

You are in: [Tivoli Network Manager](#) > Welcome to Tivoli Network Manager

## Welcome to Tivoli Network Manager



| Updated December 29, 2015 by [Sarochin T.](#) | Tags: *None*

# Welcome to Tivoli Network Manager

Tivoli Network Manager operations management software consolidates complex IT and network operation management tasks. The following content categories are available on this wiki:

[Overview](#)

[Best Practice Documents](#)

[Tips & Tricks](#)

[Feature Early Access](#)

[Troubleshooting](#)

[Requests For Enhancements](#)

[Product Documentation](#)

[Reference](#)

## Comments

*There are no comments.*

You are in: [Tivoli Network Manager](#) > [Welcome to Tivoli Network Manager](#) > Overview

## Overview



| Updated May 31, 2013 by [Keith Jeremy Posner](#) | Tags: *None*

The Network Manager Product Overview provides a clear and graphical introduction to the Network Manager product, as well as serving as an entry point into the Network Manager documentation set. The Product Overview is broken down into the following sections:

- Product and functional architectures
- Task overviews for each area of the product
- Extensive deployment information

Let's look at each of these sections in turn. And just before we start, note that you can download the entire Product Overview as a PDF document (98 pages, 1 MB) [here](#).

## Product and functional architectures

Easy-to-follow diagrams show the data flow for the entire product and for the different functional areas of the product. Real-world scenarios are used to show how data moves through the product. Key terminology is introduced. Deep dives into functionality use diagrams to communicate detailed concepts in a simple way.

For related Product Overview documentation, see the following topics:

- [Network Manager data flow](#)
- [Discovery architecture](#)

## Task overviews for each area of the product

Separate task overviews are provided for the different parts of the product, such as discovery, polling, network visualization, and event management. Administrator and operator roles are clearly distinguished, and links are provided into relevant sections of the documentation, where users can find detailed task-based information.

For related Product Overview documentation, see the following topics:

- [Discovery tasks](#)
- [Topology database administration tasks](#)
- [Network visualization tasks](#)

## Extensive deployment information

Typical deployment examples drive this section. Small, medium, large, and typical telecommunications provider deployments are matched to real-world customer networks to make it easy for you to identify the deployment that suits you. Deployment configurations show how Network Manager can be deployed within and across geographies. For each deployment examples, details are provided, including information on number of network domains, servers, processors, and client systems.

- [Typical deployments](#)
- [Details for telecommunications provider deployment](#)
- [Example of large deployment across geographies](#)

## Comments

*There are no comments.*

You are in: [Tivoli Network Manager](#) > [Welcome to Tivoli Network Manager](#) > Best Practice Documents

## Best Practice Documents



| Updated February 7, 2017 by [clarkrob](#) | Tags: [best-practices](#)

### ITNM Best Practices Guide for Deployment of Network Management for NOI

This guide approaches the deployment of the base products for NOI and Network Manager with an emphasis on preparation. The aim is to quickly get the system up and running with a minimum of fuss. Important concepts are explained with best practice suggestions so you can install it quickly the first time. The versions and maintenance levels are listed as of January 2017.

[Download](#)

---

### ITNM Best Practices Guide for Remote Discovery Set up

One of the requirements of a multi-tenancy environment is to be able to manage remote customer networks from a central location. And one of the challenges that come from such a deployment is the fact that the networks being discovered and polled are likely to be at the end of WAN links. This document describes how you can distribute Network Manager core processes so that discovery and monitoring can execute locally on the remote network. It compares this custom distributed architecture to the standard architectures.

[Download](#)

---

### ITNM 4.2 Calculator for system sizing and polled data storage

This spreadsheet is designed to help you with initial sizing plans for your IBM Tivoli Network Manager 4.2 system. It takes as input the major contributors to the system resources and will give you a good idea of the RAM, CPU, and disk space required for the ITNM core server, database, and GUI. It calculates your estimated polling load for scale and storage purposes, and the resources required.

[Download](#)

---

### ITNM Best Practices Session at InterConnect 2015

These slides cover a selection of tips and explanations to improve your outcomes working with Network Manager. Links are provided to the Knowledge Center, Best Practice documents, and Integrated Service Management (ISM) Library for details or downloads.

[Download](#)

---

### Network Manager Best Practice Guide for DB2

Read this document to understand how Network Manager uses the DB2 database with best practice tips on administration, maintenance, and tuning.

This paper is written for administrators responsible for maintaining Network Manager and assumes some basic familiarity with Network Manager and DB2.

[Download](#)

---

### Network Manager Best Practice Guide for Network Views

This guide is designed to help you efficiently create the network views for your users. It covers what constitutes a Network View and as an Administrator, shows you how to monitor, analyze and consolidate your network views to provide efficient, meaningful views that give the best possible topological visualization and monitoring to the NOC operators and clients.

[Download](#)

## Network Manager Best Practice Guide for Network Monitoring

Assuring the health of your network is one of the most important functions of Network Manager. This guide helps you take advantage of the full capabilities available in Network Manager to plan your polling policies and help you enrich the monitoring events for your operational needs.

It also helps you to configure the pollers and to use the new health metrics to ensure sustainable performance.

[Download](#)

---

## Network Manager Best Practice Guide for 3.9

Network Manager subject matter experts (SMEs) create and maintain a best practices guide that addresses the following topics.

- Discovery
- Polling
- Event Enrichment and Root-Cause Analysis
- General Performance Considerations

[Download](#)

---

## Network Manager Best Practice Event fields for enrichment and RCA

This guide explains the key event fields that affect the flow of the event through the enrichment and root cause analysis engine. These fields guide the built-in automatic procedures that map events correctly to the discovered topology, and position those events appropriately for root cause analysis and event suppression. It concludes with a simple OMNIBus automation that will analyze events in real time, populating an event field with a message so you can correct the fields in those events.

[Download](#)

## Comments

*There are no comments.*

You are in: [Tivoli Network Manager](#) > [Welcome to Tivoli Network Manager](#) > ISM Library has moved

## ISM Library has moved



| Updated June 29, 2017 by [Keith Jeremy Posner](#) | Tags: *None*

As detailed on the [ISM Library](#) page, "ISM Library will be sunset June 30, 2017. At that time reference IBM Product Specific Wikis for related integrations."

This post is to inform you that the files that were on the ISM library that are relevant to this product have been reviewed and migrated to a new location [here](#).

This site is currently the Dev Center for IBM Operations Analytics. Over time, the site will be enhanced and expanded to include further content from IBM Ops Management. The layout and navigation will be reviewed and changed to reflect this. We do not, however, plan to move the ISM Library files again. Thank you.

### Comments

*There are no comments.*

You are in: [Tivoli Network Manager](#) > [Welcome to Tivoli Network Manager](#) > Tips & Tricks

## Tips & Tricks



| Updated September 16, 2014 by [clarkrob](#) | Tags: *None*

See the following pages for many useful articles working with Network Manager:

[Installing and Configuring](#)

[Administering](#)

[Performance and Tuning](#)

[Integrating](#)

[Discovery provisioning](#)

## Comments

*There are no comments.*

You are in: [Tivoli Network Manager](#) > [Welcome to Tivoli Network Manager](#) > [Tips & Tricks](#) > Installing and Configuring

## Installing and Configuring



| Updated August 19, 2014 by [clarkrob](#) | Tags: *None*

This page includes installation and configuration information on Tivoli Network Manager best practices.

[Using multiple TIP servers for ITNM 3.9](#)

[Verification steps before performing Tivoli Network Manager Collector discovery against Alcatel 5529 Inventory Data Manager](#)

[ITNM 3.9 - ITNM 4.1 - ITNM 4.1.1 and TIP 2.2](#)

[ITNM 4.1 Example of Cognos content migration from default Derby database to DB2](#)

[Building multi-portlet pages in TIP 2.x for ITNM](#)

## Comments

*There are no comments.*



You are in: [Tivoli Network Manager](#) > [Welcome to Tivoli Network Manager](#) > [Tips & Tricks](#) > [Installing and Configuring](#) > Using multiple TIP servers for ITNM 3.9, 4.2

## Using multiple TIP servers for ITNM 3.9, 4.2



| Updated May 15, 2017 by [clarkrob](#) | Tags: [itnm](#), [tip](#)

This article applies to ITNM 3.9, 4.1, 4.1.1 using TIP. It also applies to ITNM 4.2 using DASH.

When the number of independent Tivoli Integrated Portal (TIP) servers is increased, IT Services is able to support larger numbers of concurrent users and clients by providing visibility of the state of the IT network. It also provides some measure of increased availability, because if one TIP server cannot be accessed, users can manually connect to another. Likewise, some customers maintain a form of load balancing by allocating users to TIP servers based on geography, function, and so on.

ITNM 3.9 does not support the TIP High Availability mode at this time, so this article is intended to help you understand how you can still use multiple TIP servers with ITNM.

Operators and clients who are using event views, network topology views and related tools, and reports need not be aware of being in a multi-TIP server deployment. Each user can create and modify topology views which, assuming the access permissions exist, can be seen by other users in the same manner as multiple users on the same TIP server.

Administrators must be aware of some caveats.

- When making changes to the customization or deployment of TIP, the administrator must be aware of the deployment and customization steps described in this article for multi-TIP deployments.
- With Discovery configuration, and to a lesser extent with Monitoring configuration, concurrency problems can occur, even on a single TIP server. This means if multiple provisioning administrators act independently across TIP servers, they must take care not to overwrite each other's changes.

### Deployment

Each TIP server can be installed from the Network Manager installer as per the standard distributed deployment documentation for a single TIP server. Each TIP server must connect to the same NCIM database and OMNibus object server. Note that DB2 HADR is supported for ITNM 3.9 FP4 and later, and Oracle RAC is supported for ITNM 4.1.1 and later.

Each TIP instance should use the same central Authentication service setup, whether it is the OMNibus object server or LDAP.

Manually customize each independent TIP server identically for context menu changes, tools, icons, and properties by using the files under `NCHOME/precision/profiles/TIPProfile/etc/tnm/`, and TCR for the reports.

### Post-installation configuration

There are some tasks that only one TIP server needs to perform. It is a best practice to designate one TIP server as the "administrative" server that will perform the tasks that only one TIP server needs to do. For instance only one server needs to update the network view caching. For best performance, this TIP server should be the closest to the NCIM database.

The following configuration steps should be applied only to the TIP servers indicated:

1. Select one server for "administrative" mode to update the network view caches.
2. On the designated "administrative" TIP server, set [the following property](#)

```
NCHOME/precision/profiles/TIPProfile/etc/tnm/topoviz.properties:  
topoviz.dynamicviewengine.enabled = true
```

3. On all the other non-administrative TIP servers, set the property to false

```
NCHOME/precision/profiles/TIPProfile/etc/tnm/topoviz.properties:  
topoviz.dynamicviewengine.enabled = false
```

4. Allow the "administrative" server to create the initial Auto-Provision views. On the non-administrative TIP servers, remove all the xml files in

```
NCHOME/precision/profiles/TIPProfile/etc/tnm/autoprovision/.
```

Tip: In practice, it is likely that the TIP server will be started during the installation and the default dynamic view container may be created for the itnadmin user. The easiest way to delete any duplicate network view containers is from the Network Views GUI. Further customization using the auto-provisioning mechanism should be done from the "administrative" server only. The dynamicview directory only needs to be customized on the "administrative" server for the *dynamicviewengine* to maintain.

5. Discovery Configuration stores the Pre-discovery Filter Library on the TIP server they are created on. The names of the active filters are stored on the TIP server. Both files are stored in

```
$NCHOME/precision/profiles/TIPProfile/etc/tnm/
```

```
discofilterlibrary.xml
```

```
discofilters.xml
```

(While these files control the GUI, the actual rules in the active filters are stored on the Network Manager core server in the file `$NCHOME/etc/precision/DiscoScope.<domain>.cfg`)

After making changes to the Prediscovery Filter Library, or the list of active filters, you must copy the two GUI files to the other TIP servers. Stop the TIP servers first.

### Network Manager Failover deployment

When you set up failover you have a choice with the database. Either can set up a single database for non-replicated mode, or two separate databases and let ITNM keep them in sync for replicated mode.

#### *Non-replicated database deployment*

In this scenario each independent TIP server connects to the same database, and to the primary ITNM server. You do not need to take special steps with the independent TIP servers in this scenario.

*Replicated database deployment (Not supported starting with 3.9 FP 4 - replaced with DB2 HADR and Oracle RAC)*

This scenario consists of deploying two databases that are kept in sync by the Network Manager. You deploy separate TIP servers for each of the primary and backup databases and, like the non-replicated database scenario, it involves no special steps for multiple TIPs. However, while it will work, it does not really maximize the use of all the hardware since the TIP servers connected to the secondary database are unused most of the time.

### Configuration with Netcool Configuration Manager

The GUI integration with Network Manager which involves adding views, context menus, and reports for NCM must be applied to each TIP server. However, the topology API integration has to be done only on the "administrative" TIP server.

### Performance considerations

Many factors are involved in determining how many concurrent users can be supported and it is beyond this article to quantify them. From experience the main points to consider are similar to a single TIP deployment. With Fix Pack 3, the GUI team overhauled the network topology view algorithms. For example, database communication, handling of many concurrent users, and view generation were reworked to optimize database traffic and locking mechanisms. These benefits can be seen in large single and multi-TIP deployments. Spreading the users over multiple TIP servers improves the concurrent processing of the views for each user. Since the bottleneck is with the view processing, a well-provisioned database can handle two or more TIP servers easily. As scale increases, more pressure is put on the single database, so it is important to maintain good database maintenance routines, especially if storing historical metrics from the network devices.

## Comments

*There are no comments.*

You are in: [Tivoli Network Manager](#) > [Welcome to Tivoli Network Manager](#) > [Tips & Tricks](#) > [Installing and Configuring](#) > Verification steps before performing Tivoli Network Manager Collector discovery against Alcatel 5529 Inventory Data Manager

## Verification steps before performing Tivoli Network Manager Collector discovery against Alcatel 5529 Inventory Data Manager



| Updated November 30, 2012 by [Putri Aziahanim](#) | Tags: *None*

This article will help you on the basic troubleshooting steps if you want to use Tivoli Network Manager Collector discovery against Alcatel Lucent 5529 Interface Data Manager.

This article assumes you have all 3 items below are installed and running in your environment.

Network Manager Collector : [Alcatel5529IdmSoap](#)

Target EMS : [Alcatel Lucent 5520 Access Management System](#)

NBI component on Target EMS : [Alcatel Lucent 5529 Inventory Data Manager](#)

### Alcatel5529IdmSoap Collector Introduction

Alcatel5529IdmSoap is a Perl based collector interfaces a NorthBound Interface Component called Alcatel Lucent 5529 Inventory Data Manager (IDM) in Alcatel Lucent 5520 Access Management System (AMS) EMS. The collector uses HTTP for

message negotiation and transmission with SOAP/XML as the messaging framework for exchanging request and response.

This collector downloads physical inventory data from Alcatel Lucent Managed Elements for Access Network mainly ranging from ISAM and GPON Network Element Types. It does not support layer 2 topology discovery.

Flow below depicts the high-level interaction between ITNM and the ALU5529 IDM

ITNM (Collector Discovery Agents) ↔ Alcatel5529IdmSoap Collector ↔ ALU5529 IDM

In this blog, the verification steps are divided into 2 parts.

ITNM (Collector Discovery Agents) ↔ Alcatel5529IdmSoap Collector ↔ ALU5529 IDM

└──────────────────────────┘  
Part 2

└──────────────────────────┘  
Part 1

Part 1 – Between Alcatel5529IdmSoap Collector and ALU5529 IDM

Part 2- Between Tivoli Network Manager (Collector Discovery Agents) ↔ Alcatel5529IdmSoap Collector

### Part 1 - Alcatel5529IdmSoap Collector ↔ Alcatel Lucent 5529 IDM

Part 1 covers the link between Tivoli Network Manager Collector and Target EMS, i.e. Alcatel5529IdmSoap Collector and Alcatel Lucent 5529 IDM. Part 1 determines successful handshaking and inventory data download from Alcatel Lucent 5529 IDM.

**Step 1)** Acquire the following information (1a) through (1d) from target EMS. Create user credential and its privilege **with IDM NBI** should the information is not present.

(1a) Host – Alcatel Lucent IDM Host address

(1b) Port – Alcatel Lucent IDM port listening for HTTP requests

(1c) Username - IDM username (**with IDM NBI permission**)

(1d) Password - IDM password in plaintext

**Step 2)** Do telnet to target IDM Host and IDM port using the following command

'telnet <IDM Host address> <Port>'

Example, at command prompt:

```
"telnet 1.2.3.4 8080"
```

telnet is successful and there's a service listening on the server's port.

