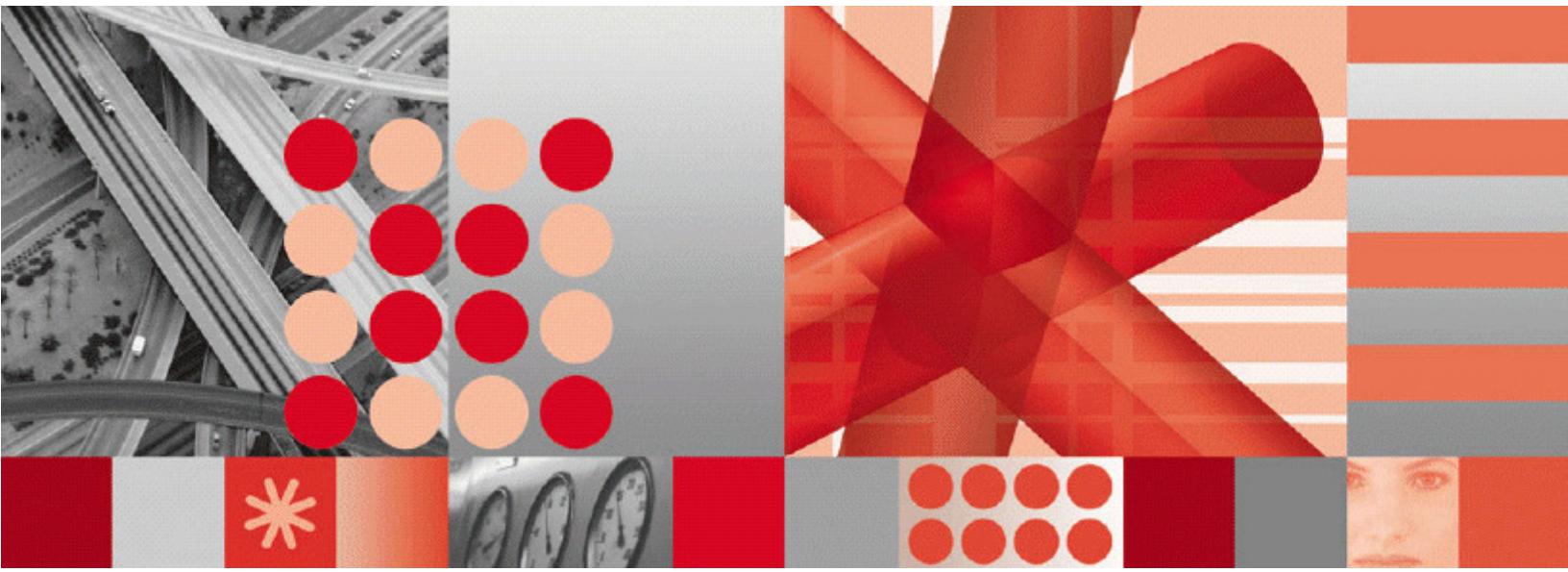
Tivoli Netcool® Performance Manager for Wireless

Version 3.5

IBM



JDBC SQL Gateways User Guide

Note: Before using this information and the product it supports, read the information in Notices on page 18.

This edition applies to Version 4.1 of IBM® Tivoli® Netcool® Performance Manager for Wireless and to all subsequent releases and modifications until otherwise indicated in new editions.

Table of Contents

1 About this Documentation 6
1.1 Audience 6
1.2 Required Skills and Knowledge 6
2 Introduction 7
3 Overview 8
3.1 The Gateway Framework 8
3.2 SQL Extractor Overview 9
3.2.1 Process Flow 9
3.2.2 Data Flow 9
4 Engine Rules and Configuration 10
4.1 RULE_TYPE 10
4.1.1 Mandatory Rule Configuration 10
4.1.2 Optional Rule Configuration 11
4.1.3 SQL Rule Configuration 12
5 Post Parser Rules and Configuration 15
6 Sample Configuration 16
Appendix A Notices and Trademarks 18

1 About this Documentation

1.1 Audience

The target audience of this document is IBM Performance Manager for Wireless customers. They should be familiar with telecommunication and IT principles and should also have a good understanding of Solaris.

