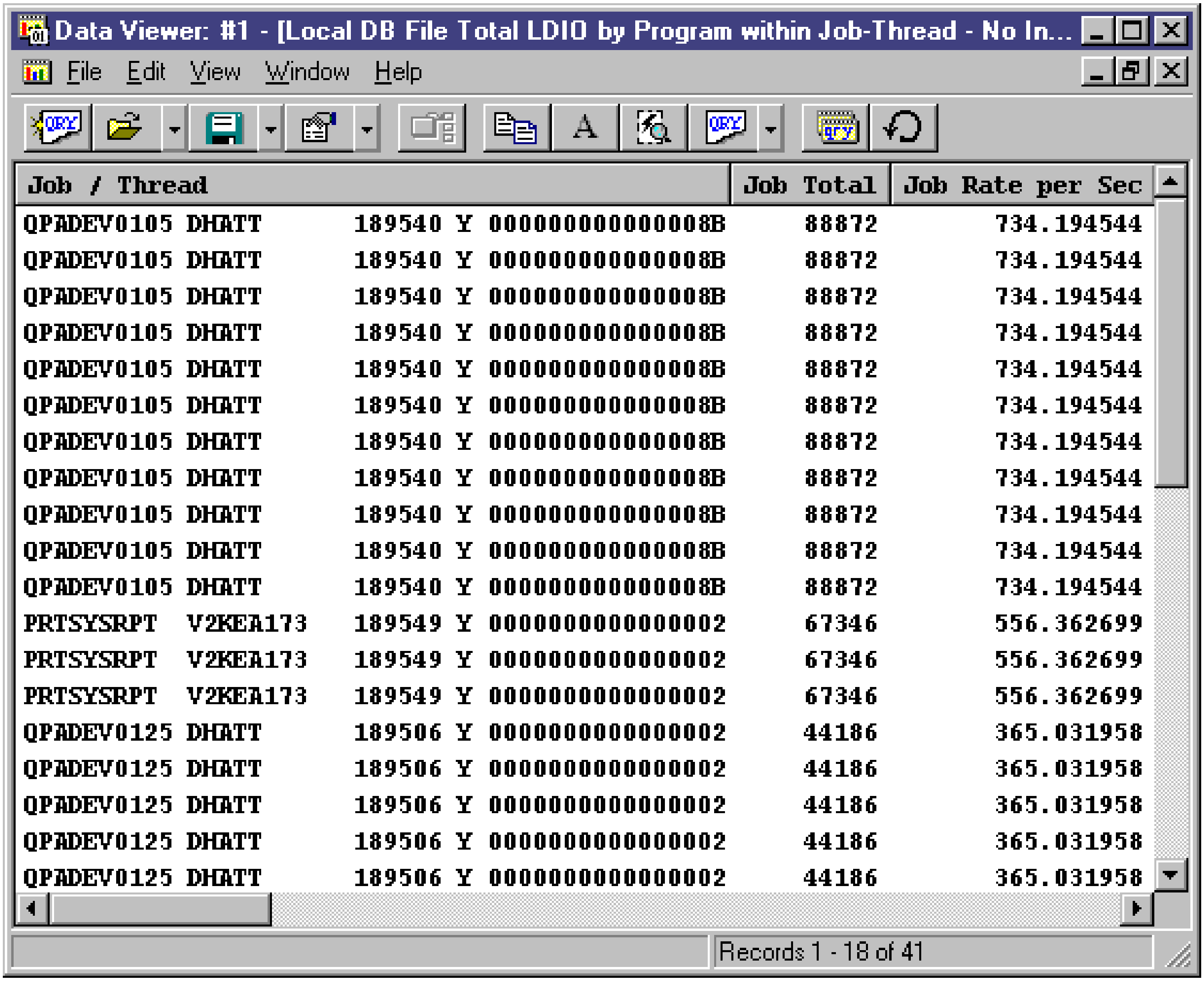
Table of Contents Previous Next

5.1.17 DB File Detail LDIO by Program within Job-thread - No Intervals

Description: This report shows the program LDIO activity for each job in the collection. Each color in the graph represents a unique program.

Note: In order for program names to be provided, *MIENTRY and *MIEXIT events must be included in the PEX collection.

Example:


Data Viewer: #1 - [Local DB File Total LDIO by Program within Job-Thread - No In...
File Edit View Window Help
Records 1 - 18 of 41

Job / Thread	Job Total	Job Rate per Sec
QPADEV0105 DHATT 189540 Y 000000000000008B	88872	734.194544
QPADEV0105 DHATT 189540 Y 000000000000008B	88872	734.194544
QPADEV0105 DHATT 189540 Y 000000000000008B	88872	734.194544
QPADEV0105 DHATT 189540 Y 000000000000008B	88872	734.194544
QPADEV0105 DHATT 189540 Y 000000000000008B	88872	734.194544
QPADEV0105 DHATT 189540 Y 000000000000008B	88872	734.194544
QPADEV0105 DHATT 189540 Y 000000000000008B	88872	734.194544
QPADEV0105 DHATT 189540 Y 000000000000008B	88872	734.194544
QPADEV0105 DHATT 189540 Y 000000000000008B	88872	734.194544
QPADEV0105 DHATT 189540 Y 000000000000008B	88872	734.194544
QPADEV0105 DHATT 189540 Y 000000000000008B	88872	734.194544
PRTSYSRPT V2KEA173 189549 Y 0000000000000002	67346	556.362699
PRTSYSRPT V2KEA173 189549 Y 0000000000000002	67346	556.362699
PRTSYSRPT V2KEA173 189549 Y 0000000000000002	67346	556.362699
QPADEV0125 DHATT 189506 Y 0000000000000002	44186	365.031958
QPADEV0125 DHATT 189506 Y 0000000000000002	44186	365.031958
QPADEV0125 DHATT 189506 Y 0000000000000002	44186	365.031958
QPADEV0125 DHATT 189506 Y 0000000000000002	44186	365.031958
QPADEV0125 DHATT 189506 Y 0000000000000002	44186	365.031958

Graph Type: stacked vertical bar

X-axis: Fully qualified job name and thread ID. The entire field is shown by placing the mouse over a bar.
Y-Axis: This value is the number of LDIO ops occurring per second for each program within job.. Each color represents a different program.
Second Y-Axis: This line shows how much each job's LDIO ops contributed to the total system LDIOs.

Example:

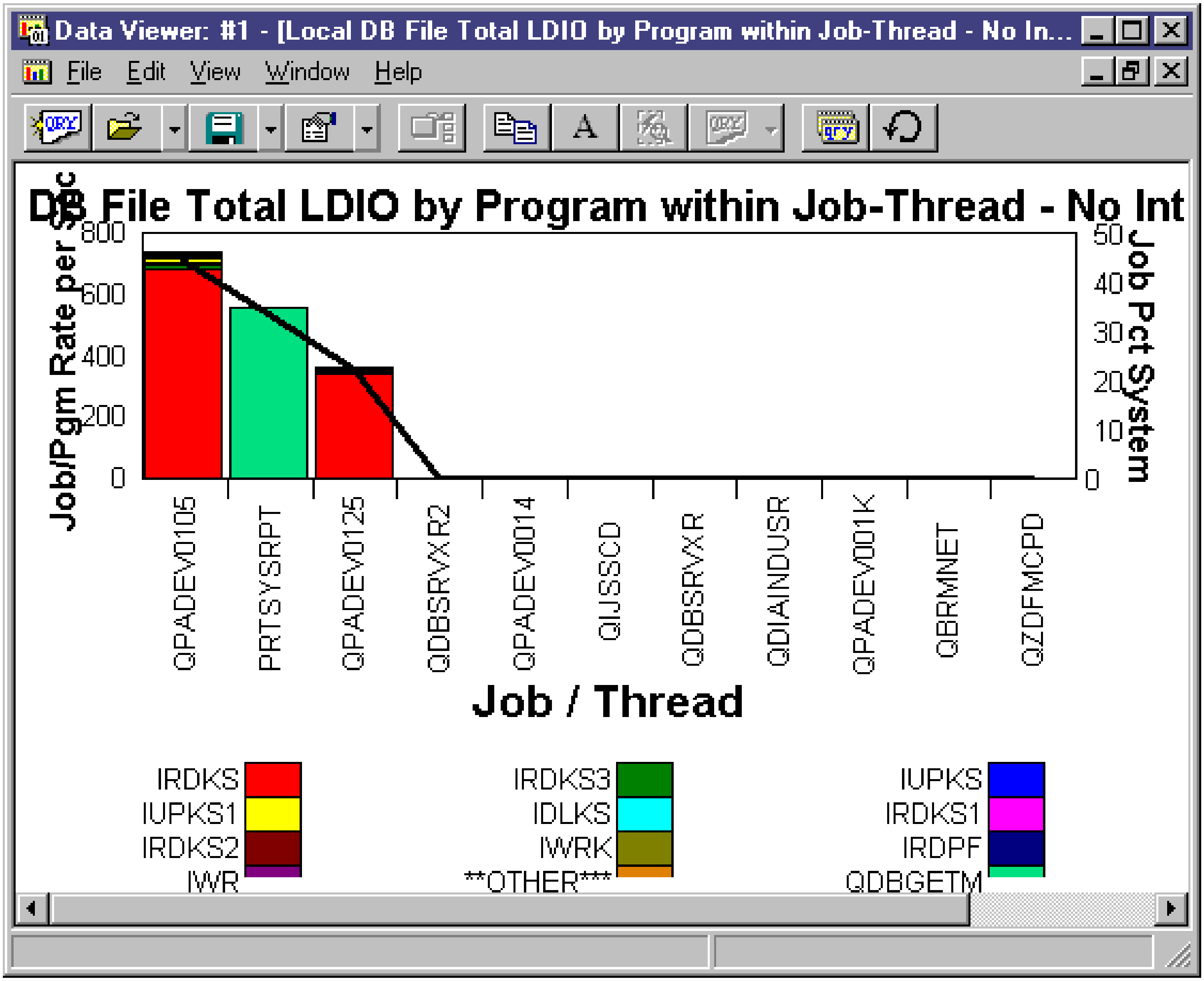


Table of Contents | Previous | Next

5.2 CPU Profile by job priority reports

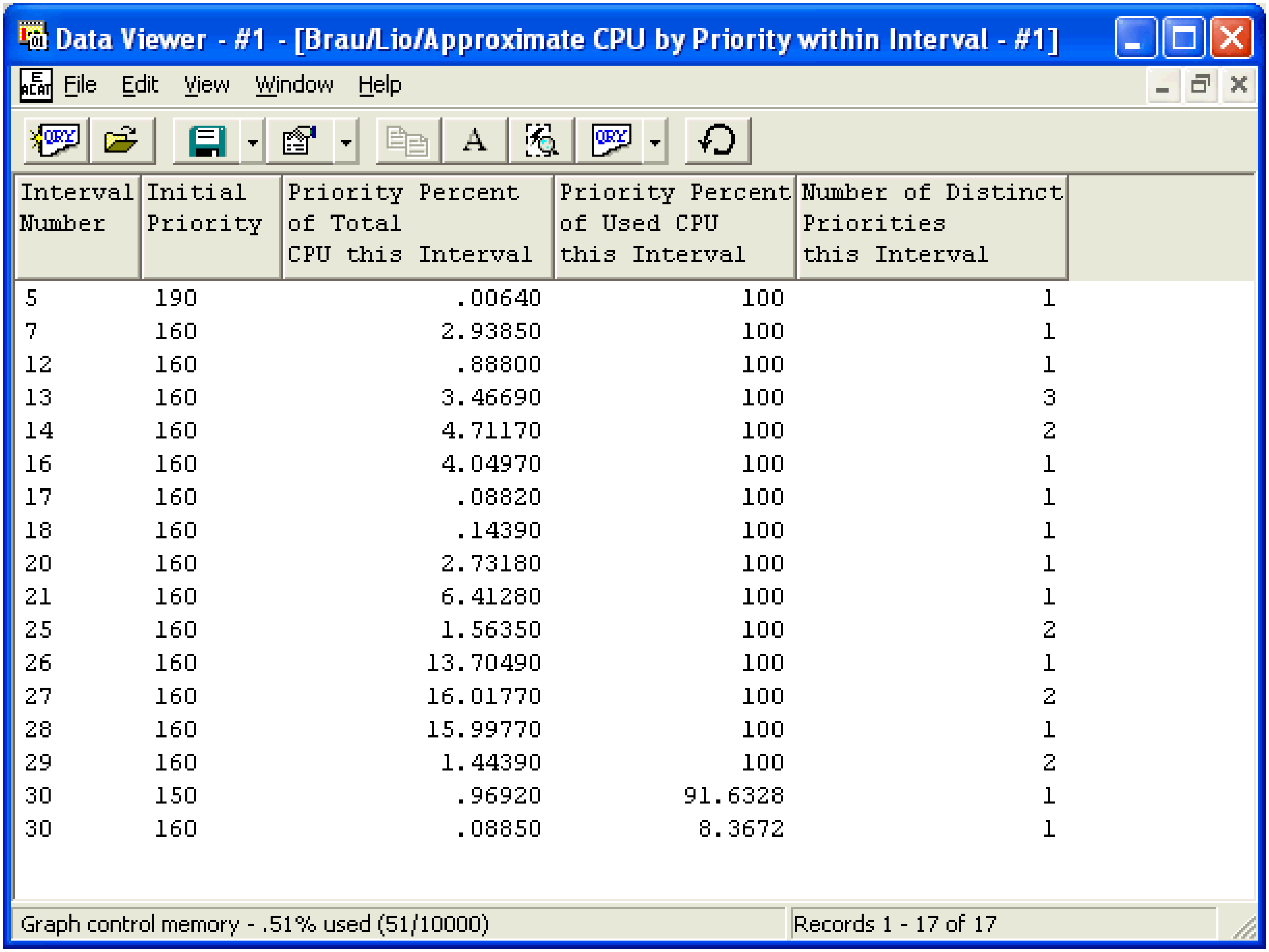
This analysis shows the job/task CPU usage percentage by time interval. Alternately you can view the CPU usage by job priority over time.

Table of Contents Previous Next

5.2.1 Approximate CPU by priority within Interval

Description: This report shows the approximate cpu usage by priority within an interval. The user can subset the analysis to any time and divide the subset time into many intervals and observe the CPU usage for each interval of time. Each priority is color coded to take a particular color. The user can get additional information by clicking on any bar in the graph. This information includes interval number, priority, percent of CPU usage, CPU percent usage in this interval, and the number of different job priorities found.

Example:


Data Viewer - #1 - [Brau/Lio/Approximate CPU by Priority within Interval - #1]
File
Edit
View
Window
Help

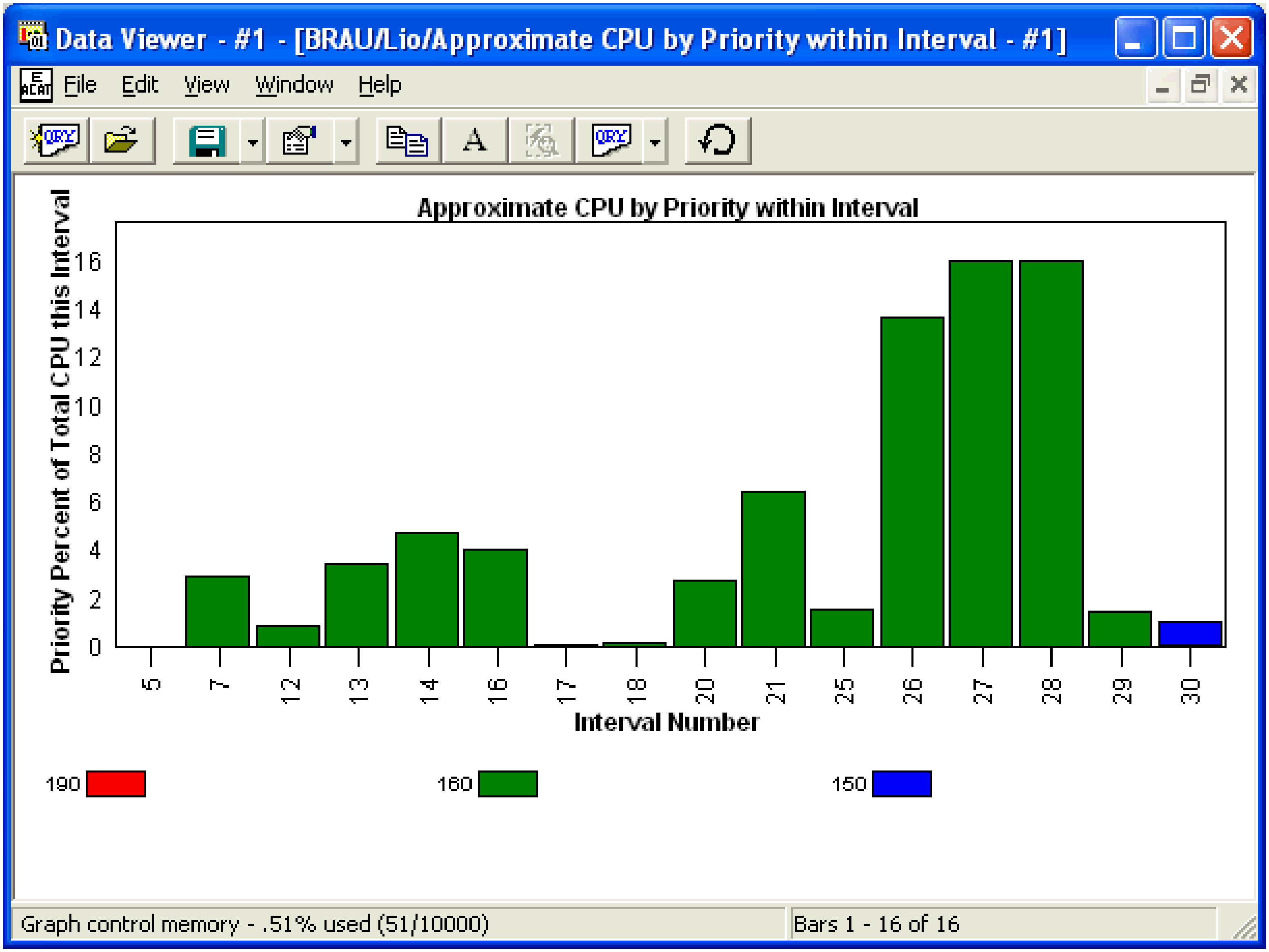
Interval Number	Initial Priority	Priority Percent of Total CPU this Interval	Priority Percent of Used CPU this Interval	Number of Distinct Priorities this Interval
5	190	.00640	100	1
7	160	2.93850	100	1
12	160	.88800	100	1
13	160	3.46690	100	3
14	160	4.71170	100	2
16	160	4.04970	100	1
17	160	.08820	100	1
18	160	.14390	100	1
20	160	2.73180	100	1
21	160	6.41280	100	1
25	160	1.56350	100	2
26	160	13.70490	100	1
27	160	16.01770	100	2
28	160	15.99770	100	1
29	160	1.44390	100	2
30	150	.96920	91.6328	1
30	160	.08850	8.3672	1

Graph control memory - .51% used (51/10000)

Records 1 - 17 of 17

Graph Type: stacked vertical bar
X-axis: Interval number
Y-Axis: This value is the approximate percentage of the total system CPU used by a priority value. Each priority value is represented by a different color.

Example:


Data Viewer - #1 - [BRAU/Lio/Approximate CPU by Priority within Interval - #1]
File
Edit
View
Window
Help
Approximate CPU by Priority within Interval
Priority Percent of Total CPU this Interval
0
2
4
6
8
10
12
14
16
5
7
12
13
14
16
17
18
20
21
25
26
27
28
29
30
Interval Number
190 🟥
160 🟩
150 🟦

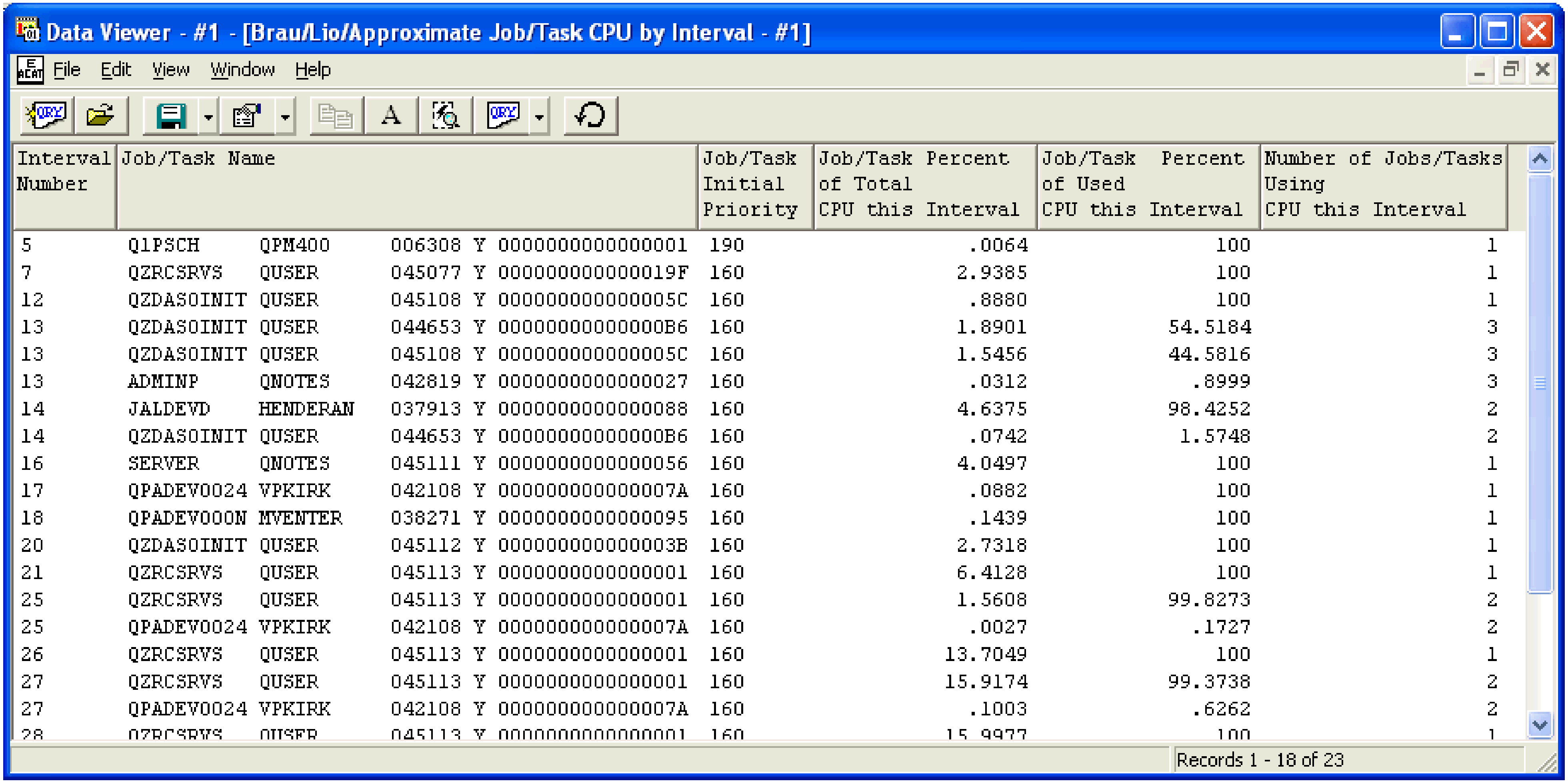
Graph control memory - .51% used (51/10000)

Bars 1 - 16 of 16

Table of Contents | Previous | Next

5.2.2 Approximate Job/Task CPU by interval

This report displays information about the CPU usage of jobs for each interval within the PEX collection.


Data Viewer - #1 - [Brau/Lio/Approximate Job/Task CPU by Interval - #1]
File
Edit
View
Window
Help

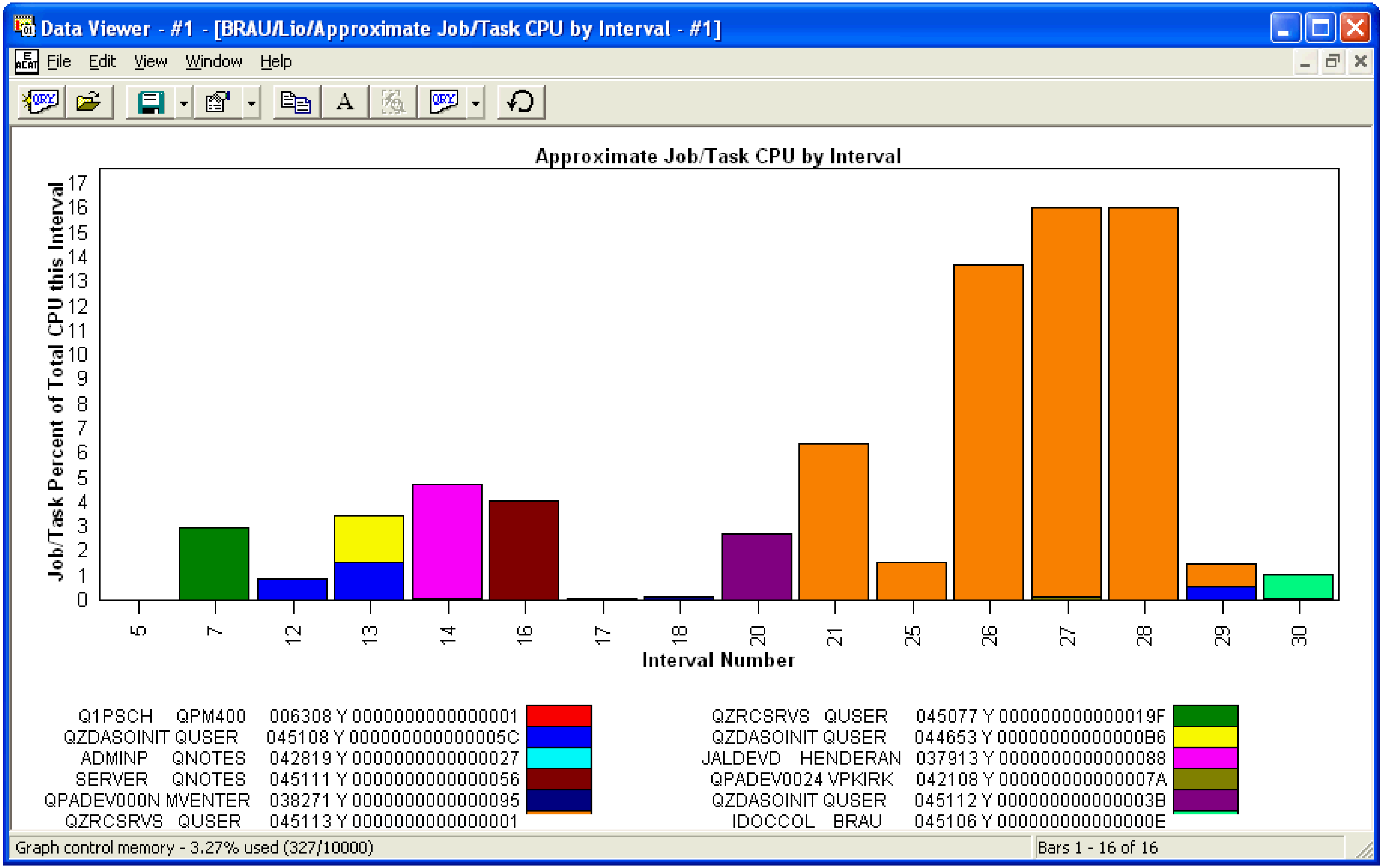
Interval Number	Job/Task Name	Job/Task Initial Priority	Job/Task Percent of Total CPU this Interval	Job/Task Percent of Used CPU this Interval	Number of Jobs/Tasks Using CPU this Interval
5	Q1PSCH QPM400 006308 Y 0000000000000001	190	.0064	100	1
7	QZRCSRVS QUSER 045077 Y 000000000000019F	160	2.9385	100	1
12	QZDASOINIT QUSER 045108 Y 000000000000005C	160	.8880	100	1
13	QZDASOINIT QUSER 044653 Y 00000000000000B6	160	1.8901	54.5184	3
13	QZDASOINIT QUSER 045108 Y 000000000000005C	160	1.5456	44.5816	3
13	ADMINP QNOTES 042819 Y 0000000000000027	160	.0312	.8999	3
14	JALDEVD HENDERAN 037913 Y 0000000000000088	160	4.6375	98.4252	2
14	QZDASOINIT QUSER 044653 Y 00000000000000B6	160	.0742	1.5748	2
16	SERVER QNOTES 045111 Y 0000000000000056	160	4.0497	100	1
17	QPADEV0024 VPKIRK 042108 Y 000000000000007A	160	.0882	100	1
18	QPADEV000N MVENTER 038271 Y 0000000000000095	160	.1439	100	1
20	QZDASOINIT QUSER 045112 Y 000000000000003B	160	2.7318	100	1
21	QZRCSRVS QUSER 045113 Y 0000000000000001	160	6.4128	100	1
25	QZRCSRVS QUSER 045113 Y 0000000000000001	160	1.5608	99.8273	2
25	QPADEV0024 VPKIRK 042108 Y 000000000000007A	160	.0027	.1727	2
26	QZRCSRVS QUSER 045113 Y 0000000000000001	160	13.7049	100	1
27	QZRCSRVS QUSER 045113 Y 0000000000000001	160	15.9174	99.3738	2
27	QPADEV0024 VPKIRK 042108 Y 000000000000007A	160	.1003	.6262	2
28	QZRCSRVS QUSER 045113 Y 0000000000000001	160	15.9977	100	1

Records 1 - 18 of 23

Graph Type: stacked vertical bar
X-axis: Interval number
Y-Axis: This value is the approximate percentage of total CPU used by a job each interval. Each color represents a different job.

Example:

5.2.2 Approximate Job/Task CPU by interval


Data Viewer - #1 - [BRAU/Lio/Approximate Job/Task CPU by Interval - #1]
File
Edit
View
Window
Help
Approximate Job/Task CPU by Interval
Job/Task Percent of Total CPU this Interval
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
5
7
12
13
14
16
17
18
20
21
25
26
27
28
29
30
Interval Number

Q1PSCH	QPM400	006308	Y	0000000000000001	🟥	QZRCSRVS	QUSER	045077	Y	000000000000019F	🟩
QZDASOINIT	QUSER	045108	Y	000000000000005C	🟦	QZDASOINIT	QUSER	044653	Y	00000000000000B6	🟨
ADMINP	QNOTES	042819	Y	0000000000000027	🟦	JALDEVD	HENDERAN	037913	Y	0000000000000088	🟪
SERVER	QNOTES	045111	Y	0000000000000056	🟫	QPADEV0024	VPKIRK	042108	Y	000000000000007A	🟫
QPADEV000N	MVENTER	038271	Y	0000000000000095	🟦	QZDASOINIT	QUSER	045112	Y	000000000000003B	🟪
QZRCSRVS	QUSER	045113	Y	0000000000000001	🟧	IDOCCOL	BRAU	045106	Y	000000000000000E	🟩

Graph control memory - 3.27% used (327/10000)

Bars 1 - 16 of 16

Table of Contents Previous Next

5.3 CPU Profile Summary (TPROF) reports

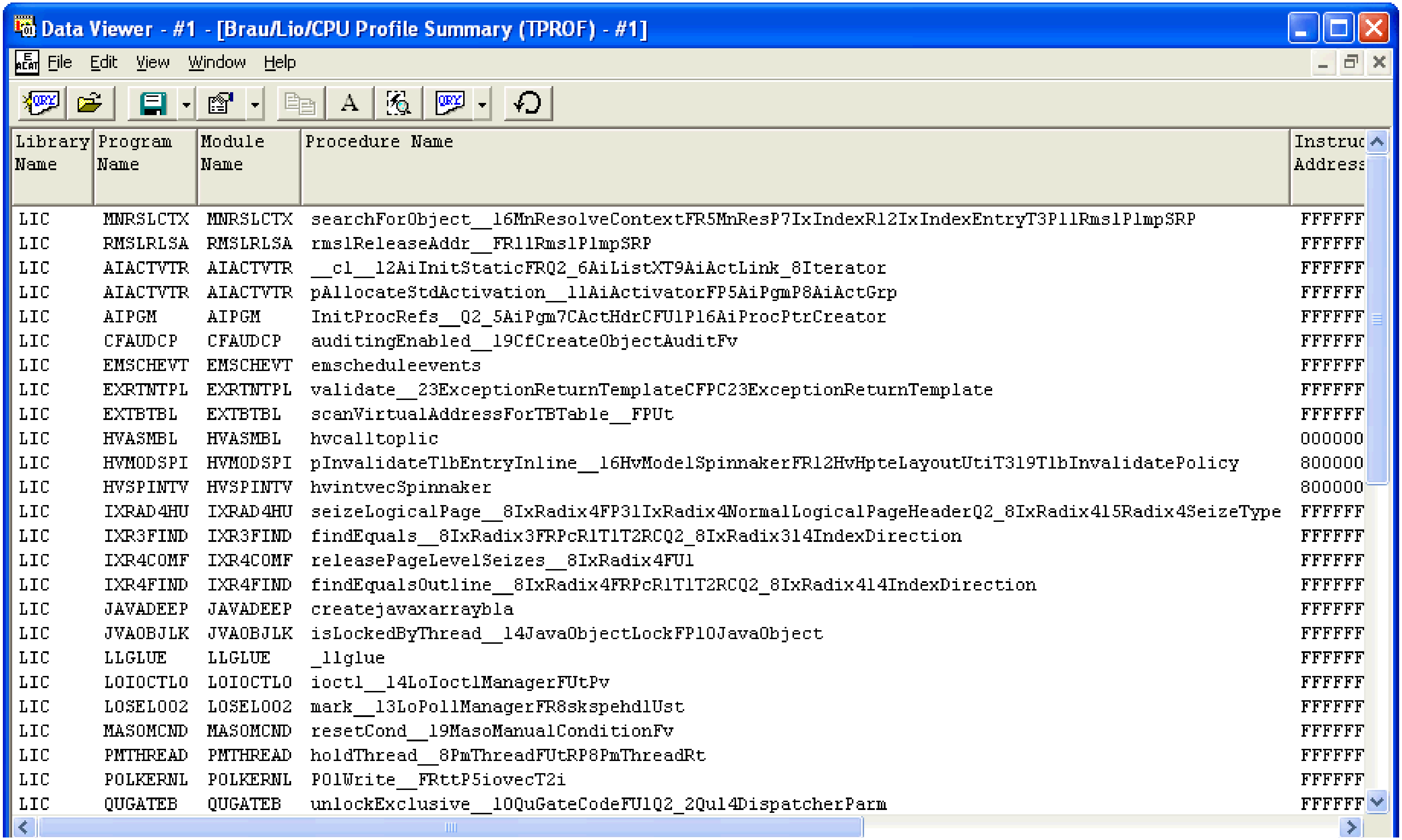
The TPROF analysis shows CPU usage for ALL types of executable code, including SLIC programs or "unhooked" programs, running in the selected job/task(s).

TPROF shows the run time CPU execution "hot spots" and indicates what programs are being used that are consuming the CPU.

5.3.1 CPU Profile Summary (TPROF)

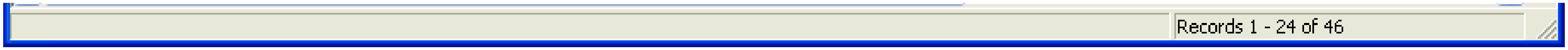
This report shows the procedures/programs and modules that used the most CPU in a PEX trace collection.

PMCO events much be collected in order to produce this analysis report.

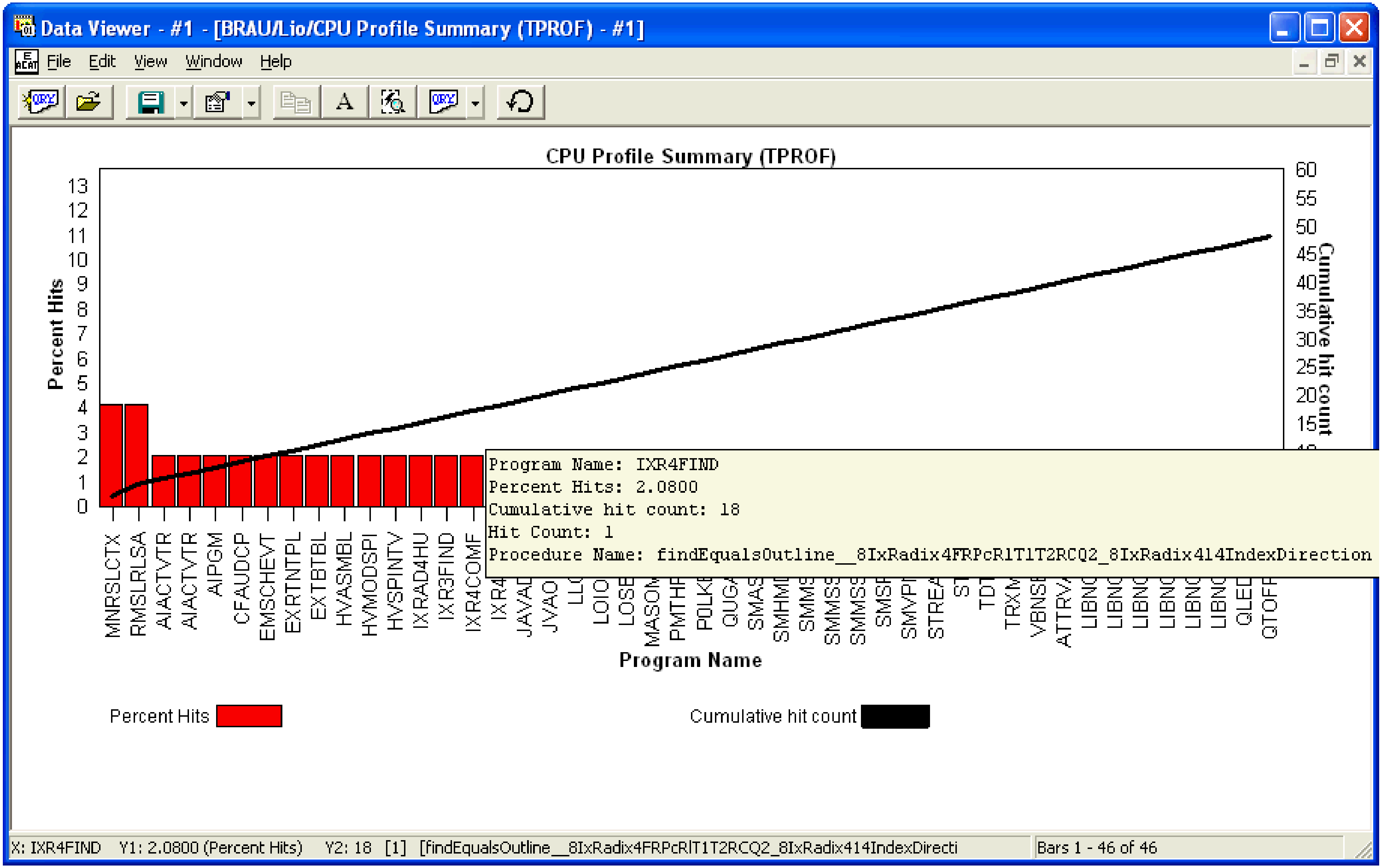


Data Viewer - #1 - [Brau/Lio/CPU Profile Summary (TPROF) - #1]
File
Edit
View
Window
Help

Library Name	Program Name	Module Name	Procedure Name	Instruc Address
LIC	MNRSLCTX	MNRSLCTX	searchForObject__16MnResolveContextFR5MnResP7IxIndexR12IxIndexEntryT3P11Rms1PlmpSRP	FFFFFF
LIC	RMSLRLSA	RMSLRLSA	rmslReleaseAddr__FR11RmslPlmpSRP	FFFFFF
LIC	AIACTVTR	AIACTVTR	__cl__12AiInitStaticFRQ2_6AiListXT9AiActLink_8Iterator	FFFFFF
LIC	AIACTVTR	AIACTVTR	pAllocateStdActivation__11AiActivatorFP5AiPgmP8AiActGrp	FFFFFF
LIC	AIPGM	AIPGM	InitProcRefs__Q2_5AiPgm7CActHdrCFUlP16AiProcPtrCreator	FFFFFF
LIC	CFAUDCP	CFAUDCP	auditingEnabled__19CfCreateObjectAuditFv	FFFFFF
LIC	EMSCHEVT	EMSCHEVT	emscheduleevents	FFFFFF
LIC	EXRTNTPL	EXRTNTPL	validate__23ExceptionReturnTemplateCFPC23ExceptionReturnTemplate	FFFFFF
LIC	EXTBTBL	EXTBTBL	scanVirtualAddressForTBTable__FPUt	FFFFFF
LIC	HVASMBL	HVASMBL	hvcalltoplic	000000
LIC	HVMODSPI	HVMODSPI	pInvalidateTlbEntryInline__16HvModelSpinnakerFR12HvHpteLayoutUtiT319TlbInvalidatePolicy	800000
LIC	HVSPINTV	HVSPINTV	hvintvecSpinnaker	800000
LIC	IXRAD4HU	IXRAD4HU	seizeLogicalPage__8IxRadix4FP31IxRadix4NormalLogicalPageHeaderQ2_8IxRadix415Radix4SeizeType	FFFFFF
LIC	IXR3FIND	IXR3FIND	findEquals__8IxRadix3FRPcR1T1T2RCQ2_8IxRadix314IndexDirection	FFFFFF
LIC	IXR4COMF	IXR4COMF	releasePageLevelSeizes__8IxRadix4FUl	FFFFFF
LIC	IXR4FIND	IXR4FIND	findEqualsOutline__8IxRadix4FRPcR1T1T2RCQ2_8IxRadix414IndexDirection	FFFFFF
LIC	JAVADEEP	JAVADEEP	createjavaxarraybla	FFFFFF
LIC	JVAOBJLK	JVAOBJLK	isLockedByThread__14JavaObjectLockFP10JavaObject	FFFFFF
LIC	LLGLUE	LLGLUE	_llglue	FFFFFF
LIC	LOIOCTLO	LOIOCTLO	ioctl__14LoIoctlManagerFUtPv	FFFFFF
LIC	LOSEL002	LOSEL002	mark__13LoPollManagerFR8skspehdlUst	FFFFFF
LIC	MASOMCND	MASOMCND	resetCond__19MasoManualConditionFv	FFFFFF
LIC	PMTHREAD	PMTHREAD	holdThread__8PmThreadFUtRP8PmThreadRt	FFFFFF
LIC	POLKERNL	POLKERNL	P01Write__FRttP5iovecT2i	FFFFFF
LIC	QUGATEB	QUGATEB	unlockExclusive__10QuGateCodeFUlQ2_2Qu14DispatcherParm	FFFFFF



Records 1 - 24 of 46



Data Viewer - #1 - [BRAU/Lio/CPU Profile Summary (TPROF) - #1]

```
Program Name: IXR4FIND
Percent Hits: 2.0800
Cumulative hit count: 18
Hit Count: 1
Procedure Name: findEqualsOutline__8IxRadix4FRPcR1T1T2RCQ2_8IxRadix414IndexDirection
```

X: IXR4FIND Y1: 2.0800 (Percent Hits) Y2: 18 [1] [findEqualsOutline__8IxRadix4FRPcRlT1T2RCQ2_8IxRadix414IndexDirecti

Bars 1 - 46 of 46

Table of Contents Previous Next

5.4 Events count reports

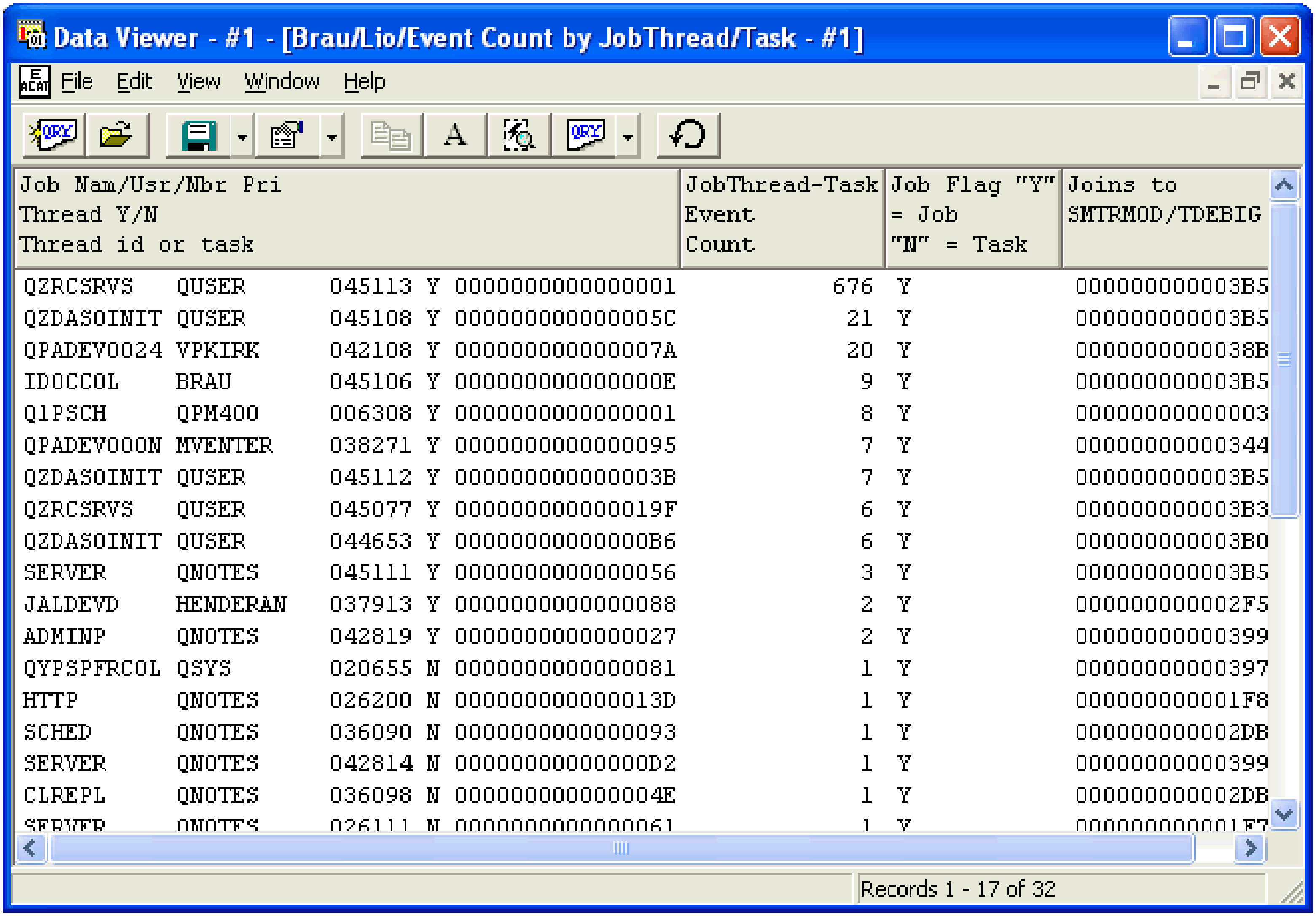
This analysis provides a breakdown of the number of PEX events that were collected per job/task in the PEX collection.

In addition to seeing the total number of PEX events per job/task, you can also see the totals for each event type/subtype combination per job/task.

Table of Contents | Previous | Next

5.4.1 Event Count by JobThread/Task

This report shows how many PEX events were captured for each job in the PEX collection.



Data Viewer - #1 - [Brau/Lio/Event Count by JobThread/Task - #1]
File
Edit
View
Window
Help
QRY
A

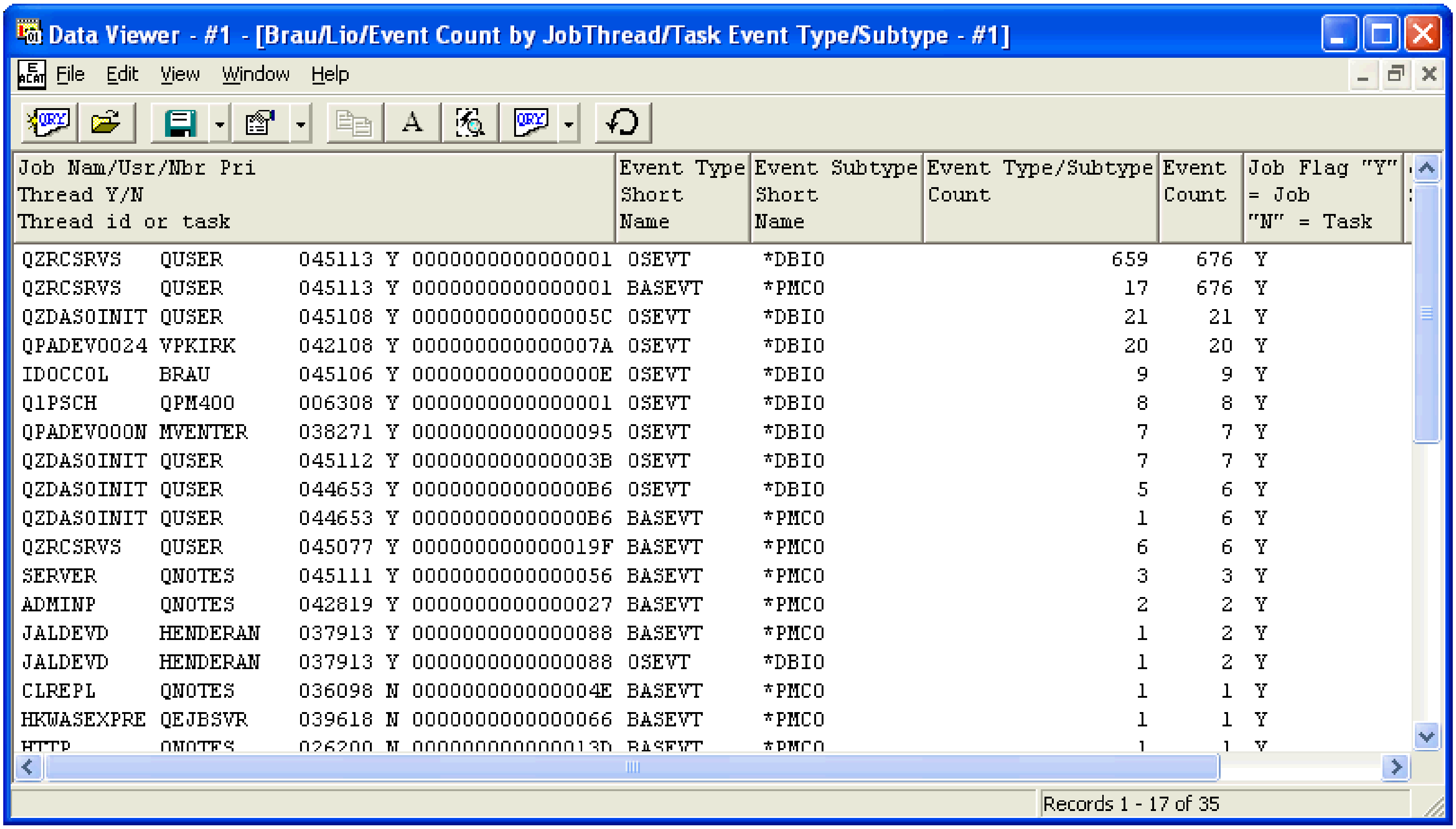
Job Nam/Usr/Nbr Pri Thread Y/N Thread id or task	JobThread-Task Event Count	Job Flag "Y" = Job "N" = Task	Joins to SMTRMOD/TDEBIG
QZRCSRVS QUSER 045113 Y 0000000000000001	676	Y	000000000003B5
QZDASOINIT QUSER 045108 Y 000000000000005C	21	Y	000000000003B5
QPADEV0024 VPKIRK 042108 Y 000000000000007A	20	Y	000000000038B
IDOCCOL BRAU 045106 Y 000000000000000E	9	Y	000000000003B5
Q1PSCH QPM400 006308 Y 0000000000000001	8	Y	00000000000003
QPADEV000N MVENTER 038271 Y 0000000000000095	7	Y	0000000000344
QZDASOINIT QUSER 045112 Y 000000000000003B	7	Y	000000000003B5
QZRCSRVS QUSER 045077 Y 000000000000019F	6	Y	000000000003B3
QZDASOINIT QUSER 044653 Y 00000000000000B6	6	Y	000000000003B0
SERVER QNOTES 045111 Y 0000000000000056	3	Y	000000000003B5
JALDEVD HENDERAN 037913 Y 0000000000000088	2	Y	000000000002F5
ADMINP QNOTES 042819 Y 0000000000000027	2	Y	0000000000399
QYPSPFRCOL QSYS 020655 N 0000000000000081	1	Y	0000000000397
HTTP QNOTES 026200 N 000000000000013D	1	Y	000000000001F8
SCHED QNOTES 036090 N 0000000000000093	1	Y	000000000002DB
SERVER QNOTES 042814 N 00000000000000D2	1	Y	0000000000399
CLREPL QNOTES 036098 N 000000000000004E	1	Y	000000000002DB
SERVER QNOTES 026111 N 0000000000000061	1	Y	000000000001F7

Records 1 - 17 of 32

Table of Contents Previous Next

5.4.2 Event Count by JobThread/Task Event Type/Subtype

This report shows how many PEX events were captured of each type for each job in the PEX collection.


Data Viewer - #1 - [Brau/Lio/Event Count by JobThread/Task Event Type/Subtype - #1]
File
Edit
View
Window
Help
QRY
A
QRY

Job Nam/Usr/Nbr Pri Thread Y/N Thread id or task	Event Type Short Name	Event Subtype Short Name	Event Type/Subtype Count	Event Count	Job Flag "Y" = Job "N" = Task
QZRCSRVS QUSER 045113 Y 0000000000000001	OSEVT	*DBIO	659	676	Y
QZRCSRVS QUSER 045113 Y 0000000000000001	BASEVT	*PMCO	17	676	Y
QZDASOINIT QUSER 045108 Y 000000000000005C	OSEVT	*DBIO	21	21	Y
QPADEV0024 VPKIRK 042108 Y 000000000000007A	OSEVT	*DBIO	20	20	Y
IDOCCOL BRAU 045106 Y 000000000000000E	OSEVT	*DBIO	9	9	Y
Q1PSCH QPM400 006308 Y 0000000000000001	OSEVT	*DBIO	8	8	Y
QPADEV000N MVENTER 038271 Y 0000000000000095	OSEVT	*DBIO	7	7	Y
QZDASOINIT QUSER 045112 Y 000000000000003B	OSEVT	*DBIO	7	7	Y
QZDASOINIT QUSER 044653 Y 00000000000000B6	OSEVT	*DBIO	5	6	Y
QZDASOINIT QUSER 044653 Y 00000000000000B6	BASEVT	*PMCO	1	6	Y
QZRCSRVS QUSER 045077 Y 000000000000019F	BASEVT	*PMCO	6	6	Y
SERVER QNOTES 045111 Y 0000000000000056	BASEVT	*PMCO	3	3	Y
ADMINP QNOTES 042819 Y 0000000000000027	BASEVT	*PMCO	2	2	Y
JALDEVD HENDERAN 037913 Y 0000000000000088	BASEVT	*PMCO	1	2	Y
JALDEVD HENDERAN 037913 Y 0000000000000088	OSEVT	*DBIO	1	2	Y
CLREPL QNOTES 036098 N 000000000000004E	BASEVT	*PMCO	1	1	Y
HKWASEXPRE QEJBSVR 039618 N 0000000000000066	BASEVT	*PMCO	1	1	Y
HTTP QNOTES 026200 N 000000000000013D	BASEVT	*PMCO	1	1	Y

Records 1 - 17 of 35

Table of Contents | Previous | Next

5.5 Physical disk IO reports

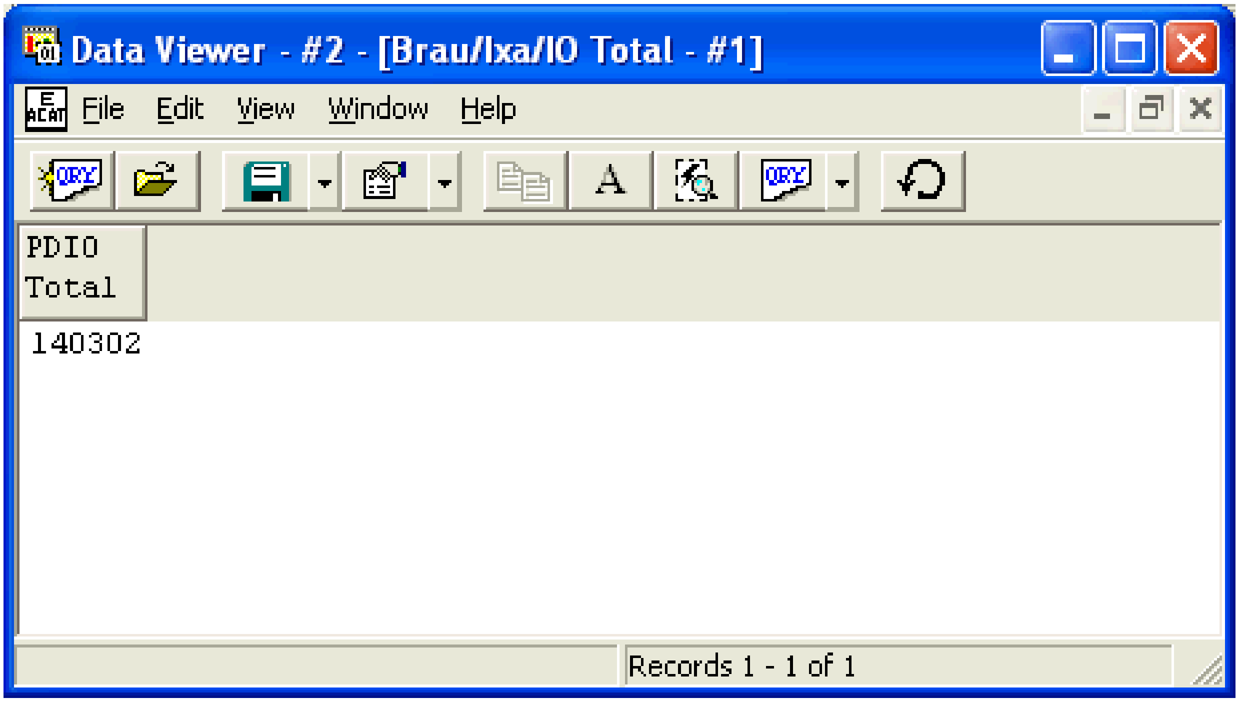
This analysis shows every physical disk IO operation over the course of the collection. The issuing TDE, time of day, object name & type, dasd unit and more are shown in this analysis.

If the MI Entry/Exit events were collected, the jobs program stack is visible by right-clicking on any output file containing the STACK field (with a value > 0) and choosing the 'Display Call Stack' menu.

Table of Contents | Previous | Next

5.5.1 IO Total

This report shows the total number of physical disk IO events that occurred over the data analyzed.



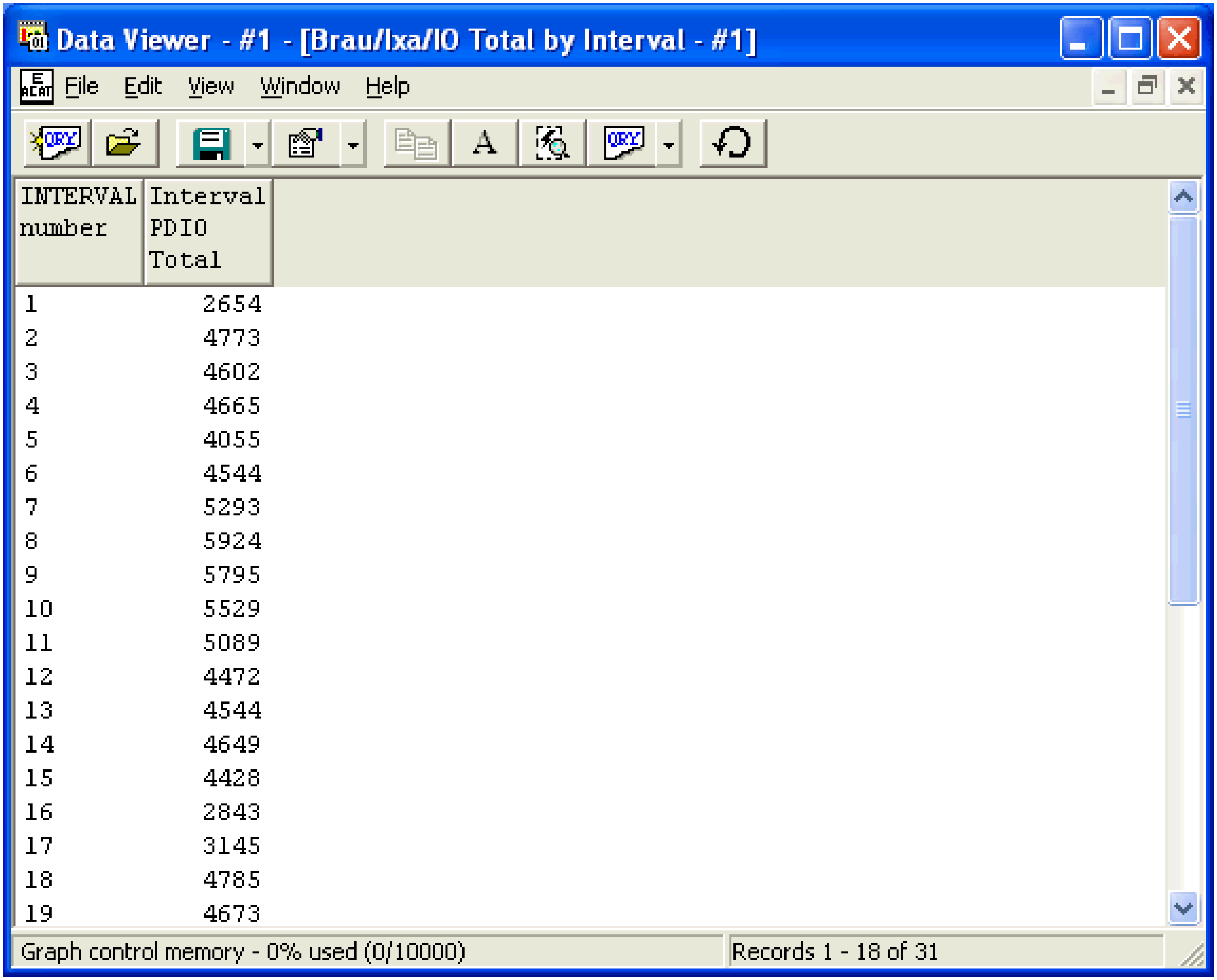
PDIO Total
140302

Records 1 - 1 of 1

Table of Contents Previous Next

5.5.2 IO Total by Interval

This report shows the total number of physical disk IO events that occurred each interval over the data analyzed.

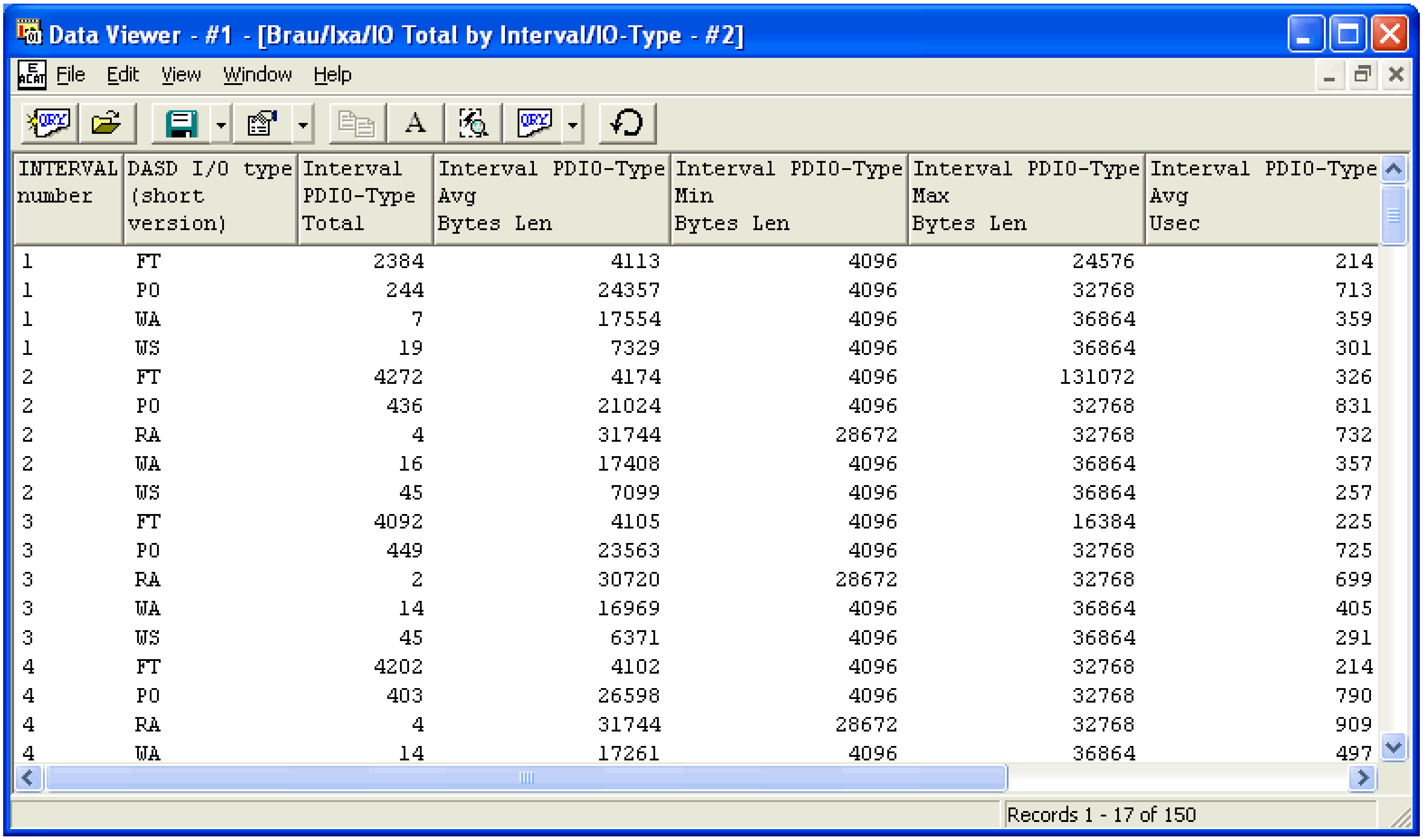

Data Viewer - #1 - [Brau/lxa/IO Total by Interval - #1]
File
Edit
View
Window
Help
QRY
QRY
A
Graph control memory - 0% used (0/10000)
Records 1 - 18 of 31

INTERVAL number	Interval PDIO Total
1	2654
2	4773
3	4602
4	4665
5	4055
6	4544
7	5293
8	5924
9	5795
10	5529
11	5089
12	4472
13	4544
14	4649
15	4428
16	2843
17	3145
18	4785
19	4673

Table of Contents Previous Next

5.5.3 IO Total by Interval/IO-Type

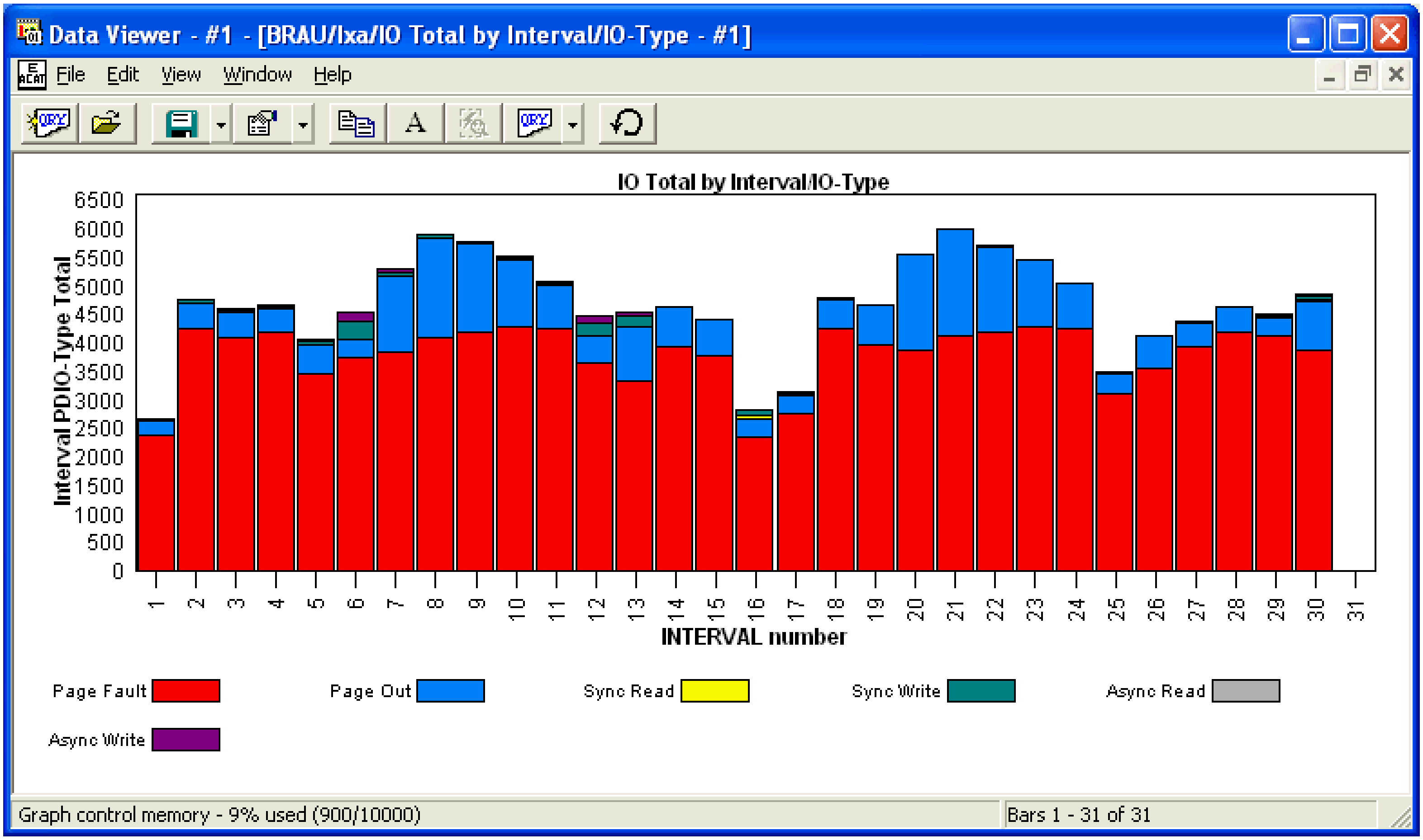
This report shows a breakdown of the physical disk IO events that occurred over the data analyzed by IO type for each interval.



Data Viewer - #1 - [Brau/Ixa/IO Total by Interval/IO-Type - #2]
File Edit View Window Help

INTERVAL number	DASD I/O type (short version)	Interval PDIO-Type Total	Interval PDIO-Type Avg Bytes Len	Interval PDIO-Type Min Bytes Len	Interval PDIO-Type Max Bytes Len	Interval PDIO-Type Avg Usec
1	FT	2384	4113	4096	24576	214
1	PO	244	24357	4096	32768	713
1	WA	7	17554	4096	36864	359
1	WS	19	7329	4096	36864	301
2	FT	4272	4174	4096	131072	326
2	PO	436	21024	4096	32768	831
2	RA	4	31744	28672	32768	732
2	WA	16	17408	4096	36864	357
2	WS	45	7099	4096	36864	257
3	FT	4092	4105	4096	16384	225
3	PO	449	23563	4096	32768	725
3	RA	2	30720	28672	32768	699
3	WA	14	16969	4096	36864	405
3	WS	45	6371	4096	36864	291
4	FT	4202	4102	4096	32768	214
4	PO	403	26598	4096	32768	790
4	RA	4	31744	28672	32768	909
4	WA	14	17261	4096	36864	497

Records 1 - 17 of 150



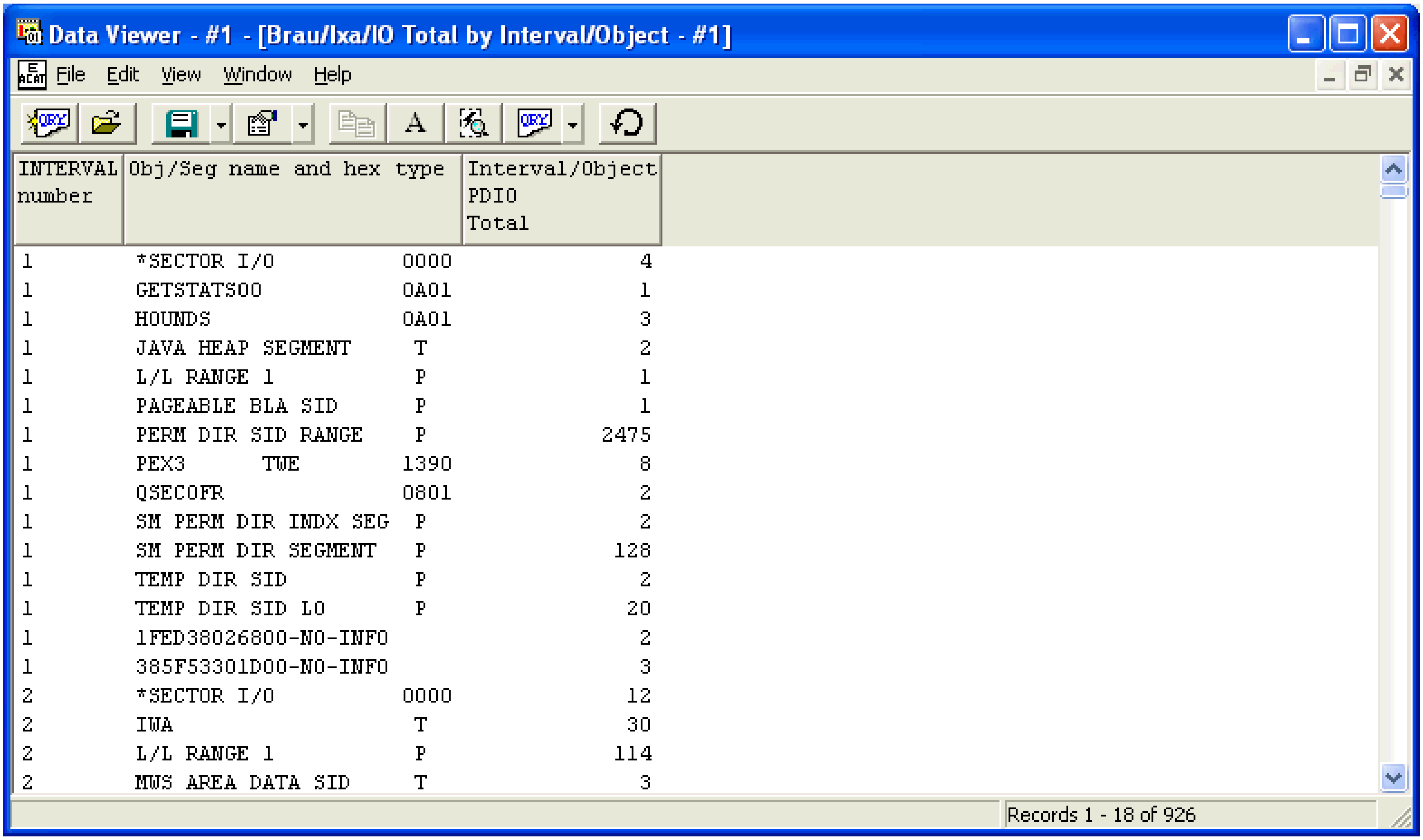
Graph control memory - 9% used (900/10000)

Bars 1 - 31 of 31

Table of Contents | Previous | Next

5.5.4 IO Total by Interval/Object

This report shows a breakdown of the physical disk IO events that occurred over the data analyzed for each object within each interval.


