

# IBM Tivoli Network Manager 3.8

## Creating custom network partition views using custom tables



Welcome to the IBM Education Assistant module for IBM Tivoli® Network Manager 3.8. This module is creating custom network partition views using custom tables.

## Objectives

After you complete this module, you should be able to:

- Create a custom table in the Network and Connectivity Information Model (NCIM) database
- Populate the custom table with information
- Make fields from the custom table available to use in network partition view filters
- Create a custom network partition view using newly added discovery data and fields
- Copy the newly created view and make it available to other users and groups in IBM Tivoli Network Manager

After you complete this module, you should be able to:

Create a custom table in the Network and Connectivity Information Model (NCIM) database

Populate the custom table with information

Make fields from the custom table available to use in network partition view filters

Create a custom network partition view using newly added discovery data and fields

Copy the newly created view and make it available to other users and groups in IBM Tivoli Network Manager

## ***Creating a custom table***

Customers often find it useful to create custom network partition views. They want these views to be based upon business-related information instead of just topology information. To use custom information to enhance topology views, the custom data must be added to the Network and Connectivity Information Model (NCIM) database. Because software updates can potentially alter the structure of default database tables, custom data should be put into a custom table.

## General guidelines for new tables

- Avoid modifying default NCIM database tables
  - The default NCIM schema might be altered by subsequent fix packs
- Create and use separate tables instead
  - Example: Add **Location** column for each chassis record using a custom MySQL table
  - Make sure the table is of type **InnoDB**; this reduces table size
  - Put **CONSTRAINTS** on entity; this helps to reduce database errors
  - Product-specific extension tables or custom tables typically have a foreign key relationship with the core-model entity table
- Define fields to include in the custom table
- Define relationships to other NCIM tables

For the reason just mentioned, avoid modifying default NCIM database tables. Instead, create and use a separate table. For example, if you wanted to add a **Location** field for each chassis entity, you could create a custom chassis table. You could put the node name, IP address, and location for each chassis entity in that table. Information from that table could then be used to create location-specific network partition views.

Creating a new table involves more than just defining the fields and data types. Some types of tables are more efficient in their use of memory than others. The InnoDB table type is used in the NCIM database because it reduces table size. It is also important to ensure that data in one table is properly related to other tables in a relational database. By putting constraints on data in a table, you can properly maintain the relationships between data.

A **constraint** specifies the relationships between values in the new table and values in another table. The constraints determine whether data is deleted from one table if the corresponding key data is deleted from the other table.

In Tivoli Network Manager, most newly created tables will have a corresponding foreign key value in the **entity** table. The constraint will indicate that if the entity data is deleted from the **entity** table, then all corresponding records in other tables will be deleted. If you delete a chassis from the **entity** table, a cascade deletion will occur. This means that all interface records associated with that chassis entity space will be deleted. The deletion will continue to cascade and delete all logical interfaces associated with the deleted physical interfaces. Proper constraints and foreign key relationships ensure that these deletions work properly and prevent the database from retaining inaccurate data.

## Creating a custom table

Easiest to create an SQL script

The first field of any new table should always be named **entityId** and be a foreign key to the **entity** table

### Specifications:

- ENGINE=InnoDB reduces database size
- CHARSET specifies character set
- COLLATE validates data against the specified standard
- CONSTRAINT defines primary and foreign key relationships

You can log on to the NCIM database and create custom tables with a series of manual commands. However, the easiest way to create a custom table is to create an SQL script. The first field of any NCIM table should be the **entityID** field. This key field uniquely identifies each entity in the NCIM database. The **entityID** field in a custom table should have a foreign key relationship to the same **entityID** in the entity table. This establishes the relationship necessary to properly cascade delete actions within the database. You can also define different character sets for the database to accommodate user language preferences.

