Hewlett Packard Enterprise

HPE Network Node Manager i Software

Step-by-Step Guide to Using NNMi Import and Export Tools

White Paper

Contents

Introduction

This document describes the some common usage scenarios for using NNMi's import and export tools, provides some helpful examples, and offers some best practices for using the tools.

The import and exports commands are listed below:

- nnmconfigimport.ovpl
- nnmconfigexport.ovpl

Useful Scenarios

Use the NNMi import and export tools when you want to:

- Make substantial changes to the NNMi configuration and be able to revert back to a known good state, if needed.
- Move the configuration of a lab's test server to a production server.
- Move one production server configuration to another production server configuration (overwriting the current configuration)
- Merge changes from one NNMi management server to another NNMi management server.

Note
The NNMi import tool performs database merges, updates, and insertions.

Tip: Use NNMi's backup and restore tools (nnmbackup.ovpl and nnmrestore.ovpl) to perform migrations involving the overwriting of data (from one computer to another). If you create a full backup with nnmbackup.ovpl, this backup can be restored to another NNMi station (assuming it has the same version and patch level) or to the same server. Note that all of the data is restored including nodes, incidents, and configuration.

Tip: Use nnmnodedelete.ovpl and nnmtriminc idents.ovpl to remove nodes and incidents.

Best Practices

Note the following best practices when exporting and importing a configuration:

- See *Export/ Import Behavior and Dependencies* in the NNMi help to learn about the specific behaviors of the configuration item you are interested in. Especially note whether the import overwrites or merges the configuration. Also note the dependencies on other configurations.Make substantial changes to the NNMi configuration and be able to revert back to a known good state, if needed.
- Avoid conflicts in the Ordering field.
- Avoid name conflicts in Node Groups, Interface Groups, and their universally unique identifiers (UUIDs).
- Import the configuration files in the correct order based on dependencies.

Learning the Specific Export and Import Behaviors

Consider the three export and import behavior choices:

- Replaces all—replaces the entire existing configuration during the import.
- Incremental—updates all object instances with matching key identifiers during the import.
- Incremental (subset)—includes configuration changes that were made by one Author.

Configuration files can have dependencies. For example, the Node Group Map Settings configuration area has a dependency Node Groups.

Before proceeding to the following examples, refer to Export/Import Behavior and Dependencies in the NNMi help.

Incremental (Merge) Import Example

For this example, remember that Node Group have an "incremental" import behavior.

1. Create a Node Group called `Node Group 1`

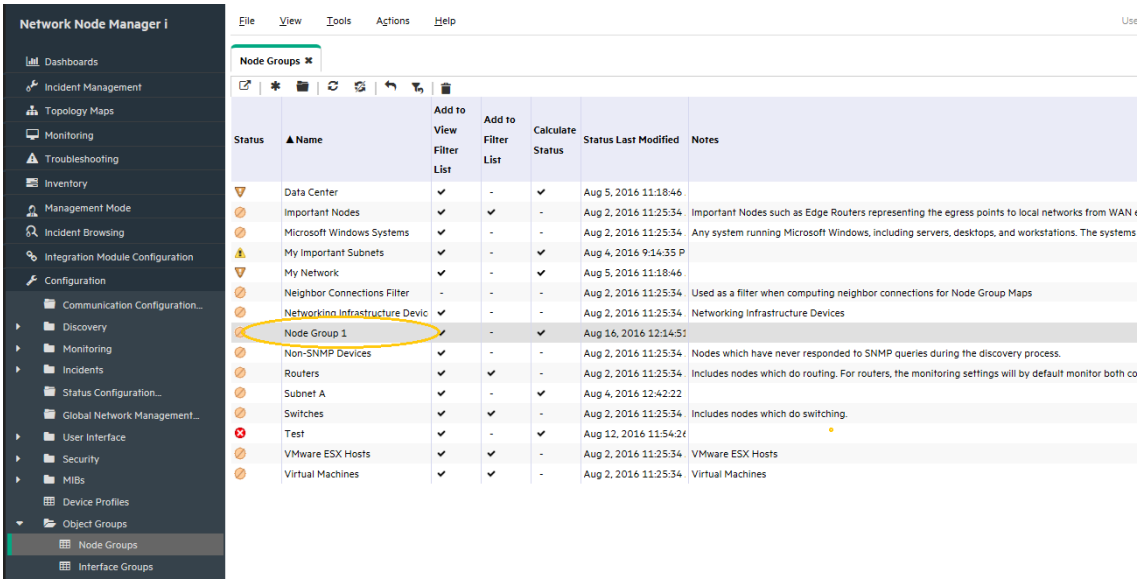

Figure 1: Configuartion Create Node Group 1

2. Export the configuration using the following command:

   `nnmconfigexport.ovpl -c nodegroup -f nodegroup.1.xml`

   You should see a message similar to the following:

   `Successfully exported /var/tmp/config/nodegroup.1.xml.`

3. Delete Node Group 1, and then create Node Group 2.

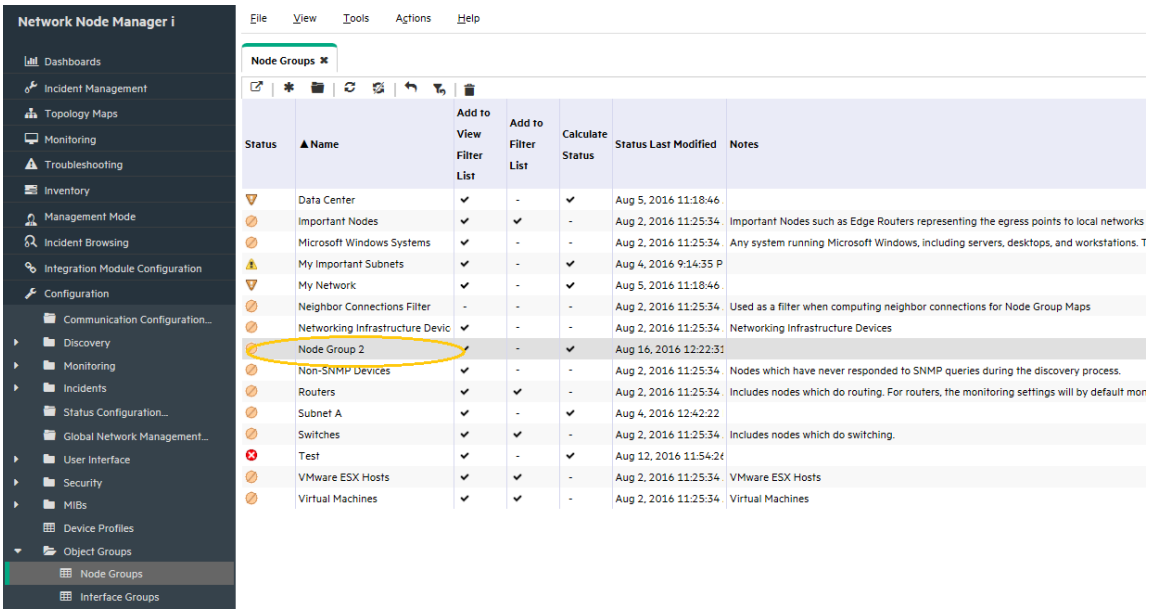

Figure 2: Configuration Create Node Group 2

4. Export the configuration again using the following command (so this export does not contain Node Group 1):

   `nnmconfigexport.ovpl -c nodegroup -f nodegroup.2.xml`

You should see a message similar to the following:

`Successfully exported /var/tmp/config/nodegroup.2.xml.`

5. Delete Node Group 2

   Now neither Node Group 1 nor 2 exists in the database.

6. Perform an import of Node Group 1 using the following command:

   `nnmconfigimport.ovpl -f nodegroup.1.xml`

   You should see a message similar to the following:

   `Successfully imported nodegroup.1.xml.`

7. Perform an import Node Group 2 using the following command:

   `nnmconfigimport.ovpl -f nodegroup.2.xml`

   You should see a message similar to the following:

   `Successfully imported nodegroup.2.xml.`

On importing the two configurations, the data is merged, as opposed to overwritten. The database contains both Node Group 1 and Node Group 2.