Data Viewer - #1 - [Brau/Ixa/IO Total by Interval/Object - #1]
File
Edit
View
Window
Help
QRY
A
QRY

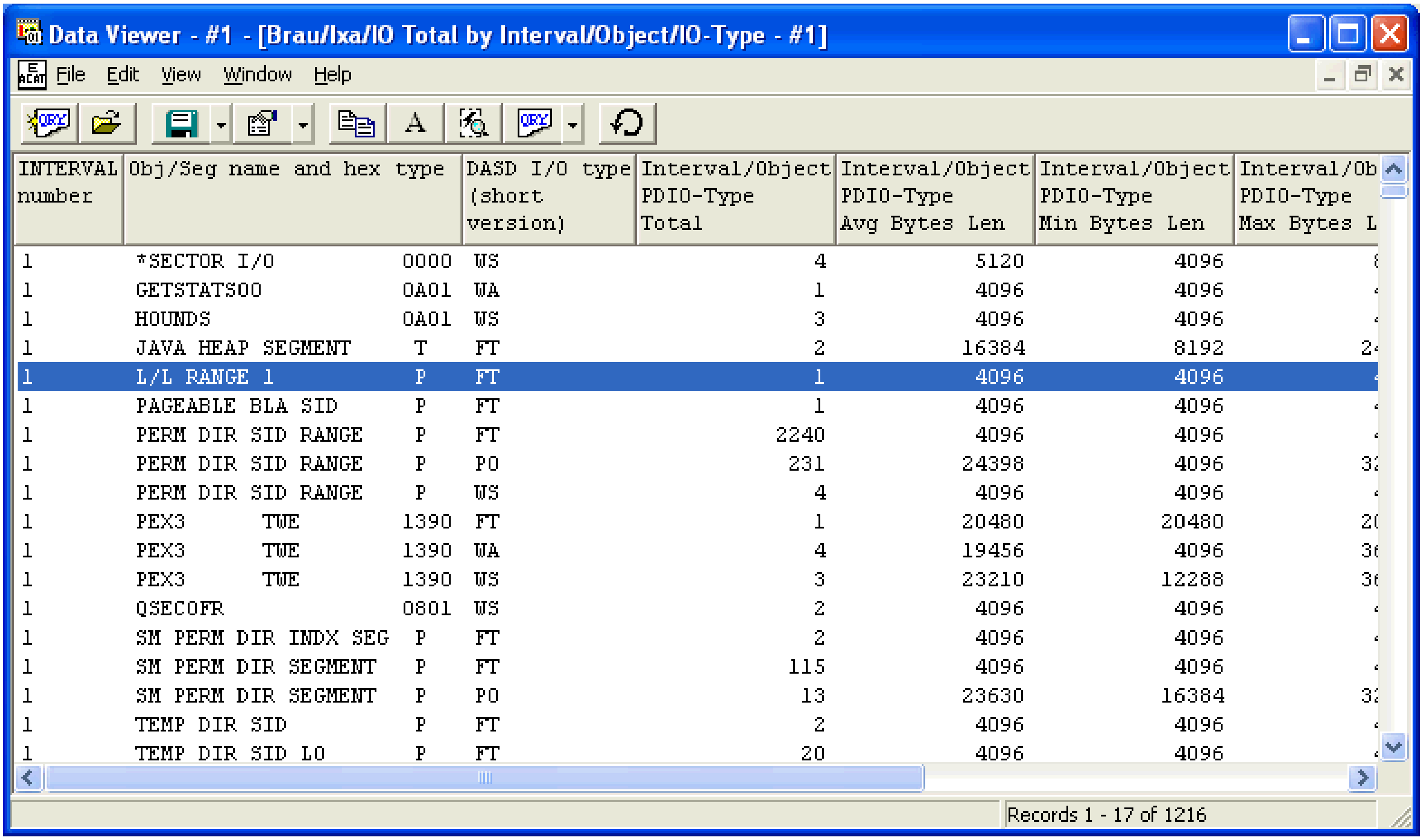
INTERVAL number	Obj/Seg name and hex type	Interval/Object PDIO Total
1	*SECTOR I/O 0000	4
1	GETSTATS00 0A01	1
1	HOUNDS 0A01	3
1	JAVA HEAP SEGMENT T	2
1	L/L RANGE 1 P	1
1	PAGEABLE BLA SID P	1
1	PERM DIR SID RANGE P	2475
1	PEX3 TWE 1390	8
1	QSECOFR 0801	2
1	SM PERM DIR INDX SEG P	2
1	SM PERM DIR SEGMENT P	128
1	TEMP DIR SID P	2
1	TEMP DIR SID LO P	20
1	1FED38026800-NO-INFO	2
1	385F53301D00-NO-INFO	3
2	*SECTOR I/O 0000	12
2	IWA T	30
2	L/L RANGE 1 P	114
2	MWS AREA DATA SID T	3

Records 1 - 18 of 926

Table of Contents Previous Next

5.5.5 IO Total by Interval/Object/IO-Type

This report shows a breakdown of the PDIO events that occurred over the data analyzed for each interval, object, and type combination.



Data Viewer - #1 - [Brau/Ixa/IO Total by Interval/Object/IO-Type - #1]
File
Edit
View
Window
Help

INTERVAL number	Obj/Seg name and hex type	DASD I/O type (short version)	Interval/Object PDIO-Type Total	Interval/Object PDIO-Type Avg Bytes Len	Interval/Object PDIO-Type Min Bytes Len	Interval/Ob PDIO-Type Max Bytes L
1	*SECTOR I/O 0000	WS	4	5120	4096	8
1	GETSTATS00 0A01	WA	1	4096	4096	4
1	HOUNDS 0A01	WS	3	4096	4096	4
1	JAVA HEAP SEGMENT T	FT	2	16384	8192	24
1	L/L RANGE 1 P	FT	1	4096	4096	4
1	PAGEABLE BLA SID P	FT	1	4096	4096	4
1	PERM DIR SID RANGE P	FT	2240	4096	4096	4
1	PERM DIR SID RANGE P	PO	231	24398	4096	32
1	PERM DIR SID RANGE P	WS	4	4096	4096	4
1	PEX3 TWE 1390	FT	1	20480	20480	20
1	PEX3 TWE 1390	WA	4	19456	4096	36
1	PEX3 TWE 1390	WS	3	23210	12288	36
1	QSECOFR 0801	WS	2	4096	4096	4
1	SM PERM DIR INDX SEG P	FT	2	4096	4096	4
1	SM PERM DIR SEGMENT P	FT	115	4096	4096	4
1	SM PERM DIR SEGMENT P	PO	13	23630	16384	32
1	TEMP DIR SID P	FT	2	4096	4096	4
1	TEMP DIR SID LO P	FT	20	4096	4096	4

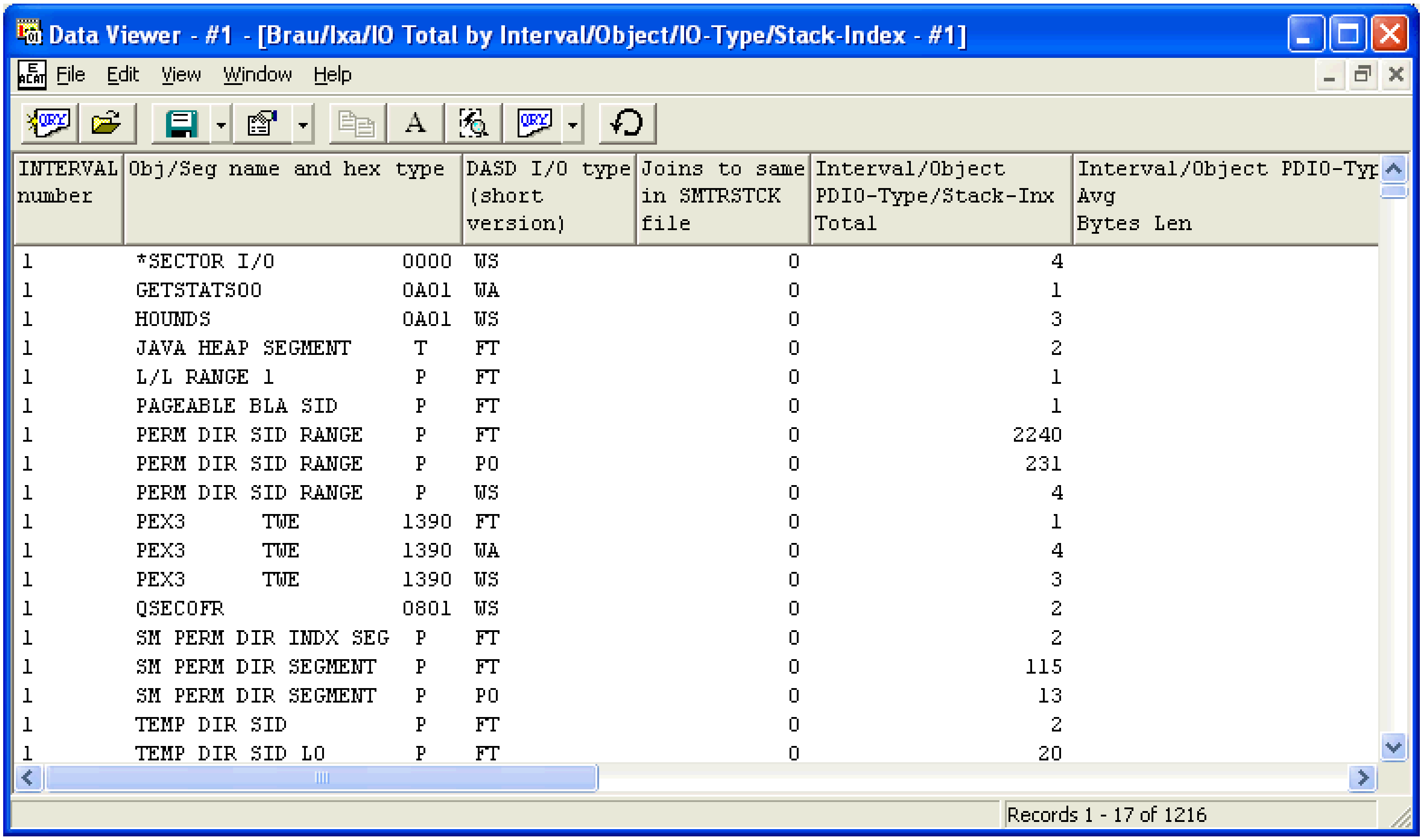
Records 1 - 17 of 1216

Table of Contents   Previous   Next

5.5.6 IO Total by Interval/Object/IO-Type/Stack-Index

This report shows a breakdown of the physical disk IO events that occurred over the data analyzed for each interval, object, type and stack index combination.

If the stack index is non zero, a call stack will appear in the properties found by double-clicking on the desired record.


Data Viewer - #1 - [Brau/lxa/IO Total by Interval/Object/IO-Type/Stack-Index - #1]
File
Edit
View
Window
Help
QRY
A
QRY

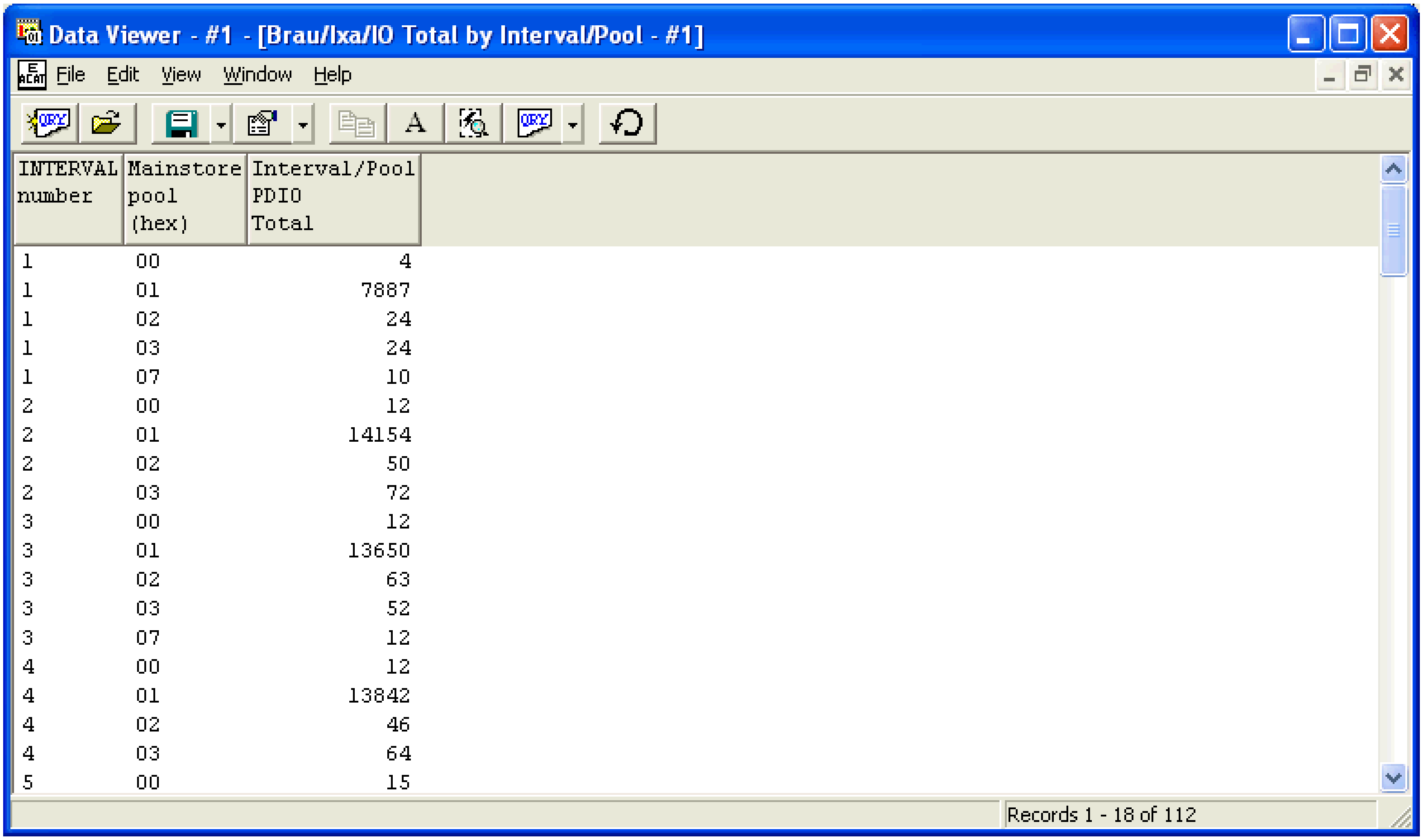
INTERVAL number	Obj/Seg name and hex type	DASD I/O type (short version)	Joins to same in SMTRSTCK file	Interval/Object PDIO-Type/Stack-Inx Total	Interval/Object PDIO-Typ Avg Bytes Len
1	*SECTOR I/O 0000	WS	0	4	
1	GETSTATS00 0A01	WA	0	1	
1	HOUNDS 0A01	WS	0	3	
1	JAVA HEAP SEGMENT T	FT	0	2	
1	L/L RANGE 1 P	FT	0	1	
1	PAGEABLE BLA SID P	FT	0	1	
1	PERM DIR SID RANGE P	FT	0	2240	
1	PERM DIR SID RANGE P	PO	0	231	
1	PERM DIR SID RANGE P	WS	0	4	
1	PEX3 TWE 1390	FT	0	1	
1	PEX3 TWE 1390	WA	0	4	
1	PEX3 TWE 1390	WS	0	3	
1	QSECOFR 0801	WS	0	2	
1	SM PERM DIR INDX SEG P	FT	0	2	
1	SM PERM DIR SEGMENT P	FT	0	115	
1	SM PERM DIR SEGMENT P	PO	0	13	
1	TEMP DIR SID P	FT	0	2	
1	TEMP DIR SID LO P	FT	0	20	

Records 1 - 17 of 1216

Table of Contents Previous Next

5.5.7 IO Total by Interval/Pool

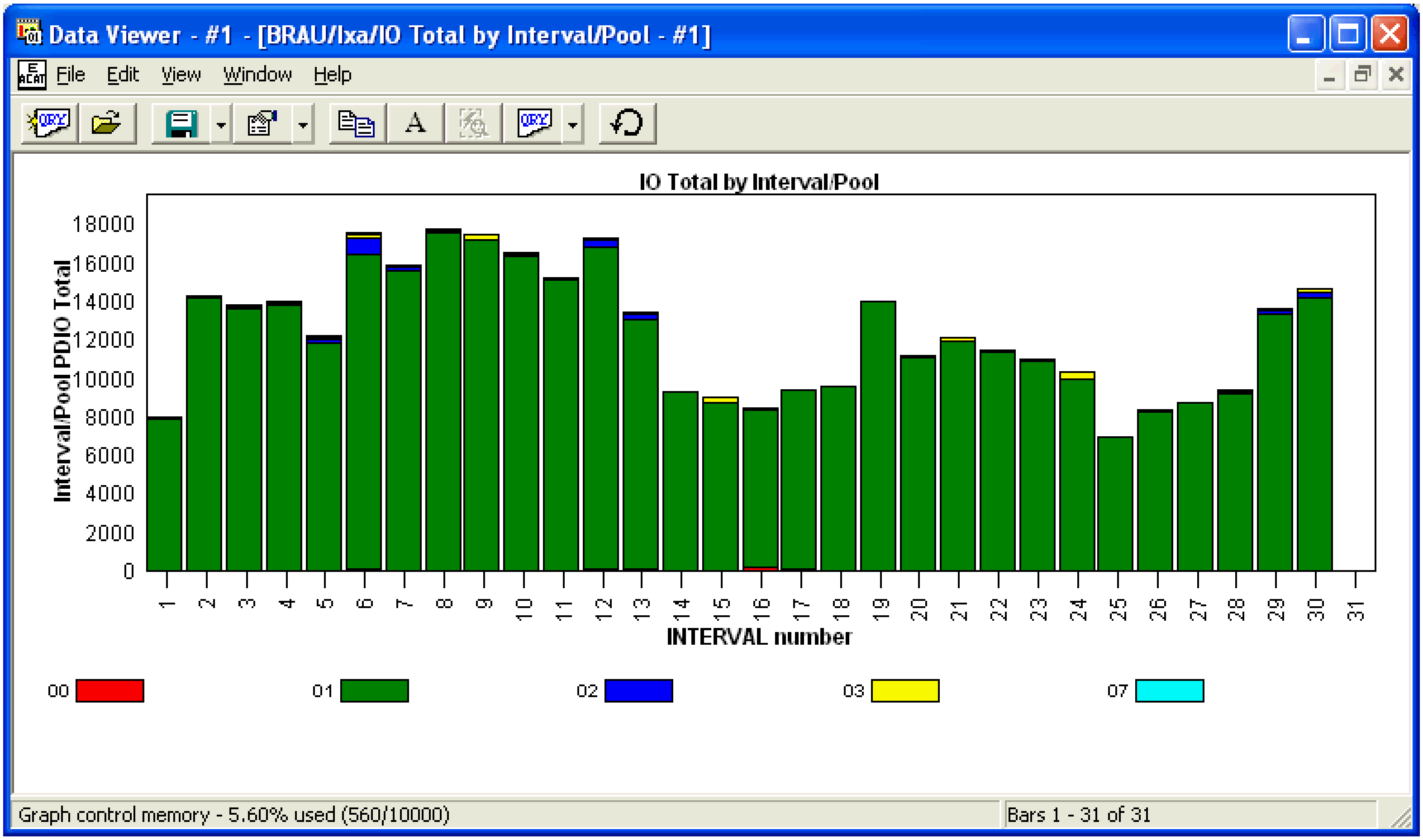
This report shows the total number of physical disk IO events that occurred over the data analyzed for each memory pool within each interval.



Data Viewer - #1 - [Brau/Ixa/IO Total by Interval/Pool - #1]
File
Edit
View
Window
Help

INTERVAL number	Mainstore pool (hex)	Interval/Pool PDIO Total
1	00	4
1	01	7887
1	02	24
1	03	24
1	07	10
2	00	12
2	01	14154
2	02	50
2	03	72
3	00	12
3	01	13650
3	02	63
3	03	52
3	07	12
4	00	12
4	01	13842
4	02	46
4	03	64
5	00	15

Records 1 - 18 of 112



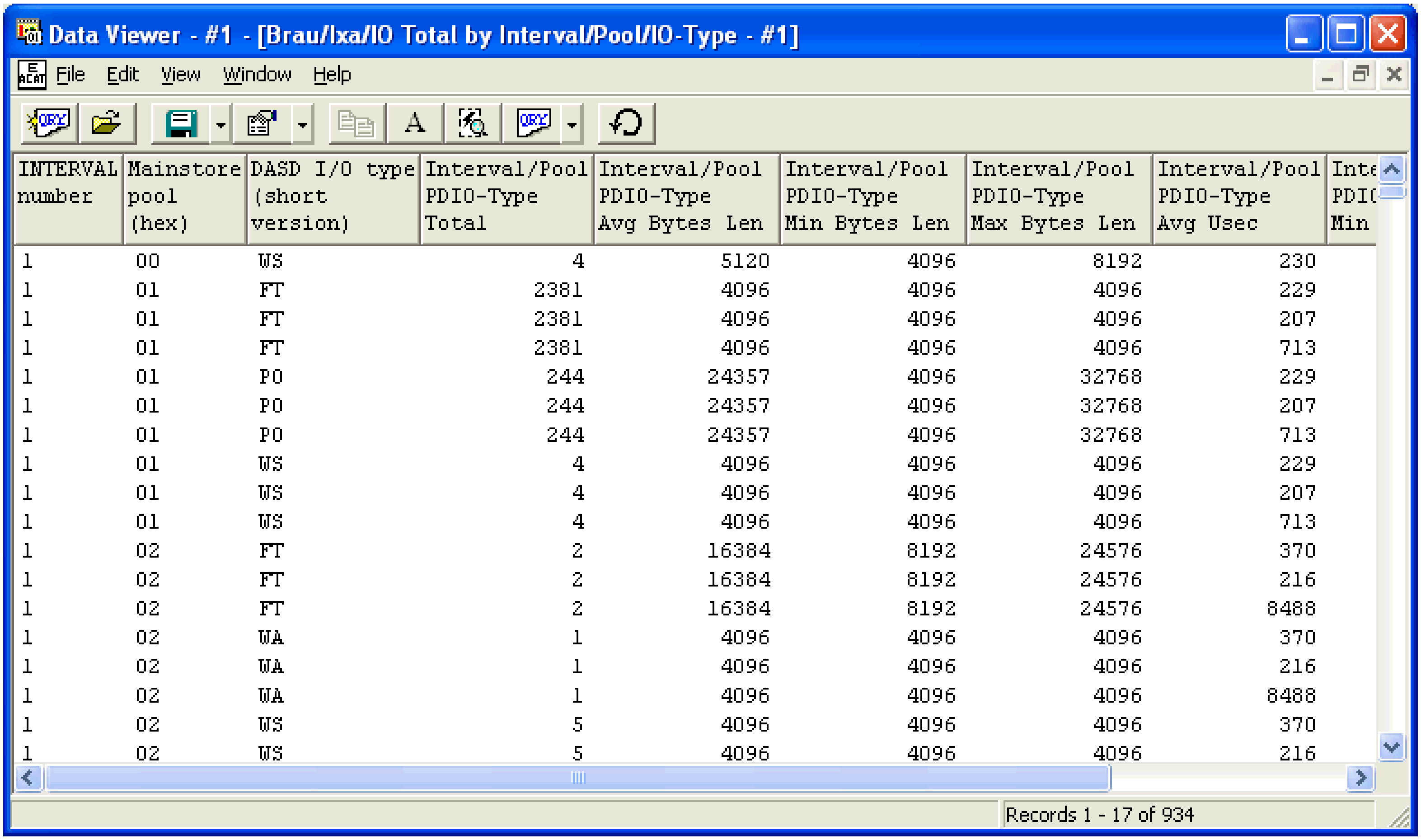
Graph control memory - 5.60% used (560/10000)

Bars 1 - 31 of 31

Table of Contents Previous Next

5.5.8 IO Total by Interval/Pool/IO-Type

This report shows a breakdown of the physical disk IO events that occurred over the data analyzed for each memory pool, io type, and interval combination.


Data Viewer - #1 - [Brau/Ixa/IO Total by Interval/Pool/IO-Type - #1]
File
Edit
View
Window
Help

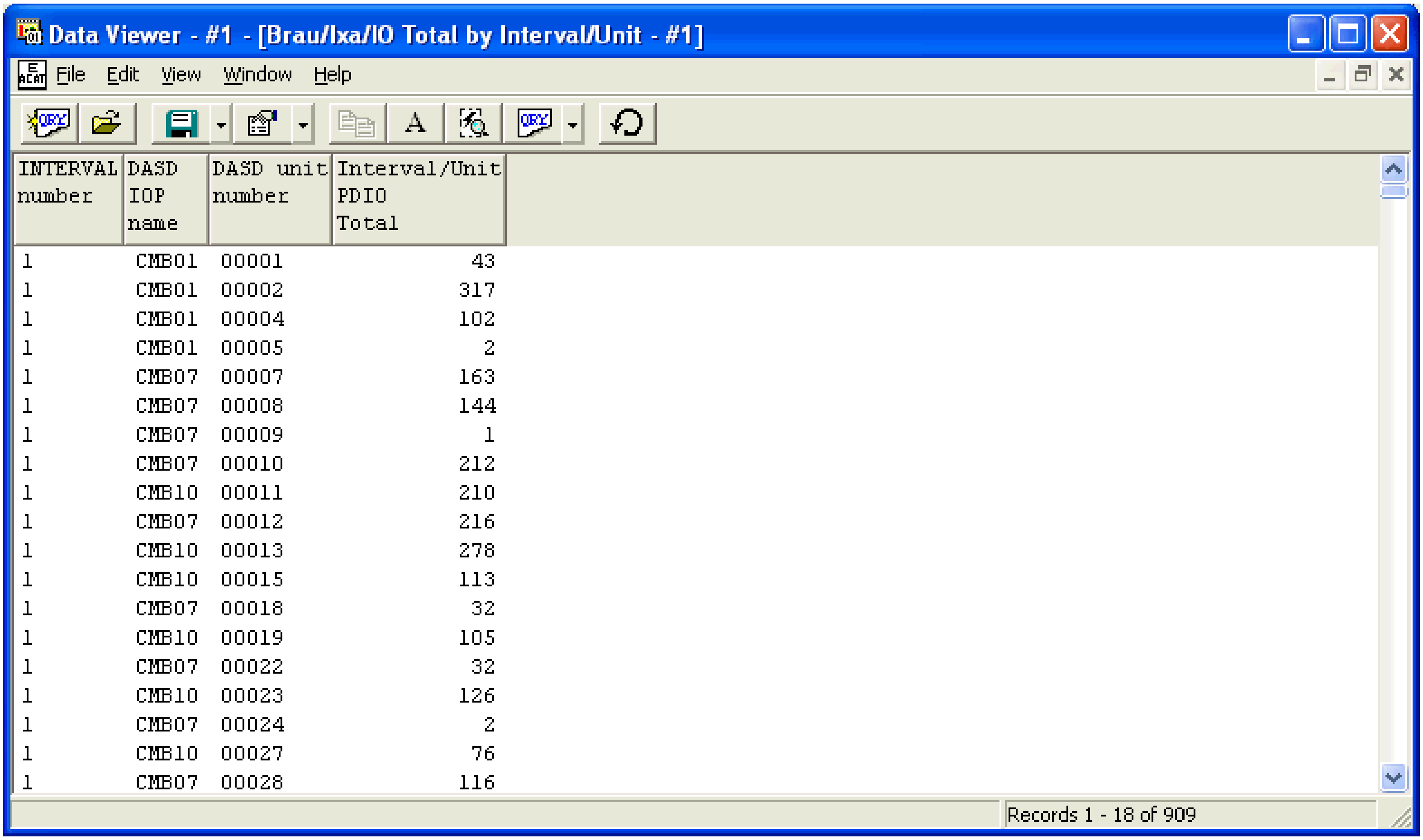
INTERVAL number	Mainstore pool (hex)	DASD I/O type (short version)	Interval/Pool PDIO-Type Total	Interval/Pool PDIO-Type Avg Bytes Len	Interval/Pool PDIO-Type Min Bytes Len	Interval/Pool PDIO-Type Max Bytes Len	Interval/Pool PDIO-Type Avg Usec	Inte PDIC Min
1	00	WS	4	5120	4096	8192	230	
1	01	FT	2381	4096	4096	4096	229	
1	01	FT	2381	4096	4096	4096	207	
1	01	FT	2381	4096	4096	4096	713	
1	01	PO	244	24357	4096	32768	229	
1	01	PO	244	24357	4096	32768	207	
1	01	PO	244	24357	4096	32768	713	
1	01	WS	4	4096	4096	4096	229	
1	01	WS	4	4096	4096	4096	207	
1	01	WS	4	4096	4096	4096	713	
1	02	FT	2	16384	8192	24576	370	
1	02	FT	2	16384	8192	24576	216	
1	02	FT	2	16384	8192	24576	8488	
1	02	WA	1	4096	4096	4096	370	
1	02	WA	1	4096	4096	4096	216	
1	02	WA	1	4096	4096	4096	8488	
1	02	WS	5	4096	4096	4096	370	
1	02	WS	5	4096	4096	4096	216	

Records 1 - 17 of 934

Table of Contents Previous Next

5.5.9 IO Total by Interval/Unit

This report shows the total number of physical disk IO events that occurred over the data analyzed for each disk unit during each interval.



Data Viewer - #1 - [Brau/Ixa/IO Total by Interval/Unit - #1]
File Edit View Window Help

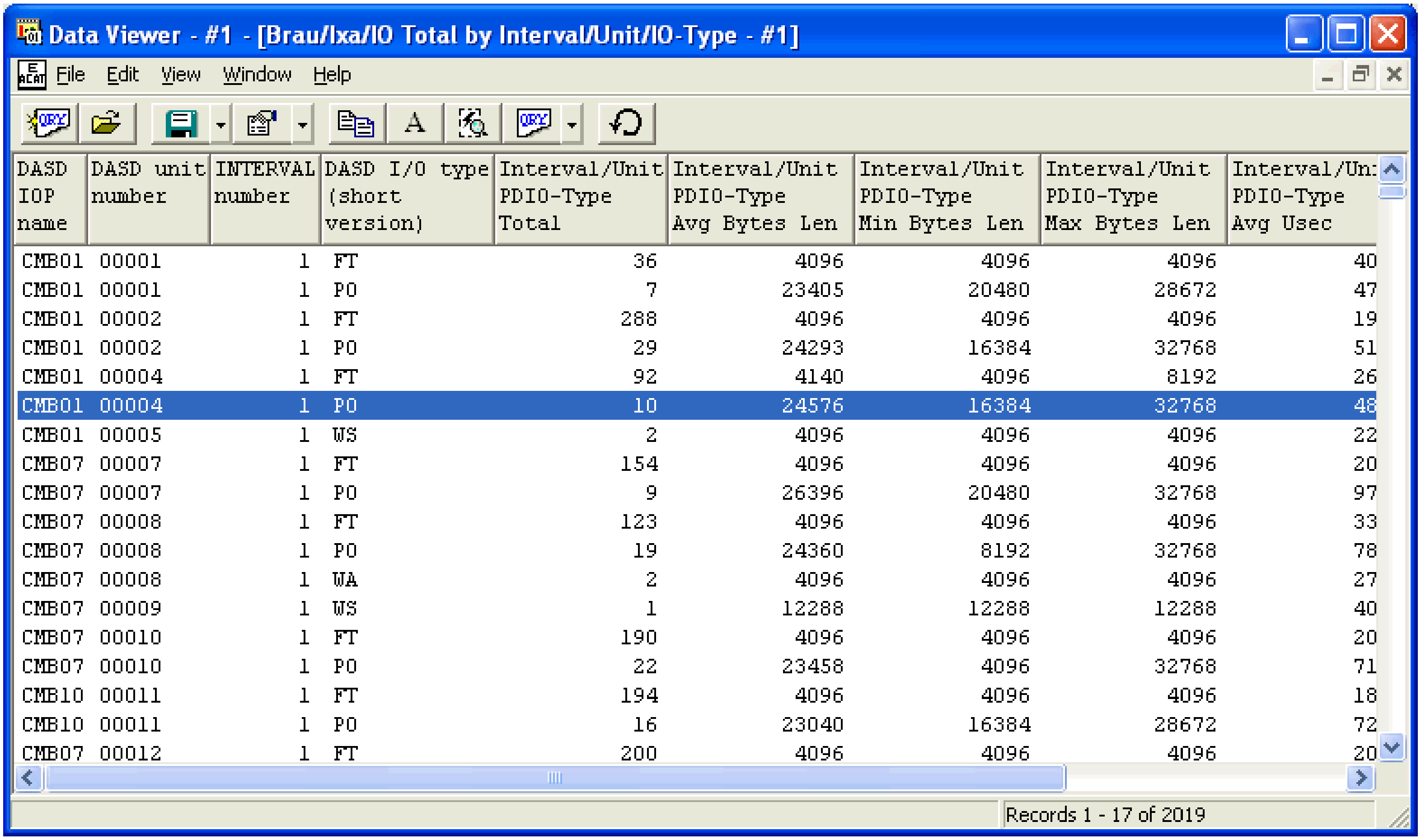
INTERVAL number	DASD IOP name	DASD unit number	Interval/Unit PDIO Total
1	CMB01	00001	43
1	CMB01	00002	317
1	CMB01	00004	102
1	CMB01	00005	2
1	CMB07	00007	163
1	CMB07	00008	144
1	CMB07	00009	1
1	CMB07	00010	212
1	CMB10	00011	210
1	CMB07	00012	216
1	CMB10	00013	278
1	CMB10	00015	113
1	CMB07	00018	32
1	CMB10	00019	105
1	CMB07	00022	32
1	CMB10	00023	126
1	CMB07	00024	2
1	CMB10	00027	76
1	CMB07	00028	116

Records 1 - 18 of 909

Table of Contents Previous Next

5.5.10 IO Total by Interval/Unit/IO-Type

This report shows a breakdown of the physical disk IO events that occurred over the data analyzed for each disk unit, io type and interval combination.

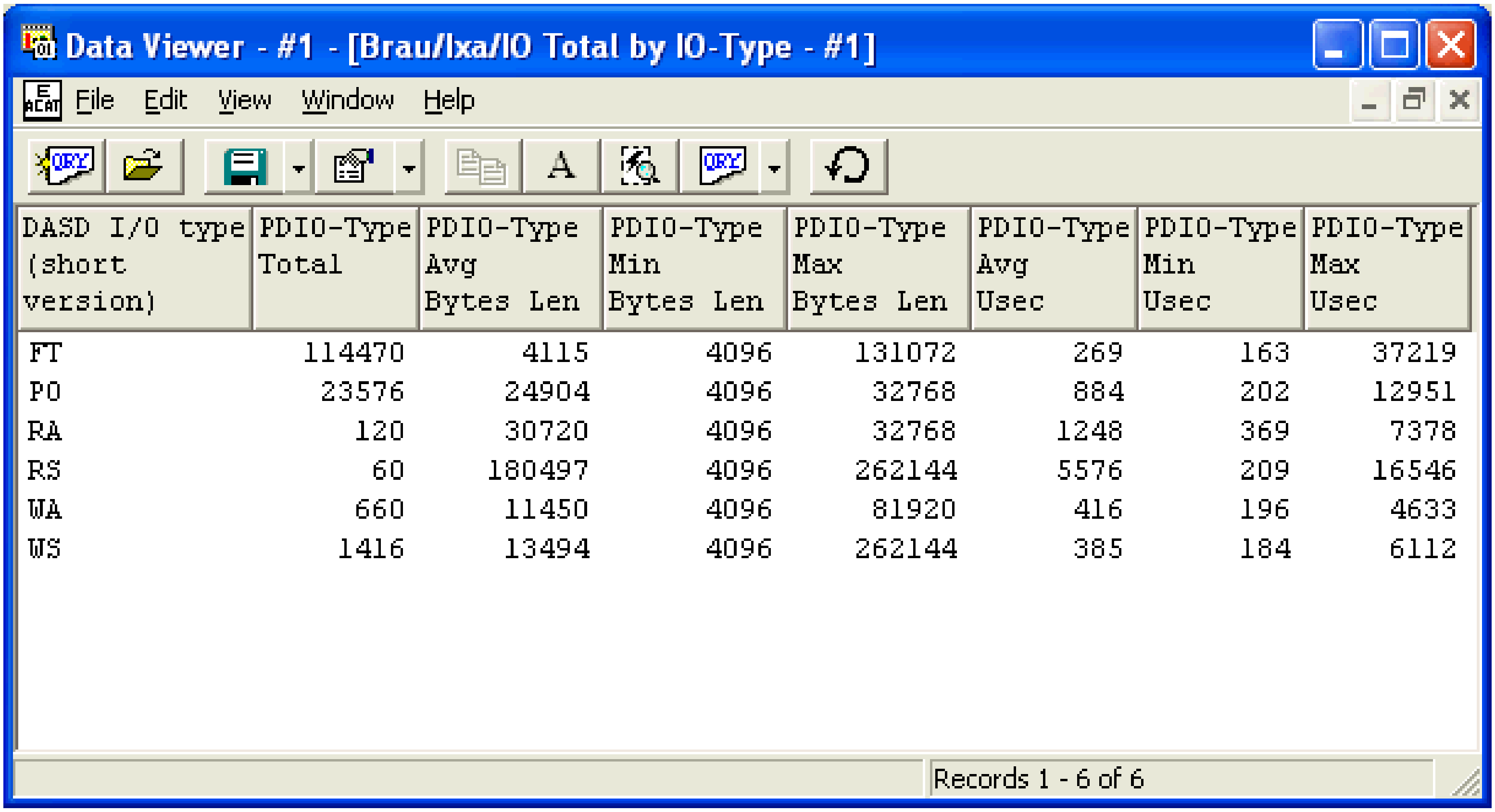

Data Viewer - #1 - [Brau/Ixa/IO Total by Interval/Unit/IO-Type - #1]
File Edit View Window Help
Records 1 - 17 of 2019

DASD IOP name	DASD unit number	INTERVAL number	DASD I/O type (short version)	Interval/Unit PDIO-Type Total	Interval/Unit PDIO-Type Avg Bytes Len	Interval/Unit PDIO-Type Min Bytes Len	Interval/Unit PDIO-Type Max Bytes Len	Interval/Un: PDIO-Type Avg Usec
CMB01	00001	1	FT	36	4096	4096	4096	40
CMB01	00001	1	PO	7	23405	20480	28672	47
CMB01	00002	1	FT	288	4096	4096	4096	19
CMB01	00002	1	PO	29	24293	16384	32768	51
CMB01	00004	1	FT	92	4140	4096	8192	26
CMB01	00004	1	PO	10	24576	16384	32768	48
CMB01	00005	1	WS	2	4096	4096	4096	22
CMB07	00007	1	FT	154	4096	4096	4096	20
CMB07	00007	1	PO	9	26396	20480	32768	97
CMB07	00008	1	FT	123	4096	4096	4096	33
CMB07	00008	1	PO	19	24360	8192	32768	78
CMB07	00008	1	WA	2	4096	4096	4096	27
CMB07	00009	1	WS	1	12288	12288	12288	40
CMB07	00010	1	FT	190	4096	4096	4096	20
CMB07	00010	1	PO	22	23458	4096	32768	71
CMB10	00011	1	FT	194	4096	4096	4096	18
CMB10	00011	1	PO	16	23040	16384	28672	72
CMB07	00012	1	FT	200	4096	4096	4096	20

Table of Contents Previous Next

5.5.11 IO Total by IO-Type

This report shows the total IOs that occurred in the collection of each type as well as the average, min, and max times and sizes.


Data Viewer - #1 - [Brau/lxa/IO Total by IO-Type - #1]
File Edit View Window Help

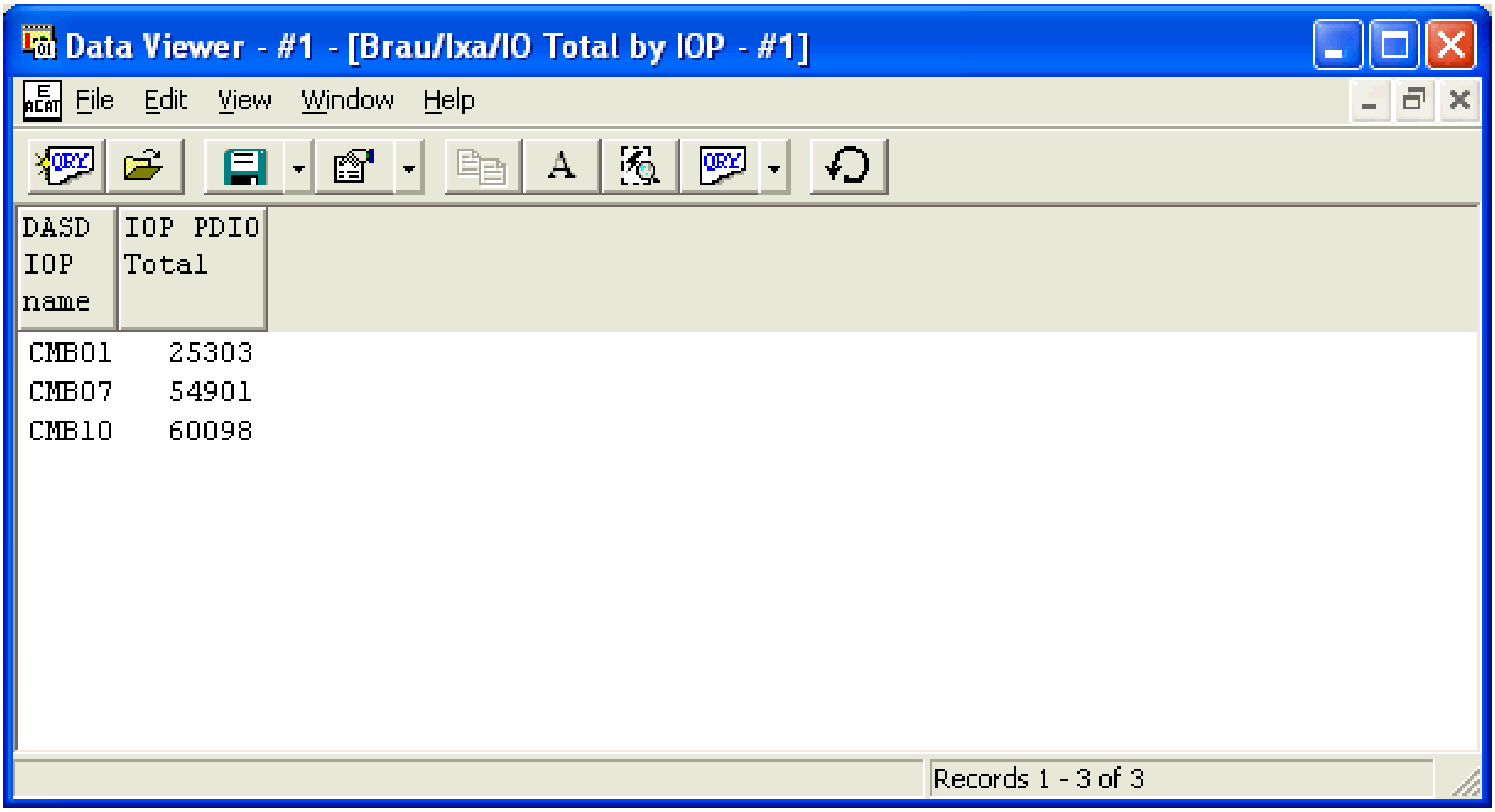
DASD I/O type (short version)	PDIO-Type Total	PDIO-Type Avg Bytes Len	PDIO-Type Min Bytes Len	PDIO-Type Max Bytes Len	PDIO-Type Avg Usec	PDIO-Type Min Usec	PDIO-Type Max Usec
FT	114470	4115	4096	131072	269	163	37219
PO	23576	24904	4096	32768	884	202	12951
RA	120	30720	4096	32768	1248	369	7378
RS	60	180497	4096	262144	5576	209	16546
WA	660	11450	4096	81920	416	196	4633
WS	1416	13494	4096	262144	385	184	6112

Records 1 - 6 of 6

Table of Contents | Previous | Next

5.5.12 IO Total by IOP

This report shows the total number of physical disk IO events that occurred for each DASD IOP.



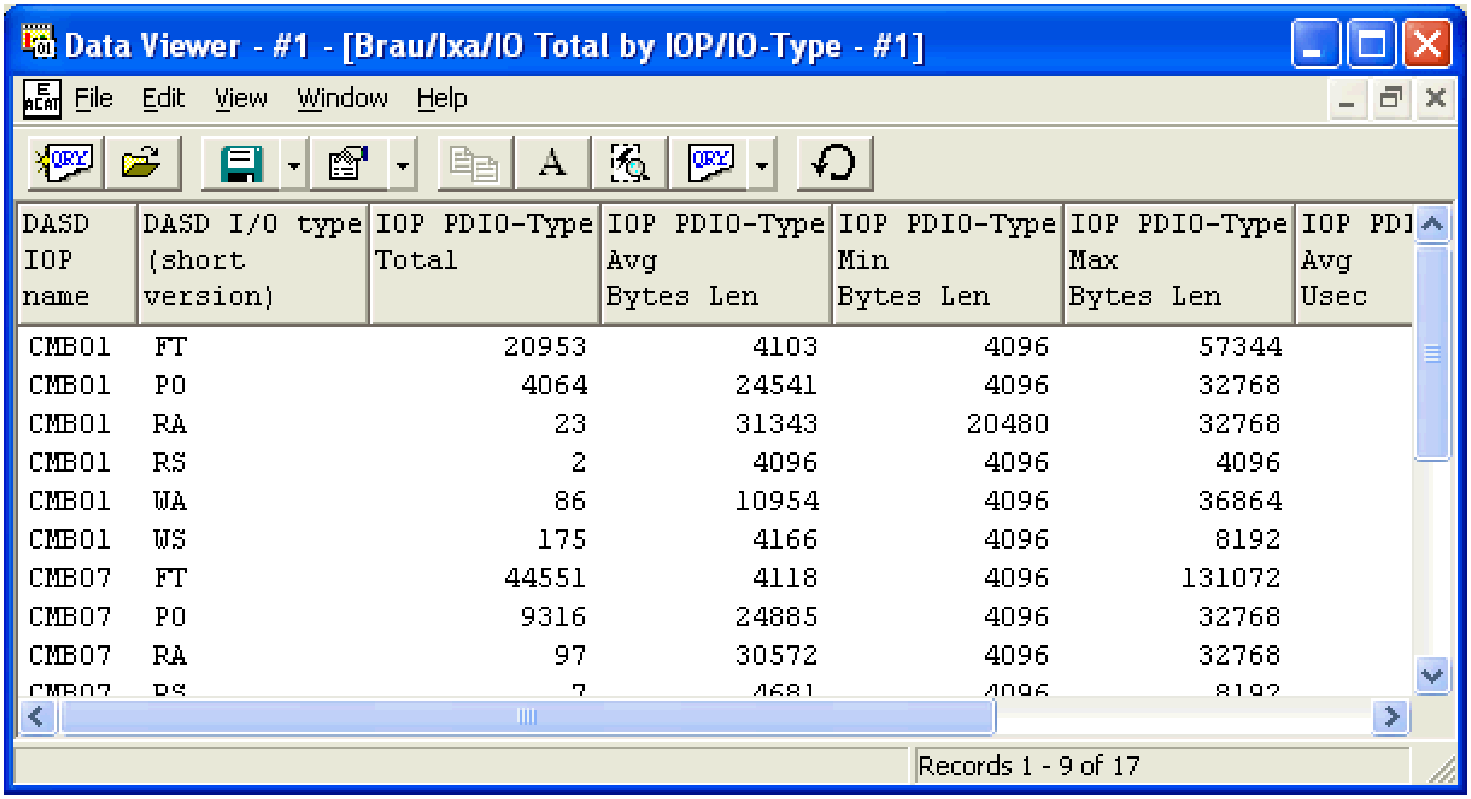
DASD IOP name	IOP PDIO Total
CMB01	25303
CMB07	54901
CMB10	60098

Records 1 - 3 of 3

Table of Contents Previous Next

5.5.13 IO Total by IOP/IO-Type

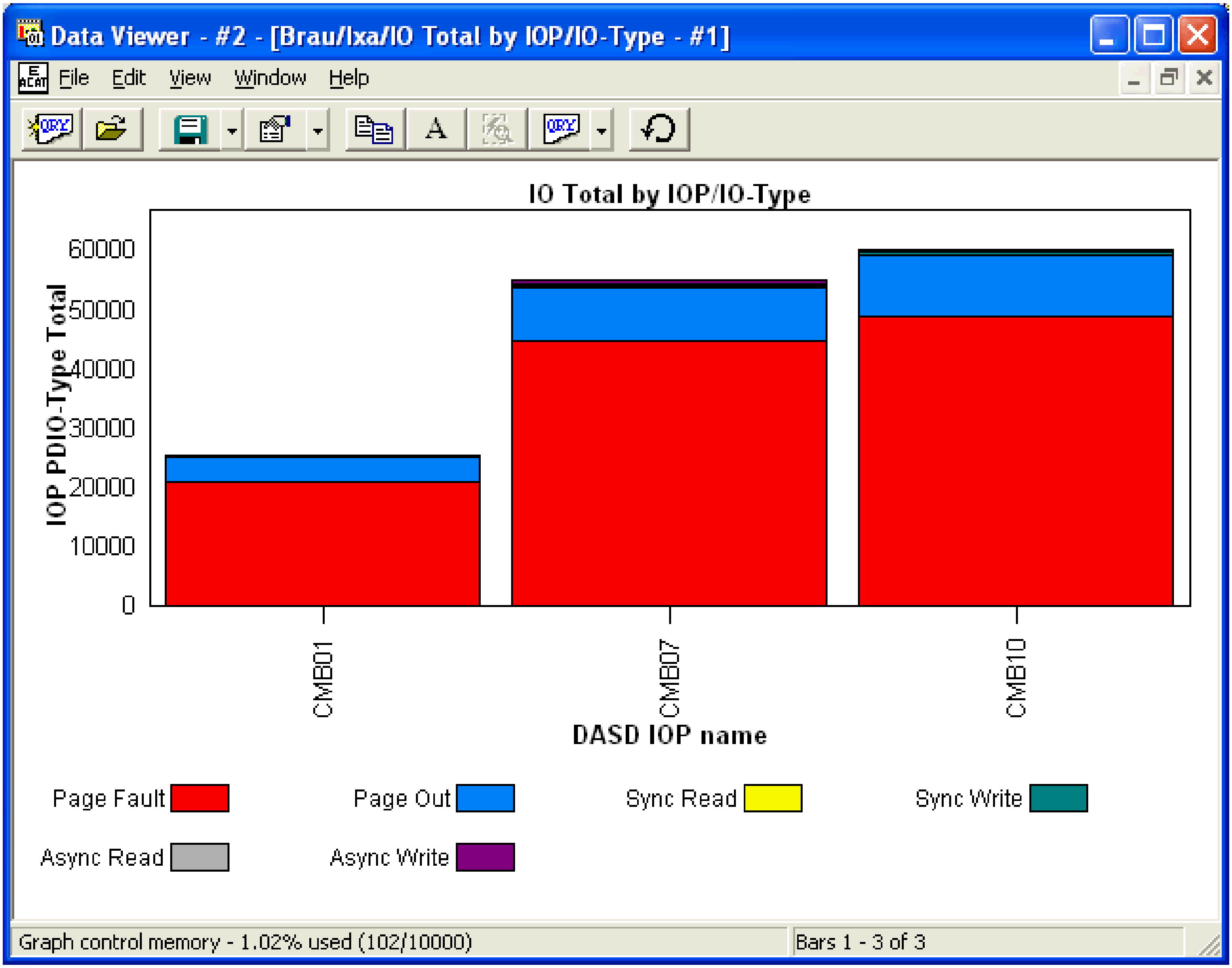
This report shows a breakdown of the physical disk IO events that occurred for each DASD IOP by IO type.



Data Viewer - #1 - [Brau/Ixa/IO Total by IOP/IO-Type - #1]

DASD IOP name	DASD I/O type (short version)	IOP PDIO-Type Total	IOP PDIO-Type Avg Bytes Len	IOP PDIO-Type Min Bytes Len	IOP PDIO-Type Max Bytes Len	IOP PDI Avg Usec
CMB01	FT	20953	4103	4096	57344	
CMB01	PO	4064	24541	4096	32768	
CMB01	RA	23	31343	20480	32768	
CMB01	RS	2	4096	4096	4096	
CMB01	WA	86	10954	4096	36864	
CMB01	WS	175	4166	4096	8192	
CMB07	FT	44551	4118	4096	131072	
CMB07	PO	9316	24885	4096	32768	
CMB07	RA	97	30572	4096	32768	
CMB07	RS	7	4681	4096	8192	

Records 1 - 9 of 17



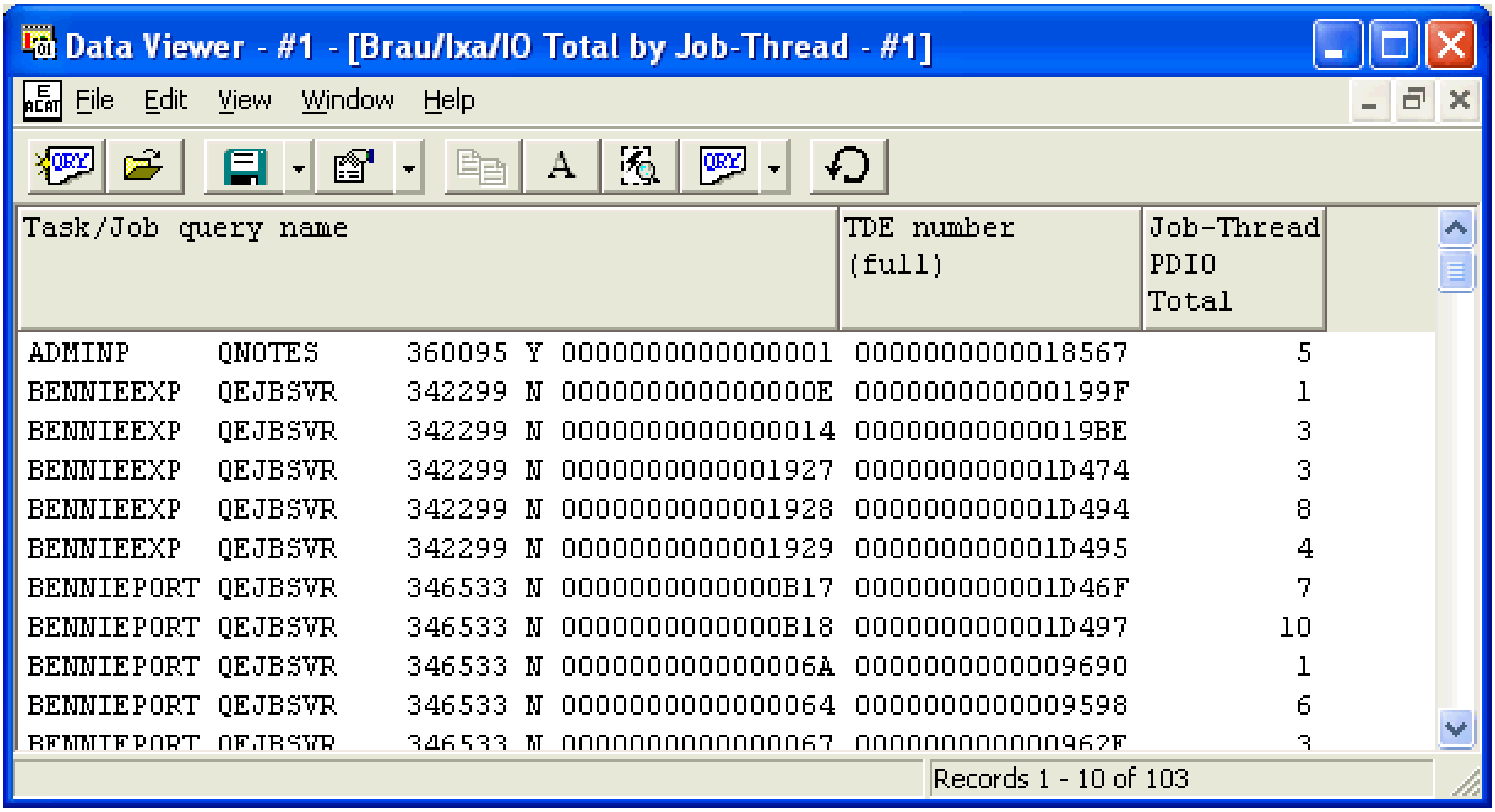
Graph control memory - 1.02% used (102/10000)

Bars 1 - 3 of 3

Table of Contents | Previous | Next

5.5.14 IO Total by Job-Thread

This report shows the total number of physical disk IO events that occurred for each job/thread within the collection.



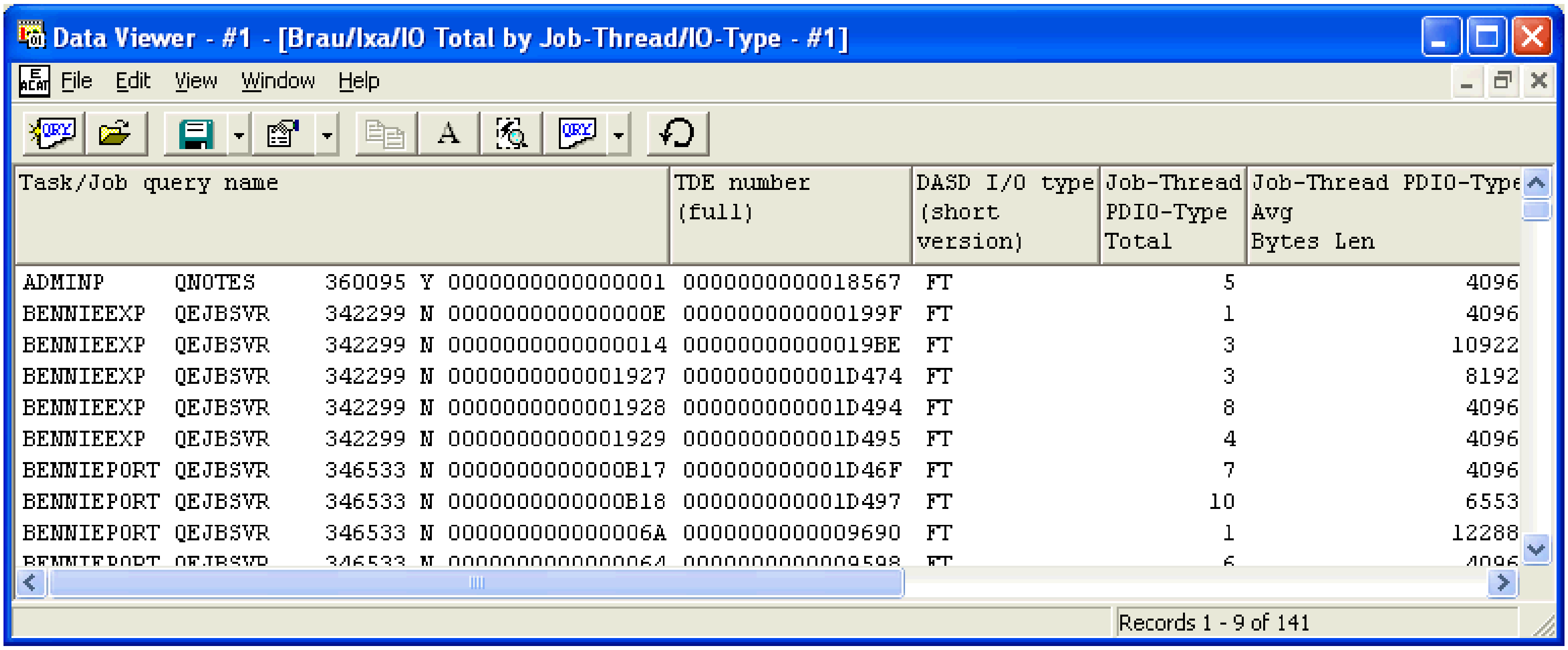
Task/Job query name					TDE number (full)	Job-Thread PDIO Total
ADMINP	QNOTES	360095	Y	0000000000000001	0000000000018567	5
BENNIEEXP	QEJBSVR	342299	N	000000000000000E	000000000000199F	1
BENNIEEXP	QEJBSVR	342299	N	0000000000000014	00000000000019BE	3
BENNIEEXP	QEJBSVR	342299	N	0000000000001927	000000000001D474	3
BENNIEEXP	QEJBSVR	342299	N	0000000000001928	000000000001D494	8
BENNIEEXP	QEJBSVR	342299	N	0000000000001929	000000000001D495	4
BENNIEPORT	QEJBSVR	346533	N	0000000000000B17	000000000001D46F	7
BENNIEPORT	QEJBSVR	346533	N	0000000000000B18	000000000001D497	10
BENNIEPORT	QEJBSVR	346533	N	000000000000006A	0000000000009690	1
BENNIEPORT	QEJBSVR	346533	N	0000000000000064	0000000000009598	6
BENNIEPORT	QEJBSVR	346533	N	0000000000000067	000000000000962F	3

Records 1 - 10 of 103

Table of Contents | Previous | Next

5.5.15 IO Total by Job-Thread/IO-Type

This report shows a breakdown of the physical disk IO events that occurred for each IO type within job/thread.



Data Viewer - #1 - [Brau/Ixa/IO Total by Job-Thread/IO-Type - #1]

Task/Job query name	TDE number (full)	DASD I/O type (short version)	Job-Thread PDIO-Type Total	Job-Thread PDIO-Type Avg Bytes Len
ADMINP QNOTES 360095 Y 0000000000000001	0000000000018567	FT	5	4096
BENNIEEXP QEJBSVR 342299 N 000000000000000E	000000000000199F	FT	1	4096
BENNIEEXP QEJBSVR 342299 N 0000000000000014	00000000000019BE	FT	3	10922
BENNIEEXP QEJBSVR 342299 N 0000000000001927	000000000001D474	FT	3	8192
BENNIEEXP QEJBSVR 342299 N 0000000000001928	000000000001D494	FT	8	4096
BENNIEEXP QEJBSVR 342299 N 0000000000001929	000000000001D495	FT	4	4096
BENNIEPORT QEJBSVR 346533 N 0000000000000B17	000000000001D46F	FT	7	4096
BENNIEPORT QEJBSVR 346533 N 0000000000000B18	000000000001D497	FT	10	6553
BENNIEPORT QEJBSVR 346533 N 000000000000006A	0000000000009690	FT	1	12288
BENNIEPORT QEJBSVR 346533 N 0000000000000064	0000000000009598	FT	6	4096

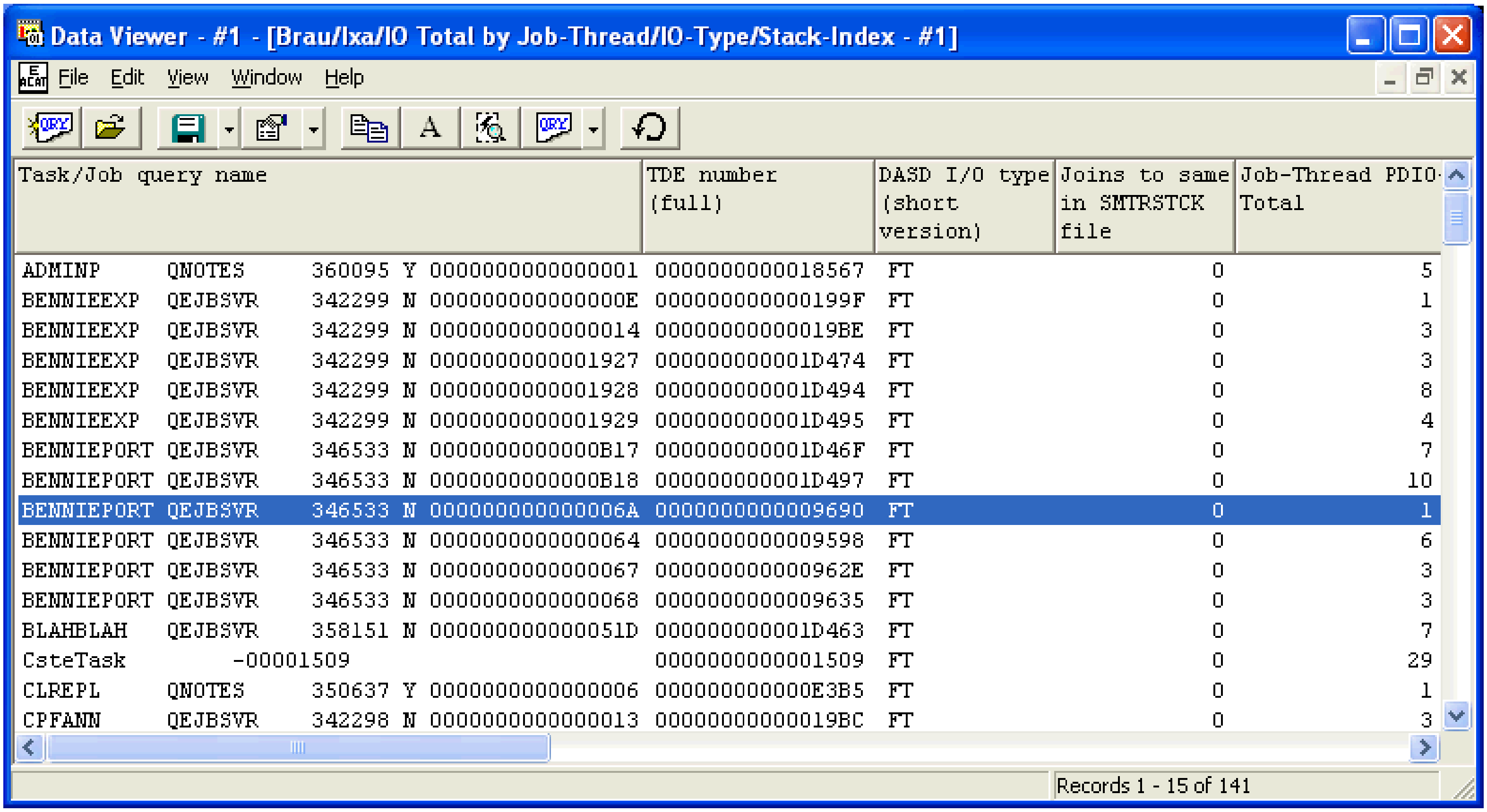
Records 1 - 9 of 141

Table of Contents | Previous | Next

5.5.16 IO Total by Job-Thread/IO-Type/Stack-Index

This report shows a breakdown of the physical disk IO events that occurred for each IO type within job/thread.

If a call stack is available for one of the events for a specific job/IO-type the STCKINDX field will be greater than 0. Double-clicking on a record will show the call stack if it is available.


Data Viewer - #1 - [Brau/Ixa/IO Total by Job-Thread/IO-Type/Stack-Index - #1]
File
Edit
View
Window
Help

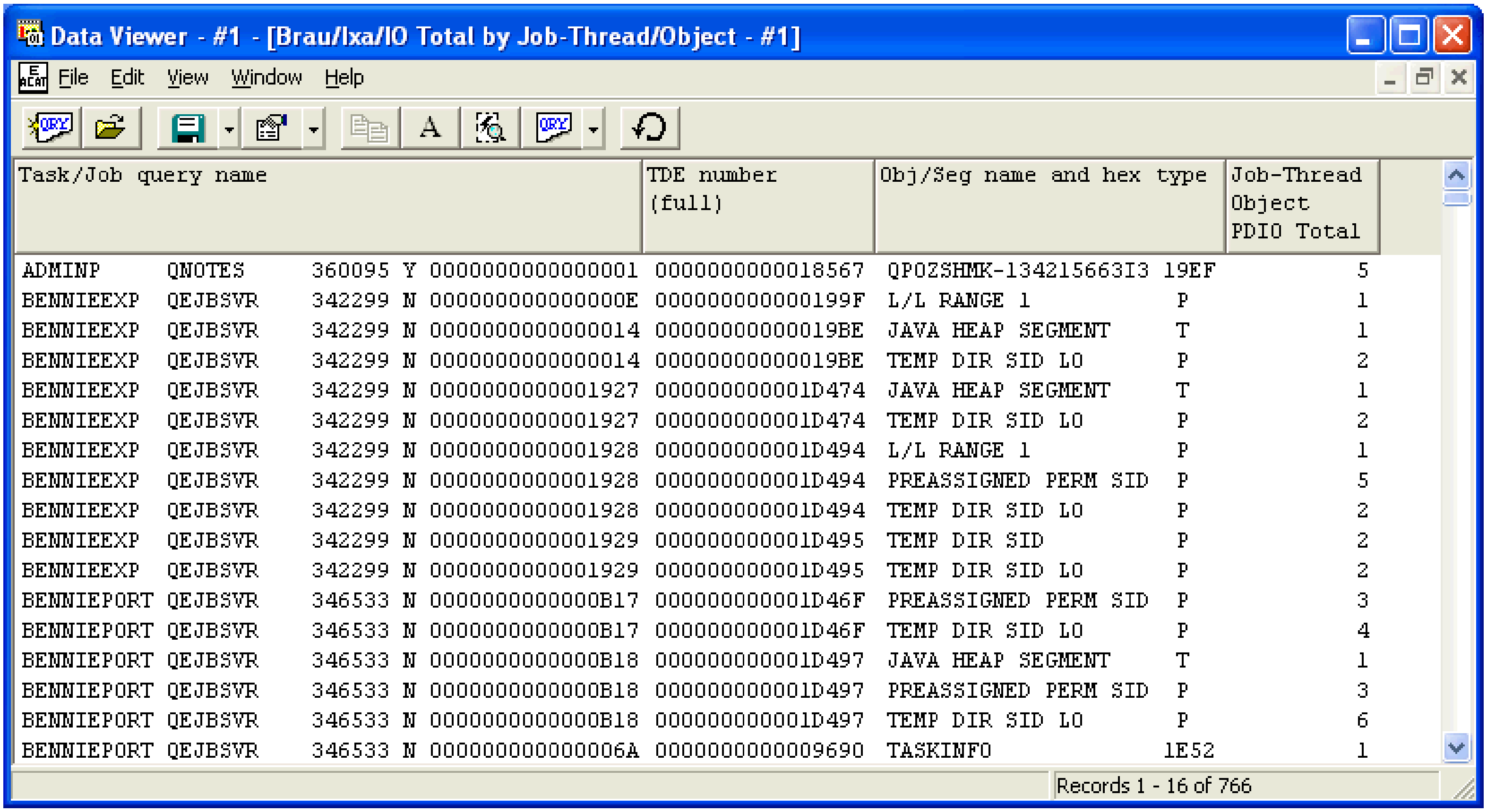
Task/Job query name	TDE number (full)	DASD I/O type (short version)	Joins to same in SMTRSTCK file	Job-Thread PDIO- Total
ADMINP QNOTES 360095 Y 0000000000000001	0000000000018567	FT	0	5
BENNIEEXP QEJBSVR 342299 N 000000000000000E	000000000000199F	FT	0	1
BENNIEEXP QEJBSVR 342299 N 0000000000000014	00000000000019BE	FT	0	3
BENNIEEXP QEJBSVR 342299 N 0000000000001927	000000000001D474	FT	0	3
BENNIEEXP QEJBSVR 342299 N 0000000000001928	000000000001D494	FT	0	8
BENNIEEXP QEJBSVR 342299 N 0000000000001929	000000000001D495	FT	0	4
BENNIEPORT QEJBSVR 346533 N 0000000000000B17	000000000001D46F	FT	0	7
BENNIEPORT QEJBSVR 346533 N 0000000000000B18	000000000001D497	FT	0	10
BENNIEPORT QEJBSVR 346533 N 000000000000006A	0000000000009690	FT	0	1
BENNIEPORT QEJBSVR 346533 N 0000000000000064	0000000000009598	FT	0	6
BENNIEPORT QEJBSVR 346533 N 0000000000000067	000000000000962E	FT	0	3
BENNIEPORT QEJBSVR 346533 N 0000000000000068	0000000000009635	FT	0	3
BLAHBLAH QEJBSVR 358151 N 000000000000051D	000000000001D463	FT	0	7
CsteTask -00001509	0000000000001509	FT	0	29
CLREPL QNOTES 350637 Y 0000000000000006	000000000000E3B5	FT	0	1
CPFANN QEJBSVR 342298 N 0000000000000013	00000000000019BC	FT	0	3

Records 1 - 15 of 141

Table of Contents Previous Next

5.5.17 IO Total by Job-Thread/Object

This report shows the total number of physical disk IO events that occurred for each job/thread by object the event was over.