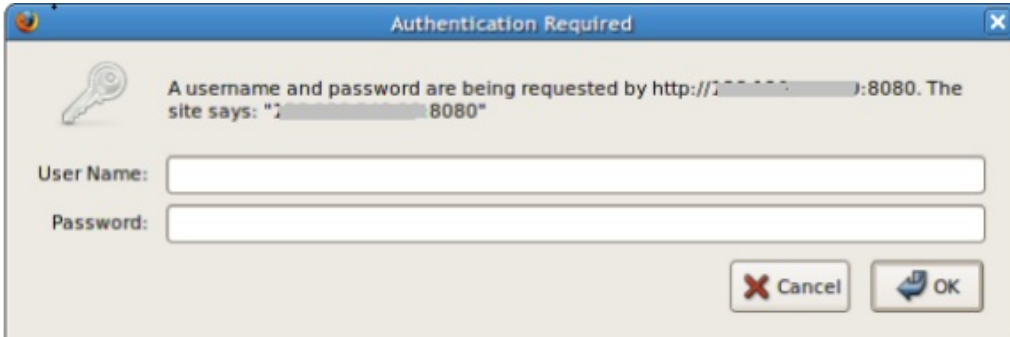
Proceed to Step 3 should Step 2 is successful.

**Step 3)** Attempt access to services on Alcatel Lucent 5529 IDM. ManagedElementMgr and InventoryRetrievalMgrExtns are the two Alcatel Lucent 5529 IDM services interested by the collector for discovery.

Launch web browser and enter the following URL. Substitute <IDM Host address> and <Port> with data acquired in Step 1 (1a) and (1b).

<http://<IDM Host address>:<Port>/idm/services/ManagedElementMgr>  
<http://<IDM Host address>:<Port>/idm/services/InventoryRetrievalMgrExtns>

Login prompt is displayed as follows:



Successful login directs user to this page:

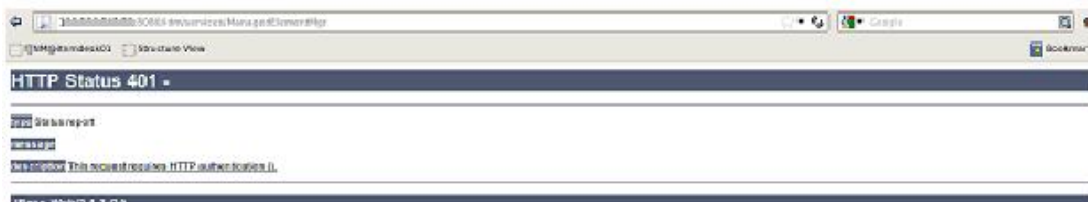
<https://<IDM Host address>:<Port>/idm/services/ManagedElementMgr>



<https://<IDM Host address>:<Port>/idm/services/InventoryRetrievalMgrExtns>



**Step 4)** Should login fails, you will get 'HTTP status – 401' message on your browser.



Remedies to login failure are :

- a) Verify correct credentials are given.
- b) Verify the user account is "IDM NBI" role access permission Via ALU 5520 AMS Client.

Implicitly it means we need to revisit Step 1.

**Step 5)** Before starting Alcatel5529IdmSoap collector, run 'ps -ef | grep Alcatel5529IdmSoap' at the command prompt. This action is to verify no Alcatel5529IdmSoap collector is running on the same box. Proceed to Step 6 should Step 5 returns

no result.

**Step 6)** Run the Alcatel5529IdmSoap collector and its successful message is as follows:

```
ncp_perl main.pl -cfg Alcatel5529IdmSoapCollector.cfg
Starting collector...
Alcatel5529IdmCollector LoadEmsDeviceList() @ 1352439114
Alcatel5529IdmCollector LoadEntityAndInterfaceList() @ 1352439119
Alcatel5529IdmCollector Completed push to data store() @ 1352439136
Listening for Precision calls...
```

You can also refer to IBM InfoCenter Document below for more details on how to configure Alcatel Lucent IDM (EMS) discovery:

[http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/index.jsp?topic=%252Fcom.ibm.networkmanagerip.doc\\_3.9%252Fitm%252Fip%252Fwip%252Fdisco%252Ftask%252Fnmip\\_dsc\\_configuringemsdsc.html](http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/index.jsp?topic=%252Fcom.ibm.networkmanagerip.doc_3.9%252Fitm%252Fip%252Fwip%252Fdisco%252Ftask%252Fnmip_dsc_configuringemsdsc.html)

## **Part 2 – Tivoli Network Manager (Collector Discovery Agents) ↔ Alcatel5529IdmSoap Collector**

Part 2 covers the link between Tivoli Network Manager Collector and Tivoli Network Manager Collector Discovery agents. A pre-requisite to Part 2 is Part 1 must yield successful results.

**Step 1 )** Verify the port mentioned in DiscoCollectorFinderSeeds.cfg is not being used by other application or user. Run 'netstat -a | grep <port>' should returns nothing.

**Step 2)** The port value in \$NCHOME/etc/precision/DiscoCollectorFinderSeeds.cfg must match with the value "Listen=<port number>" set in the Alcatel5529IdmSoap.cfg file. Listen property in Alcatel5529IdmSoap.cfg file is the Network Manager

Collector's listening port for XML-RPC requests.

**Step 3)** Enable CollectorDetails, CollectorInventory via \$NCHOME/ precision/etc/precision/DiscoAgents.cfg or Tivoli Network Manager GUI Full Discovery agents tab.

*Tips: Turn on debug level to 4 for CollectorDetails, CollectorInventory agents in \$NCHOME/ precision/etc/precision/DiscoAgents.cfg and*

*CollectorFinder (ncp\_df\_collector) in \$NCHOME/etc/precision/DiscoConfig.cfg.*

*CollectorFinder log is useful when the ITNM gui doesn't appear to be discovering any devices.*

*CollectorDetails, CollectorInventory agent logs are useful in data-processing troubleshooting.*

**Step 4)** Remove Ping Finder for Discovery Seeds in Tivoli Network Manager GUI. Ping Finder is not required for EMS based discovery because Network Elements are managed by EMS and typically they are located behind EMS. Therefore, in most cases, these Network Elements are not reachable via ping.

**Step 5)** Observe messages from the collector as shown below.

```
Starting collector...
Alcatel5529IdmCollector LoadEmsDeviceList() @ 1352439114
Alcatel5529IdmCollector LoadEntityAndInterfaceList() @ 1352439119
Alcatel5529IdmCollector Completed push to data store() @ 1352439136
Listening for Precision calls...
Reloading all device data from the IDM...
Alcatel5529IdmCollector LoadEmsDeviceList() @ 1352442871
Alcatel5529IdmCollector LoadEntityAndInterfaceList() @ 1352442876
Alcatel5529IdmCollector Completed push to data store() @ 1352442891
```

These messages describe that Tivoli Network Manager (Collector Discovery Agents) has successfully download data from the Alcatel5529IdmSoap Collector

**Step 6)** Should Discovery Phase 1 in Tivoli Network Manager GUI is completed, verify OQL data in CollectorDetails table.

Use either one of these methods to login to OQL :

(6a) Use 'ncp\_oql -service disco -username admin -domain <your domain>' command which export PATH=\$NCHOME/precision/bin:\$PATH  
Or

(6b) Use Management Database via Tivoli Network Manager GUI where service = Disco and domain = Your Domain.

At OQL prompt,

do "select \* from CollectorDetails.returns where m\_UniqueAddress = '<IP address for EMS Managed Element>';" where you must substitute <IP address for Alcatel Lucent 5520 EMS Managed Element> with IP Address for Alcatel Lucent 5520EMS Managed Element.

For example:

Assuming 1.2.3.4 is IP Address for Alcatel Lucent 5520 EMS Managed Element, your OQL command is "select \* from CollectorDetails.returns where m\_UniqueAddress = '1.2.3.4';".

Should the OQL command returns some values, you have ascertained Collector Discovery agent has successfully download data from Alcatel5529IdmCollector.

#### **Acronyms / Abbreviations :**

**ALU** – Alcatel Lucent

**EMS** – Element Management System

**IDM** – Inventory Data Manager

**NBI** – NorthBound Interface

**OQL** – Object Query Language

**SNMP** – Simple Network Management Protocol

## **Comments**

*There are no comments.*

You are in: [Tivoli Network Manager](#) > [Welcome to Tivoli Network Manager](#) > [Tips & Tricks](#) > Administering

## Administering



| Updated August 19, 2014 by [clarkrob](#) | Tags: *None*

This page includes administration information on Tivoli Network Manager best practices.

[Updating the ITNM API datasource after the ITNM database password is changed for NCM](#)

### Comments

*There are no comments.*



You are in: [Tivoli Network Manager](#) > [Welcome to Tivoli Network Manager](#) > [Tips & Tricks](#) > [Administering](#) > Updating the ITNM API datasource after the ITNM database password is changed.

## Updating the ITNM API datasource after the ITNM database password is changed.



| Updated June 19, 2013 by [EMAD\\_barry\\_mcdermott](#) | Tags: *None*

The ITNM API is used in an NCM-ITNM Integration configuration. The API connects to ITNM database via a datasource which needs to be refreshed if the ITNM database password changes.

### Updating the datasource password.

After updating the database password and the ITNM configuration run the following command to refresh the password used by the API

Unix

```
$NCHOME/precision/bin/createITNMDatasource.sh <tip_administrator_username> <tip_administrator_password>
```

Windows

```
$NCHOME/precision/bin/createITNMDatasource.bat <tip_administrator_username>
```

After running the command restart TIP.

The script above reads the new password value from `$NCHOME/precision/profiles/TIPProfile/etc/tnm/tnm.properties` and automatically updates the data source.

As a reference full Instructions for integrating NCM and ITNM can be found at the following link

[http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/index.jsp?](http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/index.jsp?topic=%2Fcom.ibm.netcool_configurationmgr.doc_6.3.0%2Fncm%2Fwip%2Fint%2Fconcept%2Fncm_int_integrating.html)

[topic=%2Fcom.ibm.netcool\\_configurationmgr.doc\\_6.3.0%2Fncm%2Fwip%2Fint%2Fconcept%2Fncm\\_int\\_integrating.html](http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/index.jsp?topic=%2Fcom.ibm.netcool_configurationmgr.doc_6.3.0%2Fncm%2Fwip%2Fint%2Fconcept%2Fncm_int_integrating.html)

### Testing the datasource after change

```
$TIPHOME/bin/wsadmin.sh -lang jython -username tipadmin -password <TIP ADMIN Password> -f
```

```
$NCHOME/precision/bin/createITNMDatasource.jy -test
```

The output should contain the following:

```
WASX7217I: Connection to provided datasource was successful.
```

## Comments

*There are no comments.*

You are in: [Tivoli Network Manager](#) > [Welcome to Tivoli Network Manager](#) > [Tips & Tricks](#) > [Administering](#) > Adding context menus to links (full disco link origin example)

## Adding context menus to links (full disco link origin example)

 2 | Updated April 8, 2016 by [lanFelstead](#) | Tags: [itnm](#), [network\\_manager](#)

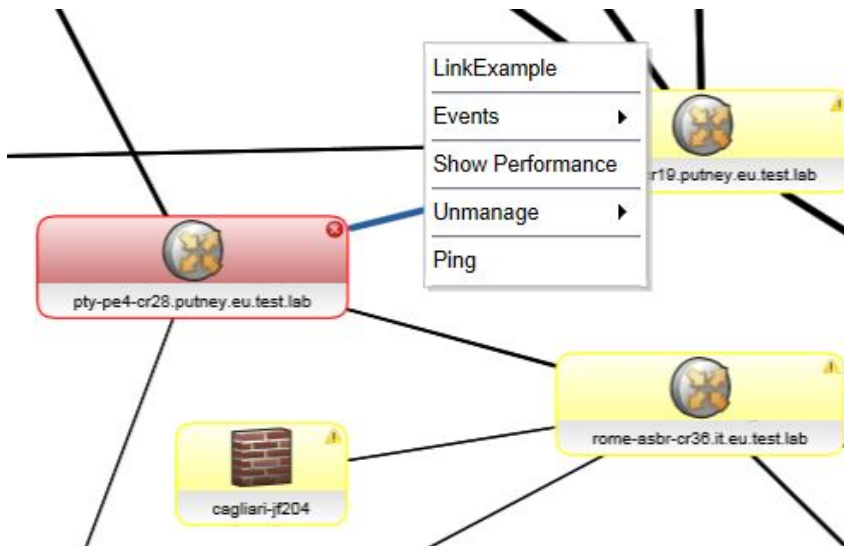
### Link Context Menus

One of the many configuration options of Network Manager is the ability to add context menus to topology links.

This article provides a small but fully working example implementation of such a link menu tool that results in a right click tool which provides a topology origin summary from ncp\_disco's current topology databases. Note: The tool makes use of a database table introduced in 4.1 and so will not work in earlier versions.

### Working Example: Link Discovery Origin Tool

By the end of this section you should have a tool (LinkExample) that appears as follows when a link is right clicked:



And when LinkExample is clicked;

Example Link Report  
2016-04-07T16:17:31

## Disco Topology Information (shows ALL links between the relevant devices)

**Topology Layer:** layer1  
No relevant connections found in this topology.

Contributing layer databases;  
No relevant connections found in any contributing link databases.

**Topology Layer:** fullTopology  
pty-pe4-cr28.putney.eu.test.lab[ Gi0/1 ] <-> pty-p2-cr19.putney.eu.test.lab[ Gi0/1/6 ]

Contributing layer databases;  
CDPLayer;  
pty-pe4-cr28.putney.eu.test.lab[ Gi0/1 ] <-> pty-p2-cr19.putney.eu.test.lab[ Gi0/1/6 ]  
pty-p2-cr19.putney.eu.test.lab[ Gi0/1/6 ] <-> pty-pe4-cr28.putney.eu.test.lab[ Gi0/1 ]

AdjustedIPLayer;  
pty-pe4-cr28.putney.eu.test.lab[ Gi0/1 ] <-> pty-p2-cr19.putney.eu.test.lab[ V165 ]  
pty-pe4-cr28.putney.eu.test.lab[ V123 ] <-> pty-p2-cr19.putney.eu.test.lab[ V123 ]  
pty-p2-cr19.putney.eu.test.lab[ V123 ] <-> pty-pe4-cr28.putney.eu.test.lab[ V123 ]  
pty-p2-cr19.putney.eu.test.lab[ V165 ] <-> pty-pe4-cr28.putney.eu.test.lab[ Gi0/1 ]

IPLayer;  
pty-pe4-cr28.putney.eu.test.lab[ V123 ] <-> pty-p2-cr19.putney.eu.test.lab[ V123 ]  
pty-pe4-cr28.putney.eu.test.lab[ Gi0/1 ] <-> pty-p2-cr19.putney.eu.test.lab[ V165 ]  
pty-p2-cr19.putney.eu.test.lab[ V165 ] <-> pty-pe4-cr28.putney.eu.test.lab[ Gi0/1 ]  
pty-p2-cr19.putney.eu.test.lab[ V123 ] <-> pty-pe4-cr28.putney.eu.test.lab[ V123 ]

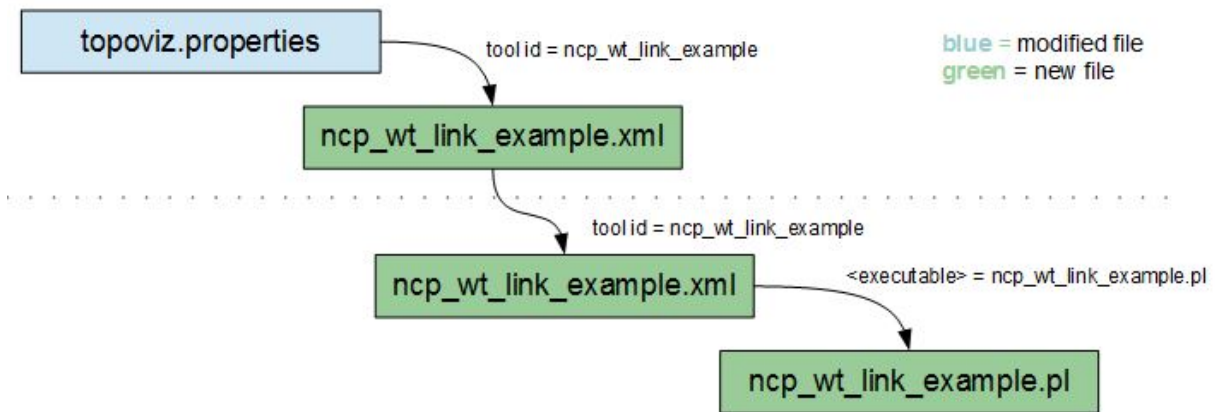
**Topology Layer:** convergedTopology  
pty-pe4-cr28.putney.eu.test.lab[ Gi0/1 ] <-> pty-p2-cr19.putney.eu.test.lab[ Gi0/1/6 ]  
pty-pe4-cr28.putney.eu.test.lab[ V123 ] <-> pty-p2-cr19.putney.eu.test.lab[ V123 ]

Contributing layer databases;  
fullTopology;  
pty-pe4-cr28.putney.eu.test.lab[ Gi0/1 ] <-> pty-p2-cr19.putney.eu.test.lab[ Gi0/1/6 ]

Tools such as this can be useful during connectivity issue investigation.

