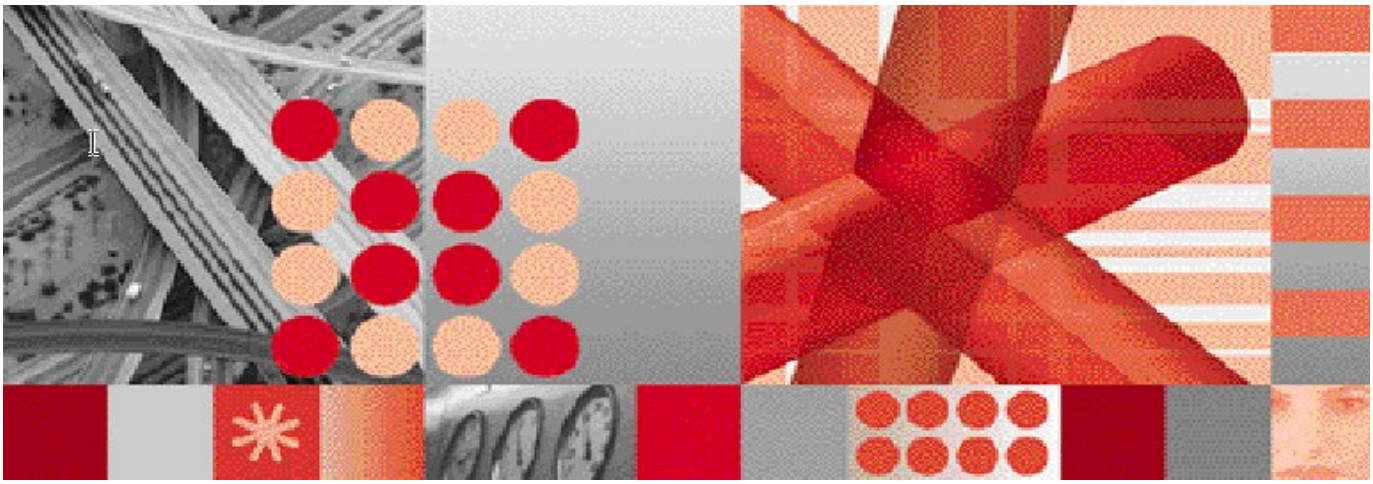
Tivoli Netcool® Gateway Framework

Version 3.5.1



Generic ASCII Gateway User Guide

Note: Before using this information and the product it supports, read the information in Notices on page 13.

This edition applies to Version 4.1 of IBM® Tivoli® Netcool® Gateway Framework and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright International Business Machines Corporation, 2009. All rights reserved.

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Table of Contents

1 About this Documentation 4
1.1 Audience 4
1.2 Required Skills and Knowledge 4
2 Associated Documents 5
2.1 Referenced Documents 5
2.2 Other Related Documents 5
3 Overview 6
3.1 The Gateway Framework 6
3.2 ASCII (Generic) Gateway Overview 6
3.2.1 Network Details 6
3.2.2 Data Types 6
3.2.3 Data Version Support 6
3.2.4 Data/File Format 6
3.2.5 Architectural extensions 7
4 Engine Rules and Configuration 8
4.1 Generic ASCII 8
4.1.1 Rule Configuration 8
4.1.2 PIF naming convention 11
5 Post Parser Rules and Configuration 11
6 Configuration Support 12
6.1 Tech Pack Support 12
Appendix A Notices and Trademarks 13

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 Gateway Framework you are strongly advised to read the release notes and any readme files distributed with your Gateway Framework 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 Associated Documents

The following documentation accompanies this release:

2.1 Referenced Documents

Document Name	Document Description
[Gateways Install Note]	This document describes the steps required to install and run a Gateway.

2.2 Other Related Documents

Document Name	Document Description
[Gateway Framework User Guide]	Gateway Framework User Guide describing the management and configuration of the Gateway Framework.

3 Overview

3.1 The Gateway Framework

The ASCII (Generic) Gateway, also referred to as the Vendor Gateway, uses the Gateway Framework as a container for the execution of its engine and post parser stages. The Gateway Framework and Vendor Gateway are decoupled 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 Vendor Engine and Post Parser rules for data transformation
- Intermediate (PIF) and output data (LIF) storage and management
- logging utilities
- cleanup and crash recovery
- statistics gathering

The Vendor Gateway plugs into the Gateway Framework and extends this functionality to provide the final Gateway that parses the vendor data.
More information on the standard Gateway configuration is contained in the Gateway Framework User Guide.
Only vendor specific configuration details will be described in this document.

3.2 ASCII (Generic) Gateway Overview

3.2.1 Network Details

The ASCII Gateway can be configured to parse any form of tabular ASCII data files from any network element in the network.

3.2.2 Data Types

The data can be either Performance data or Configuration data.

3.2.3 Data Version Support

This vendor gateway can be configured for various versions of the performance and configuration data.