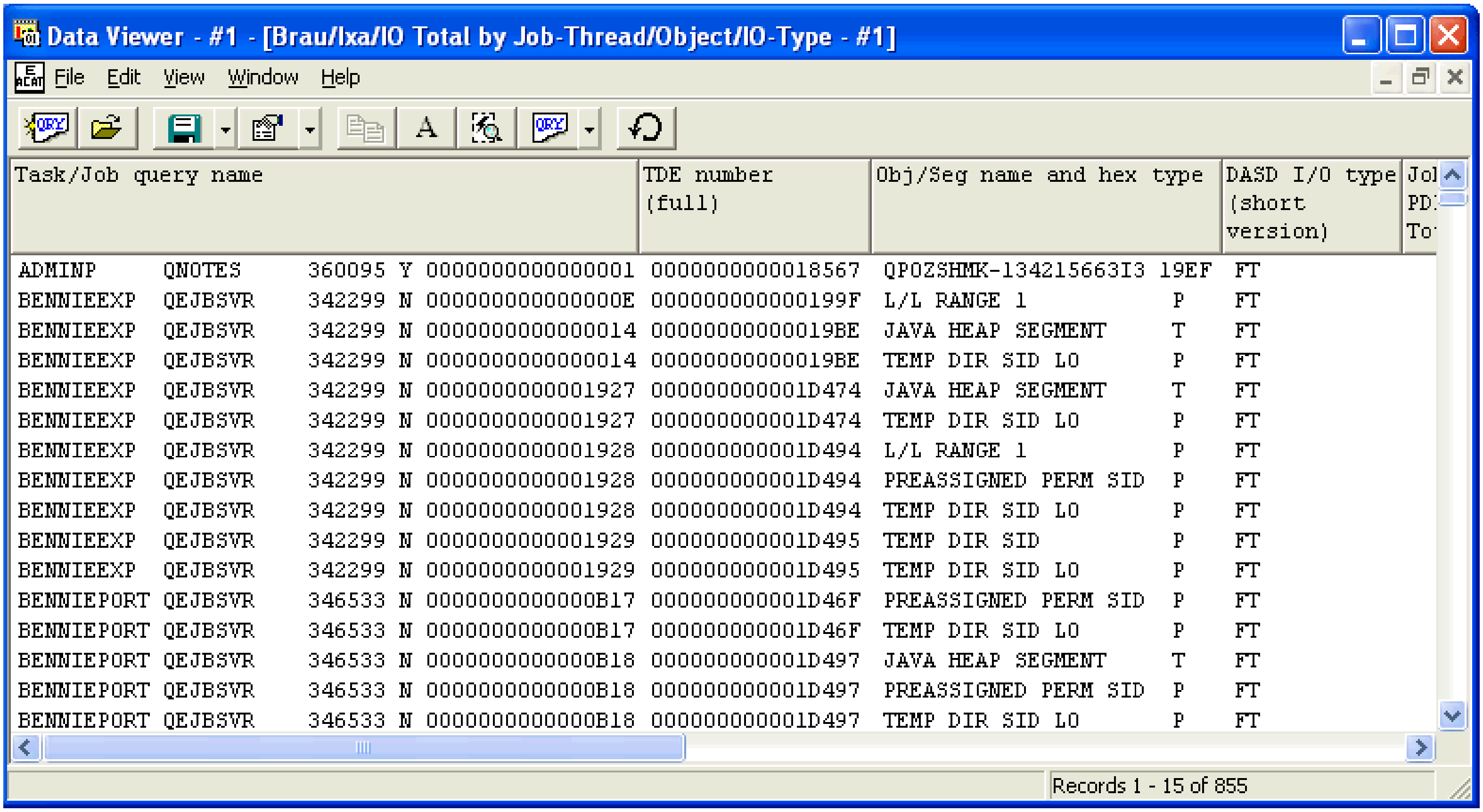

Data Viewer - #1 - [Brau/Ixa/IO Total by Job-Thread/Object - #1]
File
Edit
View
Window
Help
QRY
A
QRY
Records 1 - 16 of 766

Task/Job query name	TDE number (full)	Obj/Seg name and hex type	Job-Thread Object PDIO Total
ADMINP QNOTES 360095 Y 0000000000000001	0000000000018567	QPOZSHMK-134215663I3 19EF	5
BENNIEEXP QEJBSVR 342299 N 000000000000000E	000000000000199F	L/L RANGE 1 P	1
BENNIEEXP QEJBSVR 342299 N 0000000000000014	00000000000019BE	JAVA HEAP SEGMENT T	1
BENNIEEXP QEJBSVR 342299 N 0000000000000014	00000000000019BE	TEMP DIR SID LO P	2
BENNIEEXP QEJBSVR 342299 N 0000000000001927	000000000001D474	JAVA HEAP SEGMENT T	1
BENNIEEXP QEJBSVR 342299 N 0000000000001927	000000000001D474	TEMP DIR SID LO P	2
BENNIEEXP QEJBSVR 342299 N 0000000000001928	000000000001D494	L/L RANGE 1 P	1
BENNIEEXP QEJBSVR 342299 N 0000000000001928	000000000001D494	PREASSIGNED PERM SID P	5
BENNIEEXP QEJBSVR 342299 N 0000000000001928	000000000001D494	TEMP DIR SID LO P	2
BENNIEEXP QEJBSVR 342299 N 0000000000001929	000000000001D495	TEMP DIR SID P	2
BENNIEEXP QEJBSVR 342299 N 0000000000001929	000000000001D495	TEMP DIR SID LO P	2
BENNIEPORT QEJBSVR 346533 N 0000000000000B17	000000000001D46F	PREASSIGNED PERM SID P	3
BENNIEPORT QEJBSVR 346533 N 0000000000000B17	000000000001D46F	TEMP DIR SID LO P	4
BENNIEPORT QEJBSVR 346533 N 0000000000000B18	000000000001D497	JAVA HEAP SEGMENT T	1
BENNIEPORT QEJBSVR 346533 N 0000000000000B18	000000000001D497	PREASSIGNED PERM SID P	3
BENNIEPORT QEJBSVR 346533 N 0000000000000B18	000000000001D497	TEMP DIR SID LO P	6
BENNIEPORT QEJBSVR 346533 N 000000000000006A	0000000000009690	TASKINFO 1E52	1

Table of Contents | Previous | Next

5.5.18 IO Total by Job-Thread/Object/IO-Type

This report shows a breakdown of the physical disk IO events that occurred for each job/thread by object by IO type.


Data Viewer - #1 - [Brau/Ixa/IO Total by Job-Thread/Object/IO-Type - #1]
File
Edit
View
Window
Help

Task/Job query name	TDE number (full)	Obj/Seg name and hex type	DASD I/O type (short version)	Jo PD To
ADMINP QNOTES 360095 Y 0000000000000001	0000000000018567	QP0ZSHMK-134215663I3 19EF	FT	
BENNIEEXP QEJBSVR 342299 N 000000000000000E	000000000000199F	L/L RANGE 1 P	FT	
BENNIEEXP QEJBSVR 342299 N 0000000000000014	00000000000019BE	JAVA HEAP SEGMENT T	FT	
BENNIEEXP QEJBSVR 342299 N 0000000000000014	00000000000019BE	TEMP DIR SID LO P	FT	
BENNIEEXP QEJBSVR 342299 N 0000000000001927	000000000001D474	JAVA HEAP SEGMENT T	FT	
BENNIEEXP QEJBSVR 342299 N 0000000000001927	000000000001D474	TEMP DIR SID LO P	FT	
BENNIEEXP QEJBSVR 342299 N 0000000000001928	000000000001D494	L/L RANGE 1 P	FT	
BENNIEEXP QEJBSVR 342299 N 0000000000001928	000000000001D494	PREASSIGNED PERM SID P	FT	
BENNIEEXP QEJBSVR 342299 N 0000000000001928	000000000001D494	TEMP DIR SID LO P	FT	
BENNIEEXP QEJBSVR 342299 N 0000000000001929	000000000001D495	TEMP DIR SID P	FT	
BENNIEEXP QEJBSVR 342299 N 0000000000001929	000000000001D495	TEMP DIR SID LO P	FT	
BENNIEPORT QEJBSVR 346533 N 0000000000000B17	000000000001D46F	PREASSIGNED PERM SID P	FT	
BENNIEPORT QEJBSVR 346533 N 0000000000000B17	000000000001D46F	TEMP DIR SID LO P	FT	
BENNIEPORT QEJBSVR 346533 N 0000000000000B18	000000000001D497	JAVA HEAP SEGMENT T	FT	
BENNIEPORT QEJBSVR 346533 N 0000000000000B18	000000000001D497	PREASSIGNED PERM SID P	FT	
BENNIEPORT QEJBSVR 346533 N 0000000000000B18	000000000001D497	TEMP DIR SID LO P	FT	

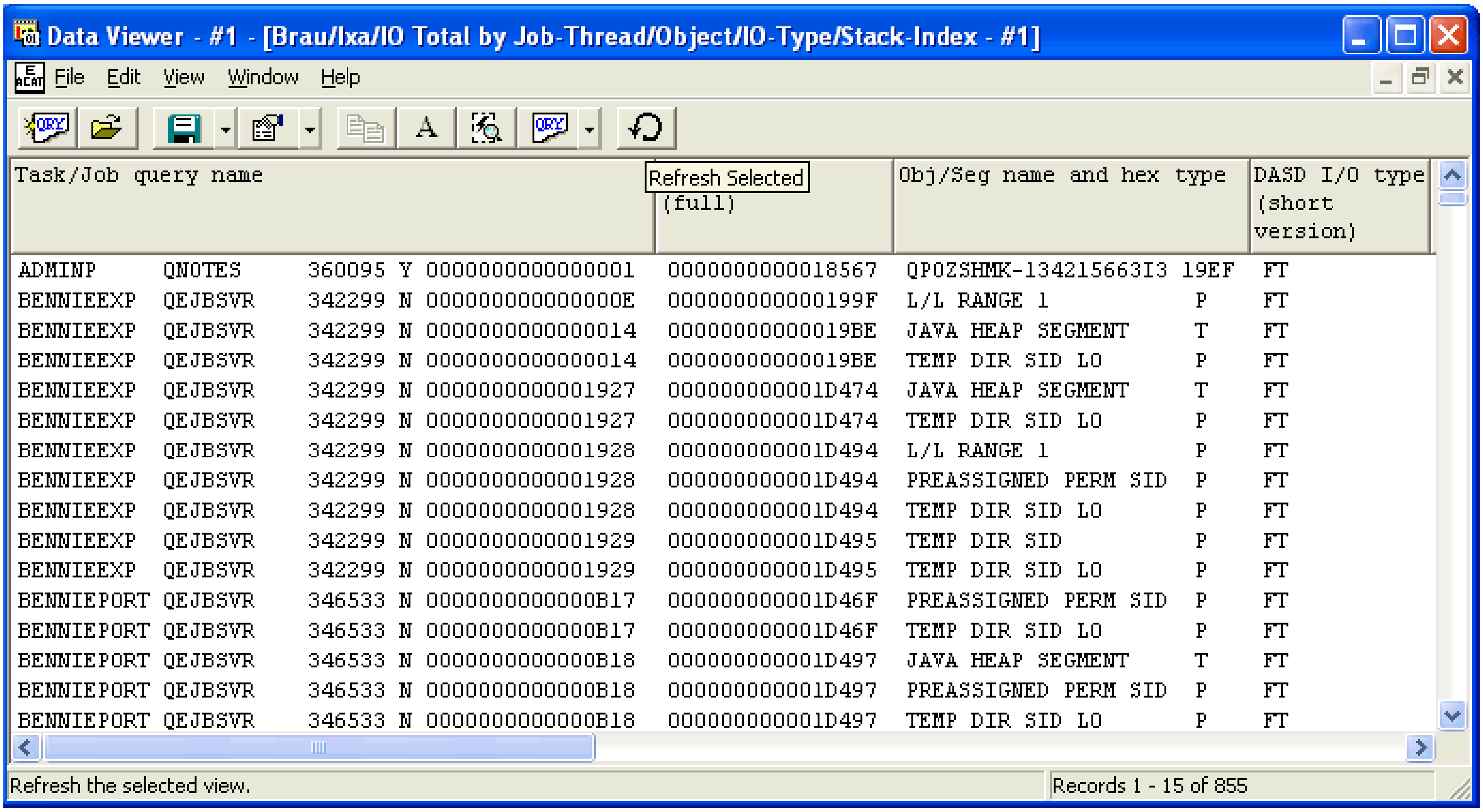
Records 1 - 15 of 855

Table of Contents | Previous | Next

5.5.19 IO Total by Job-Thread/Object/IO-Type/Stack-Index

This report shows a breakdown of the physical disk IO events that occurred for each job/thread by object by IO type.

If a call stack is available the STCKINDX field will be greater than 0. Double-clicking on a record will show the call stack if it is available.


Data Viewer - #1 - [Brau/lxa/IO Total by Job-Thread/Object/IO-Type/Stack-Index - #1]
File
Edit
View
Window
Help
QRY
A
QRY
Refresh Selected

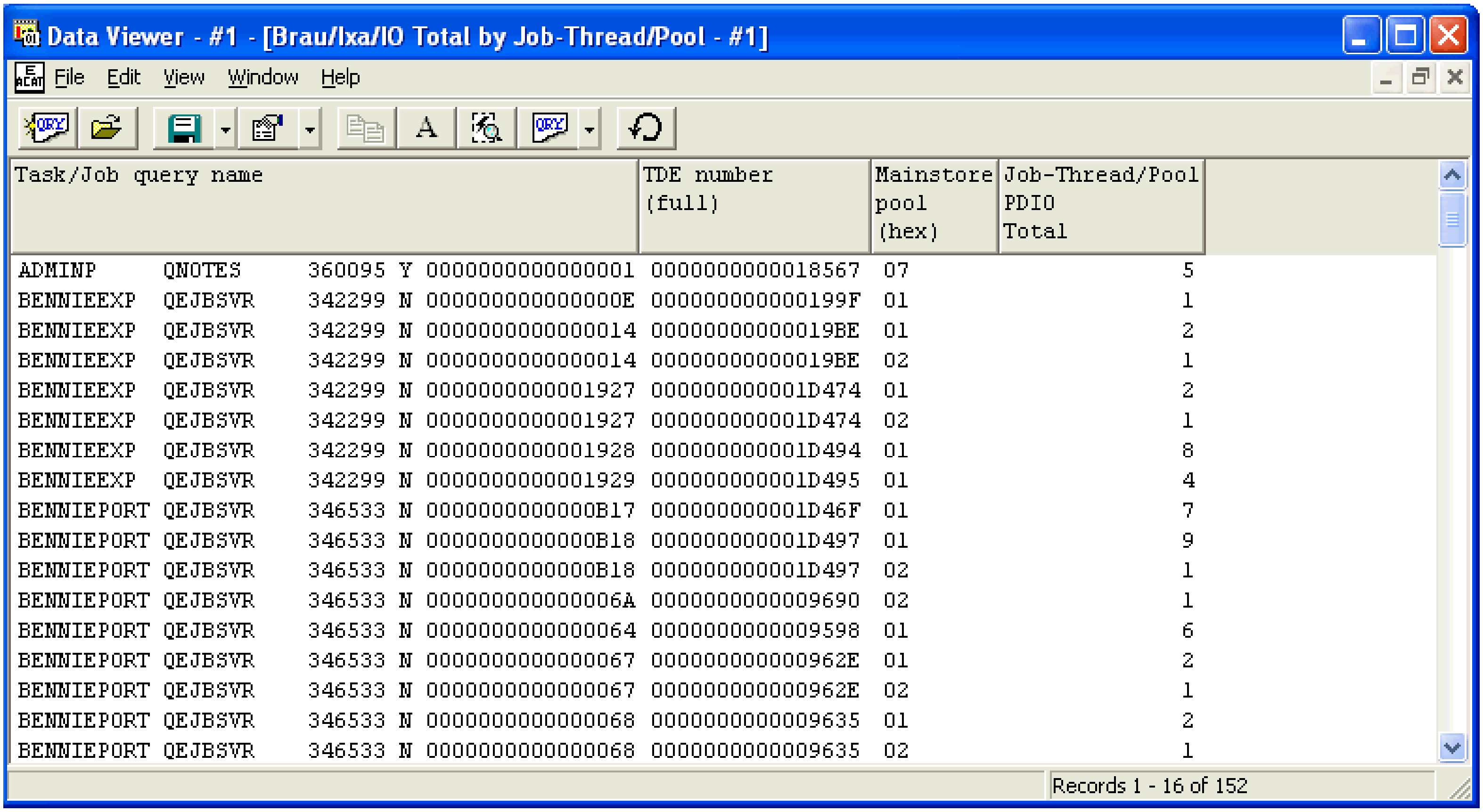
```
Task/Job query name                                  (full)              Obj/Seg name and hex type   DASD I/O type
                                                                                                     (short
                                                                                                     version)
ADMINP      QNOTES      360095 Y 0000000000000001   0000000000018567   QPOZSHMK-1342156631З 19EF   FT
BENNIEEXP   QEJBSVR     342299 N 000000000000000E   000000000000199F   L/L RANGE 1             P   FT
BENNIEEXP   QEJBSVR     342299 N 0000000000000014   00000000000019BE   JAVA HEAP SEGMENT       T   FT
BENNIEEXP   QEJBSVR     342299 N 0000000000000014   00000000000019BE   TEMP DIR SID LO         P   FT
BENNIEEXP   QEJBSVR     342299 N 0000000000001927   000000000001D474   JAVA HEAP SEGMENT       T   FT
BENNIEEXP   QEJBSVR     342299 N 0000000000001927   000000000001D474   TEMP DIR SID LO         P   FT
BENNIEEXP   QEJBSVR     342299 N 0000000000001928   000000000001D494   L/L RANGE 1             P   FT
BENNIEEXP   QEJBSVR     342299 N 0000000000001928   000000000001D494   PREASSIGNED PERM SID    P   FT
BENNIEEXP   QEJBSVR     342299 N 0000000000001928   000000000001D494   TEMP DIR SID LO         P   FT
BENNIEEXP   QEJBSVR     342299 N 0000000000001929   000000000001D495   TEMP DIR SID            P   FT
BENNIEEXP   QEJBSVR     342299 N 0000000000001929   000000000001D495   TEMP DIR SID LO         P   FT
BENNIEPORT  QEJBSVR     346533 N 0000000000000B17   000000000001D46F   PREASSIGNED PERM SID    P   FT
BENNIEPORT  QEJBSVR     346533 N 0000000000000B17   000000000001D46F   TEMP DIR SID LO         P   FT
BENNIEPORT  QEJBSVR     346533 N 0000000000000B18   000000000001D497   JAVA HEAP SEGMENT       T   FT
BENNIEPORT  QEJBSVR     346533 N 0000000000000B18   000000000001D497   PREASSIGNED PERM SID    P   FT
BENNIEPORT  QEJBSVR     346533 N 0000000000000B18   000000000001D497   TEMP DIR SID LO         P   FT
```

Refresh the selected view.   Records 1 - 15 of 855

Table of Contents | Previous | Next

5.5.20 IO Total by Job-Thread/Pool

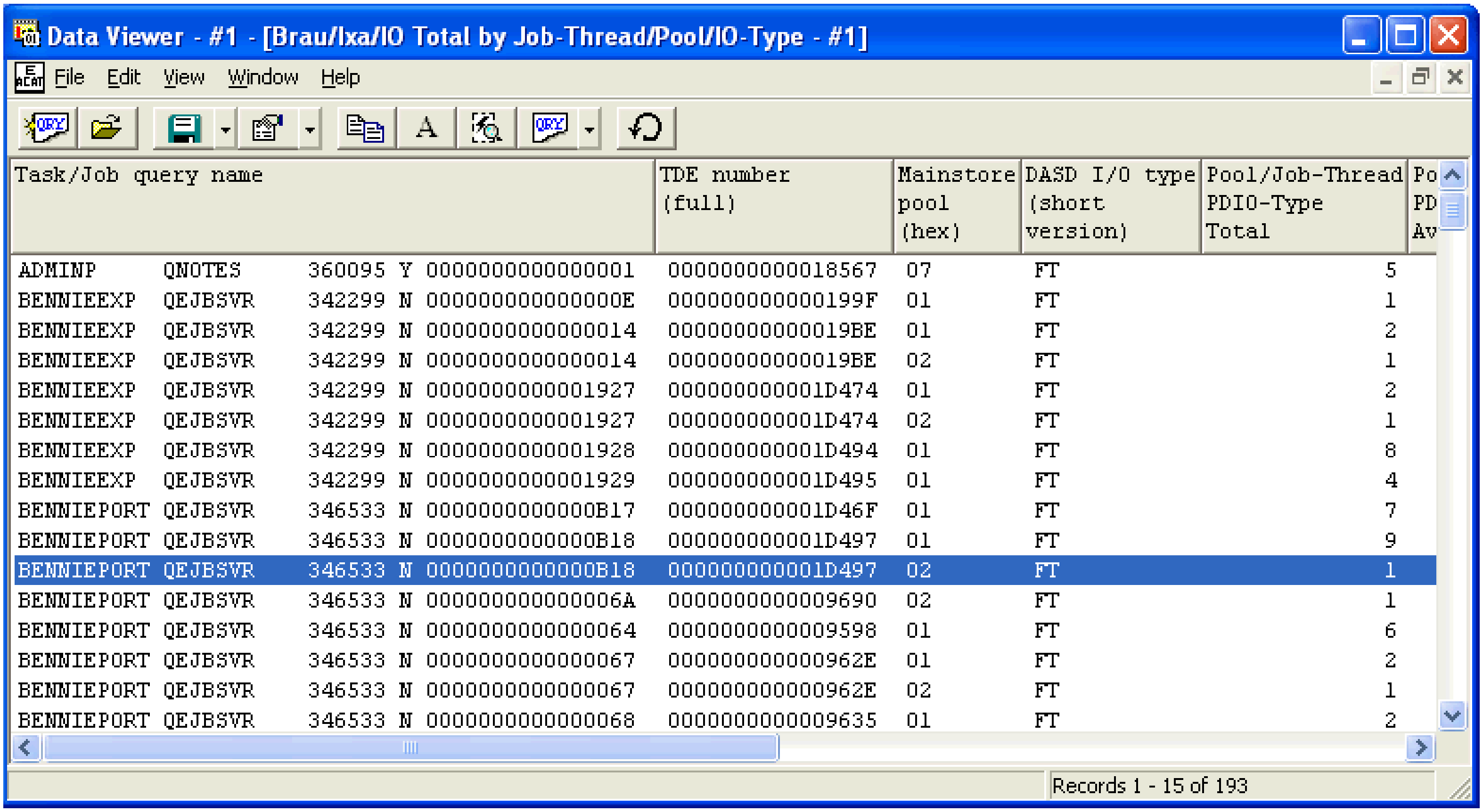
This report shows the total number of physical disk IO events that occurred for each job/thread by mainstorage pool. The pool ID is listed in HEX format within this report.


Data Viewer - #1 - [Brau/Ixa/IO Total by Job-Thread/Pool - #1]
File
Edit
View
Window
Help
QRY
A
QRY
Records 1 - 16 of 152

Task/Job query name	TDE number (full)	Mainstore pool (hex)	Job-Thread/Pool PDIO Total
ADMINP QNOTES 360095 Y 0000000000000001	0000000000018567	07	5
BENNIEEXP QEJBSVR 342299 N 000000000000000E	000000000000199F	01	1
BENNIEEXP QEJBSVR 342299 N 0000000000000014	00000000000019BE	01	2
BENNIEEXP QEJBSVR 342299 N 0000000000000014	00000000000019BE	02	1
BENNIEEXP QEJBSVR 342299 N 0000000000001927	000000000001D474	01	2
BENNIEEXP QEJBSVR 342299 N 0000000000001927	000000000001D474	02	1
BENNIEEXP QEJBSVR 342299 N 0000000000001928	000000000001D494	01	8
BENNIEEXP QEJBSVR 342299 N 0000000000001929	000000000001D495	01	4
BENNIEPORT QEJBSVR 346533 N 0000000000000B17	000000000001D46F	01	7
BENNIEPORT QEJBSVR 346533 N 0000000000000B18	000000000001D497	01	9
BENNIEPORT QEJBSVR 346533 N 0000000000000B18	000000000001D497	02	1
BENNIEPORT QEJBSVR 346533 N 000000000000006A	0000000000009690	02	1
BENNIEPORT QEJBSVR 346533 N 0000000000000064	0000000000009598	01	6
BENNIEPORT QEJBSVR 346533 N 0000000000000067	000000000000962E	01	2
BENNIEPORT QEJBSVR 346533 N 0000000000000067	000000000000962E	02	1
BENNIEPORT QEJBSVR 346533 N 0000000000000068	0000000000009635	01	2
BENNIEPORT QEJBSVR 346533 N 0000000000000068	0000000000009635	02	1

5.5.21 IO Total by Job-Thread/Pool/IO-Type

This report shows a breakdown of the physical disk IO events that occurred for each job/thread by IO type within each mainstorage pool. The pool ID is listed in HEX format within this report.


Data Viewer - #1 - [Brau/Ixa/IO Total by Job-Thread/Pool/IO-Type - #1]
File
Edit
View
Window
Help

Task/Job query name	TDE number (full)	Mainstore pool (hex)	DASD I/O type (short version)	Pool/Job-Thread PDIO-Type Total	Po PD Av
ADMINP QNOTES 360095 Y 0000000000000001	0000000000018567	07	FT	5	
BENNIEEXP QEJBSVR 342299 N 000000000000000E	000000000000199F	01	FT	1	
BENNIEEXP QEJBSVR 342299 N 0000000000000014	00000000000019BE	01	FT	2	
BENNIEEXP QEJBSVR 342299 N 0000000000000014	00000000000019BE	02	FT	1	
BENNIEEXP QEJBSVR 342299 N 0000000000001927	000000000001D474	01	FT	2	
BENNIEEXP QEJBSVR 342299 N 0000000000001927	000000000001D474	02	FT	1	
BENNIEEXP QEJBSVR 342299 N 0000000000001928	000000000001D494	01	FT	8	
BENNIEEXP QEJBSVR 342299 N 0000000000001929	000000000001D495	01	FT	4	
BENNIEPORT QEJBSVR 346533 N 0000000000000B17	000000000001D46F	01	FT	7	
BENNIEPORT QEJBSVR 346533 N 0000000000000B18	000000000001D497	01	FT	9	
BENNIEPORT QEJBSVR 346533 N 0000000000000B18	000000000001D497	02	FT	1	
BENNIEPORT QEJBSVR 346533 N 000000000000006A	0000000000009690	02	FT	1	
BENNIEPORT QEJBSVR 346533 N 0000000000000064	0000000000009598	01	FT	6	
BENNIEPORT QEJBSVR 346533 N 0000000000000067	000000000000962E	01	FT	2	
BENNIEPORT QEJBSVR 346533 N 0000000000000067	000000000000962E	02	FT	1	
BENNIEPORT QEJBSVR 346533 N 0000000000000068	0000000000009635	01	FT	2	

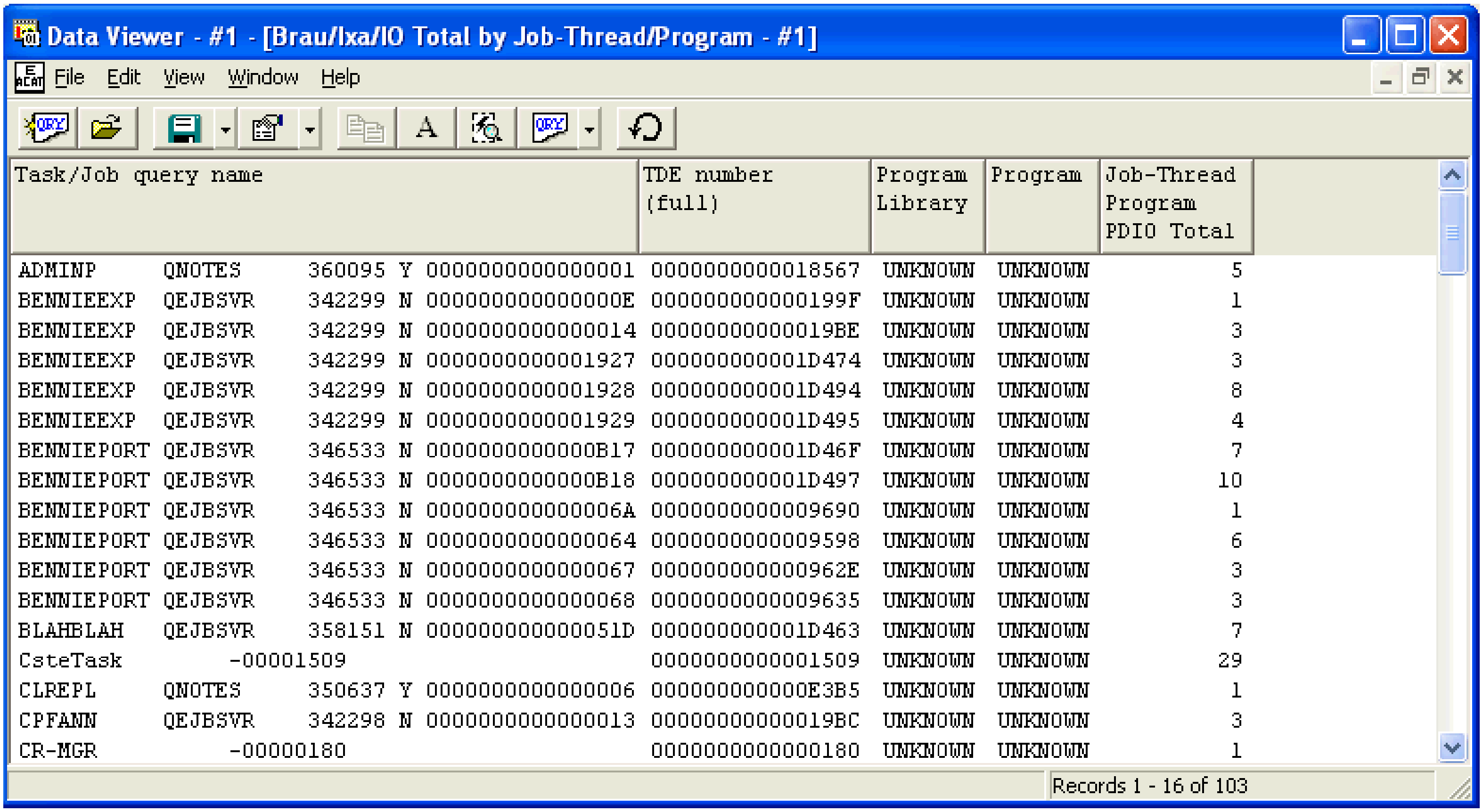
Records 1 - 15 of 193

Table of Contents Previous Next

5.5.22 IO Total by Job-Thread/Program

This report shows the total physical disk IO events that occurred for each program within each job/thread.

Note: In order for program names to be provided, *MIENTRY and *MIEXIT events must be included in the PEX collection.


Data Viewer - #1 - [Brau/Ixa/IO Total by Job-Thread/Program - #1]
File
Edit
View
Window
Help

Task/Job query name	TDE number (full)	Program Library	Program	Job-Thread Program PDIO Total
ADMINP QNOTES 360095 Y 0000000000000001	0000000000018567	UNKNOWN	UNKNOWN	5
BENNIEEXP QEJBSVR 342299 N 000000000000000E	000000000000199F	UNKNOWN	UNKNOWN	1
BENNIEEXP QEJBSVR 342299 N 0000000000000014	00000000000019BE	UNKNOWN	UNKNOWN	3
BENNIEEXP QEJBSVR 342299 N 0000000000001927	000000000001D474	UNKNOWN	UNKNOWN	3
BENNIEEXP QEJBSVR 342299 N 0000000000001928	000000000001D494	UNKNOWN	UNKNOWN	8
BENNIEEXP QEJBSVR 342299 N 0000000000001929	000000000001D495	UNKNOWN	UNKNOWN	4
BENNIEPORT QEJBSVR 346533 N 0000000000000B17	000000000001D46F	UNKNOWN	UNKNOWN	7
BENNIEPORT QEJBSVR 346533 N 0000000000000B18	000000000001D497	UNKNOWN	UNKNOWN	10
BENNIEPORT QEJBSVR 346533 N 000000000000006A	0000000000009690	UNKNOWN	UNKNOWN	1
BENNIEPORT QEJBSVR 346533 N 0000000000000064	0000000000009598	UNKNOWN	UNKNOWN	6
BENNIEPORT QEJBSVR 346533 N 0000000000000067	000000000000962E	UNKNOWN	UNKNOWN	3
BENNIEPORT QEJBSVR 346533 N 0000000000000068	0000000000009635	UNKNOWN	UNKNOWN	3
BLAHBLAH QEJBSVR 358151 N 000000000000051D	000000000001D463	UNKNOWN	UNKNOWN	7
CsteTask -00001509	0000000000001509	UNKNOWN	UNKNOWN	29
CLREPL QNOTES 350637 Y 0000000000000006	000000000000E3B5	UNKNOWN	UNKNOWN	1
CPFANN QEJBSVR 342298 N 0000000000000013	00000000000019BC	UNKNOWN	UNKNOWN	3
CR-MGR -00000180	0000000000000180	UNKNOWN	UNKNOWN	1

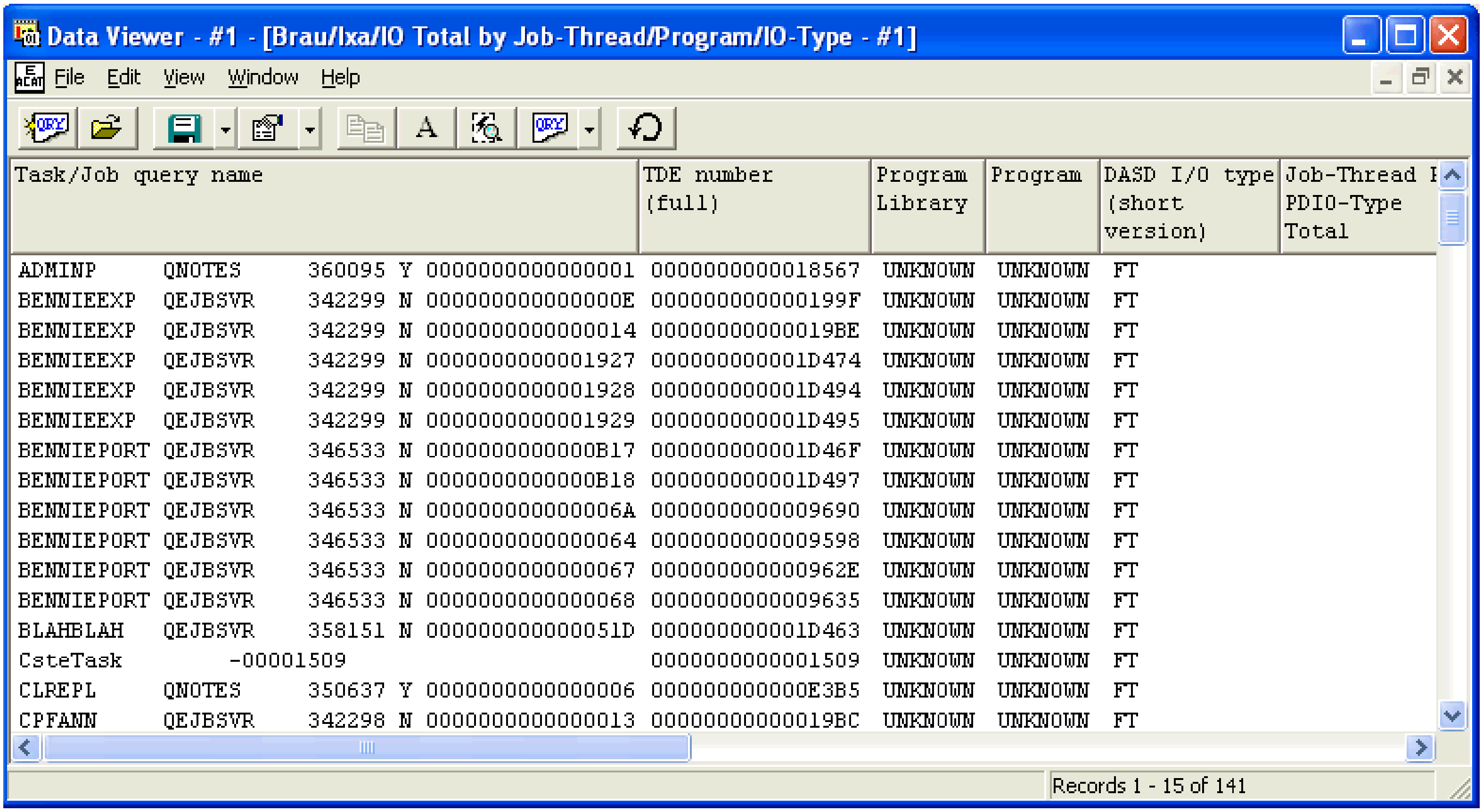
Records 1 - 16 of 103

Table of Contents | Previous | Next

5.5.23 IO Total by Job-Thread/Program/IO-Type

This report shows a breakdown of physical disk IO events that occurred for each IO type by program within each job/thread.

Note: In order for program names to be provided, *MIENTRY and *MIEXIT events must be included in the PEX collection.


Data Viewer - #1 - [Brau/lxa/IO Total by Job-Thread/Program/IO-Type - #1]
File
Edit
View
Window
Help
QRY
A
QRY
Records 1 - 15 of 141

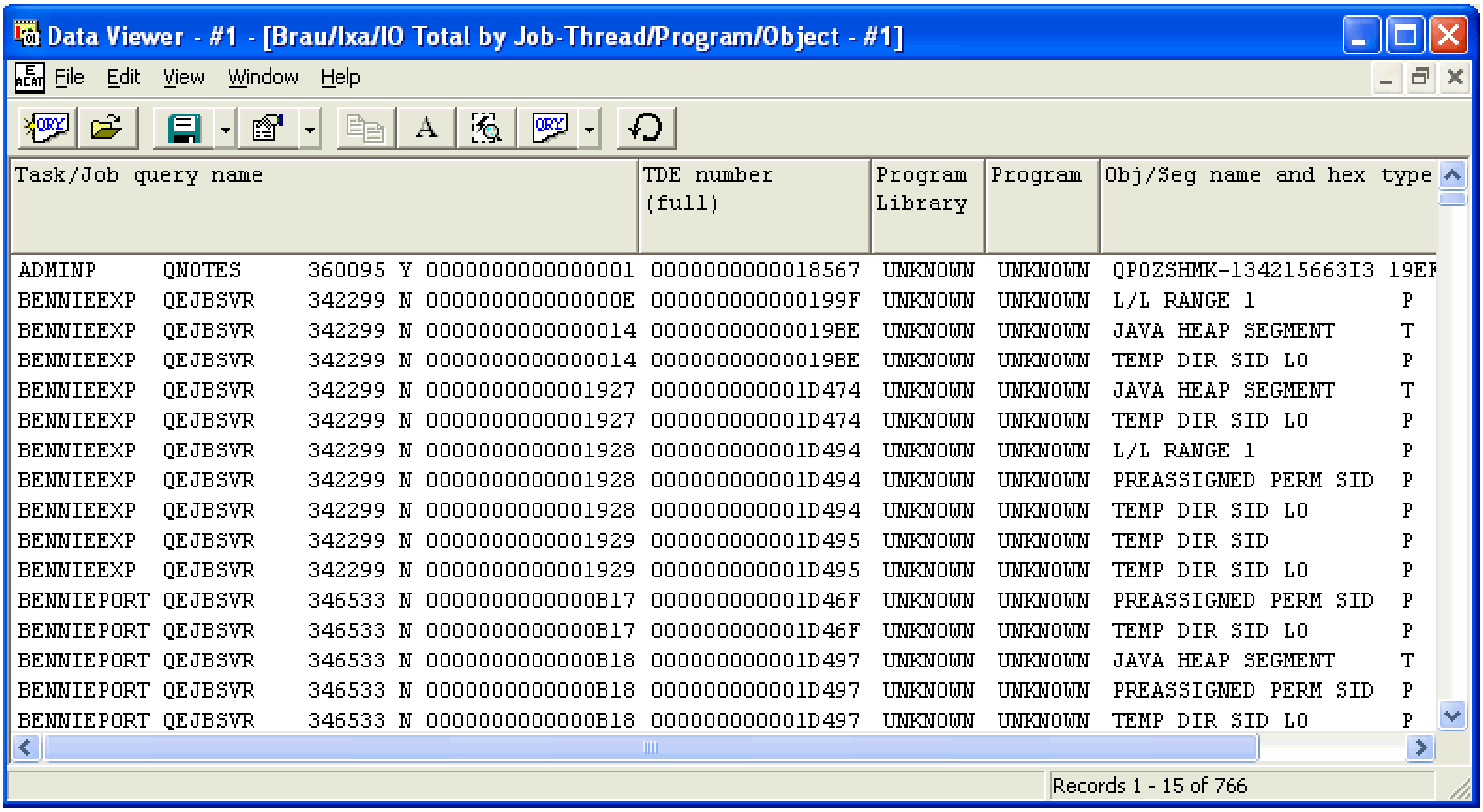
Task/Job query name	TDE number (full)	Program Library	Program	DASD I/O type (short version)	Job-Thread PDIO-Type Total	I
ADMINP QNOTES 360095 Y 0000000000000001	0000000000018567	UNKNOWN	UNKNOWN	FT		
BENNIEEXP QEJBSVR 342299 N 000000000000000E	000000000000199F	UNKNOWN	UNKNOWN	FT		
BENNIEEXP QEJBSVR 342299 N 0000000000000014	00000000000019BE	UNKNOWN	UNKNOWN	FT		
BENNIEEXP QEJBSVR 342299 N 0000000000001927	000000000001D474	UNKNOWN	UNKNOWN	FT		
BENNIEEXP QEJBSVR 342299 N 0000000000001928	000000000001D494	UNKNOWN	UNKNOWN	FT		
BENNIEEXP QEJBSVR 342299 N 0000000000001929	000000000001D495	UNKNOWN	UNKNOWN	FT		
BENNIEPORT QEJBSVR 346533 N 0000000000000B17	000000000001D46F	UNKNOWN	UNKNOWN	FT		
BENNIEPORT QEJBSVR 346533 N 0000000000000B18	000000000001D497	UNKNOWN	UNKNOWN	FT		
BENNIEPORT QEJBSVR 346533 N 000000000000006A	0000000000009690	UNKNOWN	UNKNOWN	FT		
BENNIEPORT QEJBSVR 346533 N 0000000000000064	0000000000009598	UNKNOWN	UNKNOWN	FT		
BENNIEPORT QEJBSVR 346533 N 0000000000000067	000000000000962E	UNKNOWN	UNKNOWN	FT		
BENNIEPORT QEJBSVR 346533 N 0000000000000068	0000000000009635	UNKNOWN	UNKNOWN	FT		
BLAHBLAH QEJBSVR 358151 N 000000000000051D	000000000001D463	UNKNOWN	UNKNOWN	FT		
CsteTask -00001509	0000000000001509	UNKNOWN	UNKNOWN	FT		
CLREPL QNOTES 350637 Y 0000000000000006	000000000000E3B5	UNKNOWN	UNKNOWN	FT		
CPFANN QEJBSVR 342298 N 0000000000000013	00000000000019BC	UNKNOWN	UNKNOWN	FT		

Table of Contents Previous Next

5.5.24 IO Total by Job-Thread/Program/Object

This report shows the total physical disk IO events that occurred for each object within program within each job/thread.

Note: In order for program names to be provided, *MIENTRY and *MIEXIT events must be included in the PEX collection.


Data Viewer - #1 - [Brau/Ixa/IO Total by Job-Thread/Program/Object - #1]
File
Edit
View
Window
Help

Task/Job query name	TDE number (full)	Program Library	Program	Obj/Seg name and hex type
ADMINP QNOTES 360095 Y 0000000000000001	0000000000018567	UNKNOWN	UNKNOWN	QPOZSHMK-13421566313 19EF
BENNIEEXP QEJBSVR 342299 N 000000000000000E	000000000000199F	UNKNOWN	UNKNOWN	L/L RANGE 1 P
BENNIEEXP QEJBSVR 342299 N 0000000000000014	00000000000019BE	UNKNOWN	UNKNOWN	JAVA HEAP SEGMENT T
BENNIEEXP QEJBSVR 342299 N 0000000000000014	00000000000019BE	UNKNOWN	UNKNOWN	TEMP DIR SID LO P
BENNIEEXP QEJBSVR 342299 N 0000000000001927	000000000001D474	UNKNOWN	UNKNOWN	JAVA HEAP SEGMENT T
BENNIEEXP QEJBSVR 342299 N 0000000000001927	000000000001D474	UNKNOWN	UNKNOWN	TEMP DIR SID LO P
BENNIEEXP QEJBSVR 342299 N 0000000000001928	000000000001D494	UNKNOWN	UNKNOWN	L/L RANGE 1 P
BENNIEEXP QEJBSVR 342299 N 0000000000001928	000000000001D494	UNKNOWN	UNKNOWN	PREASSIGNED PERM SID P
BENNIEEXP QEJBSVR 342299 N 0000000000001928	000000000001D494	UNKNOWN	UNKNOWN	TEMP DIR SID LO P
BENNIEEXP QEJBSVR 342299 N 0000000000001929	000000000001D495	UNKNOWN	UNKNOWN	TEMP DIR SID P
BENNIEEXP QEJBSVR 342299 N 0000000000001929	000000000001D495	UNKNOWN	UNKNOWN	TEMP DIR SID LO P
BENNIEPORT QEJBSVR 346533 N 0000000000000B17	000000000001D46F	UNKNOWN	UNKNOWN	PREASSIGNED PERM SID P
BENNIEPORT QEJBSVR 346533 N 0000000000000B17	000000000001D46F	UNKNOWN	UNKNOWN	TEMP DIR SID LO P
BENNIEPORT QEJBSVR 346533 N 0000000000000B18	000000000001D497	UNKNOWN	UNKNOWN	JAVA HEAP SEGMENT T
BENNIEPORT QEJBSVR 346533 N 0000000000000B18	000000000001D497	UNKNOWN	UNKNOWN	PREASSIGNED PERM SID P
BENNIEPORT QEJBSVR 346533 N 0000000000000B18	000000000001D497	UNKNOWN	UNKNOWN	TEMP DIR SID LO P

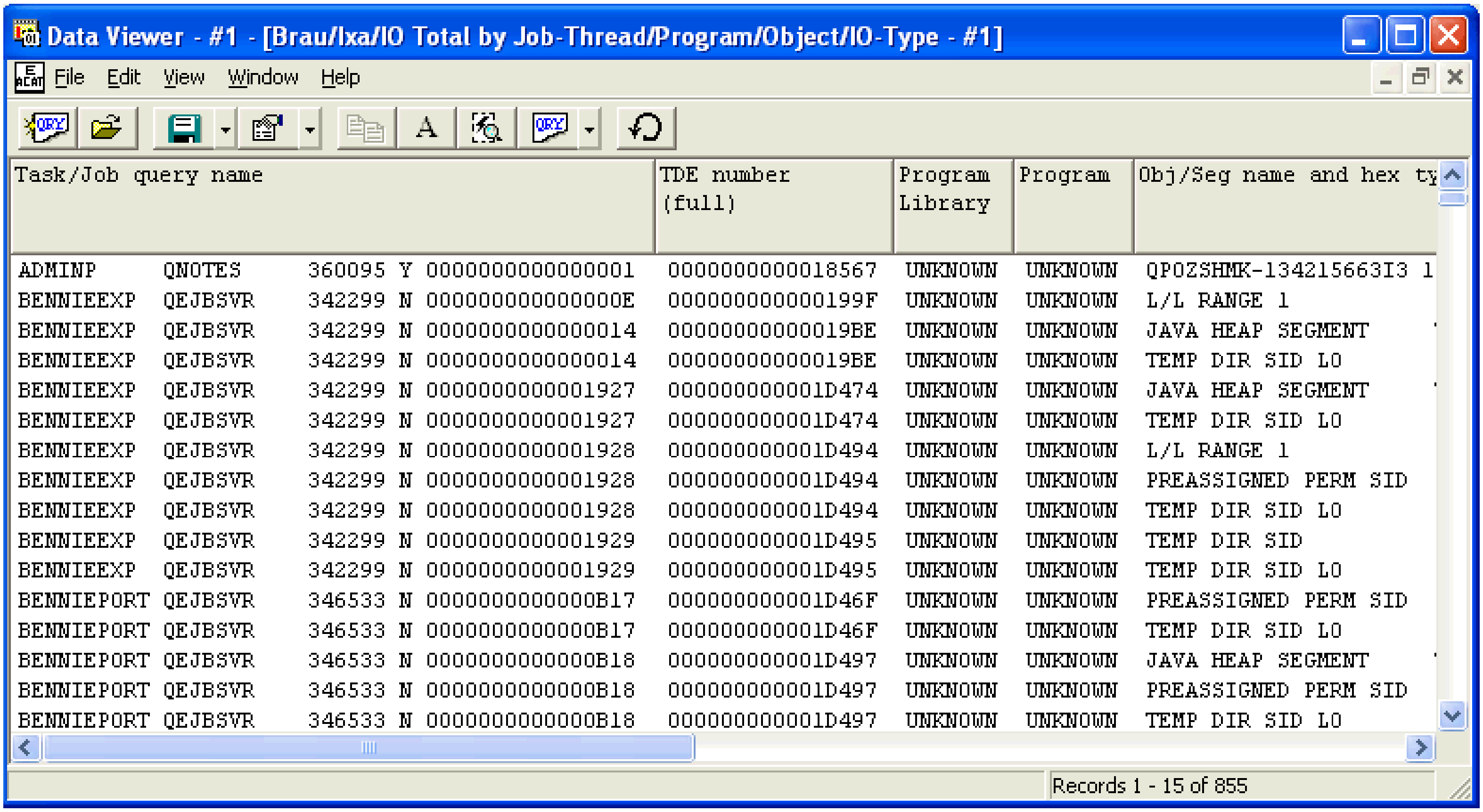
Records 1 - 15 of 766

Table of Contents Previous Next

5.5.25 IO Total by Job-Thread/Program/Object/IO-Type

This report shows a breakdown of physical disk IO events that occurred for each IO type by object within program within each job/thread.

Note: In order for program names to be provided, *MIENTRY and *MIEXIT events must be included in the PEX collection.



Data Viewer - #1 - [Brau/Ixa/IO Total by Job-Thread/Program/Object/IO-Type - #1]
File Edit View Window Help

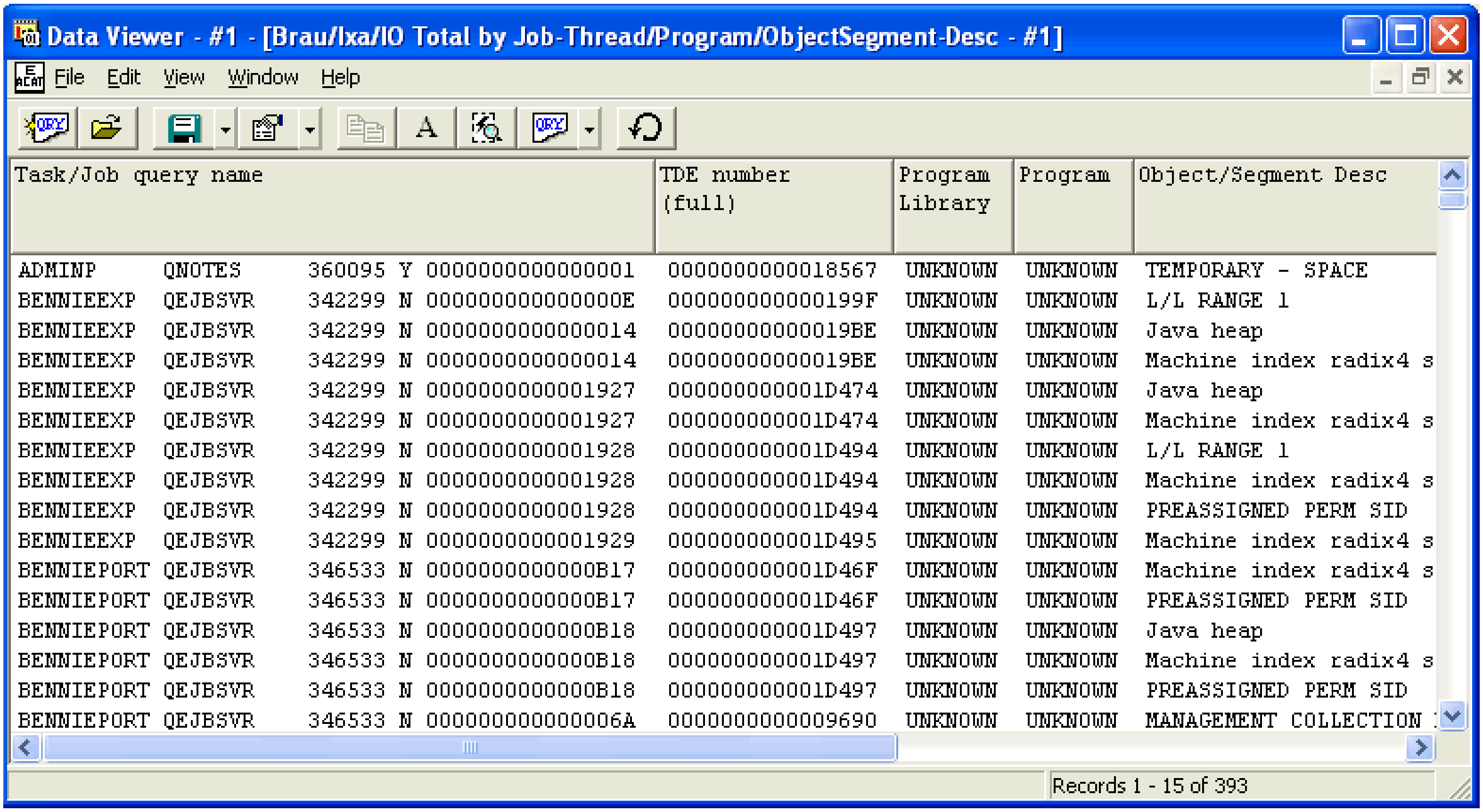
Task/Job query name	TDE number (full)	Program Library	Program	Obj/Seg name and hex ty
ADMINP QNOTES 360095 Y 0000000000000001	0000000000018567	UNKNOWN	UNKNOWN	QPOZSHMK-134215663I3 1
BENNIEEXP QEJBSVR 342299 N 000000000000000E	000000000000199F	UNKNOWN	UNKNOWN	L/L RANGE 1
BENNIEEXP QEJBSVR 342299 N 0000000000000014	00000000000019BE	UNKNOWN	UNKNOWN	JAVA HEAP SEGMENT
BENNIEEXP QEJBSVR 342299 N 0000000000000014	00000000000019BE	UNKNOWN	UNKNOWN	TEMP DIR SID LO
BENNIEEXP QEJBSVR 342299 N 0000000000001927	000000000001D474	UNKNOWN	UNKNOWN	JAVA HEAP SEGMENT
BENNIEEXP QEJBSVR 342299 N 0000000000001927	000000000001D474	UNKNOWN	UNKNOWN	TEMP DIR SID LO
BENNIEEXP QEJBSVR 342299 N 0000000000001928	000000000001D494	UNKNOWN	UNKNOWN	L/L RANGE 1
BENNIEEXP QEJBSVR 342299 N 0000000000001928	000000000001D494	UNKNOWN	UNKNOWN	PREASSIGNED PERM SID
BENNIEEXP QEJBSVR 342299 N 0000000000001928	000000000001D494	UNKNOWN	UNKNOWN	TEMP DIR SID LO
BENNIEEXP QEJBSVR 342299 N 0000000000001929	000000000001D495	UNKNOWN	UNKNOWN	TEMP DIR SID
BENNIEEXP QEJBSVR 342299 N 0000000000001929	000000000001D495	UNKNOWN	UNKNOWN	TEMP DIR SID LO
BENNIEPORT QEJBSVR 346533 N 0000000000000B17	000000000001D46F	UNKNOWN	UNKNOWN	PREASSIGNED PERM SID
BENNIEPORT QEJBSVR 346533 N 0000000000000B17	000000000001D46F	UNKNOWN	UNKNOWN	TEMP DIR SID LO
BENNIEPORT QEJBSVR 346533 N 0000000000000B18	000000000001D497	UNKNOWN	UNKNOWN	JAVA HEAP SEGMENT
BENNIEPORT QEJBSVR 346533 N 0000000000000B18	000000000001D497	UNKNOWN	UNKNOWN	PREASSIGNED PERM SID
BENNIEPORT QEJBSVR 346533 N 0000000000000B18	000000000001D497	UNKNOWN	UNKNOWN	TEMP DIR SID LO

Records 1 - 15 of 855

5.5.26 IO Total by Job-Thread/Program/ObjectSegment-Desc

This report shows the total physical disk IO events that occurred for each object type within program within each job/thread. The object type descriptions are listed in this report.

Note: In order for program names to be provided, *MIENTRY and *MIEXIT events must be included in the PEX collection.


Data Viewer - #1 - [Brau/Ixa/IO Total by Job-Thread/Program/ObjectSegment-Desc - #1]
File
Edit
View
Window
Help

Task/Job query name	TDE number (full)	Program Library	Program	Object/Segment Desc
ADMINP QNOTES 360095 Y 0000000000000001	0000000000018567	UNKNOWN	UNKNOWN	TEMPORARY - SPACE
BENNIEEXP QEJBSVR 342299 N 000000000000000E	000000000000199F	UNKNOWN	UNKNOWN	L/L RANGE 1
BENNIEEXP QEJBSVR 342299 N 0000000000000014	00000000000019BE	UNKNOWN	UNKNOWN	Java heap
BENNIEEXP QEJBSVR 342299 N 0000000000000014	00000000000019BE	UNKNOWN	UNKNOWN	Machine index radix4 s
BENNIEEXP QEJBSVR 342299 N 0000000000001927	000000000001D474	UNKNOWN	UNKNOWN	Java heap
BENNIEEXP QEJBSVR 342299 N 0000000000001927	000000000001D474	UNKNOWN	UNKNOWN	Machine index radix4 s
BENNIEEXP QEJBSVR 342299 N 0000000000001928	000000000001D494	UNKNOWN	UNKNOWN	L/L RANGE 1
BENNIEEXP QEJBSVR 342299 N 0000000000001928	000000000001D494	UNKNOWN	UNKNOWN	Machine index radix4 s
BENNIEEXP QEJBSVR 342299 N 0000000000001928	000000000001D494	UNKNOWN	UNKNOWN	PREASSIGNED PERM SID
BENNIEEXP QEJBSVR 342299 N 0000000000001929	000000000001D495	UNKNOWN	UNKNOWN	Machine index radix4 s
BENNIEPORT QEJBSVR 346533 N 0000000000000B17	000000000001D46F	UNKNOWN	UNKNOWN	Machine index radix4 s
BENNIEPORT QEJBSVR 346533 N 0000000000000B17	000000000001D46F	UNKNOWN	UNKNOWN	PREASSIGNED PERM SID
BENNIEPORT QEJBSVR 346533 N 0000000000000B18	000000000001D497	UNKNOWN	UNKNOWN	Java heap
BENNIEPORT QEJBSVR 346533 N 0000000000000B18	000000000001D497	UNKNOWN	UNKNOWN	Machine index radix4 s
BENNIEPORT QEJBSVR 346533 N 0000000000000B18	000000000001D497	UNKNOWN	UNKNOWN	PREASSIGNED PERM SID
BENNIEPORT QEJBSVR 346533 N 000000000000006A	0000000000009690	UNKNOWN	UNKNOWN	MANAGEMENT COLLECTION

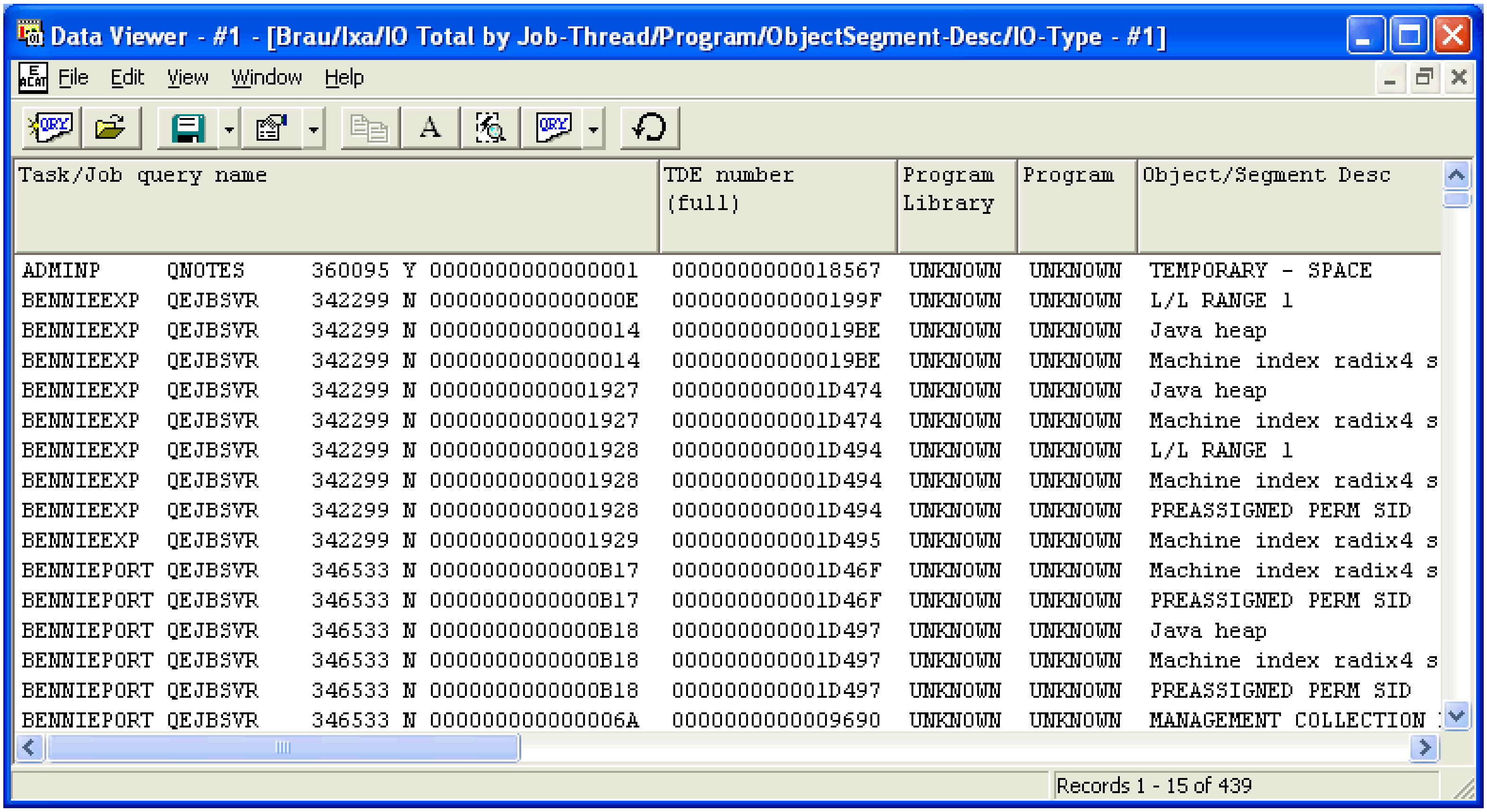
Records 1 - 15 of 393

Table of Contents Previous Next

5.5.27 IO Total by Job-Thread/Program/ObjectSegment-Desc/IO-Type

This report shows a breakdown of the physical disk IO events that occurred for each IO type within object type within program within each job/thread. The object type descriptions are listed in this report.

Note: In order for program names to be provided, *MIENTRY and *MIEXIT events must be included in the PEX collection.


Data Viewer - #1 - [Brau/lxa/IO Total by Job-Thread/Program/ObjectSegment-Desc/IO-Type - #1]
File
Edit
View
Window
Help
QRY
QRY

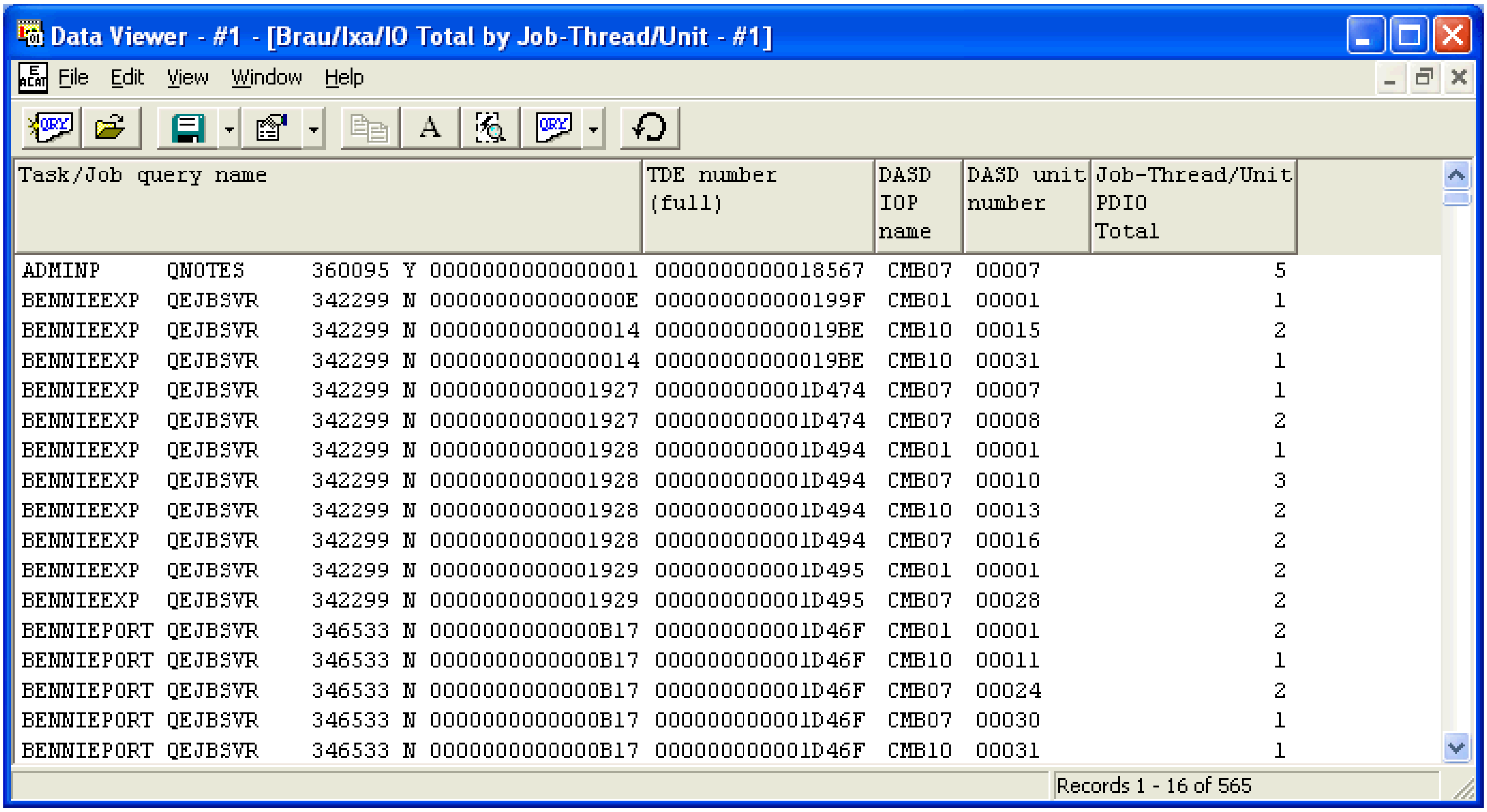
Task/Job query name	TDE number (full)	Program Library	Program	Object/Segment Desc
ADMINP QNOTES 360095 Y 0000000000000001	0000000000018567	UNKNOWN	UNKNOWN	TEMPORARY - SPACE
BENNIEEXP QEJBSVR 342299 N 000000000000000E	000000000000199F	UNKNOWN	UNKNOWN	L/L RANGE 1
BENNIEEXP QEJBSVR 342299 N 0000000000000014	00000000000019BE	UNKNOWN	UNKNOWN	Java heap
BENNIEEXP QEJBSVR 342299 N 0000000000000014	00000000000019BE	UNKNOWN	UNKNOWN	Machine index radix4 s
BENNIEEXP QEJBSVR 342299 N 0000000000001927	000000000001D474	UNKNOWN	UNKNOWN	Java heap
BENNIEEXP QEJBSVR 342299 N 0000000000001927	000000000001D474	UNKNOWN	UNKNOWN	Machine index radix4 s
BENNIEEXP QEJBSVR 342299 N 0000000000001928	000000000001D494	UNKNOWN	UNKNOWN	L/L RANGE 1
BENNIEEXP QEJBSVR 342299 N 0000000000001928	000000000001D494	UNKNOWN	UNKNOWN	Machine index radix4 s
BENNIEEXP QEJBSVR 342299 N 0000000000001928	000000000001D494	UNKNOWN	UNKNOWN	PREASSIGNED PERM SID
BENNIEEXP QEJBSVR 342299 N 0000000000001929	000000000001D495	UNKNOWN	UNKNOWN	Machine index radix4 s
BENNIEPORT QEJBSVR 346533 N 0000000000000B17	000000000001D46F	UNKNOWN	UNKNOWN	Machine index radix4 s
BENNIEPORT QEJBSVR 346533 N 0000000000000B17	000000000001D46F	UNKNOWN	UNKNOWN	PREASSIGNED PERM SID
BENNIEPORT QEJBSVR 346533 N 0000000000000B18	000000000001D497	UNKNOWN	UNKNOWN	Java heap
BENNIEPORT QEJBSVR 346533 N 0000000000000B18	000000000001D497	UNKNOWN	UNKNOWN	Machine index radix4 s
BENNIEPORT QEJBSVR 346533 N 0000000000000B18	000000000001D497	UNKNOWN	UNKNOWN	PREASSIGNED PERM SID
BENNIEPORT QEJBSVR 346533 N 000000000000006A	0000000000009690	UNKNOWN	UNKNOWN	MANAGEMENT COLLECTION

Records 1 - 15 of 439

Table of Contents | Previous | Next

5.5.28 IO Total by Job-Thread/Unit

This report shows the total number of physical disk IO events that occurred for each disk unit within job/thread.


Data Viewer - #1 - [Brau/Ixa/IO Total by Job-Thread/Unit - #1]
File
Edit
View
Window
Help
QRY
QRY

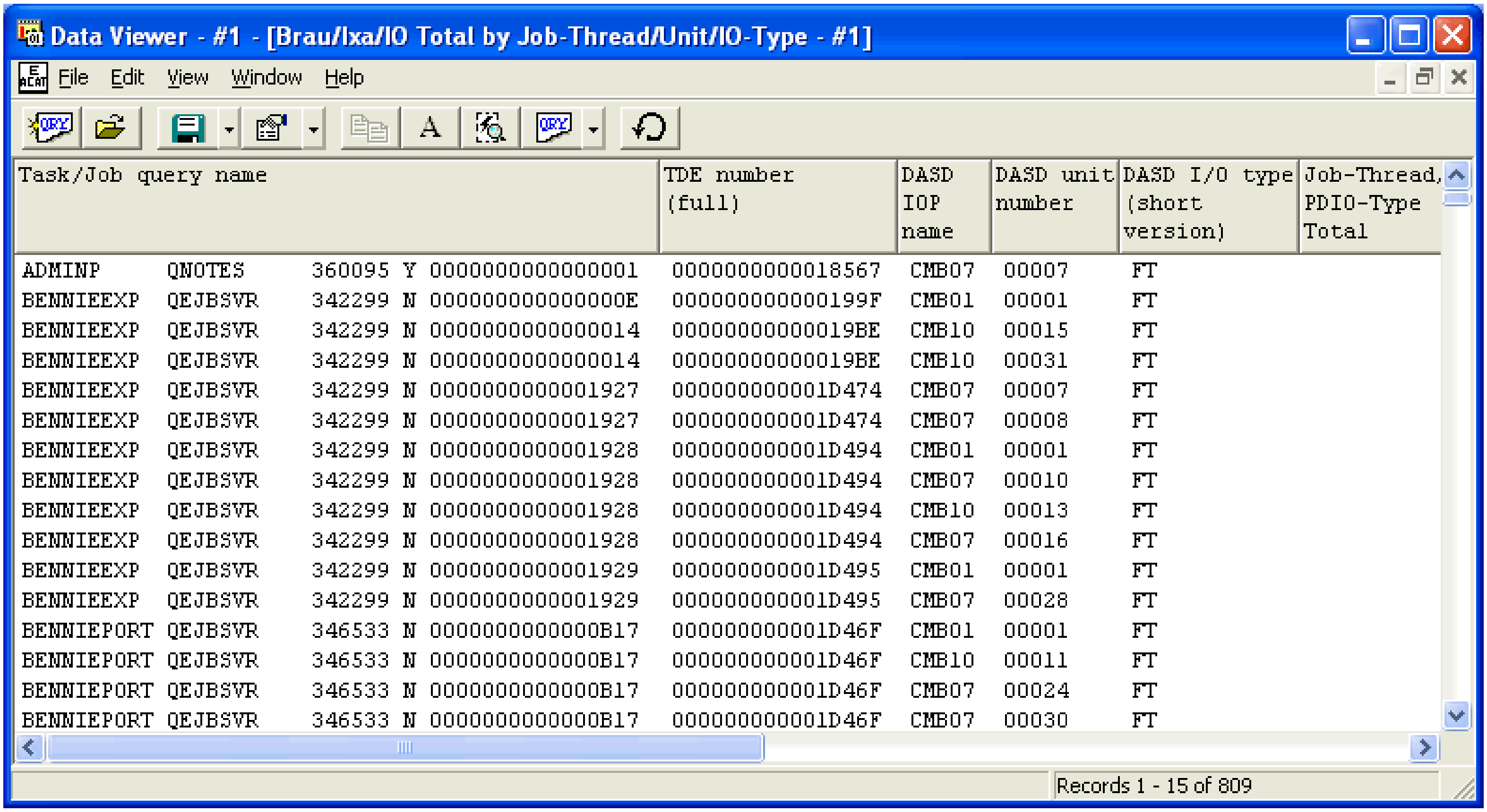
```
Task/Job query name                                    TDE number          DASD    DASD unit  Job-Thread/Unit
                                                       (full)              IOP     number     PDIO
                                                                           name               Total
ADMINP     QNOTES     360095 Y 0000000000000001  0000000000018567  CMB07  00007                     5
BENNIEEXP  QEJBSVR    342299 N 000000000000000E  000000000000199F  CMB01  00001                     1
BENNIEEXP  QEJBSVR    342299 N 0000000000000014  00000000000019BE  CMB10  00015                     2
BENNIEEXP  QEJBSVR    342299 N 0000000000000014  00000000000019BE  CMB10  00031                     1
BENNIEEXP  QEJBSVR    342299 N 0000000000001927  000000000001D474  CMB07  00007                     1
BENNIEEXP  QEJBSVR    342299 N 0000000000001927  000000000001D474  CMB07  00008                     2
BENNIEEXP  QEJBSVR    342299 N 0000000000001928  000000000001D494  CMB01  00001                     1
BENNIEEXP  QEJBSVR    342299 N 0000000000001928  000000000001D494  CMB07  00010                     3
BENNIEEXP  QEJBSVR    342299 N 0000000000001928  000000000001D494  CMB10  00013                     2
BENNIEEXP  QEJBSVR    342299 N 0000000000001928  000000000001D494  CMB07  00016                     2
BENNIEEXP  QEJBSVR    342299 N 0000000000001929  000000000001D495  CMB01  00001                     2
BENNIEEXP  QEJBSVR    342299 N 0000000000001929  000000000001D495  CMB07  00028                     2
BENNIEPORT QEJBSVR    346533 N 0000000000000B17  000000000001D46F  CMB01  00001                     2
BENNIEPORT QEJBSVR    346533 N 0000000000000B17  000000000001D46F  CMB10  00011                     1
BENNIEPORT QEJBSVR    346533 N 0000000000000B17  000000000001D46F  CMB07  00024                     2
BENNIEPORT QEJBSVR    346533 N 0000000000000B17  000000000001D46F  CMB07  00030                     1
BENNIEPORT QEJBSVR    346533 N 0000000000000B17  000000000001D46F  CMB10  00031                     1
```

Records 1 - 16 of 565

Table of Contents | Previous | Next

5.5.29 IO Total by Job-Thread/Unit/IO-Type

This report shows a breakdown of physical disk IO events that occurred for each IO type within DASD unit within job.