IMPORTANT: Before attempting an installation of Performance Manager for Wireless you are strongly advised to read the release notes and any readme files distributed with your Performance Manager for Wireless software. Readme files and release notes may contain information specific to your installation not contained in this guide. Failure to consult readme files and release notes may result in a corrupt, incomplete or failed installation.

Note: Performance Manager for Wireless Administrators should not, without prior consultation and agreement from IBM, make any changes to the Index Organized tables or database schema. Changes to the Index Organized tables or database schema may result in corruption of data and failure of the Performance Manager for Wireless System. This applies to all releases of Performance Manager for Wireless using all versions of interfaces.

1.2 Required Skills and Knowledge

This guide assumes you are familiar with the following:

- General IT Principles
- Sun Solaris Operating System
- Oracle Database
- Windows operating systems
- Graphical User Interfaces
- Network Operator's OSS and BSS systems architecture

This guide also assumes that you are familiar with your company’s network and with procedures for configuring, monitoring, and solving problems on your network.

2 Introduction

This document describes the steps required to install and run a Gateway. The steps described here are generic to all Productised Gateways from release 3.4 and above.

The layout of the Gateways installation was altered at release 3.4, and this document only applies to releases from this point.

As well as this document, readers should refer to the following documents before proceeding to install the Gateway:

- the Gateway Configuration Distribution Note
- the appropriate Vendor Gateway Distribution Note
- the Gateway Framework Distribution Note

3 Overview

3.1 The Gateway Framework

The JDBC SQL uses the Gateway Framework as a container for the execution of its engine and post parser stages. The Gateway Framework and JDBC SQL are de-coupled into two separate installations. The Gateway Framework consists of a library of Perl modules that provide functionality such as:

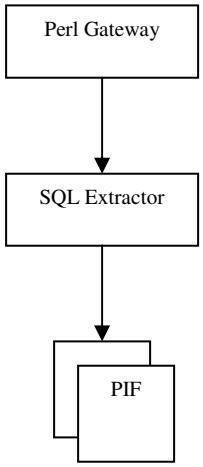
- a container for the execution of the SQL Extractor and Post Parser rules for of data transformation
- Intermediate (PIF) and output data (LIF,CSV,XML) storage and management
- logging utilities
- cleanup and crash recovery
- statistics gathering

The JDBC SQL simply plugs into the Gateway Framework and extends this base functionality to provide the final Gateway that parses a specific vendor's ASCII data.
More information on the standard Gateway configuration is contained in the Gateway Framework User Guide.
Only JDBC SQL configuration details will be described in this document.

3.2 SQL Extractor Overview

3.2.1 Process Flow

The Perl Gateway start as usual by instantiating and executing a Vendor Gateway based on the rule in EngineConfig.pm. If the rule specifies the JDBC SQL, the Vendor Gateway for JDBC SQL will be instantiated.



SQL Extractor will be connected to database as specified in the rules. If the connection fails to establish, it will try to reconnect for given numbers of retry within the interval configured. If connection still fails, SQL Extractor will exit and return control to Perl Gateway.

3.2.2 Data Flow

All configuration entries are done inside EngineConfig.pm. When a database connection successfully established, all queries specified inside the rule will be executed using this connection. The connection will be closed only after all queries are successfully executed.

For each of the result retrieved from the database, they will be process according to the rule configured by the user. Those configurations will initially be processed by Perl gateway and then converted to XML files to be loaded by SQL Extractor.

There are certain rules that having the same functionality with the sub-entries of SQL. If both of them are configured, the sub-entries rules within SQL will have higher priority in processing the data. Once they are done, then it will hand it over for later processing.

4 Engine Rules and Configuration

SQL Extractor contains its own set of rule entries. Some of the entries are mandatory while others will be optional depending on the configuration required.

4.1 RULE_TYPE

Defines which vendor gateway will be used and the access to the database for data extraction. The rule entry 'SQL' possesses its own set of sub-entries where they are required to be configured further for specific configuration of each query. All extracted data will be outputted as Parser Intermediate Format (PIF).

4.1.1 Mandatory Rule Configuration

The following are the details and example of each mandatory rule entries. All entries are scalar value except SQL as hash value.