The code and configuration behind the above screenshots will now be presented. The tool itself has been kept deliberately small (150 lines) and so has minimal error handling and other limits to functionality (such as no directionality support).

The new/modified files presented in this article fit together as follows. The files above the dotted line are GUI side files, and those below are NM core side files.



### Add To Menu: topoviz.properties & ncp\_topoviz\_link\_menu.xml

The topoviz.properties file defines which menus are presented following a right-click on a link. By default it is configured with the following;

```
topoviz.menu.link=ncp_topoviz_link_menu
```

To add a menu item for a new link tool we need to alter ncp\_topoviz\_link\_menu.xml to reference the new tool's menu configuration file.

For this example we'll call the menu config file ncp\_wt\_link\_example.xml which means we need to add the following to ncp\_topoviz\_link\_menu.xml:

```
$NMGUI_HOME/profile/etc/tnm/menus/ncp_topoviz_link_menu.xml (4.2)
$NCHOME/precision/profiles/TIPProfile/etc/tnm/menus/ncp_topoviz_link_menu.xml (4.1/4.1.1)
```

```

...
<definition>
  <tool id="ncp_wt_link_example" />
  <separator />
...

```

## Define Menu Config File: ncp\_wt\_link\_example.xml

The tool menu configuration file, `$NMGUI_HOME/profile/etc/tnm/tools/ncp_wt_link_example.xml`, defines among other things the type of the tool, the arguments and the tool id.

In this example the tool will be a Perl script that is runs on the core server via `ncp_webtool`, which means the tool type is 'url', and we want the tool run once with the relevant entityIds passed as parameter 'target'. Define `ncp_wt_link_example.xml` as follows:

`$NMGUI_HOME/profile/etc/tnm/tools/ncp_wt_link_example.xml (4.2)`

`$NCHOME/precision/profiles/TIPProfile/etc/tnm/tools/ncp_wt_link_example.xml (4.1/4.1.1)`

```

<ncp_tool id="ncp_wt_link_example" label="LinkExample" key="ncp_wt_link_example"
  runOnList="true" runForEach="false" type="url">
  <url value="ncp_webtools/ToolService" target=" blank" method="GET">
    <parameter name="domain" valueType="domainName" />
    <parameter name="ncpWebtoolRun" valueType="text" text="ncp_webtool.cgi" />
    <parameter name="toolid" valueType="text" text="ncp_wt_link_example" />
    <parameter name="toolTitle" valueType="text" text="Example Link Tool" />
    <parameter name="servletDebug" valueType="text" text="false" />
    <parameter name="removeHttpHeader" valueType="text" text="true" />
    <parameter name="target" valueType="entityIdsList" />
  </url>
</ncp_tool>

```

WARNING: use `valueType="entityIdList"` rather than `valueType="entityIdsList"` in versions prior to 4.2.

The GUI side is now complete and we can move on to defining the tool on the core side.

## Webtool Tool Configuration: ncp\_wt\_link\_example.xml

A WebTool tool configuration file associates a core tool to the GUI menu. It also configures further options such as allowing us to request that the entityIds be passed as a concatenated string to the tool. The name of the WebTool tool configuration file can differ from the GUI tool menu configuration file; it is the tool id that needs to be consistent.

For this example the WebTool tool configuration file will be as follows;

`$NCHOME/precision/scripts/webtools/etc/ncp_wt_link_example.xml`

```

<?xml version='1.0' standalone='yes'?>
<ncp_webtools>
  <tool id="ncp_wt_link_example" type="executable" timeout="60">
    <template>
      <tags>
        <pagetitle>Network Manager Example Link Tool</pagetitle>
        <resulttitle>Example Link Report</resulttitle>
      </tags>
      <file>${NCWTHOME}/etc/templates/ncp_wt_border.tmp</file>
      <processor>${NCWTHOME}/etc/processors/notitle_exec.proc</processor>
    </template>
    <parameters>
      <parameter name="domain">
        <accept>NCP::Webtool::IPv4</accept>
        <accept>NCP::Webtool::Hostname</accept>
        <accept>NCP::Webtool::String</accept>
        <argument>-domain</argument>
        <default>NCOMS</default>
        <type>single</type>
      </parameter>
      <parameter name="target" mandatory="1">
        <drivingparameter>1</drivingparameter>
        <accept>NCP::Webtool::String</accept>
        <accept>NCP::Webtool::Int</accept>
        <argument>-target</argument>
        <type>list</type>
        <mode>concatenate</mode>
      </parameter>
    </parameters>
    <command>
      <executable>${NCWTHOME}/bin/ncp_wt_link_example.pl</executable>
    </command>
  </tool>
</ncp_webtools>

```



```

        foreach my $layerDb (@$layerDbs)
        {
            my $linkReport = "";
            ReportLinks( $sql, $layerDb, $deviceA, $deviceB, 3, "$layerDb;", \$linkReport );

            $shadLink = 1 if $linkReport ne "";
            $discoReport .= $linkReport;
        }
    }
    $discoReport .= $INDENT.$INDENT.$INDENT."No relevant connections found in any contributing link databases.<br>" if
$shadLink==0;
}
}
print $discoReport;
sub ReportLinks
{
    my $sql = shift;
    my $layerDb = shift;
    my $deviceA = shift;
    my $deviceB = shift;
    my $indentSize = shift;
    my $header = shift;
    my $report = shift;

    my $linkQuery = "SELECT m_Name, m_NbrName FROM $layerDb.entityByNeighbor ".
        "WHERE (".
            "m_Name IN ((SELECT m_Name FROM workingEntities.interfaceMapping WHERE m_BaseName = '$deviceA'))".
            " AND ".
            "m_NbrName IN ((SELECT m_Name FROM workingEntities.interfaceMapping WHERE m_BaseName = '$deviceB'))".
        ") OR (".
            "m_Name IN ((SELECT m_Name FROM workingEntities.interfaceMapping WHERE m_BaseName = '$deviceB'))".
            " AND ".
            "m_NbrName IN ((SELECT m_Name FROM workingEntities.interfaceMapping WHERE m_BaseName = '$deviceA'))".
        ");";

    $sql->Send( $linkQuery, 2 );

    my ($tag, $data) = $g_App->GetResult();

    if( $data && ref($data) eq 'ARRAY' && @$data[0] )
    {
        foreach my $record (@{$data})
        {
            $$report .= ($INDENT x $indentSize).$header."<br>" if $$report eq "" and $header ne "";
            $$report .= ($INDENT x ($indentSize+1)).$record->{m_Name}." <-> ".$record->{m_NbrName}."<br>";
        }
    }
}
sub ResolveMainNodes
{
    my $targetString = shift;
    $targetString =~ s/^\s+//;

    my @entityIds = split(/ /, $targetString);
    my %ncim_access;

    $ncim_access{domain}= $g_Param->Domain();
    $ncim_access{dbid}= "NCIM";
    NCP::DBI_Factory::setLogLevel("warn");
    $ncim_access{dbh}= NCP::DBI_Factory::createDbHandle(%ncim_access);
    exit 1 unless $ncim_access{dbh};

    my $mainNodeQuery =
        "SELECT DISTINCT em.entityName AS mainNodeEntityName ".
        "FROM entityData e ".
        "INNER JOIN entityData em on em.entityId = e.mainNodeEntityId ".
        "WHERE e.entityId in (?".( "?" x scalar @entityIds-1 ).")";

    my @mainNodes;
    my $sth;
    if ($sth = $ncim_access{dbh}->prepare($mainNodeQuery) and $sth->execute( @entityIds) )
    {
        while ( my $row = $sth->fetchrow_hashref('NAME_uc') )
        {
            push @mainNodes, $row->{MAINNODEENTITYNAME};
        }
    }

    $sth->finish() if $sth;
    $ncim_access{dbh}->rollback(); # avoids DB2 warning. Could use finish() in 4.2
    $ncim_access{dbh}->disconnect(); # could use finish() in 4.2
    return ($mainNodes[0], $mainNodes[1]);
}
}
1;

```

Note: You may have to chmod 755 this file to make it executable.

The tool should now run when selected. Note: The tool only gives useful data when ncp\_disco is up, and can only provide data on ncp\_disco's current discovery.

## Comments



Krishna M Kodali commented on April 8, 2016 [Permalink](#)

Very cool post - very helpful for ITNM & Network Admins !

---

Show 10 | [25](#) | [50](#) items per page

[Previous](#) | [Next](#)

You are in: [Tivoli Network Manager](#) > [Welcome to Tivoli Network Manager](#) > [Tips & Tricks](#) > Performance and Tuning

## Performance and Tuning



| Updated August 19, 2014 by [clarkrob](#) | Tags: *None*

This page includes performance and tuning information on Tivoli Network Manager best practices.

[Oracle 11g Tuning Tips for ITNM](#)


### Comments

*There are no comments.*



You are in: [Tivoli Network Manager](#) > [Welcome to Tivoli Network Manager](#) > [Tips & Tricks](#) > [Performance and Tuning](#) > Oracle 11g Tuning Tips for ITNM

## Oracle 11g Tuning Tips for ITNM

 | Updated July 2, 2013 by [clarkrob](#) | Tags: [itnm](#), [oracle](#), [tuning](#)

This article is intended for ITNM users who find themselves with an Oracle database, perhaps managed by another group, and are starting to question the database performance. There are a few places where ITNM has a heavy transaction load on the database where scalability and performance effects can be magnified if the database is slow, especially at times when they overlap. These are,

- Immediately after discovery, as the newly discovered data is merged into the NCIM topology database
- Opening the Network View tree and individual large views
- Poller at startup
- Poller after discovery
- Poller, if storing and pruning data
- ITM agent transferring data to TDW
- Reports

Performance is very difficult to quantify since there are so many variables - machine specifications and available resources, size of discovered topology (number of entities in total and also within very large network devices), number of concurrent active GUI users, number and size of network views, polling density and storage rate. ITNM has many ways to scale to handle increasingly large networks, but this is not the focus of this article.

I'll describe in this article how to determine the start and end snapshots so we can use Oracle's Automatic Workload Repository (AWR) reports to check and resolve two main causes of database slowness - available memory and redo log size. In addition, we can run the Active Database Diagnostic Monitor (ADDM) report with the same snapshot period to identify the top "findings" affecting performance and recommended actions to resolve each issue.

Before we get into the reports, I want to mention that it is very important to establish a daily maintenance activity to update statistics on all tables and indexes across especially the busy schemas, NCIM, NCPGUI, NCPOLLDATA, but might as well include all the ITNM schemas. Choose a quiet period relative to the busy periods above for the database. This will allow the SQL query optimizer to work more effectively.

### Identify the report period

Oracle 11g automatically stores snapshot data. So to start, all you need to do is identify the time interval where ITNM is experiencing performance problems - use the list of heavy hitters as a guide here. Then find the start and end snapshot ID for this time interval from the results of the following query,

```
SQL> SELECT snap_id, begin_interval_time, end_interval_time
FROM dba_hist_snapshot
ORDER BY 1;
```

```

SNAP_ID END_INTERVAL_TIME
-----
1107 03-OCT-04 01.24.04.449 AM
1108 03-OCT-04 02.00.54.717 AM
1109 03-OCT-04 03.00.23.138 AM
1110 03-OCT-04 10.58.40.235 PM
```

### Automatic Workload Repository (AWR) Report

Generate the AWR report by running,

```
SQL> $ORACLE_HOME/rdbms/admin/awrrpt.sql
```

and entering the start and end snapshot IDs when prompted.

### Memory

The two items to consider are SGA and PGA. Together they indicate the amount of memory assigned to this database instance.

The SGA (System Global Area) is an area of memory allocated when an Oracle Instance starts up. The PGA (Program or Process Global Area) is a memory area that stores data and control information for a single process. Oracle 11g allows users to tune both PGA and SGA areas with a single parameter, called MEMORY\_TARGET.

This table in the AWR report shows the usage during the report period:

### SGA Target Advisory

SGA Target Size (M)	SGA Size Factor	Est DB Time (s)	Est Physical Reads
904	0.50	908,147	345,273,626
1,356	0.75	873,772	332,007,526
1,808	1.00	37,874	9,417,264
2,260	1.25	34,939	8,229,747
2,712	1.50	34,825	8,229,747
3,164	1.75	34,821	7,539,462
3,616	2.00	34,821	7,539,462

The SGA Target Advisory table is very useful since it projects the estimated improvement if you increase the SGA by some factor. The Size Factor row of 1.00 is the current size and the first table below indicates the benefits when SGA is increased by various factors. Based on this table, you might increase the sga\_target to 3.1GB based on this table. Avoid over-allocating memory there since it can cause swapping and be count-productive. Make sure it is less than physical memory available.

### SGA Target Advisory

SGA Target Size (M)	SGA Size Factor	Est DB Time (s)	Est Physical Reads
904	0.50	908,147	345,273,626
1,356	0.75	873,772	332,007,526
1,808	1.00	37,874	9,417,264
2,260	1.25	34,939	8,229,747
2,712	1.50	34,825	8,229,747
3,164	1.75	34,821	7,539,462
3,616	2.00	34,821	7,539,462

Similarly with PGA. The PGA Memory Advisory table shows, based on activity during this report period, that when the pga\_target was initially set to 300MB, that there is not much to gain by assigning more than 420MB:

### PGA Memory Advisory

- When using Auto Memory Mgmt, minimally choose a pga\_aggregate\_target value where Estd PGA Overallloc Count is 0

PGA Target Est (MB)	Size Factr	W/A MB Processed	Estd Extra W/A MB Read/ Written to Disk	Estd PGA Cache Hit %	Estd PGA Overallloc Count	Estd Time
38	0.13	9,260.50	5,551.70	63.00	189	3,034,779
75	0.25	9,260.50	5,551.70	63.00	189	3,034,779
150	0.50	9,260.50	5,551.70	63.00	189	3,034,779
225	0.75	9,260.50	4,878.82	65.00	155	2,896,917
300	1.00	9,260.50	469.57	95.00	0	1,993,532
360	1.20	9,260.50	61.82	99.00	0	1,909,993
420	1.40	9,260.50	0.00	100.00	0	1,897,326
480	1.60	9,260.50	0.00	100.00	0	1,897,326
540	1.80	9,260.50	0.00	100.00	0	1,897,326
600	2.00	9,260.50	0.00	100.00	0	1,897,326
900	3.00	9,260.50	0.00	100.00	0	1,897,326
1,200	4.00	9,260.50	0.00	100.00	0	1,897,326
1,800	6.00	9,260.50	0.00	100.00	0	1,897,326
2,400	8.00	9,260.50	0.00	100.00	0	1,897,326

So based on this, you might set the MEMORY\_TARGET to 3600MB (3100MB + 420MB).

### Redo Logs (transaction logs)

If ITNM is storing a lot of polled data, then it can generate a lot of redo activity. The ADDM report called out the log file sync wait times (see the ADDM examples further down) and you can see that the Instance Activity Stats table is showing the log file switches at 19.40 per hour. The log file switches occur when the log writer has filled up one log and switches over to another log, unless the DBA has configured it to switch at fixed intervals. In any case, the number of switches should be no more than 4 - 6 per hour ideally to conserve overhead. Increase the allocated log size to reduce the rates. The size here was raised from 100M to 2 GB, probably the maximum amount you would want to consider.

Before the redo log size was changed we see the log file switches at 19.40 per hour and log file sync waits over 2 million.