Data Viewer - #1 - [Brau/lxa/IO Total by Job-Thread/Unit/IO-Type - #1]
File
Edit
View
Window
Help

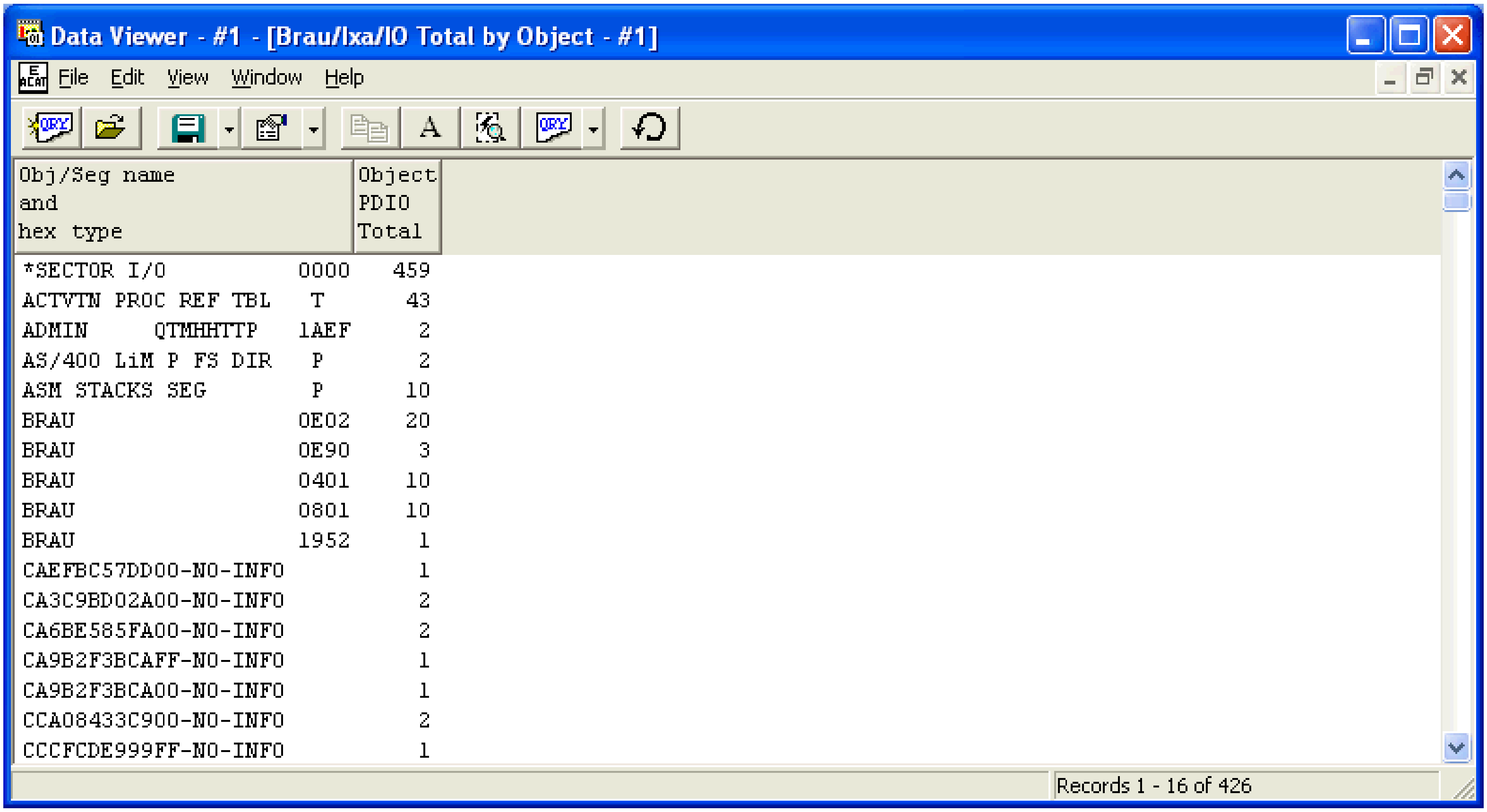
Task/Job query name	TDE number (full)	DASD IOP name	DASD unit number	DASD I/O type (short version)	Job-Thread, PDIO-Type Total
ADMINP QNOTES 360095 Y 0000000000000001	0000000000018567	CMB07	00007	FT	
BENNIEEXP QEJBSVR 342299 N 000000000000000E	000000000000199F	CMB01	00001	FT	
BENNIEEXP QEJBSVR 342299 N 0000000000000014	00000000000019BE	CMB10	00015	FT	
BENNIEEXP QEJBSVR 342299 N 0000000000000014	00000000000019BE	CMB10	00031	FT	
BENNIEEXP QEJBSVR 342299 N 0000000000001927	000000000001D474	CMB07	00007	FT	
BENNIEEXP QEJBSVR 342299 N 0000000000001927	000000000001D474	CMB07	00008	FT	
BENNIEEXP QEJBSVR 342299 N 0000000000001928	000000000001D494	CMB01	00001	FT	
BENNIEEXP QEJBSVR 342299 N 0000000000001928	000000000001D494	CMB07	00010	FT	
BENNIEEXP QEJBSVR 342299 N 0000000000001928	000000000001D494	CMB10	00013	FT	
BENNIEEXP QEJBSVR 342299 N 0000000000001928	000000000001D494	CMB07	00016	FT	
BENNIEEXP QEJBSVR 342299 N 0000000000001929	000000000001D495	CMB01	00001	FT	
BENNIEEXP QEJBSVR 342299 N 0000000000001929	000000000001D495	CMB07	00028	FT	
BENNIEPORT QEJBSVR 346533 N 0000000000000B17	000000000001D46F	CMB01	00001	FT	
BENNIEPORT QEJBSVR 346533 N 0000000000000B17	000000000001D46F	CMB10	00011	FT	
BENNIEPORT QEJBSVR 346533 N 0000000000000B17	000000000001D46F	CMB07	00024	FT	
BENNIEPORT QEJBSVR 346533 N 0000000000000B17	000000000001D46F	CMB07	00030	FT	

Records 1 - 15 of 809

Table of Contents | Previous | Next

5.5.30 IO Total by Object

This report shows the total number of physical disk IO events that occurred for each object within the collection.

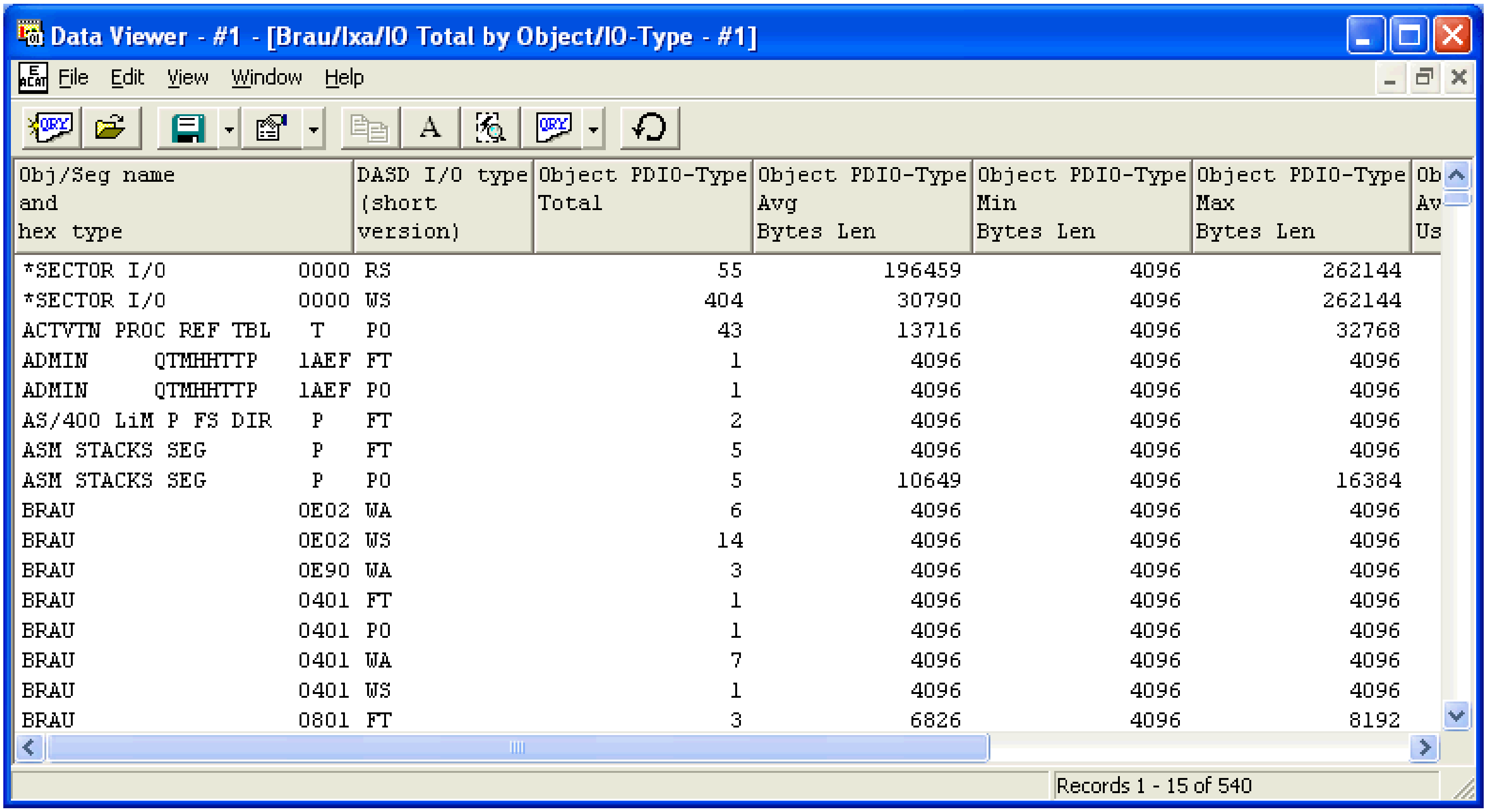


```
Obj/Seg name                 Object
and                          PDIO
hex  type                    Total
*SECTOR I/O           0000    459
ACTVTN PROC REF TBL    T       43
ADMIN     QTMHHTTP    1AEF      2
AS/400 LiM P FS DIR    P        2
ASM STACKS SEG         P       10
BRAU                  0E02     20
BRAU                  0E90      3
BRAU                  0401     10
BRAU                  0801     10
BRAU                  1952      1
CAEFBC57DD00-NO-INFO            1
CA3C9BD02A00-NO-INFO            2
CA6BE585FA00-NO-INFO            2
CA9B2F3BCAFF-NO-INFO            1
CA9B2F3BCA00-NO-INFO            1
CCA08433C900-NO-INFO            2
CCCFCDE999FF-NO-INFO            1
```

Table of Contents | Previous | Next

5.5.31 IO Total by Object/IO-Type

This report shows a breakdown of physical disk IO events that occurred by IO type within object.


Data Viewer - #1 - [Brau/Ixa/IO Total by Object/IO-Type - #1]
File
Edit
View
Window
Help
Records 1 - 15 of 540

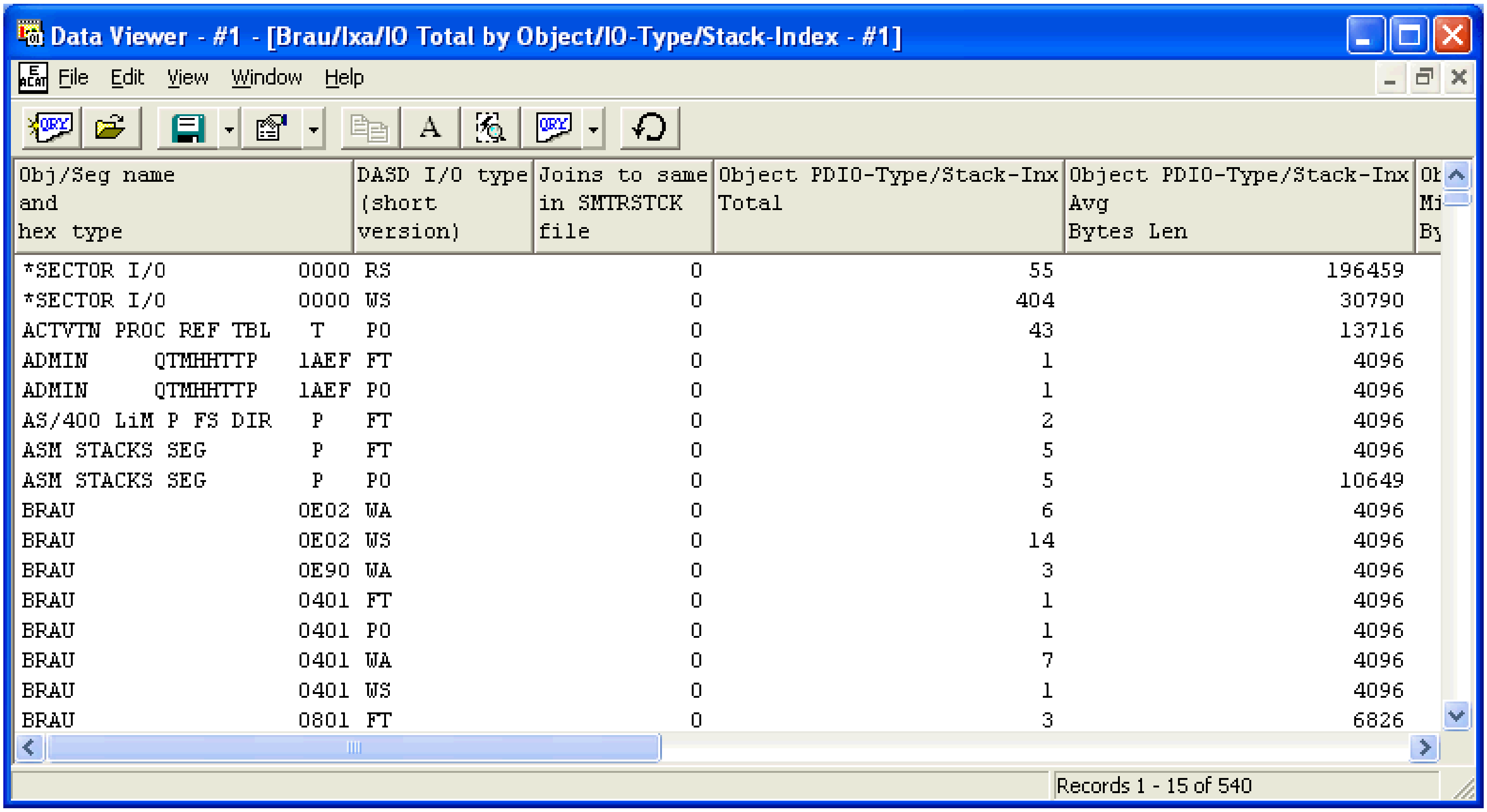
Obj/Seg name and hex type	DASD I/O type (short version)	Object PDIO-Type Total	Object PDIO-Type Avg Bytes Len	Object PDIO-Type Min Bytes Len	Object PDIO-Type Max Bytes Len	Ob Av Us
*SECTOR I/O 0000	RS	55	196459	4096	262144	
*SECTOR I/O 0000	WS	404	30790	4096	262144	
ACTVTN PROC REF TBL T	PO	43	13716	4096	32768	
ADMIN QTMHHTTP 1AEF	FT	1	4096	4096	4096	
ADMIN QTMHHTTP 1AEF	PO	1	4096	4096	4096	
AS/400 LiM P FS DIR P	FT	2	4096	4096	4096	
ASM STACKS SEG P	FT	5	4096	4096	4096	
ASM STACKS SEG P	PO	5	10649	4096	16384	
BRAU 0E02	WA	6	4096	4096	4096	
BRAU 0E02	WS	14	4096	4096	4096	
BRAU 0E90	WA	3	4096	4096	4096	
BRAU 0401	FT	1	4096	4096	4096	
BRAU 0401	PO	1	4096	4096	4096	
BRAU 0401	WA	7	4096	4096	4096	
BRAU 0401	WS	1	4096	4096	4096	
BRAU 0801	FT	3	6826	4096	8192	

Table of Contents | Previous | Next

5.5.32 IO Total by Object/IO-Type/Stack-Index

This report shows a breakdown of physical disk IO events that occurred by IO type within object.

If a call stack is available the STCKINDX field will be greater than 0. Double-clicking on a record will show the call stack if it is available.


Data Viewer - #1 - [Brau/Ixa/IO Total by Object/IO-Type/Stack-Index - #1]
File
Edit
View
Window
Help
QRY
QRY

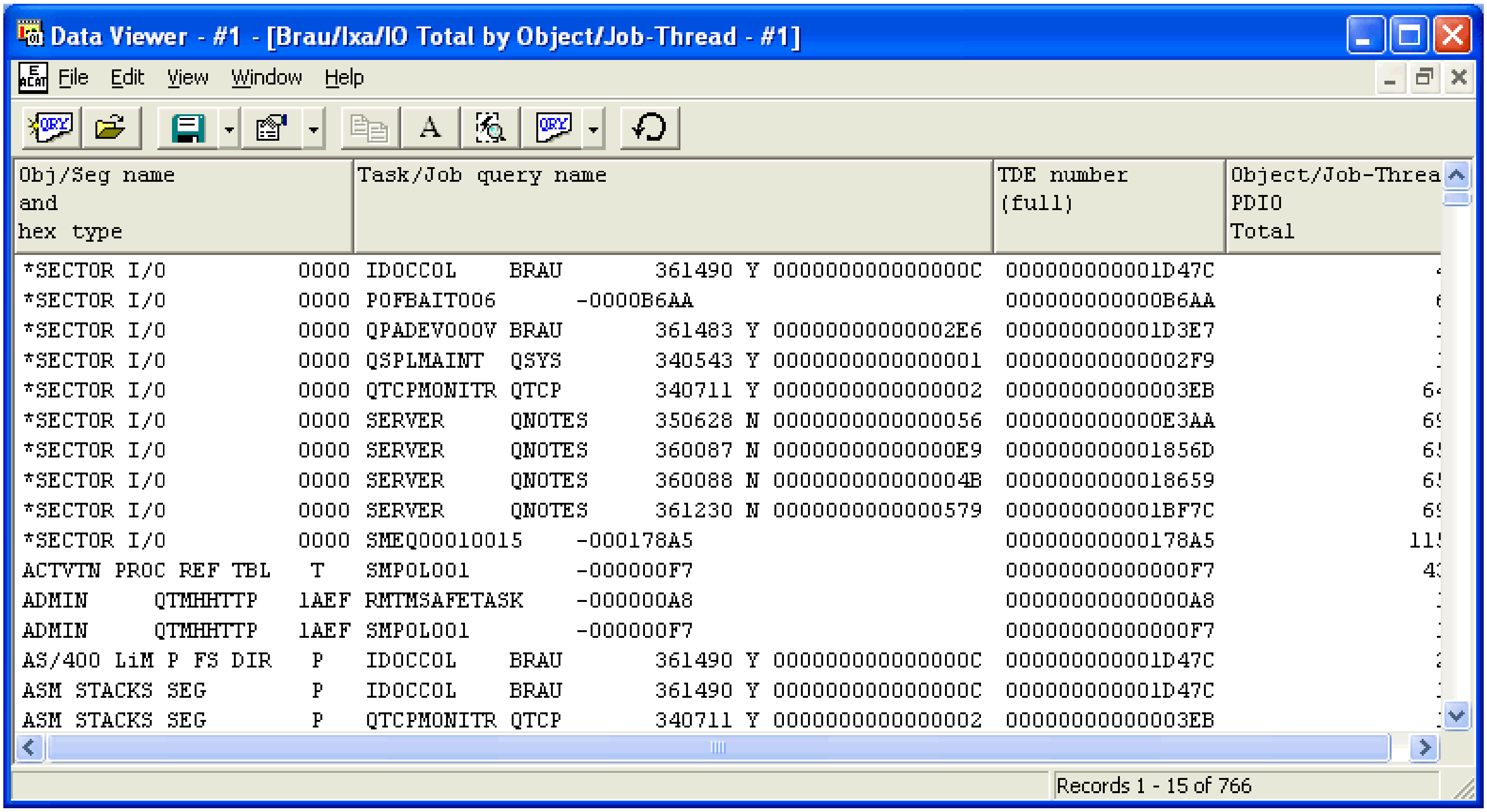
Obj/Seg name and hex type	DASD I/O type (short version)	Joins to same in SMTRSTCK file	Object PDIO-Type/Stack-Inx Total	Object PDIO-Type/Stack-Inx Avg Bytes Len	Ob Mi By
*SECTOR I/O 0000	RS	0	55	196459	
*SECTOR I/O 0000	WS	0	404	30790	
ACTVTN PROC REF TBL T	PO	0	43	13716	
ADMIN QTMHHTTP 1AEF	FT	0	1	4096	
ADMIN QTMHHTTP 1AEF	PO	0	1	4096	
AS/400 LiM P FS DIR P	FT	0	2	4096	
ASM STACKS SEG P	FT	0	5	4096	
ASM STACKS SEG P	PO	0	5	10649	
BRAU 0E02	WA	0	6	4096	
BRAU 0E02	WS	0	14	4096	
BRAU 0E90	WA	0	3	4096	
BRAU 0401	FT	0	1	4096	
BRAU 0401	PO	0	1	4096	
BRAU 0401	WA	0	7	4096	
BRAU 0401	WS	0	1	4096	
BRAU 0801	FT	0	3	6826	

Records 1 - 15 of 540

Table of Contents Previous Next

5.5.33 IO Total by Object/Job-Thread

This report shows the total number of physical disk IO events that occurred by jobs within each object.



Data Viewer - #1 - [Brau/Ixa/IO Total by Object/Job-Thread - #1]
File
Edit
View
Window
Help

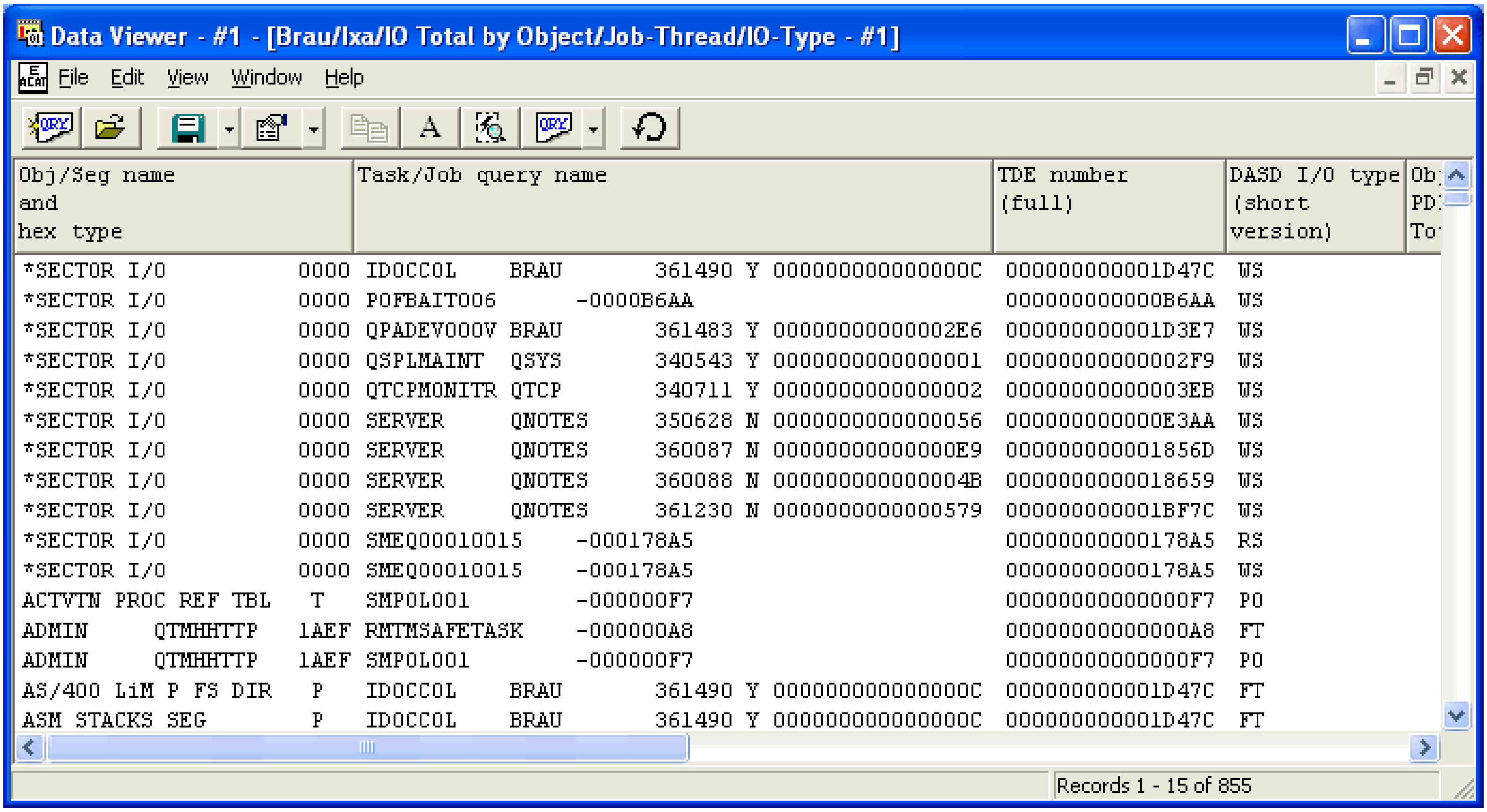
Obj/Seg name and hex type	Task/Job query name	TDE number (full)	Object/Job-Threa PDIO Total
*SECTOR I/O 0000	IDOCCOL BRAU 361490 Y 000000000000000C	000000000001D47C	4
*SECTOR I/O 0000	POFBAIT006 -0000B6AA	00000000000B6AA	6
*SECTOR I/O 0000	QPADEV000V BRAU 361483 Y 00000000000002E6	000000000001D3E7	1
*SECTOR I/O 0000	QSPLMAINT QSYS 340543 Y 0000000000000001	00000000000002F9	1
*SECTOR I/O 0000	QTCPMONITR QTCP 340711 Y 0000000000000002	00000000000003EB	64
*SECTOR I/O 0000	SERVER QNOTES 350628 N 0000000000000056	00000000000E3AA	69
*SECTOR I/O 0000	SERVER QNOTES 360087 N 00000000000000E9	000000000001856D	65
*SECTOR I/O 0000	SERVER QNOTES 360088 N 000000000000004B	0000000000018659	65
*SECTOR I/O 0000	SERVER QNOTES 361230 N 0000000000000579	000000000001BF7C	69
*SECTOR I/O 0000	SMEQ00010015 -000178A5	00000000000178A5	115
ACTVTN PROC REF TBL T	SMPOL001 -000000F7	00000000000000F7	43
ADMIN QTMHHTTP 1AEF	RMTMSAFETASK -000000A8	00000000000000A8	1
ADMIN QTMHHTTP 1AEF	SMPOL001 -000000F7	00000000000000F7	1
AS/400 LiM P FS DIR P	IDOCCOL BRAU 361490 Y 000000000000000C	000000000001D47C	2
ASM STACKS SEG P	IDOCCOL BRAU 361490 Y 000000000000000C	000000000001D47C	1
ASM STACKS SEG P	QTCPMONITR QTCP 340711 Y 0000000000000002	00000000000003EB	1

Records 1 - 15 of 766

Table of Contents | Previous | Next

5.5.34 IO Total by Object/Job-Thread/IO-Type

This report shows a breakdown of physical disk IO events that occurred by for each IO type for jobs within each object.


Data Viewer - #1 - [Brau/Ixa/IO Total by Object/Job-Thread/IO-Type - #1]
File
Edit
View
Window
Help

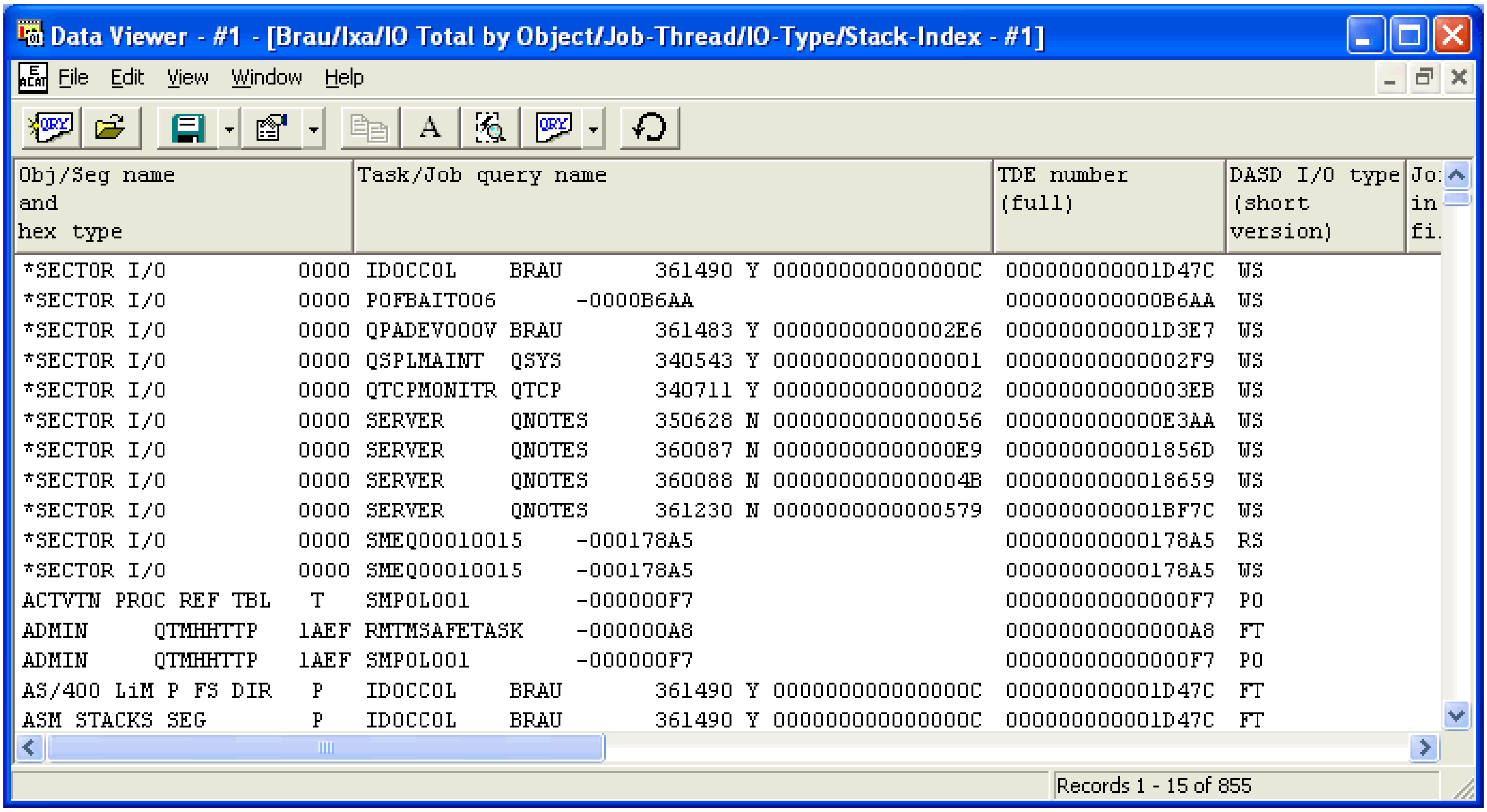
Obj/Seg name and hex type	Task/Job query name	TDE number (full)	DASD I/O type (short version)	Ob: PD: To:
*SECTOR I/O 0000	IDOCCOL BRAU 361490 Y 000000000000000C	000000000001D47C	WS	
*SECTOR I/O 0000	POFBAIT006 -0000B6AA	000000000000B6AA	WS	
*SECTOR I/O 0000	QPADEV000V BRAU 361483 Y 00000000000002E6	000000000001D3E7	WS	
*SECTOR I/O 0000	QSPLMAINT QSYS 340543 Y 0000000000000001	00000000000002F9	WS	
*SECTOR I/O 0000	QTCPMONITR QTCP 340711 Y 0000000000000002	00000000000003EB	WS	
*SECTOR I/O 0000	SERVER QNOTES 350628 N 0000000000000056	000000000000E3AA	WS	
*SECTOR I/O 0000	SERVER QNOTES 360087 N 00000000000000E9	000000000001856D	WS	
*SECTOR I/O 0000	SERVER QNOTES 360088 N 000000000000004B	0000000000018659	WS	
*SECTOR I/O 0000	SERVER QNOTES 361230 N 0000000000000579	000000000001BF7C	WS	
*SECTOR I/O 0000	SMEQ00010015 -000178A5	00000000000178A5	RS	
*SECTOR I/O 0000	SMEQ00010015 -000178A5	00000000000178A5	WS	
ACTVTN PROC REF TBL T	SMPOL001 -000000F7	00000000000000F7	PO	
ADMIN QTMHHTTP 1AEF	RMTMSAFETASK -000000A8	00000000000000A8	FT	
ADMIN QTMHHTTP 1AEF	SMPOL001 -000000F7	00000000000000F7	PO	
AS/400 LiM P FS DIR P	IDOCCOL BRAU 361490 Y 000000000000000C	000000000001D47C	FT	
ASM STACKS SEG P	IDOCCOL BRAU 361490 Y 000000000000000C	000000000001D47C	FT	

Records 1 - 15 of 855

5.5.35 IO Total by Object/Job-Thread/IO-Type/Stack-Index

This report shows a breakdown of physical disk IO events that occurred by for each IO type for jobs within each object.

If a call stack is available the STCKINDX field will be greater than 0. Double-clicking on a record will show the call stack if it is available.


Data Viewer - #1 - [Brau/Ixa/IO Total by Object/Job-Thread/IO-Type/Stack-Index - #1]
File
Edit
View
Window
Help
QRY
A
QRY

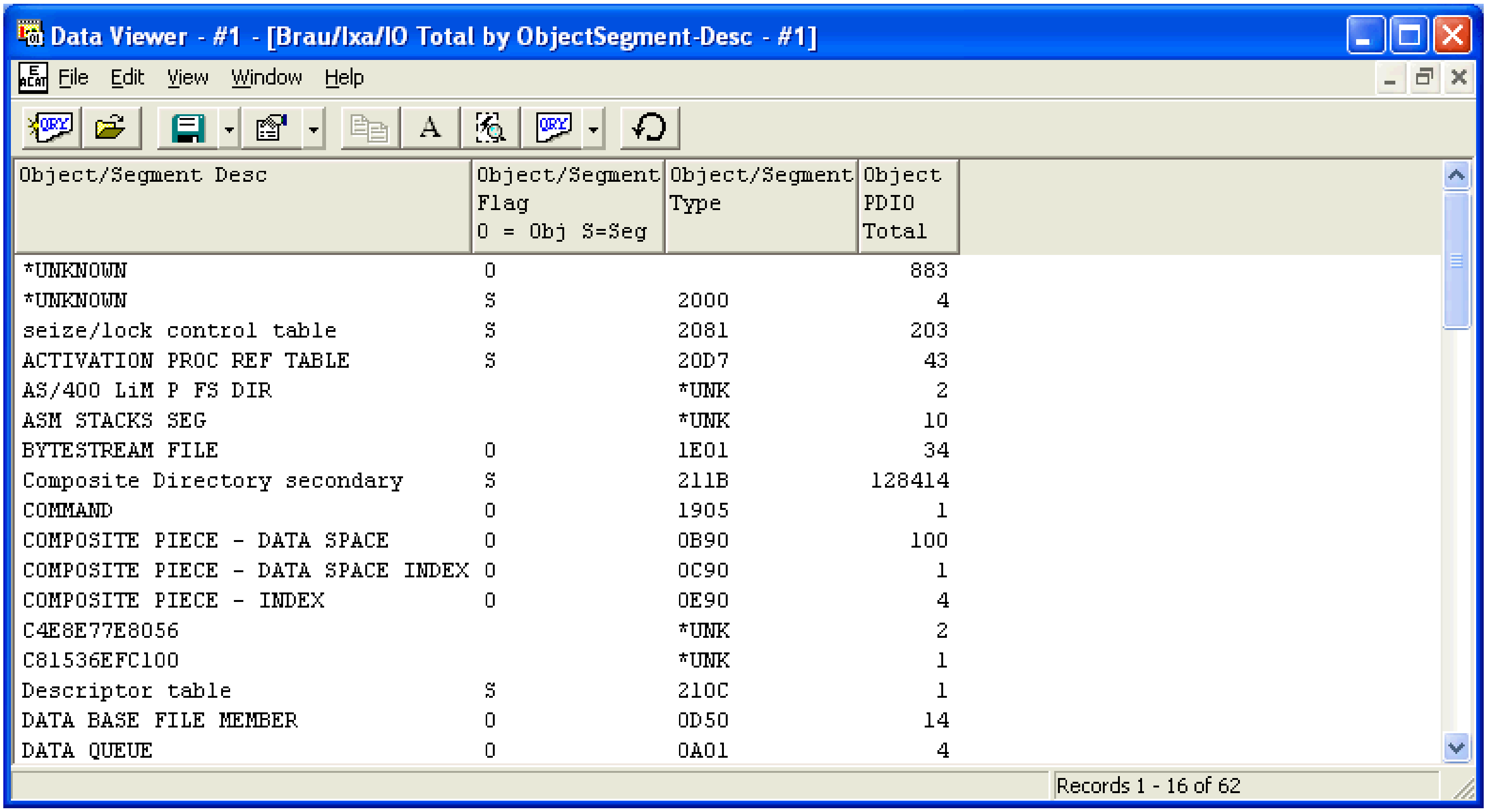
```
Obj/Seg name          Task/Job query name                                   TDE number         DASD I/O type  Jo:
and                                                                         (full)             (short         in
hex type                                                                                       version)       fi.
*SECTOR I/O          0000 IDOCCOL    BRAU       361490 Y 000000000000000C   000000000001D47C   WS
*SECTOR I/O          0000 POFBAIT006      -0000B6AA                         00000000000B6AA    WS
*SECTOR I/O          0000 QPADEV000V BRAU       361483 Y 00000000000002E6   000000000001D3E7   WS
*SECTOR I/O          0000 QSPLMAINT  QSYS       340543 Y 0000000000000001   00000000000002F9   WS
*SECTOR I/O          0000 QTCPMONITR QTCP       340711 Y 0000000000000002   00000000000003EB   WS
*SECTOR I/O          0000 SERVER     QNOTES     350628 N 0000000000000056   00000000000E3AA    WS
*SECTOR I/O          0000 SERVER     QNOTES     360087 N 00000000000000E9   000000000001856D   WS
*SECTOR I/O          0000 SERVER     QNOTES     360088 N 000000000000004B   0000000000018659   WS
*SECTOR I/O          0000 SERVER     QNOTES     361230 N 0000000000000579   000000000001BF7C   WS
*SECTOR I/O          0000 SMEQ00010015    -000178A5                         00000000000178A5   RS
*SECTOR I/O          0000 SMEQ00010015    -000178A5                         00000000000178A5   WS
ACTVTN PROC REF TBL   T   SMPOL001        -000000F7                         00000000000000F7   PO
ADMIN     QTMHHTTP   1AEF RMTMSAFETASK    -000000A8                         00000000000000A8   FT
ADMIN     QTMHHTTP   1AEF SMPOL001        -000000F7                         00000000000000F7   PO
AS/400 LiM P FS DIR   P   IDOCCOL    BRAU       361490 Y 000000000000000C   000000000001D47C   FT
ASM STACKS SEG        P   IDOCCOL    BRAU       361490 Y 000000000000000C   000000000001D47C   FT
```

Records 1 - 15 of 855

Table of Contents | Previous | Next

5.5.36 IO Total by ObjectSegment-Desc

This report shows the total number of physical disk IO events that occurred for each object type.

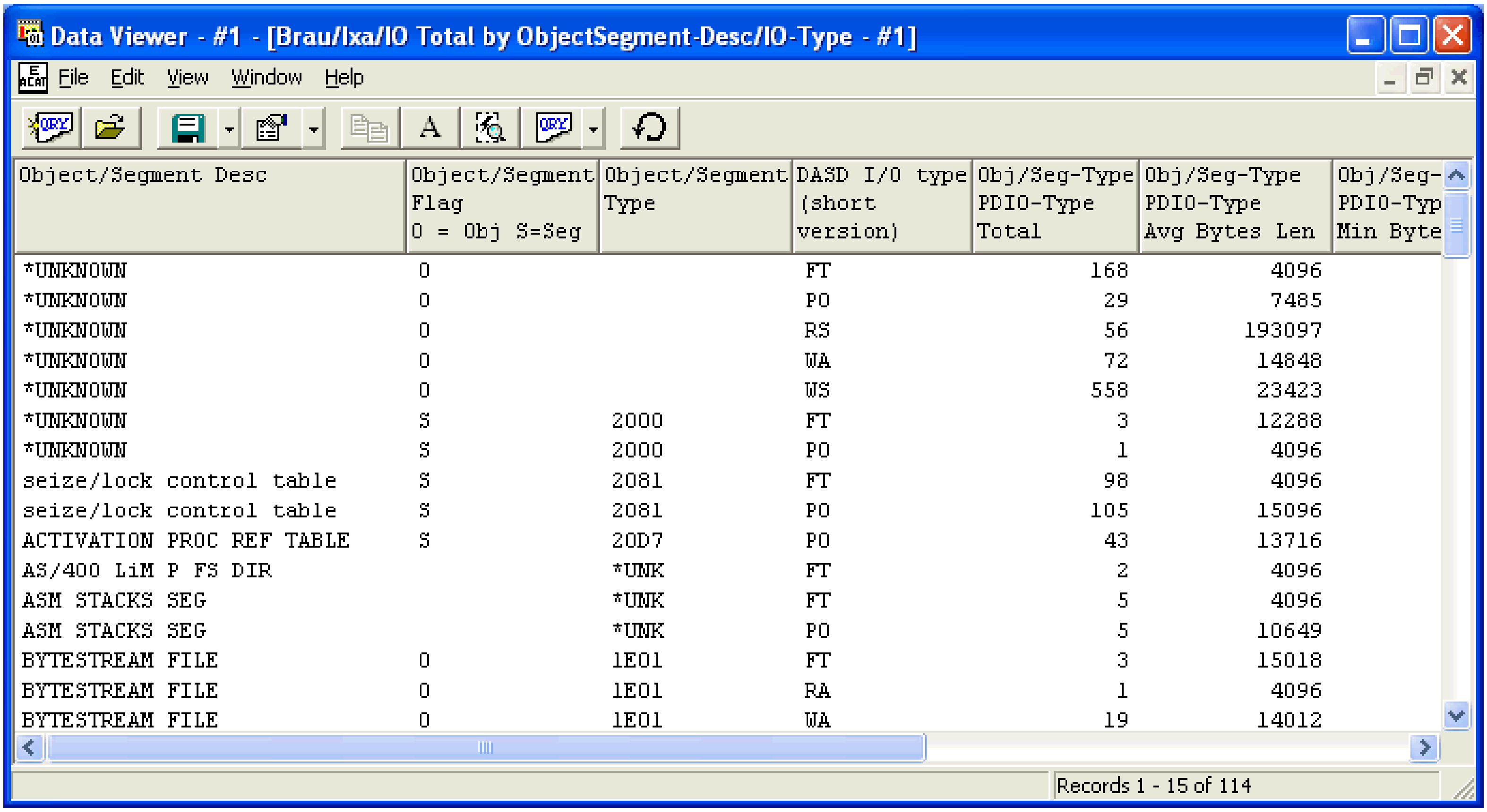

Data Viewer - #1 - [Brau/lxa/IO Total by ObjectSegment-Desc - #1]
File
Edit
View
Window
Help
Records 1 - 16 of 62

Object/Segment Desc	Object/Segment Flag O = Obj S=Seg	Object/Segment Type	Object PDIO Total
*UNKNOWN	O		883
*UNKNOWN	S	2000	4
seize/lock control table	S	2081	203
ACTIVATION PROC REF TABLE	S	20D7	43
AS/400 LiM P FS DIR		*UNK	2
ASM STACKS SEG		*UNK	10
BYTESTREAM FILE	O	1E01	34
Composite Directory secondary	S	211B	128414
COMMAND	O	1905	1
COMPOSITE PIECE - DATA SPACE	O	0B90	100
COMPOSITE PIECE - DATA SPACE INDEX	O	0C90	1
COMPOSITE PIECE - INDEX	O	0E90	4
C4E8E77E8056		*UNK	2
C81536EFC100		*UNK	1
Descriptor table	S	210C	1
DATA BASE FILE MEMBER	O	0D50	14
DATA QUEUE	O	0A01	4

Table of Contents | Previous | Next

5.5.37 IO Total by ObjectSegment-Desc/IO-Type

This report shows a breakdown of physical disk IO events that occurred for each IO type within object type.


Data Viewer - #1 - [Brau/lxa/IO Total by ObjectSegment-Desc/IO-Type - #1]
File
Edit
View
Window
Help
QRY
A
QRY

Object/Segment Desc	Object/Segment Flag 0 = Obj S=Seg	Object/Segment Type	DASD I/O type (short version)	Obj/Seg-Type PDIO-Type Total	Obj/Seg-Type PDIO-Type Avg Bytes Len	Obj/Seg-PDIO-Typ Min Byte
*UNKNOWN	0		FT	168	4096	
*UNKNOWN	0		PO	29	7485	
*UNKNOWN	0		RS	56	193097	
*UNKNOWN	0		WA	72	14848	
*UNKNOWN	0		WS	558	23423	
*UNKNOWN	S	2000	FT	3	12288	
*UNKNOWN	S	2000	PO	1	4096	
seize/lock control table	S	2081	FT	98	4096	
seize/lock control table	S	2081	PO	105	15096	
ACTIVATION PROC REF TABLE	S	20D7	PO	43	13716	
AS/400 LiM P FS DIR		*UNK	FT	2	4096	
ASM STACKS SEG		*UNK	FT	5	4096	
ASM STACKS SEG		*UNK	PO	5	10649	
BYTESTREAM FILE	0	1E01	FT	3	15018	
BYTESTREAM FILE	0	1E01	RA	1	4096	
BYTESTREAM FILE	0	1E01	WA	19	14012	

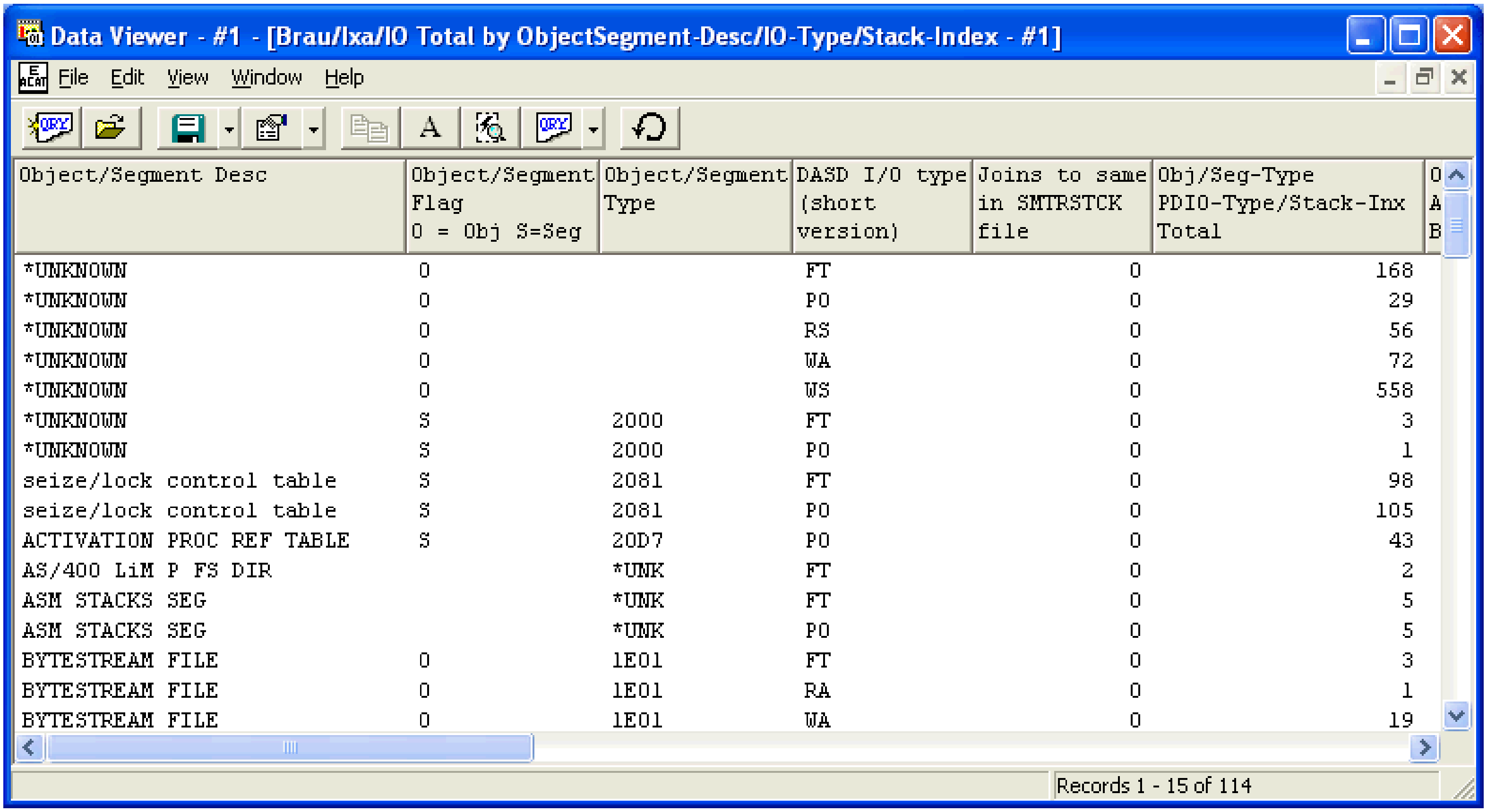
Records 1 - 15 of 114

Table of Contents | Previous | Next

5.5.38 IO Total by ObjectSegment-Desc/IO-Type/Stack-Index

This report shows a breakdown of physical disk IO events that occurred for each IO type within object type.

If a call stack is available the STCKINDX field will be greater than 0. Double-clicking on a record will show the call stack if it is available.


Data Viewer - #1 - [Brau/lxa/IO Total by ObjectSegment-Desc/IO-Type/Stack-Index - #1]
File
Edit
View
Window
Help

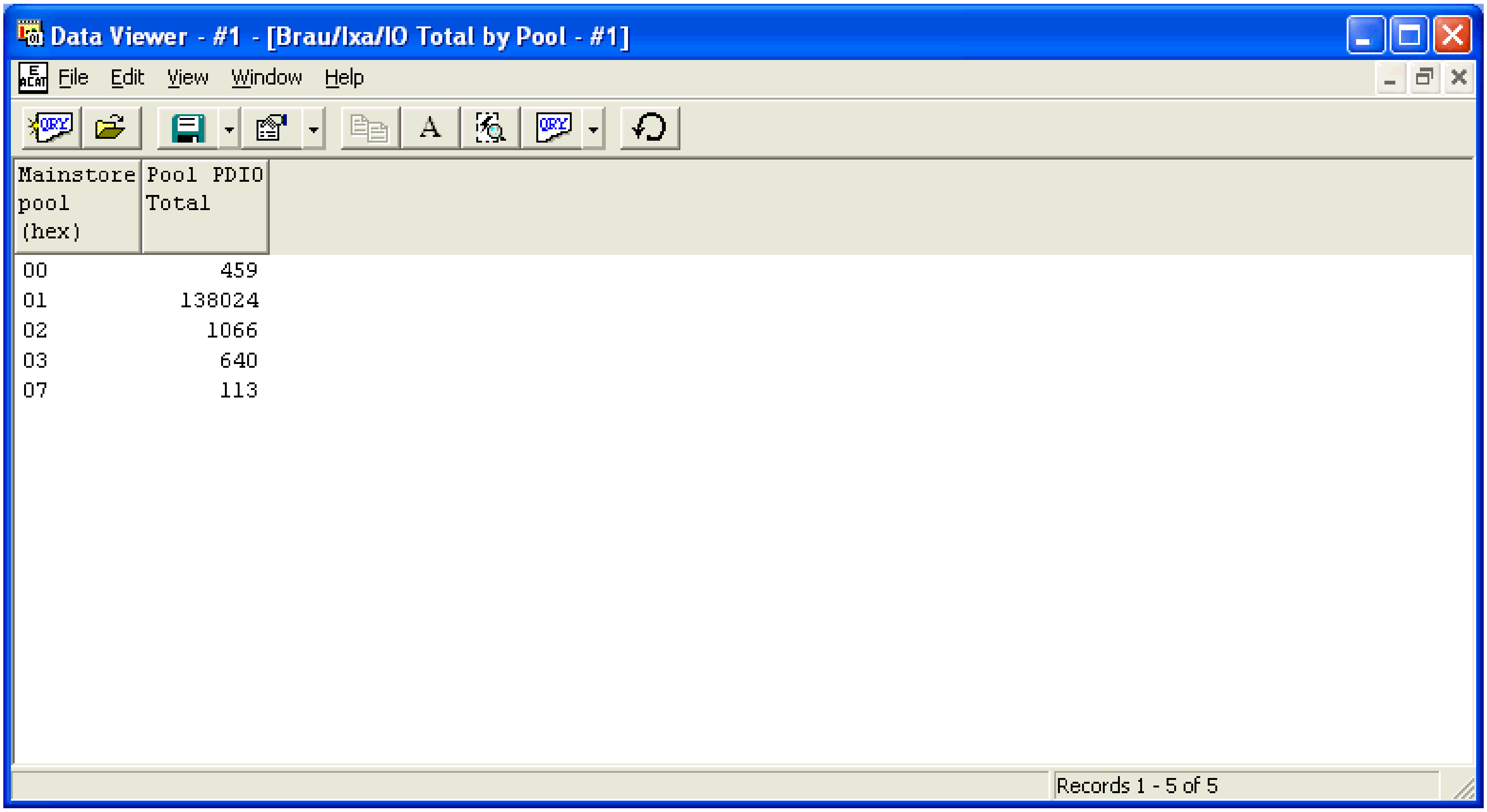
Object/Segment Desc	Object/Segment Flag O = Obj S=Seg	Object/Segment Type	DASD I/O type (short version)	Joins to same in SMTRSTCK file	Obj/Seg-Type PDIO-Type/Stack-Inx Total	O A B
*UNKNOWN	O		FT	0	168	
*UNKNOWN	O		PO	0	29	
*UNKNOWN	O		RS	0	56	
*UNKNOWN	O		WA	0	72	
*UNKNOWN	O		WS	0	558	
*UNKNOWN	S	2000	FT	0	3	
*UNKNOWN	S	2000	PO	0	1	
seize/lock control table	S	2081	FT	0	98	
seize/lock control table	S	2081	PO	0	105	
ACTIVATION PROC REF TABLE	S	20D7	PO	0	43	
AS/400 LiM P FS DIR		*UNK	FT	0	2	
ASM STACKS SEG		*UNK	FT	0	5	
ASM STACKS SEG		*UNK	PO	0	5	
BYTESTREAM FILE	O	1E01	FT	0	3	
BYTESTREAM FILE	O	1E01	RA	0	1	
BYTESTREAM FILE	O	1E01	WA	0	19	

Records 1 - 15 of 114

Table of Contents | Previous | Next

5.5.39 IO Total by Pool

This report shows the total number of physical disk IO events that occurred for each mainstorage pool within the collection. The pool is listed as a hex number.



Data Viewer - #1 - [Brau/Ixa/IO Total by Pool - #1]
File Edit View Window Help
Records 1 - 5 of 5

Mainstore pool (hex)	Pool PDIO Total
00	459
01	138024
02	1066
03	640
07	113

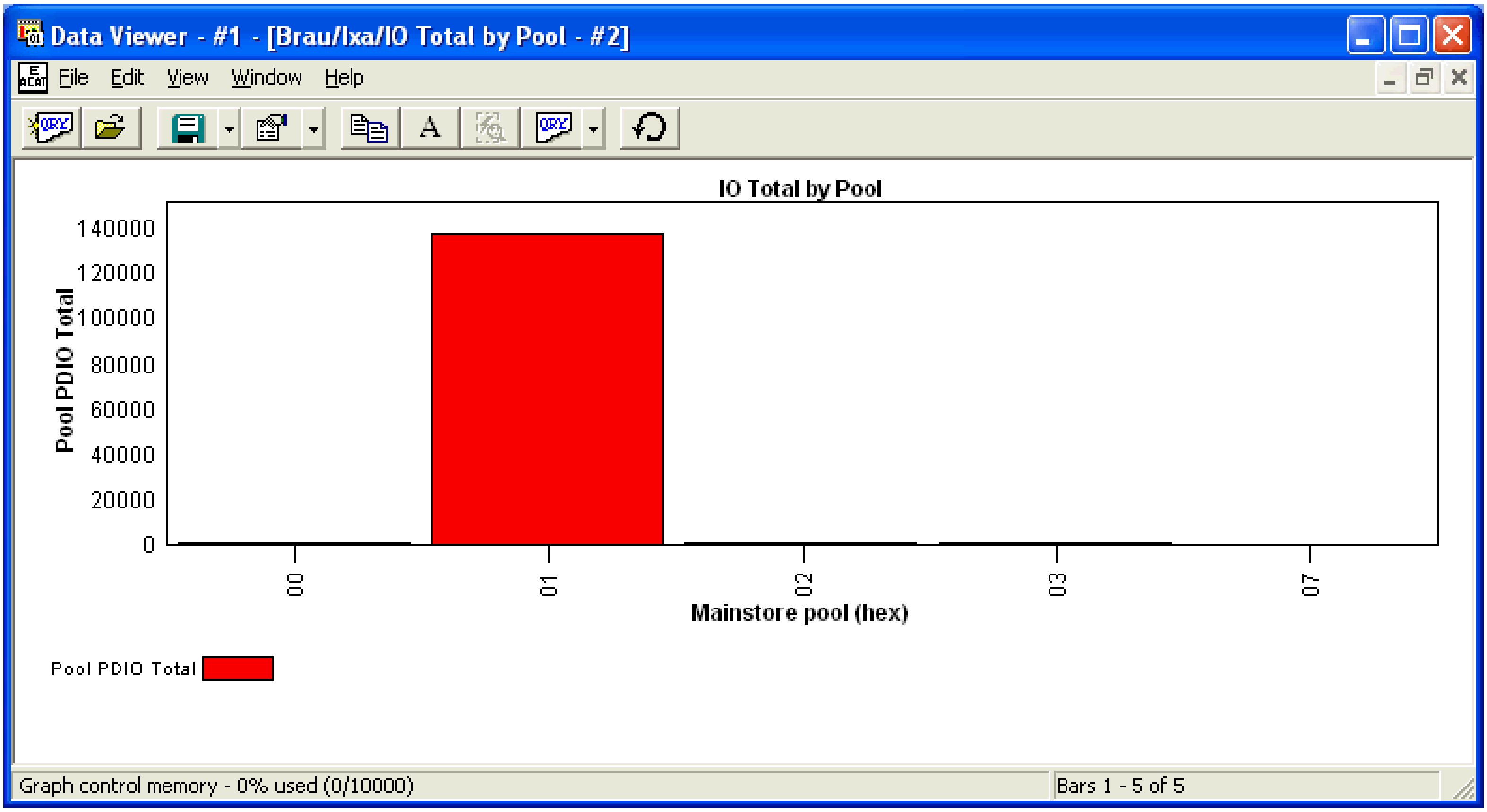
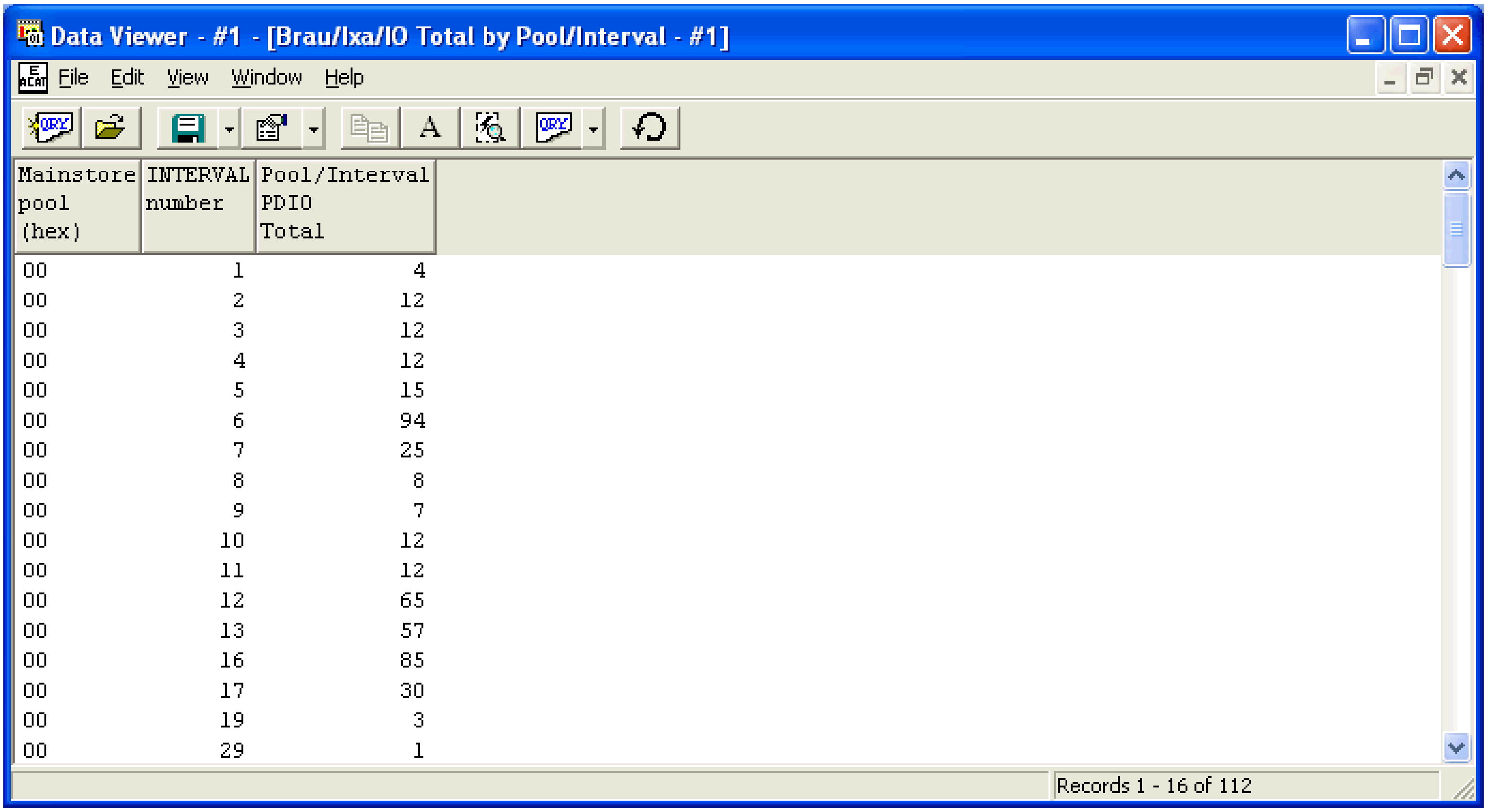

Data Viewer - #1 - [Brau/lxa/IO Total by Pool - #2]
File
Edit
View
Window
Help
IO Total by Pool
Pool PDIO Total
140000
120000
100000
80000
60000
40000
20000
0
00
01
02
03
07
Mainstore pool (hex)
Pool PDIO Total 🟥
Graph control memory - 0% used (0/10000)
Bars 1 - 5 of 5

Table of Contents Previous Next

5.5.40 IO Total by Pool/Interval

This report shows the total number of physical disk IO events that occurred for each interval within mainstorage pools.


Data Viewer - #1 - [Brau/Ixa/IO Total by Pool/Interval - #1]
File Edit View Window Help

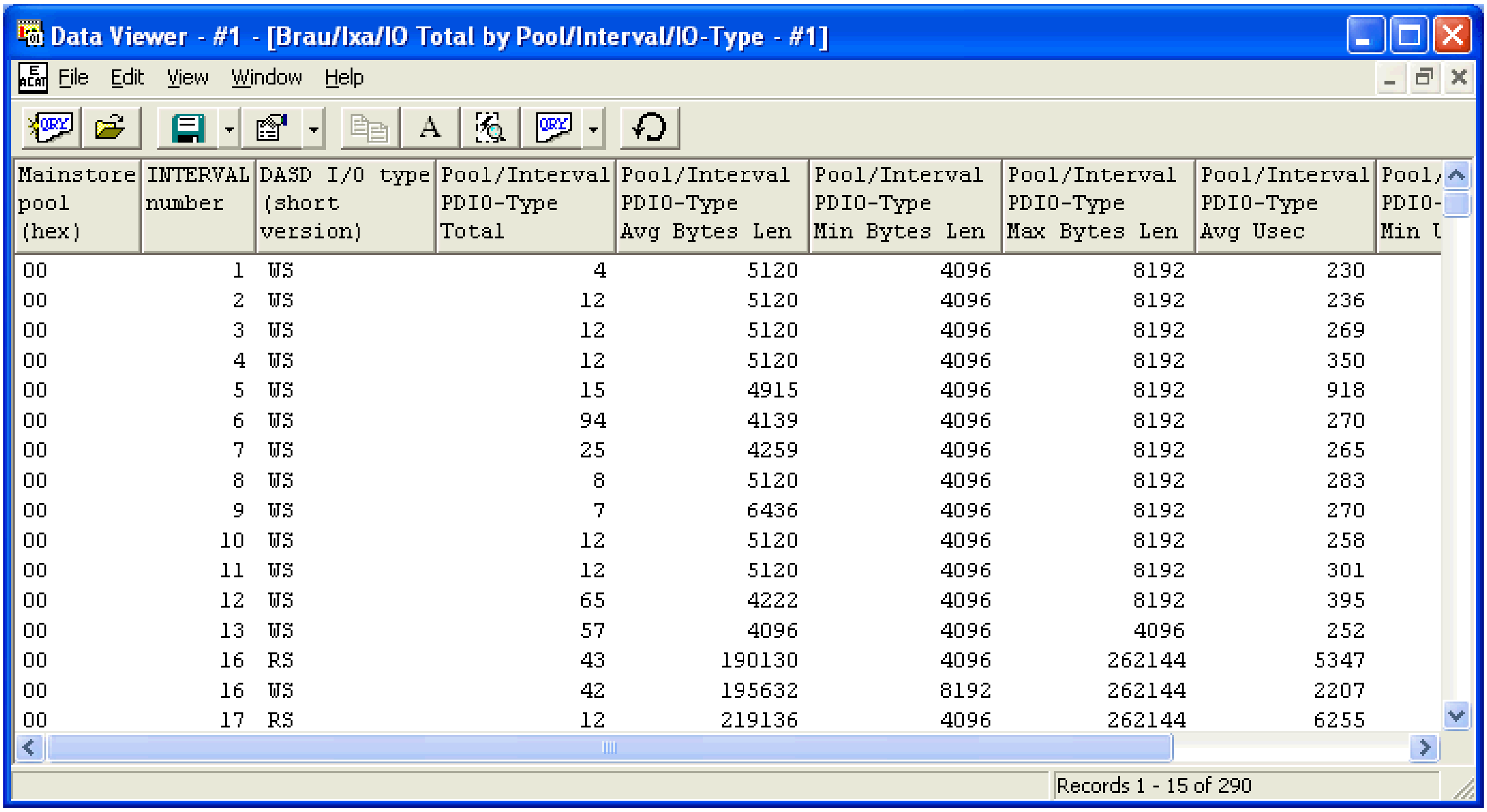
Mainstore pool (hex)	INTERVAL number	Pool/Interval PDIO Total
00	1	4
00	2	12
00	3	12
00	4	12
00	5	15
00	6	94
00	7	25
00	8	8
00	9	7
00	10	12
00	11	12
00	12	65
00	13	57
00	16	85
00	17	30
00	19	3
00	29	1

Records 1 - 16 of 112

Table of Contents | Previous | Next

5.5.41 IO Total by Pool/Interval/IO-Type

This report shows a breakdown of physical disk IO events that occurred for each IO type by interval within mainstorage pools.


Data Viewer - #1 - [Brau/Ixa/IO Total by Pool/Interval/IO-Type - #1]
File
Edit
View
Window
Help

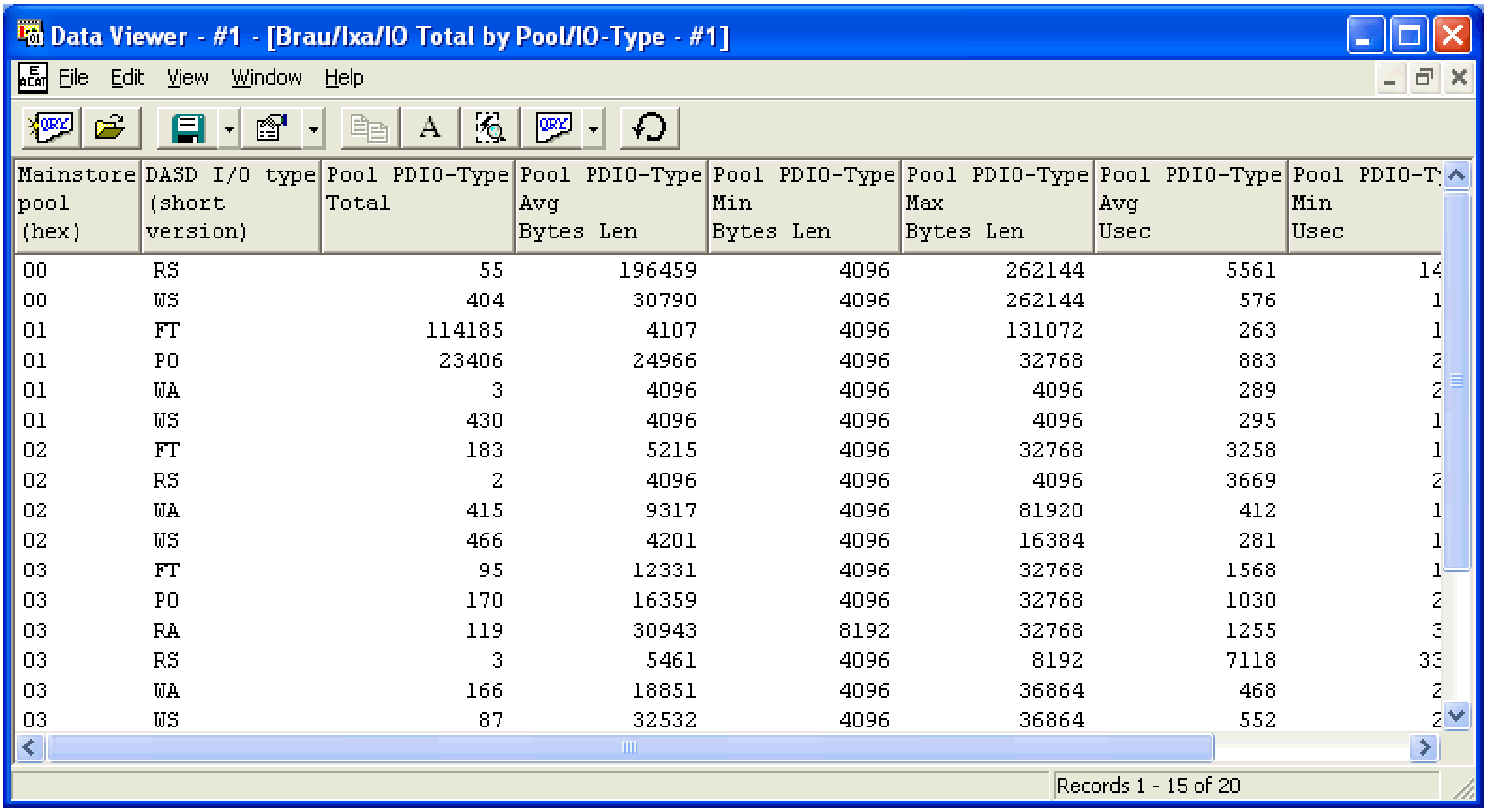
Mainstore pool (hex)	INTERVAL number	DASD I/O type (short version)	Pool/Interval PDIO-Type Total	Pool/Interval PDIO-Type Avg Bytes Len	Pool/Interval PDIO-Type Min Bytes Len	Pool/Interval PDIO-Type Max Bytes Len	Pool/Interval PDIO-Type Avg Usec	Pool/ PDIO- Min U
00	1	WS	4	5120	4096	8192	230	
00	2	WS	12	5120	4096	8192	236	
00	3	WS	12	5120	4096	8192	269	
00	4	WS	12	5120	4096	8192	350	
00	5	WS	15	4915	4096	8192	918	
00	6	WS	94	4139	4096	8192	270	
00	7	WS	25	4259	4096	8192	265	
00	8	WS	8	5120	4096	8192	283	
00	9	WS	7	6436	4096	8192	270	
00	10	WS	12	5120	4096	8192	258	
00	11	WS	12	5120	4096	8192	301	
00	12	WS	65	4222	4096	8192	395	
00	13	WS	57	4096	4096	4096	252	
00	16	RS	43	190130	4096	262144	5347	
00	16	WS	42	195632	8192	262144	2207	
00	17	RS	12	219136	4096	262144	6255	

Records 1 - 15 of 290

Table of Contents | Previous | Next

5.5.42 IO Total by Pool/IO-Type

This report shows a breakdown of physical disk IO events that occurred for each IO type within mainstorage pool. The pool is listed in hex format.



