IBM Tivoli Workload Scheduler version 8.2.1
Warehouse Enablement Pack version 1.1.1
Implementation Guide
for Tivoli Data Warehouse, versions 1.2 and 1.3

Second Edition (March 2006)

This edition applies to version 1.2 and 1.3 of Tivoli Data Warehouse and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

1 About this document

This document describes the warehouse enablement pack, version 1.1.1, for Tivoli Workload Scheduler version 8.2.1. This warehouse pack runs on Tivoli Data Warehouse versions 1.2 and 1.3. It covers the following topics:

- Installing and configuring the warehouse pack
- The data flow and data structures used by the warehouse pack

With this warehouse pack, you can enable Tivoli Workload Scheduler to save scheduling execution history data into the Tivoli Data Warehouse database in order to be retrieved and correlated with other enterprise level data to produce summaries and statistics about the scheduling activity in your enterprise.

1.1 Related documentation

You can access many Tivoli publications online using the Tivoli Information Center, which is available on the Tivoli Customer Support Web site:

http://www-3.ibm.com/software/sysmgmt/products/support/

The following sets of documentation are available to help you understand, install, and manage this warehouse pack:

- Tivoli Data Warehouse
- IBM DB2, DB2 Data Warehouse Center, and DB2 Warehouse Manager

1.1.1 IBM DB2, DB2 Data Warehouse Center, and DB2 Warehouse Manager library

The DB2 library contains important information about the database and data warehousing technology provided by IBM DB2, DB2 Data Warehouse Center, and DB2 Warehouse Manager. Refer to the DB2 library for help in installing, configuring, administering, and troubleshooting DB2, which is available on the IBM Web site:

http://www-3.ibm.com/software/data/db2/library/

After you install DB2, its library is also available on your system.

The following DB2 documents are particularly relevant for people working with Tivoli Data Warehouse:

- *IBM DB2 Universal Database for Windows Quick Beginnings*, GC09-2971

  Guides you through the planning, installation, migration (if necessary), and setup of a partitioned database system using the IBM DB2 product on Microsoft Windows.

- *IBM DB2 Universal Database for UNIX Quick Beginnings*, GC09-2970

  Guides you through the planning, installation, migration (if necessary), and setup of a partitioned database system using the IBM DB2 product on UNIX.

- *IBM DB2 Universal Database Administration Guide: Implementation*, SC09-2944

  Covers the details of implementing your database design. Topics include creating and altering a database, database security, database recovery, and administration using the Control Center, a DB2 graphical user interface.

- *IBM DB2 Universal Database Data Warehouse Center Administration Guide*, SC26-9993

  Provides information on how to build and maintain a data warehouse using the Data Warehouse Center.

- *IBM DB2 Warehouse Manager Installation Guide*, GC26-9998

  Provides the information to install the following Warehouse Manager components: Information Catalog Manager, warehouse agents, and warehouse transformers.

- *IBM DB2 Universal Database and DB2 Connect Installation and Configuration Supplement*, GC09-2957

  Provides advanced installation considerations and guides you through the planning, installation, migration (if necessary), and set up a platform-specific DB2 client. Once the DB2 client is installed, you then configure communications for both the client and server, using the DB2 GUI tools or the Command Line Processor. This supplement also contains information on binding, setting up communications on the server, the DB2 GUI tools, DRDA(tm) AS, distributed installation, the configuration of distributed requests, and accessing heterogeneous data sources.

- *IBM DB2 Universal Database Message Reference Volume 1*, GC09-2978 and *IBM DB2 Universal Database Message Reference Volume 2*, GC09-2979

  Lists the messages and codes issued by DB2, the Information Catalog Manager, and the Data Warehouse Center, and describes the actions you should take.

2 Overview

The following sections provide an overview of Tivoli Data Warehouse and the IBM Tivoli Workload Scheduler warehouse pack.

2.1 Overview of Tivoli Data Warehouse

Tivoli Data Warehouse provides the infrastructure for the following:

- Extract, transform, and load (ETL) processes through the IBM DB2 Data Warehouse Center tool
- Schema generation of the Central Data Warehouse
- Historical reporting

As shown in Figure 1, Tivoli Data Warehouse consists of a centralized data store where historical data from many management applications can be stored, aggregated, and correlated.

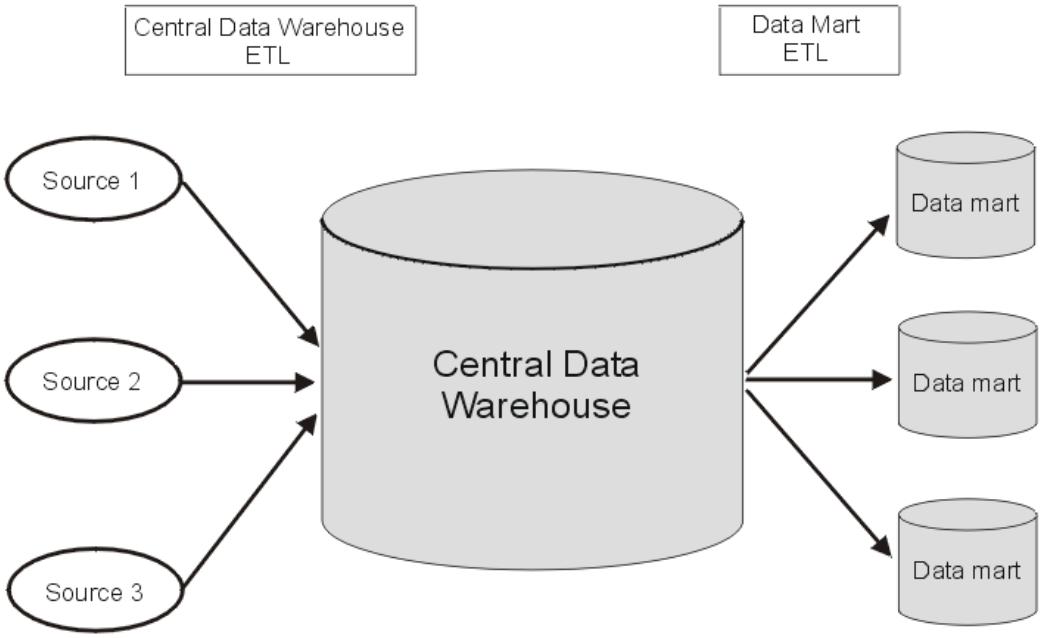


Figure 1. Tivoli Data Warehouse overview

The *Central Data Warehouse* uses a generic schema that is the same for all applications. As new components or new applications are added, more data is added to the database; however, no new tables or columns are added in the schema.

A *data mart* is a subset of a data warehouse that contains data tailored and optimized for the specific reporting needs of a department or team.

The *Central Data Warehouse ETL* reads the data from the operational data stores of the application that collects it, verifies the data, makes the data conform to the schema, and places the data into the Central Data Warehouse.

The *data mart ETL* extracts a subset of data from the Central Data Warehouse, transforms it, and loads it into one or more star schemas, which can be included in data marts to answer specific business questions.

A program that provides these ETLs is called a *warehouse enablement pack*, referred to as a *warehouse pack* in the rest of this document.

The ETLs are typically scheduled to run periodically, usually during non-peak hours. If an ETL encounters data that it cannot correctly transform, it creates an entry in an exception table.

