CA Spectrum®

TL1 Gateway User Guide

Release 9.4

CA Technologies Product References

This document references the following CA Technologies products:

- CA Spectrum® Infrastructure Manager (CA Spectrum)
- CA Spectrum® TL1 Gateway (TL1 Gateway)

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at http://ca.com/docs.

Contents

Chapter 1: Getting Started

This section contains the following topics:

What Is TL1?

Transaction Language 1 (TL1), occasionally referred to as MML (Man Machine Language), is a widely used protocol in telecommunications management. The TL1 protocol can be used to manage most telecom network elements in North America.

Unlike SNMP, TL1 is a human-machine interface that contains human-readable strings. Also, unlike SNMP, TL1 includes no concept of a MIB. TL1 was originally specified by Bellcore in 1986 and is now maintained by Ericsson.

What Is TL1 Gateway?

The TL1 Gateway for CA Spectrum translates TL1 events and alarms originating from a TL1 device into CA Spectrum events and alarms. The gateway acts as a mediator between TL1 devices and CA Spectrum.

Each TL1 device is represented by a corresponding device model within CA Spectrum. As a result, you can launch the Enterprise Alarm Manager application for a particular model so that you can check for certain alarm conditions and initiate corrective action. TL1 Gateway also supports TL1 devices that are accessible through proxy devices, also known as Gateway Network Element (GNE) devices.

TL1 Gateway implements the full range of TL1 Condition Types as specified by Telcordia. Telcordia has been acquired by Ericsson. Multiple websites provide lists of current network element and transport surveillance messages.

What Does TL1 Gateway Include?

TL1 Gateway for CA Spectrum provides the following components:

- TL1-specific inference handlers that plug into a SpectroSERVER.
- A daemon (TL1d) that handles communication between the TL1 devices and the SpectroSERVER.
- A model type (Gen_TL1_Dev) for generic TL1 devices.
- A utility (tl1map) to manage TL1 AlarmMaps.

Chapter 2: Installing and Using TL1 Gateway

This section contains the following topics:

Prerequisites for TL1 Gateway

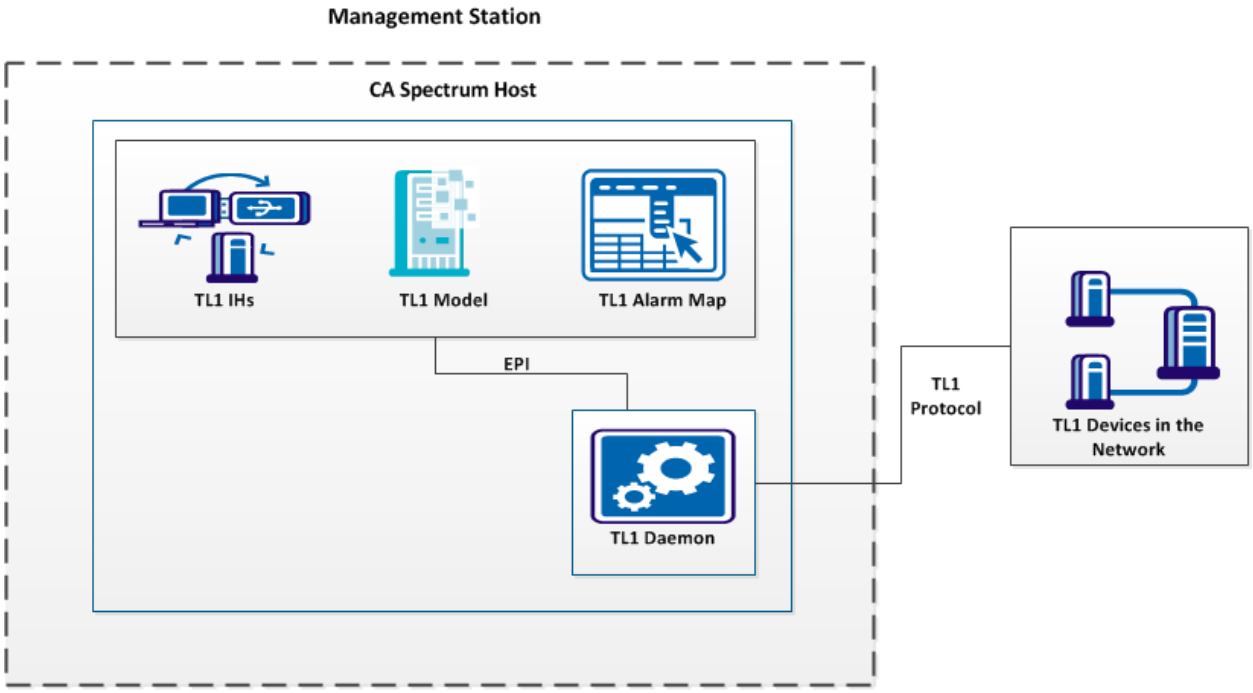
TL1 Gateway is designed such that it can either be run on the same machine CA Spectrum is running on, or on a separate machine. Diagrams illustrating both installation models are provided on the subsequent pages. Running TL1 Gateway on a separate machine is recommended if any of the following conditions apply:

- The CA Spectrum machine hosts a SpectroSERVER that has a high workload.
- The CA Spectrum machine is short on resources like CPU and RAM.
- The CA Spectrum machine is also used by applications other than CA Spectrum.
- A high volume of TL1 traffic has to be processed.

Note: TL1 Gateway does not support IPv6.