- RULE_TYPE: To select the parser.

```
RULE_TYPE    =>    'SQL_EXTRACT',
```

- DB_USERNAME: Username for database connection.

```
DB_USERNAME =>    'username',
```

- DB_PASSWORD: Password for database connection.

```
DB_PASSWORD =>    'password',
```

- DB_DATABASE: Name of service/database for the data extraction.

```
DB_DATABASE =>    'database',
```

- DB_HOST: Host address of the database.

```
DB_HOST     =>    'hostname',
```

- DB_PORT: Port number of the database connection.

```
DB_PORT     =>    '6565',
```

- SQL: This is a mandatory hash entry where it has its own sub entries which will be describe in section 2.1.3.

4.1.2 Optional Rule Configuration

The following are the details and example of each optional rule entries. All entries except COLUMN_RENAME are scalar value. COLUMN_RENAME is hash value.

- RULE_DESC: Description of current parser configuration.

```
RULE_DESCRIPTION  =>    'Parse data from database',
```

- TIMESTAMP_FILENAME: Scalar value that store the name for timestamp file. Each rule may have different or same file name.

```
TIMESTAMP_FILENAME      => 'timestamp',
```

- MAX_RETRIES: Maximum number of retries if connection fails to establish.

```
MAX_RETRIES =>    '5',
```

- RETRY_DELAY: The interval between each reconnection to the database. The scalar value will be taken in millisecond. Take the example below, it will represent the parser will try to connection to database every 5 seconds if previous connection failed.

```
RETRY_DELAY       =>    '5000',
```

- COLUMN_RENAME: A hash that rename extracted column name to new counter name. The renaming process will be performed from inner level to outer level. Therefore, the renaming for this rule is done after the SQL sub-entry RENAME_COUNTER renaming process.

```
COLUMN_RENAME     =>    {
      'start_time'      =>    'STARTTIME',
      'start_date'      =>    'DATE',
},
```

- DEFAULT_BLOCKNAME: Default blockname for the PIF. The PIF blockname is named from inner level to outer level where the BLOCKNAME entry under SQL will have higher priority than this rule entry. If both entries are not configured, the output blockname will be 'BLOCKNAME'. Blockname will be output as part of the filename and timestamp record, therefore it is recommended to configure this enrty.

```
DEFAULT_BLOCKNAME =>    'BLOCKNAME',
```

- DEFAULT_NULL_VALUE: Value for replacing the null values returned from the query result. By default, null value will be replaced with empty string.

```
DEFAULT_NULL_VALUE      =>    'null',
```

4.1.3 SQL Rule Configuration

Within the SQL hash, the configuration needs to be done in the structure as shown in following hierarchy diagram.

```
SQL
|-- <QUERY_SECTION_1>
|         |-- QUERY
|         |-- DATE_TIME_NAME
|         |-- DATE_TIME_START_VALUE
|         |-- INTERVAL
|         |-- <PARAMETER_SECTION_1>
|         |           |-- PARAMETER
|         |           |-- BLOCKNAME
|         |           |-- HEADER_FIELDS
|         |           |-- DATA_FIELDS
|         |           |-- INVALID_VALUE_MATCH
|         |-- <PARAMETER_SECTION_2>
|         |           |-- PARAMETER
|         |           |-- HEADER_FIELDS
|         |           |-- DATA_FIELDS
|         |...
|-- <QUERY_SECTION_2>
|         |...
|...
```

From the diagram above, the entries within <> can be any name provided by the user. However, the entries for QUERY_SECTION need to be different from each other as they will be used in identifying the timestamp for each query. Those entries are recommended to be named properly and without empty space. The entries within <> can be more than one as required.

The following contain details of mandatory, optional and time management rule within the SQL hash.

4.1.3.1 Mandatory Rule Configuration

- QUERY: Scalar value of query for data extraction. It can be compiled or precompiled SQL statement. If precompiled SQL is used, PARAMETER must be set with the correct number of parameters.

```
QUERY          =>   'SELECT start_time, start_date, colA,
colB, colC FROM table1 WHERE colA = ? AND colB = ?',
```

- HEADER_FIELDS: This array will extract the counters (columns) from the result of the query and will be outputted in the header section of the PIF.

```
HEADER_FIELDS  =>   ['start_time', 'start_date'],
```