2.2 Overview of Tivoli Workload Scheduler warehouse pack

Tivoli Workload Scheduler helps in managing the workload on all the distributed and z/OS systems by planning and organizing every phase of workload production; you use it to plan which jobs to run, resolve interdependencies, launch and track each job.

The Tivoli Workload Scheduler master component creates a production plan containing the work to be done in a day and then distributes this plan to the Tivoli Workload Scheduler agents that perform it. The plan is updated with the status of all the jobs that run in the Tivoli Workload Scheduler environment. At the end of the production day, the plan is archived and a new plan is created and distributed to the Tivoli Workload Scheduler agents.

The Tivoli Workload Scheduler production plan is physically mapped into a binary file (named Symphony) that contains the scheduling activities to be performed in the next 24 hours. When a new production plan is created, a new Symphony file is created, and all the activities that have not been completed in the previous plan are carried forward into the new Symphony file. The old Symphony file plan is archived in the schedlog Tivoli Workload Scheduler directory.

The archived Symphony files contain the history for all the jobs that have run during the previous production plans. The Tivoli Workload Scheduler **archiver** processes the archived Symphony files and fills the following flat files:

- Cpus
- Jobs
- Scheds

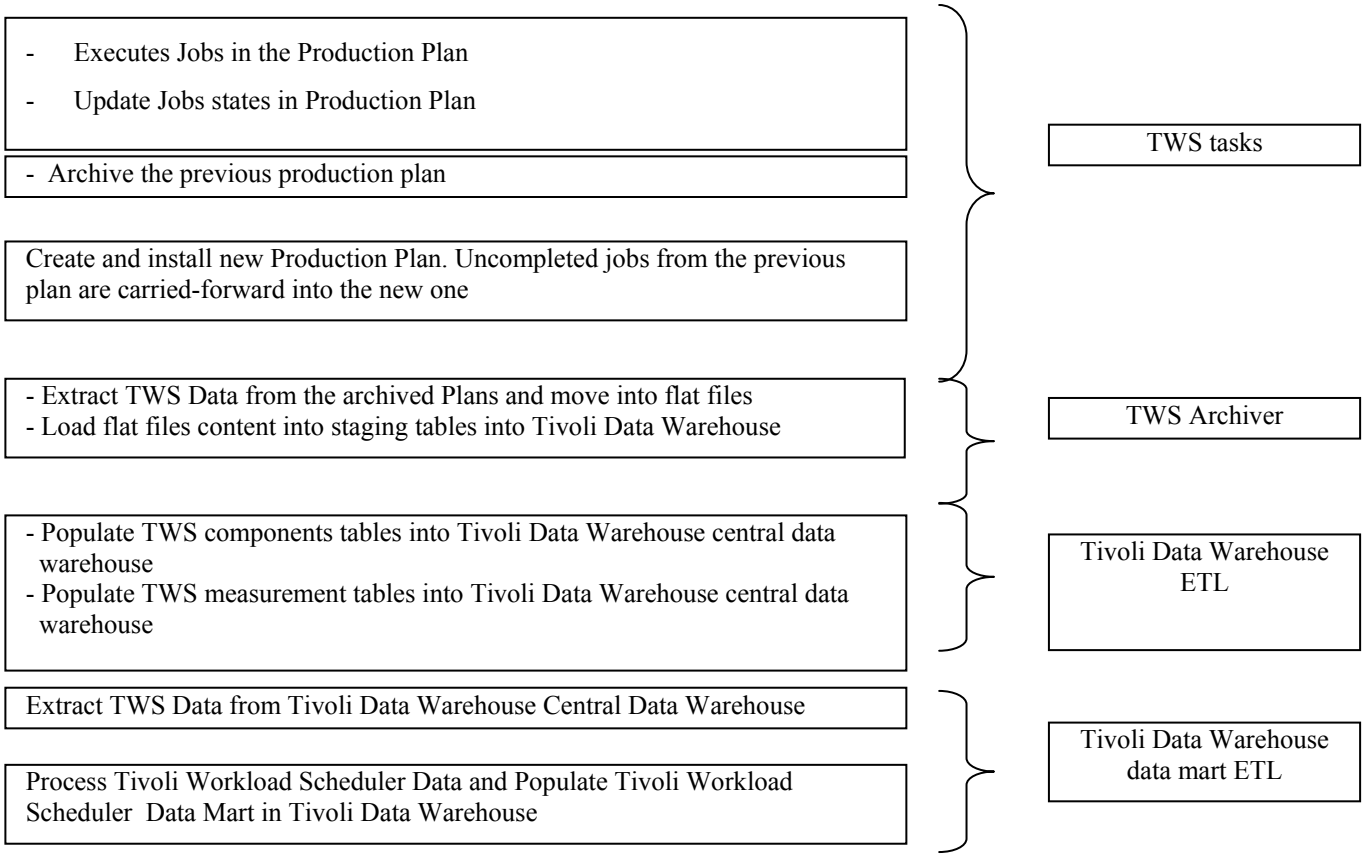
The **import** command then processes these flat files and imports the data into the following staging CDW tables contained in the AWS schema:

- TWS_WORKSTATION_P
- TWS_JOB_P
- TWS_JOBSTREAM_P

The tws_launch_archiver perl script, provided with Tivoli Workload Scheduler, calls both the **archiver** process and the **import** command. The tws_launch_archiver script can be scheduled on the Tivoli Workload Scheduler Master Domain Manager using Tivoli Workload Scheduler.

To run the tws_launch_archiver perl script use Perl level V5.8.0, contained in CD_3 /Add-On/Windows/Perl5 dir, or higher.
The tws_launch_archiver perl script can be run repeatedly, typically after a new Tivoli Workload Scheduler plan is created, and it can be scheduled to run using Tivoli Workload Scheduler itself. In this way, the Central Data Warehouse is updated with the new Tivoli Workload Scheduler data based on the jobs and job streams that have been run by Tivoli Workload Scheduler during the running of the previous Production Plan.

The picture below describes the typical Tivoli Workload Scheduler processing and the interaction with the Tivoli Data Warehouse.



The Tivoli Workload Scheduler warehouse pack for Tivoli Data Warehouse, version 1.2 and 1.3, is an implementation of the IBM DB2 Data Warehouse Center based on the data collected by Tivoli Workload Scheduler, version 8.2.1. As such, it provides a Data Warehouse Subject Area named AWS_Tivoli_Workload_Scheduler_v1.1.1_Subject_Area.

Within this Subject Area, there is the following process:

- AWS_c10_ETL_Process.

The Tivoli Workload Scheduler warehouse pack will extract the scheduling data only from the archived Production plans and not from the currently executing production plan.

3 Installing and configuring

This section describes the information about installing and configuring the warehouse pack.

3.1 Prerequisites

Before installing the Tivoli Workload Scheduler warehouse pack, the following software must be installed:

- Tivoli Workload Scheduler, version 8.2.1 at its latest Fix Pack level
- Perl level V5.8.0 (supplied with the Tivoli Workload Scheduler 8.2.1 on the Add-On directory of CD 3) or higher
- Tivoli Data Warehouse version 1.2 Fix Pack 2 or version 1.3, and prerequisites

You can obtain the Tivoli Data Warehouse Fix Pack from the Tivoli Data Warehouse Web site at the following address:

http://www.ibm.com/software/sysmgmt/products/support/TivoliDataWarehouse.html

Click the Downloads link in the Self help section.