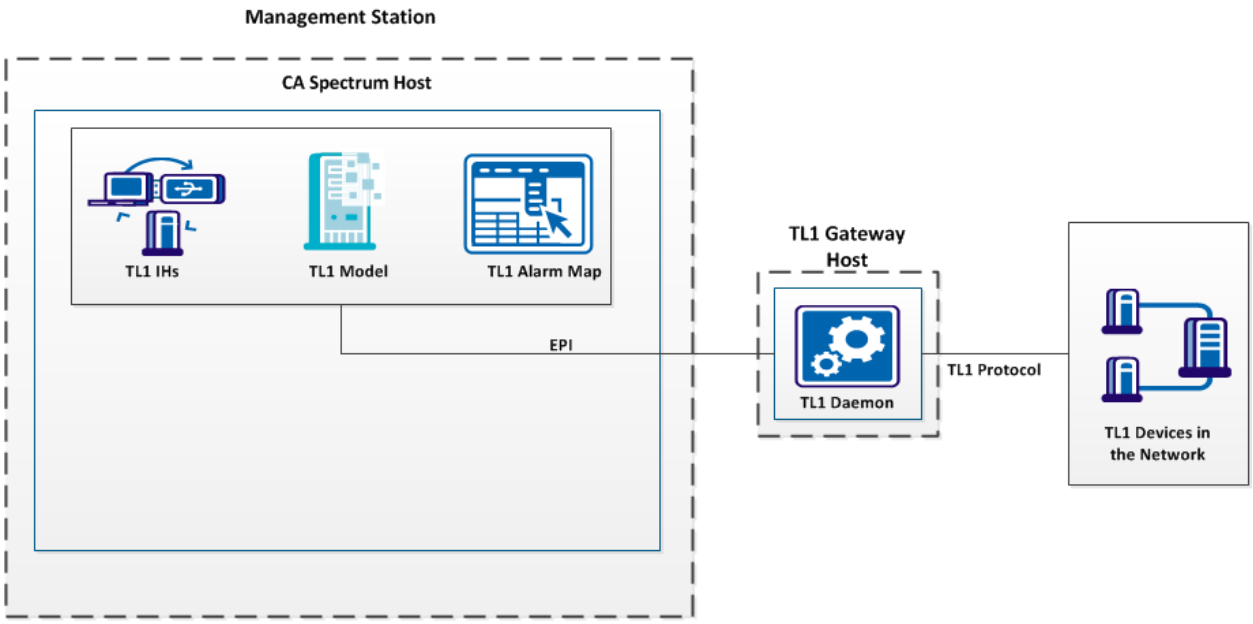
The following diagram shows an example of a TL1 Gateway installed on CA Spectrum host:

TL1 Gateway Installed on CA Spectrum Host



The following diagram shows an example of a TL1 Gateway installed on a remote host:

TL1 Gateway Installed on a Remote Host



Installation Options

If your configuration requires TL1 Gateway to be run on the same machine CA Spectrum is running on, then the gateway has already been installed during the major CA Spectrum installation. In this case, all required TL1 Gateway components can be found in a subdirectory called TL1Apps, which is located in your top-level CA Spectrum directory. The TL1 Apps directory contains the following files:

- TL1d -- the gateway daemon
- tl1map -- the AlarmMap utility

Install on a Non-CA Spectrum Machine

In some cases, where a certain level of performance is needed, it is preferable to install the TL1 daemon component on a separate server that is dedicated to TL1 message processing. The instructions vary slightly depending on the platform you are using. In either case, you must find and run the self-extracting installer file on the CA Spectrum application DVD.

The installer launches a series of dialogs that prompt you to supply the following information:

- confirmation of the license agreement
- the extraction key
- installation target area

Note: Regardless of the platform on which CA Spectrum is running (Solaris or Windows), the TL1 daemon component can be installed and operated. For example, you can run CA Spectrum on a Solaris server, while the TL1 daemon is hosted by a Windows server. Such a heterogeneous TL1 Gateway configuration is fully supported.

Follow these steps on Windows:

1. Navigate to the TL1GW folder on the CA Spectrum application DVD.
2. Run the tl1inst.exe program.
3. Copy the resulting files (listed below) to the proper locations on the NT machine:
   - TL1d.exe
   - tl1map.exe
   - orb.dll
   - orb_p.dll

   Copy the executables TL1d.exe and tl1map.exe to the preferred locations. Copy the DLL files to the system area that is used for third-party DLL files.

Follow these steps on Solaris:

1. Navigate to the TL1GW directory on the CA Spectrum DVD.
2. Run the tl1inst.bin program.
3. Copy the following resulting files to the proper locations on the Solaris server:
   - TL1d
   - tl1map
   - liborb.so

   Copy the executables TL1d and tl1map to the preferred locations. Copy the shared library liborb.so to the preferred system area for third-party libraries: /usr/lib.

   Note: If you are using a location other than /usr/lib, update the environment variable LD_LIBRARY_PATH to include that location.

Setting up Port Numbers

When you create a TL1 device model, specify the two port numbers that are described below. Before, you assign any port numbers, first check the local system administration or security policies.

TL1 Gateway Port

Used for communication between the TL1 daemon (TL1d) and the SpectroSERVER. At model creation, a default port number of 64222 is supplied. When choosing a different port number, we urge you not to use Port 65535. Ask a system administrator to find a port number in the dynamic/private range 49152 to 65534.

The TL1 Gateway port number can be specified as a command line argument when launching TL1d; otherwise, the default of 64222 is assumed.