### Instance Activity Stats - Thread Activity

- Statistics identified by '(derived)' come from sources other than SYSSTAT

Statistic	Total	per Hour
log switches (derived)	114	19.40

### Foreground Wait Events

- s - second, ms - millisecond - 1000th of a second
- Only events with Total Wait Time (s) >= .001 are shown
- ordered by wait time desc, waits desc (idle events last)
- %Timeouts: value of 0 indicates value was < .5%. Value of null is truly 0

Event	Waits	%Time -outs	Total Wait Time (s)	Avg wait (ms)	Waits /txn	% DB time
log file sync	2,292,756	0	4,436	2	1.00	20.42
db file sequential read	1,092,026	0	2,378	2	0.47	10.95
db file scattered read	545,463	0	1,158	2	0.24	5.33
SQL*Net break/reset to client	2,241,244	0	618	0	0.07	2.85

After increasing the log size, you can see the log file switches dropped to under 4 per hour, now in the recommended range. The log file sync Waits decreased substantially.

## Instance Activity Stats - Thread Activity

- Statistics identified by '(derived)' come from sources other than SYSSTAT

Statistic	Total	per Hour
log switches (derived)	7	3.49

## Foreground Wait Events

- s - second, ms - millisecond - 1000th of a second
- Only events with Total Wait Time (s) >= .001 are shown
- ordered by wait time desc, waits desc (idle events last)
- %Timeouts: value of 0 indicates value was < .5%. Value of null is truly 0

Event	Waits	%Time-outs	Total Wait Time (s)	Avg wait (ms)	Waits /txn	% DB time
log file sync	190,472	0	522	3	1.55	5.38
db file sequential read	139,927	0	412	3	1.14	4.25
enq: TX - row lock contention	24	0	217	9060	0.00	2.24
buffer busv waits	11.860	0	206	17	0.10	2.13

## Active Database Diagnostic Monitor (ADDM) Report

Obviously memory and redo logs are not the only suspects for poor database performance. The ADDM report is a great place you can easily check for other big problems. Run the following SQL script to generate this report using the same start and end snapshot IDs you used with the AWR report.

```
SQL> $ORACLE_HOME/rdbms/admin/awaddm.sql
```

The ADDM report lists the top Findings, each with a number of recommendations including actions and supporting rationales.

The top Finding on all our reports was this one for Virtual Memory Paging. It turned out to be an Oracle bug, but if you see this you should also check that other database instances on the same server are not starving your instance, just in case it is real. For instance, on Linux run vmstat and monitor the si and so (swap in, swap out) counts – they should be close to zero for your database instance.

### Finding 1: Virtual Memory Paging

Impact is 1.03 active sessions, 100% of total activity.

Significant virtual memory paging was detected on the host operating system.

#### Recommendation 1: Host Configuration

Estimated benefit is 1.03 active sessions, 100% of total activity.

#### Action

Host operating system was experiencing significant paging but no particular root cause could be detected. Investigate processes that do not belong to this instance running on the host that are consuming significant amount of virtual memory. Also consider adding more physical memory to the host.

Here is a Finding that highlighted the inadequate memory...

### Finding 3: Undersized SGA

Impact is .08 active sessions, 19.82% of total activity.

The SGA was inadequately sized, causing additional I/O or hard parses.

The value of parameter "sga\_target" was "1808 M" during the analysis period.

#### Recommendation 1: Database Configuration

Estimated benefit is .08 active sessions, 18.71% of total activity.

#### Action

Increase the size of the SGA by setting the parameter "sga\_target" to 2712 M.

The Finding on the Redo logs was reduced from up to 43% of total activity to 5.38% after increasing the redo log allocation to 2GB. The Finding itself mentions the I/O performance and Oracle recommends moving the redo logs to a separate disk to avoid contention.

### Finding 3: Commits and Rollbacks

Impact is .07 active sessions, 5.38% of total activity.

Waits on event "log file sync" while performing COMMIT and ROLLBACK operations were consuming significant database time.

#### Recommendation 1: Host Configuration

Estimated benefit is .07 active sessions, 5.38% of total activity.

#### Action

Investigate the possibility of improving the performance of I/O to the

*online redo log files.*

**Rationale**

*The average size of writes to the online redo log files was 25 K and the average time per write was 1 milliseconds.*

**Rationale**

*The total I/O throughput on redo log files was 1.3 M per second for reads and 1.4 M per second for writes.*

**Rationale**

*The redo log I/O throughput was divided as follows: 0% by RMAN and recovery, 52% by Log Writer, 0% by Archiver, 0% by Streams AQ and 47% by all other activity.*

**Symptoms That Led to the Finding:**

-----  
*Wait class "Commit" was consuming significant database time.  
Impact is .07 active sessions, 5.38% of total activity.*

I hope this provided some easy steps you can take to improve the database performance. Often you have to work within corporate guidelines for your databases when managed centrally and being able to ask for these reports lets you see if the database might be impacting ITNM performance.

## Comments

*There are no comments.*

You are in: [Tivoli Network Manager](#) > [Welcome to Tivoli Network Manager](#) > [Tips & Tricks](#) > [Integrating](#) > Direct Launch of Structure View

## Direct Launch of Structure View



| Updated April 15, 2015 by [PatrickHourigan](#) | Tags: *None*

### Pre-amble

This article should be considered as an ammendment to the set of direct URL's available in the Topoviz web application as documented [here](#).

When integrating other products with ITNM, you may want to launch the ITNM Structure View of a given Device or Interface, for example if you intend on using the ITNM DLA (Discovery Library Adapter)

### Structure Browser URL Reference

Use this information to understand how to construct a URL to display the structure and containment of a given device or an interface.

Base URL -> [https://host:port/ibm/console/ncp\\_structureview/Launch.do?](https://host:port/ibm/console/ncp_structureview/Launch.do?)

Parameters

Parameter	Explanation
domain	The domain to which this entity belongs
entityId	The Id of the entity to consider as the root of the structure view tree
entityName	The Name of the entity to consider as the root of the structure view tree. If you intend on using this parameter, you must also supply the domain
selectEntityId	The Id of the selected entity in the structure view tree. This may be an Interface selected on the overall Device containment
selectEntityName	The Name of the selected entity in the structure view tree. This may be an Interface selected on the overall Device containment

Applying these parameters we can direct launch the Structure View based on either the id or the name of the Device or Interface for a specific domain.

This facility also allows us to dive into a Device containment and select a contained entity, such as a VLAN object and/or an Interface by utilising the selectedEntityId or selectedEntityName parameters as demonstrated in Example 3, 4 and 5 provided.

Example 1: Structure View of Device by Id

[https://host:port/ibm/console/ncp\\_structureview/Launch.do?entityId=9](https://host:port/ibm/console/ncp_structureview/Launch.do?entityId=9)

[DeviceByEntityId.PNG](#)

Example 2: Structure View of Device by Name and Domain

[https://host:port/ibm/console/ncp\\_structureview/Launch.do?entityName=ny-p1-cr28.na.test.lab&domain=AUTO](https://host:port/ibm/console/ncp_structureview/Launch.do?entityName=ny-p1-cr28.na.test.lab&domain=AUTO)

Note: The domain must be supplied when launching by entityName.

[DeviceByEntityNameAndDomain.PNG](#)

Example 3: Structure View of a VLAN interface within a Device by Name and Domain (View from the Interface within the device structure)

[https://host:port/ibm/console/ncp\\_structureview/Launch.do?entityName=ny-p1-cr28.na.test.lab&domain=AUTO&selectEntityName=ny-p1-cr28.na.test.lab\[VI85\]](https://host:port/ibm/console/ncp_structureview/Launch.do?entityName=ny-p1-cr28.na.test.lab&domain=AUTO&selectEntityName=ny-p1-cr28.na.test.lab[VI85])

Note: The domain must be supplied when launching by entityName.

[ContainedInterfaceByEntityNameAndDomain.PNG](#)

Example 4: Structure View of a VLAN interface by Name and Domain (View from the Interface structure only)

[https://host:port/ibm/console/ncp\\_structureview/Launch.do?entityName=ny-p1-cr28.na.test.lab\[VI85\]&domain=AUTO&selectEntityName=ny-p1-cr28.na.test.lab\[VI85\]](https://host:port/ibm/console/ncp_structureview/Launch.do?entityName=ny-p1-cr28.na.test.lab[VI85]&domain=AUTO&selectEntityName=ny-p1-cr28.na.test.lab[VI85])

Note: The domain must be supplied when launching by entityName.

[InterfaceByEntityNameAndDomain.PNG](#)

Example 5: Structure View of a VLAN interface within a Device by Id

[https://host:port/ibm/console/ncp\\_structureview/Launch.do?entityId=9&selectEntityId=1474](https://host:port/ibm/console/ncp_structureview/Launch.do?entityId=9&selectEntityId=1474)

## Comments

*There are no comments.*

You are in: [Tivoli Network Manager](#) > [Welcome to Tivoli Network Manager](#) > [Tips & Tricks](#) > Discovery provisioning

## Discovery provisioning

 1 | Updated September 17, 2014 by [clarkrob](#) | Tags: *None*

This article explains the concepts that will help you provision Network Manager as your network changes on a day to day basis. Decide whether you will determine when a device is obsolete or not your responsibility, or if you should let the discovery handle removing devices given some assumptions.

The IP address and hostname play a significant part of determining identity within Network Manager. The discovery will do its best to adjust the topology as it finds duplicate IP addresses and hostnames. What can you do to facilitate this and thus ensure the devices continue to be monitored correctly.

### Removing obsolete devices with Discovery

When ITNM can no longer ping a device during discovery, what should the discovery do?

1. Manually remove.
  - Discovery should never delete anything - I will decide when a device is no longer under management.
  - Set `lingerTime` to 1000000, and use `RemoveNode.pl`. If they have not been physically removed from the network, obviously, you must adjust the scope and seeds in disco config.
  - 3.8/3.9 - `RemoveNode.pl` sets the device to unmanaged and `lingerTime` to 0 so that the next discovery removes the device and it is no longer polled .
  - From 4.1 - `RemoveNode.pl` removes the device immediately.
2. Discovery should automatically remove non-responding devices.
  - Discovery maintains a topology representing the as-is snapshot state of the network. Only active devices are discovered and maintained in the topology. This is typically not desirable since devices that are temporarily down will be removed from the topology and not polled until after a future, successful, discovery.
  - Set `lingerTime` to 0.
3. Configure the discovery to assume it is obsolete after failing for a certain number of consecutive discoveries.
  - Set `lingerTime` to 3 (default) so that it will automatically remove it after failing to respond for 4 consecutive discoveries. (3 discoveries to reduce the count to 0, and the next discovery will actually remove it if it is still down, so 4 in total).
  - Use the report, "Devices pending delete on next discovery", to list all the devices that will be removed on the next discovery if they don't respond, to check that the assumption is correct.

It is important to know that while a device is down and lingering out, it will still be polled and will continue to generate the appropriate events (ping fail, SNMP poll fail, etc.). This is generally desirable so that when the device is repaired and on line again, the failed events are cleared automatically as soon as possible.

If you know the devices that are being removed from the network, then you can remove them from ITNM with `RemoveNode.pl` so they will stop being polled and generating events that are not actionable. This an advantage of the first method above. (Note that with 3.8/3.9 `RemoveNode.pl` will automatically unmanage the device to prevent polling, whereas from 4.1, `RemoveNode.pl` actually removes the device from the database immediately without waiting for the next discovery.)

### LingerTime

The `lingerTime` count is used to determine how many consecutive discoveries the device has not responded to a ping. Each discovery will decrement the `lingerTime` count if the device does not respond. When it does respond, the discovery always sets the `lingerTime` back to the default value.

You can change the default value for the `lingerTime` variable by editing the file, `$NCHOME/etc/precision/ModelSchema.<domain>.cfg`. Find the `insert` section for the table `model.config`, and change the `LingerTime` variable to the desired value:

```
insert into model.config
(
  LingerTime,
  ChassisCreationEvents,
  IpInterfaceCreationEvents,
  MaintenanceStateEvents,
  ManagedStatusUpdateInterval,
  BuildNeighborAndContainers,
  DeleteRenamedDevices
)
values
(
  3, // LingerTime (3 discoveries)
  0, // Raise creation/deletion events for chassis (0 = no; 1 = yes)
  0, // Raise creation/deletion events for interfaces with IP addresses
    // (0 = no; 1 = yes)
  0, // Raise maintenance state change events for chassis and for
    // interfaces with IP addresses (0 = no; 1 = yes)
  30, // check managedStatus table for changes every 30 seconds
  0, // Should model build master.entityByNeighbor and master.containers
  0 // if set to 1, a renamed devices are deleted, by default, linger time is decremented
);
```

After restarting the Network Manager processes, new devices will use the new default value. To change the current values for existing devices, refer to the Knowledge Center:

For 4.1/4.1.1

[http://www.ibm.com/support/knowledgecenter/SSSHRK\\_4.1.1/com.ibm.networkmanagerip.doc\\_4.1.1/itnm/ip/wip/disco/task/nmip\\_dsc\\_mng\\_redsc\\_force\\_redsc\\_set\\_linger.html?cp=SSSHRK\\_4.1.1](http://www.ibm.com/support/knowledgecenter/SSSHRK_4.1.1/com.ibm.networkmanagerip.doc_4.1.1/itnm/ip/wip/disco/task/nmip_dsc_mng_redsc_force_redsc_set_linger.html?cp=SSSHRK_4.1.1)

For 3.8/3.9

[http://www.ibm.com/support/knowledgecenter/SSSHRK\\_3.9.0/com.ibm.networkmanagerip.doc\\_3.9/itnm/ip/wip/disco/task/nmip\\_dsc\\_mng\\_redsc\\_force\\_redsc\\_set\\_linger.html](http://www.ibm.com/support/knowledgecenter/SSSHRK_3.9.0/com.ibm.networkmanagerip.doc_3.9/itnm/ip/wip/disco/task/nmip_dsc_mng_redsc_force_redsc_set_linger.html)

## Provisioning changes

The discovery will attempt to maintain an accurate topology in the database as the network grows, and IP addresses and hostnames change. The discovery will do its best to make decisions when it encounters duplicate IP addresses and hostnames resulting from device and addressing changes.

If you are re-IP'ing sections of the network, or changing some IP addresses on devices, ITNM will handle this automatically. However, it can take a few discoveries to linger out obsolete devices in the topology, depending on the `lingerTime` value, and this can cause the poller to issue false alerts for the obsolete devices. This section suggests steps to take to minimize this transition and help the discovery engine to maintain an accurate topology quickly.

Various changes have been made, starting in Network Manager 3.8 up to the present 4.1.1, that improve the logic behind handling duplicates, so the steps vary a little depending on the version you are using.

### Is the unwanted IP address in ITNM still active on the network?

If so, prevent rediscovery by adjusting the discovery configuration,

- (i) Remove from Scope, or add exclude entries
- (ii) Remove from seedlists and ping finder

### Will the discovery find the new device?

Ensure it is found by using the Discovery Configuration GUI to,

- (i) Set the Scope
- (ii) Add the IP address to a seed file or to the ping finder

### Restart the disco process

When you start a discovery you have the option of stopping the disco processes first, and then restarting them to trigger the discovery. You can do this from the Discovery Status GUI by pressing the Stop button, followed by the Start button.

This refreshes the in-memory cached data and will cleanly discover the device interfaces. This also ensures IP changes for devices that do not respond to SNMP are handled efficiently in a single discovery in all circumstances.

### Is the device being renamed, or DNS name change for IP address

A new entity is created for the new device name, and the old device entity will either be removed, or linger out.

For 3.8 FP6 and forward, you can set the option `DeleteRenamedDevices` to 1 so that the device is renamed and no duplicate entity is created. This can be set in the same `model.config` table above in section "LingerTime". For more details, see <http://www-01.ibm.com/support/docview.wss?uid=swg21592568> (Note that the `DeleteRenamedDevices` setting will have no effect if `sysName` is used for device naming.)

### Are devices that don't respond to SNMP different?

Yes. If SNMP fails, it is not possible to determine the set of interfaces on a device, so typically a separate device is created to model each IP address that fails SNMP. Although IP addresses that resolve to the same name are assumed to belong to the same device in the absence of SNMP data.

When the `noSNMPAccess` device responds to SNMP on the next discovery, it will create the device correctly, but the duplicate obsolete devices may linger in the topology.

For 3.8 FP6 and forward, set the option `DeleteRenamedDevices` to 1, as described above, to handle the transition for `noSNMPAccess` devices to fully SNMP responding devices.