3.2.4 Data/File Format

This section provides examples of the tabular ASCII data files.

Performance Data Layout

ASCII performance data must have distinctive header and data sections.
The header section may span across multiple lines as long as they are within a single continous section.
The data section may span across multiple lines after the header section, and may contain header information as well.
Example 1 below is a sample of a tabular ASCII performance data.

```
NodeName=RN_10.10.50.59
Category=Abis_InterfaceTraffic
ConfName=AbisTraffic
OIDs(Variables):
Var-1 = 1.3.6.1.4.1.6587.4.30.1.4.1.6(airAbisPeerMsgsSent)
Var-2 = 1.3.6.1.4.1.6587.4.30.1.4.1.5(airAbisPeerHellosReceived)
Var-3 = 1.3.6.1.4.1.6587.4.30.1.4.1.4(airAbisPeerHellosSent)
Var-4 = 1.3.6.1.4.1.6587.4.30.1.4.1.7(airAbisPeerMsgsReceived)

IndexOids:
1.3.6.1.4.1.6587.4.30.1.4.1.1(airAbisPeerIPAddr)
airAbisPeerIPAddr   Time                        Var-1  Var-2  Var-3  Var-4
-------------------------------------------------------------------------
10.10.50.1          02/05/2003 23:57:44        *N/A   N/A    N/A    N/A
10.10.50.1          02/05/2003 23:58:45        N/A    N/A    N/A    N/A
10.10.50.1          02/05/2003 23:59:44        N/A    N/A    N/A    N/A
10.10.50.1          02/06/2003 00:00:46        N/A    N/A    N/A    N/A
*N/A: “not available”
```

Example 1:

Configuration Data Layout

There is no specific configuration data layout. All data layout must refer to the respective vendor data specification.

File naming specification

There are no specific file naming conventions. All data layout must refer to the respective vendor data specification.

3.2.5 Architectural extensions

No external tools were used to parse the Nortel performance data.

4 Engine Rules and Configuration

The ASCII Gateway engine is able to parse raw data and configuration data.

4.1 Generic ASCII

This vendor gateway parses tabular ASCII raw data and configuration files into PIF format.

4.1.1 Rule Configuration

To evoke the Generic ASCII module the configuration option 'RULE_TYPE' must be set to 'Generic_ASCII'. The configuration options 'RULE_TYPE', 'RULE_DESC', 'INPUT_FILE_DESCRIPTION', 'INPUT_DIR_DEPTH', 'FILENAME_HEADER_FIELDS' and 'DIRECTORY_HEADER_FIELDS' are common to all productised gateways and are fully described in Gateways Framework User Guide; all other configuration options listed are specific to the Generic ASCII module.

Mandatory configuration entries:

- **HEADER_LINE_PROCESSING**: This configuration entry is used to parse the header section of the raw data.
- **DATA_LINE_PROCESSING**: This configuration entry is similar to HEDAER_LINE_PROCESSING except that it is configured to parse the data record. Both configuration entries must be an array of hashes with two sub-configuration entries.
    - **LINE_DESCRIPTION**: a regular expression matching the line
    - **LINE_PROCESSING**: a function to process the line. Arguments passed into the LINE_PROCESSING are as follows:

      $rule_obj - the current rule object. Object variables are passed into the function for extraction.
      $line_sref – the current line reference to be read.

      ```
      $$line_sref =~ /^START: (.+)$/;
      ```

      $h_href – the hash reference of the header record. The hash keys will be the header field name while the hash values are the header value, e.g.:

      ```
      $h_href->{CLASS} = $1;
      ```

      $d_href – the hash reference of the data record. The hash keys will be the data field name while the hash values are the data counters, e.g.:

      ```
      $d_href->{TRAFFIC} = shift @counters;
      ```

At the end of each LINE_PROCESSING functions should return one of the following strings:

- **NEXTLINE** – go to next line of the data.
- **GODATA** – end the HEADER_LINE_PROCESSING entry and begin the DATA_LINE_PROCESSING entry. Usually returned from within the HEADER_LINE_PROCESSING loop.
- **GOHEADER** – end the DATA_LINE_PROCESSING entry and begin HEADER_LINE_PROCESSING. Usually returned from within the DATA_LINE_PROCESSING loop.
- **NOMATCH** – skip the current LINE_PROCESSING, and continue with the next LINE_PROCESSING within the array.
- **WRITEPIF** – once a complete data record is read, begin writing the data record into PIF. Usually returned from within the DATA_LINE_PROCESSING loop.
- **-1** – File failed to parse due to a bad condition which stops the parsing and marks the file as bad.
- **-2** – File failed to parse which stops the parsing and log an error message.