TL1 Device Port

Used for communication between the physical TL1 device and the TL1 daemon. Check with the network administrator for the port number to use; it depends on device configuration. It is likely to be a port number specified in the IANA standard document. Check the “Registered Port Numbers” section. Likely candidates are: 2361, 3081, 3082 and 3083. IANA port assignments are available on the Internet.

Supply the TL1 port number in the model creation dialog.

Creating a TL1 Device Model

The TL1 Device models are used to check for alarm conditions and initiate corrective action through Enterprise Alarm Manager. The TL1 devices must be modeled manually as they cannot be modeled through Spectrum Discovery. For more information about Discovery, see the *Modeling and Managing Your IT Infrastructure Administrator Guide*.

Follow these steps:

1. From any OneClick Topology tab view, select the Model by Type tool bar icon.
2. Select the 'All Model Types' tab and enter TL1 into the Filter field.
3. Select Gen_TL1_Dev to model a generic TL1 device and click OK.
4. Complete the following fields to configure the TL1 device model:

   Model Name

   Specifies a unique name for this model.

   TL1 Gateway Address

   Specifies the IP address of the server that hosts the TL1 daemon (TL1d).

TL1 Gateway Port

Specifies the port number for communication between the TL1 daemon and CA Spectrum.

Default: 64222

TL1 Device Address

Specifies the IP address of the TL1 device.

Note: In some cases, this address is the IP address of a dedicated TL1 device that acts as a proxy for all the other TL1 devices in the network.

TL1 Device Port

Specifies the port number where the TL1 device agent is listening (also known as the “craft” port).

Max Telnet Sessions

Specifies the total number of Telnet sessions TL1 Gateway can make to Gateway Network Element (GNE).

Default: 1

Max Logins Per Telnet

Specifies the number of logins that are permitted to be active simultaneously per telnet session. This field value depends on the GNE.

TID

Specifies the “Target ID” for the TL1 device. This parameter is part of the TL1 addressing concept, and is required to uniquely identify a device. The TID is a predefined alphanumeric string. Obtain it from the local network administrator.

User Name

Specifies the User Id for the maintenance user, as configured in the TL1 device.

Password

Specifies the password for the maintenance user.

Security String

Prevents selected users from viewing this model.

Poll Interval (sec)

Specifies the frequency at which the device is polled for status updates. Change the value as needed to increase or decrease the polling interval.

Default: 60 (300 for some model types)

Note: If you increase the time between polling intervals, less bandwidth is required for traffic management. However, you receive device status updates less frequently.We recommend using the default polling interval for critical devices and using 600 seconds for less important devices.

Log Ratio

Defines how many times CA Spectrum polls devices for updates before logging the results.

Default: 10 (CA Spectrum logs the polling results after it polls the device every tenth time).

Note: No entries or adjustments to the defaults are required for the remaining fields. For more information about field settings, see the *Administrator Guide.*

5. Click OK.

   A TL1 device model icon is displayed.

Max Telnet Sessions and Max Logins Per Telnet Attribute Considerations

Max Telnet Sessions

The Max Telnet Sessions field enables you to specify the total number of telnet sessions that TL1 Gateway can make to Gateway Network Element (GNE). Default is 1.

TL1 devices exist in a ring or group. The Gateway Network Element acts as a go between the rest of the network and the TL1 devices within the ring. Therefore, CA Spectrum makes a telnet connection to the GNE.

This will vary with hardware maximum device specifications. CA Spectrum can use as many sessions as the Max Telnet Sessions value has been set to allow. You can specify Max Telnet Sessions to one less than the hardware maximum in order to leave one available for the Network Administrator, but it is not required.

For any given ring, the number of telnet sessions used is the highest Max Telnet Sessions value among all the Gen_TL1_Dev models that are associated with that ring. So, if you have ten devices modeled for a single ring, if all the Max Telnet Sessions values are 1, then the TL1 daemon (TL1d) uses at most one telnet session when communicating with all the devices on that ring. If nine of the models have a Max Telnet Sessions value of 1 and one of the models has a Max Telnet Sessions value of 4, then the TL1d can use up to four concurrent telnet sessions when communicating with ANY of the devices on that ring.

The recommended usage for Max Telnet Sessions is to set it to the desired value on the Gen_TL1_Dev model which represents the TL1 proxy device, and to leave the value at its default for other Gen_TL1_Dev models. That way, if you want to change the value, you only have to change it on one model.

Max Logins Per Telnet

The Max Logins Per Telnet field enables you to enter the number of logins that are permitted to be active simultaneously per telnet session. This depends on the GNE. Some authorize one active login at a time, others allocate more.

As with Max Telnet Session, when two or more devices from the same ring are modeled, the TL1 deamon (TL1d) uses the highest Max Logins Per Telnet value among all the Gen_TL1-Dev models that are associated with that ring.

The recommended usage for Max Logins Per Telnet is to set it to the desired value on the Gen_TL1_Dev model which represents the TL1 proxy device, and to leave the value at it's default for other Gen_TL1_Dev models. That way, if you want to change the value, you only have to change it on one model.

Important Processing Information for Max Telnet Sessions and Max Logins Per Telnet

It is important to understand the difference between having a value for Max Logins Per Telnet of 1 versus a value greater than 1.

When Max Logins Per Telnet Equals 1

When Max Logins Per Telnet equals one, the telnet connections are time-shared. Therefore, polls cannot occur simultaneously, but instead are carried out in sequence. The maximum number of devices manageable is dependent primarily on the polling interval and the value of Max Telnet Sessions. Congestion and slow device response can adversely affect CA Spectrum's management of the devices.