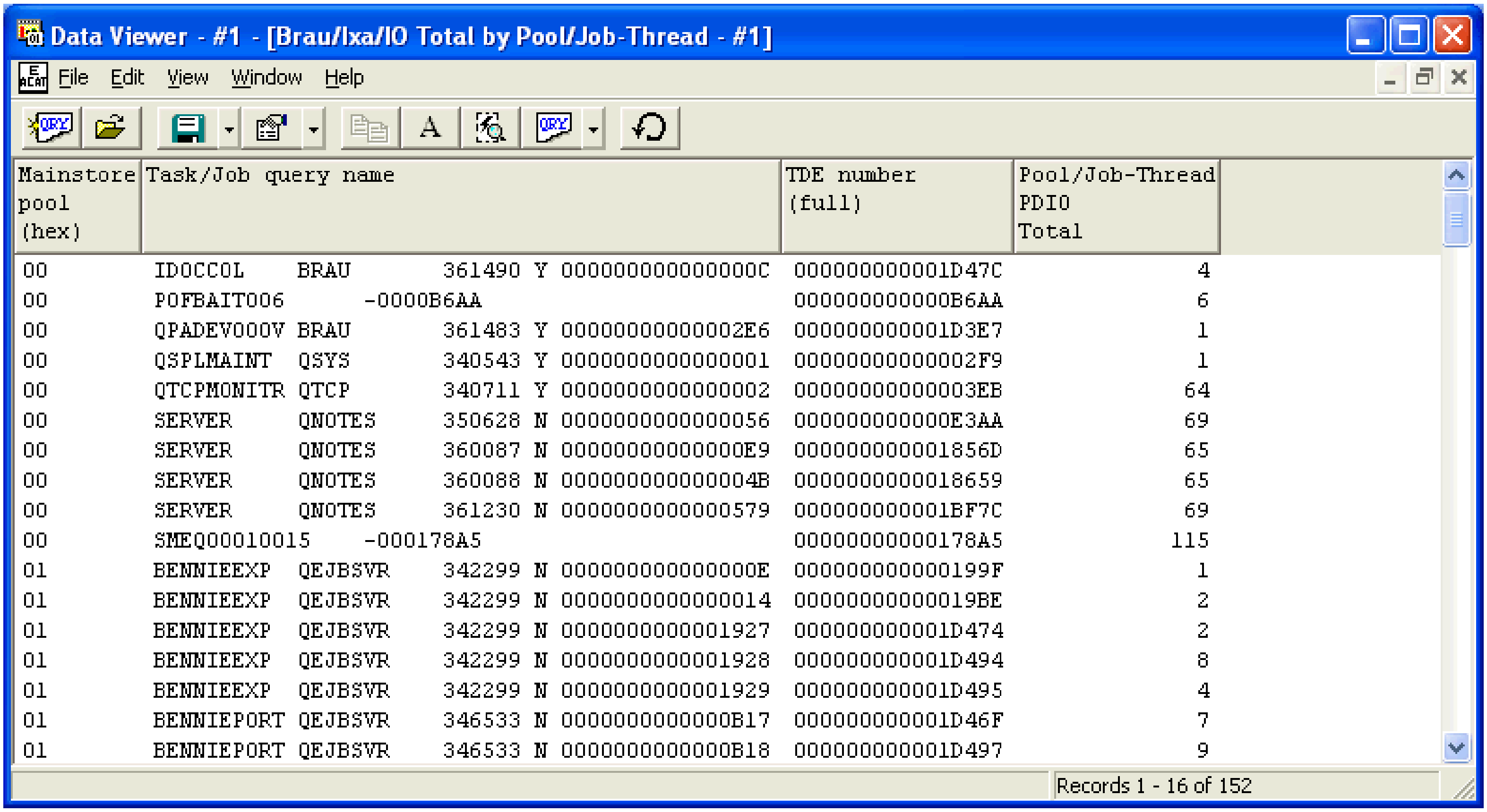
Mainstore pool (hex)	DASD I/O type (short version)	Pool PDIO-Type Total	Pool PDIO-Type Avg Bytes Len	Pool PDIO-Type Min Bytes Len	Pool PDIO-Type Max Bytes Len	Pool PDIO-Type Avg Usec	Pool PDIO-T Min Usec
00	RS	55	196459	4096	262144	5561	14
00	WS	404	30790	4096	262144	576	1
01	FT	114185	4107	4096	131072	263	1
01	PO	23406	24966	4096	32768	883	2
01	WA	3	4096	4096	4096	289	2
01	WS	430	4096	4096	4096	295	1
02	FT	183	5215	4096	32768	3258	1
02	RS	2	4096	4096	4096	3669	2
02	WA	415	9317	4096	81920	412	1
02	WS	466	4201	4096	16384	281	1
03	FT	95	12331	4096	32768	1568	1
03	PO	170	16359	4096	32768	1030	2
03	RA	119	30943	8192	32768	1255	3
03	RS	3	5461	4096	8192	7118	33
03	WA	166	18851	4096	36864	468	2
03	WS	87	32532	4096	36864	552	2

Records 1 - 15 of 20

Table of Contents | Previous | Next

5.5.43 IO Total by Pool/Job-Thread

This report shows the total number of physical disk IO events that occurred for each job within mainstorage pool. The pool id is listed in hex format.


Data Viewer - #1 - [Brau/Ixa/IO Total by Pool/Job-Thread - #1]
File Edit View Window Help
QRY
A
QRY

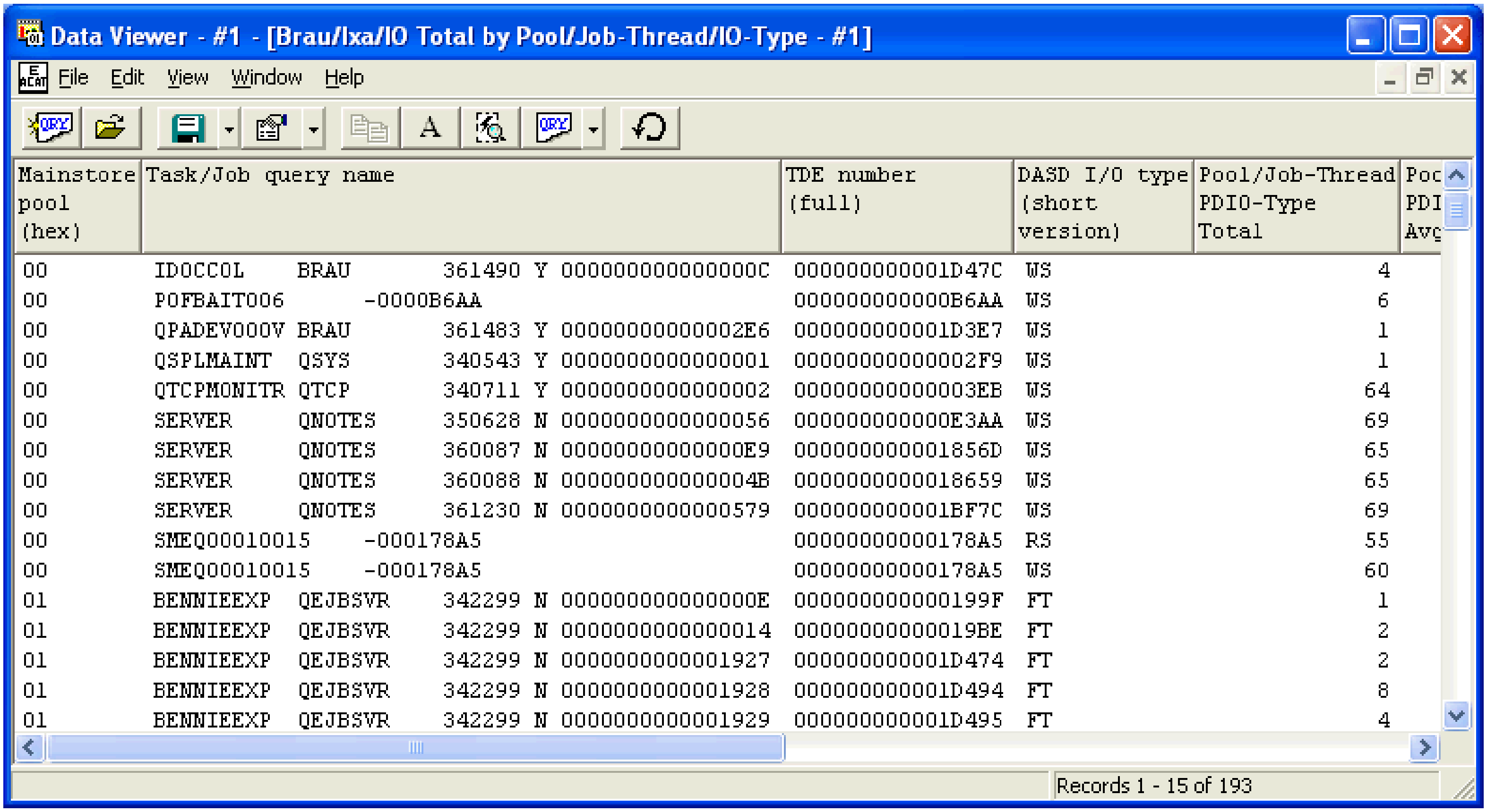
Mainstore pool (hex)	Task/Job query name	TDE number (full)	Pool/Job-Thread PDIO Total
00	IDOCCOL BRAU 361490 Y 000000000000000C	000000000001D47C	4
00	POFBAIT006 -0000B6AA	000000000000B6AA	6
00	QPADEV000V BRAU 361483 Y 00000000000002E6	000000000001D3E7	1
00	QSPLMAINT QSYS 340543 Y 0000000000000001	00000000000002F9	1
00	QTCPMONITR QTCP 340711 Y 0000000000000002	00000000000003EB	64
00	SERVER QNOTES 350628 N 0000000000000056	000000000000E3AA	69
00	SERVER QNOTES 360087 N 00000000000000E9	000000000001856D	65
00	SERVER QNOTES 360088 N 000000000000004B	0000000000018659	65
00	SERVER QNOTES 361230 N 0000000000000579	000000000001BF7C	69
00	SMEQ00010015 -000178A5	00000000000178A5	115
01	BENNIEEXP QEJBSVR 342299 N 000000000000000E	000000000000199F	1
01	BENNIEEXP QEJBSVR 342299 N 0000000000000014	00000000000019BE	2
01	BENNIEEXP QEJBSVR 342299 N 0000000000001927	000000000001D474	2
01	BENNIEEXP QEJBSVR 342299 N 0000000000001928	000000000001D494	8
01	BENNIEEXP QEJBSVR 342299 N 0000000000001929	000000000001D495	4
01	BENNIEPORT QEJBSVR 346533 N 0000000000000B17	000000000001D46F	7
01	BENNIEPORT QEJBSVR 346533 N 0000000000000B18	000000000001D497	9

Records 1 - 16 of 152

Table of Contents Previous Next

5.5.44 IO Total by Pool/Job-Thread/IO-Type

This report shows a breakdown of physical disk IO events that occurred for each IO type by job within mainstorage pool. Pool id is listed in hex format.


Data Viewer - #1 - [Brau/Ixa/IO Total by Pool/Job-Thread/IO-Type - #1]
File
Edit
View
Window
Help

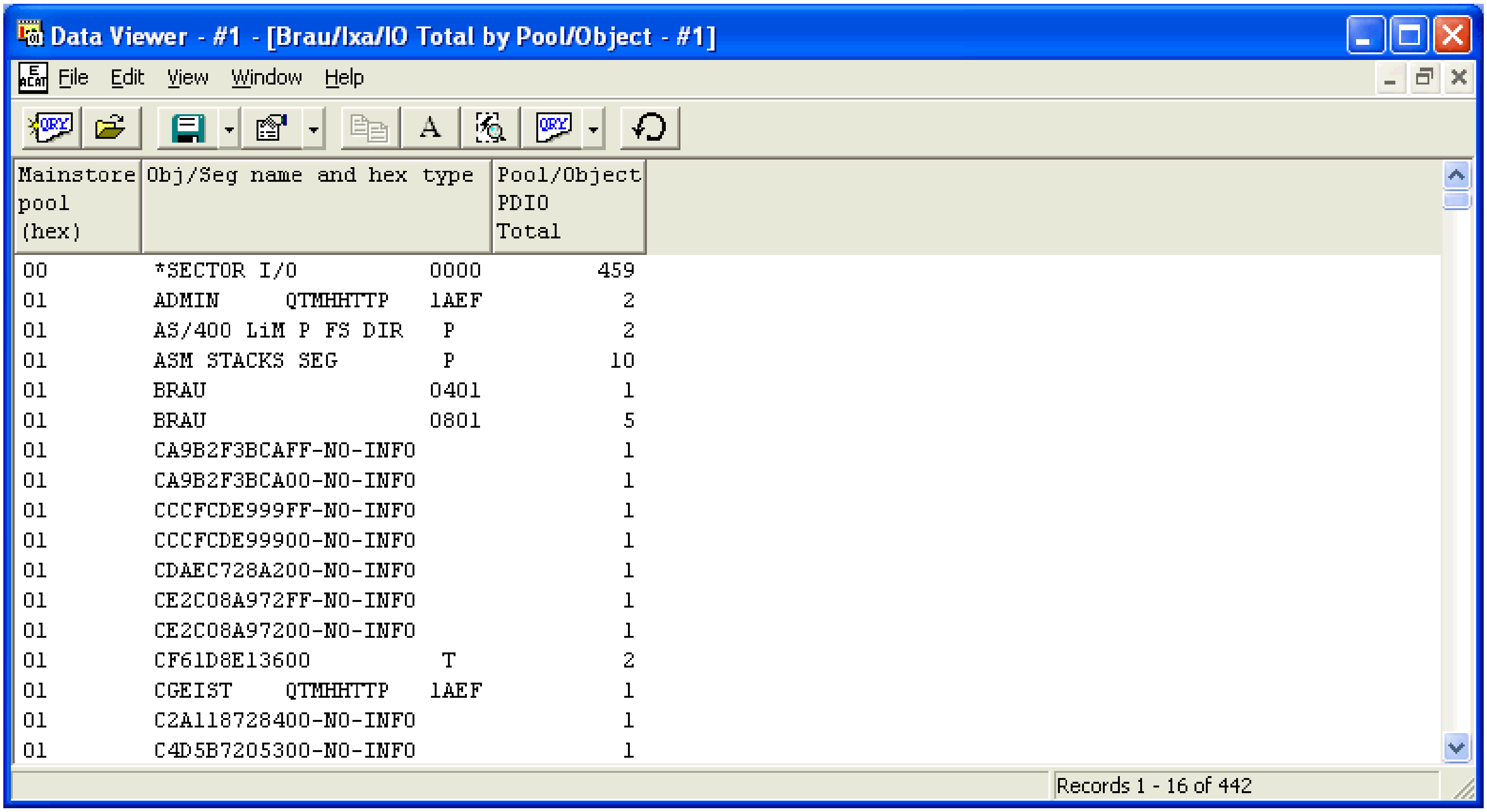
Mainstore pool (hex)	Task/Job query name	TDE number (full)	DASD I/O type (short version)	Pool/Job-Thread PDIO-Type Total	Poc PDI Avg
00	IDOCCOL BRAU 361490 Y 000000000000000C	000000000001D47C	WS	4	
00	POFBAIT006 -0000B6AA	00000000000B6AA	WS	6	
00	QPADEV000V BRAU 361483 Y 00000000000002E6	000000000001D3E7	WS	1	
00	QSPLMAINT QSYS 340543 Y 0000000000000001	00000000000002F9	WS	1	
00	QTCPMONITR QTCP 340711 Y 0000000000000002	00000000000003EB	WS	64	
00	SERVER QNOTES 350628 N 0000000000000056	000000000000E3AA	WS	69	
00	SERVER QNOTES 360087 N 00000000000000E9	000000000001856D	WS	65	
00	SERVER QNOTES 360088 N 000000000000004B	0000000000018659	WS	65	
00	SERVER QNOTES 361230 N 0000000000000579	000000000001BF7C	WS	69	
00	SMEQ00010015 -000178A5	00000000000178A5	RS	55	
00	SMEQ00010015 -000178A5	00000000000178A5	WS	60	
01	BENNIEEXP QEJBSVR 342299 N 000000000000000E	000000000000199F	FT	1	
01	BENNIEEXP QEJBSVR 342299 N 0000000000000014	00000000000019BE	FT	2	
01	BENNIEEXP QEJBSVR 342299 N 0000000000001927	000000000001D474	FT	2	
01	BENNIEEXP QEJBSVR 342299 N 0000000000001928	000000000001D494	FT	8	
01	BENNIEEXP QEJBSVR 342299 N 0000000000001929	000000000001D495	FT	4	

Records 1 - 15 of 193

Table of Contents | Previous | Next

5.5.45 IO Total by Pool/Object

This report shows the total number of physical disk IO events that occurred by object within mainstorage pool



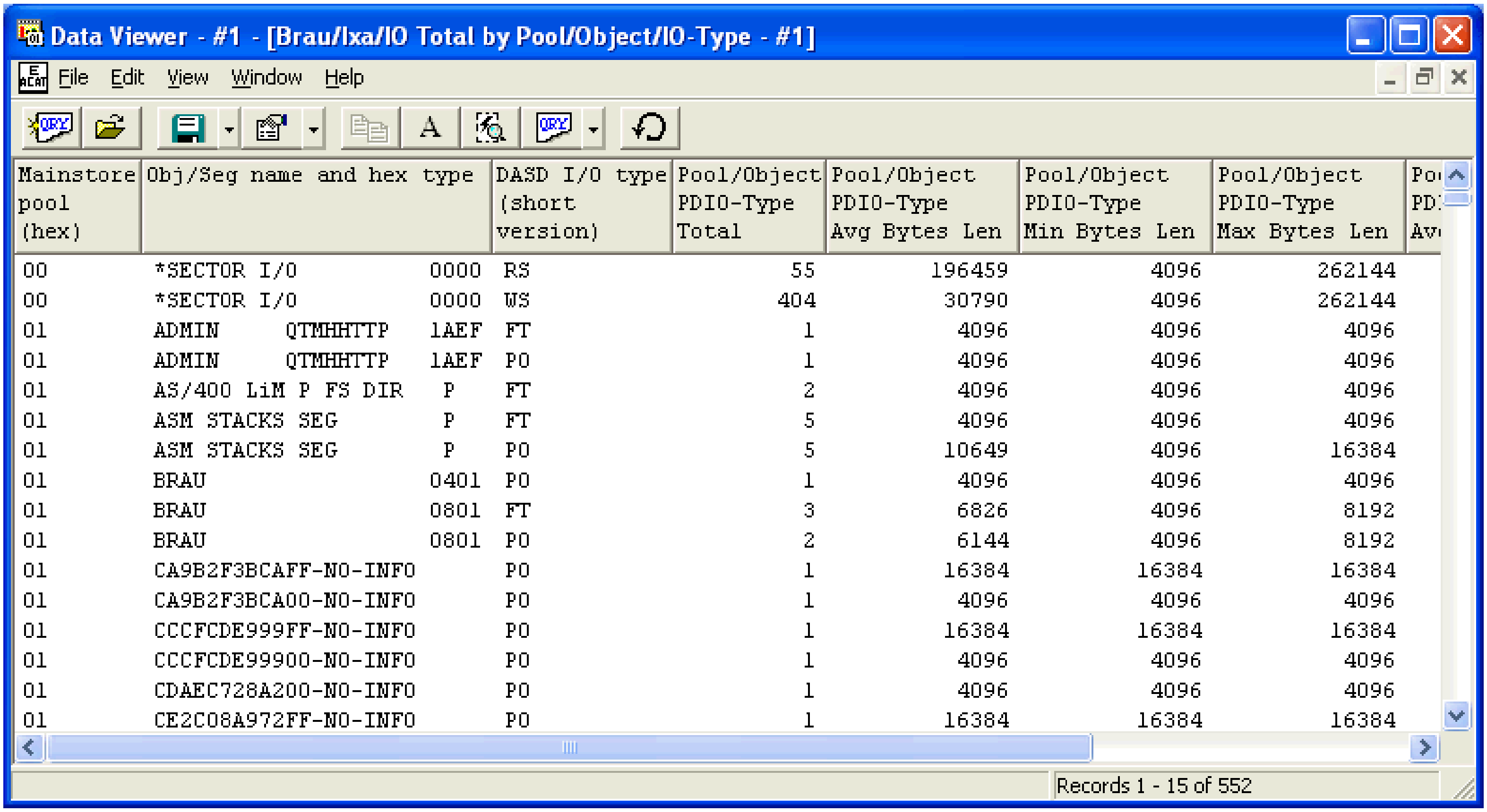
Mainstore pool (hex)	Obj/Seg name and hex type	Pool/Object PDIO Total
00	*SECTOR I/O 0000	459
01	ADMIN QTMHHTTP 1AEF	2
01	AS/400 LiM P FS DIR P	2
01	ASM STACKS SEG P	10
01	BRAU 0401	1
01	BRAU 0801	5
01	CA9B2F3BCAFF-NO-INFO	1
01	CA9B2F3BCA00-NO-INFO	1
01	CCCFCDE999FF-NO-INFO	1
01	CCCFCDE99900-NO-INFO	1
01	CDAEC728A200-NO-INFO	1
01	CE2C08A972FF-NO-INFO	1
01	CE2C08A97200-NO-INFO	1
01	CF61D8E13600 T	2
01	CGEIST QTMHHTTP 1AEF	1
01	C2A118728400-NO-INFO	1
01	C4D5B7205300-NO-INFO	1

Records 1 - 16 of 442

Table of Contents | Previous | Next

5.5.46 IO Total by Pool/Object/IO-Type

This report shows a breakdown of physical disk IO events that occurred by IO type within object within mainstorage pool. Pool id is listed in hex format.


Data Viewer - #1 - [Brau/lxa/IO Total by Pool/Object/IO-Type - #1]
File
Edit
View
Window
Help

Mainstore pool (hex)	Obj/Seg name and hex type	DASD I/O type (short version)	Pool/Object PDIO-Type Total	Pool/Object PDIO-Type Avg Bytes Len	Pool/Object PDIO-Type Min Bytes Len	Pool/Object PDIO-Type Max Bytes Len	Po PD Av
00	*SECTOR I/O 0000	RS	55	196459	4096	262144	
00	*SECTOR I/O 0000	WS	404	30790	4096	262144	
01	ADMIN QTMHHTTP 1AEF	FT	1	4096	4096	4096	
01	ADMIN QTMHHTTP 1AEF	PO	1	4096	4096	4096	
01	AS/400 LiM P FS DIR P	FT	2	4096	4096	4096	
01	ASM STACKS SEG P	FT	5	4096	4096	4096	
01	ASM STACKS SEG P	PO	5	10649	4096	16384	
01	BRAU 0401	PO	1	4096	4096	4096	
01	BRAU 0801	FT	3	6826	4096	8192	
01	BRAU 0801	PO	2	6144	4096	8192	
01	CA9B2F3BCAFF-NO-INFO	PO	1	16384	16384	16384	
01	CA9B2F3BCA00-NO-INFO	PO	1	4096	4096	4096	
01	CCCFCDE999FF-NO-INFO	PO	1	16384	16384	16384	
01	CCCFCDE99900-NO-INFO	PO	1	4096	4096	4096	
01	CDAEC728A200-NO-INFO	PO	1	4096	4096	4096	
01	CE2C08A972FF-NO-INFO	PO	1	16384	16384	16384	

Records 1 - 15 of 552

Table of Contents | Previous | Next

5.5.47 IO Total by Program

This report shows the total number of physical disk IO events that occurred for each program in the collection.

Note: In order for program names to be provided, *MIENTRY and *MIEXIT events must be included in the PEX collection.

Table of Contents Previous Next

5.5.48 IO Total by Program/IO-Type

This report shows a breakdown of physical disk IO events that occurred for each IO type within program in the collection.

Note: In order for program names to be provided, *MIENTRY and *MIEXIT events must be included in the PEX collection.

Table of Contents Previous Next

5.5.49 IO Total by Program/Object

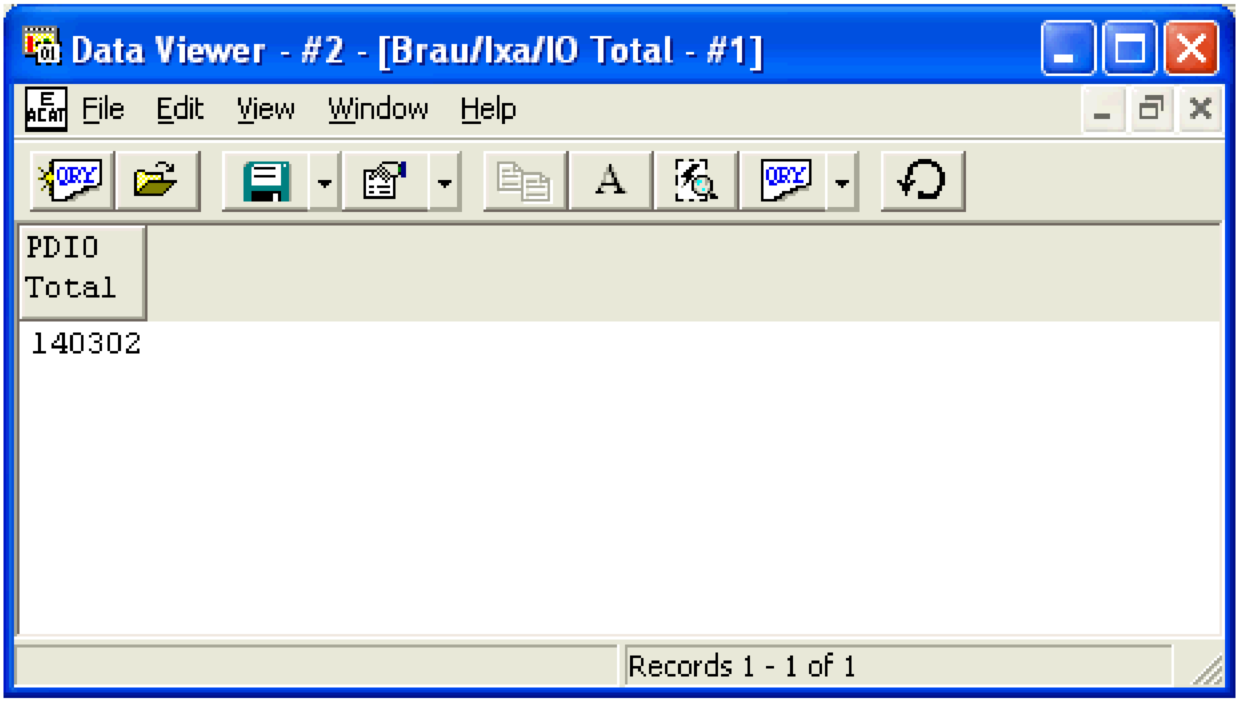
This report shows the total number of physical disk IO events that occurred for each IO type within object within program in the collection.

Note: In order for program names to be provided, *MIENTRY and *MIEXIT events must be included in the PEX collection.

Table of Contents Previous Next

5.5.50 IO Total by Program/Object/IO-Type

This report shows the total number of physical disk IO events that occurred over the data analyzed.


Data Viewer - #2 - [Brau/Ixa/IO Total - #1]
File Edit View Window Help
QRY
A
QRY

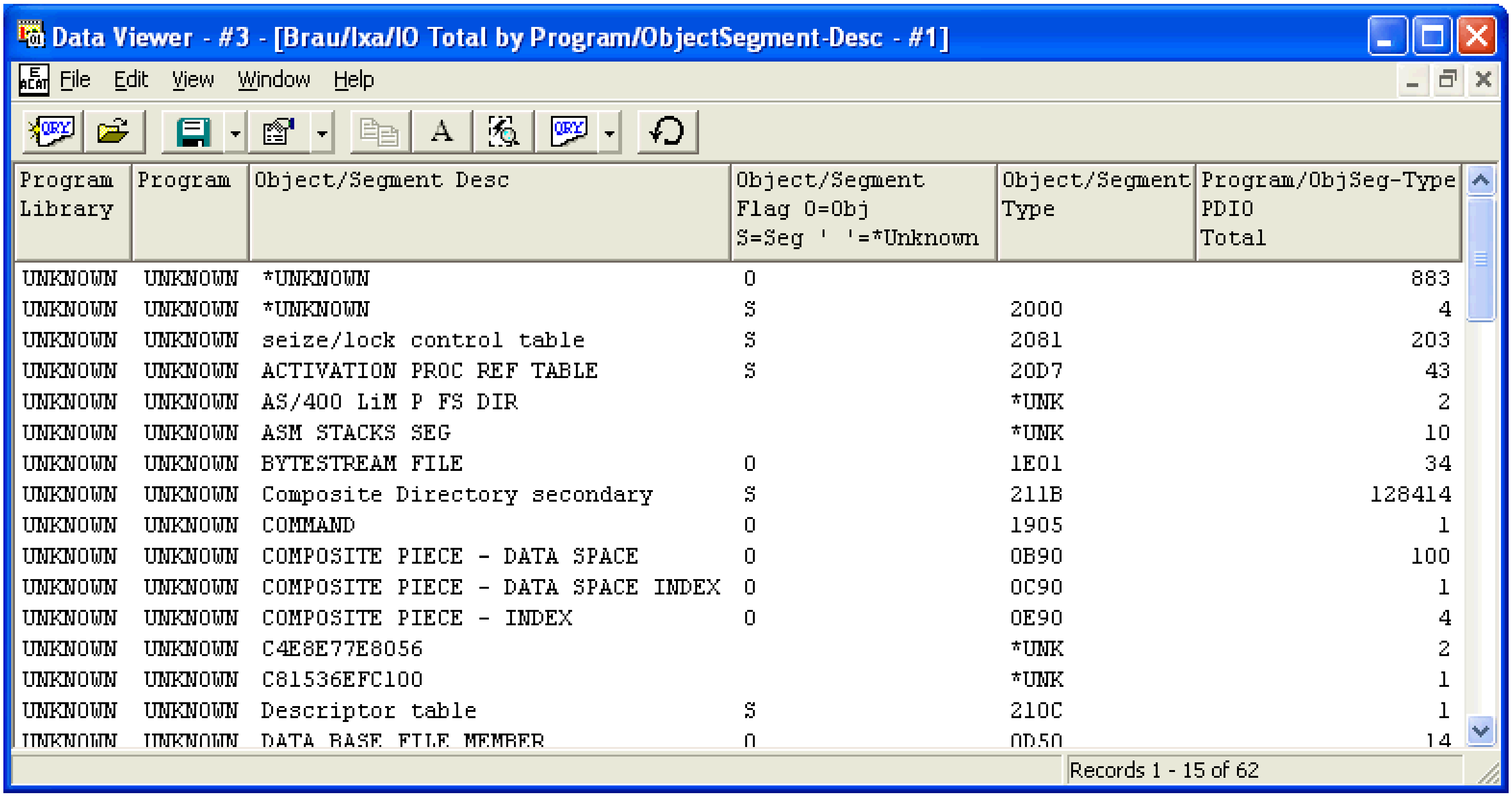
PDIO Total
140302

Records 1 - 1 of 1

Table of Contents | Previous | Next

5.5.51 IO Total by Program/ObjectSegment-Desc

This report shows the total number of physical disk IO events that occurred for each object type by program.

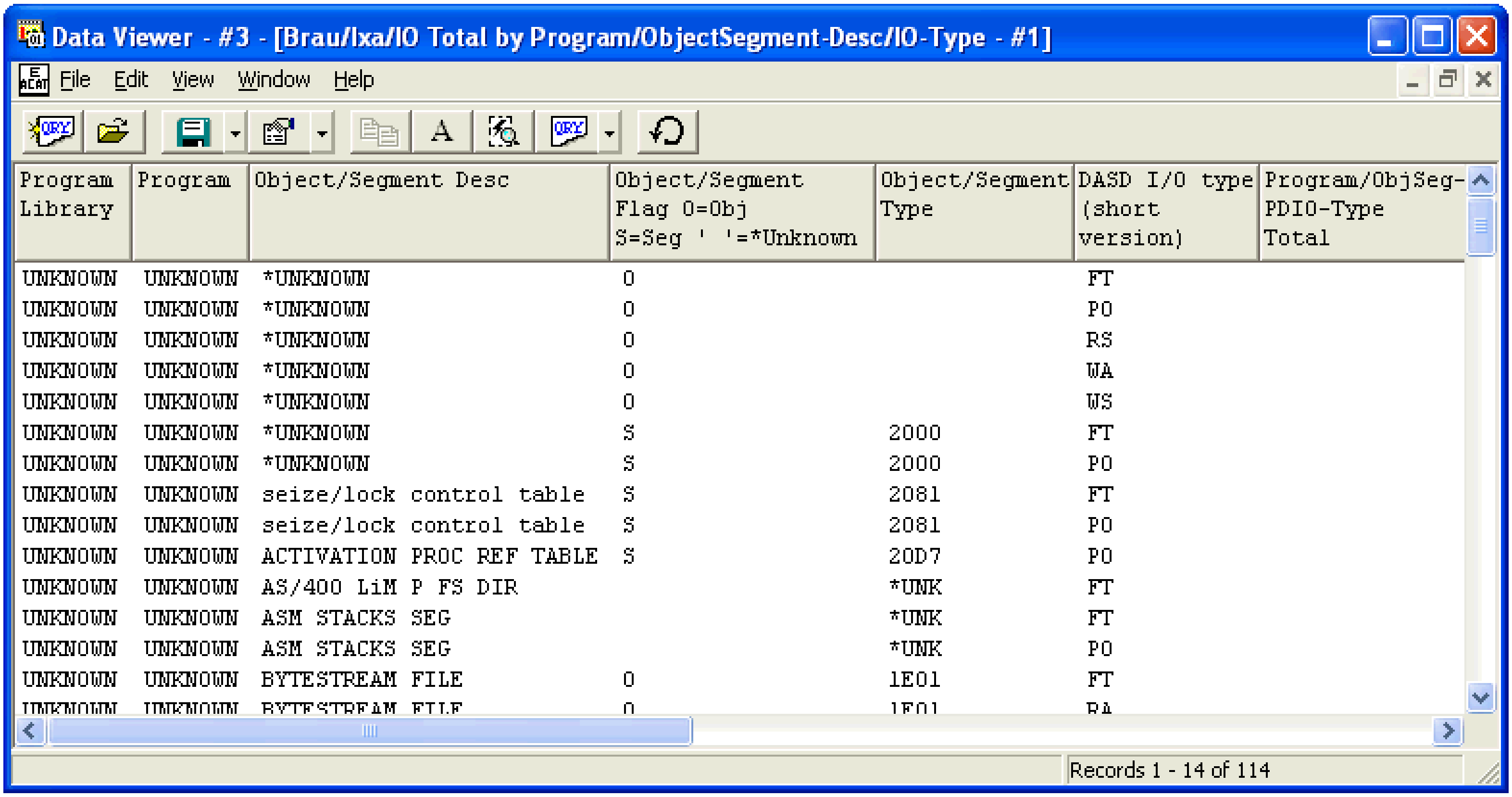

Data Viewer - #3 - [Brau/Ixa/IO Total by Program/ObjectSegment-Desc - #1]
File
Edit
View
Window
Help
Records 1 - 15 of 62

Program Library	Program	Object/Segment Desc	Object/Segment Flag O=Obj S=Seg ' '=*Unknown	Object/Segment Type	Program/ObjSeg-Type PDIO Total
UNKNOWN	UNKNOWN	*UNKNOWN	O		883
UNKNOWN	UNKNOWN	*UNKNOWN	S	2000	4
UNKNOWN	UNKNOWN	seize/lock control table	S	2081	203
UNKNOWN	UNKNOWN	ACTIVATION PROC REF TABLE	S	20D7	43
UNKNOWN	UNKNOWN	AS/400 LiM P FS DIR		*UNK	2
UNKNOWN	UNKNOWN	ASM STACKS SEG		*UNK	10
UNKNOWN	UNKNOWN	BYTESTREAM FILE	O	1E01	34
UNKNOWN	UNKNOWN	Composite Directory secondary	S	211B	128414
UNKNOWN	UNKNOWN	COMMAND	O	1905	1
UNKNOWN	UNKNOWN	COMPOSITE PIECE - DATA SPACE	O	0B90	100
UNKNOWN	UNKNOWN	COMPOSITE PIECE - DATA SPACE INDEX	O	0C90	1
UNKNOWN	UNKNOWN	COMPOSITE PIECE - INDEX	O	0E90	4
UNKNOWN	UNKNOWN	C4E8E77E8056		*UNK	2
UNKNOWN	UNKNOWN	C81536EFC100		*UNK	1
UNKNOWN	UNKNOWN	Descriptor table	S	210C	1
UNKNOWN	UNKNOWN	DATA BASE FILE MEMBER	O	0D50	14

Table of Contents | Previous | Next

5.5.52 IO Total by Program/ObjectSegment-Desc/IO-Type

This report shows the total number of physical disk IO events that occurred for each IO type by object type within program.

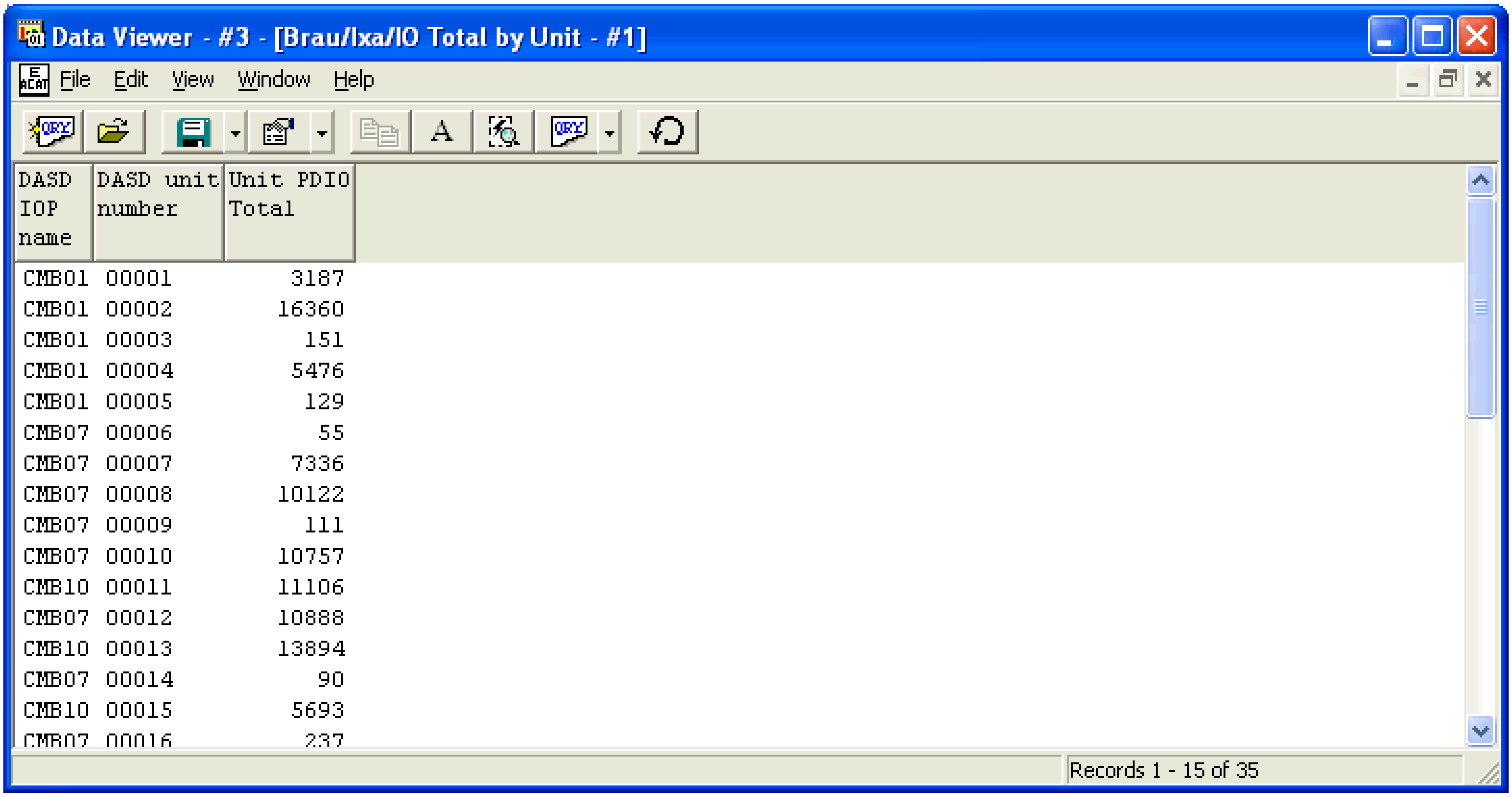

Data Viewer - #3 - [Brau/Ixa/IO Total by Program/ObjectSegment-Desc/IO-Type - #1]
File
Edit
View
Window
Help
Records 1 - 14 of 114

Program Library	Program	Object/Segment Desc	Object/Segment Flag O=Obj S=Seg ' '=*Unknown	Object/Segment Type	DASD I/O type (short version)	Program/ObjSeg-PDIO-Type Total
UNKNOWN	UNKNOWN	*UNKNOWN	O		FT	
UNKNOWN	UNKNOWN	*UNKNOWN	O		PO	
UNKNOWN	UNKNOWN	*UNKNOWN	O		RS	
UNKNOWN	UNKNOWN	*UNKNOWN	O		WA	
UNKNOWN	UNKNOWN	*UNKNOWN	O		WS	
UNKNOWN	UNKNOWN	*UNKNOWN	S	2000	FT	
UNKNOWN	UNKNOWN	*UNKNOWN	S	2000	PO	
UNKNOWN	UNKNOWN	seize/lock control table	S	2081	FT	
UNKNOWN	UNKNOWN	seize/lock control table	S	2081	PO	
UNKNOWN	UNKNOWN	ACTIVATION PROC REF TABLE	S	20D7	PO	
UNKNOWN	UNKNOWN	AS/400 LiM P FS DIR		*UNK	FT	
UNKNOWN	UNKNOWN	ASM STACKS SEG		*UNK	FT	
UNKNOWN	UNKNOWN	ASM STACKS SEG		*UNK	PO	
UNKNOWN	UNKNOWN	BYTESTREAM FILE	O	1E01	FT	
UNKNOWN	UNKNOWN	BYTESTREAM FILE	O	1E01	RA	

Table of Contents Previous Next

5.5.53 IO Total by Unit

This report shows the total number of physical disk IO events that occurred for each disk unit.

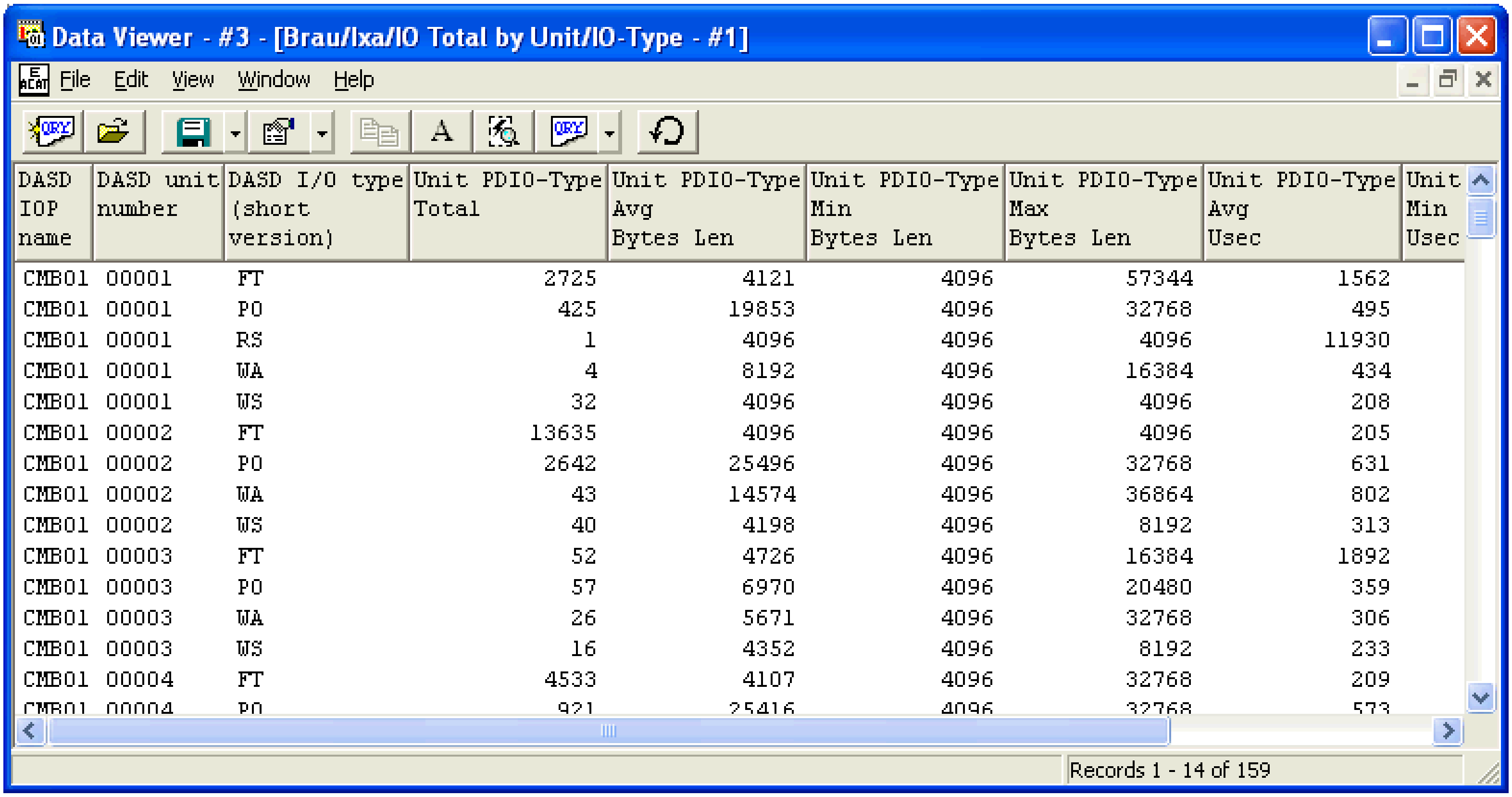


DASD IOP name	DASD unit number	Unit PDIO Total
CMB01	00001	3187
CMB01	00002	16360
CMB01	00003	151
CMB01	00004	5476
CMB01	00005	129
CMB07	00006	55
CMB07	00007	7336
CMB07	00008	10122
CMB07	00009	111
CMB07	00010	10757
CMB10	00011	11106
CMB07	00012	10888
CMB10	00013	13894
CMB07	00014	90
CMB10	00015	5693
CMB07	00016	237

Table of Contents | Previous | Next

5.5.54 IO Total by Unit/IO-Type

This report shows a breakdown of physical disk IO events that occurred by each IO type by disk unit.

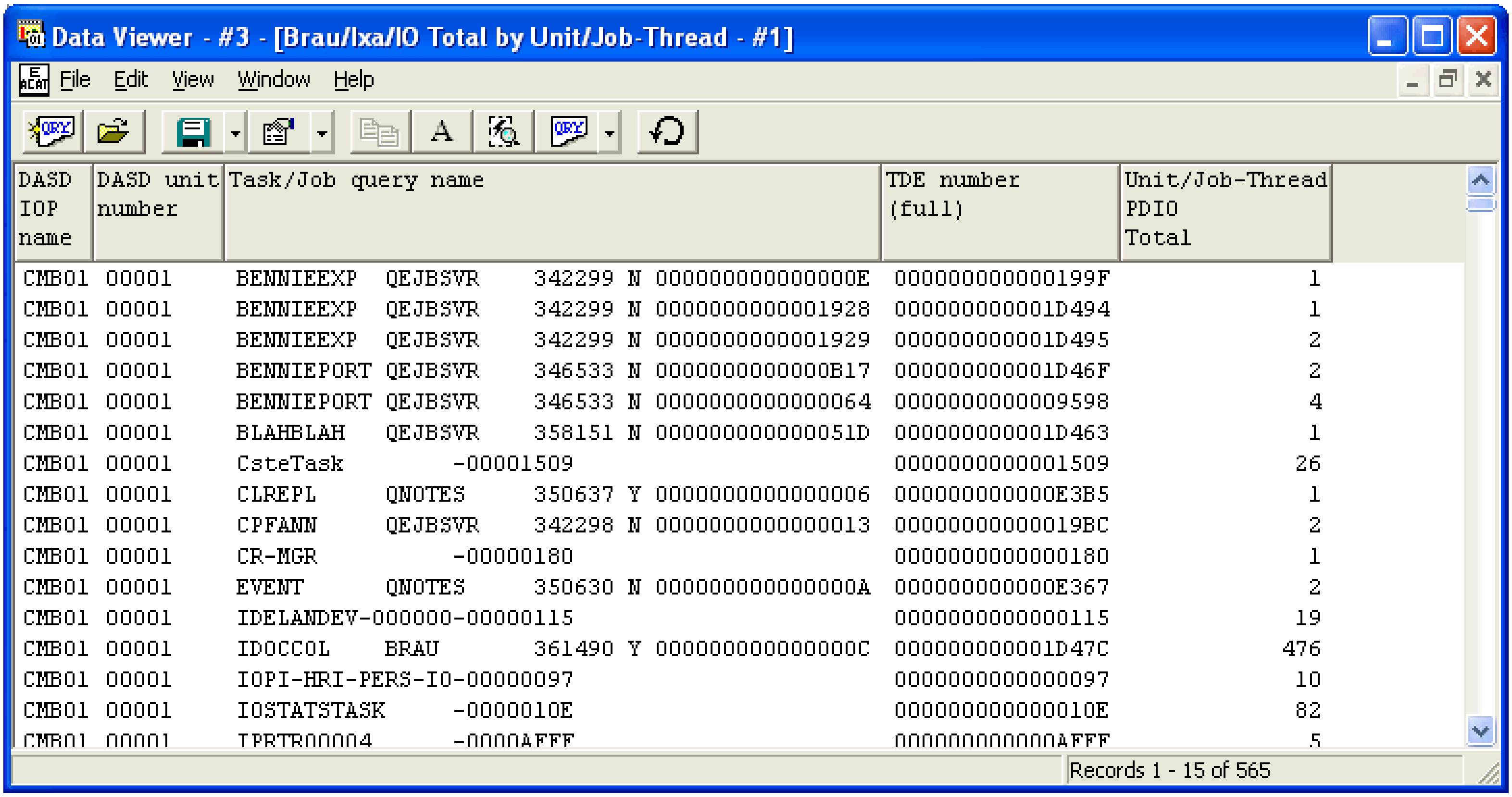


DASD IOP name	DASD unit number	DASD I/O type (short version)	Unit PDIO-Type Total	Unit PDIO-Type Avg Bytes Len	Unit PDIO-Type Min Bytes Len	Unit PDIO-Type Max Bytes Len	Unit PDIO-Type Avg Usec	Unit Min Usec
CMB01	00001	FT	2725	4121	4096	57344	1562	
CMB01	00001	PO	425	19853	4096	32768	495	
CMB01	00001	RS	1	4096	4096	4096	11930	
CMB01	00001	WA	4	8192	4096	16384	434	
CMB01	00001	WS	32	4096	4096	4096	208	
CMB01	00002	FT	13635	4096	4096	4096	205	
CMB01	00002	PO	2642	25496	4096	32768	631	
CMB01	00002	WA	43	14574	4096	36864	802	
CMB01	00002	WS	40	4198	4096	8192	313	
CMB01	00003	FT	52	4726	4096	16384	1892	
CMB01	00003	PO	57	6970	4096	20480	359	
CMB01	00003	WA	26	5671	4096	32768	306	
CMB01	00003	WS	16	4352	4096	8192	233	
CMB01	00004	FT	4533	4107	4096	32768	209	
CMB01	00004	PO	921	25416	4096	32768	573	

Table of Contents Previous Next

5.5.55 IO Total by Unit/Job-Thread

This report shows the total number of physical disk IO events that occurred for each disk unit by job/thread.


Data Viewer - #3 - [Brau/Ixa/IO Total by Unit/Job-Thread - #1]
File
Edit
View
Window
Help

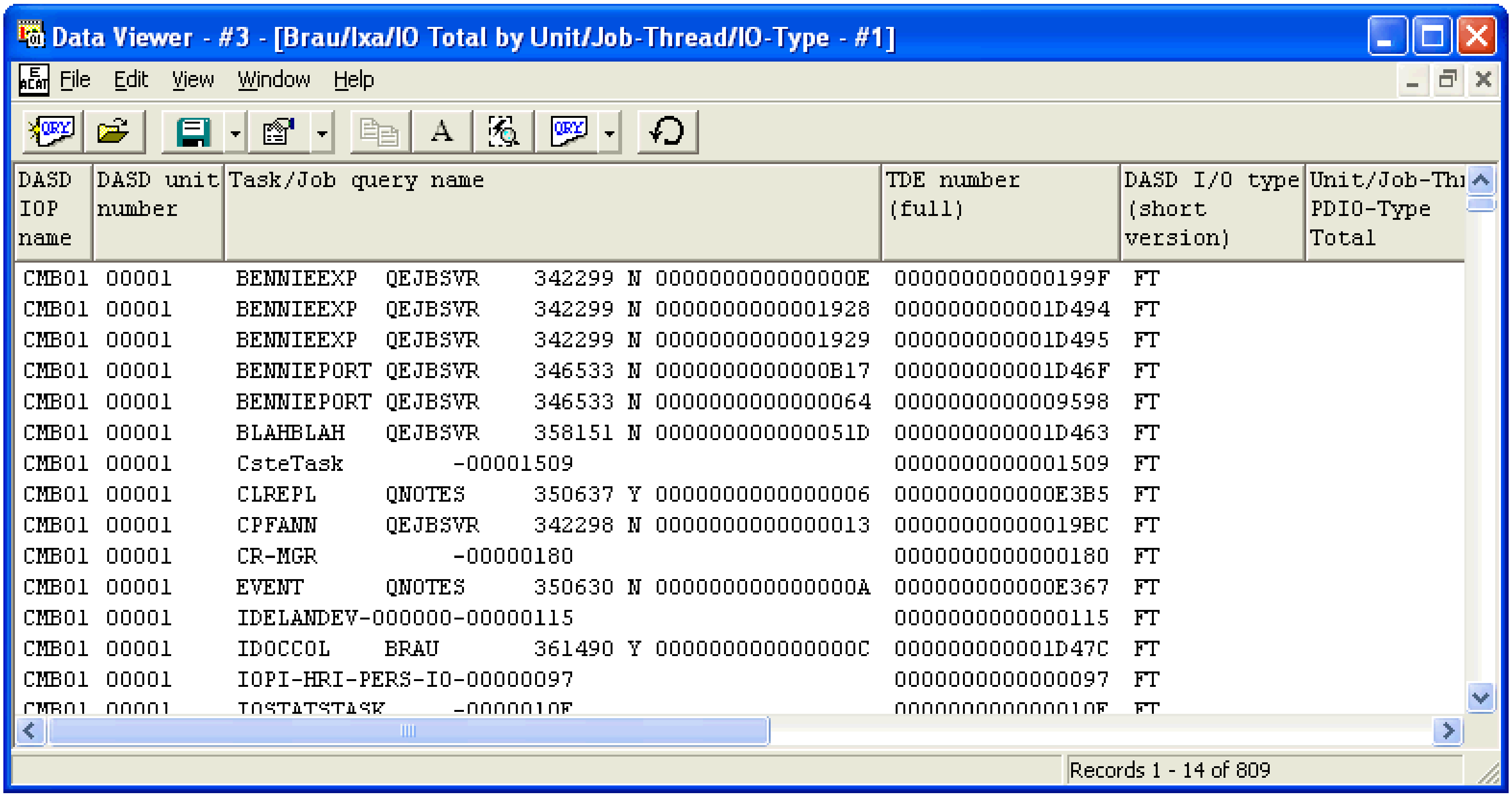
DASD IOP name	DASD unit number	Task/Job query name	TDE number (full)	Unit/Job-Thread PDIO Total
CMB01	00001	BENNIEEXP QEJBSVR 342299 N 000000000000000E	000000000000199F	1
CMB01	00001	BENNIEEXP QEJBSVR 342299 N 0000000000001928	000000000001D494	1
CMB01	00001	BENNIEEXP QEJBSVR 342299 N 0000000000001929	000000000001D495	2
CMB01	00001	BENNIEPORT QEJBSVR 346533 N 0000000000000B17	000000000001D46F	2
CMB01	00001	BENNIEPORT QEJBSVR 346533 N 0000000000000064	0000000000009598	4
CMB01	00001	BLAHBLAH QEJBSVR 358151 N 000000000000051D	000000000001D463	1
CMB01	00001	CsteTask -00001509	0000000000001509	26
CMB01	00001	CLREPL QNOTES 350637 Y 0000000000000006	000000000000E3B5	1
CMB01	00001	CPFANN QEJBSVR 342298 N 0000000000000013	00000000000019BC	2
CMB01	00001	CR-MGR -00000180	0000000000000180	1
CMB01	00001	EVENT QNOTES 350630 N 000000000000000A	000000000000E367	2
CMB01	00001	IDELANDEV-000000-00000115	0000000000000115	19
CMB01	00001	IDOCCOL BRAU 361490 Y 000000000000000C	000000000001D47C	476
CMB01	00001	IOPI-HRI-PERS-IO-00000097	0000000000000097	10
CMB01	00001	IOSTATSTASK -0000010E	000000000000010E	82
CMB01	00001	IPRTR00004 -0000AFFF	000000000000AFFF	5

Records 1 - 15 of 565

Table of Contents | Previous | Next

5.5.56 IO Total by Unit/Job-Thread/IO-Type

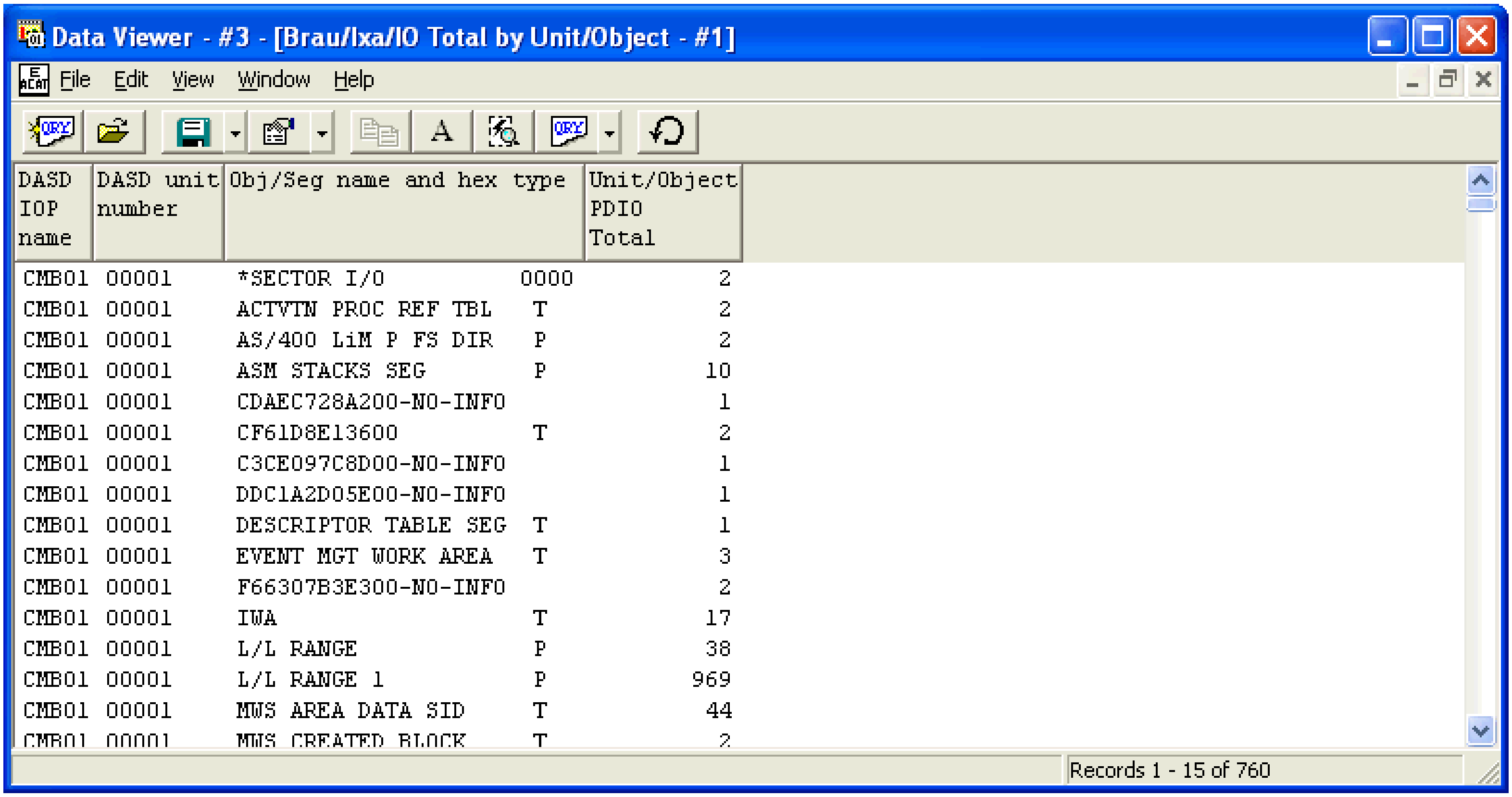
This report shows the total number of physical disk IO events that occurred for each IO type by disk unit by job/thread.


Data Viewer - #3 - [Brau/Ixa/IO Total by Unit/Job-Thread/IO-Type - #1]
File Edit View Window Help
QRY
QRY
Records 1 - 14 of 809

DASD IOP name	DASD unit number	Task/Job query name	TDE number (full)	DASD I/O type (short version)	Unit/Job-Thı PDIO-Type Total
CMB01	00001	BENNIEEXP QEJBSVR 342299 N 000000000000000E	000000000000199F	FT	
CMB01	00001	BENNIEEXP QEJBSVR 342299 N 0000000000001928	000000000001D494	FT	
CMB01	00001	BENNIEEXP QEJBSVR 342299 N 0000000000001929	000000000001D495	FT	
CMB01	00001	BENNIEPORT QEJBSVR 346533 N 0000000000000B17	000000000001D46F	FT	
CMB01	00001	BENNIEPORT QEJBSVR 346533 N 0000000000000064	0000000000009598	FT	
CMB01	00001	BLAHBLAH QEJBSVR 358151 N 000000000000051D	000000000001D463	FT	
CMB01	00001	CsteTask -00001509	0000000000001509	FT	
CMB01	00001	CLREPL QNOTES 350637 Y 0000000000000006	000000000000E3B5	FT	
CMB01	00001	CPFANN QEJBSVR 342298 N 0000000000000013	00000000000019BC	FT	
CMB01	00001	CR-MGR -00000180	0000000000000180	FT	
CMB01	00001	EVENT QNOTES 350630 N 000000000000000A	000000000000E367	FT	
CMB01	00001	IDELANDEV-000000-00000115	0000000000000115	FT	
CMB01	00001	IDOCCOL BRAU 361490 Y 000000000000000C	000000000001D47C	FT	
CMB01	00001	IOPI-HRI-PERS-IO-00000097	0000000000000097	FT	
CMB01	00001	IOSTATSTASK -0000010F	000000000000010F	FT	

5.5.57 IO Total by Unit/Object

This report shows the total number of physical disk IO events that occurred for each object within disk unit.



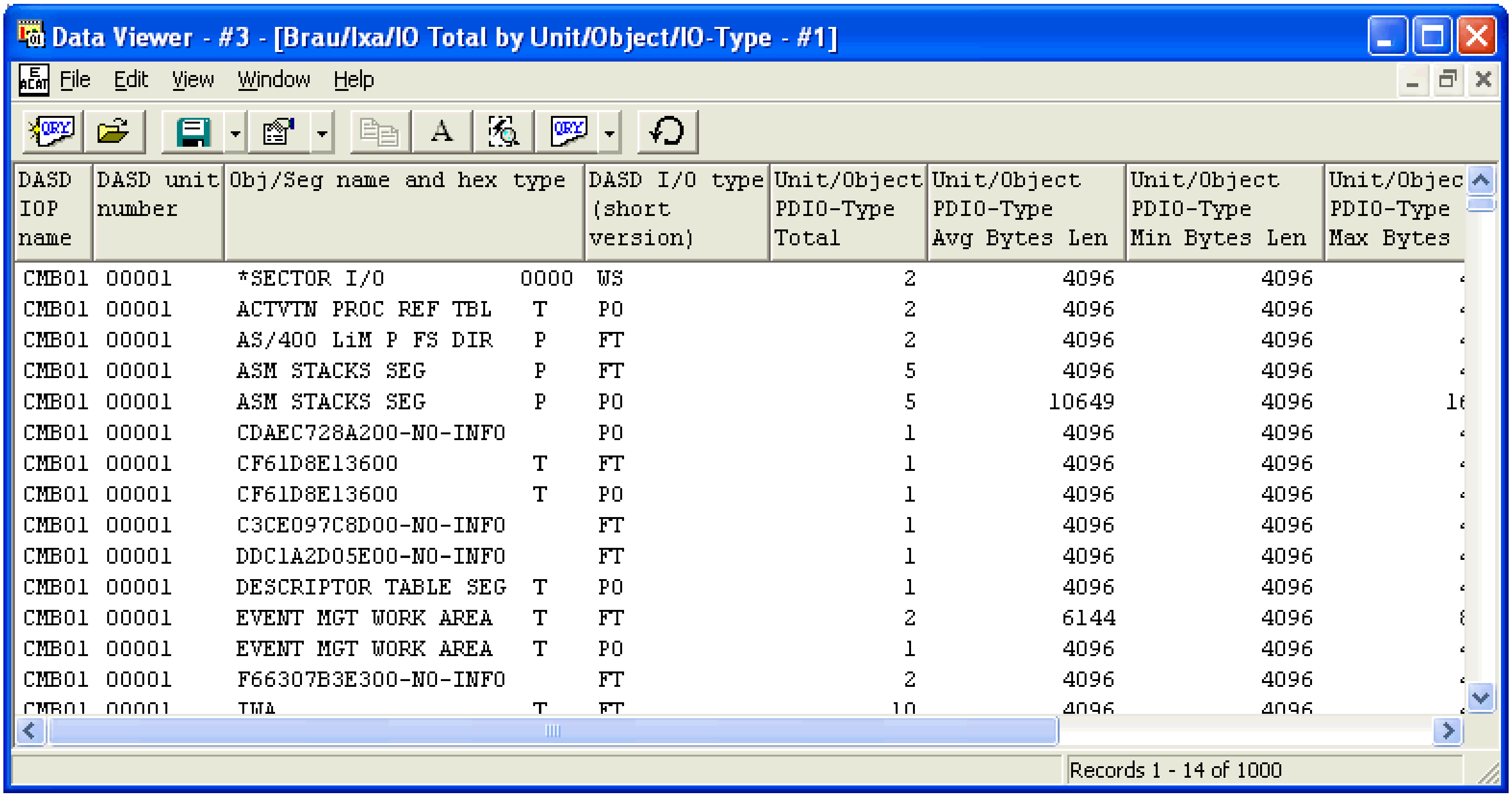
DASD IOP name	DASD unit number	Obj/Seg name and hex type	Unit/Object PDIO Total
CMB01	00001	*SECTOR I/O 0000	2
CMB01	00001	ACTVTN PROC REF TBL T	2
CMB01	00001	AS/400 LiM P FS DIR P	2
CMB01	00001	ASM STACKS SEG P	10
CMB01	00001	CDAEC728A200-NO-INFO	1
CMB01	00001	CF61D8E13600 T	2
CMB01	00001	C3CE097C8D00-NO-INFO	1
CMB01	00001	DDC1A2D05E00-NO-INFO	1
CMB01	00001	DESCRIPTOR TABLE SEG T	1
CMB01	00001	EVENT MGT WORK AREA T	3
CMB01	00001	F66307B3E300-NO-INFO	2
CMB01	00001	IWA T	17
CMB01	00001	L/L RANGE P	38
CMB01	00001	L/L RANGE 1 P	969
CMB01	00001	MWS AREA DATA SID T	44
CMB01	00001	MWS CREATED BLOCK T	2

Records 1 - 15 of 760

Table of Contents Previous Next

5.5.58 IO Total by Unit/Object/IO-Type

This report shows a breakdown of physical disk IO events for each IO type by object within disk unit.


Data Viewer - #3 - [Brau/lxa/IO Total by Unit/Object/IO-Type - #1]
File
Edit
View
Window
Help
Records 1 - 14 of 1000

DASD IOP name	DASD unit number	Obj/Seg name and hex type	DASD I/O type (short version)	Unit/Object PDIO-Type Total	Unit/Object PDIO-Type Avg Bytes Len	Unit/Object PDIO-Type Min Bytes Len	Unit/Objec PDIO-Type Max Bytes
CMB01	00001	*SECTOR I/O 0000	WS	2	4096	4096	4
CMB01	00001	ACTVTN PROC REF TBL T	PO	2	4096	4096	4
CMB01	00001	AS/400 LiM P FS DIR P	FT	2	4096	4096	4
CMB01	00001	ASM STACKS SEG P	FT	5	4096	4096	4
CMB01	00001	ASM STACKS SEG P	PO	5	10649	4096	16
CMB01	00001	CDAEC728A200-NO-INFO	PO	1	4096	4096	4
CMB01	00001	CF61D8E13600 T	FT	1	4096	4096	4
CMB01	00001	CF61D8E13600 T	PO	1	4096	4096	4
CMB01	00001	C3CE097C8D00-NO-INFO	FT	1	4096	4096	4
CMB01	00001	DDC1A2D05E00-NO-INFO	FT	1	4096	4096	4
CMB01	00001	DESCRIPTOR TABLE SEG T	PO	1	4096	4096	4
CMB01	00001	EVENT MGT WORK AREA T	FT	2	6144	4096	8
CMB01	00001	EVENT MGT WORK AREA T	PO	1	4096	4096	4
CMB01	00001	F66307B3E300-NO-INFO	FT	2	4096	4096	4
CMB01	00001	TWA T	FT	10	4096	4096	4

Table of Contents | Previous | Next

5.6 Size change to objects and segments reports

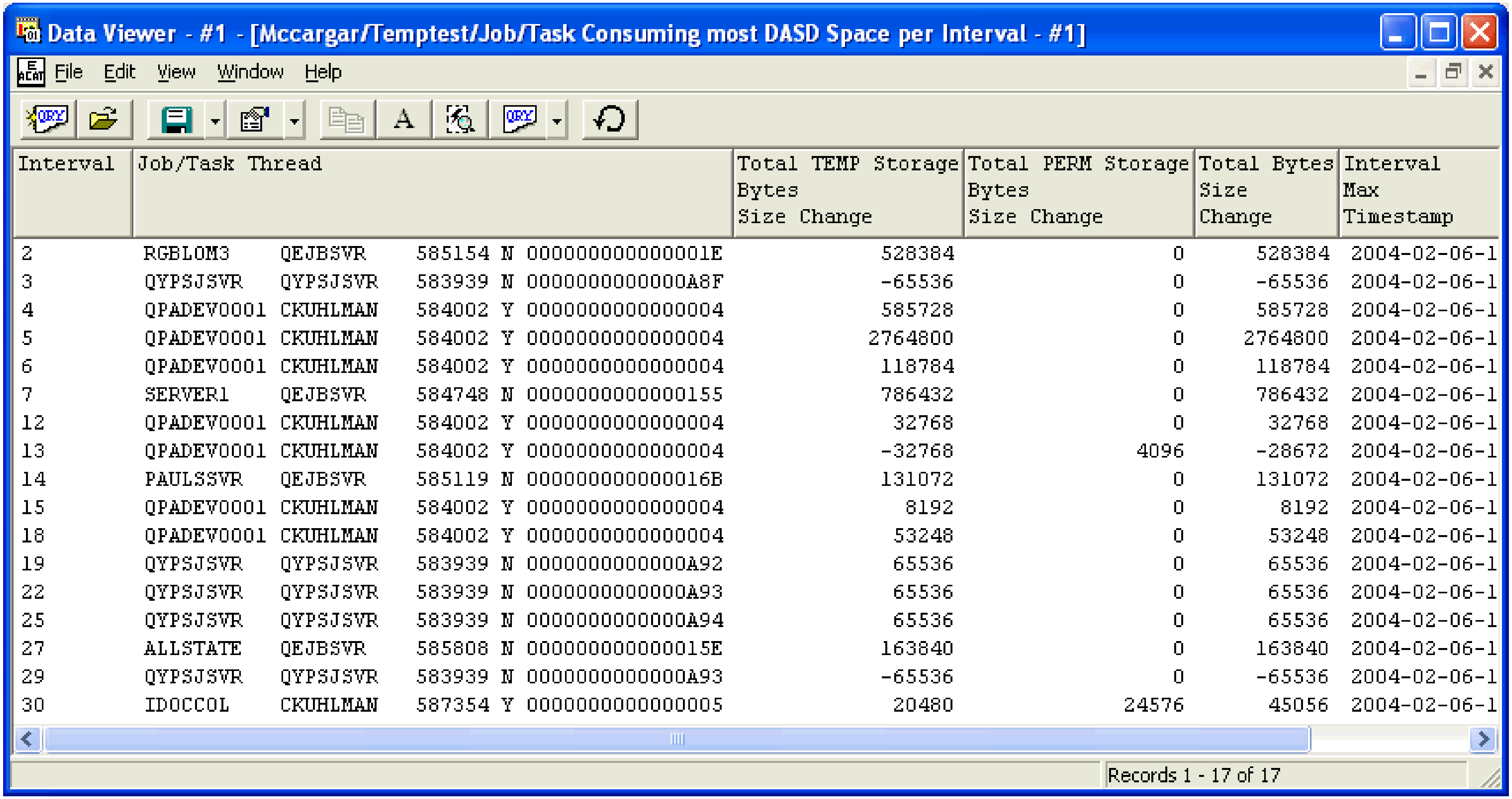
This analysis shows every dasd space create/destroy or size change of external objects/segments. It covers both permanent and temporary objects.
The analysis detail includes the object name and object type, causing job/task and length (size) of the create/destroy.