### Should the old devices be removed from the database first?

The discovery will do its best to make decisions when it finds duplicate IP addresses. However, there are several complicating factors that can affect this outcome, for example,

1. If the old IP addresses are still active on the network (respond to ping)
2. When moving an IP address from one device in the ITNM topology to another
3. If you use `sysName` for naming devices
4. A partial discovery is run instead of a full discovery. (The `lingerTime` algorithm is only activated on a full discovery or rediscovery)

To ensure a more predictable and efficient discovery result when changing IP addresses, it is a best practice to remove the devices in ITNM with the old IP addresses before attempting to discover the changes.

For 3.8 and 3.9:

- a) Use `RemoveNode.pl` which will,
  - Unmanage the device and its interfaces to stop polling
  - Set `lingerTime` to 0 so the next full discovery (or rediscovery) will actually remove it
- b) Complete a full discovery
- c) Add the new provisioning changes to the discovery configuration and complete a second full or partial discovery

For 4.1:

- a) Use `RemoveNode.pl` to completely remove the device (no need for a full discovery)
- b) Add the new provisioning changes to the discovery configuration and complete a full or partial discovery

### What is the affect on the poller?

The poller is updated automatically when new devices are added to the topology or devices are removed from the topology. However, the story is different for changes in the IP interfaces on a device.

Starting with Network Manager 3.9 FP4, the poller automatically adjusts its scheduling with any interface IP address changes at the end of the discovery. Sometimes if an obsolete device is lingering out, the poller will issue ping fails and attribute them to the obsolete device while it remains in the database in the managed state.

Prior to 3.9 FP4, the policy must be manually re-enabled (or poller restarted) to pick up new or changed IP interface addresses.

### Should I use `AddNode.pl` for provisioning new devices in ITNM?



`$NCHOME/precision/bin/AddNode.pl` is a convenience script that simply starts a partial discovery for a set of IP addresses.

Whether you start a partial discovery via `AddNode.pl` or a full or partial discovery via the Discovery Status GUI, you must first ensure the discovery Scope includes the new IP addresses. Depending on your practice for discovery, you may also want to add them to either a seed file, or to the ping finder in the Discovery Configuration GUI to ensure they are always part of future discoveries.

## Comments

---

1-1 of 1

[Previous](#) | [Next](#)



[Karl\\_Gerriets](#) commented on January 8, 2015 [Permalink](#)  
..well done, Robert, perfect documentation .. :-)

---

Show 10 | [25](#) | [50](#) items per page

[Previous](#) | [Next](#)

You are in: [Tivoli Network Manager](#) > [Welcome to Tivoli Network Manager](#) > [Feature Early Access](#) > Tivoli Network Manager - Geographic Discovery and Mapping Solution

## Tivoli Network Manager - Geographic Discovery and Mapping Solution



| Updated March 1, 2017 by [PatrickHourigan](#) | Tags: *None*

### **Update : ITNM 4.2 Fp2 GA of NCP GIS**

As of ITNM 4.2 FP2 the GIS feature has been GA'ed.

Customers wishing to use the GIS feature are strongly encouraged to upgrade to the ITNM 4.2 FP2. Subsequent fix-pack's of ITNM 4.2 are now going to include further features such as offline maps, regional aggregation, etc.

#### ***Note: Existing GIS beta Customers.***

If you have installed via the beta zip delivered on ISM, you must first un-install the GIS feature prior to updating to ITNM 4.2 FP2 following these steps outlined here:

1. Get the beta distro from our [BOX location](#).
2. `unzip -d /tmp ncp_gis-distro**.zip`
3. `cd /tmp`
4. `sh bin/install.sh -u` (This will un-install the application WARs and restart the WAS application server)
5. Install the ITNM 4.2 FP2

If you have installed FP2 atop of a beta GIS installation you must, un-install FP2 first and follow the steps above.

### **Feature Overview**

This capability provides a JazzSM-DASH based integration with Google, OpenStreetMaps, Bing and OpenLayers compliant XYZ maps for Tivoli Network Manager 4.1.1 or later discoveries, giving the following features:

- Exploitation of network manager topology, connectivity, and location data to drive the maps.
- Integration with OMNIBus for state information - intra/inter-site connectivity and per site.
- Site and device-type based map rendering with summarized inter-site connectivity by network layer. Intra-site connectivity and state is also represented for each site.
- Flexible filtering options to control map rendering.
- Access to maps from network manager topology displays via context sensitive menus.
- Access to network manager device and topology displays from the maps.

Below is a snapshot of some of IBM SWG test networks in the U.K, illustrating devices and datacenters overlaid with polled status and discovery context.

Device ID: 128488  
 Name: sbk-pe5-or58.southbank.eu.test.lab  
 Status: ✖  
 Event Distribution: 🔥 2 🔥 3  
 Device Class: CiscoSbox  
 Access IP Address: 172.20.1.5  
 Sys Contact: williamking@uk.ibm.com  
 Description: Cisco IOS Software, 3800 Software (C3825-ADVIPSERVICESK9M), Version 12.4(24)T5, RELEASE SOFTWARE (fc5) Technical Support: http://www.cisco.com/techsupport Copyright (c) 1988-2011 by Cisco Systems, Inc. Compiled Fri 04-Mar-11 05:52 by prod\_rel\_team  
 Domain: SBK  
 Last Discovered: 15/03/2016 21:18  
 Address: St Thomas Hospital Westminster Bridge Rd London SE1 7EH UK  
 Lat/Long: Latitude (51.4903185) Longitude (0.11701336)

Sev	Ack	Node	Alert Group	Summary	First Occurrence	Last Occurrence	Count	Type	ExpireTime	Agent	Manager
<span style="color: red;">✖</span>	No	172.24.2.9	ITNM Monitor	ConfirmDeviceDown/Default Interface Ping fail for 172.24.2.9: ICMP timeout	07/03/2016, 16:29	16/03/2016, 17:27	50,934	Problem	Not Set	nop_poler	ITNM
<span style="color: red;">✖</span>	No	172.20.7.7	ITNM Monitor	ConfirmDeviceDown/Default Interface Ping fail for 172.20.7.7: ICMP timeout	07/03/2016, 16:29	16/03/2016, 17:27	50,934	Problem	Not Set	nop_poler	ITNM
<span style="color: red;">✖</span>	No	172.20.12.1	ITNM Monitor	ConfirmDeviceDown/Default Interface Ping fail for 172.20.12.1: ICMP timeout	07/03/2016, 16:29	16/03/2016, 17:27	50,934	Problem	Not Set	nop_poler	ITNM
<span style="color: yellow;">⚠</span>	No	173.30.5.0	ITNM Monitor	Unknown device on connection to DNS	11/03/2016, 14:42	16/03/2016, 17:27	607	Problem	Not Set	nop_poler	ITNM

Total 5 Selected: 0

## Downloads and Requirements

Please download the latest install image, available via our Sprint deliverable [BOX location](#).

## Installation and Getting Started

Refer to the [Getting Started](#) guide to understand how to install the feature and to get your sample discovery for GIS enrichment complete.

Refer to the [Mandatory Google Map Licensing](#) instructions to ensure, as a Deployer and EndUser, you are compliant with the Terms of Use and licensing for Google Maps.

## Usage

The following are some feature content videos to help you get more familiar with the capabilities of the solution, these cover the different entry points available to gain a geographical context to your network topology.

- [Device Right Click from Hop/Net/Path Views](#) demonstrates how EndUsers can get a geographical context to their devices via right click tools. Alternatively, you can download this recording [here](#).
- [View Right Click from Network Views](#) demonstrates how EndUsers can get a geographical context to their Network Views and Devices via right click tools. Alternatively, you can download this recording [here](#).

- **Geographic Availability Dashboard** demonstrates how EndUsers can see all GIS enriched devices represented on a Map with in-context eventing to drive the AEL for the relevant alarms for each links, device or location. Alternatively, you can download this recording [here](#).

Recent sprints have built upon the original featureset to deliver more functionality, such as full i18n support, further stitchers for rapid prototyping, full right click tool support and much more.

For the latest set of sprint features, check out the [Sprint Release Notes](#).

## Further Integration

This [section demonstrates some of the integration capabilities](#) and URL API's into the mapping provider, which Clients and Business Partners can leverage to define their own GIS dashboards and integrations.

## Map Performance, Tuning and Best Practices

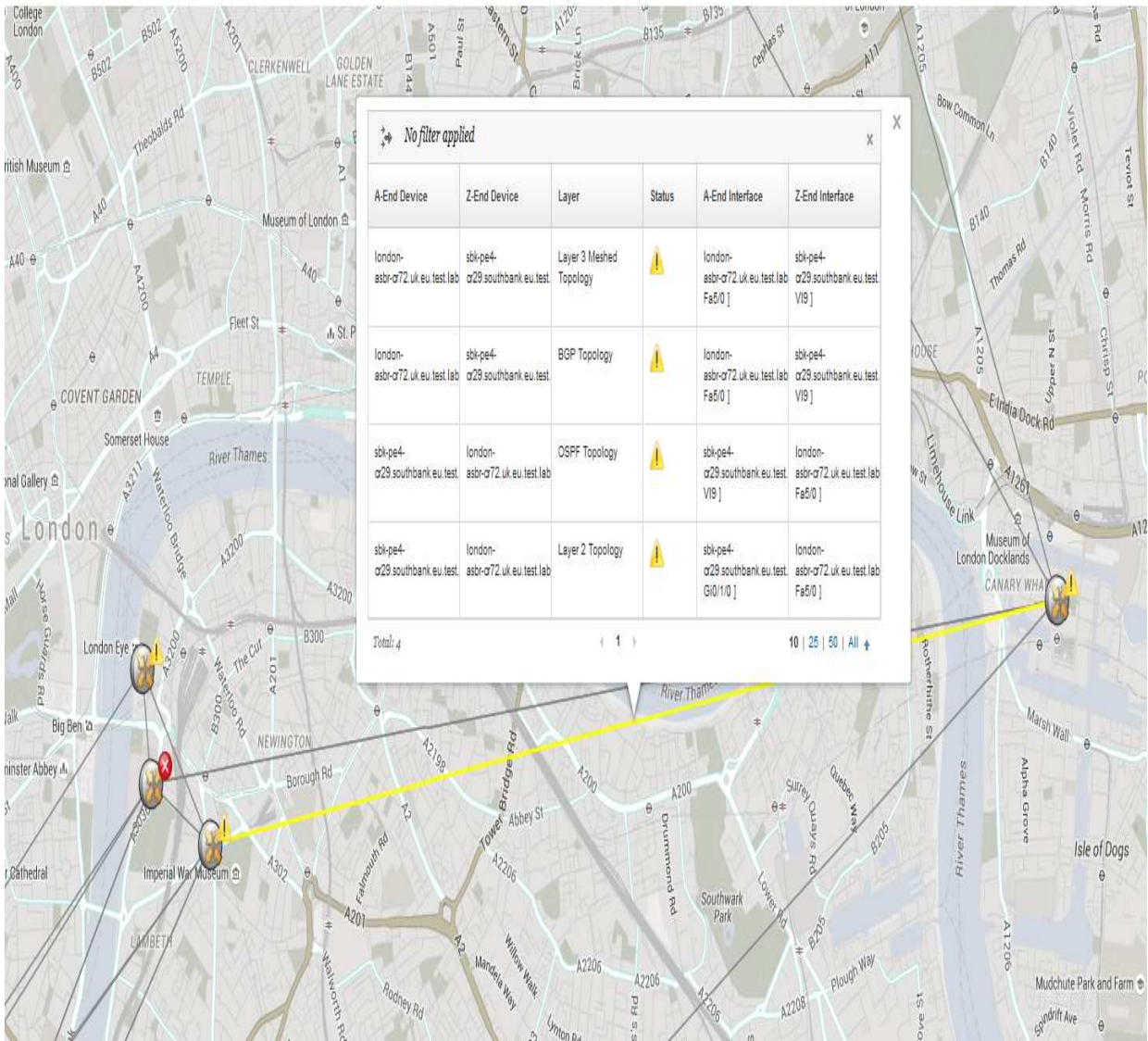
If you're interested in how we compute topology, status and how to tune your feature installation, [refer to this section](#) which covers detail on how to get the best performance for the mapping solution for enterprise grade networks.

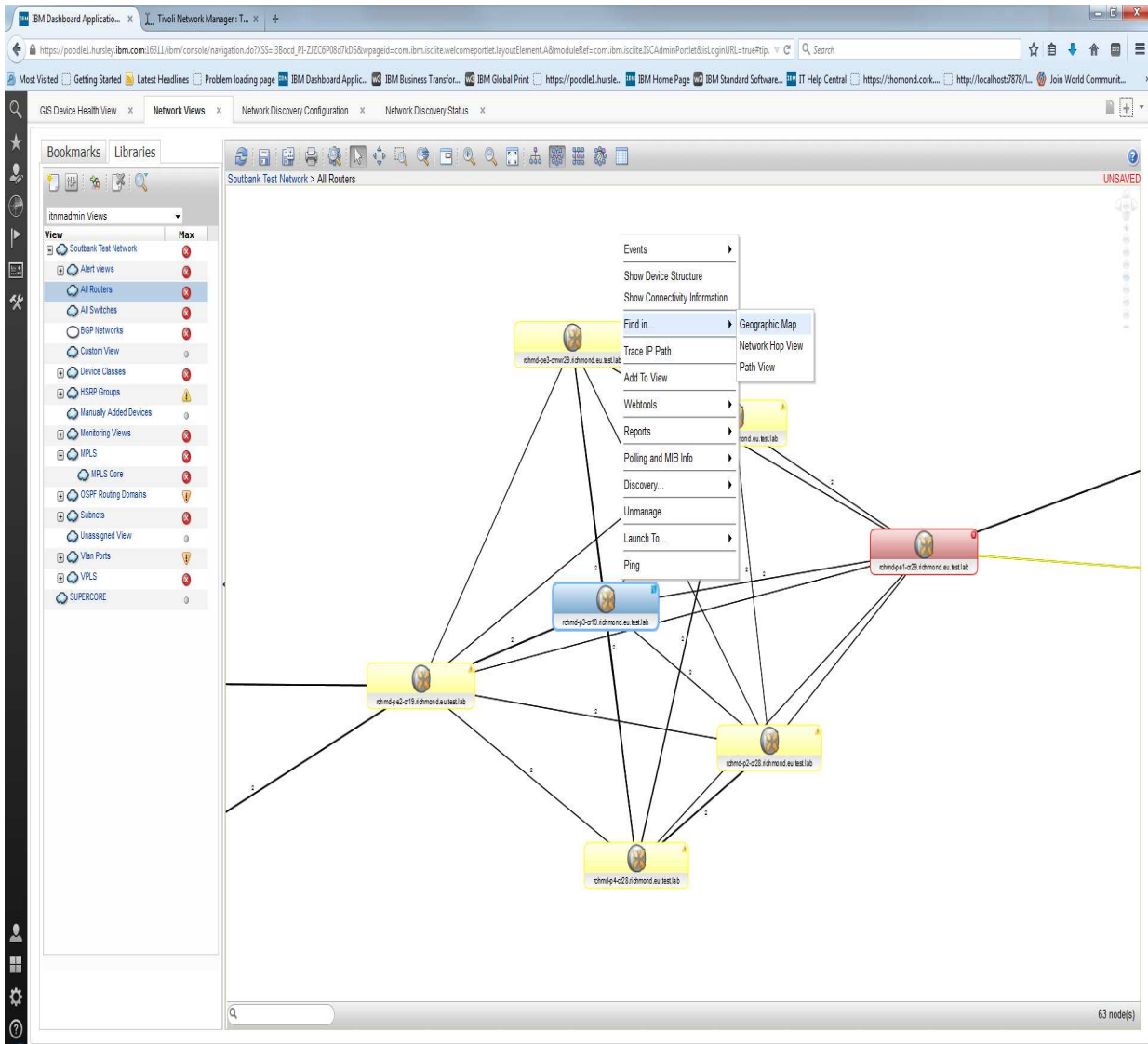
## Feedback and Request for Enhancements

We are keen to take feedback on how to improve and extend this capability. If you have any suggestions on how to improve, we'd love to hear from you. For example some of the potential enhancements could include:

1. **Geographic region bounding boxes.** Scheduled for delivery in ITNM 4.2 FP3 - Available in our Beta from box above
2. **Open Maps Integration.** Scheduled for delivery in ITNM 4.2 FP3 - Available in our Beta from box above. Open Street Maps is becoming the default and Google Maps is being deprecated.