When Max Logins Per Telnet Is Greater Than 1

When Max Logins Per Telnet is greater than one, polls can occur simultaneously. The maximum number of devices manageable is a hard limit of Max Telnet Sessions multiplied with Max Logins Per Telnet. Therefore, congestion and slow device response has less impact on CA Spectrum's ability to properly manage the devices.

The TL1 daemon (TL1d) communicates with TL1 devices over shared telnet sessions. That is, if you have say, ten TL1 devices on a single TL1 ring, TL1 Gateway can send ping requests to and receive messages from all ten devices over a single telnet connection.

The TL1 daemon (TL1d) can also have several concurrent telnet sessions open to the same TL1 ring. So, TL1 Gateway could be using three telnet sessions open at the same time to communicate with ten TL1 devices on a single ring.

Congestion Considerations

Because communication is shared over the connections, congestion can occur. The amount of congestion is determined by:

- Number of concurrent telnet sessions available to TL1 Gateway for use to the TL1 ring in question
- Number of TL1 devices on the ring that are managed in CA Spectrum
- Polling interval of the TL1 models
- Frequency of alerts sent to TL1 Gateway from the TL1 devices
- Time delay in the TL1 devices responding to commands from TL1 Gateway

If congestion over the telnet session is bad enough, ping requests sent by the SpectroSERVER will not return in time, and the TL1 models will sporadically go into a contact lost state.

If the network congestion is caused by too many alerts from the TL1 devices, a solution might be to specify an autonomous port for the devices.

If Max Logins Per Telnet equals one, the following two solutions are available:

- Increase the polling interval of the TL1 models -- the devices are polled less often causing less congestion, resulting in better performance.
- Increase the telnet sessions to TL1 devices ratio, by either increasing the number of concurrent telnet sessions the TL1d can use, or by decreasing the number of TL1 devices on the ring.

Upgrade Considerations

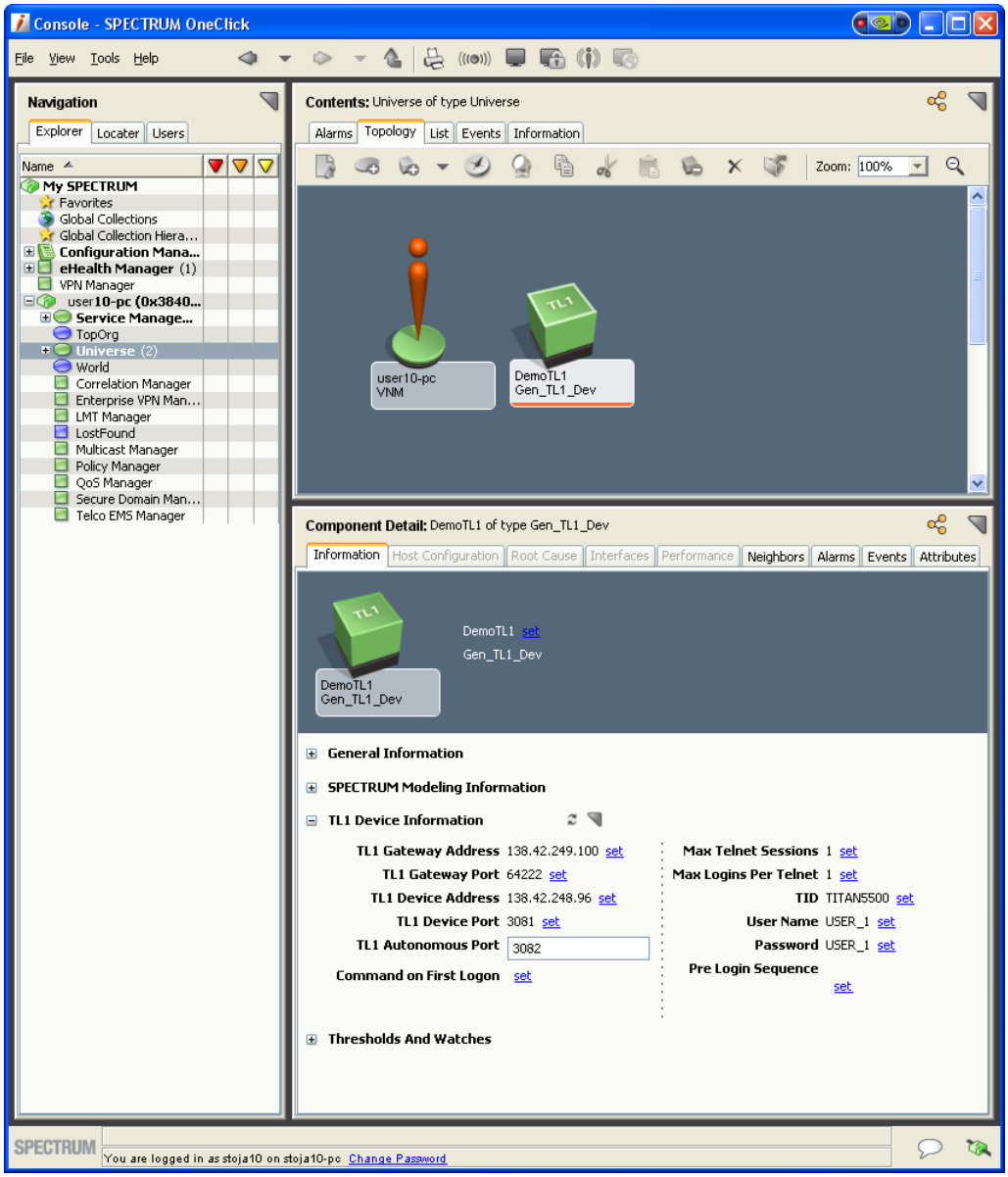
Of particular importance to customers who are upgrading:

- Due to the overhead caused by the new functionality, we have increased the default value for the DCM Timeout attribute (0x110c4) from 3000 to 5000. However, if the customer is upgrading, this change will not be made because the Timeout attribute has the 'Preserve Value' flag set.
- If the DMC timeout value is left at 3000, the ping requests sent by the SpectroSERVER will probably timeout sporadically. After upgrading, customers should change the Timeout value for all Gen_TL1_Dev models that they have currently modeled to be 2000 milliseconds higher than their current value. Likewise, they must also increase the Gen_TL1_Dev model type default value for Timeout by 2000 milliseconds.

TL1 Devices with Autonomous Port

Some TL1 devices support command ports only and have a port dedicated to autonomous messages. TL1 Gateway supports this option.

To use it, highlight the TL1 device icon and select the Information tab in the Component Detail view. The TL1 Autonomous Port can be set in the TL1 Device Information section.



Pre Login Sequence

Before logging into a TL1 device, the TL1 daemon can send a few characters to the device to wake up the telnet session. These characters are stored in the Pre Login Sequence attribute. The default is a single newline character.

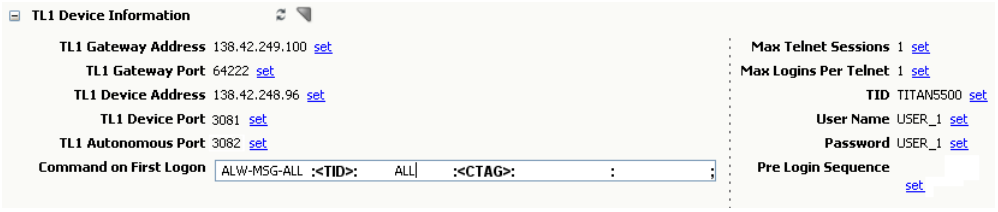
For some cases, multiple newlines may be more appropriate. To change the Pre Login Sequence, highlight the TL1 device icon and select the Information tab in the Component Detail view. The Pre Login Sequence can be set in the TL1 Device Information section.

Command on First Logon

The value that appears in the Command on First Logon field is issued to the device the first time you log into that device. In the image shown below, the ALW-MSG-ALL command enables the device to send alerts to the TL1 Gateway. Note that some devices do not send alerts automatically. The gateway can then relay the alerts to CA Spectrum as it receives them.

You can customize the Command on First Logon for any command that requires customization. CA Spectrum automatically replaces the <TID> and <CTAG> values. The <TID> value uses the TID value that you specified in the Create Model Type dialog. The <CTAG> value is automatically generated by the TL1 Gateway for proper communication.

The following image shows an example of the Command on First Logon field in the TL1 Device Information subview.



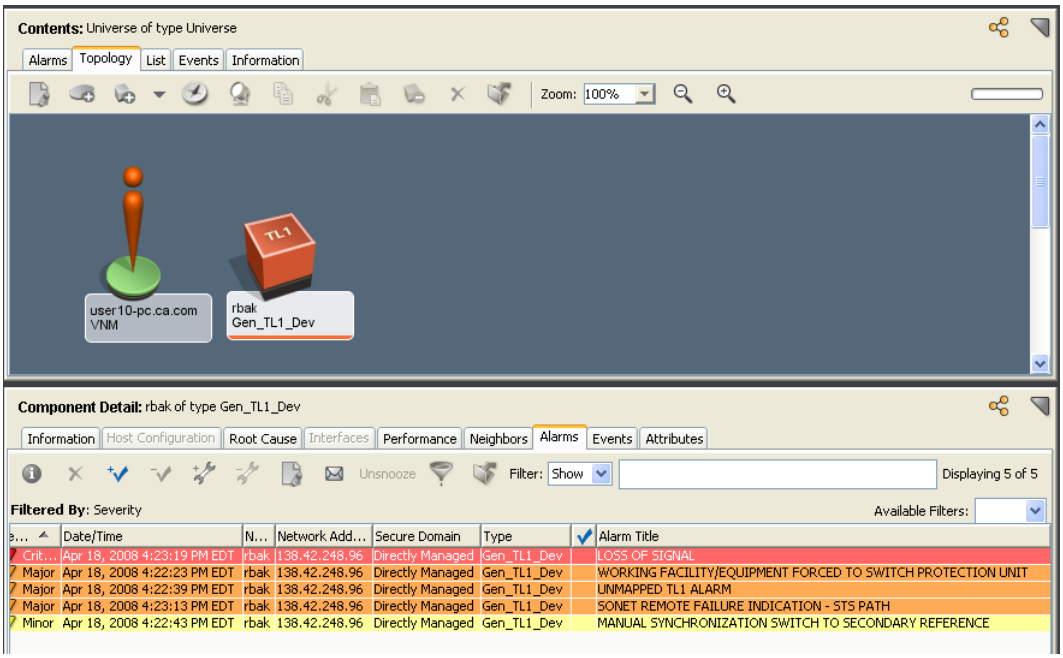
The TL1 AlarmMap

The TL1 Gateway AlarmMap is an internal mapping of TL1 events and alarms to CA Spectrum alerts. The AlarmMap is used by the TL1 daemon to determine whether an incoming TL1 event or alarm will be discarded or be converted to an alert code and forwarded to CA Spectrum. Thus the AlarmMap functions as a filter to discard unnecessary TL1 events or alarms.