## Defining a custom table

```
create table ncim.customChassis
(entityId      integer not null,
 Location      varchar(64),
 Department    varchar(64),
 Continent     varchar(64),
 Country       varchar(64),
 City          varchar(64),
 DeviceType    varchar(16),
 ShortName     varchar(16),
```

- The **varchar** data type has a variable field length. The numeric designation specifies a maximum field length
- The **char** data type has a fixed field length designated by the number

An SQL script to create a new table begins by defining the table name, field names, and data types. The example shown here creates a **customChassis** table in the NCIM database. It then creates **Location**, **Department**, **Continent**, **Country**, **City**, and **Devicetype** fields in that table. You will notice that the data types for string variables are defined as **varchar** fields. A **varchar** field only uses as much memory as is needed to store a string variable. The number in parentheses next to the **varchar** designation indicates a maximum field length. In the example shown, the location field has a maximum field length of 64 characters. However, if the device was located in Dallas, only the memory required to store six characters would be used. If you used a **char** data type, the number would designate a fixed field length. With a **char(64)** data type, the system would always allocate memory to store 64 characters of data. It would do this even if the location only required six characters. For this reason it is always more efficient to use a **varchar** character type. For more information on the syntax shown here, consult the *IBM Tivoli Network Manager Topology Database Reference* manual.

## Defining a custom table (continued)

```
CONSTRAINT customChassis_pk PRIMARY KEY (entityId),  
CONSTRAINT customChassis_entity_fk FOREIGN KEY (entityId) REFERENCES  
entity(entityId) ON DELETE CASCADE)  
ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
```

The first constraint seen in this example defines the **entityID** field as the primary key of the **customChassis** table. The second constraint creates a relationship between **entityID** fields in the **customChassis** table and the **entity** table. This second constraint also instructs the NCIM database to delete any records in any **customChassis** table that correspond to a deleted entity in the entity table.

The **ENGINE** designation uses the efficient **InnoDB** table type. The standard English character set is selected. Then the **COLLATE** statement instructs the database to ensure that all data entered into the database conforms to the specified character set.

## Create and verify the new table

```

▪ Run SQL script file to create table:
  mysql -u root -p < scriptName.sql

▪ Verify table creation:
  mysql> connect ncim;

  mysql> show tables;
+-----+
| Tables_in_ncim |
+-----+
| chassis        |
| classMembers  |
| collects       |
| connects       |
| contains       |
| customChassis |
+-----+

```

After the SQL script has been created and saved, it can be used to create the new table. In the example shown here, the user logs in to the MySQL server as the root user. The `-p` command line option will prompt the user for a password. The less-than sign instructs MySQL to process the instructions in the specified SQL file. If you do not have root user privileges for your database, enlist the help of your database administrator. The NCIM database is not limited to MySQL. It can also be installed to run on DB2® or Oracle. Future versions of IBM Tivoli Network Manager will include support for the Informix® database as well. Differences in the database platform can affect the syntax requirements of your SQL script. You will need to know the correct syntax for your database platform.

Table creation will happen quickly. To verify that the table has been created, you can log on to the MySQL server. Then connect to the NCIM database and use the **show tables** command. All commands terminate with a semicolon character. A list of tables will be shown and will include your newly defined table if the syntax of your SQL script was correct.



## Section

***Populate the custom table with data***

After the custom table has been completed, you must configure IBM Tivoli Network Manager to populate the custom table with data. Business-specific data can be added to the discovery through the use of custom discovery stitchers. Examples of these custom discovery stitchers can be seen in other IBM Education Assistant modules for IBM Tivoli Network Manager. The examples in this module assume that a custom discovery stitcher has already been created and is running. This discovery stitcher assigns Location, Department, Continent, Country, and Device type to each chassis entity in the network.

## Make MODEL populate the custom table

- Edit the **dbModel.entityMap** section of **dbEntityDetails.domainName.cfg**

```
insert into dbModel.entityMap
(EntityFilter, TableName, FieldMap)values
( "(EntityType = 1 or EntityType = 8) and ExtraInfo->
m_Location is not null",
"customChassis",
{ entityId = "eval(int, '&ObjectId')",
Location = "eval(text, '&ExtraInfo->m_Location')",
Department = "eval(text, '&ExtraInfo-> m_Department')"} );
```
- The custom database will be populated with data by a custom discovery stitcher

A custom discovery stitcher can add business-specific information to the default **master.entityByName** table. To use this new data for topology visualization (TopoViz), the data must be sent to the NCIM database.

To do this, edit the **DbEntityDetails.cfg** file. Never modify the default **DbEntityDetails.cfg** file. As with all configuration files, make a domain-specific copy of the file and then edit that file. To make a domain-specific file, copy the file and insert the domain name and another period before the **.cfg** extension. Doing this ensures that you will always have a working original configuration file if you ever want to use it.