3.2 Supported hardware and software

Tivoli Workload Scheduler warehouse pack, version 1.1.1, supports Tivoli Workload Scheduler, version 8.2.1. It supports all versions of databases supported by Tivoli Data Warehouse.

For information about the hardware and software requirements of Tivoli Data Warehouse, see the *Tivoli Data Warehouse Release Notes*.

3.3 Limitations

This warehouse pack must be installed using the user "db2". If that is not the user name used when installing the Tivoli Data Warehouse core application, you must create a user temporary tablespace for use by the installation program. The user temporary tablespace that is created in each Central Data Warehouse database and data mart database during the installation of Tivoli Data Warehouse is accessible only to the user that performed the installation.

If you are installing the warehouse pack using the same database user that installed Tivoli Data Warehouse, or if your database user has access to another user temporary tablespace in the target databases, no additional action is required.

If you do not know the user name that was used to install Tivoli Data Warehouse, you can determine whether the tablespace is accessible by attempting to declare a temporary table while connected to each database as the user that will install the warehouse pack. The following commands are one way to do this:

```
db2 "connect to TWH_CDW user installing_user using password"
db2 "declare global temporary table t1 (c1 char(1))with replace on commit preserve
rows not logged"
db2 "disconnect TWH_CDW"
db2 "connect to TWH_MART user installing_user using password"
db2 "declare global temporary table t1 (c1 char(1))with replace on commit preserve
rows not logged"
db2 "disconnect TWH_CDW"
```

Where:

installing_user Identifies the database user that will install the warehouse pack.

password Specifies the password for the installing user.

If the declare command is successful, the specified database user can install the warehouse pack. No additional action is required.

If the declare command fails, run the following DB2 commands to create a new tablespace for the installation in both the Central Data Warehouse database and data mart databases:

```
db2 "connect to TWH_CDW user installing_user using password"
db2 "create user temporary tablespace usertmp2 managed by system using ('
usertmp2)"
db2 "disconnect TWH_CDW"
db2 "connect to TWH_MART user installing_user using password"
db2 "create user temporary tablespace usertmp3 managed by system using ('
usertmp3')"
db2 "disconnect TWH_MART"
```

Where:

installing_user	Identifies the database user that will install the warehouse pack.
password	Specifies the password for the installing user.

3.4 Database sizing considerations

The following formula will give approximate indications concerning the size that has to be considered for the Central Data Warehouse for this package:

1. Consider the component tree (Table Comp) from the root to the farthest leaf; then give a sample of repeating values for each component based on your environment.
2. Consider the number of measurements (msmtTyp table) applicable for each component type (Table MsmtRul).
3. Multiply the above by the number of days before the data will be deleted (table Prune_msmt_Control), which default value for daily measurements is 365.
4. Multiply the above for the record length of Msmt table (89 bytes).

#components x #Msmt_Types x PruneDays x Msmt reclength = msmt_TableSize for one component type

where:

- #components: estimated number of different instances of a component type, added for all systems for which you are collecting data (for instance, if you have two Tivoli Workload Scheduler networks and you estimate 700 jobs in one network and 300 in the second one, your number of components will be 1000)
- #Msmt_Types: number of measurement types applicable for this measurement type (from the MsmtRul table)
- PruneDays: the number of days from the field PMsmtC_Age_inDays in the Prune_Msmt_Control table, before the old data are deleted.
- Msmt reclength: record length of Msmt table (89 bytes)
- Msmt_TableSize: estimated size for Msmt table.

For example, if in your environment you estimate 1000 components for a particular component type for which the MsmtRul table indicates 7 different measurement types, you get:

1000 x 7 x 365 x 89 bytes = 227.395.000 bytes ~ 216 MBs

3.5 Data sources and targets

The following table shows the corresponding Tivoli Workload Scheduler data source locations, the Central Data Warehouse database targets, together with the component types and measurement types managed by the warehouse pack.

Tivoli Workload Scheduler data source **Symphony record**	**Tivoli Workload Scheduler** **Field or Source formula**	**Tivoli Data Warehouse** **CompTyp_Cd (C)** **MsmtTyp_Nm (M)**
Workstation	Master CPU name	(C) “SCHED_ENGINE”
	Name	(C) “SCHED_WORKSTATION”
	Node	(C) “IP_HOST”
Schedule	Name	(C) ‘SCHED_JOBSTREAM
	SUM(status) WHERE status = SUCCESSFUL	(M) Number of Successful Runs
	SUM(status) WHERE status = ABEND or FAILED	(M) Number of Unsuccessful Runs
	Elapsed_time	(M) “Duration”
	Date_started- Sched_date	(M) “Start Delay”
Job definition	Name	(C) ‘SCHED_JOB’
	Status	(M) Run States counters and total (MsmtTyp_ID 1 to 5)
	Elapsed_time	(M) “Duration”
	Date_started – sched_date	(M) “Start Delay”
	Cpu_time	(M) 'CPU Time'

3.6 Installation of the warehouse pack

To install the Tivoli Workload Scheduler warehouse pack, do the following:

1. Ensure that you have installed Tivoli Workload Scheduler.
2. Ensure that you have installed all the prerequisite product patches.
3. Ensure that Tivoli Data Warehouse is installed. Refer to the documentation available with the Tivoli Data Warehouse if you need to install it.
4. Install the Tivoli Workload Scheduler warehouse pack. Refer to the documentation available with the Tivoli Data Warehouse.
5. Perform the post-installation steps as described in the 3.7 post-installation procedures.

Note: If the Tivoli Workload Scheduler Master Domain Manager and the Tivoli Data Warehouse control server reside on two different machines, a DB2 client must be installed on the Tivoli Workload Scheduler Master Domain Manager, for the **import** process to upload data to the Central Data Warehouse database.

3.7 Post-installation procedures

After you have installed the Tivoli Workload Scheduler warehouse pack, set the properties including the User ID, Password, and Verify Password, for the following Warehouse Sources and Targets as described in the getting started information in *Installing and Configuring Tivoli Data Warehouse*:

- CDW_TWH_CDW_Target
- TWS_TWH_CDW_Target
- TWS_TWH_MART_TARGET

These properties are in the Database page.

- In the User ID field, type the user ID used to access the Tivoli Data Warehouse Central Data Warehouse database. The default value is db2admin.
- In the Password field, type the password used to access the Central Data Warehouse database.
- Do not change the value of the Data Source field. It must be TWH_CDW.

If you want to schedule the AWS_c10_ETL_Process you can use one of the following products:

- Tivoli Data Warehouse. Use the procedure to schedule ETL processes described in "*Installing and Configuring Tivoli Data Warehouse"*
- Tivoli Workload Scheduler. Create a Tivoli Workload Scheduler job stream that schedules the following:
  - The archiver process
  - The DB2 process that imports data into staging tables

4 Maintenance

4.1 Backing up and restoring

No special requirements are needed for backing up or restoring data.

4.2 Pruning

Removing old data from the Msmt table is implemented with a combination of triggers and warehouse processes; CDW_c05_Prune_and_Mark_Active_Process is a process in the CDW Subject Area. You schedule when you want to run the process: weekly, monthly, daily.

The Prune_Msmt_Control table controls what data gets pruned. The default value set by Tivoli Workload Scheduler is 365 days. You can modify the value by running the following SQL statement, where *X* is a date duration whose format is *yyyymmdd* (for example: *X* = 00000108 for 0000 years, 01 month, 08 days).