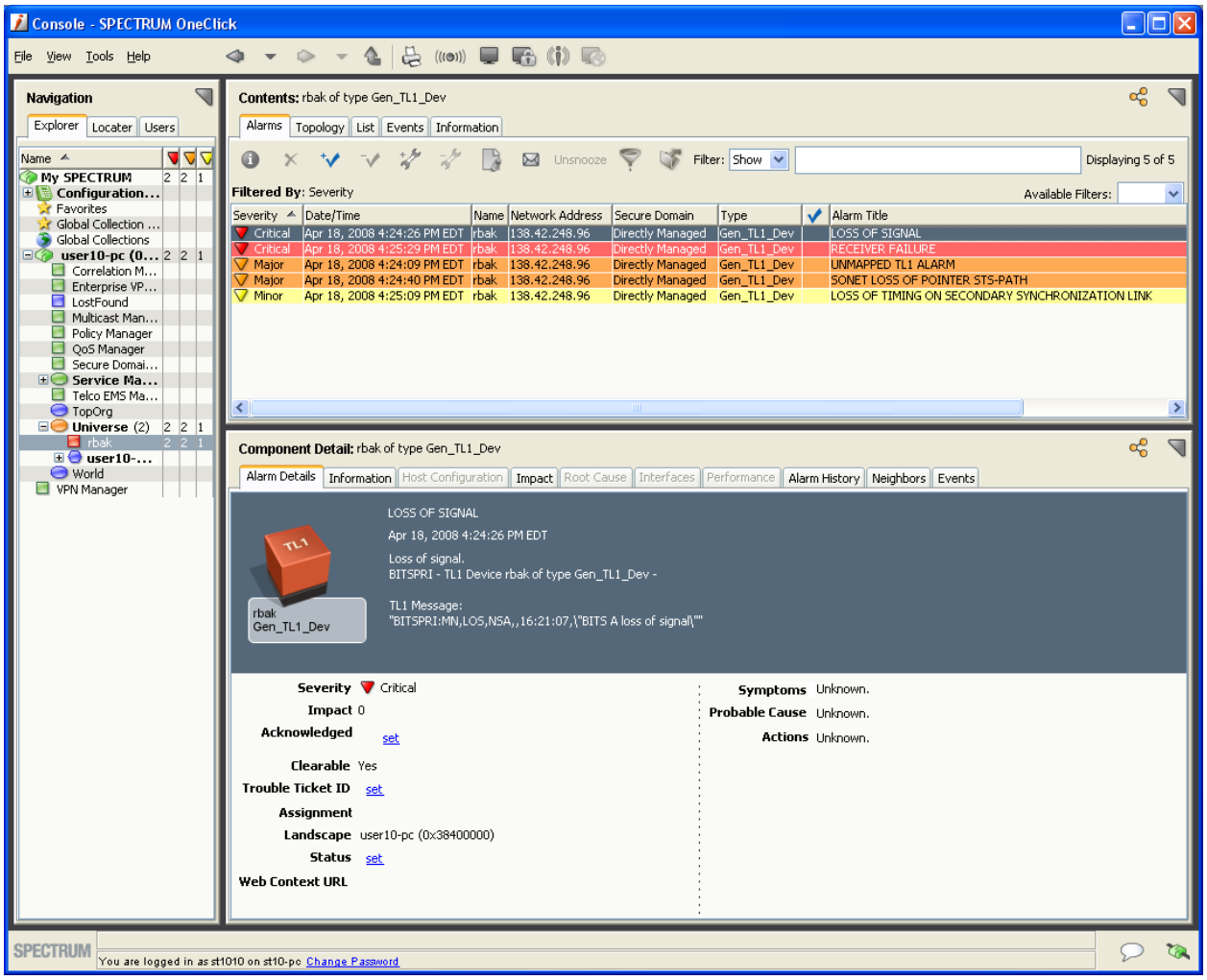
Another mapping file is also required to support the AlarmMap. The alert code that is specified in the TL1 AlarmMap is processed by a corresponding AlertMap file on the SpectroSERVER. If the AlertMap file is missing, TL1 alerts are not properly mapped to events. For more information, see AlertMaps for TL1 (see page 28).

An AlarmMap usually includes a default entry, which is used for any TL1 events or alarms that lack a specific mapping. That is, all unmapped TL1 events/alarms result in the same CA Spectrum alert that is specified in the default entry.

TL1 alarms are displayed in the following image:



TL1 details per alarms are displayed in the following image:



Format of the AlarmMap

The AlarmMap is a sequence of records, each comprising the following five comma-separated fields:

```
<type>,<condition>,<alert-code>,<clr-alert-code>,<process>
```

<type>

A text string specifying the entry type as either “ALM” (for alarm) or “EVT” (for event).

<condition>

A text string identifying a TL1 condition type (TL1 alarms) as specified in the Telcordia standards document GR-833-CORE. If this string is blank, or empty, the record serves as the default entry for the AlarmMap. If the string ends in a hyphen (e.g., OGCCS-), all conditions that start with that string are mapped by this entry. It resembles a “wildcard.”

<alert-code>

A hexadecimal literal with a leading “0x” that specifies the alert code to be used by CA Spectrum for this TL1 event/alarm.

