

# **Siemens TNMS Gateway User Guide**

Gateway Release: 3.4.0

Date: 10 July 2008

---

## Contents

<b>CONTENTS</b>	<b>2</b>
<b>REFERENCES</b>	<b>3</b>
<b>GLOSSARY</b>	<b>3</b>
<b>PREFACE</b>	<b>4</b>
ABOUT THIS GUIDE	4
<b>1. OVERVIEW</b>	<b>5</b>
1.1 THE GATEWAY FRAMEWORK	5
1.2 SIEMENS TNMS GATEWAY OVERVIEW	5
1.2.1 Data Types	5
1.2.2 Data Version Support	5
1.2.3 Data/File Formats	6
1.2.4 File Naming Specification	7
1.2.5 DTD Validation	7
<b>2. ENGINE RULES AND CONFIGURATION</b>	<b>8</b>
2.1 SIEMENS_TNMS_XML	8
2.1.1 Processing Sequence	8
2.1.2 Rule Configuration	8
2.1.3 PIF file name convention	9
<b>3. POST PARSER RULES AND CONFIGURATION</b>	<b>9</b>
<b>4. TECH PACK SUPPORT</b>	<b>10</b>
<b>5. INSTALLATION SPECIFIC INFORMATION</b>	<b>10</b>

## **References**

[Gateway Framework User Guide]

## **Glossary**

LIF	Loader Input File
PIF	Parser Intermediate File
XML	EXtensible Mark-up Language

## Preface

### About this Guide

This guide details the Vendor specific information on the Gateway release for the Siemens TNMS Gateway. It contains the following information:

- *Chapter 1. Overview.* This chapter gives a brief description of the Vendor Gateway and the raw data formats it parses.
- *Chapter 2: Engine Rules and Configuration.* This chapter details the vendor specific rules for parsing the raw data and their configuration.
- *Chapter 3: Tech Pack Support.* This chapter describes any standard support for Tech Packs included with the Gateway.
- *Chapter 4: Installation specific information.* This chapter contains the customer installation specific information.

## 1. Overview

### 1.1 The Gateway Framework

The Siemens TNMS Gateway uses the Gateway Framework as a container for the execution of its engine and post parser stages. The Gateway Framework and Siemens TNMS Gateway 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 Siemens TNMS Gateway 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 Siemens TNMS Gateway simply plugs into the Gateway Framework and provides the functionality to parse the Siemens TNMS XML format.

More information on the standard Gateway configuration is contained in the Gateway Framework User Guide.

Only Siemens TNMS Gateway configuration details will be described in this document.

### 1.2 Siemens TNMS Gateway Overview

#### 1.2.1 Data Types

The *Siemens TNMS Gateway* can be configured to parse the Siemens TNMS XML format.

#### 1.2.2 Data Version Support

These are examples of systems currently using the Siemens TNMS Gateway

Vendor Subsystem	Release	Format
Siemens TNMS	V11.1.2	XML

### 1.2.3 Data/File Formats

The *Siemens TNMS Gateway* was developed to support the Siemens TNMS XML configuration file.

The sample XML configuration files are as below:

```
<?xml version="1.0" encoding = 'ISO-8859-1' ?>
<!-- TNMS Export Start-Time: Tue, 17 Jun 2008, 16:37:05 -->
<!-- TNMS Server: 'TNMSCOREBG' -->

<?xml-stylesheet type="text/xsl" href="TnmsV1.xsl" ?>
<TNMSData xmlns="http://www.siemens.com/tnms"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xsi:schemaLocation="http://www.siemens.com/tnms
TnmsV1.xsd"
            Version="1.8.5">

    <!-- PmLogDefs -->
    <PmLogDefs>
        <!-- PmLogDef -->
        <PmLogDef Name="KUKUSH_DC_201-CIIT_DC_203"
PmLogState="Active" FullCondition="WrapAround" FullWarningThreshold="0"
MeasurementIntervalType="15min" MeasurementIntervalFactor="15"
TandemConnectionMonitor="false" >
            <Size>200</Size>
            <PmPs>
                <PmP Name="MS64-TTP/0" AdminState="Unlocked"
EndType="Unknown">
                    <PortID EID= "Prt: {KUKUSH DC} {SPI-OS64-
TTP-wW 1-201}"/>
                </PmP>
                <PmP Name="RS64-TTP/0" AdminState="Unlocked"
EndType="Unknown">
                    <PortID EID= "Prt: {KUKUSH DC} {SPI-OS64-
TTP-wW 1-201}"/>
                </PmP>
                <PmP Name="MS64-TTP/0" AdminState="Unlocked"
EndType="Unknown">
                    <PortID EID= "Prt: {CIIT DC} {SPI-OS64-TTP-
eW 1-203}"/>
            </PmPs>
        </PmLogDef>
    </PmLogDefs>
</TNMSData>
```

```

        </PmP>
        <PmP Name="RS64-TTP/0" AdminState="Unlocked"
EndType="Unknown">
            <PortID EID= "Prt: {CIIT DC} {SPI-OS64-TTP-
eW 1-203}"/>
            </PmP>
        </PmPs>
        <PmCapabilities>
            <PmCapability Name="BBE"/>
            <PmCapability Name="ES"/>
            <PmCapability Name="SES"/>
            <PmCapability Name="UAS"/>
        </PmCapabilities>
    </PmLogDef>
    ...
    ...
</PmLogDefs>
</TNMSData>

```

#### 1.2.4 File Naming Specification

The naming convention for the XML configuration files are:

Export\_GroupTps\_Paths\_<n>.xml,

Export\_PerformanceLogs\_<n>.xml,

Export\_PortConnections\_<n>.xml. Where <n> equal to 1 or 2.