UPDATE TWG.Prune_Msmt_Control

SET PMsmtc_age_in_days = X WHERE TmSum_Cd = 'H' AND Msrc_Cd = 'AWS' ;

5 ETL processes

This warehouse pack has the following processes.

5.1 AWS_c10_ETL_Process

The **AWS_c10_ETL_Process** processes the Tivoli Workload Scheduler data from the CDW staging tables and loads it into the Central Data Warehouse and in the Data Mart tables. The procedure that Tivoli Workload Scheduler uses to load the data into the CDW staging tables is described in the Overview section.

The AWS_c10_ETL_Process extracts only the jobs and job streams that have been run since the last time it ran. Therefore, once a job or a job stream record has been moved into the Central Data Warehouse, it should not be extracted a second time. This function of extracting only new data from a data source is referred to as "Extract Control". More details can be found in the sub-section Incremental extraction.

Another important Central Data Warehouse feature, that is used to control the amount of data extracted from the Tivoli Workload Scheduler plans, is the Prune Measurement Control feature. This feature automatically prunes measurement data when it is older than a defined age. As supplied by default, Tivoli Workload Scheduler measurements are maintained in the Central Data Warehouse for one year. The Prune Measurement Control feature is driven by the *PMsmtc_age_in_days* parameters. Details for altering this value are provided in the section "Pruning".

The AWS_c10_ETL_Process is composed of the following steps that you can run either manually following the order indicated below or automatically by scheduling them using Tivoli Data Warehouse. Each step must start if the previous one is successful.

1. **AWS_c10_s040_LoadComp**

   This step loads the Tivoli Workload Scheduler data from the Tivoli Workload Scheduler staging tables and populates the COMP table in Tivoli Data Warehouse Central Data Warehouse.

2. **AWS_c10_40_LoadCompAttr**

   This step loads the Tivoli Workload Scheduler data from the Tivoli Workload Scheduler staging tables and populates the COMPATTR table in Tivoli Data Warehouse Central Data Warehouse.

3. **AWS_c10_s040_LoadCompReln**

   This step loads the Tivoli Workload Scheduler data from the Tivoli Workload Scheduler staging tables and populates the COMPRELN table in Tivoli Data Warehouse Central Data Warehouse

4. **AWS_c10_s50_LoadMsmt**

   This step loads the Tivoli Workload Scheduler data from the Tivoli Workload Scheduler staging tables and populates the MSMT table in Tivoli Data Warehouse Central Data Warehouse.

5. **AWS_m05_s010_martTWS**

   This step loads the Tivoli Workload Scheduler data from the Tivoli Data Warehouse Central Data Warehouse database to the Tivoli Data Warehouse Mart database.

The running of each step produces a log file as described in the log file information in *Installing and Configuring Tivoli Data Warehouse*. Analyze these log files and verify that no errors occur. If an error occurs correct it and run the step that failed again.

5.2 Incremental extraction

The incremental extraction is a way to automatically extract only the IBM Tivoli Workload Scheduler data that is not present in the Central Data Warehouse database.

The value in EXTCTL_TO_DTTM column of the Twg.Extract_Control table determines which Job and Job Stream records will be extracted from the Tivoli Workload Scheduler tables by the AWS_c10_ETL_Process. The record to consider is the one having "AWS_Stage_TWS_Archive" value in the column EXTCTL_TARGET.

The incremental extraction for Tivoli Workload Scheduler will be handled using the timestamps of the archived Symphony files. In fact any archived Symphony file is named by Tivoli Workload Scheduler with the following naming convention:

M-yyyymmddhhmmss where yyyy are four digits representing the year, mm represents the month, dd the day of the month, and hhmmss the time stamps in hours-minute-seconds.

The ETL process will save in the ExtCtl_From_DtTm field the date and time of the last ETL run, so that next time, only the archived Symphony files that have been created after the last ETL run will be processed by the new ETL run.

To view the ExtCtl_From_DtTm value, run the following SQL command:

SELECT ExtCtl_From_DtTm FROM TWG.Extract_Control

WHERE ExtCtl_Target='AWS_Stage_TWS_Archive'

This value represents the latest Job or jobstream execution date/time that was last extracted from the Tivoli Workload Scheduler plan tables. By default, this value is initialized to 1970-01-01-00.00.00.000000, and it is updated automatically after each run of the AWS_c10_ETL_Process so that at the next run of the process only jobs and jobstreams executed by Tivoli Workload Scheduler since the preceding run of the process will be extracted.

This value should not be changed manually as altering it could result in either missing scheduling data (setting the value to a later timestamp) or duplication of data (setting the value to an earlier timestamp). All timestamps in the Central Data Warehouse are stored in Universal Time Coordinated (UTC). If this value needs to be reset for any reason, you must use the aws_clean_and_reset.sql script. This script is described in detail in the next section.

5.3 Additional scripts

The following additional scripts, located in the apps/aws/v111/misc directory under the directory where Tivoli Data Warehouse (TWH_TOPDIR) is installed, are provided with this warehouse pack:

- **aws_clean_and_reset.db2**

  The aws_clean_and_reset.db2 script removes all the Central Data Warehouse (TWH_CDW) measurements, components, component attributes, and component relationships associated with Tivoli Workload Scheduler data. It also removes any Tivoli Workload Scheduler data that might exist in any staging table used by the Central Data Warehouse ETL process. It then resets the Extract Control timestamp (ExtCtl_From_DtTm, ExtCtl_To_DtTm) back to 1970-01-01-00.00.00.000000.

- **aws_clean_mart.db2**

  The aws_clean_mart.db2 script removes all Tivoli Workload Scheduler Mart data (TWH_MART).

To clean out any Tivoli Workload Scheduler data, both the above scripts you must run the scripts in the following order:

1. **aws_clean_and_reset.db2**
2. **aws_clean_mart.db2**

Note: These scripts delete all the Tivoli Workload Scheduler data from the Central Data Warehouse and the Mart databases. Perform a backup before running the scripts if you want to save any data.

These scripts can be useful when the Central Data Warehouse ETL has been running in a Tivoli Workload Scheduler test network in a testing environment. Before moving from a testing environment to a production environment, run these scripts to remove all the test data and reset the extract control value.

These scripts can also be run to recover from a database corruption. Ideally, you should ensure that the Central Data Warehouse (TWH_CDW) and Mart (TWH_MART) databases are backed up to allow proper recovery. However, in an extreme situation, providing you have maintained a Tivoli Workload Scheduler History Plan, you can use these scripts and then rerun the ETLs against the Tivoli Workload Scheduler History tables to re-create the data in the Central Data Warehouse and Mart databases.

6 Central Data Warehouse information

Before reading this section, read about the generic schema for the Tivoli Data Warehouse Central Data Warehouse, which is described in *Enabling an Application for Tivoli Data Warehouse*. That document defines the content of each table and explains the relationships between the tables in this document.

Shaded columns in the following tables are translated. These columns are also marked with an asterisk (*) after the column name.