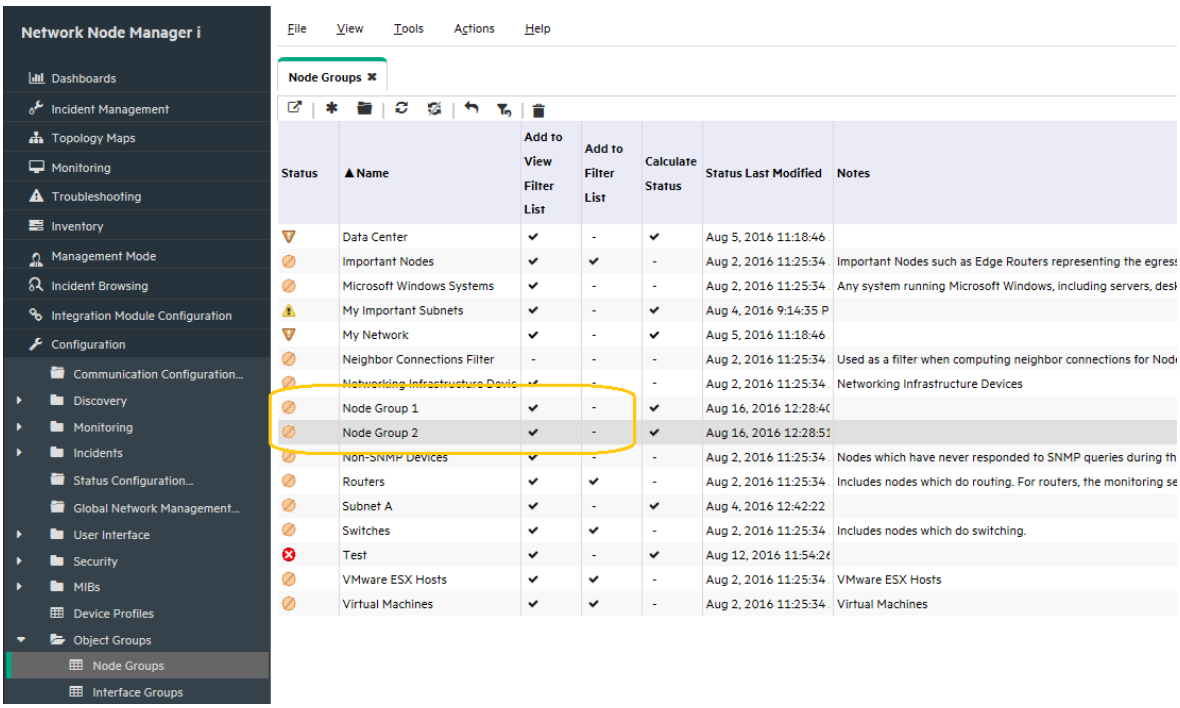

Figure 3: Configuration both Node Groups Displayed

"Replace All" Import Example

The Discovery configuration import uses the "Replaces all" import behavior. So instead of a merge, the import performs a replacement of data.

1. Create an Auto-Discovery rule called `discorange1`.

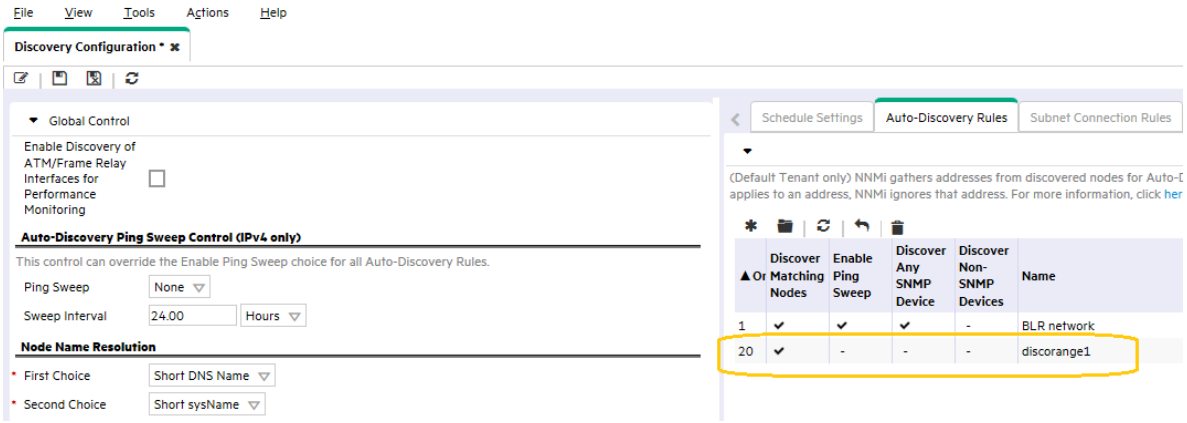

Figure 4: Discovery Configuration - Create discorange1 Auto-Discovery Rule

2. Perform an export of the `discorange1` configuration using the following command:

   `nnmconfigexport.ovpl -c disco -f disco.1.xml`

   You should see a message similar to the following:

   `Successfully exported /var/tmp/config/disco.1.xml.`

3. Delete the discorange1 Auto-Discovery Rule.
4. Create a new Auto-Discovery rule called `discorange2`.

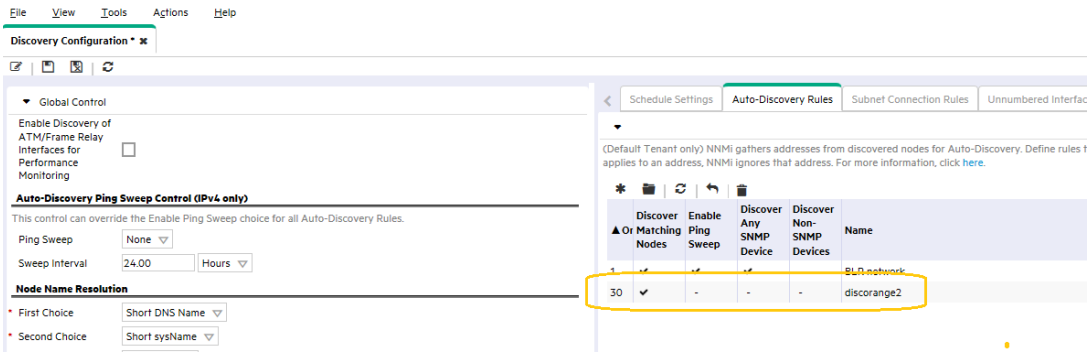

Figure 5: Discovery Configuration - Create discorange2 Auto-Discovery Rule

5. Perform an import of the `discorange1` configuration using the following command:

   `nnmconfigimport.ovpl -f disco.1.xml`

   You should see a message similar to the following:

   `Successfully imported disco.1.xml.`

   Even though there was not a conflict with `discorange2`, the `disco.1.xml` configuration overwrote the configuration.

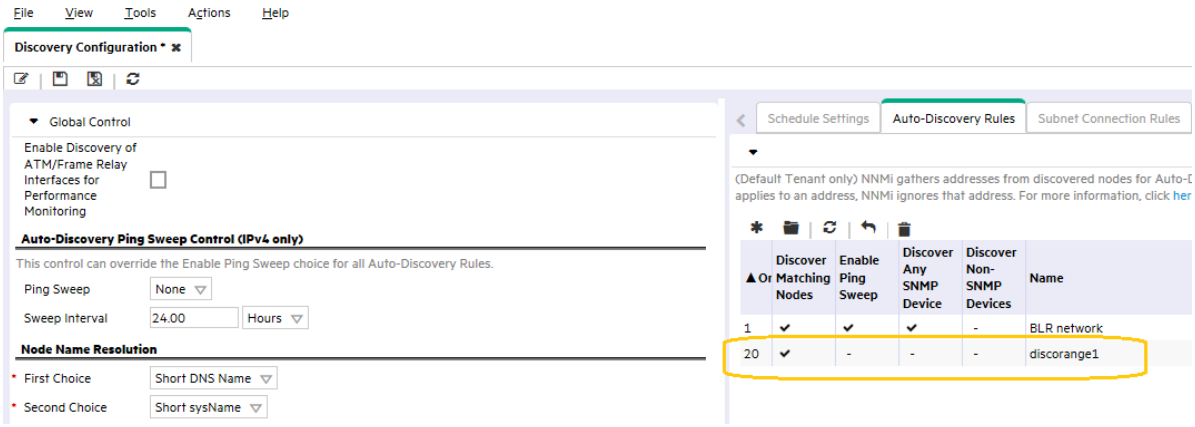

Figure 6: Discovery Configuration: disco.1.xml File Overwrote Configuration

Avoid Conflicts in the Ordering Fields

Many configurations, such as Monitoring Configuration, have an `Ordering` field.

Note
While the Monitoring Configuration import has a “Replaces all” behavior, it only replaces all if there are no conflicts in the `Ordering` field.

The following figure shows the default configuration of Monitoring Configuration:

For this example, start by exporting this default configuration so that you can revert back to it if necessary.