In the example shown here, a custom database called **customChassis** has been added to the NCIM database. The insert statement adds the **Location** and **Department** fields to each chassis record in the **dbModel.entityMap** table. More fields can be added in the same manner. After modifying this file, you must restart the MODEL service so that it will read the changes in this configuration file.

## Make MODEL populate the custom table (continued)

- Stop the MODEL service:
  - `pkill ncp_model`
- Restart the MODEL service
  - The `ncp_model` process will reread the configuration file
- After MODEL restarts, it will update the NCIM database
- To force MODEL to update NCIM, use a query:
  - `ncp_oql -domain domainName -username admin -password 'password' -service MODEL -query "update master.entityByName set ActionType=1;"`

After you have edited the **DbEntityDetails.cfg** file, stop and restart the MODEL service. To stop the MODEL service, you can use the **pkill ncp\_model** command. Because this process runs under IBM Tivoli Network Manager process control (`ncp_ctrl`), the service will be restarted automatically. When MODEL restarts, it will update the NCIM database. If you do not see your custom table populated soon after restarting the MODEL service, you can manually force this update. To do this, use an SQL query similar to the one shown, substituting your domain name and password values.

## ***View custom table data in network partition views***

You now have your business-specific data in your custom table in the NCIM database. By configuring an Extensible Markup Language (XML) file, you can enable the custom data to be used in network partition views.

## TopoViz GUI modification overview

- Edit the **ncimMetaData.xml** file
  - Found in the **\$TIPHOME/profiles/TIPProfile/etc/tnm** directory
  - Specify new fields to include in TIP GUI
- Stop Tivoli Integrated Portal
  - **itnm\_stop tip**
- Restart Tivoli Integrated Portal
  - **itnm\_start tip**
- New fields become available for use in creating filtered network partition views

Configuring the TopoViz GUI to use fields from the custom table involves editing one file and then stopping and restarting Tivoli Integrated Portal.

Begin by editing the **ncimMetaData.xml** file located in the **\$TIPHOME/profiles/TIPProfile/etc/tnm** directory. The **\$TIPHOME** environment variable usually corresponds to the **/opt/IBM/tivoli/tip** directory.

The process of stopping or starting Tivoli Integrated Portal is most easily accomplished using the **itnm\_stop tip** and **itnm\_start tip** commands. After you restart Tivoli Integrated Portal, the new fields specified in the **ncimMetaData.xml** file become available to the GUI.

## Modify TopoViz to display new fields

- Edit **ncimMetaData.xml** on the Tivoli Integrated Portal server to configure TopoViz

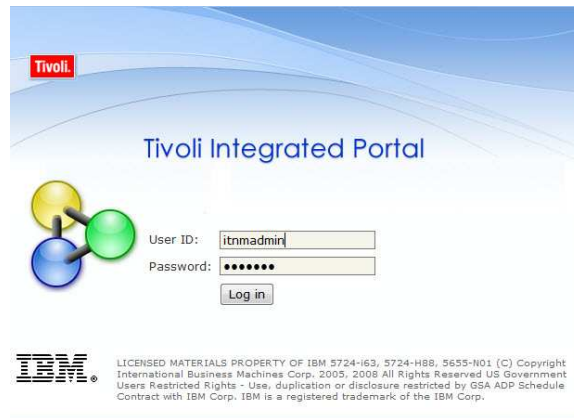
```
<!-- END ... EXAMPLE CUSTOMER EXTENSION -->
  <entityMetaData table="customChassis" manager="AllManagers"
entitySearch="true">
    <dataField dataType="str" column="Location" />
    <dataField dataType="str" column="Department" />
    <dataField dataType="str" column="Continent" />
    <dataField dataType="str" column="Country" />
    <dataField dataType="str" column="City" />
    <dataField dataType="str" column="DeviceType" />
    <dataField dataType="str" column="ShortName" />
  </entityMetaData>
</ncimMetaData>
```

Make your modifications to the **ncimMetaData.xml** file at the end of the file, after the EXAMPLE CUSTOMER EXTENSION section marker and before the closing **/ncimMetaData** keyword. The example shown here begins by specifying the new **customChassis** table, and enables searches on data in that table. A list of data fields follows. Each data field is specified as containing text string values, as indicated by the “**str**” setting for the data type. You can also add integer fields to the display in a similar manner.

After these modifications are complete, save the file and then stop and restart Tivoli Integrated Portal as previously explained.