In this section, the sample data shown below is used to illustrate how the event data will be used to implement the generic Central Data Warehouse schema:

TWSNetw	Domain	Hostname	WorkSt	Stream	Job	State	PlnStart	AcStart	Duration	Cpu time
ProdMaster	Account	Abc.abc.com	Dm1	Payroll		abend	10:00:00	10:00:05		
ProdMaster	Account	db2.abc.com	SAPXA1	Payroll	Extract	abend	10:00:00	10:12:01	01:31:00	00:22:00
ProdMaster	Finance	Pay.xyz.com	Fta1	Payroll	CalcPayroll	succ	12:00:00	12:00:30	00:00:23	00:00:07
ProdMaster	Dev	aix1.abc.com	cmvc_dev	Build		Succ	15:30:00	15:38:22		
ProdMaster	Dev	aix1.abc.com	cmvc_dev	Build	Extract	succ	15:30:00	15:45:10	00:25:01	00:11:00
ProdMaster	Dev	aix1.abc.com	Build_dev	Build	MakeAll	succ	14:20:00	14:21:59	02:00:01	00:55:00

6.1 Component configuration

The following sections describe the component configuration.

6.1.1 Component type (table CompTyp)

CompTyp_Cd CHAR(17)	CompTyp_Parent_Cd CHAR(17)	CompTyp_Nm * VARCHAR(120)	CompTyp_Strt_DtTm TIMESTAMP	CompTyp_End_DtTm TIMESTAMP
IPHOST	NULL	IP Host	2002-01-01-00.00.00.000000	9999-01-01-00.00.000000
IP_INTERFACE	NULL	IP Interface	2002-01-01-00.00.00.000000	9999-01-01-00.00.000000
SCHED_ENGINE	NULL	Workload Scheduler Master	2002-01-01-00.00.00.000000	9999-01-01-00.00.000000
SCHED_WORKSTATION	NULL	Workload Scheduler Workstation	2002-01-01-00.00.00.000000	9999-01-01-00.00.000000
SCHED_JOB	NULL	Workload Scheduler Job	2002-01-01-00.00.00.000000	9999-01-01-00.00.000000
SCHED_JOBSTREAM	NULL	Workload Scheduler Jobstream	2002-01-01-00.00.00.000000	9999-01-01-00.00.000000
* This column is translated.				

6.1.2 Component (table Comp)

Comp_ID INTEGER	CompTyp_Cd CHAR (17)	Centr_Cd CHAR (6)	Cust_ID INTEGER	Comp_Corr_ID INTEGER	Comp_Nm VARCHAR (254)	Comp_Corr_Val VARCHAR (254)	Comp_Strt_DtTm TIMESTAMP	Comp_End_DtTm TIMESTAMP	Comp_Ds VARCHAR (254)
1	SCHED_ENGINE	CDW	1		ProdMaster		2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000	Master
2	IP_HOST	CDW	1		abc.abc.com		2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000	IP Host
3	SCHED_WORKSTATION	CDW	1		DM1	SCHED_ENGINE – ProdMaster ,IP_HOST - abc.abc.com	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000	Workstation
4	IP_HOST	CDW	1		db2.abc.com		2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000	IP Host
5	SCHED_WORKSTATION	CDW	1		SAPXA1	SCHED_ENGINE – ProdMaster ,IP_HOST - db2.abc.com	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000	Workstation
6	IP_HOST	CDW	1		Pay.xyz.com		2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000	IP Host
7	SCHED_WORKSTATION	CDW	1		FTA1	SCHED_ENGINE – ProdMaster ,IP_HOST - Pay.xyz.com	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000	Workstation
8	IP_HOST	CDW	1		aix1.abc.com		2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000	IP Host
9	SCHED_WORKSTATION	CDW	1		Cmvc_dev	SCHED_ENGINE – ProdMaster ,IP_HOST - aix1.abc.com	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000	Workstation
10	SCHED_WORKSTATION	CDW	1		Build_dev	SCHED_ENGINE – ProdMaster ,IP_HOST - aix1.abc.com	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000	Workstation
11	SCHED_JOBSTREAM	CDW	1		Payroll	SCHED_ENGINE – ProdMaster	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000	Job Stream
12	SCHED_JOB	CDW	1		Extract	SCHED_ENGINE – ProdMaster ,SCHED_WORKSTATION – SAPXA1 ,SCHED_JOBSTREAM - Payroll	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000	Job
13	SCHED_JOB	CDW	1		CalcPayroll	SCHED_ENGINE – ProdMaster ,SCHED_WORKSTATION – FTA1 ,	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000	Job

						CHED_JOBSTREAM - Payroll			
14	SCHED_JOBSTREAM	CDW	1		Build	SCHED_ENGINE – ProdMaster	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000	Job Stream
15	SCHED_JOB	CDW	1		MakeAll	SCHED_ENGINE – ProdMaster ,SCHED_WORKSTATION – Build_dev , SCHED_JOBSTREAM - Build	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000	Job

6.1.3 Component relationship type (table RelnTyp)

RelnTyp_Cd CHAR(6)	**RelnTyp_Nm * VARCHAR(120)**
PCHILD	Parent Child Relation
LCONT	Logical Containment Relation
RUNSON	Runs on Relation
* This column is translated.	

6.1.4 Component relationship rule (table RelnRul)

CompTyp_Source_Cd CHAR(17)	**CompTyp_Target_Cd CHAR(17)**	**RelnTyp_Cd CHAR(6)**	**RelnRul_Strt_DtTm TIMESTAMP**	**RelnRul_End_DtTm TIMESTAMP**
SCHED_ENGINE	SCHED_WORKSTATION	PCHILD	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000
SCHED_ENGINE	SCHED_JOBSTREAM	PCHILD	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000
SCHED_ENGINE	SCHED_JOB	PCHILD	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000
SCHED_JOB	SCHED_JOBSTREAM	LCONT	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000
SCHED_JOB	SCHED_WORKSTATION	RUNSON	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000
IP_HOST	SCHED_WORKSTATION	RUNSON	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000

6.1.5 Component relationship (table CompReln)

CompReln_ID INTEGER	**Comp_Source_ID INTEGER**	**Comp_Target_ID INTEGER**	**RelnTyp_Cd CHAR(6)**	**CompReln_Strt_DtTm TIMESTAMP**	**CompReln_End_DtTm TIMESTAMP**
1	1	3	PCHILD	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000
2	2	3	RUNSON	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000
3	1	5	PCHILD	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000

CompReln_ID INTEGER	Comp_Source_ID INTEGER	Comp_Target_ID INTEGER	RelnTyp_Cd CHAR(6)	CompReln_Strt_DtTm TIMESTAMP	CompReln_End_DtTm TIMESTAMP
4	4	5	RUNSON	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000
5	1	7	PCHILD	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000
6	6	7	RUNSON	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000
7	1	9	PCHILD	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000
8	8	9	RUNSON	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000
9	1	10	PCHILD	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000
10	8	10	RUNSON	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000
11	1	11	PCHILD	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000
12	1	12	PCHILD	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000
13	12	11	LCONT	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000
14	12	5	RUNSON	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000
15	1	13	PCHILD	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000
16	13	11	LCONT	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000
17	13	7	RUNSON	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000
18	1	14	PCHILD	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000
19	12	14	LCONT	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000
20	12	10	RUNSON	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000
21	1	15	PCHILD	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000
22	15	14	LCONT	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000
23	15	10	RUNSON	2002-01-01-00.00.00.000000	9999-01-01-0.00.00.000000

6.1.6 Attribute type (table AttrTyp)

AttrTyp_Cd CHAR(17)	AttrTyp_Nm * VARCHAR(120)
LAST_IP_ADDRESS	Last IP Address
SCHED_WRKST_TYPE	Workload Scheduler Workstation Type
OS_NAME	Operating System Name
SCHED_DOMAIN	Workload Scheduler Domain
SCHED_PRIORITY	Workload Scheduler Priority
SCHED_LOGIN	Workload Scheduler Login User Name
SCHED_COMMAND	Workload Scheduler Command String
* This column is translated.	