<clr-alert-code>

A hexadecimal literal with a leading “0x” that specifies the code that CA Spectrum uses to clear the TL1 event/alarm.

<process>

Is a boolean value (TRUE or FALSE) that specifies whether the event/alarm is processed. If TRUE, an alert is generated for CA Spectrum. If FALSE, the incoming TL1 event/alarm is discarded and nothing is forwarded to CA Spectrum.

The following is an example of an AlarmMap:

```
ALM,              ,0x3d50001  ,0x3d50002  ,TRUE  <------ DEFAULT
ALM,LINETERM      ,0x3d500f3  ,0x3d500f4  ,TRUE
ALM,ECOI          ,0x3d501b9  ,0x3d501ba  ,TRUE
ALM,SLOR          ,0x3d501c9  ,0x3d501ca  ,TRUE
EVT,NTYDGSP       ,0x3d5009d ,0x3d5009e ,TRUE
```

The commas can be surrounded by spaces. Everything after the <process> field and a space is treated as comment for better readability.

The contents of the AlarmMap are critical to the performance of TL1 Gateway. Configure the AlarmMap such that only those TL1 alarms in which you are interested are processed. In other words, verify that the <process> value is set to FALSE for all TL1 alarms that you do not want to generate a CA Spectrum alert.

Be aware that if an “unwanted” TL1 event/alarm does not appear in the map, it is still processed through the default mapping, assuming that the map has a default entry. To prevent this extra processing, you can take one of the following steps:

- Include all unwanted TL1 events/alarms in the map with their <*process*> values set to FALSE.
- Remove the default entry.

The tl1map Utility

TL1 Gateway includes a command-line tool called tl1map that lets you inspect the AlarmMap and perform either of the following two operations:

- Extract an AlarmMap from either a TL1 device model or from the TL1 device model type (Gen_TL1_Dev).
- Import an AlarmMap in the form of a text file either to a TL1 device model or to the Gen_TL1_Dev model type.

The tl1map tool is located in your top-level CA Spectrum directory's TL1Apps subdirectory. You can run tl1map either from the shell command line or from within scripts (if you need to perform more complicated AlarmMap manipulations). In either case, tl1map is a SpectroSERVER client, so SpectroSERVER must be up and running.

Note: The tl1map utility is dedicated to models/model types of the Gen_TL1_Dev hierarchy only. It is not intended to be used with any other models/model types and could cause potential data corruption if not used properly.

Syntax

```
.tl1map {load | dump} -f mfile {-t | -m} handle [host]
```

dump

Extracts the internal AlarmMap in text format and saves it to the file specified by the -f option, in text format.

load

Processes the input file specified by the -f option and writes an AlarmMap to the corresponding attribute.

-f mfile

Indicates that the next entry will provide the name of a particular map file (mfile). In load mode, the supplied input file is expected to have AlarmMap syntax, and it will be converted into an internal TL1 AlarmMap object. When in dump mode, the contents of the already existing, internal AlarmMap will be extracted, formatted, and written to the specified mfile. This file is then suitable for visual inspection and processing by a text editor, so it can be changed, and re-imported using the load option.

-t

Specifies that the handle provided will be the model type handle of the TL1 device model type.

-m

Specifies that the handle provided will the model handle of a TL1 device model.

handle

Specifies the handle of either the TL1 device model type (if the -t flag is used) or the TL1 device model (if the -m flag is used).

Limits: Hexadecimal format

host

The name of the CA Spectrum host. This is only needed when accessing a remote host, or if the Unix hostname command reports a mixed-case name, while the “true” host name was advertised in lower-case format. In any case, tl1map should be run on the local host.

Note: The -f, -t, and -m -f options can appear in any order.

AlarmMap Extraction

If tl1map is invoked with the dump option, it will read the current AlarmMap attribute value from either a model (-m option), or a model-type (-t option), convert it into a human-readable format, and write it to the file specified by the -f option. That file will also contain an informational header, which includes things like model name, model type name, model handle, model type handle, host name, and the current time stamp (based on the machine where the utility was invoked). The header is formatted as a comment, so it will not affect any subsequent import operation using this machine-generated file.

Typically, you would examine this AlarmMap file with a text editor, make any desired modifications, and then import the file again by invoking tl1map with the load option.

Note: The entries in the output file are in no particular order. If you wish to compare two different AlarmMaps, you need to do the following:

- Strip off the comment header.
- Sort the files.
- Perform the actual comparison.

Importing an AlarmMap

To import an AlarmMap under any of the following scenarios, use the tl1map utility with the load option:

- Creating an initial AlarmMap based on documents from a TL1 device vendor or on TL1 standards documents.
- Extracting a previously stored AlarmMap, applying some changes, and importing the changed version again.
- Extracting the AlarmMap from one model/model type, and importing it to another model/model type (possibly on another host)

When importing an initial AlarmMap from a file, you have to make sure that the file conforms with the format specifications for AlarmMap files.

The tl1map utility will display limited diagnostic messages if it encounters illegal entries in the AlarmMap file. If there were any errors, no data will be written to the model/model type attribute.

Examples of tl1map Usage

The following examples illustrate how the tl1map utility might be used to perform various specific tasks. Note that the examples use /dev/stdin and /dev/stdout. Since these are not available on NT, you would therefore use intermediate files instead of Unix-style pipes.