## Log on to Tivoli Integrated Portal

- Open the Uniform Resource Locator (URL) for Tivoli Integrated Portal  
– **https://<host\_IP\_address>:16316/ibm/console**
- Log on as a Tivoli Integrated Portal user with rights to use the MIB graphing tool



After Tivoli Integrated Portal has been restarted, log on to the graphical user interface. To do this, open your compatible browser and enter the uniform resource locator (URL) of the Tivoli Integrated Portal server. The default installation uses https on port 16316. Make certain that you log on as a user with sufficient rights to create and view network partition views. The administrator for Tivoli Network Manager or Tivoli Integrated Portal has the authority to define these permissions.

## Creating a new view

- Select **Network Views** in the Tivoli Network Manager GUI
- Create a new network view
  - Select the user or group for the view
  - Click the **New View** icon



From the menu at the left of your screen, click **Network Views** under the **Network Availability** category. From the pull-down menu, select the user view or group view in which you want to place the new view. Click the new view icon.



## Creating a new view (continued)

- Enter a name for the new view.
- Select **Filtered** as the view type.
- Click the **Filter** tab.

The screenshot shows a dialog box with two tabs: 'General' and 'Filter'. The 'General' tab is active. At the top, there is a label '\*required field'. Below it, the 'Name' field contains 'North\_America'. The 'Parent' field contains 'NONE'. The 'Type' dropdown menu is set to 'Filtered'. The 'Layout' dropdown menu is set to 'Symmetric'. There are three empty text input fields for 'Map Icon', 'Tree Icon', and 'Background image'. The 'Background Style' dropdown menu is set to 'Centered'. At the bottom, there is a help icon (?), an 'OK' button, and a 'Cancel' button.

On the **General** tab of the GUI, enter a name for the view. Select **Filtered** as the view type. Then click the **Filter** tab.

## Creating a new view (continued)

Domain: ITNM38GO ▾

- Select the correct domain name
- Select the new custom field you want from the **Field** pull-down menu, the comparator, and the value for your filter
  - You can combine multiple fields using the All (Boolean AND) or Any (Boolean OR) values to create more specific filters

**Filter Builder**

Basic Advanced

All  Any

Field	Comparator	Value	Action
Continent ▾	like ▾	[Nn][Aa]	<input type="button" value="X"/> <input type="button" value="+"/>

18

Creating custom network partition views using custom tables

© 2010 IBM Corporation

On the Filter tab of the GUI, select the correct domain name. From the Field pull-down menu, select the new custom field you want to use. Select the appropriate comparator and specify the value on which you want to filter.

## Creating a new view (continued)

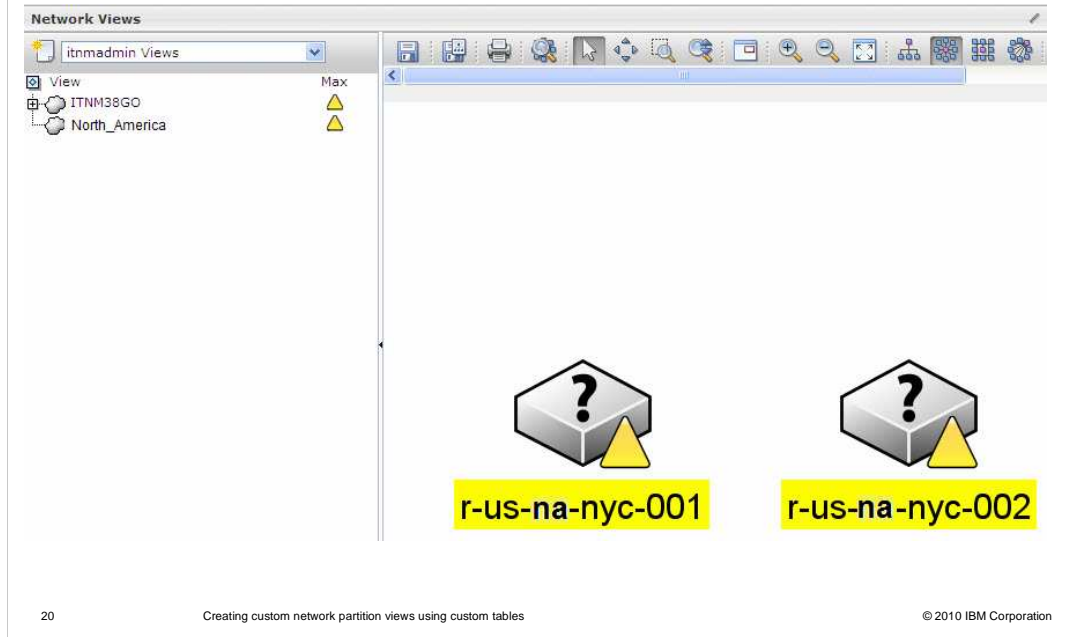
- Select whether to view **End Nodes**
- Select the level of **Connectivity** to view
- Click **OK**