6.1.7 Attribute rule (table AttrRul)

CompTyp_Cd CHAR(17)	AttrTyp_Cd CHAR(17)	AttrRul_Strt_DtTm TIMESTAMP	AttrRul_End_DtTm TIMESTAMP	AttrRul- _Dom_Ind CHAR
IP_HOST	LAST_IP_ADDRESS	2002-01-01- 00.00.00.000000	9999-01-01- 00.00.00.000000	N
SCHED_WORKSTATION	SCHED_WRKST_TYPE	2002-01-01- 00.00.00.000000	9999-01-01- 00.00.00.000000	N
SCHED_WORKSTATION	OS_NAME	2002-01-01- 00.00.00.000000	9999-01-01- 00.00.00.000000	N
SCHED_WORKSTATION	SCHED_DOMAIN	2002-01-01- 00.00.00.000000	9999-01-01- 00.00.00.000000	
SCHED_JOBSTREAM	SCHED_PRIORITY	2002-01-01- 00.00.00.000000	9999-01-01- 00.00.00.000000	N
SCHED_JOB	SCHED_PRIORITY	2002-01-01- 00.00.00.000000	9999-01-01- 00.00.00.000000	N
SCHED_JOB	SCHED_LOGIN	2002-01-01- 00.00.00.000000	9999-01-01- 00.00.00.000000	N
SCHED_JOB	SCHED_COMMAND	2002-01-01- 00.00.00.000000	9999-01-01- 00.00.00.000000	N

6.1.8 Attribute domain (table AttrDom)

AttrDom_ID INTEGER	CompTyp_Cd CHAR(17)	AttrTyp_Cd CHAR(17)	AttrDom_Strt_DtTm TIMESTAMP	AttrDom_End_DtTm TIMESTAMP	AttrDom_Val VARCHAR(254)	AttrDom_Ds VARCHAR(254)

6.1.9 Component attribute (table CompAttr)

CompAttr_ID INTEGER	Comp_ID INTEGER	AttrTyp_Cd CHAR(17)	CompAttr_Strt_DtTm TIMESTAMP	CompAttr_End_DtTm TIMESTAMP	CompAttr_Val VARCHAR(254)
1	3	SCHED_WRKST_TYPE	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000	Fault Tolerant Agent
2	3	OS_NAME	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000	Windows
3	3	SCHED_DOMAIN	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000	Account
4	5	SCHED_WRKST_TYPE	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000	SAP-X-Agent
5	5	OS_NAME	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000	Unix
6	5	SCHED_DOMAIN	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000	Account
7	7	SCHED_WRKST_TYPE	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000	Standard Agent
8	7	OS_NAME	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000	Unix
9	7	SCHED_DOMAIN	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000	Finance
10	9	SCHED_WRKST_TYPE	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000	Fault Tolerant Agent
11	9	OS_NAME	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000	Windows
12	9	SCHED_DOMAIN	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000	Dev
13	10	SCHED_WRKST_TYPE	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000	Fault Tolerant Agent
14	10	OS_NAME	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000	Windows
15	10	SCHED_DOMAIN	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000	Dev
16	11	SCHED_PRIORITY	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000	10
17	12	SCHED_PRIORITY	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000	1
18	12	SCHED_LOGIN	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000	Twsusr22
19	12	SCHED_COMMAND	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000	Extract –db db2 –h ppp –v 23

CompAttr_ID INTEGER	Comp_ID INTEGER	AttrTyp_Cd CHAR(17)	CompAttr_Strt_DtTm TIMESTAMP	CompAttr_End_DtTm TIMESTAMP	CompAttr_Val VARCHAR(254)
20	13	SCHED_PRIORITY	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000	20
21	13	SCHED_LOGIN	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000	Twsusr33
22	13	SCHED_COMMAND	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000	Calcpay –in extr.txt –out pay.form
23	14	SCHED_PRIORITY	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000	50
24	15	SCHED_PRIORITY	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000	10
25	15	SCHED_LOGIN	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000	Twsbuild1
26	15	SCHED_COMMAND	2002-01-01-00.00.00.000000	9999-01-01-00.00.00.000000	Make 2>&1 | tee build.log

6.2 Component measurement

The following sections describe the component measurement.

6.2.1 Measurement group type (table MGrpTyp)

MGrpTyp_Cd CHAR(6)	MGrpTyp_Nm * VARCHAR(120)
CATEG	Category
GROUP	Aggregate Types or Group Functions
STATE	State
TRANS	State Transition Groups
* This column is translated.	

6.2.2 Measurement group (table MGrp)

MGrp_Cd CHAR(6)	MGrpTyp_Cd CHAR(6)	MGrp_Parent_Cd CHAR(6)	MGrp_Nm * VARCHAR(120)
AVG_E	GROUP		Average Value Exists
MIN_E	GROUP		Minimum Value Exists
MAX_E	GROUP		Maximum Value Exists
TOT_E	GROUP		Total value Exists
AVL	CATEG		Availability
PERF	CATEG		Performance
STATE	CATEG		Percentage State Measurements
STORAG	CATEG		Storage
UTIL	CATEG		Utilization

MGrp_Cd CHAR(6)	MGrpTyp_Cd CHAR(6)	MGrp_Parent_Cd CHAR(6)	MGrp_Nm * VARCHAR(120)
* This column is translated.			

6.2.3 Measurement group member (table MGrpMbr)

MGrp_Cd CHAR(6)	MGrpTyp_Cd CHAR(6)	MsmtTyp_ID INTEGER
TOT_E	GROUP	41
TOT_E	GROUP	15
TOT_E	GROUP	42
TOT_E	GROUP	43
TOT_E	GROUP	14
MIN_E	GROUP	16
MAX_E	GROUP	16
AVG_E	GROUP	16
TOT_E	GROUP	16
MIN_E	GROUP	44
MAX_E	GROUP	44
AVG_E	GROUP	44
TOT_E	GROUP	44
MIN_E	GROUP	45
MAX_E	GROUP	45
AVG_E	GROUP	45
TOT_E	GROUP8	45

6.2.4 Measurement unit category (table MUnitCat)

MunitCat_Cd CHAR(6)	MunitCat_Nm * VARCHAR(120)
TM	Time Duration
QTY	Quantity
PRC	Percentage
* This column is translated.	

6.2.5 Measurement unit (table MUnit)

MUnit_Cd CHAR(6)	MUnitCat_Cd CHAR(6)	Munit_Nm * VARCHAR(120)
PRC	PRC	Percentage
QTY	QTY	Quantity
Sec	TM	Seconds
…	…	…
* This column is translated.		

6.2.6 Time summary (table TmSum)

The period over which a measurement may be summarized.

TmSum_Cd CHAR	TmSum_Nm * VARCHAR(120)
D	Daily
W	Weekly
M	Monthly
Q	Quarterly
Y	Yearly
H	Hourly
P	Point
* This column is translated.	