To import an AlarmMap to model 0x146007

```
tl1map load -f custom.amap -m 0x146007
```

To export the AlarmMap to a file

Exporting the AlarmMap to a file enables it to be modified. The modified file can be reloaded/imported.

```
tl1map dump -f exp.amap -t 0x4010000
```

...then, after you edit and save exp.amap ...

```
tl1map load -f exp.amap -t 0x4010000
```

To copy the AlarmMap from one model to another

```
tl1map dump -m0x1503fcc -f/dev/stdout |
tl1map load -m0x1504003 -f/dev/stdin
```

To delete the "LINETERM" entry from the AlarmMap

```
tl1map dump -m0x1504003 -f/dev/stdout |
sed '/LINETERM/d' |
tl1map load -m0x1504003 -f/dev/stdin
```

To copy a model-based map from one host to a model type-based map on another host:

```
tl1map dump -m0x1504003 -f/dev/stdout hosta |
tl1map load -t0x25040cc -f/dev/stdin hostb
```

The following example shows a shell script that could be used to implement a simple AlarmMap editor:

```
#! /bin/ksh
#
# A simple AlarmMap editor
#
# Syntax:  tl1edit { -m | -f } handle [ host ]
#

# Your favorite text editor
if [[ -z $EDITOR ]]; then
EDITOR=vi
fi

# Generate a unique, temp file name
MF=`date '+%Y%m%d%H%M%S.almap'`

# Dump the AlarmMap to that temp file
tl1map dump -f $MF $*

# Make a backup copy
cp $MF $MF.orig

# Invoke your favorite editor on that file
$EDITOR $MF

# If there are no changes, remove temp mapfiles & exit
diff $MF $MF.orig > /dev/null && rm -rf $MF $MF.orig && exit

# If there were changes, load them into the attribute
tl1map load -f $MF $*
```

AlertMaps for TL1

The TL1 AlarmMap converts a TL1 autonomous message into an alert (not an event) that is processed by CA Spectrum. CA Spectrum handles these alerts similarly to SNMP traps. Configure the AlertMap with the alert code that is specified in the TL1 Alarm Map.

Note: The alert code that is specified in the AlertMap is a decimal number rather than an SNMP trap OID.

Each alert from the TL1 Gateway offers the following standard, unchangeable varbinds:

- Varbind 1: AID
- Varbind 2: Severity
- Varbind 3: Condition
- Varbind 4: Alarm Message
- Varbind 5: Alert Code (as specified in the TL1 AlarmMap)

The AlertMap must map these alert variables to event variables, one for one.

A supported entry in the AlertMap resembles the following syntax:

```
64352257 0x3d5f001 1(1,0) 2(2,0) 3(3,0) 4(4,0) 5(5,0)
```

The above entry maps alert code 64352257 (0x3d5f001 in decimal) to event code 0x3d5f001, and copies each of the five alert variables to the event.

For more information about the AlertMap, EventDisp, CsEvFormat, and CsPCause files, see the *Event Configuration User Guide*.

Chapter 3: Managing TL1 Gateway Daemon

This section contains the following topics:

TL1 Gateway Daemon

A TL1 Gateway daemon (TL1d) must be up and running in order to translate alarms from the modeled TL1 device into CA Spectrum events and alarms. The daemon is located in your top-level CA Spectrum directory's TL1Apps subdirectory. It has the following syntax:

```
TL1d [-p gw-port]
```

-p gw-port

Specifies that a TL1 Gateway port (gw-port) other than the one specified at model creation will be designated for communication between the daemon and SpectroSERVER.

Default: 64222

Note: CA recommends that you *not* use the value 65535, as it is likely to conflict with other applications.

Start the TL1 Daemon on a CA Spectrum Server

Start the TL1 daemon to start handling alarms from TL1 devices.

This procedure varies only slightly (see Step 3) for the Solaris and Windows platforms.

Follow these steps:

1. Launch a command shell window.
2. Navigate to the TL1Apps subdirectory in your top-level CA Spectrum directory.

3. Enter one of the following commands:

   ```
   TL1d ( or ./TL1d on Solaris)
   ```

   or, if you want to use a different gateway port...

   ```
   TL1d ( or ./TL1d on Solaris) -p <your-gw-port number>
   ```

   As soon as you see “@(#)” followed by the actual port number, the daemon is running. You can minimize the command shell window.

Start the TL1 Daemon on a Remote Server

This procedure is described separately for NT and Solaris below. In either case, you need to know the IP address of the server that hosts the CORBA osagent in your environment. This server is most likely the same computer where CA Spectrum is running.

Take a few steps to start the TL1 Daemon on a remote server on Windows.

Follow these steps:

1. Launch a command shell window.
2. Navigate to the directory where your TL1d.exe file resides.
3. Enter one of the following commands:

   ```
   TL1d
   ```

   Or, to use a different gateway port, enter the following command:

   ```
   TL1d -p <your-gw-port number>
   ```

   As soon as you see “@(#)” followed by the actual port number, the daemon is running. You can minimize the command shell window.

Take a few steps to start the TL1 Daemon on a remote server on Solaris.

Follow these steps:

1. Launch a command shell window.
2. Navigate to the directory where your TL1d file resides.
3. Enter one of the following commands:

   ```
   ./TL1d
   ```

   Or, to use a different gateway port, enter the following command:

   ```
   ./TL1d -p <your-gw-port number>
   ```

   As soon as you see “@(#)” followed by the actual port number, the daemon is running. You can minimize the command shell window.

Terminating the TL1 Daemon

The specific command or procedure for gracefully terminating the TL1 daemon depends on the platform the daemon is running on.

- On Solaris, use either the kill command or Control-C.
- On NT, bring up the Task Manager, select the TL1d process, and then terminate it.

Index