#### 1.2.5 DTD Validation

No DTD validation is supported in the Gateway. Deployments requiring such validation should seek to have this process completed by the equipment vendor element management systems supplying the XML.

## 2. Engine Rules and Configuration

### 2.1 SIEMENS\_TNMS\_XML

The essential task of the SIEMENS\_TNMS\_XML engine is to parse the XML data into output PIF files. This gateway uses its configuration to extract the data required for the performance tool.

#### 2.1.1 Processing Sequence

The SIEMENS\_TNMS\_XML processes the configuration data by parsing the XML file for header counters and data counters.

Header counters are parsed based on the XML comments defined within the **XML\_PIF\_HEADER\_COMMENT\_DESCRIPTION** entries.

Unique PIF files are created for each PIF mapping defined within **XML\_HEADER\_INFO\_FOR\_PIF\_FILENAME**.

All the data counters are parsed for each data block defined within the entry **XML\_UNIQUE\_PIF\_BLOCK\_ELEMENT\_NAMES**. Counter names and counter values are mapped to a single row in the PIF file.

#### 2.1.2 Rule Configuration

The following details the vendor specific rule entries for the Siemens TNMS XML engine rule.

- **XML\_UNIQUE\_PIF\_BLOCK\_ELEMENT\_NAMES**: This mandatory entry contains the data blocks to be parsed into a unique PIF file

```
XML_UNIQUE_PIF_BLOCK_ELEMENT_NAMES => [ 'PmLogDef' ],
```

- **XML\_IGNORE\_ELEMENTS**: This optional scalar or array entry specifies the XML element to be ignored by the parser. All child elements will be ignored as well if exist.

```
XML_IGNORE_ELEMENTS => [ 'Positions', 'Position1' ],
```

- **XML\_LINE\_COUNT\_TO\_TOP\_TRIM**: This optional scalar entry allows for a configurable number of lines from the top of the XML file to be removed before passing to the XML::Parser. These lines may include entries unsupported by XML::Parser, and need to be removed before parsing.

```
XML_LINE_COUNT_TO_TOP_TRIM => 5,
```

- **XML\_DUPLICATE\_COUNTERS**: An atomic element name, appearing as an actual counter in the PIF file should be included in this array list, if the name of the counter appears more than once in the XML block for the PIF record. This facilitates the placing of a unique incrementing numeric value on these values (vsDataType\_1, vsDataType\_2 etc.).

```
XML_DUPLICATE_COUNTERS => [ 'Layer' ],
```

- **XML\_ENABLE\_FULL\_ELEMENTS\_NAME**: This scalar entry enables or disables the element name to include the full element tree structure. For example the full

element name for <PmPs> in the example XML file will be TNMSData\_PmLogDefs\_PmLogDef\_PmPs.

```
XML_ENABLE_FULL_ELEMENTS_NAME => 'True',
```

- XML\_HEADER\_INFO\_FOR\_PIF\_FILENAME: Values of PIF header entries can be included in the output PIF filename by including their element names in this array. It is critical that enough header counters are included here, so as to ensure a unique PIF file.

```
XML_HEADER_INFO_FOR_PIF_FILENAME => [ 'FILE_NO' ],
```

- XML\_PIF\_HEADER\_COMMENT\_DESCRIPTION: This optional entry consists of tokens of the XML comment and the corresponding Perl Regular Expressions that matches the XML comment name. Each token will be validated and recorded as the header counter. This data is common to all PIF output files.

```
XML_PIF_HEADER_COMMENT_DESCRIPTION => {
    Network_Id => '^TNMS\sServer\:\s\'(.*)\'$'
},
```

- HEADER\_DATA\_RECORD\_PROCESSING: This is an optional entry whose value points to a configured Perl subroutine. The purpose of this entry is to add custom processing for a deployment, such as add a new counter. But it could be any processing on the header and data record. This entry deals with counter name/value pairs in the current header record and is invoked for each header record encountered. Example:

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

    # Replace whitespace in NAME with underscore
    $d_ref->{NAME} =~ s/ /\_/g;
}
```

### 2.1.3 PIF file name convention

The name of the PIF file will be in the following format:

<BLOCK\_NAME>-#-<XML\_HEADER\_INFO\_FOR\_PIF\_FILENAME>-#-CONFIG-#-I.pif

## 3. Post Parser Rules and Configuration

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

## **4. Tech Pack Support**

Tech pack support is included in the Siemens TNMS Gateway for the following Performance Manager solutions.

- Siemens TNSM V11.1.2

The EngineConfig.pm and UserConfig.pm configuration files for these Tech Packs are located in the tech pack.

## **5. Installation Specific Information**

None.