6.2.7 Measurement source (table MSrc)

MSrc_Cd CHAR(6)	MSrc_Parent_Cd CHAR(6)	MSrc_Nm * VARCHAR(120)
Tivoli		Tivoli Application
AWS	Tivoli	Tivoli Workload Scheduler
* This column is translated.		

6.2.8 Measurement type (table MsmtTyp)

MsmtTyp_ID INTEGER	MUnit_Cd CHAR(6)	MSrc_Cd CHAR (6)	MsmtTyp_Nm * VARCHAR(120)	MsmtTyp_Ds * VARCHAR(254)
41	QTY	AWS	Number of Successful Runs	Number of runs that were successful for a job or a job stream
15	QTY	MODEL1	Number of Unsuccessful Runs	Number of failed runs
42	QTY	AWS	Number of Canceled Runs	Number of runs that were canceled for a job or a job stream
43	QTY	AWS	Number of Unknown Runs	Number of unknown status runs for a job or job stream
14	QTY	MODEL1	Number or Runs	The number of runs
16	Sec	MODEL1	Duration	The amount of time needed to complete a process
44	Sec	AWS	Start Delay	Start delay, that is the time elapsed between the planned start time and the effective start time of a job or job stream
45	Sec	AWS	CPU Time	CPU utilization time
* This column is translated.				

6.2.9 Component measurement rule (table MsmtRul)

CompTyp_Cd CHAR(17)	MsmtTyp_ID INTEGER
SCHED_JOB	41
SCHED_JOB	15

SCHED_JOB	42
SCHED_JOB	43
SCHED_JOB	14
SCHED_JOB	16
SCHED_JOB	44
SCHED_JOB	45
SCHED_JOBSTREAM	41
SCHED_JOBSTREAM	15
SCHED_JOBSTREAM	42
SCHED_JOBSTREAM	43
SCHED_JOBSTREAM	14
SCHED_JOBSTREAM	16
SCHED_JOBSTREAM	44

6.2.10 Measurement (table Msmt)

Msmt_ID BIGINT	**Comp_ID INTEGER**	**MsmtTyp_ID INTEGER**	**TmSum_Cd CHAR**	**Msmt_Strt_Dt DATE**	**Msmt_Strt_Tm TIME**	**Msmt_Min_Val FLOAT**	**Msmt_Max_Val FLOAT**	**Msmt_Avg_Val FLOAT**	**Msmt_Tot_Val FLOAT**	**Msmt_Smpl_Cnt INTEGER**	**Msmt_Err_Cnt INTEGER**
1	11	15 Unsuccessful. runs	D	2002-09-14	06.00.00.000000				1		
2	11	14 Number of runs	D	2002-09-14	06.00.00.000000				1		
3	11	44 Start Delay	D	2002-09-14	06.00.00.000000	5 (00.00.05)	5 (00.00.05)	5 (00.00.05)			
4	11	44 Start Delay	D	2002-09-14	06.00.00.000000				5 (00.00.05)		
5	12	41 Unsuccessful. Runs	D	2002-09-14	06.00.00.000000				1		
6	12	15 successful. Runs	D	2002-09-14	06.00.00.000000				1		
7	12	14 Number of runs	D	2002-09-14	06.00.00.000000				2		

8	12	16 Duration	D	2002-09-14	06.00.00.000000	1501 (00.25.01)	5460 (01.31.00)	58 (00.00.58)			
9	12	16 Duration	D	2002-09-14	06.00.00.000000				3960 (01.06.00)		
10	12	44 Start Delay	D	2002-09-14	06.00.00.000000	721 (00.12.01)	910 (00.15.10)	784 (00.13.04)			
11	12	44 Start Delay	D	2002-09-14	06.00.00.000000				1621 (00.27.11)		
12	12	45 CPU Time	D	2002-09-14	06.00.00.000000	660 (00.11.00)	1320 (00.22.00)	990 (00:16:30)			
13	12	45 CPU Time	D	2002-09-14	06.00.00.000000				1980 (00.33.00)		
14	13	41 Unsuccessful. Runs	D	2002-09-14	06.00.00.000000				1		
15	13	14 Number of runs	D	2002-09-14	06.00.00.000000				1		
16	13	16 Duration	D	2002-09-14	06.00.00.000000	23 (00.00.23)	23 (00.00.23)	23 (00.00.23)			
17	13	16 Duration	D	2002-09-14	06.00.00.000000				23 (00.00.23)		
18	13	44 Start Delay	D	2002-09-14	06.00.00.000000	30 (00.00.30)	30 (00.00.30)	30 (00.00.30)			
19	13	44 Start Delay	D	2002-09-14	06.00.00.000000				30 (00.00.30)		
20	13	45 CPU Time	D	2002-09-14	06.00.00.000000	7 (00.00.07)	7 (00.00.07)	7 (00.00.07)			
21	13	45 CPU Time	D	2002-09-14	06.00.00.000000				7 (00.00.07)		
22	14	41 Unsuccessful. Runs	D	2002-09-14	06.00.00.000000				1		
23	14	14 Number of runs	D	2002-09-14	06.00.00.000000				1		
24	14	16 Duration	D	2002-09-14	06.00.00.000000	502 (00.08.22)	502 (00.08.22)	502 (00.08.22)			
25	14	16 Duration	D	2002-09-14	06.00.00.000000				502 (00.08.22)		

26	15	41 Unsuccessful. Runs	D	2002-09-14	06.00.00.000000				1		
27	15	14 Number of runs	D	2002-09-14	06.00.00.000000				1		
28	15	16 Duration	D	2002-09-14	06.00.00.000000	7201 (02.00.01)	7201 (02.00.01)	7201 (02.00.01)			
29	15	16 Duration	D	2002-09-14	06.00.00.000000				7201 (02.00.01)		
30	15	44 Start Delay	D	2002-09-14	06.00.00.000000	119 (00.01.59)	119 (00.01.59)	119 (00.01.59)			
31	15	44 Start Delay	D	2002-09-14	06.00.00.000000				119 (00.01.59)		
32	15	45 CPU Time	D	2002-09-14	06.00.00.000000	3300 (00.55.00)	3300 (00.55.00)	3300 (00.55.00)			
33	15	45 CPU Time	D	2002-09-14	06.00.00.000000				3300 (00.55.00)		

7 Data mart schema information

The following sections contain the definition of star schemas, metric dimension tables, data marts, and reports provided with the Tivoli Workload Scheduler warehouse pack.

Shaded columns in the following tables are translated. *Installing and Configuring Tivoli Data Warehouse* contains instructions for installing support for additional languages.

7.1 Star schemas

Before using this section, read about the star schemas in *Enabling an Application for Tivoli Data Warehouse*. That document defines the content of each table and explains the relationships between the tables in this document.

This warehouse pack provides the following star schemas.

7.1.1 TWS Daily Run States star schema

The following table defines the star schema. The description of the star schema can be translated.