## Some Examples





## Comments

1-1 of 1

Previous | Next



Krishna M Kodali commented on February 2, 2017 [Permalink](#)

GIS capability is now Generally Available as part of ITNM v4.2 in FixPack#2, more information can be read at - <http://www-01.ibm.com/support/docview.wss?uid=swg21995773>

ITNM v4.2 - Fixpack#2 can be downloaded from - <https://www-01.ibm.com/support/docview.wss?uid=swg24042143>

Show 10 | 25 | 50 items per page

Previous | Next

You are in: [Tivoli Network Manager](#) > [Welcome to Tivoli Network Manager](#) > [Feature Early Access](#) > [Tivoli Network Manager - Geographic Discovery and Mapping Solution](#) > Getting Started

## Getting Started



| Updated May 23, 2016 by [PatrickHourigan](#) | Tags: *None*

### Before you start...

This feature is supported and installable on JazzSM-DASH only. The original TIP compliant version has been deprecated and whilst still available to download from the ISM library is not being maintained or supported. However, the Tivoli Network Manager 4.1.1 client base is supported, since this feature is compliant with a Tivoli Network Manager 4.1.1 Core and NCIM dB, for which we have an anonymous reference from a Asia-Pac, based client.

### Installation

This package can be installed on DASH v3.1.+ only. Noting, it is compliant with both ITNM 4.1.1 or higher backend ITNM installations.

#### 4.2 Compliant Installation

##### **Step 1 of 1: Install the GIS widget on the JazzSM-DASH Server**

1. unzip ncp\_gis-distro-4.2\_GA.zip -d tmp
2. cd tmp
3. Copy the Menus and Tools from the unzipped distro image cp tmp/profiles/TIPProfile/etc/tnm/\* /opt/IBM/netcool/precision\_gui/profile/etc/tnm/
4. sh bin/install.sh -r

Well done! your installed..Next step is to setup your discovery to GIS enrich the devices, the tmp/samples directory has 2 samples, 1 for GeoBySysLocation and another for GeoByLookup

#### 4.1.1 Compliant Installation

##### **Step 1 of 3: Pre-Req - Prepare your Network Manager NCIM dB by running the dB modify script**

- It's necessary to update the NCIM schema geographic tables for the solution to function.The distribution image contains a scripts to modify the format of the geographic tables in your relevant NCIM schema. There are 2 variants of this script one for Oracle and one dB2.
  - Below covers an example of the dB2 execution.
  - On the system where you have deployed the Network Manager db2/Oracle instance or a system that has db2/oracle client access to the db instance.
1. Unpack the ncp\_gis-distro-TIP(DASH).zip i.e. unzip ncp\_distro-TIP.zip -d ~/installs/tmp
  2. cd ~/installs/tmp
  3. Run the bin/db2/update\_geo\_location.sh
    - [root@server ncp\_gis-distro]# sh bin/db2/update\_geo\_location.sh GEOTEST ncim
    - Usage: bin/db2/update\_geo\_location.sh <database\_name> <user\_name> <password>
    - [root@serverncp\_gis-distro]# sh bin/db2/update\_geo\_location.sh GEOTEST ncim ncim

- Example output:

```
* Summary Table:
Type      Number      Repetitions Total Time (s) Min Time (s)  Max Time (s)  Arithmetic Mean Geometric Mean Row(s) Fetched Row(s) Output
-----
Statement 1          1          0.000143    0.000143     0.000143     0.000143     0.000143     0          0
Statement 2          1          0.000288    0.000288     0.000288     0.000288     0.000288     0          0
Statement 3          1          0.000365    0.000365     0.000365     0.000365     0.000365     0          0
Statement 4          1          0.020881    0.020881     0.020881     0.020881     0.020881     0          0
Statement 5          1          0.000000    0.000000     0.000000     0.000000     0.000000     0          0
Statement 6          1          0.000210    0.000210     0.000210     0.000210     0.000210     0          0
Statement 7          1          0.000188    0.000188     0.000188     0.000188     0.000188     0          0
Statement 8          1          0.000000    0.000000     0.000000     0.000000     0.000000     0          0
Statement 9          1          0.000000    0.000000     0.000000     0.000000     0.000000     0          0
Statement 10         1          0.000000    0.000000     0.000000     0.000000     0.000000     0          0
Statement 11         1          0.000000    0.000000     0.000000     0.000000     0.000000     0          0
Statement 12         1          0.876429    0.876429     0.876429     0.876429     0.876429     0          0

* Total Entries:      12
* Total Time:         0.898504 seconds
* Minimum Time:      0.000000 seconds
* Maximum Time:      0.876429 seconds
* Arithmetic Mean Time: 0.074875 seconds
* Geometric Mean Time: 0.000004 seconds

-----
* Timestamp: Wed Jan 21 2015 13:05:00 GMT
DB20000I The SQL DISCONNECT command completed successfully.
```

### **Step 2 of 3 : Export your 4.1.1 TIP configuration**

1. Tar/Zip the \$NCHOME/precision/profiles/TIPProfile/etc/tnm/ folder and transfer it to the DASH server at /tmp or somewhere

### **Step 3 of 3: Install the GIS widget on the JazzSM-DASH Server**

1. Make a directory called /opt/IBM/tivoli/ncp\_netcool/precision\_gui/profile/
2. Unpack the etc/tnm tar/zip you transferred to /tmp from the TIP Server
3. unzip ncp\_gis-distro-4.2\_GA.zip -d tmp
4. cd tmp
5. Copy the Menus and Tools from the unzipped distro image cp tmp/profiles/TIPProfile/etc/tnm/\* /opt/IBM/netcool/precision\_gui/profile/etc/tnm/
6. sh bin/install.sh -r

Well done! your installed..Next step is to setup your discovery to GIS enrich the devices, the tmp/samples directory has 2 samples, 1 for GeoBySysLocation and another for GeoByLookup



The install distro comes with 2 example stitchers to help you get devices geo enriched. These are located in the distro image, within the samples dir as:

- `samples/GeoByLookup` : This method is used to stitch the geo data to devices and build geo and region collections based on a lookup to a 3rd party table. See the install and deployment notes for this stitcher at `tmp/samples/GeoByLookup/getting-started.txt`
- `samples/GeoBySysLocation` : This method is used to stitch the geo data from a devices `syslocation` field in order to build the minimum geo collection. See the install and deployment notes for this stitcher at `tmp/samples/GeoBySysLocation /getting-started.txt`

The format of the `SysLocation` field should be:

`< location address >;< longitude >;< latitude >`



Once you have setup the stitchers, run a discovery of that domain and validate the Geographic tables have been populated as expected and the devices are enriched with the relevant `SysLocation` data.

To verify the relevant devices have been discovered:

1. Query the `NCIM.GEOGRAPHICLOCATION` table to verify, there are locations present as a collection for each of your known Geo enriched devices.
2. Alternatively, login to JazzSM-DASH and open the GIS Device Map under Availability/Incident->Network Geography . Verify that your devices are rendered on the Map
3. Alternatively, create a Network View based on the following Collection view to render all of your Geo enriched devices in a single network view.

## Validating your Install

Once the feature has been installed and devices are geo-enriched you can verify in a number of ways

1. Validate a RESTful Service result, if you have geo enriched discovery data present in your NCIM dB.
  - o Test a valid JSON CURI URL such  
as: [https://\\${SERVER}:\\${PORT}/ibm/console/nm\\_rest/topology/devices/geo/all](https://${SERVER}:${PORT}/ibm/console/nm_rest/topology/devices/geo/all)
  - o The JSON payload will have a body of attributes for geo enriched devices, if the payload is empty, then you may not have properly enriched the devices.
2. Validate the Google Map Widget loads correctly within the JazzSM-DASH container. Note there are license obligations for the use of Google Maps and it's APIs. These are documented for public consumption in the relevant SMC Article section, for development purposes the general open license applies.
  - o Navigate to the  
URL [https://\\${SERVER}:\\${PORT}/ibm/console/ncp\\_gis/NetKitMapWidget.jsp](https://${SERVER}:${PORT}/ibm/console/ncp_gis/NetKitMapWidget.jsp)

## Comments

---

1-1 of 1

[Previous](#) | [Next](#)



[DominicLehr](#) commented on June 30, 2016 [Permalink](#)

Hi Pat,

nice thing - especially the integration into DASH, nice and smooth. One tiny thing: How about adding a capability to use custom maps? There are a lot of OpenSource maps out there, OpenStreetMap just to name one.

BR

Dom

---

Show 10 | [25](#) | [50](#) items per page

[Previous](#) | [Next](#)

You are in: [Tivoli Network Manager](#) > [Welcome to Tivoli Network Manager](#) > [Feature Early Access](#) > [Tivoli Network Manager - Geographic Discovery and Mapping Solution](#) > Mandatory Google Licensing

## Mandatory Google Licensing



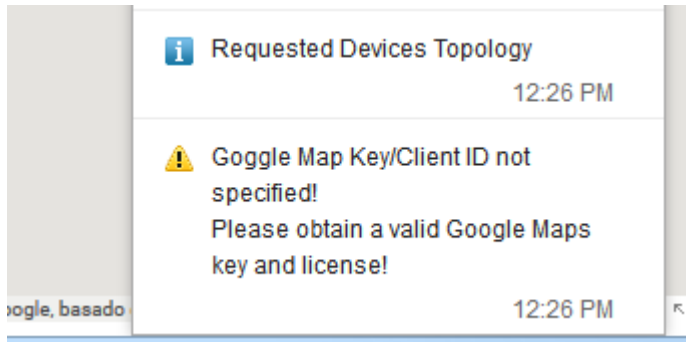
Like | Updated May 23, 2016 by [PatrickHourigan](#) | Tags: *None*

This widget is based on an integration with the Google Maps API.

**IMPORTANT!** - As a Deployer and User of this solution you must ensure you are compliant with the [Terms of Use for Google Maps](#)

As distributed, there is a default no license applied and when End Users launch the Maps in JazzSM-DASH to following warning is displayed on the Information Toaster.

For example:



To apply your license:

1. On your JazzSM-DASH Server.
2. `cd /opt/IBM/JazzSM/profile/installedApps/JazzSMNode01Cell/isc.ear/ncp_gis.war/`
3. Create a backup of resources/config.json
4. Edit resources/config.json, amend the property `mapProvider.baseLayers.baseLayerURL` to include your API or ClientID key and save the file, the changes will take effect one the map has been reloaded, ensure the browser cache is cleared for modifications to take effect.

**For Example:**

Base Unlicensed

```
"mapProvider": {  
  "baseLayers": [  
    {  
      "baseLayerName": "Google",  
      "baseLayerURL": "https://maps.google.com/maps/api/js?libraries=visualization,drawing,geometry&sensor=true&v=3.exp"  
    }  
  ]  
}
```

API Key Applied

```
"mapProvider": {  
  "baseLayers": [  
    {  
      "baseLayerName": "Google",  
      "baseLayerURL": "https://maps.google.com/maps/api/js?libraries=visualization,drawing,geometry&sensor=true&v=3.45&key=AIzaSyD6f9Rb7ktZIVdrjzax1X7K2_uk6f7U" }  
    ]  
  }  
}
```

If you fail to provide a valid Google Maps License you will see the following error on Maps load.

This page was unable to display a Google Maps element. The provided Google API key is invalid or this site is not authorized to use it. Error Code: InvalidKeyOrUnauthorizedURLMapError


OK

## Comments

*There are no comments.*

You are in: [Tivoli Network Manager](#) > [Welcome to Tivoli Network Manager](#) > [Feature Early Access](#) > [Tivoli Network Manager - Geographic Discovery and Mapping Solution](#) > [Further Maps Integration](#)

## Further Maps Integration

 Like | Updated July 31, 2015 by [PatrickHourigan](#) | Tags: *None*

The Mapping widget is fully URL addressable to facilitate custom integration's.

### Available URLs

The mapping widget can be launched by navigating to the URL:

[https://{server}:{port}/ibm/console/ncp\\_gis/NetKitMapWidget.jsp](https://{server}:{port}/ibm/console/ncp_gis/NetKitMapWidget.jsp)

There are a set of parameters which can be used to scope the device and link membership of the mapped topology. These are as follows:

Parameter	Behavior	Allowed Range/Value(s)	Example
<b>NONE</b>	<p>Renders the entire GIS enriched topology.</p> <p>This covers all devices, locations and links for all domains in your discovery NCIM dB which have been GIS enriched.</p> <p>This query is used in the GIS Availability Dashboard to render all topology.</p>	n/A	<p><a href="https://{server}:{port}/ibm/console/ncp_gis/NetKitMapWidget.jsp">https://{server}:{port}/ibm/console/ncp_gis/NetKitMapWidget.jsp</a></p>
<b>viewId</b>	<p>Scopes the topology based on the device membership of the specified Network View.</p> <p>This will render all devices and links within the Network View which have been GIS enriched.</p> <p>This query is used from the View right click tool "Show On Map".</p>	<p>NCP_VIEW_{\$viewId} or \$viewId</p> <p>Where viewId is a positive numeric value only.</p> <p>The regex for this range is: <code>((NCP_VIEW_)(\d+))(\d+)</code></p>	<p><a href="https://{server}:{port}/ibm/console/ncp_gis/NetKitMapWidget.jsp?viewId=760">https://{server}:{port}/ibm/console/ncp_gis/NetKitMapWidget.jsp?viewId=760</a></p> <p>or</p> <p><a href="https://{server}:{port}/ibm/console/ncp_gis/NetKitMapWidget.jsp?viewId=NCP_VIEW_760">https://{server}:{port}/ibm/console/ncp_gis/NetKitMapWidget.jsp?viewId=NCP_VIEW_760</a></p>
<b>deviceId</b>	<p>Scopes the topology based on a specific device or set of devices.</p> <p>This will render all devices and links specified which have been GIS enriched.</p> <p>This query is used from the Device right click tool "Show On Map". This is available in the Hop, Network and Path views.</p>	<p>A comma separated list of positive numeric value device Ids.</p> <p>The regex for this range is: <code>([1-9][0-9]*.[ ])*[1-9][0-9]*</code></p>	<p><a href="https://{server}:{port}/ibm/console/ncp_gis/NetKitMapWidget.jsp?deviceId=19078&amp;deviceId=19062&amp;deviceId=19067">https://{server}:{port}/ibm/console/ncp_gis/NetKitMapWidget.jsp?deviceId=19078&amp;deviceId=19062&amp;deviceId=19067</a></p> <p>or</p> <p><a href="https://{server}:{port}/ibm/console/ncp_gis/NetKitMapWidget.jsp?deviceId=19078,19062,19067">https://{server}:{port}/ibm/console/ncp_gis/NetKitMapWidget.jsp?deviceId=19078,19062,19067</a></p>
<b>bounds</b>	<p>Scope the topology based on the geographic range specified. The range is defined as the N.W -&gt; S.E geographical area.</p> <p>This will render all devices and links which are determined to be within the specified range.</p> <p>This query can be useful if you intend on finding devices within a range of a specific LONG:LAT. For example: Topology within 5 miles of my location.</p>	<p>A defined NW to SE geographical area which matches the format: <code>((LAT, LONG), (LAT, LONG))</code></p> <p>Each LAT and LONG are signed numeric values.</p> <p>The regex for this range is: <code>((\d*)(\d+)([+-]? \d* \d+)(?![-+0-9\d.])?)((\d*)(\d+)([+-]? \d* \d+)(?![-+0-9\d.])?)((\d*)(\d+)([+-]? \d* \d+)(?![-+0-9\d.])?)((\d*)(\d+)([+-]? \d* \d+)(?![-+0-9\d.])?)</code></p>	<p><a href="https://{server}:{port}/ibm/console/ncp_gis/NetKitMapWidget.jsp?bounds=%28%2851.50,-0.15%29,%2851.45,-0.09%29%29">https://{server}:{port}/ibm/console/ncp_gis/NetKitMapWidget.jsp?bounds=%28%2851.50,-0.15%29,%2851.45,-0.09%29%29</a></p>

Parameter	Behavior	Allowed Range/Value(s)	Example
locationId	<p>Scope the topology based on a Network Manager Location Collection Id.</p> <p>This will render a single location, with devices at that location listed in the right click datagrid.</p>	<p>A positive numeric value representing the location collection entityId in ITNM.</p> <p>The regex for this range is : (\d+)</p>	<p><a href="https://{server}:{port}/ibm/console/nkmap/NetKitMap.jsp?locationId=47672">https://{server}:{port}/ibm/console/nkmap/NetKitMap.jsp?locationId=47672</a></p>

**Auto Filtering via URL Parameters**

Also included, is the ability to auto filter the rendered topology based on a collection of URL parameters which can be supplied with any of the above scoping parameters.

The auto filter capability allows the map to be filtered by any of the Map Configuration interactive options, for example, by Domain, Device Class, Connectivity Layer, Enabling/Disabling Status, etc.