4.1.3.2 Optional Rule Configuration

- PARAMETER: Array of parameters for the precompiled SQL statement. This is optional entry as it depends on the query statement configured for the data extraction. For example, if the query is "SELECT * FROM ?, ?" The values in the PARAMETER entry will replace the '?' accordingly. Therefore the number of '?' sign in the query must same with the number of values provided inside the PARAMETER array.

```
PARAMETER =>    ['1130', '15MAY2008'],
```

- DATA_FIELDS: This array containing counters (columns) where they will be outputted in the data fields section. It will output all the columns retrieved from the query if it is not configured.

```
DATA_FIELDS          =>    ['colA', 'colB', 'colC'],
```

- INVALID_VALUE_MATCH: A hash which replace invalid characters in the results of the query with valid characters.

```
INVALID_VALUE_MATCH =>    {
           '@' => '_',
             '^' => '_',
},
```

- FIELDS_TO_KEY_PIF_FILENAME: An array containing field values to be used as part of the filename. The default is to take the header fields as filename if this entry is not configured.

```
FIELDS_TO_KEY_PIF_FILENAME    =>    ['start_time',
                            'start_date', 'colA'],
```

- BLOCKNAME: Scalar value for the PIF blockname specific to the output of that query. It has higher priority than DEFAULT_BLOCKNAME if both are configured.

```
BLOCKNAME =>    'M118723142',
```

- RENAME_COUNTER: A hash which rename the extracted counter name to new counter name. It has higher priority to COLUMN_RENAME if both are configured.

```
RENAME_COUNTER =>    {
           'colA'     =     'BTS_ID',
           'colB'     =     'CELL_ID',
},
```

4.1.3.3 Interval Management Configuration

For interval management, all the entries are mandatory with the condition that the user needs to select which group of time column name and start value to specify the time. For example, if DATE_TIME_NAME and DATE_TIME_START_VALUE entries group is configured, the other group that contains TIME_NAME, TIME_START_VALUE, DATE_NAME and DATE_START_VALUE will no be necessary and vice versa.

- INTERVAL: Scalar value that represent the interval in retrieving the data.

```
INTERVAL  =>   30,
```

- DATE_TIME_NAME: Scalar value for the date and time column in the database.

```
DATE_TIME_NAME =>   'START_TIME',
```

- DATE_TIME_START_VALUE: Scalar value for the date and time value in the database. The date for this entry need to be formatted as 'DD-MM-YYYY HH24:MI:SS'.

```
DATE_TIME_START_VALUE    =>   '20-08-2008 10:00:00',
```

- DATE_NAME: Scalar value that represents the date column in the database for retrieving certain amount of files based on the interval.

```
DATE_NAME =>   'START_DATE',
```

- DATE_START_VALUE: Scalar value that stores the start date of the interval in the format of 'DD-MM-YYYY'.

```
DATE_START_VALUE    =>   '03-07-2008',
```

- TIME_NAME: Scalar value that represents the time column in the database for retrieving certain amount of files based on the interval.

```
TIME_NAME =>   'START_TIME',
```

- TIME_START_VALUE: Scalar value that stores the start time of the interval in the format of 'HH24:MI:SS'.

```
TIME_START_VALUE    =>   '13:00:00',
```

The timestamp file will be generated if it is specified in the configuration. Filename is specified manually and it allows different file names for each SQL_EXTRACT rule. The timestamps recorded inside include the local gateway last runtime and the timestamp for the previous query. Both of them are stored as Perl hash reference format where the value will be converted into seconds. For example:

```
$time_hash = {
        'BLOCKNAME' => {
                'QUERY_4' => {
                        'lastruntime' => 1217207716,
                        'timestamp' => '1218909660'
                        },
                'QUERY_5' => {
                        'lastruntime' => 1217207716,
                        'timestamp' => '1218909720'
                        },
                'QUERY_6' => {
                        'lastruntime' => 1217207716,
                        'timestamp' => '1218373260'
                        }
                }
        };
```

From the example above, the hash key within the $time_hash will be taken from the DEFAULT_BLOCKNAME entry while its values will comprise of another set of hashes where their keys are taken from the QUERY_SECTION specified based on hierarchy diagram above.

5 Post Parser Rules and Configuration

Only standard Post Parser Rules Are Supplied. There are no vendor specific rules as the parser is not vendor specific.