Example of HEADER_LINE_PROCESSING and DATA_LINE_PROCESSING:

```
HEADER_LINE_PROCESSING     => [
   {
      LINE_DESCRIPTION => '^Header Lines$',
      LINE_PROCESSING  => sub {
         my ($rule_obj, $line_sref, $data_flag_sref, $h_href, $d_href) = @_;

         ..... # (initialize variables and code for parsing the data line)
         $$h_href->{HEADER_KEY} = 'HEADER_VALUE';
         .....
         return 'NEXTLINE';
      },
   },
   {
      LINE_DESCRIPTION => '^End of Header section*$',
      LINE_PROCESSING  => sub {
         my ($rule_obj, $line_sref, $data_flag_sref, $h_href, $d_href) = @_;

         ..... # (more parsing code)
         .....

         return 'GODATA';
      },
   },
],
DATA_LINE_PROCESSING => [
   {
      LINE_DESCRIPTION => '^Data Lines$',
      LINE_PROCESSING  => sub {
         my ($rule_obj, $line_sref, $data_flag_sref, $h_href, $d_href) = @_;
```

```
         ..... # (code to parse data line here)
         $$d_href->{DATA_KEY} = 'DATA_VALUE';
         .....
         return 'NEXTLINE';
      },
   },
   {
      LINE_DESCRIPTION => '^End of a data record*$',
      LINE_PROCESSING  => sub {
         my ($rule_obj, $line_sref, $data_flag_sref, $h_href, $d_href) = @_;

         ..... # (more parsing code)
         .....

         return 'WRITEPIF';
      },
   },
   {
      LINE_DESCRIPTION => '^End of all data records and start of a new
header$',
      LINE_PROCESSING  => sub {
         my ($rule_obj, $line_sref, $data_flag_sref, $h_href, $d_href) = @_;

         ..... # (reset variables)
         .....

         return 'GOHEADER';
      },
   },
],
```

Optional components:

- **OUTPUT_BLOCK_NAME**: A string to be output as data block name. By default is 'ASCII_DATA'.

  ```
  OUTPUT_BLOCK_NAME => 'ASCII_DATA',
  ```

- **INVALID_VALUE_MATCH**: Removes all the items from the raw input data file which match the characters listed before the gateway engine processes it.

  ```
  INVALID_VALUE_MATCH => [ "'", '"' ],
  ```

- **WHITESPACE_MODIFIER**: This option allows the user to change any " " (space) characters to underscores or other more useful character. This configuration entry will work only with INVALID_VALUE_MATCH configured.

  ```
  WHITESPACE_MODIFIER => '_',
  ```

- **HEADER_COUNTERS_TO_KEY_PIF_FILENAME**: A list of header info to construct the output file name.

  ```
  HEADER_COUNTERS_TO_KEY_PIF_FILENAME => [ 'TYPE', 'TIME', 'FILENAME' ],
  ```

- **DEFAULT_NULL_VALUE**: A value that is output when data is null.

```
DEFAULT_NULL_VALUE => 'NULL',
```

- **TIMESTAMP_MODE**: The name for the header info for entity name.

```
OBJECT_TYPE_NAME_FOR_HEADER_INFO => 'ENTITY',
```

- **HEADER_DATA_RECORD_PROCESSING**: This entry enables extra Perl code in the engine to do various manipulations to header and data records. This is the last process within the engine before the data is being output to the PIF files.

```
HEADER_DATA_RECORD_PROCESSING => sub {
   my ($$blkname_ref, $h_ref, $d_ref) = @_;

   # Header record processing
   if ($h_ref->{TIME} =~ /(\d{4})(\d{2})(\d{2})(\d{2})(\d{2})/) {
      my $day = "$1\-$2\-$3";
      my $time = "$4\:$5";
      &GenUtils::convert_date_format(\$day);

      # TIME will be output as a key PIF filename
      # as defined in HEADER_COUNTERS_TO_KEY_PIF_FILENAME
      # It is used by the UNPEGGER during post parsing
      $h_ref->{TIME} = "$day\_$time";
   }

   return 0; # Return 0 if successful
},
```

4.1.2 PIF naming convention

The PIF will be output with a name in the following format where '-#-' is used as a delimiter:
<header_counters_to_key_pif_filename>-#-<output_block_name>-#-I.pif

5 Post Parser Rules and Configuration

No customized post parser rules were implemented for this vendor gateway.

6 Configuration Support

6.1 Tech Pack Support

Tech pack Gateway Configurations which uses this vendor gateway are as follows:

- Huawei NSS – v61, v33
- Nortel NBSS – v13, v14, v15

The EngineConfig.pm and UserConfig.pm configurations for these Tech Packs are available with the Tech Packs respectively.

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

IBM, IBM logo, Tivoli, and Netcool are trademarks of International Business Machines Corporation in the United States, other countries or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Intel and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Other company, product or service names may be trademarks or service marks of others.

© Copyright IBM Corporation 2009

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

Printed in the Republic of Ireland
All Rights Reserved
IBM, IBM logo, Tivoli, and Netcool are trademarks of
International Business Machines Corporation in the
United States, other countries or both.

Other company, product and service names may be
trademarks or service marks of others.