Table of Contents Previous Next

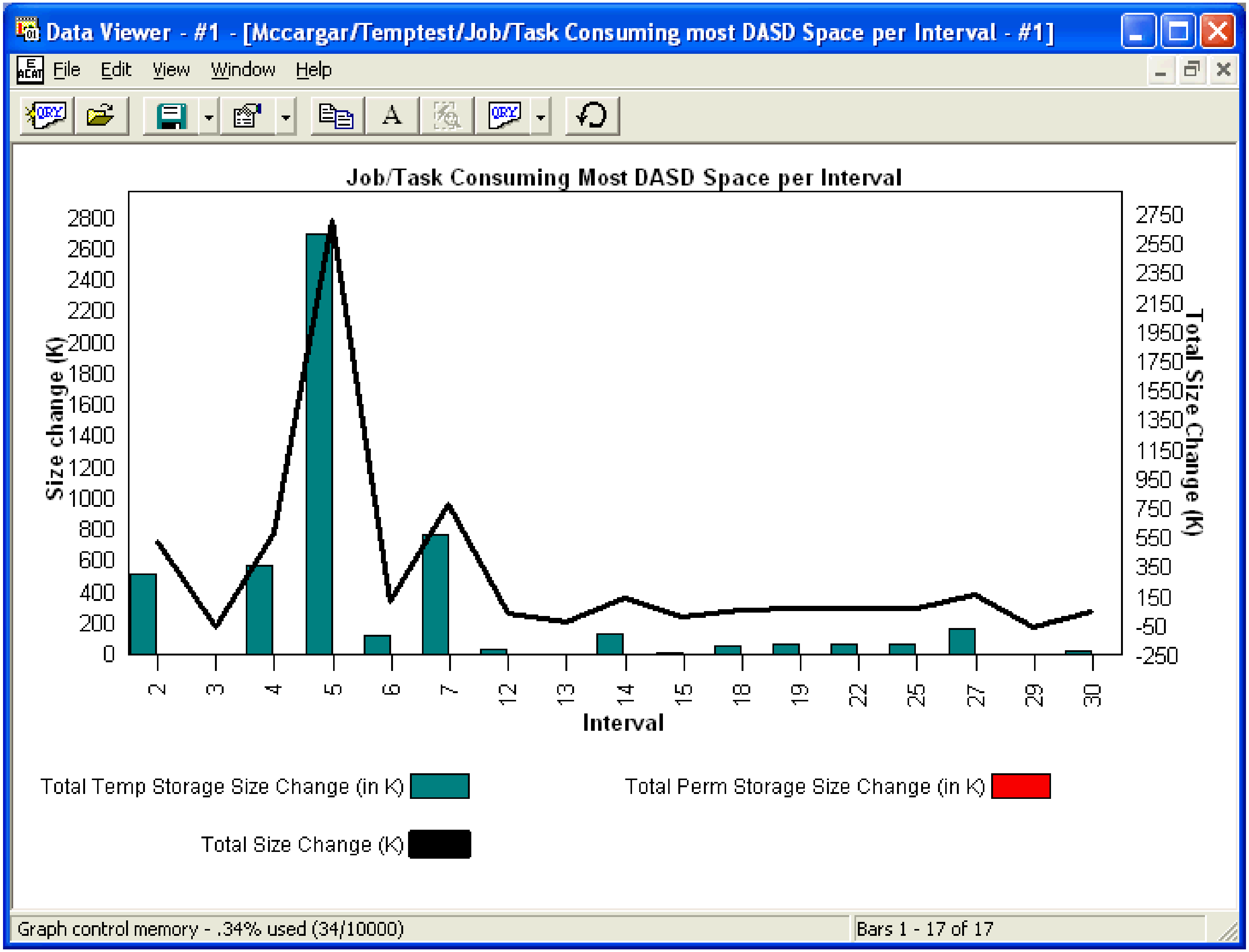
5.6.1 Job/Task Consuming most DASD Space per Interval

Description: This report shows the job within each interval of the collection that consumed the most DASD space. The delta bytes consumed by this job is provided.

Example:


Data Viewer - #1 - [Mccargar/Temptest/Job/Task Consuming most DASD Space per Interval - #1]
File
Edit
View
Window
Help
Records 1 - 17 of 17

Interval	Job/Task Thread	Total TEMP Storage Bytes Size Change	Total PERM Storage Bytes Size Change	Total Bytes Size Change	Interval Max Timestamp
2	RGBLOM3 QEJBSVR 585154 N 000000000000001E	528384	0	528384	2004-02-06-1
3	QYPSJSVR QYPSJSVR 583939 N 0000000000000A8F	-65536	0	-65536	2004-02-06-1
4	QPADEV0001 CKUHLMAN 584002 Y 0000000000000004	585728	0	585728	2004-02-06-1
5	QPADEV0001 CKUHLMAN 584002 Y 0000000000000004	2764800	0	2764800	2004-02-06-1
6	QPADEV0001 CKUHLMAN 584002 Y 0000000000000004	118784	0	118784	2004-02-06-1
7	SERVER1 QEJBSVR 584748 N 0000000000000155	786432	0	786432	2004-02-06-1
12	QPADEV0001 CKUHLMAN 584002 Y 0000000000000004	32768	0	32768	2004-02-06-1
13	QPADEV0001 CKUHLMAN 584002 Y 0000000000000004	-32768	4096	-28672	2004-02-06-1
14	PAULSSVR QEJBSVR 585119 N 000000000000016B	131072	0	131072	2004-02-06-1
15	QPADEV0001 CKUHLMAN 584002 Y 0000000000000004	8192	0	8192	2004-02-06-1
18	QPADEV0001 CKUHLMAN 584002 Y 0000000000000004	53248	0	53248	2004-02-06-1
19	QYPSJSVR QYPSJSVR 583939 N 0000000000000A92	65536	0	65536	2004-02-06-1
22	QYPSJSVR QYPSJSVR 583939 N 0000000000000A93	65536	0	65536	2004-02-06-1
25	QYPSJSVR QYPSJSVR 583939 N 0000000000000A94	65536	0	65536	2004-02-06-1
27	ALLSTATE QEJBSVR 585808 N 000000000000015E	163840	0	163840	2004-02-06-1
29	QYPSJSVR QYPSJSVR 583939 N 0000000000000A93	-65536	0	-65536	2004-02-06-1
30	IDOCCOL CKUHLMAN 587354 Y 0000000000000005	20480	24576	45056	2004-02-06-1



Graph control memory - .34% used (34/10000)

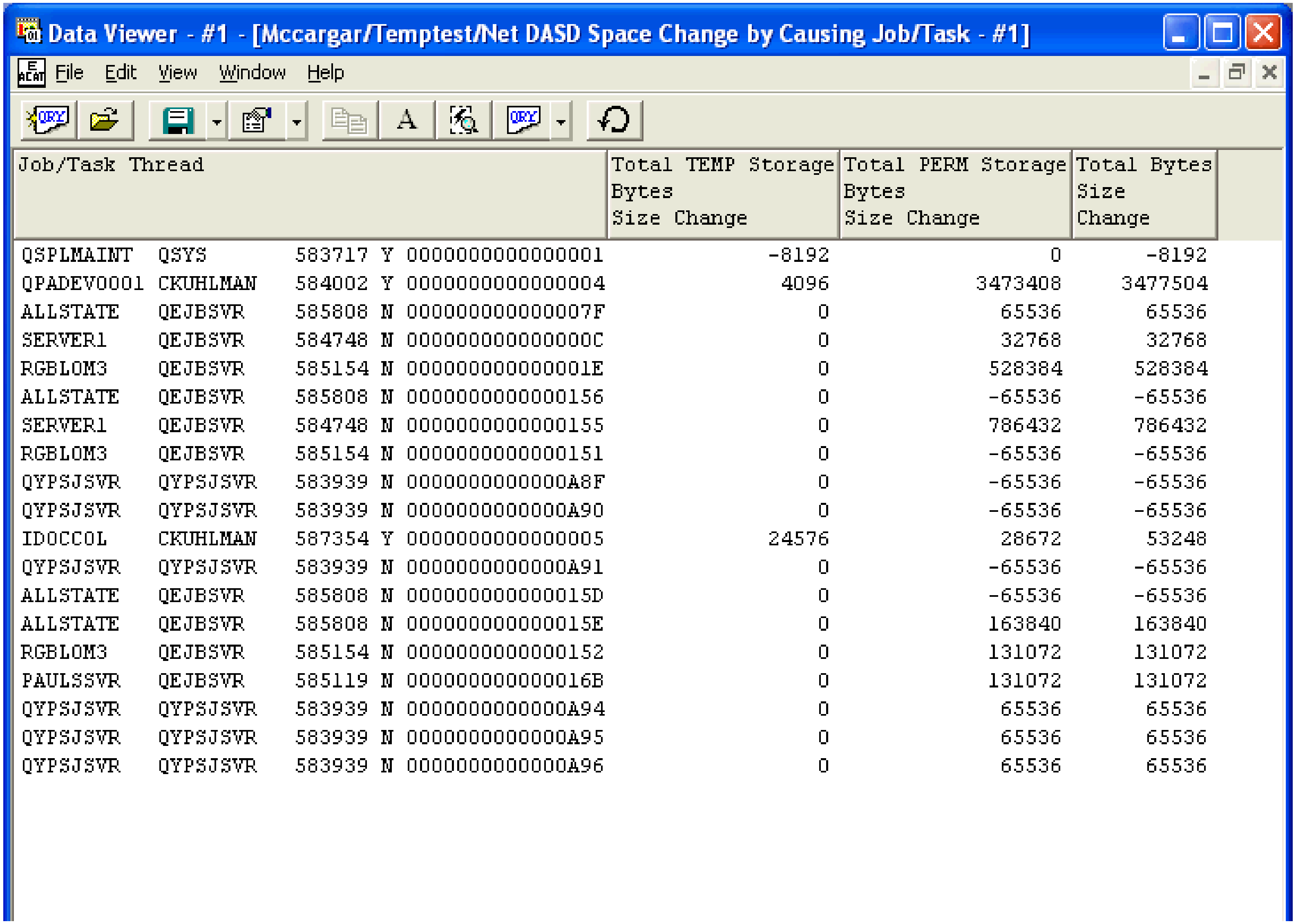
Bars 1 - 17 of 17

Table of Contents | Previous | Next

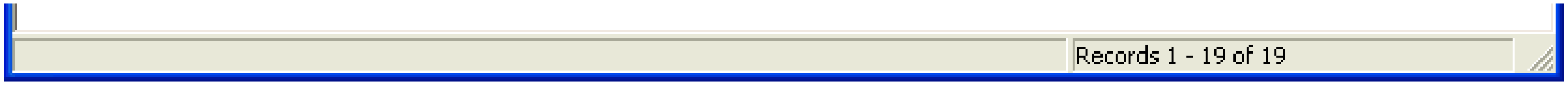
5.6.2 Net DASD Space Change by Causing Job/Task

Description: The report shows the temporary and permanent Dasd space change for each job/task. In the graph the user can click on any bar (temporary or permanent space change) and get information on the job/task(fully qualified job name or task name) causing it, and space change. The graph's right y-axis provides the total of the temporary and permanent Dasd space change.

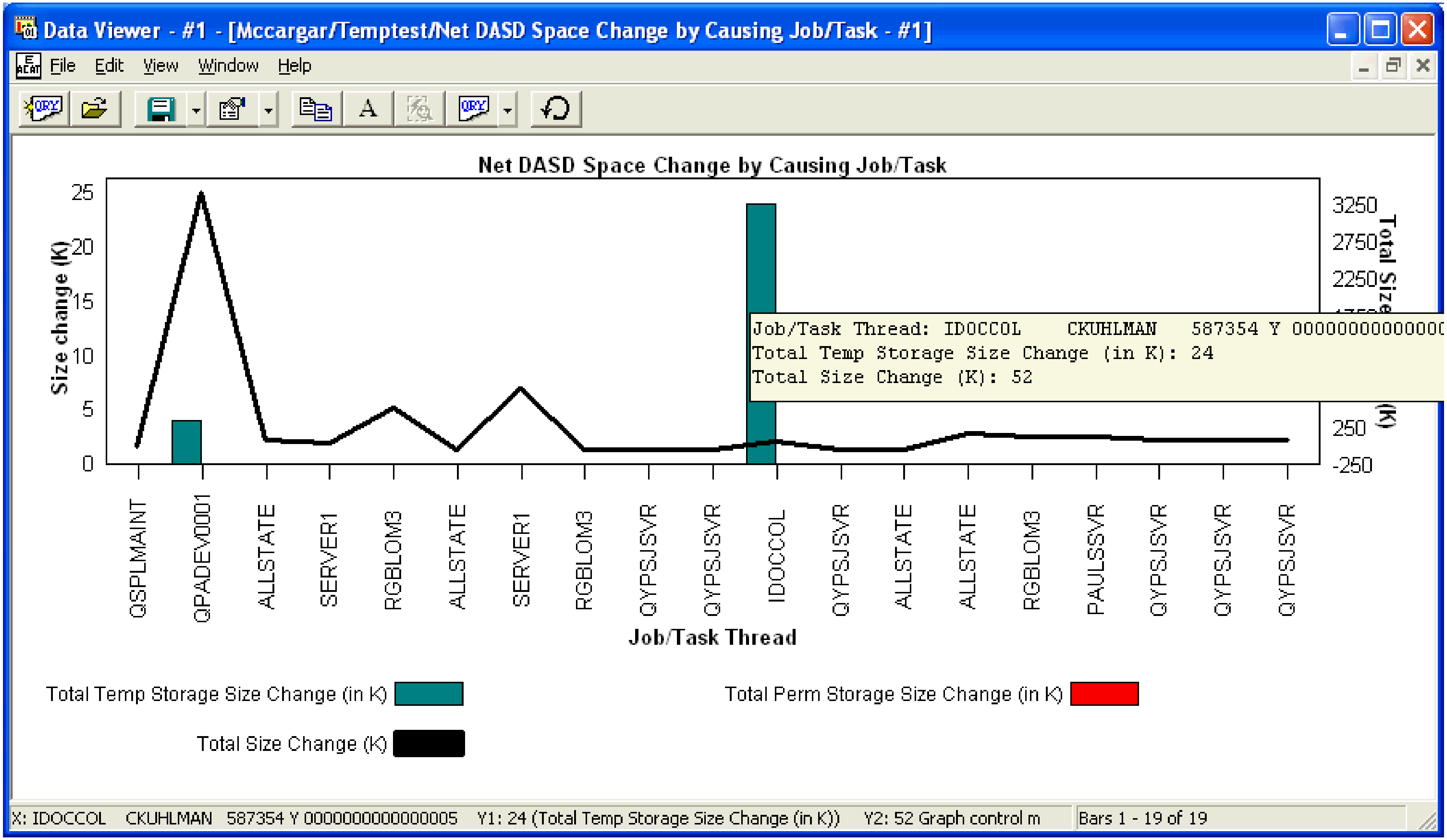
Example:


Data Viewer - #1 - [Mccargar/Temptest/Net DASD Space Change by Causing Job/Task - #1]
File Edit View Window Help

Job/Task Thread	Total TEMP Storage Bytes Size Change	Total PERM Storage Bytes Size Change	Total Bytes Size Change
QSPLMAINT QSYS 583717 Y 0000000000000001	-8192	0	-8192
QPADEV0001 CKUHLMAN 584002 Y 0000000000000004	4096	3473408	3477504
ALLSTATE QEJBSVR 585808 N 000000000000007F	0	65536	65536
SERVER1 QEJBSVR 584748 N 000000000000000C	0	32768	32768
RGBLOM3 QEJBSVR 585154 N 000000000000001E	0	528384	528384
ALLSTATE QEJBSVR 585808 N 0000000000000156	0	-65536	-65536
SERVER1 QEJBSVR 584748 N 0000000000000155	0	786432	786432
RGBLOM3 QEJBSVR 585154 N 0000000000000151	0	-65536	-65536
QYPSJSVR QYPSJSVR 583939 N 0000000000000A8F	0	-65536	-65536
QYPSJSVR QYPSJSVR 583939 N 0000000000000A90	0	-65536	-65536
IDOCCOL CKUHLMAN 587354 Y 0000000000000005	24576	28672	53248
QYPSJSVR QYPSJSVR 583939 N 0000000000000A91	0	-65536	-65536
ALLSTATE QEJBSVR 585808 N 000000000000015D	0	-65536	-65536
ALLSTATE QEJBSVR 585808 N 000000000000015E	0	163840	163840
RGBLOM3 QEJBSVR 585154 N 0000000000000152	0	131072	131072
PAULSSVR QEJBSVR 585119 N 000000000000016B	0	131072	131072
QYPSJSVR QYPSJSVR 583939 N 0000000000000A94	0	65536	65536
QYPSJSVR QYPSJSVR 583939 N 0000000000000A95	0	65536	65536
QYPSJSVR QYPSJSVR 583939 N 0000000000000A96	0	65536	65536



Records 1 - 19 of 19



```
Job/Task Thread: IDOCCOL    CKUHLMAN   587354 Y 00000000000000
Total Temp Storage Size Change (in K): 24
Total Size Change (K): 52
```

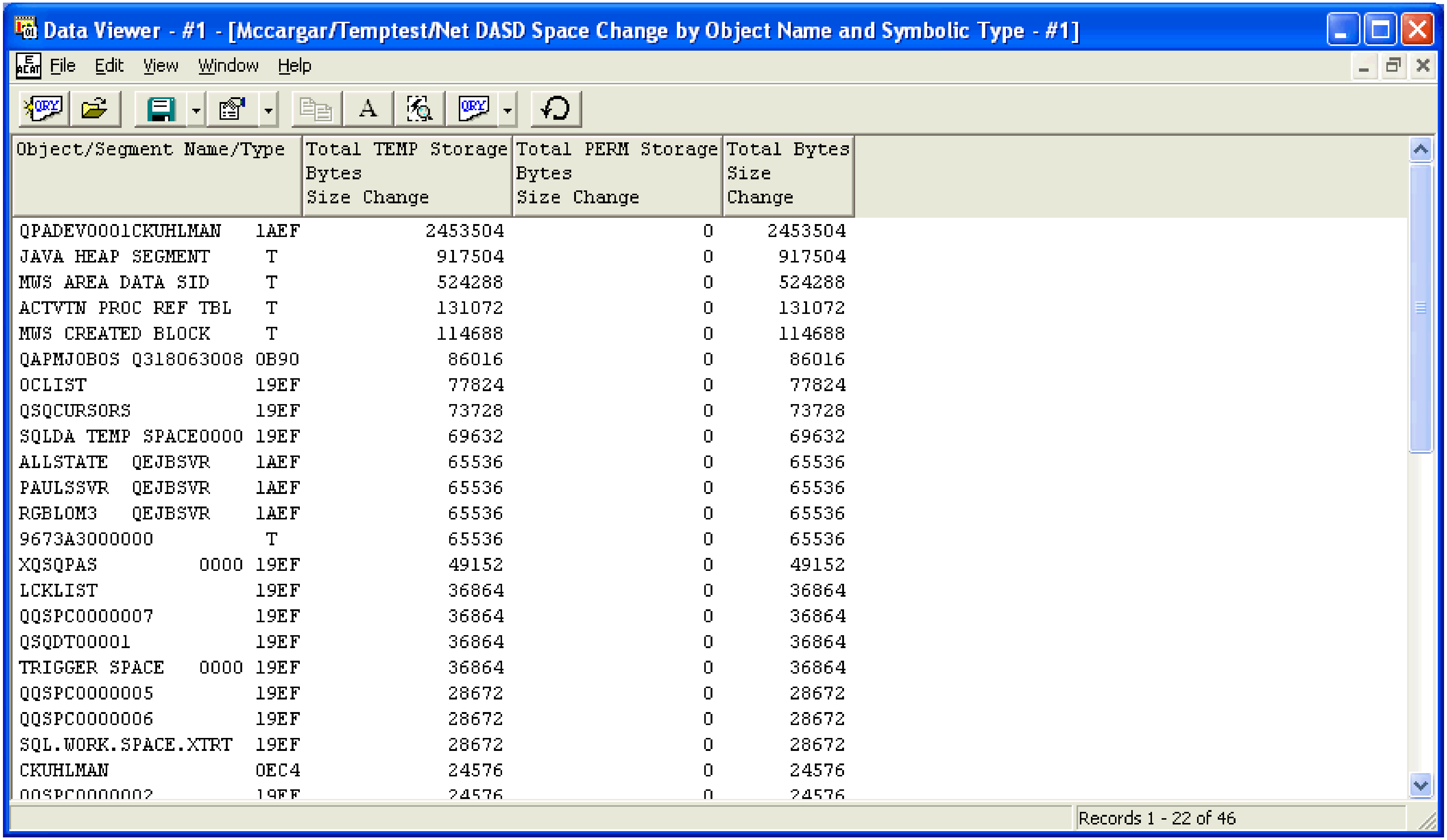
X: IDOCCOL   CKUHLMAN 587354 Y 0000000000000005 Y1: 24 (Total Temp Storage Size Change (in K)) Y2: 52 Graph control m   Bars 1 - 19 of 19

Table of Contents Previous Next

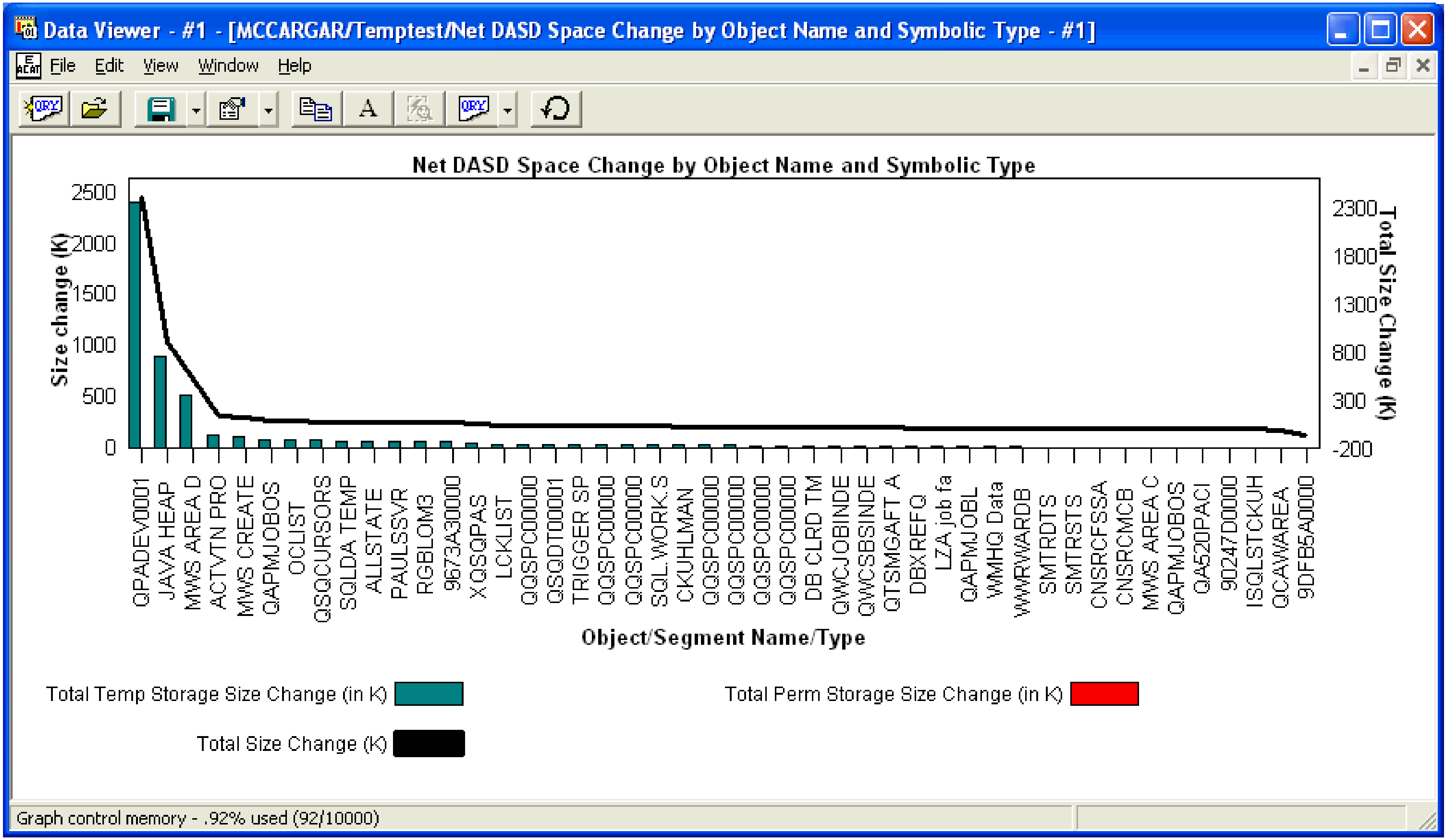
5.6.3 Net DASD Space Change by Object Name and Symbolic Type

Description:

Example:


Data Viewer - #1 - [Mccargar/Temptest/Net DASD Space Change by Object Name and Symbolic Type - #1]
File
Edit
View
Window
Help
Records 1 - 22 of 46

Object/Segment Name/Type	Total TEMP Storage Bytes Size Change	Total PERM Storage Bytes Size Change	Total Bytes Size Change
QPADEV0001CKUHLMAN 1AEF	2453504	0	2453504
JAVA HEAP SEGMENT T	917504	0	917504
MWS AREA DATA SID T	524288	0	524288
ACTVTN PROC REF TBL T	131072	0	131072
MWS CREATED BLOCK T	114688	0	114688
QAPMJOBOS Q318063008 0B90	86016	0	86016
OCLIST 19EF	77824	0	77824
QSQCURSORS 19EF	73728	0	73728
SQLDA TEMP SPACE0000 19EF	69632	0	69632
ALLSTATE QEJBSVR 1AEF	65536	0	65536
PAULSSVR QEJBSVR 1AEF	65536	0	65536
RGBLOM3 QEJBSVR 1AEF	65536	0	65536
9673A3000000 T	65536	0	65536
XQSQPAS 0000 19EF	49152	0	49152
LCKLIST 19EF	36864	0	36864
QQSPC0000007 19EF	36864	0	36864
QSQDT00001 19EF	36864	0	36864
TRIGGER SPACE 0000 19EF	36864	0	36864
QQSPC0000005 19EF	28672	0	28672
QQSPC0000006 19EF	28672	0	28672
SQL.WORK.SPACE.XTRT 19EF	28672	0	28672
CKUHLMAN 0EC4	24576	0	24576
QQSPC0000002 19EF	24576	0	24576



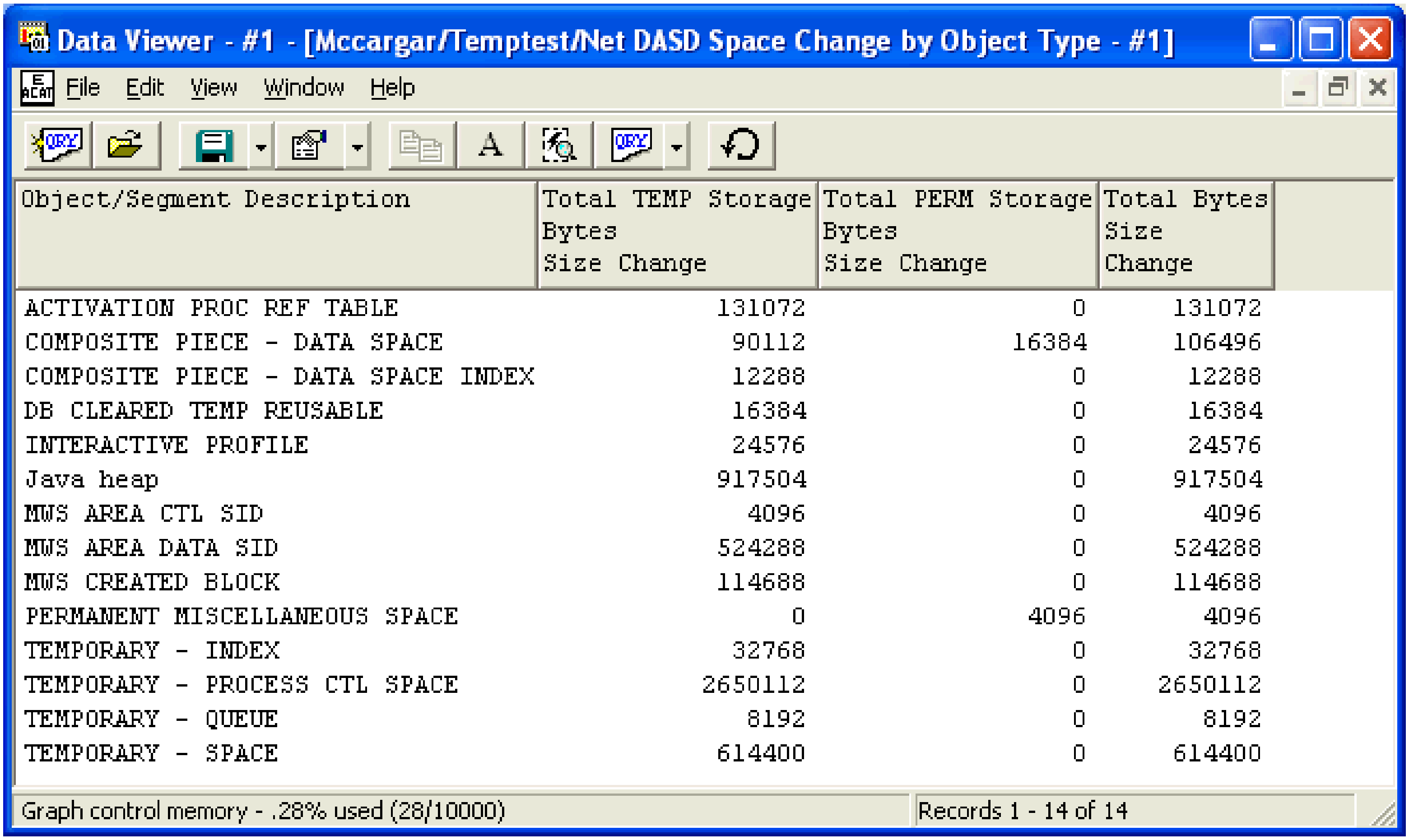
Graph control memory - .92% used (92/10000)

Table of Contents | Previous | Next

5.6.4 Net DASD Space Change by Object Type

Description:

Example:


Data Viewer - #1 - [Mccargar/Temptest/Net DASD Space Change by Object Type - #1]
File
Edit
View
Window
Help

Object/Segment Description	Total TEMP Storage Bytes Size Change	Total PERM Storage Bytes Size Change	Total Bytes Size Change
ACTIVATION PROC REF TABLE	131072	0	131072
COMPOSITE PIECE - DATA SPACE	90112	16384	106496
COMPOSITE PIECE - DATA SPACE INDEX	12288	0	12288
DB CLEARED TEMP REUSABLE	16384	0	16384
INTERACTIVE PROFILE	24576	0	24576
Java heap	917504	0	917504
MWS AREA CTL SID	4096	0	4096
MWS AREA DATA SID	524288	0	524288
MWS CREATED BLOCK	114688	0	114688
PERMANENT MISCELLANEOUS SPACE	0	4096	4096
TEMPORARY - INDEX	32768	0	32768
TEMPORARY - PROCESS CTL SPACE	2650112	0	2650112
TEMPORARY - QUEUE	8192	0	8192
TEMPORARY - SPACE	614400	0	614400

Graph control memory - .28% used (28/10000)

Records 1 - 14 of 14

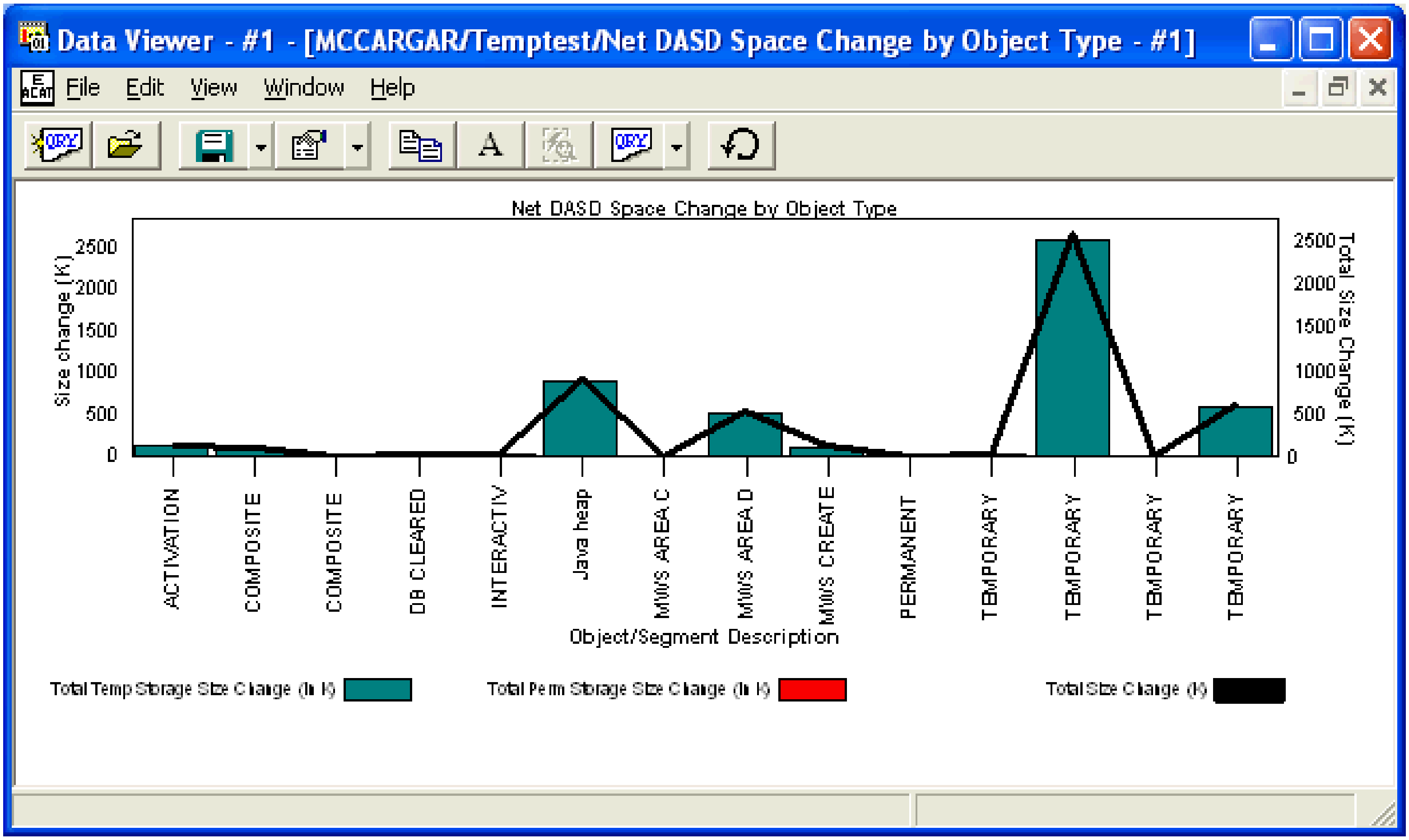
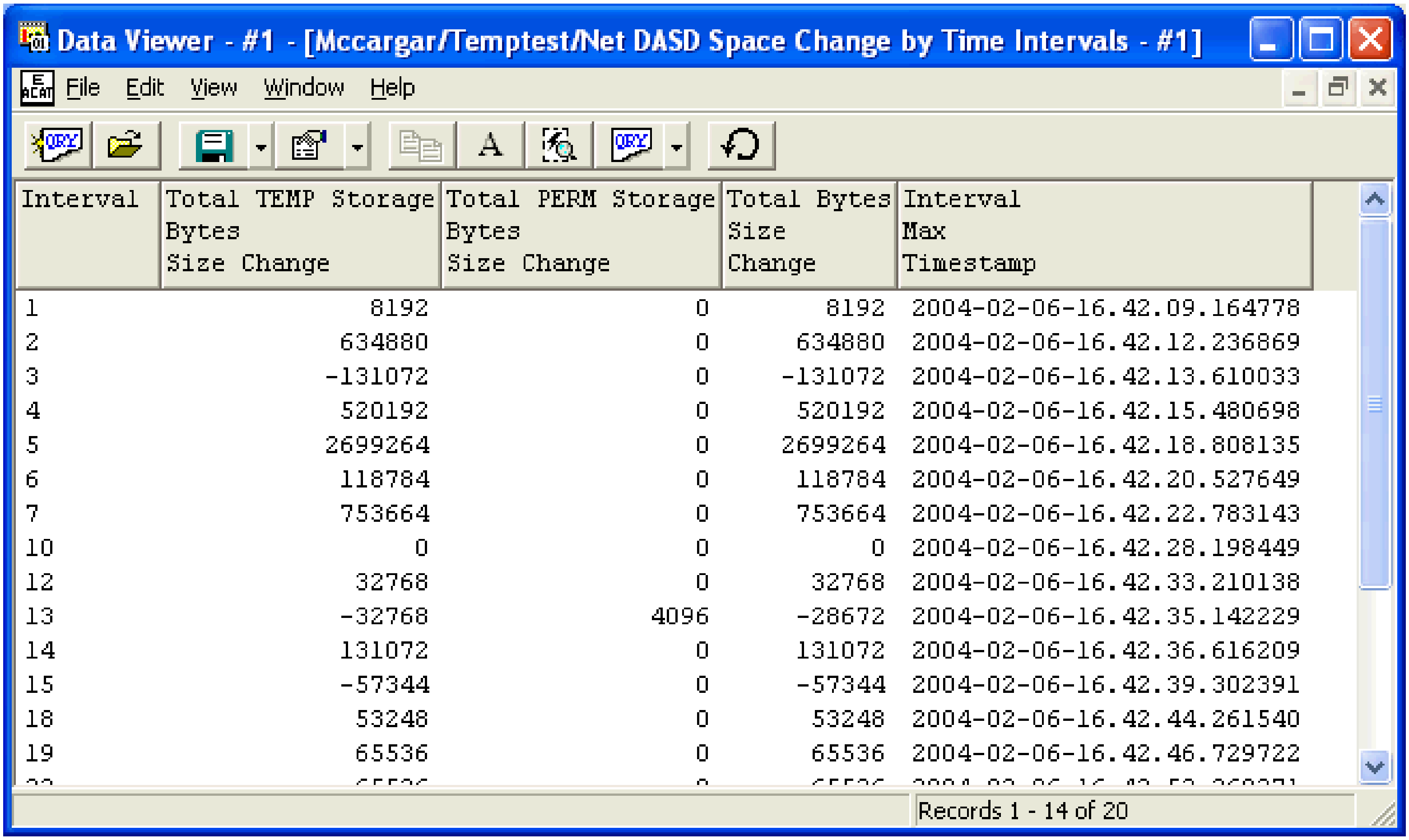

Data Viewer - #1 - [MCCARGAR/Temptest/Net DASD Space Change by Object Type - #1]
File
Edit
View
Window
Help
Net DASD Space Change by Object Type
Size change (K)
2500
2000
1500
1000
500
0
Total Size Change (K)
ACTIVATION
COMPOSITE
COMPOSITE
DB CLEARED
INTERACTIV
Java heap
MI/IS AREA C
MI/IS AREA D
MI/IS CREATE
PERMANENT
TEMPORARY
TEMPORARY
TEMPORARY
TEMPORARY
Object/Segment Description
Total Temp Storage Size Change (In K)
Total Perm Storage Size Change (In K)
Total Size Change (K)

Table of Contents Previous Next

5.6.5 Net DASD Space Change by Time Intervals

Description:

Example:



Data Viewer - #1 - [Mccargar/Temptest/Net DASD Space Change by Time Intervals - #1]
File Edit View Window Help
QRY
A
QRY
Records 1 - 14 of 20

Interval	Total TEMP Storage Bytes Size Change	Total PERM Storage Bytes Size Change	Total Bytes Size Change	Interval Max Timestamp
1	8192	0	8192	2004-02-06-16.42.09.164778
2	634880	0	634880	2004-02-06-16.42.12.236869
3	-131072	0	-131072	2004-02-06-16.42.13.610033
4	520192	0	520192	2004-02-06-16.42.15.480698
5	2699264	0	2699264	2004-02-06-16.42.18.808135
6	118784	0	118784	2004-02-06-16.42.20.527649
7	753664	0	753664	2004-02-06-16.42.22.783143
10	0	0	0	2004-02-06-16.42.28.198449
12	32768	0	32768	2004-02-06-16.42.33.210138
13	-32768	4096	-28672	2004-02-06-16.42.35.142229
14	131072	0	131072	2004-02-06-16.42.36.616209
15	-57344	0	-57344	2004-02-06-16.42.39.302391
18	53248	0	53248	2004-02-06-16.42.44.261540
19	65536	0	65536	2004-02-06-16.42.46.729722

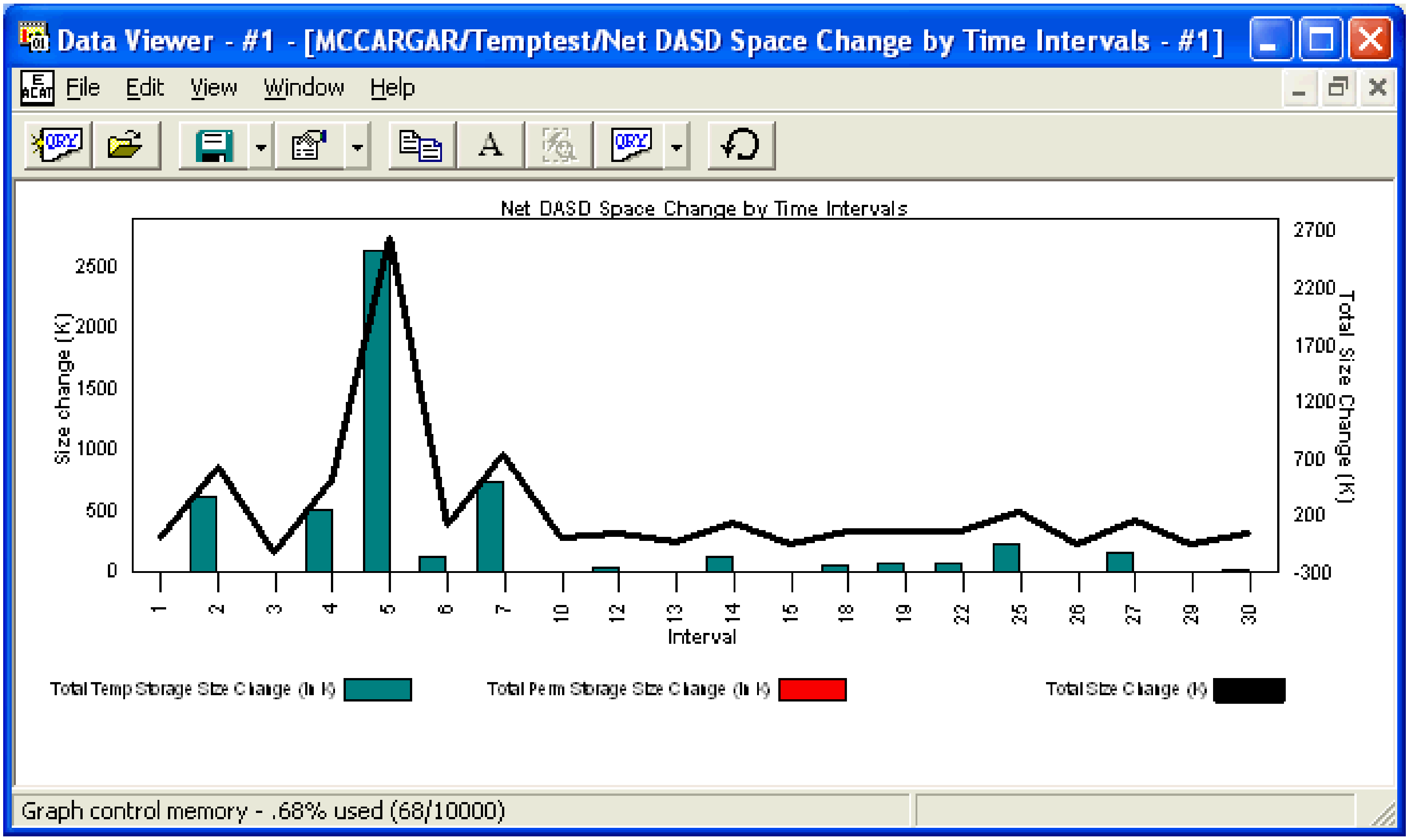
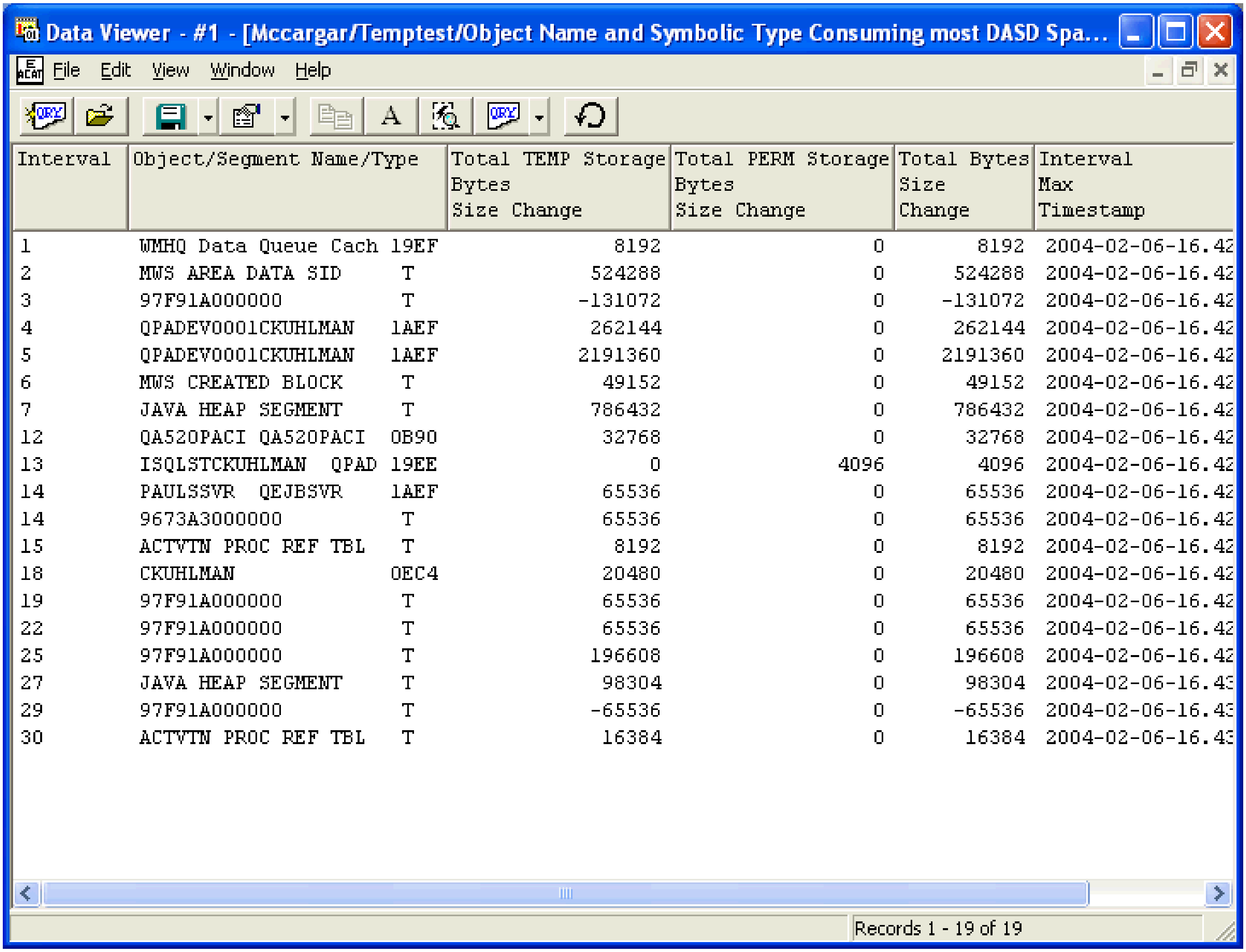

Data Viewer - #1 - [MCCARGAR/Temptest/Net DASD Space Change by Time Intervals - #1]
File
Edit
View
Window
Help
QRY
A
QRY
Net DASD Space Change by Time Intervals
Size change (K)
2500
2000
1500
1000
500
0
Total Size Change (K)
2700
2200
1700
1200
700
200
-300
1
2
3
4
5
6
7
10
12
13
14
15
18
19
22
25
26
27
29
30
Interval
Total Temp Storage Size Change (In K)
Total Perm Storage Size Change (In K)
Total Size Change (K)
Graph control memory - .68% used (68/10000)

Table of Contents | Previous | Next

5.6.6 Object Name and Symbolic Type Consuming most DASD Space per Interval

Description:

Example:

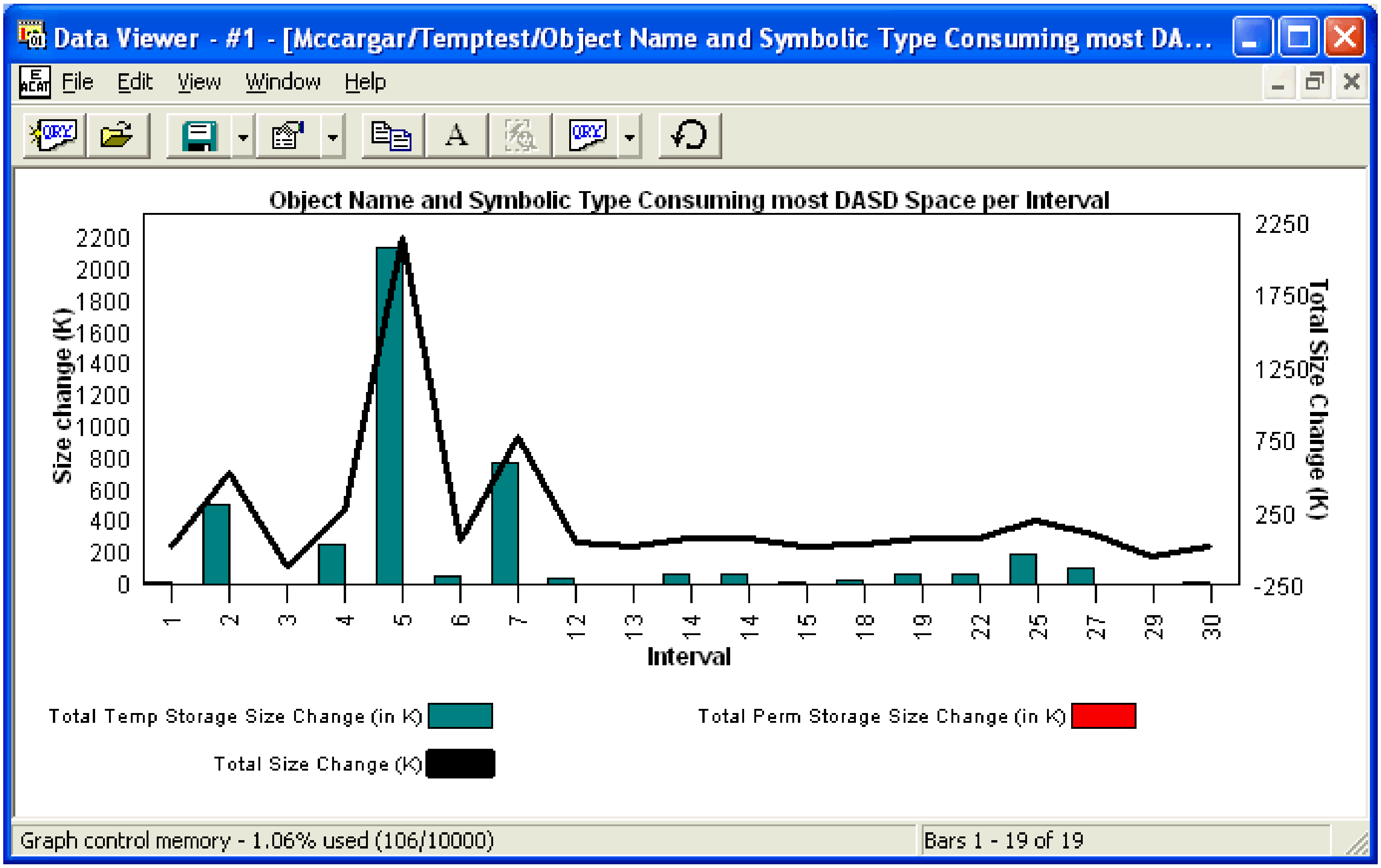


Data Viewer - #1 - [Mccargar/Temptest/Object Name and Symbolic Type Consuming most DASD Spa...
File
Edit
View
Window
Help

Interval	Object/Segment Name/Type	Total TEMP Storage Bytes Size Change	Total PERM Storage Bytes Size Change	Total Bytes Size Change	Interval Max Timestamp
1	WMHQ Data Queue Cach 19EF	8192	0	8192	2004-02-06-16.42
2	MWS AREA DATA SID T	524288	0	524288	2004-02-06-16.42
3	97F91A000000 T	-131072	0	-131072	2004-02-06-16.42
4	QPADEV0001CKUHLMAN 1AEF	262144	0	262144	2004-02-06-16.42
5	QPADEV0001CKUHLMAN 1AEF	2191360	0	2191360	2004-02-06-16.42
6	MWS CREATED BLOCK T	49152	0	49152	2004-02-06-16.42
7	JAVA HEAP SEGMENT T	786432	0	786432	2004-02-06-16.42
12	QA520PACI QA520PACI 0B90	32768	0	32768	2004-02-06-16.42
13	ISQLSTCKUHLMAN QPAD 19EE	0	4096	4096	2004-02-06-16.42
14	PAULSSVR QEJBSVR 1AEF	65536	0	65536	2004-02-06-16.42
14	9673A3000000 T	65536	0	65536	2004-02-06-16.42
15	ACTVTN PROC REF TBL T	8192	0	8192	2004-02-06-16.42
18	CKUHLMAN 0EC4	20480	0	20480	2004-02-06-16.42
19	97F91A000000 T	65536	0	65536	2004-02-06-16.42
22	97F91A000000 T	65536	0	65536	2004-02-06-16.42
25	97F91A000000 T	196608	0	196608	2004-02-06-16.42
27	JAVA HEAP SEGMENT T	98304	0	98304	2004-02-06-16.43
29	97F91A000000 T	-65536	0	-65536	2004-02-06-16.43
30	ACTVTN PROC REF TBL T	16384	0	16384	2004-02-06-16.43

Records 1 - 19 of 19

5.6.6 Object Name and Symbolic Type Consuming most DASD Space per Interval


Data Viewer - #1 - [Mccargar/Temptest/Object Name and Symbolic Type Consuming most DA...
File
Edit
View
Window
Help
QRY
A
QRY
Object Name and Symbolic Type Consuming most DASD Space per Interval
Size change (K)
2200
2000
1800
1600
1400
1200
1000
800
600
400
200
0
Total Size Change (K)
2250
1750
1250
750
250
-250
1
2
3
4
5
6
7
12
13
14
14
15
18
19
22
25
27
29
30
Interval
Total Temp Storage Size Change (in K) 🟩
Total Perm Storage Size Change (in K) 🟥
Total Size Change (K) ■

Graph control memory - 1.06% used (106/10000)

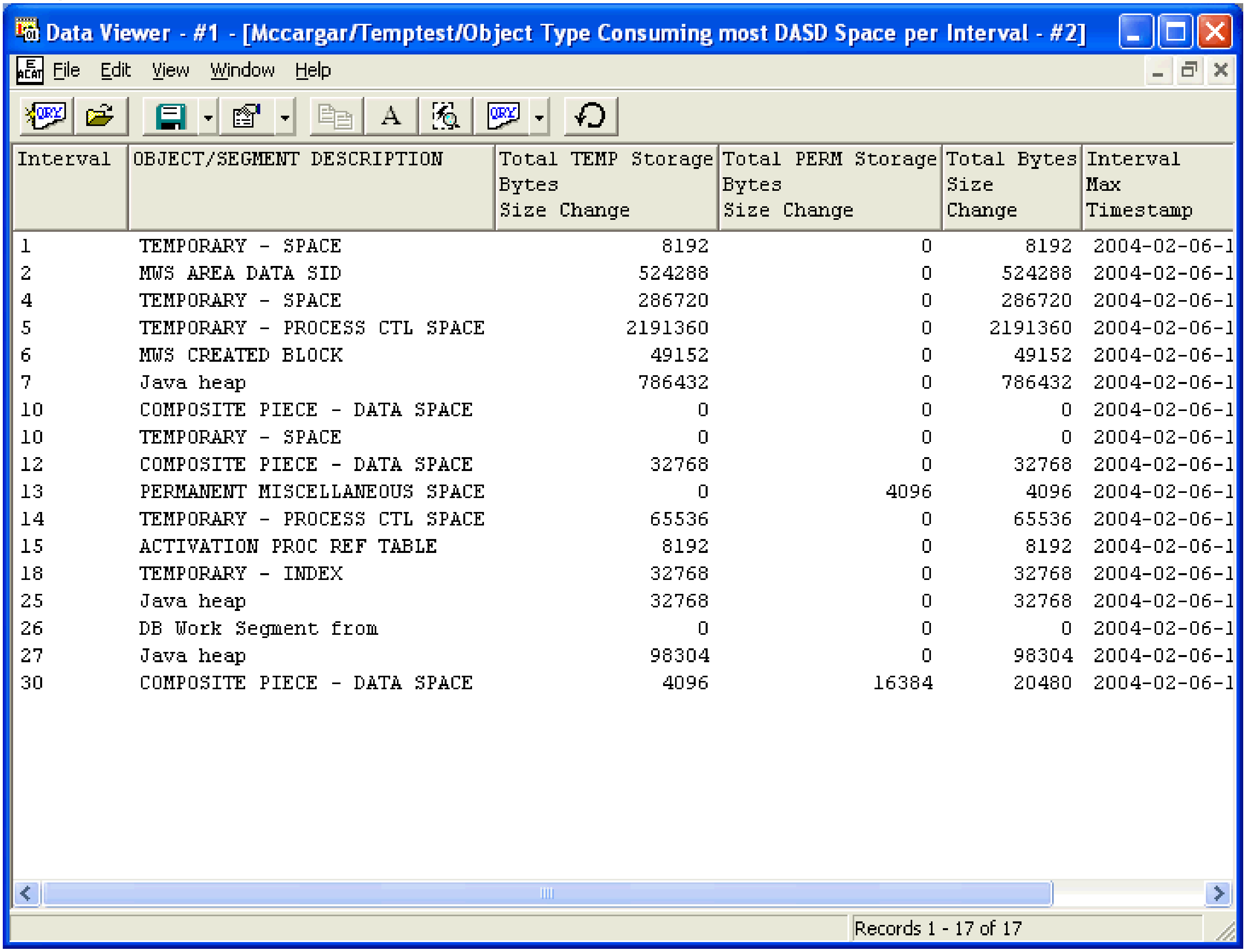
Bars 1 - 19 of 19

Table of Contents | Previous | Next

5.6.7 Object Type Consuming most DASD Space per Interval

Description: This report enables the user to know the Dasd space changes for each interval of an analysis and the amount of change and the object/segment type causing it. Within the graphs the user can click on any bar (temporary or permanent space change) and get information on the interval, space change, the object/segment type causing it, and the date-time stamp.

Example:


Data Viewer - #1 - [Mccargar/Temptest/Object Type Consuming most DASD Space per Interval - #2]
File
Edit
View
Window
Help
QRY
QRY
A

Interval	OBJECT/SEGMENT DESCRIPTION	Total TEMP Storage Bytes Size Change	Total PERM Storage Bytes Size Change	Total Bytes Size Change	Interval Max Timestamp
1	TEMPORARY - SPACE	8192	0	8192	2004-02-06-1
2	MWS AREA DATA SID	524288	0	524288	2004-02-06-1
4	TEMPORARY - SPACE	286720	0	286720	2004-02-06-1
5	TEMPORARY - PROCESS CTL SPACE	2191360	0	2191360	2004-02-06-1
6	MWS CREATED BLOCK	49152	0	49152	2004-02-06-1
7	Java heap	786432	0	786432	2004-02-06-1
10	COMPOSITE PIECE - DATA SPACE	0	0	0	2004-02-06-1
10	TEMPORARY - SPACE	0	0	0	2004-02-06-1
12	COMPOSITE PIECE - DATA SPACE	32768	0	32768	2004-02-06-1
13	PERMANENT MISCELLANEOUS SPACE	0	4096	4096	2004-02-06-1
14	TEMPORARY - PROCESS CTL SPACE	65536	0	65536	2004-02-06-1
15	ACTIVATION PROC REF TABLE	8192	0	8192	2004-02-06-1
18	TEMPORARY - INDEX	32768	0	32768	2004-02-06-1
25	Java heap	32768	0	32768	2004-02-06-1
26	DB Work Segment from	0	0	0	2004-02-06-1
27	Java heap	98304	0	98304	2004-02-06-1
30	COMPOSITE PIECE - DATA SPACE	4096	16384	20480	2004-02-06-1

Records 1 - 17 of 17

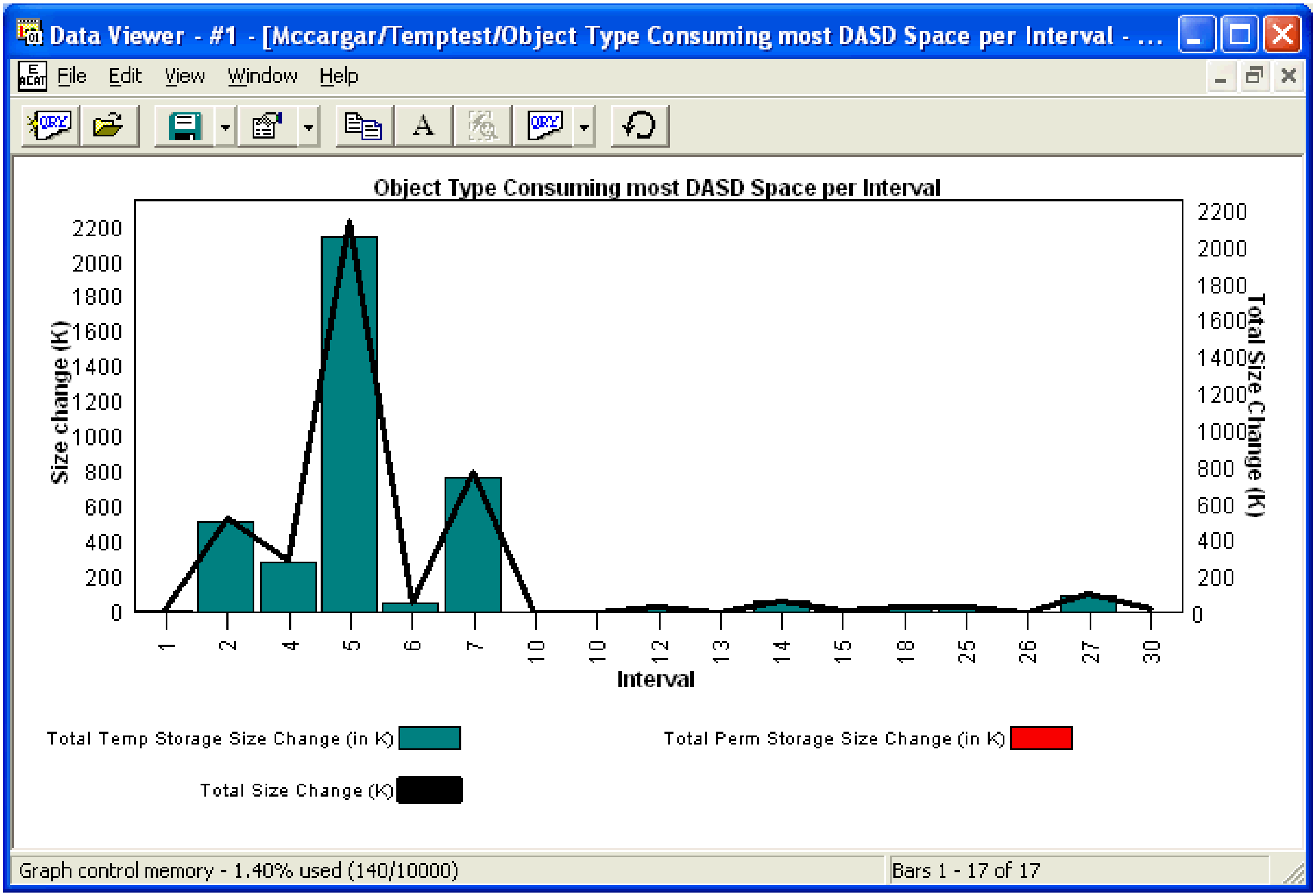

Data Viewer - #1 - [Mccargar/Temptest/Object Type Consuming most DASD Space per Interval - ...
File Edit View Window Help
Object Type Consuming most DASD Space per Interval
Size change (K)
2200
2000
1800
1600
1400
1200
1000
800
600
400
200
0
Total Size Change (K)
Interval
1
2
4
5
6
7
10
10
12
13
14
15
18
25
26
27
30
Total Temp Storage Size Change (in K) 🟩
Total Perm Storage Size Change (in K) 🟥
Total Size Change (K) ■
Graph control memory - 1.40% used (140/10000)
Bars 1 - 17 of 17

Table of Contents Previous Next

5.7 Database file full opens/closes reports

This analysis traces all full opens and closes showing the causing job/thread, and file/member name.

If MIENTRY and MIEXIT events are included in the collection, the analysis will also include the causing program name, elapsed time, and cpu time of each open/close. This analysis currently only supports local files (not DDM).