6 Sample Configuration

```
{
        RULE_TYPE      => "SQL_EXTRACT",
        RULE_DESC      => "FIRST_RULE",
        DB_USERNAME    => "SYSTEM",
        DB_PASSWORD    => "password",
        DB_HOST        => "127.0.0.1",
        DB_PORT        => "1521",
        DB_DATABASE    => "database",
        DEFAULT_NULL_VALUE     => "NULL",
        DEFAULT_BLOCKNAME      => "BLOCKNAME",
        RETRY_DELAY            => "5000",
        MAX_RETRIES            => 5,
        COLUMN_RENAME          => {
            COLUMN1 => 'BTS_ID',
            COLUMN2  => 'CELL_ID',
            COLUMN3   => 'CARRIER_ID',
        },
        TIMESTAMP_FILENAME     => 'timestamp1',
        SQL => {
            QUERY_1 => {
                QUERY   => "SELECT to_char(START_TIME, 'DD-MM-YYYY') AS START_DATE, to_char(START_TIME, 'HH24MI') AS
START_TIME, STRING1, STRING2, NUMBER1, NUMBER2 FROM TABLE1 WHERE STRING1 = ? AND NUMBER1 = ?",
                DATE_TIME_NAME => 'START_TIME',
                DATE_TIME_START_VALUE => "16-08-2008 17:50:00",
                INTERVAL => 1,
                PARAM_1 => {
                    PARAMETER      => ['VALUE01', '101'],
                    BLOCKNAME      => 'M115723131',
                    RENAME_COUNTER  => {
                        STRING2 => 'CARRIER_ID',
                        NUMBER1 => 'CELL_ID',
                    },
                    HEADER_FIELDS   => ['START_DATE', 'START_TIME'],
                    FIELDS_TO_KEY_PIF_FILENAME => ['START_DATE', 'START_TIME'],
                    DATA_FIELDS     => ['STRING1', 'STRING2', 'NUMBER1', 'NUMBER2'],
                    INVALID_VALUE_MATCH => {
                        '^' => '_',
                        '$' => '_',
                    },
                },
                PARAM_2 => {
                    PARAMETER      => ['VALUE01', '101'],
                    BLOCKNAME      => 'M115723132',
```

```
                        HEADER_FIELDS   => ['START_DATE', 'START_TIME'],
                        FIELDS_TO_KEY_PIF_FILENAME => ['START_DATE', 'START_TIME'],
                        DATA_FIELDS     => ['STRING1', 'STRING2', 'NUMBER1', 'NUMBER2'],
                        RENAME_COUNTER  => {
                            NUMBER1 => 'HOME_ID',
                            NUMBER2 => 'DISTANCE_ID',
                        },
                        INVALID_VALUE_MATCH => {
                            '^' => '_',
                            '$' => '_',
                        },
                    }
                },
                QUERY_2 => {
                    QUERY   => "SELECT to_char(START_TIME, 'DD-MM-YYYY') AS START_DATE, to_char(START_TIME, 'HH24MI') AS
START_TIME, STRING1, STRING2, NUMBER1, NUMBER2 FROM TABLE2 WHERE STRING1 = ? AND NUMBER1 = ?",
                    DATE_NAME => 'STARTDATE',
                    DATE_START_VALUE => "10-08-2008",
                    TIME_NAME => 'STARTTIME',
                    TIME_START_VALUE => "13:00:00",
                    INTERVAL => 1,
                    PARAM_1 => {
                        PARAMETER       => ['VALUE01', '101'],
                        BLOCKNAME       => 'M115723133',
                        HEADER_FIELDS   => ['START_DATE', 'START_TIME'],
                        FIELDS_TO_KEY_PIF_FILENAME => ['START_DATE', 'START_TIME'],
                        DATA_FIELDS     => ['STRING1', 'STRING2', 'NUMBER1', 'NUMBER2'],
                        INVALID_VALUE_MATCH => {
                            '^' => '_',
                            '$' => '_',
                        },
                    }
                },
            }
        },
```

S

Appendix A Notices and Trademarks

This appendix contains the following:

- Notices
- Trademarks

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome
Minato-ku
Tokyo 106-0032
Japan.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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
5300 Cork Airport Business Park
Kinsale Road
Cork
Ireland.

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.

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.

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

Trademarks

International Business Machines Corporation
5300 Cork Airport
Business Park
Kinsale Road
Cork
Ireland

Printed in the Republic of Ireland