Parameter	Behavior	Allowed Range/Value(s)	Example
config.filter.domain	Specifying this parameter will reduce the scoped topology to only render topology in the specified domain.	<p>A comma separated list of positive numeric value domain Ids.</p> <p>The regex for this range is: ((1-9)[0-9]*,[ ])*[1-9][0-9]*</p>	<p><a href="https://{server}:{port}/ibm/console/ncp_gis?config.filter.domain=5">https://{server}:{port}/ibm/console/ncp_gis?config.filter.domain=5</a></p> <p>Show all topology and auto filter to only specified domain.</p>
config.filter.class	Specifying this parameter will reduce the scoped topology to only render topology which matches the specified class type.	<p>A comma separated list of positive numeric value entity class type Ids.</p> <p>The regex for this range is: ((1-9)[0-9]*,[ ])*[1-9][0-9]*</p>	<p><a href="https://{server}:{port}/ibm/console/ncp_gis?bounds=((51.50,-0.15),(51.45,-0.09))&amp;config.filter.class=110">https://{server}:{port}/ibm/console/ncp_gis?bounds=((51.50,-0.15),(51.45,-0.09))&amp;config.filter.class=110</a></p> <p>Show devices in a geographic range and CiscoCat35xx class devices only.</p>
config.filter.layer	Specifying this parameter will reduce the scoped topology to only render topology links which match the defined topology layer type.	<p>A comma separated list of positive numeric value topology layer type Ids.</p> <p>The regex for this range is: ((1-9)[0-9]*,[ ])*[1-9][0-9]*</p>	<p><a href="https://{server}:{port}/ibm/console/ncp_gis?deviceId=19078&amp;deviceId=19062&amp;config.filter.layer=72">https://{server}:{port}/ibm/console/ncp_gis?deviceId=19078&amp;deviceId=19062&amp;config.filter.layer=72</a></p> <p>Show specific devices and auto filter to connections.</p>
config.enableLinks	Specifying this parameter will exclude/include links from the scoped topology.	<p>true or false.</p> <p>The regex for this range is: ((true) (false))</p>	<p><a href="https://{server}:{port}/ibm/console/ncp_gis?config.enableLinks=false">https://{server}:{port}/ibm/console/ncp_gis?config.enableLinks=false</a></p> <p>Show all topology and auto filter exclude</p>
config.enableLinkStatus	Specifying this parameter will exclude/include link status from the scoped topology.	<p>true or false.</p> <p>The regex for this range is: ((true) (false))</p>	<p><a href="https://{server}:{port}/ibm/console/ncp_gis?config.enableLinkStatus=false">https://{server}:{port}/ibm/console/ncp_gis?config.enableLinkStatus=false</a></p> <p>Show all topology and auto filter exclude rendered without severity colouring.</p>
config.enableTopologyStatus	Specifying this parameter will exclude/include device/location status from the scoped topology.	<p>true or false.</p> <p>The regex for this range is: ((true) (false))</p>	<p><a href="https://{server}:{port}/ibm/console/ncp_gis?config.enableTopologyStatus=false">https://{server}:{port}/ibm/console/ncp_gis?config.enableTopologyStatus=false</a></p> <p>Show all topology and auto filter exclude Device and Location markers will be rendered severity icons.</p>
config.minTopologyStatus	Specifying this parameter will specify the minimum device/location status to render for the scoped topology.	<p>0 : Clear or Higher 1: Indeterminate or Higher 2: Warning or Higher 3: Minor or Higher 4: Major or Higher 5: Critical only</p> <p>The regex for this range is: [0-5]</p>	<p><a href="https://{server}:{port}/ibm/console/ncp_gis?config.minTopologyStatus=2">https://{server}:{port}/ibm/console/ncp_gis?config.minTopologyStatus=2</a></p> <p>Show all topology and auto filter to only status of Warning or higher.</p>
config.minLinkStatus	Specifying this parameter will specify the minimum links status to render for the scoped topology.	<p>0 : Clear or Higher 1: Indeterminate or Higher 2: Warning or Higher 3: Minor or Higher 4: Major or Higher 5: Critical only</p> <p>The regex for this range is: [0-5]</p>	<p><a href="https://{server}:{port}/ibm/console/ncp_gis?config.minLinkStatus=5">https://{server}:{port}/ibm/console/ncp_gis?config.minLinkStatus=5</a></p> <p>Show all topology and auto filter to only Critical.</p>



## Comments

*There are no comments.*

You are in: [Tivoli Network Manager](#) > [Welcome to Tivoli Network Manager](#) > [Feature Early Access](#) > [Tivoli Network Manager - Geographic Discovery and Mapping Solution](#) > Maps Performance, Tuning and Best Practice

## Maps Performance, Tuning and Best Practice



Like | Updated March 23, 2015 by [Paul Partington](#) | Tags: *None*

There are 2 main drivers of map performance:

1. The scope of the topology to render, which is a function of the number of devices by the number of links or topological relationships between these devices.
2. The refresh rate for status collection on the topology.

Topology is queried upon map load or when a new discovery cycle is detected for the listed domain(s). This is based on :

- A single query to the Network Manager NCIM dB to fetch the devices topology.
- A query for each location aggregation in the inspected topology.
- A single query for links relationships between the devices/locations about to be rendered on the map.

Therefore, the larger the topology in scope the greater the number of location aggregations and link relationships to be processed.

Luckily, we've given the ability to scope a topology in many ways, as described in this document, you can scope by a Network View, select devices, a geographical range or indeed by all GIS enriched topology.

It's possible to render a very large topology (many many thousands of devices) with many degrees of topological layer over several domains, but this method may not give you the best visibility of your network. There may be simply too much map markers present in close proximity to make the map useful to an Ops persona.

Instead consider some of these techniques:

1. If you are making use of Geographic Region based views, then you can use the RightClick for Views or specify the viewId parameter to scope the topology on just devices in that specific views membership. The same applies for any Network View, this is also useful to segregate by domains in a multi-domain environment.
2. If you wish to render geography around a fixed geographic area, i.e. a City, then you have the capability of using a geographic bounds by specifying the bounds parameter.
3. If you do render the entire network, then the Map Filtering capability allows you to reduce the mapped markers to suit your needs.

Status is queried in 3 separate tracks from the defined Omnibus WebGUI server:

1. Device Status Track, whereby a single query is made to return the event summary data for all visible devices on map.
2. Location Status Track, whereby a single query is made to return the event summary data for all aggregate locations on map.
3. Link Status Track, whereby a single query is made to return the event summary data for all aggregate links on map.

Using this technique allows us to reduce the volume of status invocations being made and to post process the status on a per device, aggregate location or aggregate link basis.

Status is queried on a N second basis and is configured in the `$TIPHOME/profiles/TIPProfile/installedApps/TIPCell/isc.ear/ncp_gis.war/resources/config.json` file by the property: "mapStatusRefreshInterval", which is set to a 20 second default refresh. When deploying you should consider the refresh rate in tandem with your Omnibus refresh rate and any configured Network Manager poll policy cycle.

There is a second status refresh rate property, "gridStatusRefreshInterval", which is used to refresh the status indicator only when a data grid is made visible on a Device, Link or Location. In general there is not need to modify this rate, unless the intended overall refresh rate is significantly higher.

## Comments

---

1-1 of 1

[Previous](#) | [Next](#)

---



[DominicLehr](#) commented on September 4, 2017 [Permalink](#)

Hi,

is there a way to limit the amount of GIS data that is initially loaded when opening the GIS views? Reason me asking is that our customer has a very large topology - if this is loaded on init, DASH crashes due to running out of memory. For further details, please refer to PMR 20192,999,724.

Thanks and BR,  
Dominic

---

Show 10 | [25](#) | [50](#) items per page

[Previous](#) | [Next](#)

---

You are in: [Tivoli Network Manager](#) > [Welcome to Tivoli Network Manager](#) > [Troubleshooting](#) > [Network Manager TopologyAPI](#) > JACC Authorization failed for bean - ConfigurationAccessFacade

## JACC Authorization failed for bean - ConfigurationAccessFacade

 Like | Updated April 20, 2015 by [Pimmy](#) | Tags: *None*

### Use case:

This issue has surfaced for the first time in an environment where the IBM Tivoli Network Manager (ITNM) has been integrated with the IBM Tivoli Netcool Configuration Manager (ITNCM) via the ITNM Topology API.

### Problem:

When the user changes the authentication mechanism (for example from *File Based Authentication* to *Active Directory Authentication*), ITNCM throws the following exception:

```
org.omg.CORBA.UNKNOWN:
>> SERVER (id=4773e3aa, host=nmstcr02.gtb.viasat.com) TRACE START:
>> javax.ejb.EJBAccessException: SECJ0053E: Authorization failed for defaultWIMFileBasedRealm/itnadmin while invoking (Bean)isc#itnm_topo_api_server-
ejb.jar#ConfigurationAccessFacade getITNMConfigurationInfo::1 JACC Authorization failed for bean: ConfigurationAccessFacade
>> at com.ibm.ws.security.core.SecurityCollaborator.performAuthorization(SecurityCollaborator.java:685)
>> at com.ibm.ws.security.core.EJSSecurityCollaborator.preInvoke(EJSSecurityCollaborator.java:275)
>> at com.ibm.ejs.container.EJSContainer.preInvokeAfterActivate(EJSContainer.java:4066)
>> at com.ibm.ejs.container.EJSContainer.EjbPreInvoke(EJSContainer.java:3353)
```

Please note that here the issue is the authorization stage and not the authentication - the 'itnadmin' user can successfully login to TIP.

### Reason:

When the authentication mechanism has been changed from *File Based* to *Active Directory*, the 'itnadmin' user roles were not propagated to the new Federated Repository. This means that despite the 'itnadmin' user is present in the and the user can login, this user has no rules in the new Federated Repository - (the user can log in, but is not authorized to access).

### Remedy:

1. Remove all authorization records including groups for the problematic user.
2. Reassign all required roles.

### Troubleshooting

Such authorization failures are common for environments which authentication mechanism has been changed. In order to troubleshoot such issues the following traces would be required:

**Object Request Broker (ORB) MustGather:** <http://www-01.ibm.com/support/docview.wss?uid=swg21157005#show-hide>

#### WAS Security trace:

```
*=info:com.ibm.ws.security.*=all:com.ibm.websphere.security.*=all:com.ibm.websphere.wim.*=all:com.ibm.wsspi.wim.*=all:com.ibm.ws.wim.*=all:SASRas=all:ORBRas=a
```

Please note that this will generate massive amount of data, hence in order not to loose valuable information, please increase the trace file size to 100Mb and set the log rotation to 20 files. Also, prior the test, delete any old trace files, restart the server, carry out the test and collect the files as soon as the test finishes (ot the issue appears).

#### Client JVM trace:

In the ITNM/ITNCM integration the client JVM is started with the /opt/IBM/tivoli/netcool/ncm/bin/itncm.sh script. In order to enable tracing in the client JVM, the following properties have been added to the script:

```
WORKER_TRACE_ARGS=" -DtraceSettingsFile=TraceSettings.properties -
Dcom.ibm.ejs.ras.lite.traceSpecification=SASRas=all:com.ibm.ws.security.*=all:com.ibm.sec.authz.*=all -
Dcom.ibm.ejs.ras.lite.traceFileName=MyTraceFile.log -Dcom.ibm.CORBA.Debug=true -Dcom.ibm.CORBA.CommTrace=true -
Dcom.ibm.CORBA.Debug.Output=${INSTALL_DIR}/MyTraceFile.log"
CLASSPATH="${CLASSPATH}:${INSTALL_DIR}/drivers/lib_legacy/itnm/com.ibm.ws.ejb.thinclient_7.0.0.jar:${INSTALL_DIR}/drivers/lib_legacy/itnm/com.ibm.ws.admin.cli
```

#### JACC trace: com.tivoli.pd.as.\*=all

(there should also be some log files under the `WebSphere\AppServer\java\jre\PolicyDirector\log` which may contain extra information)

### Comments

*There are no comments.*

You are in: [Tivoli Network Manager](#) > [Welcome to Tivoli Network Manager](#) > [Troubleshooting](#) > [Network Manager TopologyAPI](#) > Authentication failure in an ITNCM and ITNM Topology API integration due to inconsistently set TIP realm (TIPRealm)

## Authentication failure in an ITNCM and ITNM Topology API integration due to inconsistently set TIP realm (TIPRealm)



Like | Updated April 20, 2015 by [Pimmy](#) | Tags: *None*

### Use case:

This issue is particular to *IBM Tivoli Netcool Configuration Manager* (ITNCM) and *IBM Tivoli Network Manager* integrations where the ITNCM client runs in a separate JVM started via the `/opt/IBM/tivoli/netcool/ncm/bin/itncm.sh` script.

### Problem:

Authentication failure - Client logs state that realm name is 'TIPRealm', while it has been set to 'defaultWIMFileBasedRealm'. Following entries in the traces could give a clue:

```
[15/04/2015 13:03:28:192 GMT+00:00] 00000000 < jectCSiv2UtilityImpl.CSiv2EffectivePerformPolicy
getPerformClientAuthMechOID oid:2.23.130.1.1.1 Exit 1324699381
[15/04/2015 13:03:28:192 GMT+00:00] 00000000 3 com.ibm.ws.security.auth.WSCredentialImpl
isForwardable current_realm is: TIPRealm
[15/04/2015 13:03:28:192 GMT+00:00] 00000000 3 com.ibm.ws.security.auth.WSCredentialImpl Target
realm does not match current realm, target realm: defaultWIMFileBasedRealm, current realm:
TIPRealm.
[15/04/2015 13:03:28:192 GMT+00:00] 00000000 3 com.ibm.ws.security.auth.WSCredentialImpl
isForwardable() -> false
[15/04/2015 13:03:28:192 GMT+00:00] 00000000 < com.ibm.ws.security.auth.WSCredentialImpl
isForwardable Exit false
[15/04/2015 13:03:28:192 GMT+00:00] 00000000 3 rityLocalObjectBaseL13Impl.CSICredentialsManager
There is no invocation subject on the current thread; Login will be performed for
defaultWIMFileBasedRealm/null <<==== Realm mismatch TIPRealm vs defaultWIMFileBasedRealm.
```

```
org.omg.CORBA.UNKNOWN:
```

```
>> SERVER (id=4773e3aa, host=nmstcr02.gtb.viasat.com) TRACE START:
>> javax.ejb.EJBAccessException: SECJ0053E: Authorization failed for TIPRealm/itnadmin while
invoking (Bean)isc#itnm_topo_api_server-ejb.jar#ConfigurationAccessFacade
getITNMConfigurationInfo::1 JACC Authorization failed for bean: ConfigurationAccessFacade
>> at
com.ibm.ws.security.core.SecurityCollaborator.performAuthorization(SecurityCollaborator.java:685)
>> at com.ibm.ws.security.core.EJSSECollaborator.preInvoke(EJSSECollaborator.java:275)
>> at com.ibm.ejs.container.EJSContainer.preInvokeAfterActivate(EJSContainer.java:4066)
>> at com.ibm.ejs.container.EJSContainer.EjbPreInvoke(EJSContainer.java:3353)
```

### Resolution:

By default the Topology API use 'TIPRealm' for a realm name. This can be changed however by adding the "itnm\_api\_login\_realm" property to the server.xml configuration file. For example:

```
<systemProperties xmi:id="Property_3" name="itnm_api_login_realm" value="defaultWIMFileBasedRealm"
required="false"/>
```

or as a System.property if the TopologyAPI is invoked through an external JVM

## Comments

17/12/2019

Tivoli Network Manager : Authentication failure in an ITNCM and ITNM Topology API integration due to inconsistently set TIP realm (TIPReal...

*There are no comments.*

You are in: [Tivoli Network Manager](#) > [Welcome to Tivoli Network Manager](#) > [Reference](#) > Collector Developer Guides

## Collector Developer Guides



Like | Updated February 29, 2016 by [IanFelstead](#) | Tags: [collector](#), [itnm](#)

# ITNM Collector Developer Documentation

For ITNM 3.9 and earlier please see the attached Field Guide; [CollectorDevGuide\\_V1.4\\_For\\_ITNM\\_v3.9.pdf](#) (this is the TFG\_Collector\_Developer\_Guide\_V1.4.pdf file that was previously available via <http://www-01.ibm.com/support/docview.wss?uid=swg27014787>).

For ITNM 4.1 please see the attached file [CollectorDevGuide\\_v1.5\\_For\\_ITNM\\_v4.1.tar.gz](#) which contains the Collector Developer Guide, Perl Framework documentation and Java Framework documentation.