Table of Contents | Previous | Next

5.7.1 Local DB File Full Closes by File-Library within Interval

Description:

Example:

Table of Contents | Previous | Next

5.7.2 Local DB File Full Closes by File-Library within Job-Thread - No Intervals

Description:

Example:

Table of Contents | Previous | Next

5.7.3 Local DB File Full Closes by File-Library within Program - No Intervals

Description:

Example:

Table of Contents Previous Next

5.7.4 Local DB File Full Closes by Job within Interval

Description:

Example:

Table of Contents | Previous | Next

5.7.5 Local DB File Full Closes by Job-Thread within File-Library - No Intervals

Description:

Example:

Table of Contents | Previous | Next

5.7.6 Local DB File Full Closes by Job-Thread within Program - No Intervals

Description:

Example:

Table of Contents Previous Next

5.7.7 Local DB File Full Closes by Program within File-Library - No Intervals

Description:

Example:

Table of Contents | Previous | Next

5.7.8 Local DB File Full Closes by Program within Interval

Description:

Example:

Table of Contents | Previous | Next

5.7.9 Local DB File Full Closes Program within Job-Thread - No Intervals

Description:

Example:

Table of Contents | Previous | Next

5.7.10 Local DB File Full Opens by File-Library within Interval

Description:

Example:

Table of Contents | Previous | Next

5.7.11 Local DB File Full Opens by File-Library within Job-Thread - No Intervals

Description:

Example:

Table of Contents | Previous | Next

5.7.12 Local DB File Full Opens by File-Library within Program - No Intervals

Description:

Example:

Table of Contents Previous Next

5.7.13 Local DB File Full Opens by Job within Interval

Description:

Example:

Table of Contents | Previous | Next

5.7.14 Local DB File Full Opens by Job-Thread within File-Library - No Intervals

Description:

Example:

Table of Contents Previous Next

5.7.15 Local DB File Full Opens by Job-Thread within Program - No Intervals

Description:

Example:

Table of Contents | Previous | Next

5.7.16 Local DB File Full Opens by Program within File-Library - No Intervals

Description:

Example:

Table of Contents | Previous | Next

5.7.17 Local DB File Full Opens by Program within Interval

Description:

Example:

Table of Contents | Previous | Next

5.7.18 Local DB File Full Opens by Program within Job-Thread - No Intervals

Description:

Example:

Table of Contents | Previous | Next

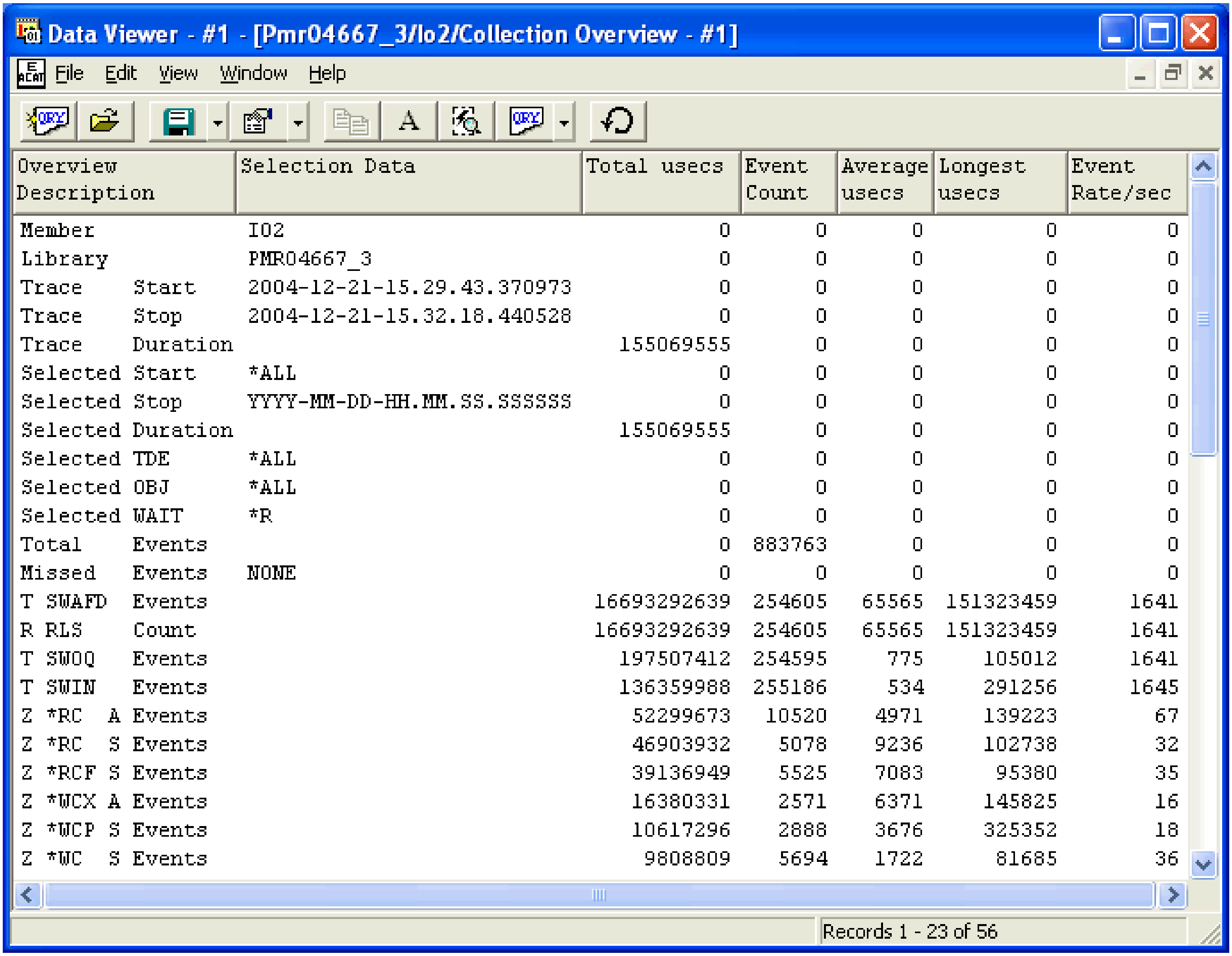
5.8 Task switch data analysis reports

Table of Contents | Previous | Next

5.8.1 Collection Overview

Description:

Example:


Data Viewer - #1 - [Pmr04667_3/Io2/Collection Overview - #1]
File
Edit
View
Window
Help

Overview Description	Selection Data	Total usecs	Event Count	Average usecs	Longest usecs	Event Rate/sec
Member	I02	0	0	0	0	0
Library	PMR04667_3	0	0	0	0	0
Trace Start	2004-12-21-15.29.43.370973	0	0	0	0	0
Trace Stop	2004-12-21-15.32.18.440528	0	0	0	0	0
Trace Duration		155069555	0	0	0	0
Selected Start	*ALL	0	0	0	0	0
Selected Stop	YYYY-MM-DD-HH.MM.SS.SSSSSS	0	0	0	0	0
Selected Duration		155069555	0	0	0	0
Selected TDE	*ALL	0	0	0	0	0
Selected OBJ	*ALL	0	0	0	0	0
Selected WAIT	*R	0	0	0	0	0
Total Events		0	883763	0	0	0
Missed Events	NONE	0	0	0	0	0
T SWAFD Events		16693292639	254605	65565	151323459	1641
R RLS Count		16693292639	254605	65565	151323459	1641
T SWOQ Events		197507412	254595	775	105012	1641
T SWIN Events		136359988	255186	534	291256	1645
Z *RC A Events		52299673	10520	4971	139223	67
Z *RC S Events		46903932	5078	9236	102738	32
Z *RCF S Events		39136949	5525	7083	95380	35
Z *WCX A Events		16380331	2571	6371	145825	16
Z *WCP S Events		10617296	2888	3676	325352	18
Z *WC S Events		9808809	5694	1722	81685	36

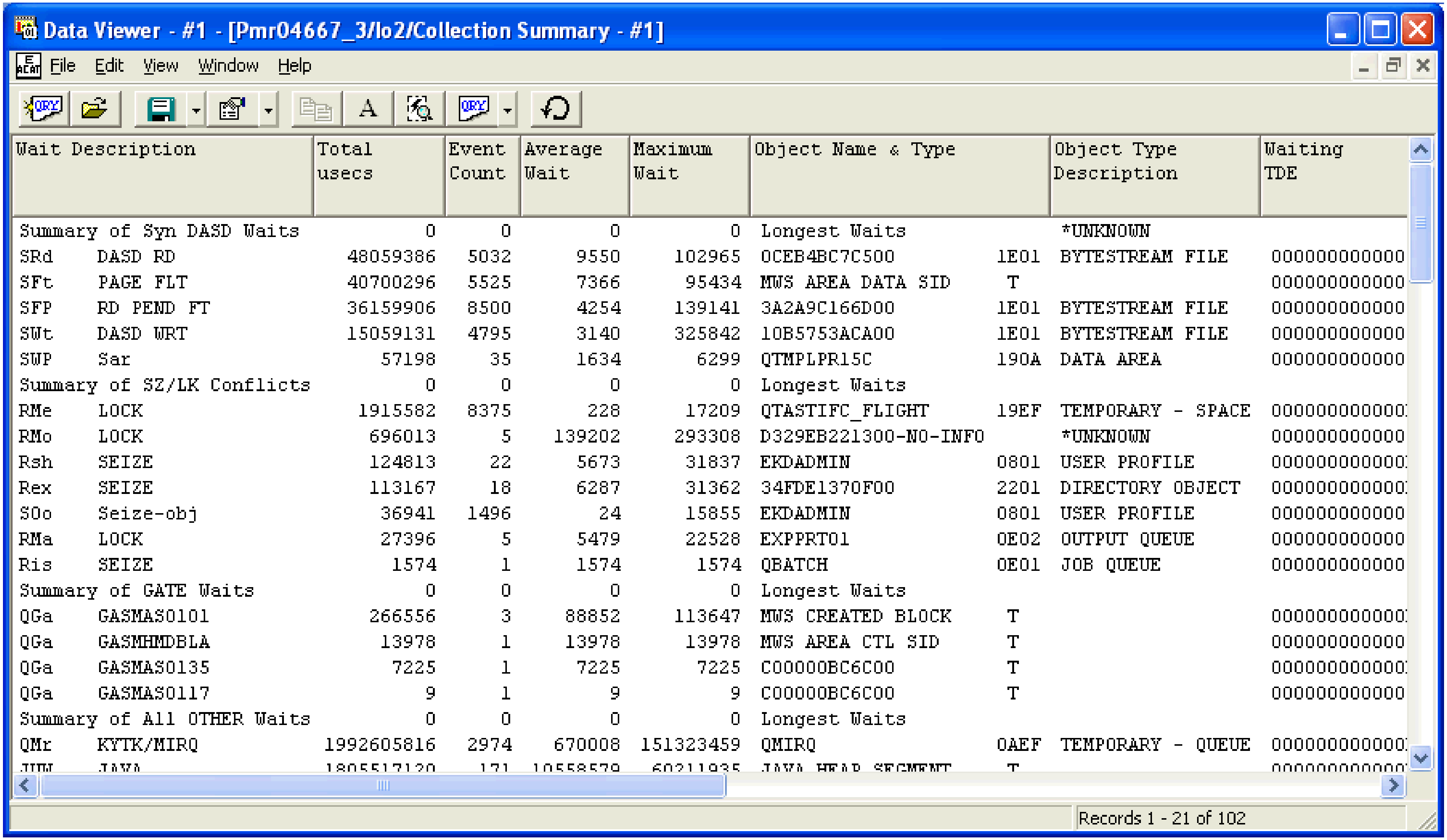
Records 1 - 23 of 56

Table of Contents | Previous | Next

5.8.2 Collection Summary

Description:

Example:



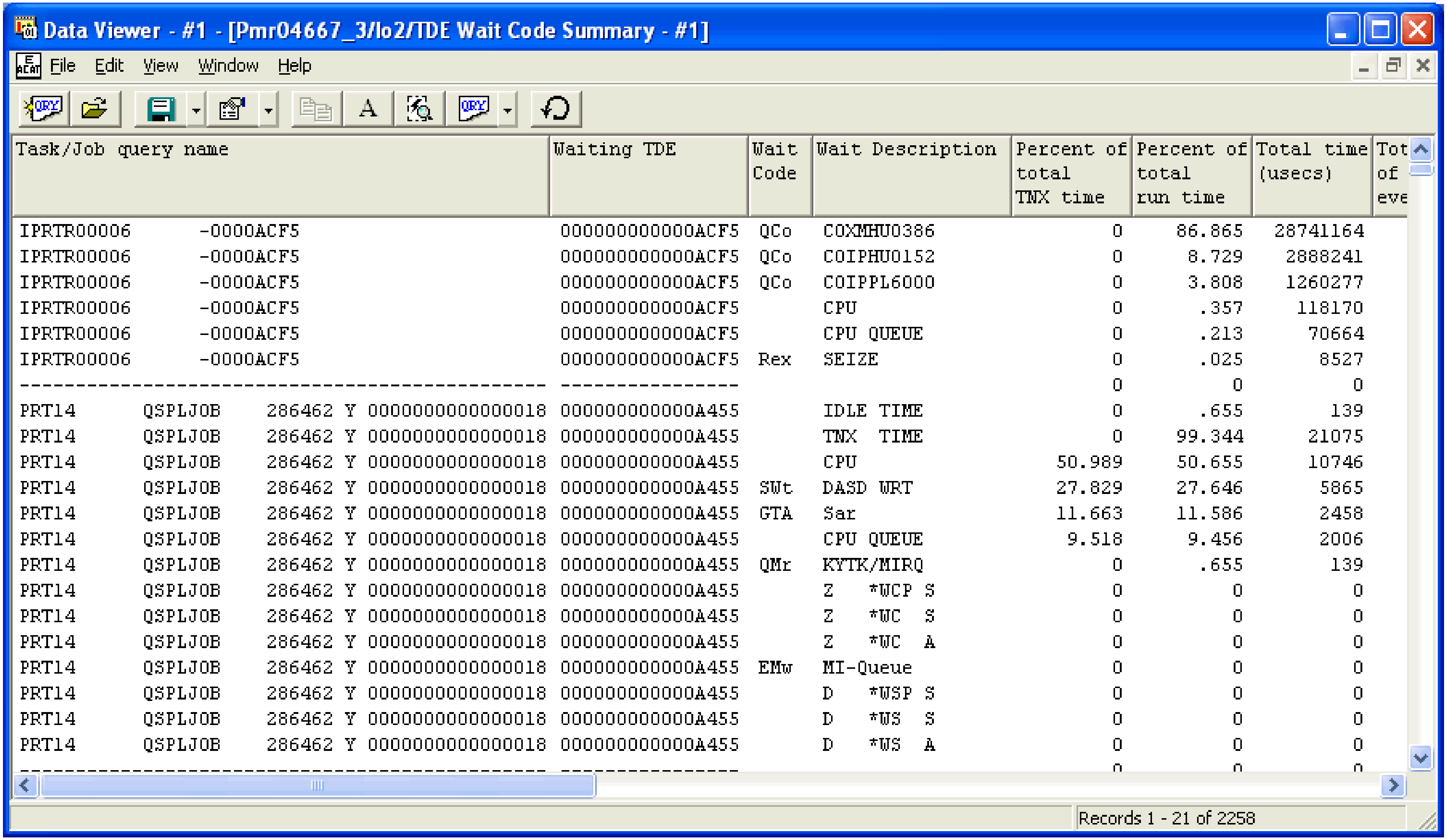
Data Viewer - #1 - [Pmr04667_3/lo2/Collection Summary - #1]
File Edit View Window Help
QRY
QRY
Records 1 - 21 of 102

Wait Description		Total usecs	Event Count	Average Wait	Maximum Wait	Object Name & Type		Object Type Description	Waiting TDE
Summary of Syn DASD Waits		0	0	0	0	Longest Waits		*UNKNOWN	
SRd	DASD RD	48059386	5032	9550	102965	0CEB4BC7C500	1E01	BYTESTREAM FILE	000000000000
SFt	PAGE FLT	40700296	5525	7366	95434	MWS AREA DATA SID	T		000000000000
SFP	RD PEND FT	36159906	8500	4254	139141	3A2A9C166D00	1E01	BYTESTREAM FILE	000000000000
SWt	DASD WRT	15059131	4795	3140	325842	10B5753ACA00	1E01	BYTESTREAM FILE	000000000000
SWP	Sar	57198	35	1634	6299	QTMPLPR15C	190A	DATA AREA	000000000000
Summary of SZ/LK Conflicts		0	0	0	0	Longest Waits			
RMe	LOCK	1915582	8375	228	17209	QTASTIFC_FLIGHT	19EF	TEMPORARY - SPACE	000000000000
RMo	LOCK	696013	5	139202	293308	D329EB221300-NO-INFO		*UNKNOWN	000000000000
Rsh	SEIZE	124813	22	5673	31837	EKDADMIN	0801	USER PROFILE	000000000000
Rex	SEIZE	113167	18	6287	31362	34FDE1370F00	2201	DIRECTORY OBJECT	000000000000
SOo	Seize-obj	36941	1496	24	15855	EKDADMIN	0801	USER PROFILE	000000000000
RMa	LOCK	27396	5	5479	22528	EXPPRT01	0E02	OUTPUT QUEUE	000000000000
Ris	SEIZE	1574	1	1574	1574	QBATCH	0E01	JOB QUEUE	000000000000
Summary of GATE Waits		0	0	0	0	Longest Waits			
QGa	GASMAS0101	266556	3	88852	113647	MWS CREATED BLOCK	T		000000000000
QGa	GASMHMDBLA	13978	1	13978	13978	MWS AREA CTL SID	T		000000000000
QGa	GASMAS0135	7225	1	7225	7225	C00000BC6C00	T		000000000000
QGa	GASMAS0117	9	1	9	9	C00000BC6C00	T		000000000000
Summary of All OTHER Waits		0	0	0	0	Longest Waits			
QMr	KYTK/MIRQ	1992605816	2974	670008	151323459	QMIRQ	0AEF	TEMPORARY - QUEUE	000000000000
JUW	JAVA	1805517120	171	10558579	60211935	JAVA HEAP SEGMENT	T		000000000000

5.8.3 TDE Wait Code Summary

Description:

Example:


Data Viewer - #1 - [Pmr04667_3/lo2/TDE Wait Code Summary - #1]
File
Edit
View
Window
Help
QRY
Records 1 - 21 of 2258

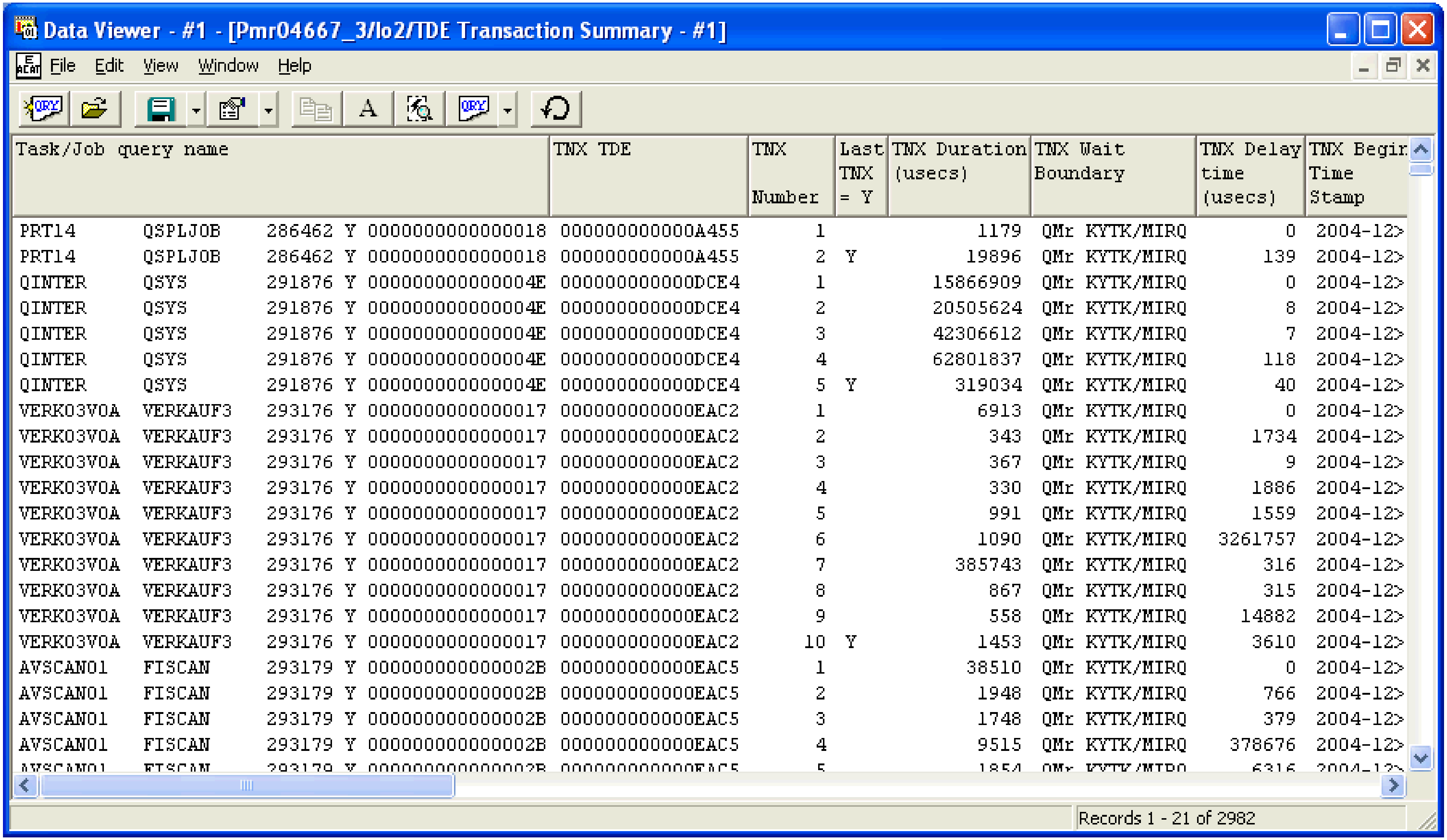
Task/Job query name	Waiting TDE	Wait Code	Wait Description	Percent of total TNX time	Percent of total run time	Total time (usecs)	Tot of eve
IPRTR00006 -0000ACF5	000000000000ACF5	QCo	COXMHU0386	0	86.865	28741164	
IPRTR00006 -0000ACF5	000000000000ACF5	QCo	COIPHU0152	0	8.729	2888241	
IPRTR00006 -0000ACF5	000000000000ACF5	QCo	COIPPL6000	0	3.808	1260277	
IPRTR00006 -0000ACF5	000000000000ACF5		CPU	0	.357	118170	
IPRTR00006 -0000ACF5	000000000000ACF5		CPU QUEUE	0	.213	70664	
IPRTR00006 -0000ACF5	000000000000ACF5	Rex	SEIZE	0	.025	8527	
--	----------------			0	0	0	
PRT14 QSPLJOB 286462 Y 0000000000000018	000000000000A455		IDLE TIME	0	.655	139	
PRT14 QSPLJOB 286462 Y 0000000000000018	000000000000A455		TNX TIME	0	99.344	21075	
PRT14 QSPLJOB 286462 Y 0000000000000018	000000000000A455		CPU	50.989	50.655	10746	
PRT14 QSPLJOB 286462 Y 0000000000000018	000000000000A455	SWt	DASD WRT	27.829	27.646	5865	
PRT14 QSPLJOB 286462 Y 0000000000000018	000000000000A455	GTA	Sar	11.663	11.586	2458	
PRT14 QSPLJOB 286462 Y 0000000000000018	000000000000A455		CPU QUEUE	9.518	9.456	2006	
PRT14 QSPLJOB 286462 Y 0000000000000018	000000000000A455	QMr	KYTK/MIRQ	0	.655	139	
PRT14 QSPLJOB 286462 Y 0000000000000018	000000000000A455		Z *WCP S	0	0	0	
PRT14 QSPLJOB 286462 Y 0000000000000018	000000000000A455		Z *WC S	0	0	0	
PRT14 QSPLJOB 286462 Y 0000000000000018	000000000000A455		Z *WC A	0	0	0	
PRT14 QSPLJOB 286462 Y 0000000000000018	000000000000A455	EMw	MI-Queue	0	0	0	
PRT14 QSPLJOB 286462 Y 0000000000000018	000000000000A455		D *WSP S	0	0	0	
PRT14 QSPLJOB 286462 Y 0000000000000018	000000000000A455		D *WS S	0	0	0	
PRT14 QSPLJOB 286462 Y 0000000000000018	000000000000A455		D *WS A	0	0	0	
--	----------------			0	0	0	

Table of Contents Previous Next

5.8.4 TDE Transaction Summary

Description:

Example:

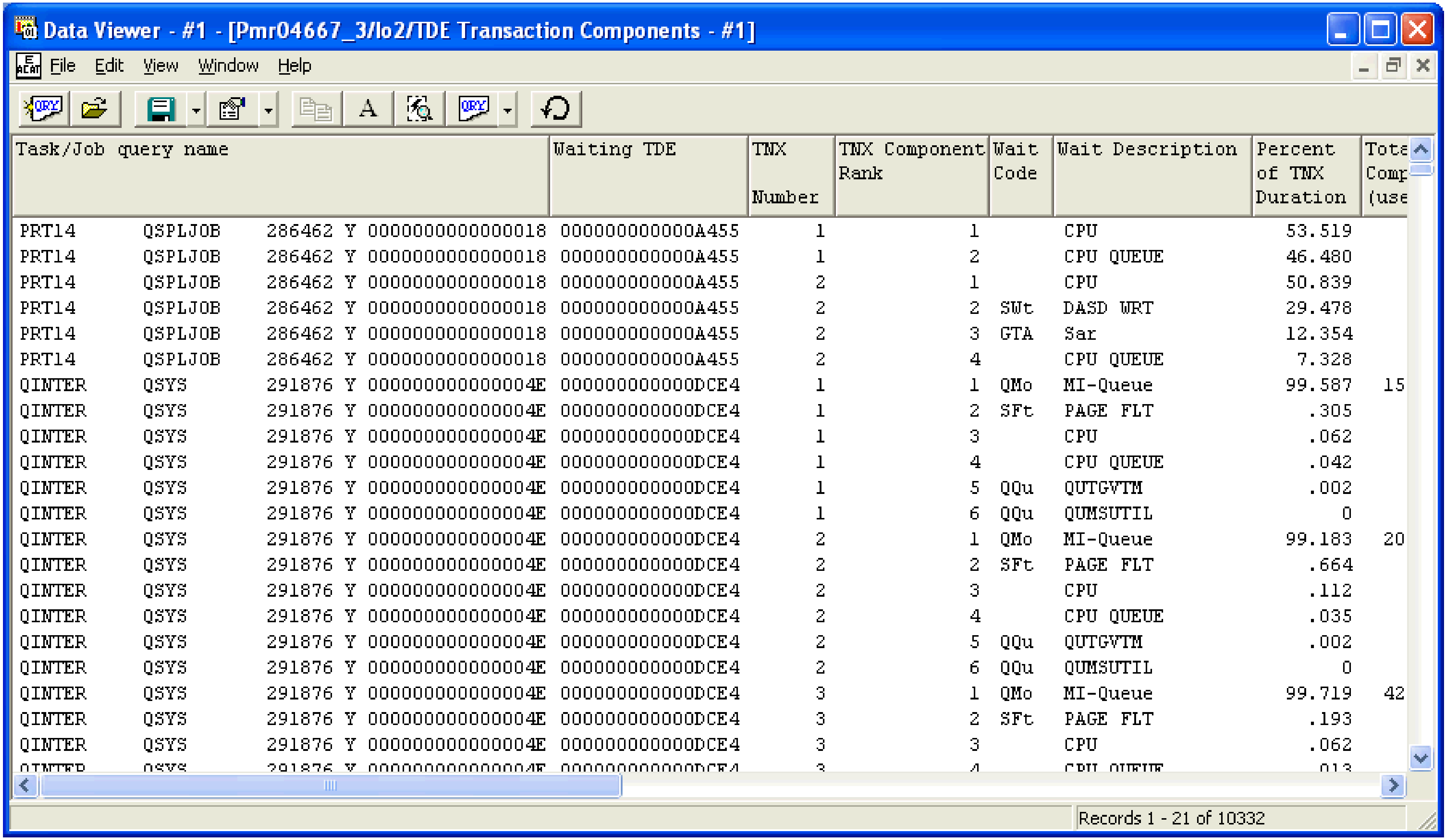

Data Viewer - #1 - [Pmr04667_3/lo2/TDE Transaction Summary - #1]
File
Edit
View
Window
Help
QRY
QRY
A
Records 1 - 21 of 2982

Task/Job query name					TNX TDE	TNX Number	Last TNX = Y	TNX Duration (usecs)	TNX Wait Boundary		TNX Delay time (usecs)	TNX Begin Time Stamp
PRT14	QSPLJOB	286462	Y	0000000000000018	000000000000A455	1		1179	QMr	KYTK/MIRQ	0	2004-12>
PRT14	QSPLJOB	286462	Y	0000000000000018	000000000000A455	2	Y	19896	QMr	KYTK/MIRQ	139	2004-12>
QINTER	QSYS	291876	Y	000000000000004E	000000000000DCE4	1		15866909	QMr	KYTK/MIRQ	0	2004-12>
QINTER	QSYS	291876	Y	000000000000004E	000000000000DCE4	2		20505624	QMr	KYTK/MIRQ	8	2004-12>
QINTER	QSYS	291876	Y	000000000000004E	000000000000DCE4	3		42306612	QMr	KYTK/MIRQ	7	2004-12>
QINTER	QSYS	291876	Y	000000000000004E	000000000000DCE4	4		62801837	QMr	KYTK/MIRQ	118	2004-12>
QINTER	QSYS	291876	Y	000000000000004E	000000000000DCE4	5	Y	319034	QMr	KYTK/MIRQ	40	2004-12>
VERK03V0A	VERKAUF3	293176	Y	0000000000000017	000000000000EAC2	1		6913	QMr	KYTK/MIRQ	0	2004-12>
VERK03V0A	VERKAUF3	293176	Y	0000000000000017	000000000000EAC2	2		343	QMr	KYTK/MIRQ	1734	2004-12>
VERK03V0A	VERKAUF3	293176	Y	0000000000000017	000000000000EAC2	3		367	QMr	KYTK/MIRQ	9	2004-12>
VERK03V0A	VERKAUF3	293176	Y	0000000000000017	000000000000EAC2	4		330	QMr	KYTK/MIRQ	1886	2004-12>
VERK03V0A	VERKAUF3	293176	Y	0000000000000017	000000000000EAC2	5		991	QMr	KYTK/MIRQ	1559	2004-12>
VERK03V0A	VERKAUF3	293176	Y	0000000000000017	000000000000EAC2	6		1090	QMr	KYTK/MIRQ	3261757	2004-12>
VERK03V0A	VERKAUF3	293176	Y	0000000000000017	000000000000EAC2	7		385743	QMr	KYTK/MIRQ	316	2004-12>
VERK03V0A	VERKAUF3	293176	Y	0000000000000017	000000000000EAC2	8		867	QMr	KYTK/MIRQ	315	2004-12>
VERK03V0A	VERKAUF3	293176	Y	0000000000000017	000000000000EAC2	9		558	QMr	KYTK/MIRQ	14882	2004-12>
VERK03V0A	VERKAUF3	293176	Y	0000000000000017	000000000000EAC2	10	Y	1453	QMr	KYTK/MIRQ	3610	2004-12>
AVSCAN01	FISCAN	293179	Y	000000000000002B	000000000000EAC5	1		38510	QMr	KYTK/MIRQ	0	2004-12>
AVSCAN01	FISCAN	293179	Y	000000000000002B	000000000000EAC5	2		1948	QMr	KYTK/MIRQ	766	2004-12>
AVSCAN01	FISCAN	293179	Y	000000000000002B	000000000000EAC5	3		1748	QMr	KYTK/MIRQ	379	2004-12>
AVSCAN01	FISCAN	293179	Y	000000000000002B	000000000000EAC5	4		9515	QMr	KYTK/MIRQ	378676	2004-12>
AVSCAN01	FISCAN	293179	Y	000000000000002B	000000000000EAC5	5		1854	QMr	KYTK/MIRQ	6316	2004-12>

5.8.5 TDE Transaction Components

Description:

Example:



Data Viewer - #1 - [Pmr04667_3/lo2/TDE Transaction Components - #1]
File Edit View Window Help
QRY
QRY

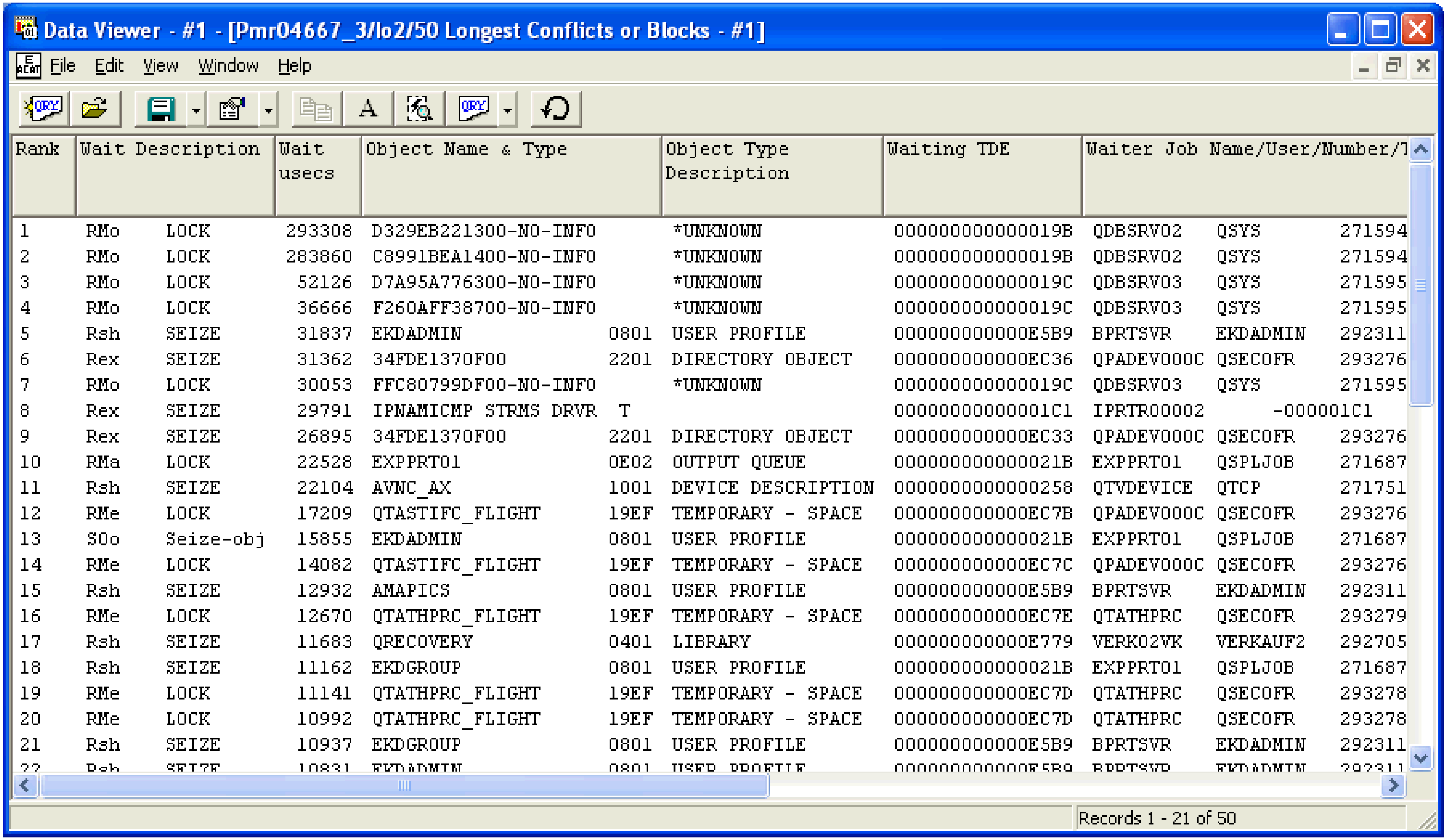
Task/Job query name	Waiting TDE	TNX Number	TNX Component Rank	Wait Code	Wait Description	Percent of TNX Duration	Tota Comp (use
PRT14 QSPLJOB 286462 Y 0000000000000018	000000000000A455	1	1		CPU	53.519	
PRT14 QSPLJOB 286462 Y 0000000000000018	000000000000A455	1	2		CPU QUEUE	46.480	
PRT14 QSPLJOB 286462 Y 0000000000000018	000000000000A455	2	1		CPU	50.839	
PRT14 QSPLJOB 286462 Y 0000000000000018	000000000000A455	2	2	SWt	DASD WRT	29.478	
PRT14 QSPLJOB 286462 Y 0000000000000018	000000000000A455	2	3	GTA	Sar	12.354	
PRT14 QSPLJOB 286462 Y 0000000000000018	000000000000A455	2	4		CPU QUEUE	7.328	
QINTER QSYS 291876 Y 000000000000004E	000000000000DCE4	1	1	QMo	MI-Queue	99.587	15
QINTER QSYS 291876 Y 000000000000004E	000000000000DCE4	1	2	SFt	PAGE FLT	.305	
QINTER QSYS 291876 Y 000000000000004E	000000000000DCE4	1	3		CPU	.062	
QINTER QSYS 291876 Y 000000000000004E	000000000000DCE4	1	4		CPU QUEUE	.042	
QINTER QSYS 291876 Y 000000000000004E	000000000000DCE4	1	5	QQu	QUTGVTM	.002	
QINTER QSYS 291876 Y 000000000000004E	000000000000DCE4	1	6	QQu	QUMSUTIL	0	
QINTER QSYS 291876 Y 000000000000004E	000000000000DCE4	2	1	QMo	MI-Queue	99.183	20
QINTER QSYS 291876 Y 000000000000004E	000000000000DCE4	2	2	SFt	PAGE FLT	.664	
QINTER QSYS 291876 Y 000000000000004E	000000000000DCE4	2	3		CPU	.112	
QINTER QSYS 291876 Y 000000000000004E	000000000000DCE4	2	4		CPU QUEUE	.035	
QINTER QSYS 291876 Y 000000000000004E	000000000000DCE4	2	5	QQu	QUTGVTM	.002	
QINTER QSYS 291876 Y 000000000000004E	000000000000DCE4	2	6	QQu	QUMSUTIL	0	
QINTER QSYS 291876 Y 000000000000004E	000000000000DCE4	3	1	QMo	MI-Queue	99.719	42
QINTER QSYS 291876 Y 000000000000004E	000000000000DCE4	3	2	SFt	PAGE FLT	.193	
QINTER QSYS 291876 Y 000000000000004E	000000000000DCE4	3	3		CPU	.062	
QINTER QSYS 291876 Y 000000000000004E	000000000000DCE4	3	4		CPU QUEUE	.013	

Records 1 - 21 of 10332

5.8.6 50 Longest Conflicts or Blocks

Description:

Example:



Data Viewer - #1 - [Pmr04667_3/lo2/50 Longest Conflicts or Blocks - #1]
File
Edit
View
Window
Help

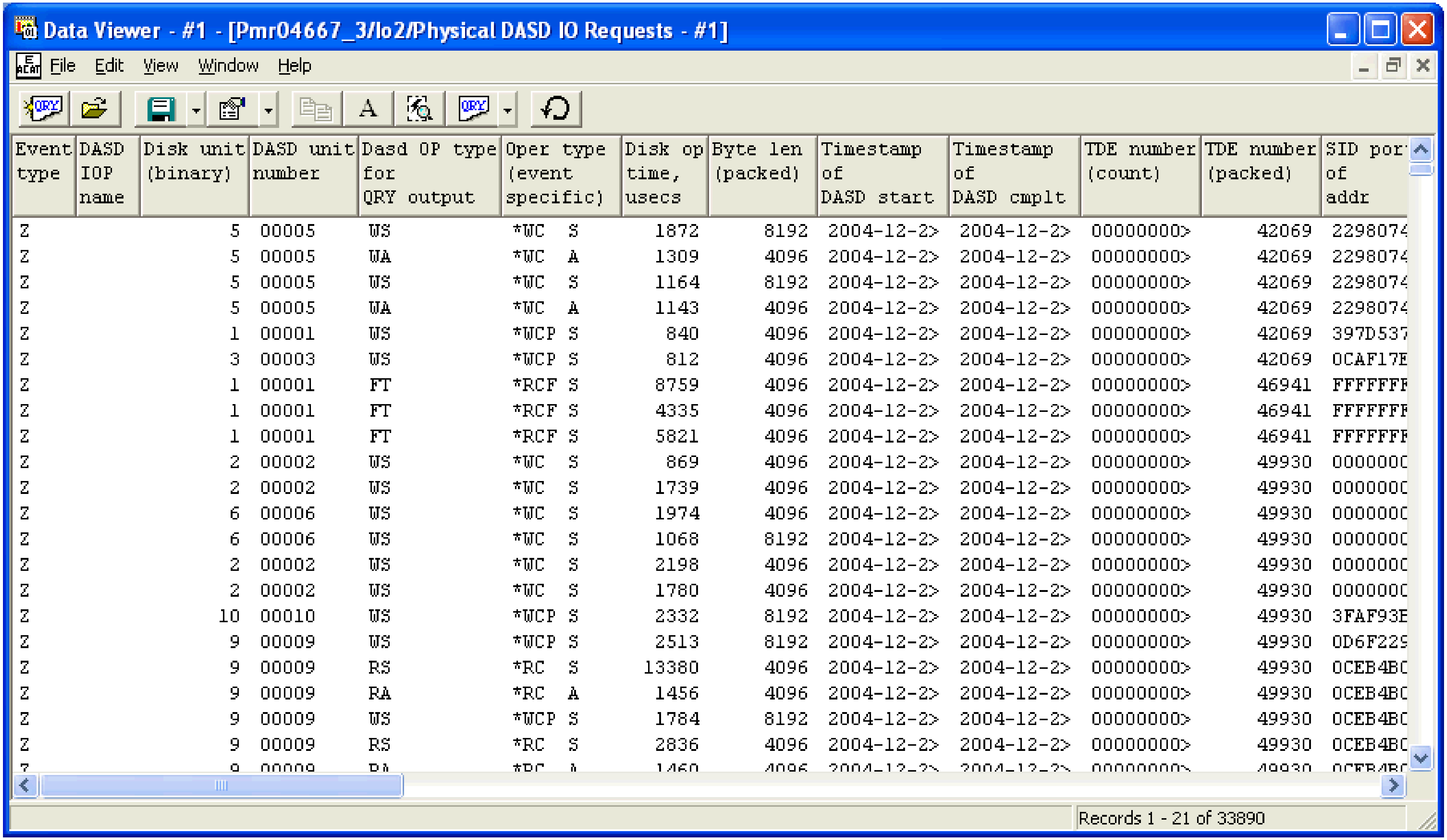
Rank	Wait Description		Wait usecs	Object Name & Type		Object Type Description	Waiting TDE	Waiter Job Name/User/Number/T		
1	RMo	LOCK	293308	D329EB221300-NO-INFO		*UNKNOWN	000000000000019B	QDBSRV02	QSYS	271594
2	RMo	LOCK	283860	C8991BEA1400-NO-INFO		*UNKNOWN	000000000000019B	QDBSRV02	QSYS	271594
3	RMo	LOCK	52126	D7A95A776300-NO-INFO		*UNKNOWN	000000000000019C	QDBSRV03	QSYS	271595
4	RMo	LOCK	36666	F260AFF38700-NO-INFO		*UNKNOWN	000000000000019C	QDBSRV03	QSYS	271595
5	Rsh	SEIZE	31837	EKDADMIN	0801	USER PROFILE	000000000000E5B9	BPRTSVR	EKDADMIN	292311
6	Rex	SEIZE	31362	34FDE1370F00	2201	DIRECTORY OBJECT	000000000000EC36	QPADEV000C	QSECOFR	293276
7	RMo	LOCK	30053	FFC80799DF00-NO-INFO		*UNKNOWN	000000000000019C	QDBSRV03	QSYS	271595
8	Rex	SEIZE	29791	IPNAMICMP STRMS DRVR	T		00000000000001C1	IPRTR00002	-000001C1	
9	Rex	SEIZE	26895	34FDE1370F00	2201	DIRECTORY OBJECT	000000000000EC33	QPADEV000C	QSECOFR	293276
10	RMa	LOCK	22528	EXPPRT01	0E02	OUTPUT QUEUE	000000000000021B	EXPPRT01	QSPLJOB	271687
11	Rsh	SEIZE	22104	AVNC_AX	1001	DEVICE DESCRIPTION	0000000000000258	QTVDEVICE	QTCP	271751
12	RMe	LOCK	17209	QTASTIFC_FLIGHT	19EF	TEMPORARY - SPACE	000000000000EC7B	QPADEV000C	QSECOFR	293276
13	SOo	Seize-obj	15855	EKDADMIN	0801	USER PROFILE	000000000000021B	EXPPRT01	QSPLJOB	271687
14	RMe	LOCK	14082	QTASTIFC_FLIGHT	19EF	TEMPORARY - SPACE	000000000000EC7C	QPADEV000C	QSECOFR	293276
15	Rsh	SEIZE	12932	AMAPICS	0801	USER PROFILE	000000000000E5B9	BPRTSVR	EKDADMIN	292311
16	RMe	LOCK	12670	QTATHPRC_FLIGHT	19EF	TEMPORARY - SPACE	000000000000EC7E	QTATHPRC	QSECOFR	293279
17	Rsh	SEIZE	11683	QRECOVERY	0401	LIBRARY	000000000000E779	VERK02VK	VERKAUF2	292705
18	Rsh	SEIZE	11162	EKDGROUP	0801	USER PROFILE	000000000000021B	EXPPRT01	QSPLJOB	271687
19	RMe	LOCK	11141	QTATHPRC_FLIGHT	19EF	TEMPORARY - SPACE	000000000000EC7D	QTATHPRC	QSECOFR	293278
20	RMe	LOCK	10992	QTATHPRC_FLIGHT	19EF	TEMPORARY - SPACE	000000000000EC7D	QTATHPRC	QSECOFR	293278
21	Rsh	SEIZE	10937	EKDGROUP	0801	USER PROFILE	000000000000E5B9	BPRTSVR	EKDADMIN	292311
22	Rsh	SEIZE	10831	EKDADMIN	0801	USER PROFILE	000000000000E5B9	BPRTSVR	EKDADMIN	292311

Records 1 - 21 of 50

5.8.7 Physical DASD IO Requests

Description:

Example:



Data Viewer - #1 - [Pmr04667_3/Io2/Physical DASD IO Requests - #1]
File
Edit
View
Window
Help

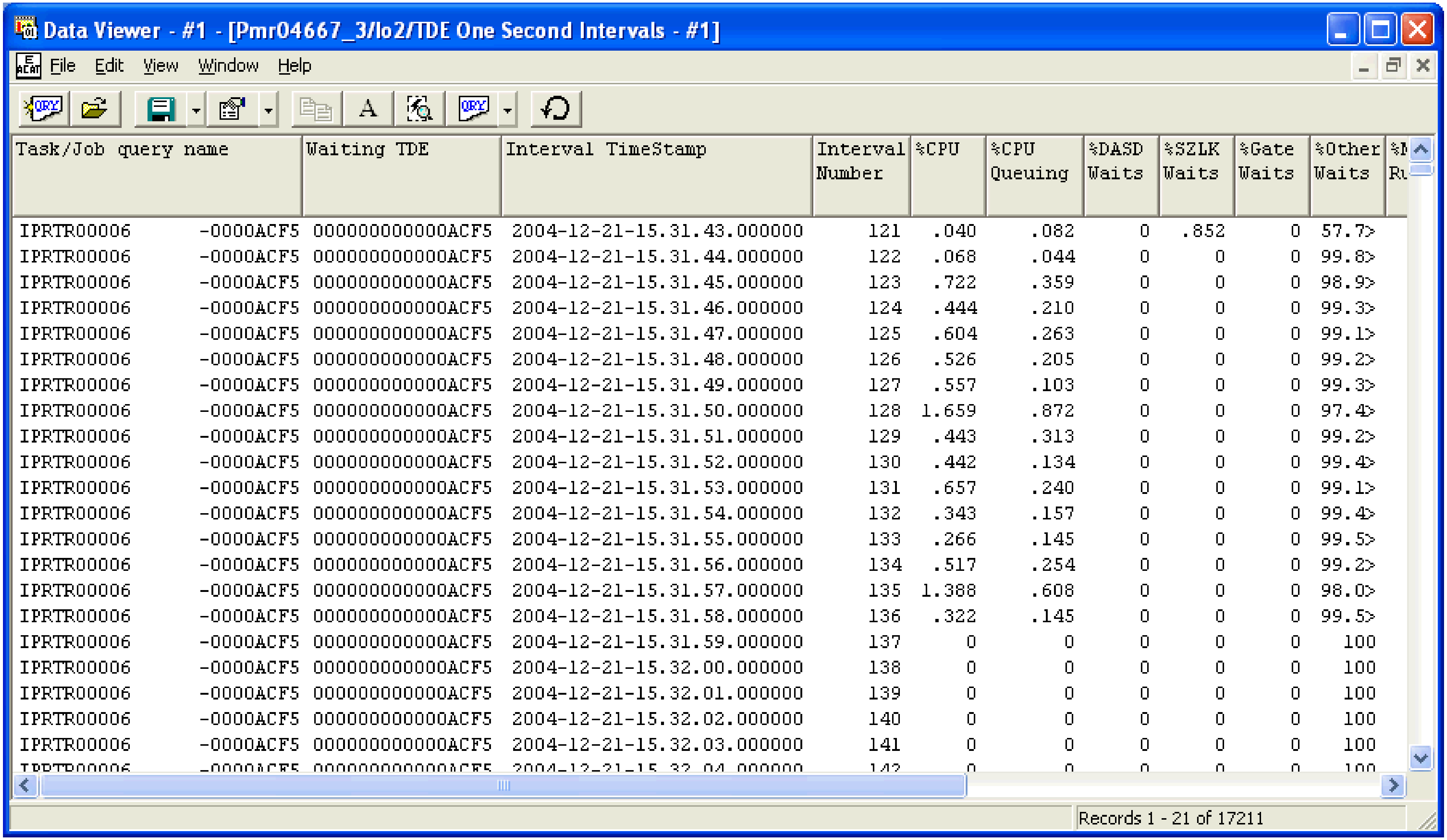
Event type	DASD IOP name	Disk unit (binary)	DASD unit number	Dasd OP type for QRY output	Oper type (event specific)	Disk op time, usecs	Byte len (packed)	Timestamp of DASD start	Timestamp of DASD cmplt	TDE number (count)	TDE number (packed)	SID por of addr
Z		5	00005	WS	*WC S	1872	8192	2004-12-2>	2004-12-2>	00000000>	42069	2298074
Z		5	00005	WA	*WC A	1309	4096	2004-12-2>	2004-12-2>	00000000>	42069	2298074
Z		5	00005	WS	*WC S	1164	8192	2004-12-2>	2004-12-2>	00000000>	42069	2298074
Z		5	00005	WA	*WC A	1143	4096	2004-12-2>	2004-12-2>	00000000>	42069	2298074
Z		1	00001	WS	*WCP S	840	4096	2004-12-2>	2004-12-2>	00000000>	42069	397D537
Z		3	00003	WS	*WCP S	812	4096	2004-12-2>	2004-12-2>	00000000>	42069	0CAF17E
Z		1	00001	FT	*RCF S	8759	4096	2004-12-2>	2004-12-2>	00000000>	46941	FFFFFFF
Z		1	00001	FT	*RCF S	4335	4096	2004-12-2>	2004-12-2>	00000000>	46941	FFFFFFF
Z		1	00001	FT	*RCF S	5821	4096	2004-12-2>	2004-12-2>	00000000>	46941	FFFFFFF
Z		2	00002	WS	*WC S	869	4096	2004-12-2>	2004-12-2>	00000000>	49930	0000000
Z		2	00002	WS	*WC S	1739	4096	2004-12-2>	2004-12-2>	00000000>	49930	0000000
Z		6	00006	WS	*WC S	1974	4096	2004-12-2>	2004-12-2>	00000000>	49930	0000000
Z		6	00006	WS	*WC S	1068	8192	2004-12-2>	2004-12-2>	00000000>	49930	0000000
Z		2	00002	WS	*WC S	2198	4096	2004-12-2>	2004-12-2>	00000000>	49930	0000000
Z		2	00002	WS	*WC S	1780	4096	2004-12-2>	2004-12-2>	00000000>	49930	0000000
Z		10	00010	WS	*WCP S	2332	8192	2004-12-2>	2004-12-2>	00000000>	49930	3FAF93E
Z		9	00009	WS	*WCP S	2513	8192	2004-12-2>	2004-12-2>	00000000>	49930	0D6F229
Z		9	00009	RS	*RC S	13380	4096	2004-12-2>	2004-12-2>	00000000>	49930	0CEB4BC
Z		9	00009	RA	*RC A	1456	4096	2004-12-2>	2004-12-2>	00000000>	49930	0CEB4BC
Z		9	00009	WS	*WCP S	1784	8192	2004-12-2>	2004-12-2>	00000000>	49930	0CEB4BC
Z		9	00009	RS	*RC S	2836	4096	2004-12-2>	2004-12-2>	00000000>	49930	0CEB4BC
Z		9	00009	RA	*RC A	1460	4096	2004-12-2>	2004-12-2>	00000000>	49930	0CEB4BC

Records 1 - 21 of 33890

5.8.8 TDE One Second Intervals

Description:

Example:


Data Viewer - #1 - [Pmr04667_3/lo2/TDE One Second Intervals - #1]
File
Edit
View
Window
Help
QRY
QRY

Task/Job query name	Waiting TDE	Interval TimeStamp	Interval Number	%CPU	%CPU Queuing	%DASD Waits	%SZLK Waits	%Gate Waits	%Other Waits	%N Ru
IPRTR00006 -0000ACF5	0000000000000ACF5	2004-12-21-15.31.43.000000	121	.040	.082	0	.852	0	57.7>	
IPRTR00006 -0000ACF5	0000000000000ACF5	2004-12-21-15.31.44.000000	122	.068	.044	0	0	0	99.8>	
IPRTR00006 -0000ACF5	0000000000000ACF5	2004-12-21-15.31.45.000000	123	.722	.359	0	0	0	98.9>	
IPRTR00006 -0000ACF5	0000000000000ACF5	2004-12-21-15.31.46.000000	124	.444	.210	0	0	0	99.3>	
IPRTR00006 -0000ACF5	0000000000000ACF5	2004-12-21-15.31.47.000000	125	.604	.263	0	0	0	99.1>	
IPRTR00006 -0000ACF5	0000000000000ACF5	2004-12-21-15.31.48.000000	126	.526	.205	0	0	0	99.2>	
IPRTR00006 -0000ACF5	0000000000000ACF5	2004-12-21-15.31.49.000000	127	.557	.103	0	0	0	99.3>	
IPRTR00006 -0000ACF5	0000000000000ACF5	2004-12-21-15.31.50.000000	128	1.659	.872	0	0	0	97.4>	
IPRTR00006 -0000ACF5	0000000000000ACF5	2004-12-21-15.31.51.000000	129	.443	.313	0	0	0	99.2>	
IPRTR00006 -0000ACF5	0000000000000ACF5	2004-12-21-15.31.52.000000	130	.442	.134	0	0	0	99.4>	
IPRTR00006 -0000ACF5	0000000000000ACF5	2004-12-21-15.31.53.000000	131	.657	.240	0	0	0	99.1>	
IPRTR00006 -0000ACF5	0000000000000ACF5	2004-12-21-15.31.54.000000	132	.343	.157	0	0	0	99.4>	
IPRTR00006 -0000ACF5	0000000000000ACF5	2004-12-21-15.31.55.000000	133	.266	.145	0	0	0	99.5>	
IPRTR00006 -0000ACF5	0000000000000ACF5	2004-12-21-15.31.56.000000	134	.517	.254	0	0	0	99.2>	
IPRTR00006 -0000ACF5	0000000000000ACF5	2004-12-21-15.31.57.000000	135	1.388	.608	0	0	0	98.0>	
IPRTR00006 -0000ACF5	0000000000000ACF5	2004-12-21-15.31.58.000000	136	.322	.145	0	0	0	99.5>	
IPRTR00006 -0000ACF5	0000000000000ACF5	2004-12-21-15.31.59.000000	137	0	0	0	0	0	100	
IPRTR00006 -0000ACF5	0000000000000ACF5	2004-12-21-15.32.00.000000	138	0	0	0	0	0	100	
IPRTR00006 -0000ACF5	0000000000000ACF5	2004-12-21-15.32.01.000000	139	0	0	0	0	0	100	
IPRTR00006 -0000ACF5	0000000000000ACF5	2004-12-21-15.32.02.000000	140	0	0	0	0	0	100	
IPRTR00006 -0000ACF5	0000000000000ACF5	2004-12-21-15.32.03.000000	141	0	0	0	0	0	100	
IPRTR00006 -0000ACF5	0000000000000ACF5	2004-12-21-15.32.04.000000	142	0	0	0	0	0	100	

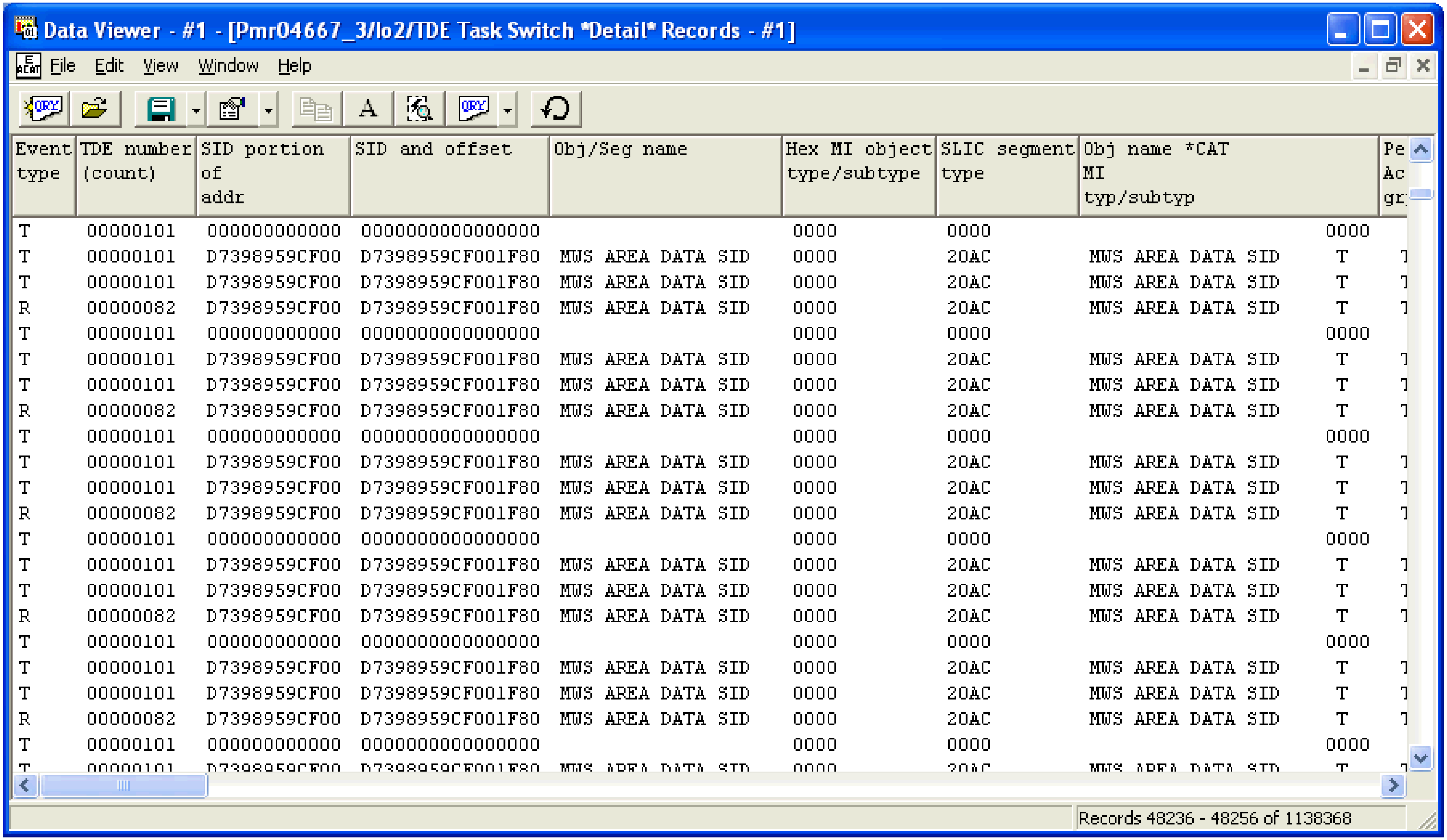
Records 1 - 21 of 17211

Table of Contents Previous Next

5.8.9 TDE Task Switch *Detail* Records

Description:

Example:


Data Viewer - #1 - [Pmr04667_3/lo2/TDE Task Switch *Detail* Records - #1]
File
Edit
View
Window
Help

Event type	TDE number (count)	SID portion of addr	SID and offset	Obj/Seg name	Hex MI object type/subtype	SLIC segment type	Obj name *CAT MI typ/subtyp	Pe Ac gr
T	00000101	000000000000	0000000000000000		0000	0000	0000	
T	00000101	D7398959CF00	D7398959CF001F80	MWS AREA DATA SID	0000	20AC	MWS AREA DATA SID T	1
T	00000101	D7398959CF00	D7398959CF001F80	MWS AREA DATA SID	0000	20AC	MWS AREA DATA SID T	1
R	00000082	D7398959CF00	D7398959CF001F80	MWS AREA DATA SID	0000	20AC	MWS AREA DATA SID T	1
T	00000101	000000000000	0000000000000000		0000	0000	0000	
T	00000101	D7398959CF00	D7398959CF001F80	MWS AREA DATA SID	0000	20AC	MWS AREA DATA SID T	1
T	00000101	D7398959CF00	D7398959CF001F80	MWS AREA DATA SID	0000	20AC	MWS AREA DATA SID T	1
R	00000082	D7398959CF00	D7398959CF001F80	MWS AREA DATA SID	0000	20AC	MWS AREA DATA SID T	1
T	00000101	000000000000	0000000000000000		0000	0000	0000	
T	00000101	D7398959CF00	D7398959CF001F80	MWS AREA DATA SID	0000	20AC	MWS AREA DATA SID T	1
T	00000101	D7398959CF00	D7398959CF001F80	MWS AREA DATA SID	0000	20AC	MWS AREA DATA SID T	1
R	00000082	D7398959CF00	D7398959CF001F80	MWS AREA DATA SID	0000	20AC	MWS AREA DATA SID T	1
T	00000101	000000000000	0000000000000000		0000	0000	0000	
T	00000101	D7398959CF00	D7398959CF001F80	MWS AREA DATA SID	0000	20AC	MWS AREA DATA SID T	1
T	00000101	D7398959CF00	D7398959CF001F80	MWS AREA DATA SID	0000	20AC	MWS AREA DATA SID T	1
R	00000082	D7398959CF00	D7398959CF001F80	MWS AREA DATA SID	0000	20AC	MWS AREA DATA SID T	1
T	00000101	000000000000	0000000000000000		0000	0000	0000	
T	00000101	D7398959CF00	D7398959CF001F80	MWS AREA DATA SID	0000	20AC	MWS AREA DATA SID T	1
T	00000101	D7398959CF00	D7398959CF001F80	MWS AREA DATA SID	0000	20AC	MWS AREA DATA SID T	1
R	00000082	D7398959CF00	D7398959CF001F80	MWS AREA DATA SID	0000	20AC	MWS AREA DATA SID T	1
T	00000101	000000000000	0000000000000000		0000	0000	0000	
T	00000101	D7398959CF00	D7398959CF001F80	MWS AREA DATA SID	0000	20AC	MWS AREA DATA SID T	1

Records 48236 - 48256 of 1138368

Table of Contents Previous Next

5.9 Data area reports

This analysis shows all data area changes and retrieves. It indicates the causing job/thread, data area and library names, data area type, local or DDM, and time of day each operation occurred.

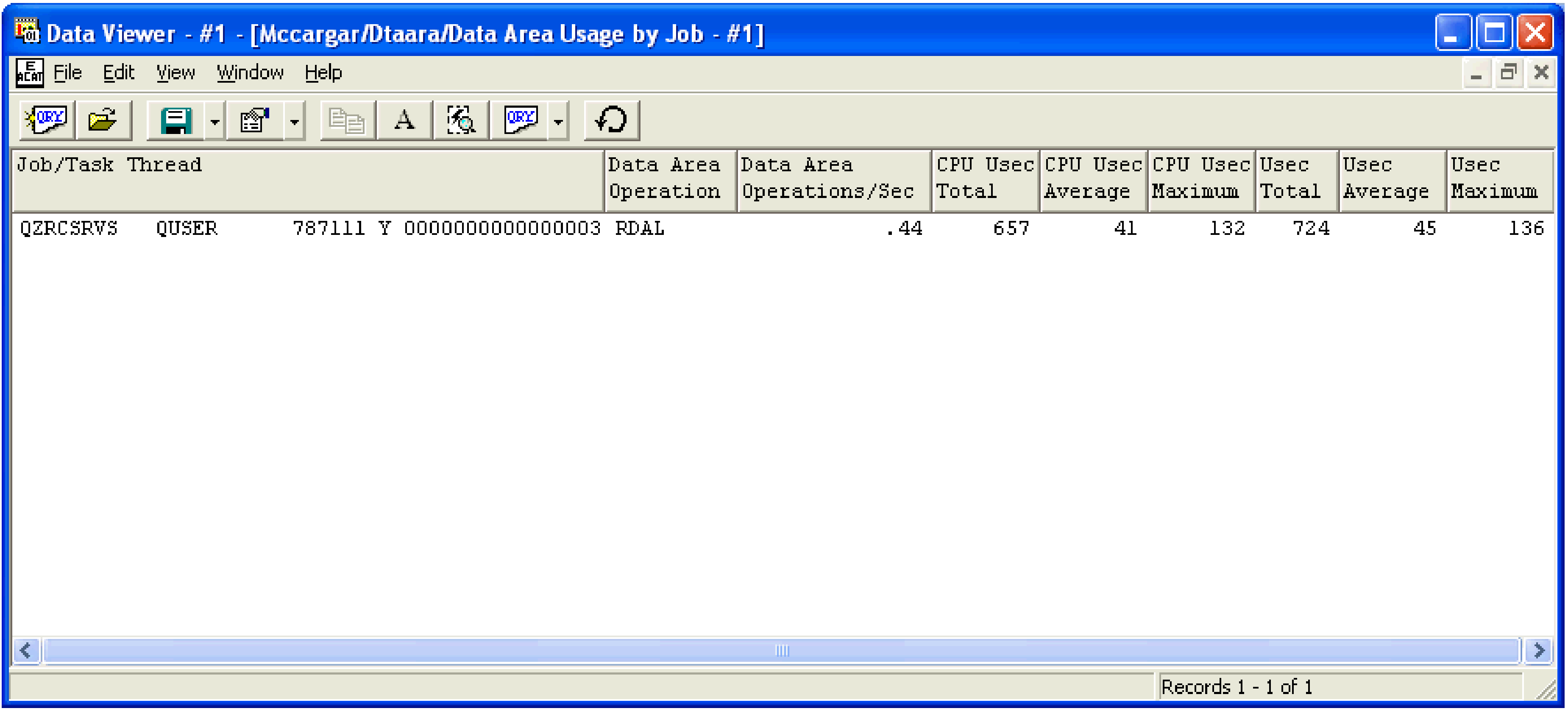
If the program events *MIENTRY and *MIEXIT are collected, then the PEX Analysis shows the calling program name, elapsed time, CPU, and disk I/O during each data area operation.

Table of Contents | Previous | Next

5.9.1 Data area usage by job

Description:
This report displays the jobs that experienced data area usage within the collection.

Example:


Data Viewer - #1 - [Mccargar/Dtaara/Data Area Usage by Job - #1]
File
Edit
View
Window
Help
Records 1 - 1 of 1

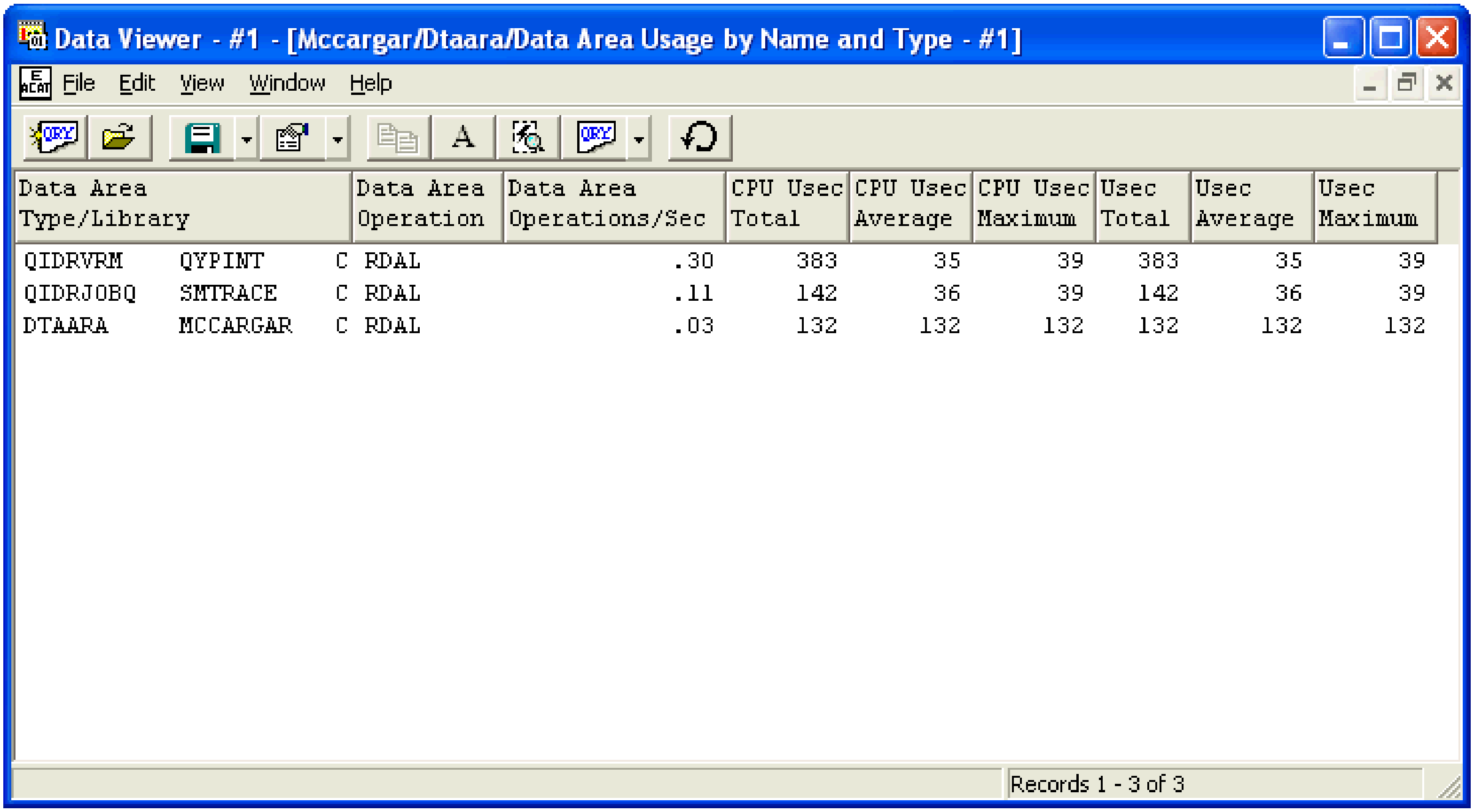
Job/Task Thread	Data Area Operation	Data Area Operations/Sec	CPU Usec Total	CPU Usec Average	CPU Usec Maximum	Usec Total	Usec Average	Usec Maximum
QZRCSRVS QUSER 787111 Y 0000000000000003	RDAL	.44	657	41	132	724	45	136

Table of Contents Previous Next

5.9.2 Data area usage by name and type

Description:
This report provides the data areas that were operated on as well as statistics for each data area during the collection.

Example:


Data Viewer - #1 - [Mccargar/Dtaara/Data Area Usage by Name and Type - #1]
File
Edit
View
Window
Help

Data Area Type/Library	Data Area Operation	Data Area Operations/Sec	CPU Usec Total	CPU Usec Average	CPU Usec Maximum	Usec Total	Usec Average	Usec Maximum
QIDRVRM QYPINT C	RDAL	.30	383	35	39	383	35	39
QIDRJOBQ SMTRACE C	RDAL	.11	142	36	39	142	36	39
DTAARA MCCARGAR C	RDAL	.03	132	132	132	132	132	132

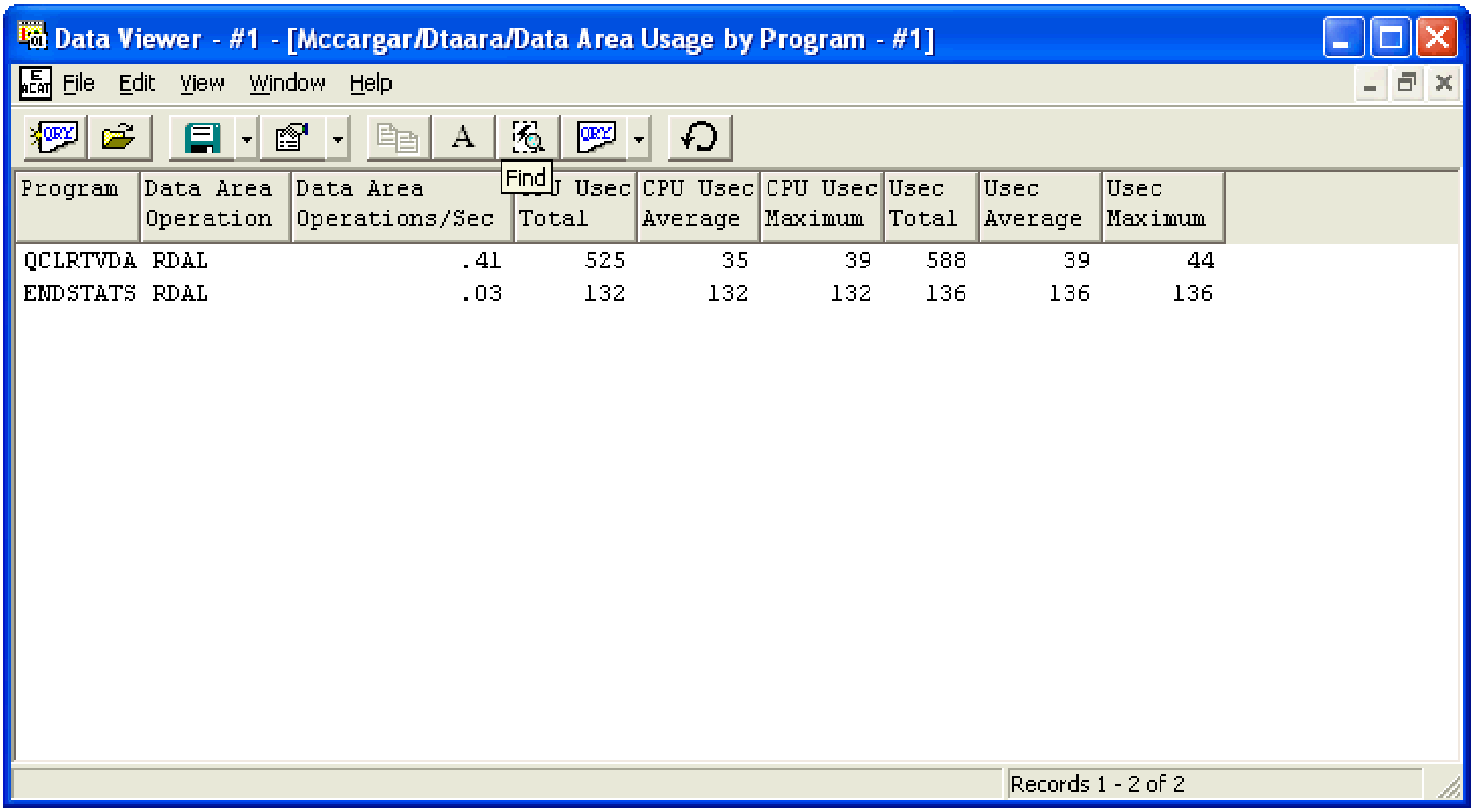
Records 1 - 3 of 3

Table of Contents Previous Next

5.9.3 Data area usage by program

Description:
This report provides the program names that were operated on as well as statistics for each program during the collection.

Example:


Data Viewer - #1 - [Mccargar/Dtaara/Data Area Usage by Program - #1]
File
Edit
View
Window
Help
Find
Records 1 - 2 of 2

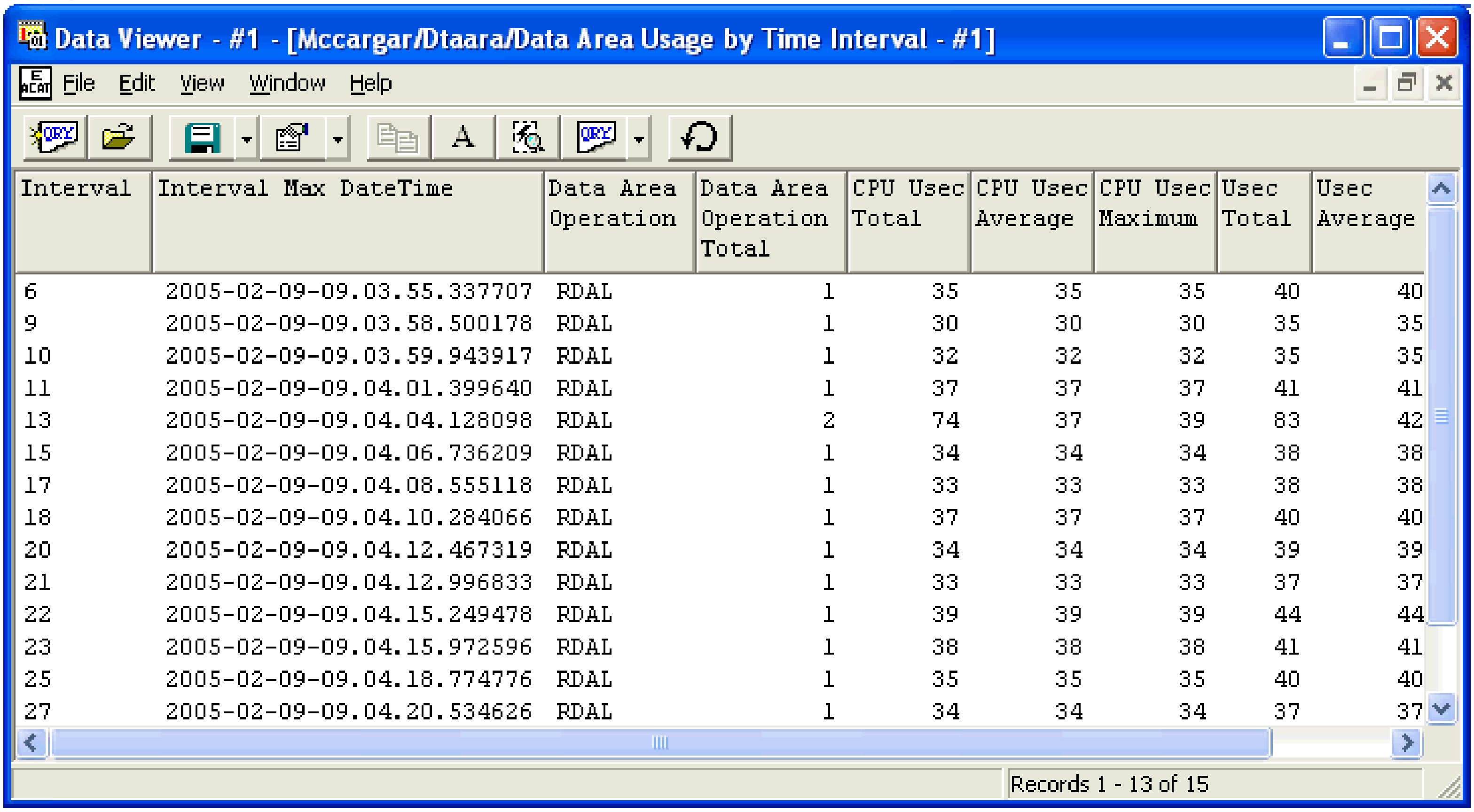
Program	Data Area Operation	Data Area Operations/Sec	U Usec Total	CPU Usec Average	CPU Usec Maximum	Usec Total	Usec Average	Usec Maximum
QCLRTVDA	RDAL	.41	525	35	39	588	39	44
ENDSTATS	RDAL	.03	132	132	132	136	136	136

Table of Contents Previous Next

5.9.4 Data area usage by time interval

Description:

Example:


Data Viewer - #1 - [Mccargar/Dtaara/Data Area Usage by Time Interval - #1]
File
Edit
View
Window
Help
QRY
A
QRY
Records 1 - 13 of 15

Interval	Interval Max DateTime	Data Area Operation	Data Area Operation Total	CPU Usec Total	CPU Usec Average	CPU Usec Maximum	Usec Total	Usec Average
6	2005-02-09-09.03.55.337707	RDAL	1	35	35	35	40	40
9	2005-02-09-09.03.58.500178	RDAL	1	30	30	30	35	35
10	2005-02-09-09.03.59.943917	RDAL	1	32	32	32	35	35
11	2005-02-09-09.04.01.399640	RDAL	1	37	37	37	41	41
13	2005-02-09-09.04.04.128098	RDAL	2	74	37	39	83	42
15	2005-02-09-09.04.06.736209	RDAL	1	34	34	34	38	38
17	2005-02-09-09.04.08.555118	RDAL	1	33	33	33	38	38
18	2005-02-09-09.04.10.284066	RDAL	1	37	37	37	40	40
20	2005-02-09-09.04.12.467319	RDAL	1	34	34	34	39	39
21	2005-02-09-09.04.12.996833	RDAL	1	33	33	33	37	37
22	2005-02-09-09.04.15.249478	RDAL	1	39	39	39	44	44
23	2005-02-09-09.04.15.972596	RDAL	1	38	38	38	41	41
25	2005-02-09-09.04.18.774776	RDAL	1	35	35	35	40	40
27	2005-02-09-09.04.20.534626	RDAL	1	34	34	34	37	37

Table of Contents Previous Next

5.10 Data queue activity reports

This analysis shows all data queue sends and receives. It indicates the causing job/thread, data queue and library names, and time of day each operation occurred.

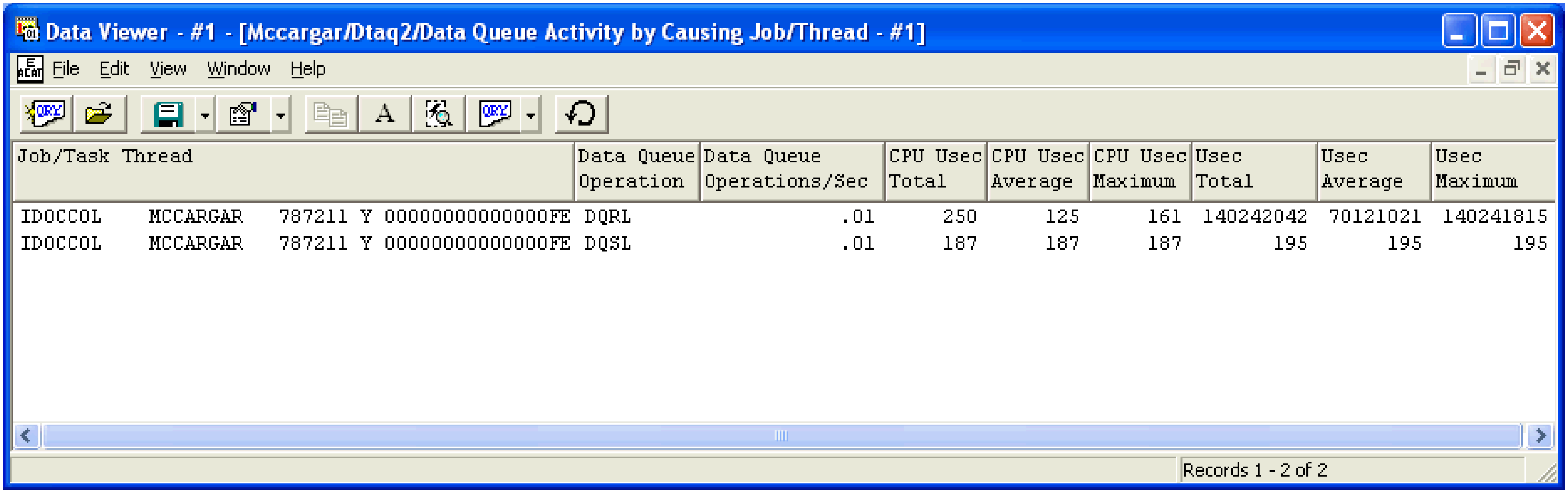
If the program events *MIENTRY and *MIEXIT are collected, then the analysis shows the calling program name, elapsed time, CPU, and disk I/O during each data queue operation.