Description of star schema (in IWH_STARSCHEMA)	This star schema groups daily information related to the job run states.
Name of fact table	**AWS.F_RunStates_DAY**
Name of metric dimension table	**AWS.D_RunStates_METRIC**
Names of other dimension tables	**AWS.D_JOB**
	AWS.D_WORKSTATION

7.1.1.1 Fact table AWS.F_RunStates_DAY

Metric_ID INTEGER	**Job_ID INTEGER**	**Workstation_ID INTEGER**	**Meas_date TIMESTAMP**	**Min_value DOUBLE**	**Max_value DOUBLE**	**Avg_value DOUBLE**	**Total_value DOUBLE**	**Sample_count DOUBLE**
1	1	1	2002-07-15				23	
2	1	1	2002-07-15				4	
5	1	1	2002-07-15				31	
……	……	……	……	……	……	……	……	……

7.1.2 TWS Daily Run Times star schema

The following table defines the star schema. The description of the star schema can be translated.

Description of star schema (in IWH_STARSCHEMA)	This star schema groups information related to the daily average and total processing times for a particular job.
Name of fact table	**AWS.F_RunTimes_DAY**
Name of metric dimension table	**AWS.D_RunTimes_METRIC**
Names of other dimension tables	**AWS.D_JOB**
	AWS.D_WORKSTATION

7.1.2.1 Fact table AWS.F_RunTimes_DAY

Metric_ID INTEGER	Job_ID INTEGER	Workstation_ID INTEGER	Meas_date TIMESTAMP	Min_value DOUBLE	Max_value DOUBLE	Avg_value DOUBLE	Total_value DOUBLE	Sample_count DOUBLE
1	1	1	2002-07-15-00.00.00.0000	13	177	60	445	
2	1	1	2002-07-15-00.00.00.0000	2	12	4	34	
3	1	1	2002-07-15-00.00.00.0000	0	300	100		
……	……	……	……	……	……	……	……	……

7.2 Metric dimension tables

This section describes the metric dimension tables used by the star schemas in this warehouse pack. Shaded columns indicate text that is translated. These column headings are also marked with an asterisk (*).

7.2.1 AWS.D_RunStates_METRIC

metric_ID INTEGER	met_category * VARCHAR(254)	met_desc * VARCHAR(254)	met_name * VARCHAR(254)	met_units* VARCHAR(254)	min_exists CHAR(1)	max_exists CHAR(1)	ave_existsir CHAR(1)	total exists CHAR(1)	msrc_nm VARCHAR(254)
*	Not used	Number of successful runs for a job or a job stream	Number of Successful Runs	Number of job runs	N	N	Y	N	AWS
2	Not used	Number of abended and failed runs for a job or a job stream	Number of Unsuccessful Runs	Number of job runs	N	N	Y	N	AWS
3	Not used	Number of canceled runs for a job or a job stream	Number of Canceled Runs	Number of job runs	N	N	Y	N	AWS
4	Not used	Number of unknown status runs for a job or a job stream	Number of Unknown Runs	Number of job runs	N	N	Y	N	AWS
5	Not used	Total number of runs for a job or a job stream	Number of Runs	Number of job runs	N	N	N	Y	AWS
* This column is translated.									

7.2.2 AWS.D_RunTimes_METRIC

metric_ID INTEGER	met_category * VARCHAR(254)	met_desc * VARCHAR(254)	met_name * VARCHAR(254)	met_units * VARCHAR(254)	min_exists CHAR(1)	max_exists CHAR(1)	ave_existsir CHAR(1)	total exists CHAR(1)	msrc_nm * VARCHAR(254)
1	Not used	Run duration time of a job	Duration	Seconds	Y	Y	Y	Y	AWS
2	Not used	CPU utilization time of a job	CPU Time	Seconds	Y	Y	Y	Y	AWS
3	Not used	Start delay, that is the time elapsed between the planned start time and the effective start time of a job	Start Delay	Seconds	Y	Y	Y	N	AWS
* This column is translated.									

7.3 Dimension tables

The following sections describe the dimension tables (other than metric dimension tables) used by the star schemas in this warehouse pack.

7.3.1 Dimension table AWS.D_JOB

The following columns are used in this dimension table.

Job_ID INTEGER	Job_Name CHAR(40)	Job_Stream_Name CHAR(16)	Priority INTEGER	OWNER CHAR(64)	Login CHAR(32)
1	Extract	Payroll	1	Orlando Salazar	Twsusr22
2	Extract	Build	1	Jack Straw	Twsusr22
3	CalcPayroll	Payroll	20	Richard Smith	Twsusr33
4	Makeall	Build	10	GBE tools support department	Twsbuild1

7.3.2 Dimension table AWS.D_WORKSTATION

The following columns are used in this dimension table.

Workstation_ID INTEGER	Name CHAR(16)	Operating_System CHAR(32)	IP_Host CHAR(254)	Network CHAR(16)
1	DM1	Windows	Abc.abc.com	ProdMaster
2	SAPXA1	UNIX	Db2.abc.com	ProdMaster

Workstation_ID INTEGER	Name CHAR(16)	Operating_System CHAR(32)	IP_Host CHAR(254)	Network CHAR(16)
3	FTA1	UNIX	Pay.abc.com	ProdMaster
4	Cmvc_dev	Windows	Aix1.abc.com	ProdMaster
5	Buid-dev	Windows	Aix1.abc.com	ProdMaster

7.4 Data marts and reports

This warehouse pack provides the following data marts.

7.4.1 Data mart AWS Daily RunStates data mart

This data mart uses the following star schema:

- TWS daily Run States star schema

7.4.2 Reports

7.4.2.1 Jobs with highest number of unsuccessful runs

This extreme case report shows jobs with the highest number of days when it have at least one or more unsuccessful runs (that is the job is abended or failed) during the time range selected.

7.4.2.2 Workstations with highest number of unsuccessful runs

This extreme case report shows workstations with the highest number of unsuccessful jobs run on those workstations during the time range selected. This number is the sum of the days when the job is abended or failed.

7.4.2.3 Run states statistics for all jobs

This health check report compares the number of successful, unsuccessful, canceled and unknown runs for all the jobs that were run during the day.

7.4.3 Data mart AWS Daily RunTimes data mart

This data mart uses the following star schema:

- TWS daily Run Times star schema

7.4.4 Reports

This data mart provides the following prepackaged reports.

7.4.4.1 Jobs with the highest average duration time

This extreme case report shows jobs with the highest average run duration time.

7.4.4.2 Workstations with highest CPU utilization

This extreme case report shows workstations with the highest average CPU utilization.

7.4.4.3 Run times statistics for all jobs

This health check report compares the average run duration and average CPU utilization time measurements for all the jobs that run during the day.

7.4.4.4 Start Delay statistics for all jobs

This health check report shows the average delay time measurement for all the jobs that run during the day.

8 Troubleshooting

This section provides information on how to recover from errors that you might encounter when using Tivoli Workload Scheduler Warehouse Pack.

8.1 Error while loading a report

While loading a report, if the following error message is displayed:

"Maximum processing time or Maximum records limit reached"

On the workstation where you installed the Crystal Server, perform the following steps:

1. Open the Windows registry using the regedit command.
2. Modify the "ImagePath" key located in the HKEY_LOCAL_MACHINE->SYSTEM->ControlSet001->Services->pageserver registry path, by adding the " –maxDBResultRecords 0" string at the end of its current value.
3. Restart the Crystal Page Server service.

Notices

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.
Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee. The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental. This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

If you are viewing this information in softcopy form, the photographs and color illustrations might not appear.

Trademarks

IBM, the IBM logo, AIX, DB2, IBMLink, Informix, OS/2, OS/400, Tivoli, the Tivoli logo, Tivoli Enterprise Console, and TME are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.