For ITNM 4.1.1 please see the attached file [CollectorDevGuide\\_v1.6\\_For\\_ITNM\\_v4.1.1.tar.gz](#) which contains the LTE updates to the Collector Developer Guide and Perl/Java Framework documentation

For ITNM 4.2 please see the attached file [CollectorDevGuide\\_v1.7\\_For\\_ITNM\\_v4.2.tar.gz](#) . This release sees the addition of a new migration sections to the generic and perl-API documentation packages which provide advice on migration and act as a quick reference showing what features have been added/removed/modified since 3.9.

Further Collector development related materials can be found [here](#).

## Comments

*There are no comments.*

You are in: [Tivoli Network Manager](#) > [Welcome to Tivoli Network Manager](#) > [Reference](#) > [Collector Developer Guides](#) > [Additional Collector Development Articles](#)

## Additional Collector Development Articles



[Like](#) | Updated June 2, 2015 by [IanFelstead](#) | Tags: *None*

The following pages provide more information regarding Collector development.

[Using Collectors to Enrich a Network Discovery With 'Undiscoverable' Data \(incl. full example collector\)](#)

Blog Entry: [Interrogating ITNM Collectors & cloning an EMS \(with ncp\\_query\\_collector.pl & fake\\_collector.pl\)](#)

### Comments

*There are no comments.*



You are in: [Tivoli Network Manager](#) > [Welcome to Tivoli Network Manager](#) > [Reference](#) > [Collector Developer Guides](#) > [Additional Collector Development Articles](#) > Using Collectors to Enrich a Network Discovery With 'Undiscoverable' Data (incl. full example collector)

## Using Collectors to Enrich a Network Discovery With 'Undiscoverable' Data (incl. full example collector)

 Like | Updated June 2, 2015 by [IanFeistead](#) | Tags: [collector](#), [itm](#)

### Introduction

Collectors aren't just about EMS support - they can also be a convenient way to enhance traditional SNMP discoveries too.

In this article I'll be demonstrating how you could write a simple topology enhancing Collector for use in scenarios where topology data gathered via device or EMS interrogation is unavailable or incomplete. For example we could be talking about a traditional ping/SNMP discovery which contains some types of devices for which topology data is unavailable via SNMP, or even a Collector/EMS based discovery where the EMS can't provide all necessary topology data.

### The Problem and a Proposed Collector Based Solution

Imagine that as things stand our layer 1 and layer 2 topologies have been discovered but we have a problem device DeviceA.ibm.com which doesn't allow us to discover its layer 1 link to '172.20.1.2' or its layer 2 link to 172.20.1.101 (on ifIndex 11) and so our topology looks something like this;



Network Manager is already flexible enough to assist in such scenarios, for example one way would be to use the GUI to manually add links (see 'Editing network topology' in the user documentation), however in this post I'll be assuming that we're talking about hundreds/thousands of missing links and that the customer has provided us with a CSV file that contains the topology information they require.

So the goal is to demonstrate how a small custom Collector might be written that can enrich data discovered via other methods (e.g. SNMP).

Some key points about the Collector developed in this example:

- Small Perl Collector
- Driven by a single CSV file (the additional topology in our case)
- No device definition (e.g. type, inventory) as we assume this is done via the existing discovery
- Selectable topology levels (e.g. Layer1, Layer2, Layer3, Microwave and RAN)
- Tested in V4.1+ (it won't work as is in 3.9)


### The Collector (full working code presented)

This is the sole CSV file that we'll be loading in this example to provide the additional link information is as follows:

#### Topology.csv:

```
-- Topology.csv
--
-- Rows;
--   TopologyType|Source|SourceIfIndex|SourceInterfaceId|Destination|DestinationIfIndex|DestinationInterfaceId|Unidirectional|extraInfo
--
-- Where;
--   TopologyType is either Layer1, Layer2, Layer3, Microwave or RAN
--   Note: At time of writing the Agent/Stitchers force the following restrictions;
--     Layer 1 links use interfaceId rather than ifIndex, and support chassis-chassis links (i.e. no interfaceId)
--     Layer 2 & 3 links require either ifIndex or interfaceId (i.e. don't support chassis-to-chassis links)
--   Source & Destination can be an IP or device name. The discovery stitchers will use this name/address so it
--     should match the value discovered via the other discovery source (e.g. SNMP)
--   Unidirectional and extraInfo are optional
```

```
--
Layer2|DeviceA.ibm.com|10||172.20.1.101|11||
Layer1|DeviceA.ibm.com|||172.20.1.2|||
```

 **A note about device naming:** When using a Collector in a multiple data source scenario like this (i.e. device data coming from, say, SNMP and additional data coming from a Collector for that same device) we need to make sure that our Collector refers to the device in such a way that the discovery stitchers will be able to resolve and merge the data for that device from all sources. Most of the time this means we just need to use a consistent name or common IP address for that device across the data sources. Advanced: If the data sources don't provide a consistent name/IP then there is a configurable custom mapping stitcher (CustomerManagedEntityMapping.stch) that can add advanced resolution based on custom tags.

We'll use the XSV module to load the CSV data and so we need an XSV .cfg and .drv to describe how the CSV data is to be parsed which we'll define as follows;

#### topology.cfg:

```
# topology.cfg
(
  driver => { file => './topology.drv' },
  file => { TopologyData => './Topology.csv' },
  lineMatch => { TopologyData => '.*', },
  delimiter => { TopologyData => '|', }
)
```

#### topology.drv:

```
# topology.drv
(
  'type' => 'columnMap',

  'mappingGroupDescriptions' =>
  {
    'TopologyData' => { DelimitedBy => '|', UseColumns => '.*', },
  },

  'mappingGroupDefinitions' =>
  {
    'TopologyData' =>
    {
      1 => { Name => 'TopologyType', MapsTo => 'TopologyType' },
      2 => { Name => 'Source', MapsTo => 'Source' },
      3 => { Name => 'SourceIfIndex', MapsTo => 'SourceIfIndex' },
      4 => { Name => 'SourceInterfaceId', MapsTo => 'SourceInterfaceId' },
      5 => { Name => 'Destination', MapsTo => 'Destination' },
      6 => { Name => 'DestinationIfIndex', MapsTo => 'DestinationIfIndex' },
      7 => { Name => 'DestinationInterfaceId', MapsTo => 'DestinationInterfaceId' },
      8 => { Name => 'Unidirectional', MapsTo => 'Unidirectional' },
      9 => { Name => 'extraInfo', MapsTo => 'extraInfo' },
    },
  },
)
```

Now to write the Collector to load this CSV data, convert it and make it available to the discovery...

We'll need a Collector config file (again .cfg) to state which CSV .cfg to load and which port to listen to Network Manager on;

#### CollectorConfig.cfg:

```
# CollectorConfig.cfg
(
  General =>
  {
    Listen => 8087, # use this as the seed in DiscoCollectorFinderSeeds.cfg
  },
)
```

```

DataSource =>
{
    CsvCfg => 'topology.cfg',      # .cfg defining how csv is parsed
}
)

```

For Perl Collectors we usually have an initial script, main.pl usually, that processes the Collector config, defines the supported XML-RPC calls (the implementations of which usually just call the standard Collector Library implementations), sets up the XML-RPC server, and creates an instance of the main Collector class to handle the Collector specific processing.

Our main.pl will be as follows;

#### main.pl:

```

use lib "../lib/";          # path to perl collector library

use TopologyCsvCollector;   # The module that holds our collector class
use XMLRPC::Transport::HTTP; # For our XML-RPC server
use Getopt::Long;          # For command line options

use sigtrap 'handler' => \&CollectorExit, 'normal-signals', 'error-signals';

my $g_Collector;           # The module that takes care of data load/conversion
my $g_XmlRpcServer;       # The Collector-to-Network Manager XML-RPC server

print( "Starting collector...\n" );
sub Main();
Main();
print( "...Ending collector\n" );

sub Main()
{
    my @availableRpcMethods = (
        'UpdateData', 'GetInfo', 'GetDeviceList',
        'GetDeviceInfo', 'GetAssociatedAddresses',
        'GetConnections', 'GetLayer1Connections',
        'GetLayer2Connections', 'GetLayer3Connections'
    );
    # Get config file (-cfg <file>) from command line and load it.
    # It must have the source CSV cfg and listen port defined.
    #
    my %cmdLineConfig;
    GetOptions( "cfg=s" => \%cmdLineConfig{CfgName} );
    my $config = Collector::Collector::LoadCfgFile( $cmdLineConfig{CfgName} );
    if( ($config->{DataSource}->{CsvCfg}) and ($config->{General}->{Listen}) )
    {
        # Pretty much all Collector just support a single data source (1)
        # but we still have to specify it
        #
        my @sourceInfo;
        my %defaultSourceInfo = ( Id=>1, Descr=>"Primary Data Source" );
        push( @sourceInfo, \%defaultSourceInfo );

        # Create & initialise the collector and XML-RPC server, then just listen...
        #
        $g_Collector = TopologyCsvCollector->new( \@sourceInfo );
        if( $g_Collector )
        {
            if( $g_Collector->Initialise($config->{DataSource}) )
            {
                $g_XmlRpcServer = XMLRPC::Transport::HTTP::Daemon->new(
                    Listen=>40,

```

```
                ReuseAddr => 1,
                LocalPort=>${config->{General}->{Listen} } );
    if( $g_XmlRpcServer )
    {
        $g_XmlRpcServer->dispatch_to( @availableRpcMethods );
        print( "Listening for RPC calls...\n" );

        $g_XmlRpcServer->handle(); # listen for RPC calls. never returns
    }
}

sub CollectorExit
{
    $g_XmlRpcServer->close(); # close so can re-use socket on restart
    exit 1;
}

##
#####
# Supported XML-RPC Methods: mostly redirected to base class implementation.
# These are the calls the discovery can make to our Collector to retrieve data.
#####
##

sub UpdateData()
{
    shift; # drop self
    return $g_Collector->UpdateData( @_ );
}

sub GetInfo()
{
    return $g_Collector->StdGetInfo();
}

sub GetDeviceList()
{
    shift; # drop self
    return $g_Collector->StdGetDeviceList( @_ );
}

sub GetDeviceInfo()
{
    shift; # drop self
    return $g_Collector->StdGetDeviceInfo( @_ );
}

sub GetAssociatedAddresses()
{
    shift; # drop self
    return $g_Collector->StdGetAssociatedAddresses( @_ );
}

sub GetLayer3Connections()
{
    shift; # drop self
    return $g_Collector->StdGetLayer3Connections( @_ );
}

sub GetLayer2Connections()
```

```

{
    shift; # drop self
    return $g_Collector->StdGetLayer2Connections( @_ );
}

sub GetLayer1Connections()
{
    shift; # drop self
    return $g_Collector->StdGetLayer1Connections( @_ );
}

sub GetConnections()
{
    shift; # drop self
    return $g_Collector->StdGetConnections( @_ );
}

```

Finally we write the class that represents the Collector specific code (inherits from the Perl Collector Library's Collector::Collector base class);

#### TopologyCsvCollector.pm:

```

package TopologyCsvCollector;

use lib "../lib/";          # path to perl collector library

use Collector::Collector;  # collector base class offering common functionality
use Collector::Store;     # used to store collector data
use NCP::XSV;            # provides CSV loading functionality

@ISA = qw(Collector::Collector);

sub new()
{
    my $className = $_[0];
    my $sourceInfo = $_[1];

    my $store = Collector::Store->new();

    $store->SetupDataSources( $sourceInfo );

    $store->Initialise( "TopologyCsv", "Example topology data Collector");

    my $self = Collector::Collector::new( $store );

    $self->{Store} = $store;    # We hold a reference in this object too
    $self->{DataSource} = {};

    bless( $self );
    return $self;
}

sub Initialise()
{
    my $self = $_[0];
    my $configuration = $_[1];

    Collector::Collector::Initialise( $self, 0 );
    $self->{DataSource} = $configuration;
    $self->{CsvLoader} = NCP::XSV->new( $self->{DataSource}->{CsvCfg} )
        or die "Error whilst creating new XSV handler: $_[1]";
}

```

```

    return 1;
}

sub LoadTopology
{
    my $self = $_[0];      # Ref to an instance of this class
    my $store = $_[1];    # Ref to a Collector::Store. Updated with loaded data
    my $sourceId = $_[2]; # usually 1 as most collectors only have 1 data source

    my @links = $self->{CsvLoader}->load( "TopologyData", 1 );

    # Get rid of any empty fields i.e. where user hasn't specified a value (X,,Z)
    foreach my $fields (@links)
    {
        foreach my $field (keys %$fields)
        {
            if( $fields->{$field} eq '' )
            {
                delete( $fields->{$field} );
            }
        }
    }

    foreach my $link (@links)
    {
        if( exists $link->{TopologyType} and exists $link->{Source} and exists $link->{Destination} )
        {
            # Create a device entry if one not already present
            my $deviceId = $link->{Source};
            if( $deviceId and not $store->GetDeviceData($sourceId,$deviceId) )
            {
                my %deviceData;
                if( $deviceId =~ m/^[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}/ )
                {
                    $deviceData{ManagementIpAddress} = $deviceId;
                }
                else
                {
                    $deviceData{BaseName} = $deviceId;
                }
                $store->AddDeviceData( $sourceId, $deviceId, \%deviceData );
            }

            # Adjust the field names to match those expected by Collector library
            $link->{SourceName} = $link->{Source};
            delete $link->{Source};
            $link->{DestinationIP} = $link->{Destination};
            delete $link->{Destination};
            my $topologyType = $link->{TopologyType};
            delete $link->{TopologyType};
            # Add the link
            $store->AddConnection( $sourceId, $deviceId, $topologyType, $link );
        }
    }
}

# Overridden RPC Method: UpdateData() - load the topology data, delete the old
sub UpdateData()
{
    my $self = $_[0];
    my $dataSourceId = $_[1];

```

```

my $store = $self->{Store};
$store->DeleteData( $dataSourceId );

$self->LoadTopology( $store, $dataSourceId );

return $self->StdUpdateData();
}

return 1;

```

And that's it, we're ready to run.

## Testing The Collector

However first it is wise to test the Collector with `ncp_query_collector.pl`, so run the Collector with;

```
ncp_perl main.pl -cfg CollectorConfig.cfg&
```

... and then `ncp_query_collector.pl` with (stating the same port as configured in the Collector's configuration file (`CollectorConfig.cfg` in this case));

```
ncp_perl ncp_query_collector.pl -port 8087
```

Then 'ping' the Collector with `GetInfo()`...

```

>GetInfo()
<collectorInfo>
  <name>TopologyCsv</name>
  <datasources>
    <datasource>
      <id>1</id>
      <descr>Primary Data Source</descr>
    </datasource>
  </datasources>
  <descr>Example topology data Collector</descr>
</collectorInfo>

```

Our Collector appears to be up, so make it load the CSV data (this call is usually issues by the Collector Finder during a discovery);

```

>UpdateData(1,0,'','')
<updateResponse></updateResponse>

```

Now we can check the data (i.e. issue the calls that the Collector Agents would usually make);

```

>GetDeviceList(1,0,'','')
<deviceList>
  <device>
    <id>DeviceA.ibm.com</id>
    <ip>DeviceA.ibm.com</ip>
    <protocol>4</protocol>
  </device>
</deviceList>

```

Note that only the source devices are returned; that's fine as in this scenario the main device data is being discovered via another source. We're only providing this skeleton source device so that we have something to 'hang' our data on.

```
>GetLayer2Connections(1,DeviceA.ibm.com)
<layer2Connections>
  <connection>
    <dest>172.20.1.101</dest>
    <destIfIndex>11</destIfIndex>
    <src>DeviceA.ibm.com</src>
    <srcIfIndex>10</srcIfIndex>
  </connection>
</layer2Connections>
>GetLayer1Connections(1,DeviceA.ibm.com)
<layer1Connections>
  <connection>
    <dest>172.20.1.2</dest>
    <src>DeviceA.ibm.com</src>
  </connection>
</layer1Connections>
```

So our Layer 1 and 2 links look good and we can now try an actual discovery, i.e. add the seed (port) of this Collector to the discovery configuration for a domain that already discovers these devices (e.g. via SNMP) but which lacks the connections we've defined.

#### **DiscoCollectorFinderSeeds.ACME.cfg;**

```
insert into collectorFinder.collectorRules ( m_Port) values ( 8087 );
```

Run a discovery. This should now produce the topology that we wanted;



## Comments

*There are no comments.*