End Nodes: Include ▼

Connectivity: Layer 2 ▼

If you want your network partition view to display end nodes, select **Include** as the **End Nodes** value. Specify whether you want layer 2, layer 3, or subnet connectivity to be shown. Click **OK** to complete configuration of this network partition view.

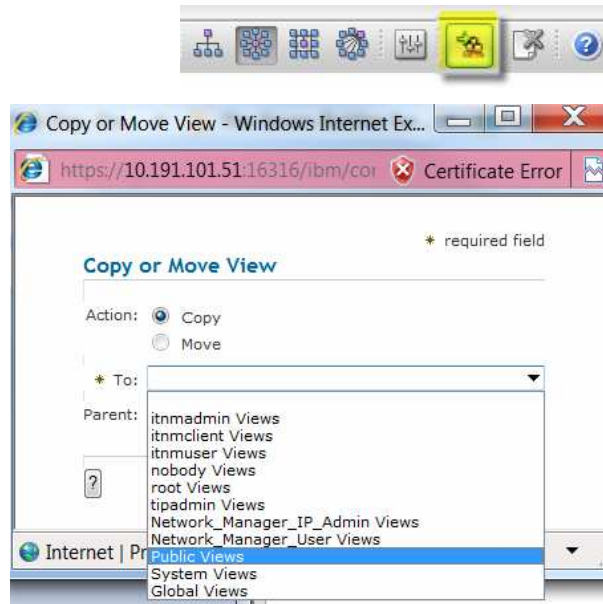
## Viewing the new network partition view



The new network partition view is now displayed in the Network Views GUI.

## Copying a network partition view

- Click the **Copy View** icon to copy the newly created view and make it available to other groups or users.
- Select the appropriate views to receive the new network partition view.



You can copy a newly created network partition view and make it available to other users or groups in IBM Tivoli Network Manager. To do this, click the **Copy View** icon. Select **Copy** or **Move** as the action. Select from a list of user and group views to receive the copy of the network partition view. Click **OK** to copy or move the view. Click the question mark if you need further explanation of these options.

## Summary

This module has shown you how to:

- Create a custom table in the Network and Connectivity Information Model (NCIM) database
- Populate the custom table with information
- Make fields from the custom table available to use in network partition view filters
- Create a custom network partition view using newly added discovery data and fields
- Copy the newly created view and make it available to other users and groups in IBM Tivoli Network Manager

This module has shown you how to:

Create a custom table in the Network and Connectivity Information Model (NCIM) database

Populate the custom table with information

Make fields from the custom table available to use in network partition view filters

Create a custom network partition view using newly added discovery data and fields

Copy the newly created view and make it available to other users and groups in IBM Tivoli Network Manager

## Resources

- For more information on creating network partition views, consult the *Tivoli Network Manager IP Edition Version 3 Release 8 Topology Visualization Guide*
- Use the question mark in any IBM Tivoli Network Manager screen to access context-sensitive online documentation for the product

For more information on creating network partition views, consult the *Tivoli Network Manager IP Edition Version 3 Release 8 Topology Visualization Guide*.

Also, use the question mark in any IBM Tivoli Network Manager screen to access context-sensitive online documentation for the product. A new browser window will open with product documentation specific to the context in which you clicked the question mark.



## Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

[mailto:iea@us.ibm.com?subject=Feedback\\_about\\_creatingcustomnetworkviewsanddbtables.ppt](mailto:iea@us.ibm.com?subject=Feedback_about_creatingcustomnetworkviewsanddbtables.ppt)

This module is also available in PDF format at: <.../creatingcustomnetworkviewsanddbtables.pdf>

You can help improve the quality of IBM Education Assistant content by providing feedback.





## Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, DB2, Informix, and Tivoli are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the Web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.

© Copyright International Business Machines Corporation 2010. All rights reserved.