Table of Contents | Previous | Next

5.10.1 Data queue activity by causing job/thread

Description:

Example:

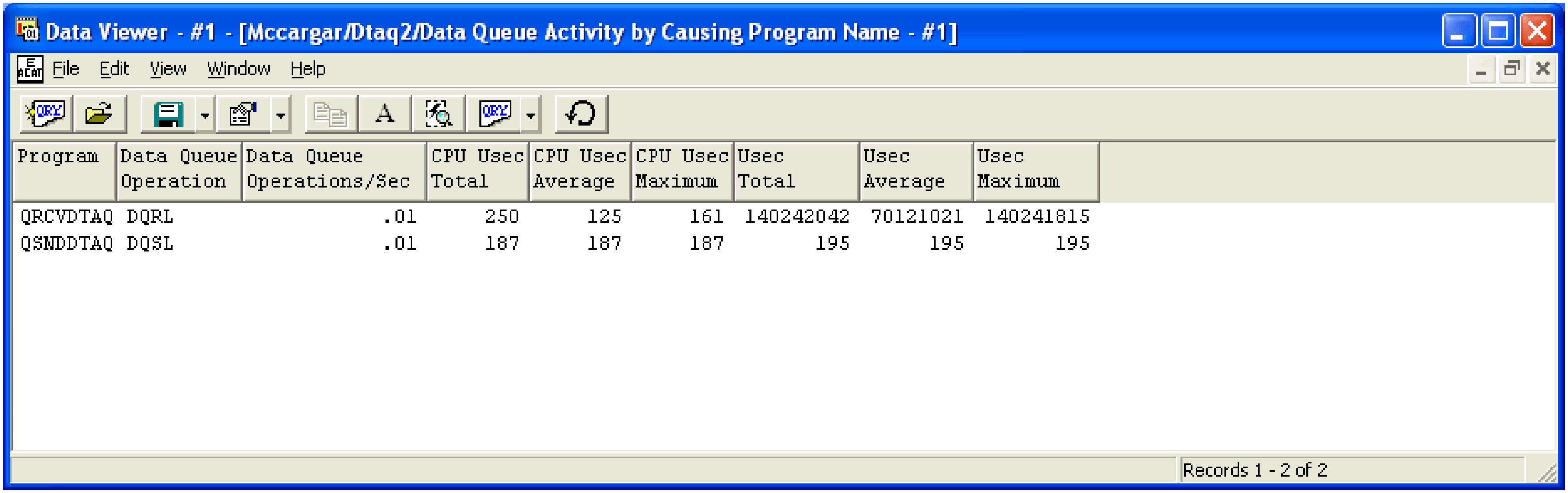

Data Viewer - #1 - [Mccargar/Dtaq2/Data Queue Activity by Causing Job/Thread - #1]
File
Edit
View
Window
Help
Records 1 - 2 of 2

Job/Task Thread	Data Queue Operation	Data Queue Operations/Sec	CPU Usec Total	CPU Usec Average	CPU Usec Maximum	Usec Total	Usec Average	Usec Maximum
IDOCCOL MCCARGAR 787211 Y 00000000000000FE	DQRL	.01	250	125	161	140242042	70121021	140241815
IDOCCOL MCCARGAR 787211 Y 00000000000000FE	DQSL	.01	187	187	187	195	195	195

5.10.2 Data queue activity by causing program name

Description:

Example:



Data Viewer - #1 - [Mccargar/Dtaq2/Data Queue Activity by Causing Program Name - #1]
File Edit View Window Help

Program	Data Queue Operation	Data Queue Operations/Sec	CPU Usec Total	CPU Usec Average	CPU Usec Maximum	Usec Total	Usec Average	Usec Maximum
QRCVDTAQ	DQRL	.01	250	125	161	140242042	70121021	140241815
QSNDDTAQ	DQSL	.01	187	187	187	195	195	195

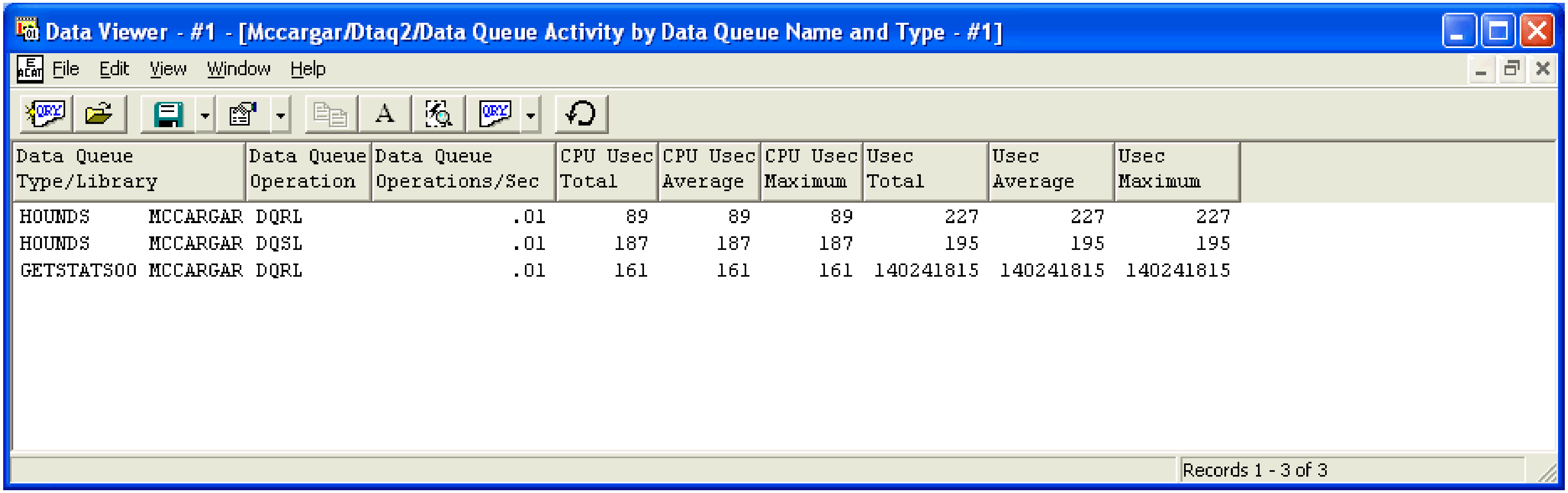
Records 1 - 2 of 2

Table of Contents Previous Next

5.10.3 Data queue activity by data queue name and type

Description:

Example:



Data Viewer - #1 - [Mccargar/Dtaq2/Data Queue Activity by Data Queue Name and Type - #1]
File
Edit
View
Window
Help
Records 1 - 3 of 3

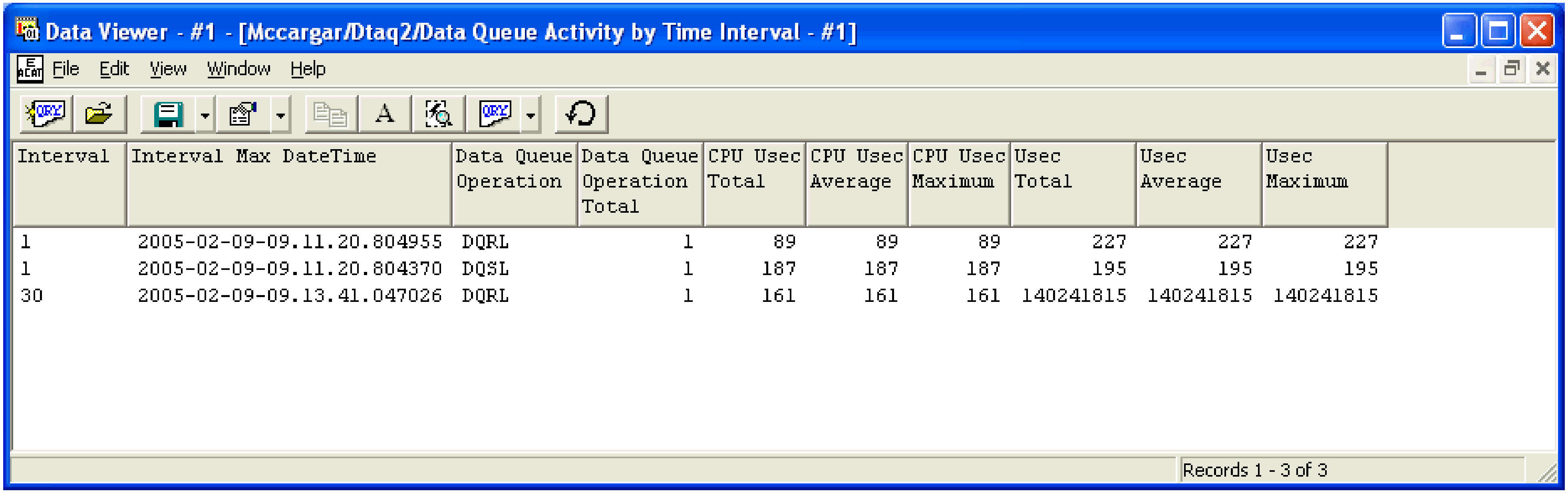
Data Queue Type/Library	Data Queue Operation	Data Queue Operations/Sec	CPU Usec Total	CPU Usec Average	CPU Usec Maximum	Usec Total	Usec Average	Usec Maximum
HOUNDS MCCARGAR	DQRL	.01	89	89	89	227	227	227
HOUNDS MCCARGAR	DQSL	.01	187	187	187	195	195	195
GETSTATS00 MCCARGAR	DQRL	.01	161	161	161	140241815	140241815	140241815

Table of Contents Previous Next

5.10.4 Data queue activity by time interval

Description:

Example:



Data Viewer - #1 - [Mccargar/Dtaq2/Data Queue Activity by Time Interval - #1]
File Edit View Window Help

Interval	Interval Max DateTime	Data Queue Operation	Data Queue Operation Total	CPU Usec Total	CPU Usec Average	CPU Usec Maximum	Usec Total	Usec Average	Usec Maximum
1	2005-02-09-09.11.20.804955	DQRL	1	89	89	89	227	227	227
1	2005-02-09-09.11.20.804370	DQSL	1	187	187	187	195	195	195
30	2005-02-09-09.13.41.047026	DQRL	1	161	161	161	140241815	140241815	140241815

Records 1 - 3 of 3

Table of Contents | Previous | Next

Chapter 6 PTDV

This chapter provides an overview of the modes of operation and interfaces for the iDoctor for iSeries PTDV component.

Table of Contents | Previous | Next

6.1 Modes of Operation

There are three basic modes of operation for PTDV which provide a GUI. Manual mode, which was supported in previous releases of PTDV, is no longer supported.

Thin Client

This is the default mode for PTDV. In this mode, there is a PTDV server which runs on your iSeries, and does most of the data processing. The PTDV client acts mostly as a presentation layer on your client system. This mode is generally the fastest method of running the PTDV, as all of the database access is done on the local system. Using this mode also allows you to process the largest number of events (currently the largest trace processed has been about 10 million events), as the data is kept on the iSeries, which can generally handle large amounts of data more effectively than a PC. The primary disadvantage to this mode is that a connection to your iSeries must be active the entire time you are using PTDV, and some processing time on the server will be used. Therefore, this method is not ideal for viewing data on production systems. In addition, the PTDV server must be installed on the iSeries, which is sometimes not practical/desirable.

Thick Client

This mode is very similar to the way that the original version of PTDV ran. In this case, the data processing code and the presentation are both done on your client system, and the iSeries is used only to access the database. This mode tends to be somewhat slower than thin-client mode, and can require more network bandwidth. In addition, the amount of data which can be processed is limited to the amount of memory on your PC -- most systems will run out of memory if more than 300,000 events (or sometimes even less) are processed. However, the thick client mode can be useful when dealing with small collections (of only a few thousand events), since it does not require any PTDV code to be installed on the iSeries. Since it does not require a connection to the server after the events are initially processed, this mode can also be useful when it is not acceptable to do processing on the server over an extended period of time.

Three Tier

This mode is a mix between thin client and thick client. In this case, one iSeries holds the PEX data. Another iSeries runs the PTDV server, and your client system runs the PTDV client. This results in a mix of the advantages and disadvantages above. Processing can sometimes take even longer than in the Thick Client mode. However, the Three Tier mode can process as many events as the Thin Client mode.

In addition, the connection to the first iSeries (database server) is closed after the initial processing (as in Thick Client), so this mode can work well for retrieving data from production systems. In general, the Thin Client mode is preferable unless it is necessary to do only minimal processing on the system with the database. Even in this case, it is generally better to end PEX with the *FILE option, and then move the database over to another iSeries system which can be used for the processing.

A graphic and chart follow to aid in understanding.

Run Modes Graphic

Run Modes Quick Reference Chart

Run Mode	Speed	Number of Events Possible	iSeries Setup, JV1 LPP Required?	Active Connection to iSeries Required?
Thin Client	Fastest	Most (~10 millions)	Yes	At all times
Thick Client	Medium	Least (low 100s of thousands)	No	Only during event processing
Three Tier	Slowest	Most (low millions)	Yes*	At all times*

Table of Contents | Previous | Next

6.2 Navigating through Frames and Panes

One of the unique features of PTDV is the way in which the user navigates through the performance data. Data is organized into a set of tables or trees, which are then grouped together into frames. The contents of the frames and panes depends on the types of events in the collection being processed. As you navigate through the data and find areas of interest, you can get to more detailed information by selecting a row from one of the panes and clicking on it.

The Main Frame is the starting point for processing. The panes available on the Main Frame give you views of the data for the entire collection. After selecting a pane, you can click on a row in the table or tree to get to its corresponding frame, which will then display more details on the item you are interested in.

5.2.1 Frames

A frame is a set of one or more tabbed panes, and is usually associated with an entity in the collection. The data in the frame may be scoped to a level other than the entire collection; it might represent data for only a single job or thread.

The following frames are available:

5.2.1.1 Main Trace Frame

For a trace collection, the initial panes are:

- **Collection Information Pane**
  The collection information is displayed in an Informational Pane, which contains details on the current collection.

- **Event Counts Pane**
  This pane contains information about the events that appear in the current trace collection. It provides the event counts organized by event type and subtype.

- **Job/Thread Pane**
  The Job/Thread panedisplays the jobs and threads in the collection, along with performance data for each job and thread. Once processing has been completed the following panes are displayed:

- **Procedure Summary Pane**
  Information about the procedures used within the collection is displayed in a Procedure

Summary Pane. This data represents procedure information over the entire collection.

- **Object Summary Pane**
  A summary of the objects used in this collection are displayed as an Object Summary Pane.

5.2.1.2. Job Frame

The Job Frame contains information about a specific Job within the collection. The Job Frame consists of the following panes:

- **Job Information Pane**
  This pane is an Informational Pane, providing details on the job associated with this frame.
- **Threads Pane**
  A table containing the threads associated with the current job is the Threads Pane.
- **Procedure Summary Pane**
  The Procedure Summary Panefor this job contains performance data collected for the procedures within the current job.
- **Object Summary Pane**
  The Object Summary Panefor this job includes information on the objects used by the current job, organized by class name (object type).
- **Garbage Collection Sweep Pane**
  This pane is optional depending on whether the collection contains garbage collection sweep events.Garbage Collection Sweep Pane

  The WebSphere Summary Pane summarizes information about the WebSphere events that have occurred in this collection within the current job. This pane will appear only if there are WebSphere events in the current collection.

5.2.1.3. Thread Frame

The Thread Frame contains information about a specific Thread or Task within the collection. The Thread Frame consists of the following panes:

- **Procedure Summary Pane**
  The Procedure Summary Paneprovides summary information about the procedures/methods appearing within the current thread or task.
- **Call Trace pane**
  The Call Trace Panefor a Thread Frame displays the call flow for all procedures/methods

within the current thread.

- **Object Summary Pane**
  The Object Summary Panefor a Thread contains the summary information about the objects appearing within the thread associated with the current frame. Data in this table is grouped by class name (object type).

5.2.1.4. Procedure Summary Frame

The procedure summary frame contains several panes with detailed information about a specific procedure within a given collection level. The information in the panes for a given frame might contain data for the entire collection, or for a specific job or thread. This frame consists of the following panes:

- **Procedure Information Pane**
  The Procedure Information Pane is an Informational Panewhich provides summary information about the procedure within the current collection level.

- **Call site pane**
  The call site pane is a Procedure Call Pane. It contains all of the individual call sites (i.e., single calls of the procedure) for the current procedure within the frame's collection level.

- **Cumulative callers pane**
  This table is a Procedure Summary Panewhich contains the summarized information about the callers of the current procedure within the frame's collection level.

- **Cumulative callees pane**
  This table is also a Procedure Summary Panewhich contains the summarized information about the callees (procedures that are called by the current procedure and its descendants) of the current procedure.

- **Inline objects pane**
  This table is an Object Summary Pane, containing the summarized information about the java objects that are associated with the java events that have occurred inline within the current procedure.

- **Cumulative objects pane**
  The cumulative objects table is also a Object Summary Pane, containing information about the objects that are associated with the java events that occur by the current procedure or procedures that are called cumulatively by the current procedure.

5.2.1.5. Procedure Call Frame

The procedure frame contains information about a single procedure call within the collection. It consists of the following panes:

- **Information pane**
  This pane is an Informational Paneproviding details about the procedure represented by this frame.

- **Call ancestors pane**
  This pane provides a view of the call stack at the point of the procedure call associated with the current frame. This table is a Procedure Call Pane.

- **Call subtree pane**
  This pane contains the tree representation of the calls that occur beneath the current procedure. It is actually a subtree of the entire call tree, with the current procedure as its root. It is a form of a Call Trace Pane.

- **Cumulative callees pane**
  This table is a Procedure Summary Panecontaining the summarized information about the callees (procedures that are called by the current procedure and its descendants) of the current procedure.

- **Inline objects pane**
  This table is an Object Summary Panecontaining the summarized information about the java objects that are associated with java events within the current procedure. The data is in this table is organized by object type.

- **Cumulative objects pane**
  This table is also an Object Summary Panecontaining the summarized information about the java objects that are associated with java events within the cumulative callees of the current procedure. The data is in this table is organized by object type.

5.2.1.6. Object Group Frame

The Object Group Frame contains information about objects of a single class, containing data over a specific collection level. The Object Group Frame consists of the following panes:

- **Object Group Information Pane**
  This pane is an Informational Pane, providing some summary information about the objects within the group for the current frame.

- **Associated methods**
  This table is an Object Summary Pane, containing information about the objects for the class for the current object group organized by method name. The data within each row represents the objects that have associated events within a specified method; for example, a row of the table contains information about the create and all locks for objects within a given class for a specific method.

- **Object instances**
  This table is an Object Instances Pane, containing the set of individual objects within the current object group. In some cases, this table can be very large, and is usually represented as a dynamic data pane.

5.2.1.7. Object Info Frame

The Object Info Frame contains information about a single object instance. In most cases, this frame will contain all information for this object over the entire collection. The Object Info Frame consists of the following panes:

- **Create Call Ancestors Pane**
  This pane provides a view of the call stack at the point of the object create event for the current object. This table is a Procedure Call Pane.The table for this pane is only generated when there are object create events and java method entry and exit events in the collection.

- **Object Locks Pane**
  This table contains the list of all locks that have occurred on the object associated with the current frame. This data is displayed in an Object Locks Pane. The information in this table is only generated when there are object lock events in the collection.

- **Notify/Wait Events Pane**
  This table is a Object Notify/Wait Pane, containing information about thread notify or thread wait events that have occurred on the current object. The information in this table is only generated when there are thread notify or thread wait events in the collection.

5.2.1.8. Object Lock Frame

The Object Lock Frame contains information about a single object lock, which consists of a grouping of want, get, and release lock events on a single object. It consists of the following panes:

- **Lock Information Pane**
  This is an Informational Paneproviding information about the lock associated with the current frame.
- **Concurrent Locks in Thread**
  This is an Object Lock Pane, containing the locks that occur within the current thread during the hold time for the current lock.
- **Concurrent Locks in Job**
  This is also an Object Lock Pane, containing the locks that occur within the current job during the hold time for the current lock.
- **Locks During Hold Pane**
  This is a Object Lock Pane, containing all the locks that occur on the current object during the hold time of the current lock.
- **Methods During Hold Pane**
  This table is a Call Trace Pane, displaying in tree form the set of methods that are called

while the current lock is being held.

5.2.1.9. Event Summary Frame

The Event Summary Frame summarizes information about a specific event that is included in the collection. It contains the following panes:

- **WebSphere Event Information Pane**
  This is an Informational Paneproviding detailed information about the current WebSphere event.

- **Call ancestors pane**
  This pane provides a view of the call stack when the WebSphere event occurred. This table is a Procedure Call Pane.

5.2.1.10. WebSphere Event Frame

The WebSphere Event Frame contains detailed information about an instance of a WebSphere event. It contains the following panes:

- **WebSphere Event Information Pane**
  This is an Informational Paneproviding detailed information about the current WebSphere event.

- **Call ancestors pane**
  This pane provides a view of the call stack when the WebSphere event occurred. This table is a Procedure Call Pane.

5.2.1.11. WebSphere Event Summary Frame

The WebSphere Event Summary Frame contains summary information about a single WebSphere event within a specific collection level. FOr example, it might contain the set of WebSphere events that occurred in the entire collection, or for a specific job or thread. It contains the following panes:

- **Individual WebSphere Events Pane**
  This is an WebSphere Event Paneproviding information about the individual WebSphere events for the type associated with the current frame.

- **Cumulative Callers**
  This is an Procedure Summary Pane, containing counts and other details about the procedures that called the current WebSphere event.

5.2. Panes

A frame consists of one or more panes, which can be selected by clicking on its tab. A pane can contain a table or tree of data. A tree is used in those cases where the data is best represented in a parent/child relationship, such as a call trace. For both tables and trees, the data is organized by rows and columns, and in many cases, clicking on a row of a table or tree will generate a new frame containing more detailed information about that row.

5.2.1. Informational Pane

In most frames, the first pane contains a description for that frame. On the main frame, the informational pane contains information about the collection itself. Each frame type has a specific format.

5.2.2. Event counts pane

This pane contains information about the events that appear in the current trace collection. It provides the event counts organized by event type and subtype.

5.2.3. Job/Thread Pane

This pane appears in the main frame for most collection types. It contains information about the jobs and threads (tasks) in the collection currently being processed. It displays information in a tree format, showing the relationships between jobs and threads.

This table contains performance data for each job and thread in the collection, such as cycle counts, instruction counts, total time, total events, and other information depending on the events found in the collection.

Double-clicking on a job or thread row in this table will generate and show the corresponding Job Frame or Thread Frame. These frames provide performance information scoped to the selected Job or Thread.

5.2.4. Threads Pane

The Threads Pane is a table containing information about the threads (tasks) associated with the job for the current frame. Each row of the table corresponds to a thread, and contains performance data, such as cycles, instructions, time, create and delete timestamps, as well as status flags about the thread.

This table is generated and available for all collections. The availability of some of the data is dependent on the events in the collection, such as thread create and delete timestamps.

Double-clicking on a row of this table will generate and display the Thread Frame associated with the selected thread.

5.2.5. Procedure Summary Pane

This pane contains summary information about the procedures within the collection. This data can be scoped to a specific collection level, such as a job, thread, or the entire collection.

The rows in this table correspond to a specific procedure. Each row of the table contains the procedure name along with the associated performance data, such as times called, time spent inline for the procedure, time spent cumulatively for the procedure (i.e., including time for all procedures called between entry and exit of the procedure), cycles, etc. Other data is available, depending on the events in the collection, such as object create information, object lock information, or WebSphere data.

Information for this table will be generated only if there are program events in the collection being processed. This includes Java entry and exit events, MI program entry and exit events, and Procedure entry and exit events.

Double-clicking on a row in this table will open the corresponding Procedure Summary Frame.

5.2.6. Procedure Call Pane

The procedure call pane is a table containing individual procedure call instances (i.e., a single call of a procedure). Each row of the table contains performance data about a single call, such as time, cycles, and other event counts for events from the collection.

This table can only be generated if there are program events in the current collection.

Double-clicking on a row of this pane will generate and display the Procedure Call Frame for the selected procedure call.

5.2.7. Call Trace Pane

This pane contains a tree representing the call flow based on the entry/exit event information in the collection. It displays the parent/child call relationships for a subset of the collection. Each row corresponds to a procedure call within the tree, and contains performance data about the specific procedure call. Other information may also be available depending on the types of events that are present in the collection, such as object events or WebSphere events.

Double-clicking on a procedure row of this table will generate and display the Procedure Call Frame associated with the selected procedure call.

5.2.8. Object Summary Pane

The Object Summary Pane is a table containing information on objects that are associated with the java events occurring in the current collection. The collection may contain object create, object lock, thread notify, or class load events, all of which will have a class associated with them. The data in this table represents the objects that occur in the events of the collection, grouped by class name (i.e., object type). On an associated method pane, the data may be further grouped by method name.

The data in this table is generated only when there are java object events in the collection, such as object create, object lock, or thread notify.

Double-clicking on a row of this pane will generate and display the Object Group Frame for the selected *class name.*

5.2.9. Object Instances Pane

This table contains a set of individual object instances. It will contain information about the create and delete timestamps, the creating procedure, and summary information about the locks for the object. The data in this table will depend on the types of object events found in the collection.

Double-clicking on a row in this table will generate and display the corresponding Object Info Frame for the selected object.

5.2.10. Object Lock Pane

This pane contains a table of locks. A "lock" is considered to be a set of want, get, and/or release events operating on a single object, representing the actual lock/unlock behavior for an object. The data in this table provides the timestamps for these events, the hold and wait times, as well as the name of the method, thread, and job locking the object.

Double-clicking on a procedure row of this table will generate and display the Object Lock Frame for the selected lock.

5.2.11. Object Notify/Wait Pane

This pane contains a table of wait and notify events that have occurred for a specific object. One of these events will occur whenever the java.lang.Thread.notify(), notifyAll(), or wait() methods have been invoked. The table contains information about the notify or notifyAll, the wait time, the method that invoked the notify or wait event, and links up the corresponding wait and notify events.

5.2.12. Garbage Collector Sweep Pane

This table contains information about the garbage collector sweep events that have occurred in the collection. This data is similar to what is output to the screen when using the *verbose option on the java command.

This table will be generated only if there are garbage collection sweep events in the trace collection being processed.

5.2.13. WebSphere Summary Pane

The WebSphere Summary Pane is a table containing summary information for all the WebSphere events that have occurred within a specific collection level.

This pane can only be generated when there are WebSphere events in the collection.

Double-clicking on a row of this table will generate and display the WebSphere Event Summary Frame associated with the selected WebSphere event.

5.2.14. WebSphere Event Pane

The WebSphere Event Pane is a table containing information on the individual WebSphere events of the same type as the current WebSphere Event Frame. It contains detailed event information based on the data in each WebSphere event.

This pane can only be generated when there are WebSphere events in the collection.

Double-clicking on a row of this table will generate and display the WebSphere Event Frame associated with the selected individual WebSphere event.

Table of Contents | Previous | Next

Chapter 8 Heap Analyzer

This chapter provides an overview of the interfaces within the iDoctor for iSeries - Heap Analyzer component.

Table of Contents | Previous | Next

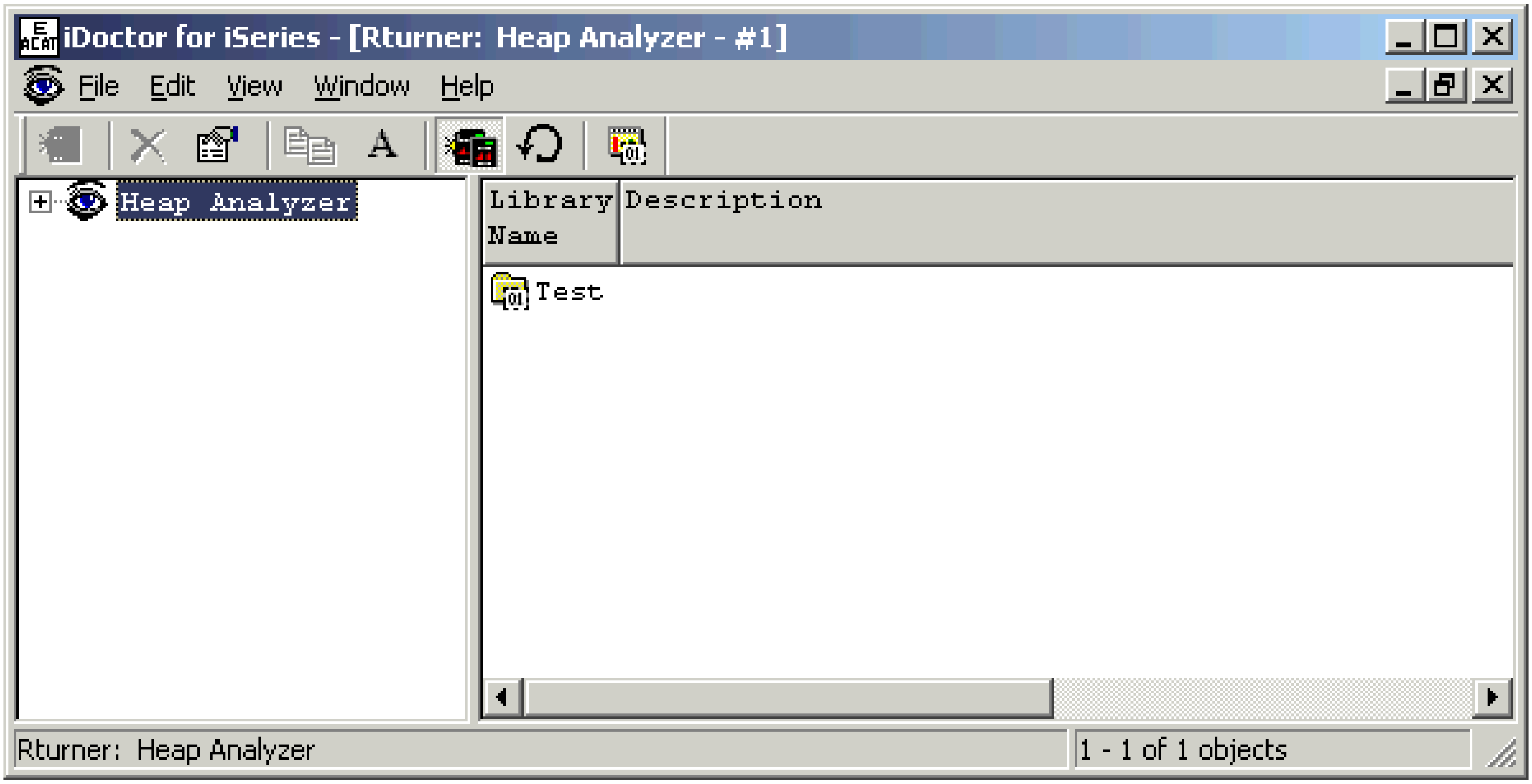
8.1 Heap Analyzer Basics

The Heap Analyzer component allows a Java^{TM} application developer on the iSeries to diagnose Java performance issues such as heap growth issues in the Java Virtual Machine.

Starting Heap Analyzer

Heap Analyzer is a component of the iDoctor for iSeries suite of tools. iDoctor for iSeries can be started using the Start menu: Start->Programs->iDoctor for iSeries. Once the iDoctor for iSeries application appears, the Heap Analyzer component is started from the Connection List View by right-clicking on a system name and choosing the Heap Analyzer menu.

After starting Heap Analyzer you will see a window similar to the following:



[The Heap Analyzer component displaying a list of libraries containing "collections" on a system.]

The 'Heap Analyzer' folder contains a list of library folders, each representing a library on the iSeries system that contains Job Watcher database files (collectiones). The list displays each library's name and description.

Heap Analyzer Objects

There are various types of objects within the tree/list views of Heap Analyzer in the following order: **Libraries**, **collections**, and **reports.** Each of these will covered in more detail in the next sections.

Heap Analyzer Menu Options

The following menu options are available by right-clicking on the 'Heap Analyzer' folder in the tree/list view above.

Menu Item	Description
Explore	Displays the contents of the Heap Analyzer folder (list of libraries on the system containing Heap Analyzer data) in the right pane of the tree/list window.
Start Heap Watch...	This menu will open the Start Heap Watch Wizard where you can define and run a Heap Watch.
Open New Data Viewer	Opens a new Data Viewer window. This window is used to display tables and graphs on the system. You can open Heap Analyzer reports into this window or you can also open any other type of physical file and view as a graph or table.
Properties	Use this menu to display Heap Analyzer version information.

Table of Contents Previous Next

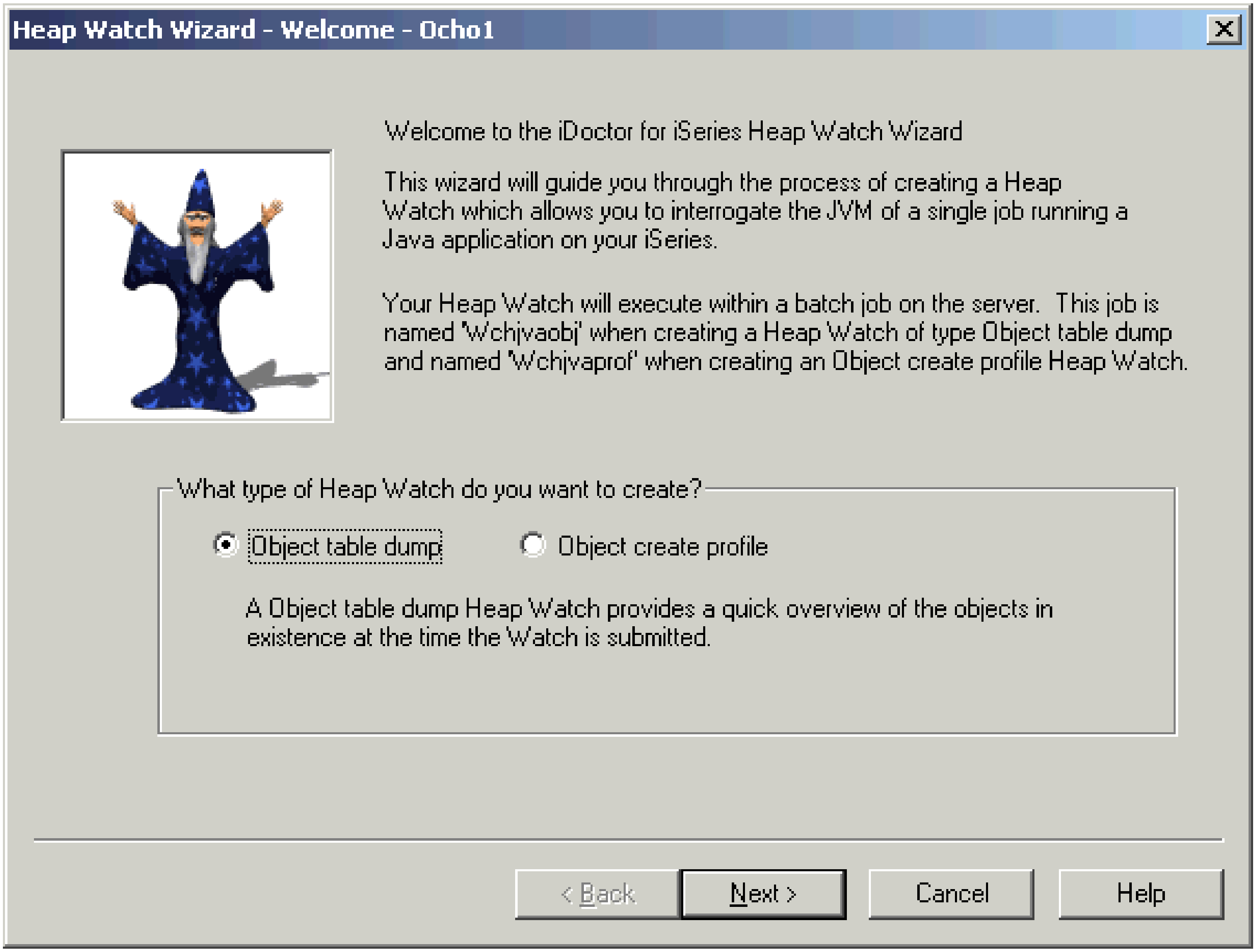
8.2 Start Heap Watch Wizard

Heap Analyzer provides the capability to collect detailed information about any currently running job on the system that is running Java. There are two modes of collection. The first is the called the 'object table snapshot' and the second type is the 'object create profile'. The 'object table snapshot' is a one shot look at the JVM for a particular job showing the breakdown of number of objects/(Java classes) of each type, and how much memory is being consumed by these objects.

The 'object create profile' mode lets you sample information from the JVM as fast as possible until a limit condition is reached. The limit may be provided as a time value (seconds), disk space (megabytes) or number of samples.

To create a new Heap Watch use the Heap Watch Wizard. The Wizard is accessible via the Start Heap Watch menu on the Heap Analyzer or library folder icons.

An example of the Wizard is shown below:



Welcome to the iDoctor for iSeries Heap Watch Wizard

This wizard will guide you through the process of creating a Heap Watch which allows you to interrogate the JVM of a single job running a Java application on your iSeries.

Your Heap Watch will execute within a batch job on the server. This job is named 'Wchjvaobj' when creating a Heap Watch of type Object table dump and named 'Wchjvaprof' when creating an Object create profile Heap Watch.

What type of Heap Watch do you want to create?

● Object table dump ○ Object create profile

A Object table dump Heap Watch provides a quick overview of the objects in existence at the time the Watch is submitted.

[The Start Heap Watch Wizard]

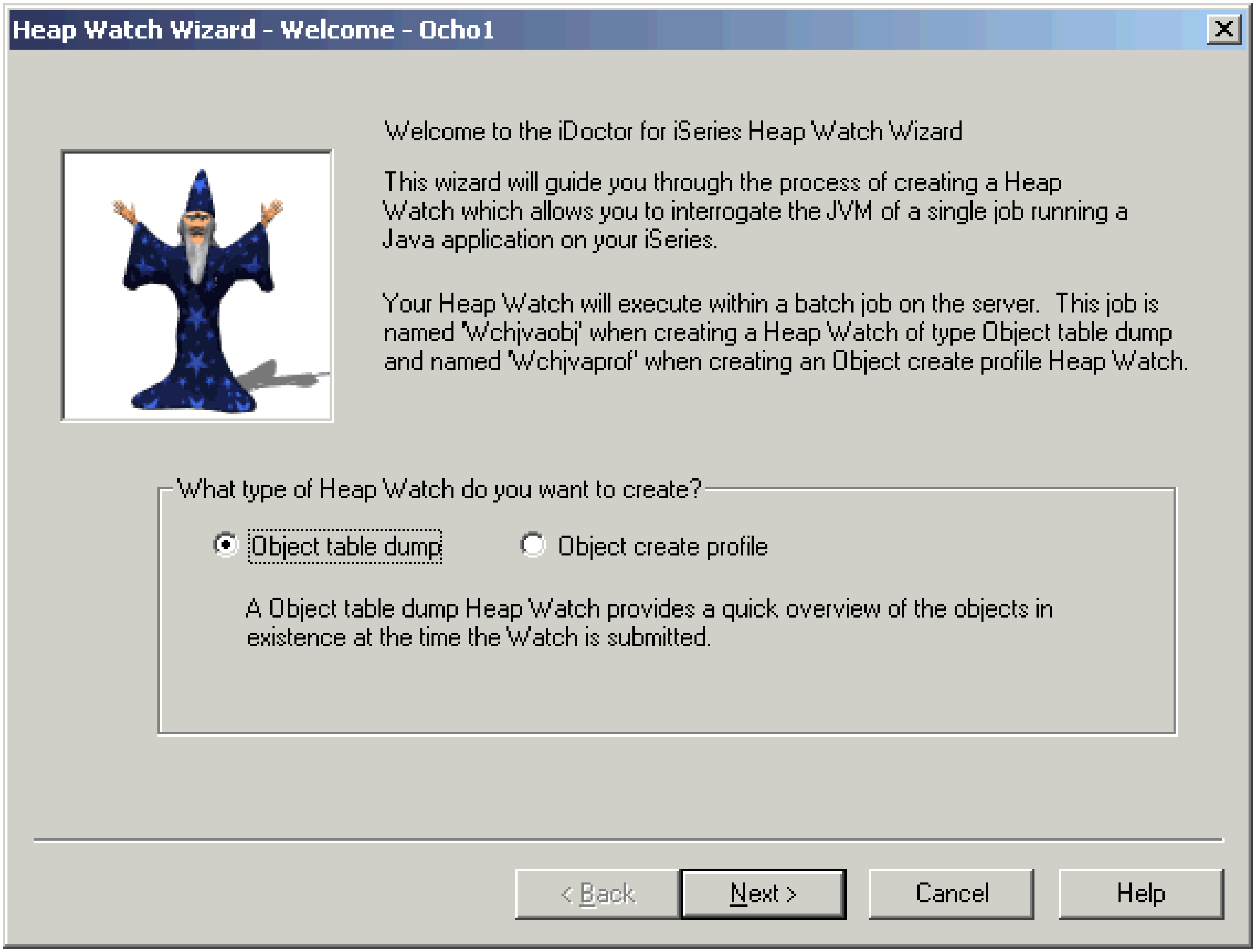
The Heap Watch Wizard guides you step by step through the process of creating the Heap Watch. Each page is covered in detailed within the next sections.

Table of Contents | Previous | Next

8.2.1 Welcome

The Welcome page in the Heap Watch Wizard introduces the user to the wizard and offers information about what the wizard will do. The page also explains the two types of Heap Watches and within what job name the Watches will run in.

On this page you must specify the type of Heap Watch to create.



Heap Watch Wizard - Welcome - Ocho1

Welcome to the iDoctor for iSeries Heap Watch Wizard

This wizard will guide you through the process of creating a Heap Watch which allows you to interrogate the JVM of a single job running a Java application on your iSeries.

Your Heap Watch will execute within a batch job on the server. This job is named 'Wchjvaobj' when creating a Heap Watch of type Object table dump and named 'Wchjvaprof' when creating an Object create profile Heap Watch.

What type of Heap Watch do you want to create?

● Object table dump ○ Object create profile

A Object table dump Heap Watch provides a quick overview of the objects in existence at the time the Watch is submitted.

[Heap Watch Wizard - Welcome Page]

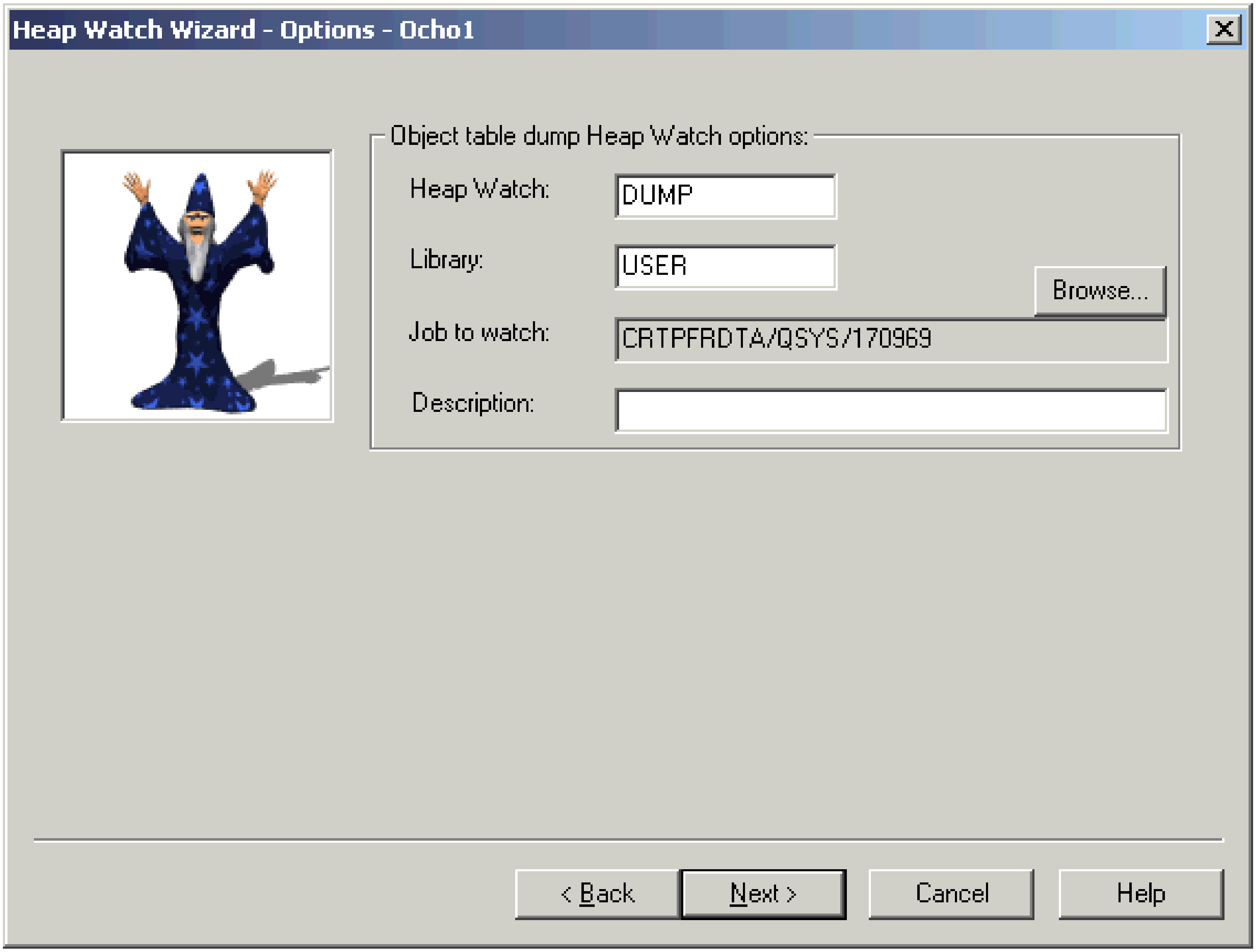
Table of Contents Previous Next

8.2.2 'Object table snapshot' Options page

Note: This page does not apply to 'object create profile' Heap Watches.

The following parameters are defined on this window:

Field	**Description**
Heap Watch	The up to 10 character member name that uniquely identifies the Heap Watch in the library. Files QPYRTJVMH1 and QPYRTJVMH2 will have contain this member after the Heap Watch is started.
Library	Name of the library to create the Heap Watch into.
Job to watch	Name of the job to watch. You can only watch one job/user/number per instance.
Description	50 byte description shown in the client when viewing a list of active/completed Heap Watches.


Heap Watch Wizard - Options - Ocho1

Object table dump Heap Watch options:

Heap Watch: DUMP

Library: USER

Job to watch: CRTPFRDTA/QSYS/170969

Description:

[Heap Watch Wizard - Object table snapshot Heap Watch options]

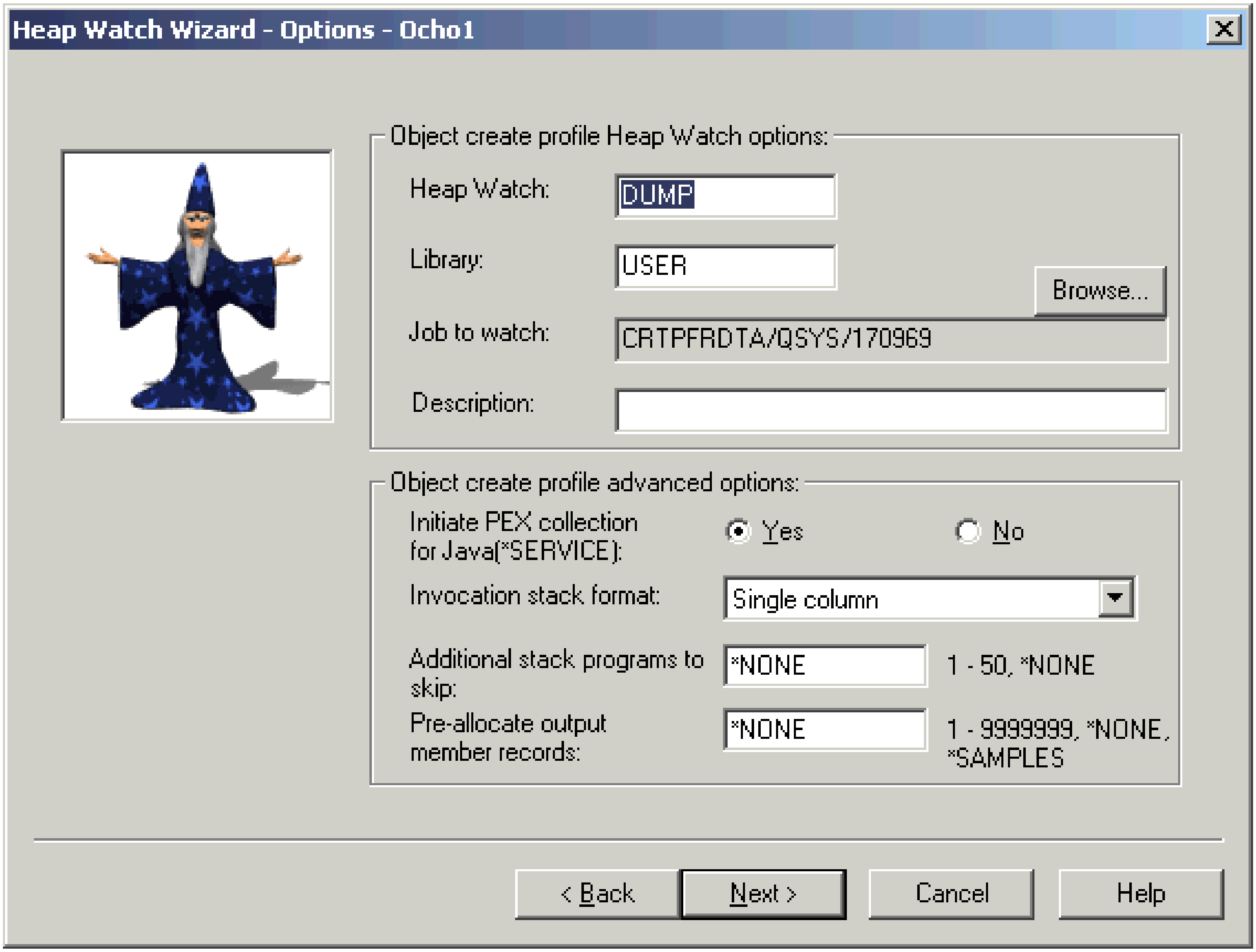
Table of Contents Previous Next

8.2.3 'Object create profile' Options page

Note: This page does not apply to 'object table snapshot' Heap Watches.

The following parameters are defined on this window:

Field	Description
Heap Watch	The up to 10 character member name that uniquely identifies the Heap Watch in the library. Files QPYRTJVMH1 and QPYRTJVMH2 will have contain this member after the Heap Watch is started.
Library	Name of the library to create the Heap Watch into.
Job to watch	Name of the job to watch. You can only watch one job/user/number per instance.
Description	50 byte description shown in the client when viewing a list of active/completed Heap Watches.
Initiate PEX collection for Java (*SERVICE)	Indicates whether a PEX collection should be started over the Job to watch during execution of the Heap Watch. The PEX collection will not collect data, but will turn a flag on in the job to make information retrieval for the profile possible. The PEX collection will be deleted automatically when the Heap Watch completes.
Invocation stack format	Determines if the call stack should be harvested into a single column in the database , or into multiple columns. Depending on this selection determines which file the stack information is written to. This will either be QPYRTJVMF2 (multiple columns) or QPYRTJVMF3 (single column).
Additional stack programs to skip	Only 5 stack entries are harvested due to space constraints. Because there may be many many entries on the call stack, you may want to use this option to throw away everything except each nth entry on the call stack. Note: You can only have one stack skip value setting used for any JVM instance at a time. This means that if two users are watching the JVM at the same time with different stack skip values, the actual value will likely toggle between the two because at the beginning of each interval the stack skip value is sent down to the JVM.
Pre-allocate output member records	Indicates how many records of data to preallocate before starting the Heap Watch. This can be used to remove a delay during sampling in allocating the space. However using this it is possible to allocate much more data than is actually needed.



Heap Watch Wizard - Options - Ocho1

Object create profile Heap Watch options:

Heap Watch: DUMP

Library: USER

Job to watch: CRTPFRDTA/QSYS/170969

Description:

Object create profile advanced options:

Initiate PEX collection for Java(*SERVICE): ● Yes ○ No

Invocation stack format: Single column

Additional stack programs to skip: *NONE 1 - 50, *NONE

Pre-allocate output member records: *NONE 1 - 9999999, *NONE, *SAMPLES

[Heap Watch Wizard - Object create profile Heap Watch options]

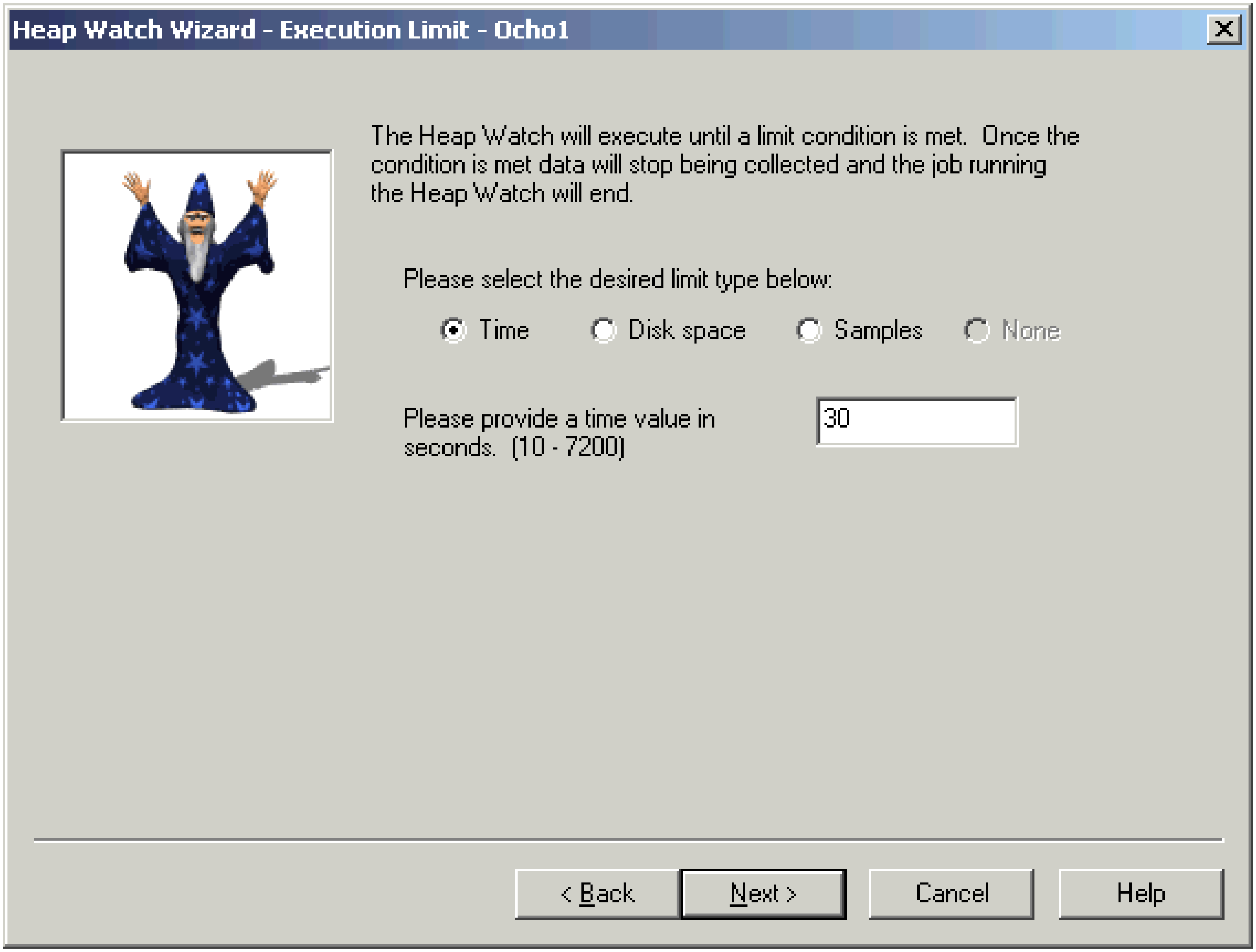
Table of Contents Previous Next

8.2.4 Execution limits page

This page lets you define the execution limit. This limit indicates when the Heap Watch will stop executing.

The following parameters are defined on this window:

Field	Description
Limit type	The type of limit that will end execution. The possible limit types is different depending on if you are creating a object table snapshot Heap Watch or a object create profile Heap Watch. A object table snapshot can have a time limit (seconds) or none. Because dumping the JVM can take a very long time depending on how many objects exist within it, you can use the time limit so the job will stop executing if it takes longer than n seconds. If the dump does not complete you will only get partial data though. If you don't want a time limit when running this type of Heap Watch select 'none' for the limit type. A object create profile can have a limit type of time, disk space (megabytes) or samples. When an object create profile runs it will create records as fast as possible, providing information about the objects created during each 'interval'. It's important to choose the desired limit type here because you may end up with many GBs of undesired data if not careful. Example: Setting a time limit of 60 seconds could easily generate 1 GB or more of data. It may be better to set a disk space limit or a limit to the number of samples if disk space is a concern.
Limit value	The value for the execution limit. Its meaning is totally dependent on the limit type. If limit type is 'time' then this value is in seconds. If limit type is 'disk space' then this value is in megabytes. If limit type is 'samples' then this value is the number of samples If limit type is 'none' then there is no limit value and this field is greyed out.



Heap Watch Wizard - Execution Limit - Ocho1

The Heap Watch will execute until a limit condition is met. Once the condition is met data will stop being collected and the job running the Heap Watch will end.

Please select the desired limit type below:

● Time ○ Disk space ○ Samples ○ None

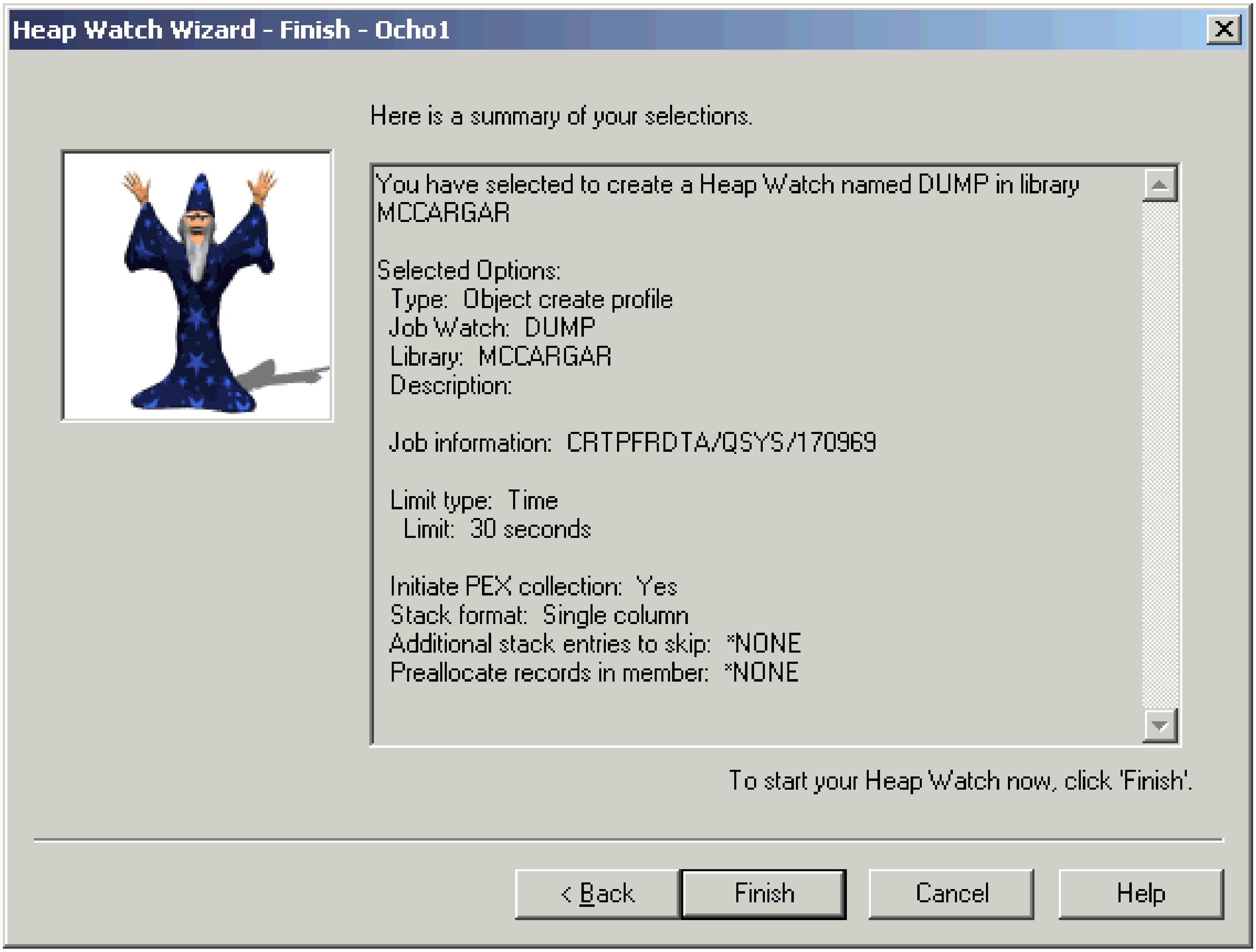
Please provide a time value in seconds. (10 - 7200) 30

[Heap Watch Wizard - Execution limits page]

Table of Contents | Previous | Next

8.2.5 Summary page

This page simply summarizes your selections. Clicking the 'Finish' button will start the Heap Watch and close the wizard.



Heap Watch Wizard - Finish - Ocho1

Here is a summary of your selections.

```
You have selected to create a Heap Watch named DUMP in library
MCCARGAR

Selected Options:
  Type:  Object create profile
  Job Watch:  DUMP
  Library:  MCCARGAR
  Description:

  Job information:  CRTPFRDTA/QSYS/170969

  Limit type:  Time
    Limit:  30 seconds

  Initiate PEX collection:  Yes
  Stack format:  Single column
  Additional stack entries to skip:  *NONE
  Preallocate records in member:  *NONE
```

To start your Heap Watch now, click 'Finish'.

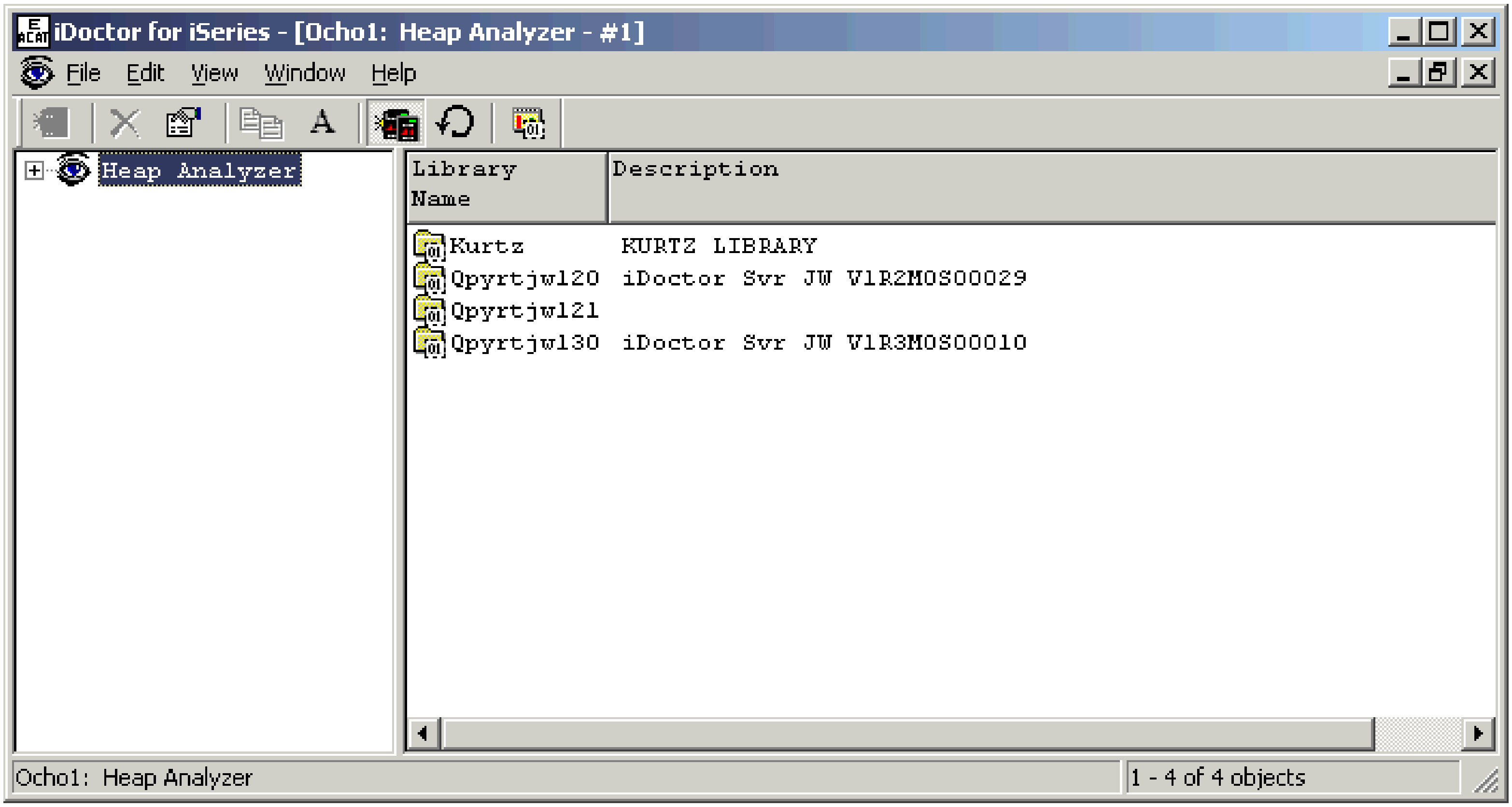
[Heap Watch Wizard - Summary page]

Table of Contents Previous Next

8.3 Libraries

The 'Heap Analyzer' folder contains a list of library folders, each representing a library on the iSeries system that contains Heap Analyzer database files. The list displays each library's name and description.

By clicking on a library in the tree you will see its contents (a list of Heap Watches).



Library Name	Description
Kurtz	KURTZ LIBRARY
Qpyrtjwl20	iDoctor Svr JW V1R2M0S00029
Qpyrtjwl21	
Qpyrtjwl30	iDoctor Svr JW V1R3M0S00010

Ocho1: Heap Analyzer

1 - 4 of 4 objects

[The Heap Analyzer component displaying a list of libraries containing Heap Analyzer data on system.]

Library Menu Options

The following menu options are available by right-clicking on a library in the tree/list view above.

Menu Item	Description
Explore	Displays the contents of the library (list of Heap Watches within the library) in the right pane of the tree/list window.
Select fields...	Brings up a window that lets you modify what fields are shown when displaying the contents of library. The contents of a library are Heap Watch objects so this menu lets you hide/display/reorder fields that are relevant to a Heap Watch.
Start Heap Watch...	This menu will open the Heap Watch Wizard so you can create a Heap Watch in the selected library.
Copy...	Allows you to copy the library's contents into a new library or into an existing one.
Save...	This option lets you save the library's contents into a save file.
Clear	This option clears a library (deletes all objects in the library).
Delete	Deletes the library.

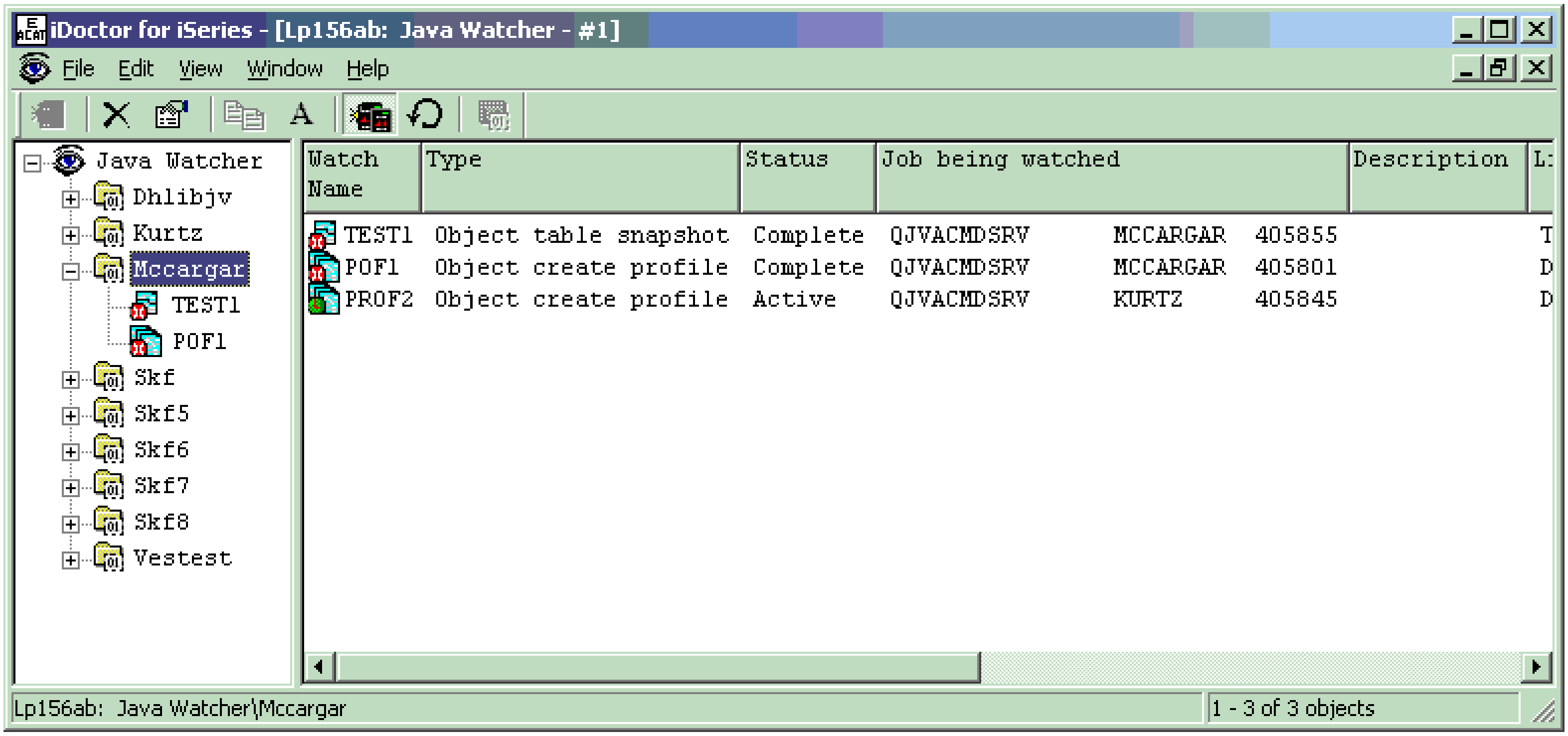
Rename	Renames the library.
Properties	Use this menu to display the library property pages. Basic information similar to that provided by DSPOBJD is available through these property pages. For more information on the library property pages click here ->

Table of Contents | Previous | Next

8.4 Heap Watches

Moving down the tree within each Library folder are zero or more Heap Watches that have been created (or are being created) within the current library. The icons showing a green clock indicate active Heap Watches and the icons with red stop signs indicate Heap Watches which have completed.

There are two types of Heap Watches: 'object table snapshot' and 'object create profile'. The type of watch will be shown in the 2nd column in the list by default (see example below).



Watch Name	Type	Status	Job being watched	Description	L:
TEST1	Object table snapshot	Complete	QJVACMDSRV MCCARGAR 405855		T
POF1	Object create profile	Complete	QJVACMDSRV MCCARGAR 405801		D
PROF2	Object create profile	Active	QJVACMDSRV KURTZ 405845		D

[Heap Analyzer displaying the list of Heap Watches within library 'Mccargar']

Heap Watch Status

Each Watch has a status field indicating whether it is currently running. You can also tell the status by the color of the icon: Green = active, Red = not active.

Heap Watch Fields

The list of Heap Watches displays the watch name, watch type, status, description as well as several additional fields.

Each Heap Watch in the list has a set of fields available which can be optionally reordered and displayed. To change the current field selections for the Heap Watch list, use the Select fields... menu from the library folder. A listing of the available fields and a short description is provided in the table below:

Field	Description
Watch name	Name of the Heap Watch. This name matches the member name used in the output files named QPYRTJVM* that exist on the system.
Type	The type of Heap Watch. Possible values are 'object table snapshot' and 'object create profile'.
Status	The status field indicates the status of the job on the system running the Heap Watch (if active) or if not active the status indicates whether the watch can be analyzed and/or has data available.
Job being watched	The name of the job this Heap Watch is watching.

Description	50 byte description of the Heap Watch specified at creation time.
Limit type	Execution limit type. Indicates when the Watch will stop executing. Possible values are: time, disk space, or samples.
Limit value	The value of the execution limit.
Job running the watch	The name of the job running the Heap Watch. This will typically be named WCHJVAOBJ for an object table snapshot Heap Watch and WCHJVAPROF for an object create profile Heap Watch. The job running the watch will be an interactive job if the user created the watch using the green screen command WCHJVA.
Start time	The time the Heap Watch started.
End time	The time the Heap Watch ended (stopped collecting data).
Stack format	Determines if the call stack should be harvested into a single column in the database , or into multiple columns. Depending on this selection determines which file the stack information is written to. This will either be QPYRTJVMF2 (multiple columns) or QPYRTJVMF3 (single column).
Additoinal stack entries to skip	Indicates how many entries are skipped if any per every saved call stack entry. Example: A value of 3 means, skip 3 entries, save/write 1 entry, skip 3 entries, save/write 1 entry, etc.

Heap Watch Menu Options

The table below outlines the different types of operations that may be performed on a Heap Watch.

Menu Item	**Description**
Explore	Displays the contents of the Heap Watch (its reports) in the right pane of the tree/list window.
Record Quick View	Displays the fields for a single Heap Watch in a list view vertically for easier viewing. Not available from the tree side, only the list side.
Compare...	Runs a comparison analysis over two object table snapshots Heap Watches.
Delete...	Delete a Heap Watch. Select multiple Heap Watches in order to delete more than one watch at a time.

Table of Contents | Previous | Next

8.4.1 Object table snapshot comparison

Heap Analyzer features an analysis over two object table snapshot Heap Watches. This analysis identifies heap differences between snapshot A and snapshot B. **Note:** This documentation refers to snapshot/watch/collection interchangeably.

The analysis outputs consist three reports:

1. Shows the object count and object/heap size differences between snapshot A and snapshot B.
2. Displays information about snapshot A.
3. Displays information about snapshot B.

By running the analysis from the client the command QPYRTJW/HCOMPARE is executed under the covers. This command creates the output files J_DIFF, J_OBJ1, J_OBJ2 in library QTEMP within the current job (in this case the QZDASOINIT job that the client is using). The member name is randomly generated based on the current timestamp.

After the command finishes executing the reports are shown in the Data Viewer. Once opened the reports may be queried but any queries created cannot be saved. You cannot copy the data or save it via the client. If you want to save these comparisons, you must use the HCOMPARE command and copy the files from QTEMP within a green screen session.

To run this analysis use the 'Compare...' menu found by right-clicking on a Heap Watch of type 'object table snapshot.'