1. Perform an export of the `monitoring1` configuration using the following command:

   ```
   nnmconfigexport.ovpl -c monitoring -f monitoring.1.xml
   ```

   You should see a message similar to the following:

   ```
   Successfully exported /var/tmp/config/monitoring.1.xml.¥
   ```

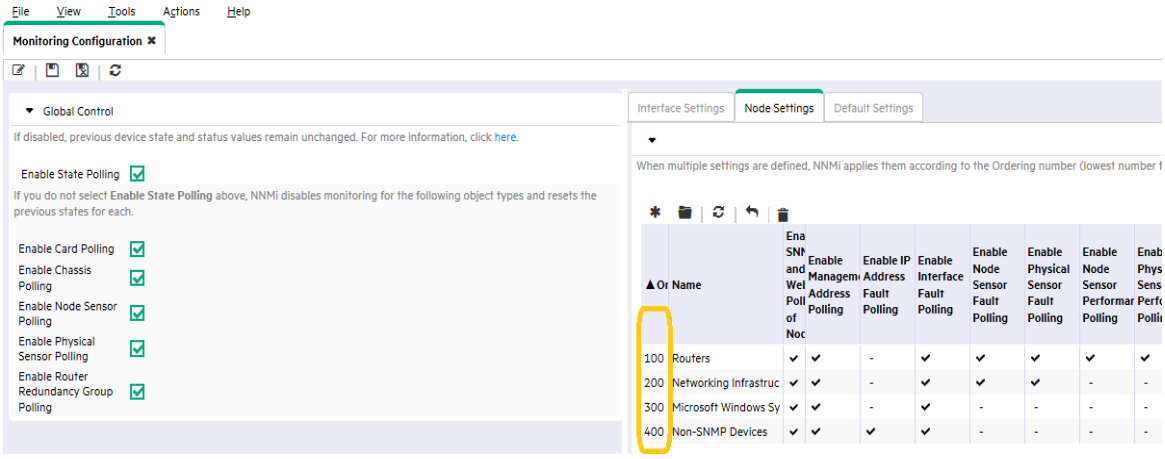

Figure 7: Monitoring Configuration - Default

2. Delete the Monitoring policy based on `Routers` (`Ordering 100`).

3. Create a Monitoring policy based on `Data Center` and reuse the `Ordering` value of `100`.

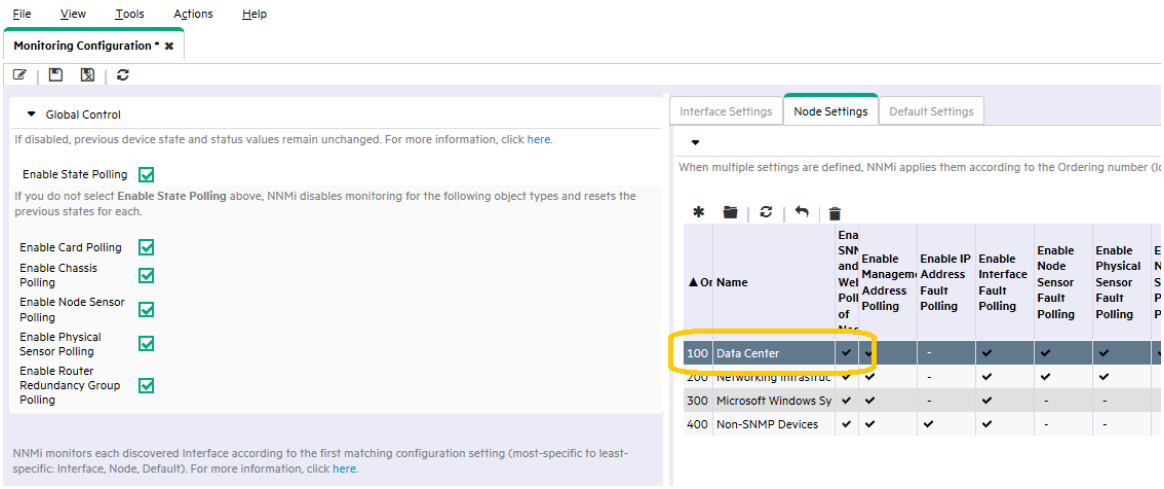

Figure 8: Monitoring Configuration: Replace Routers with Data Center

4. Assuming that you want to revert back to the original configuration, perform an import using the following command:

   `nnmconfigimport.ovpl -f monitoring.1.xml`

   You should see the following error message:

   `Configuration Import Error:`

   `Database constraint violation: Could not execute JDBC batch update`

The import fails because the original configuration had Routers at 100 and the new configuration has Data Center at 100. To resolve this problem, you need to remove the conflict, save the configuration, and then re-import. To remove the conflict, you can either change the Ordering number of the current configuration (change Data Center to 99) or you can completely remove the policy for Data Center if you no longer need it.

Tip: When you change `Data Center` to 99, you avoid the `Ordering` conflict but the `Data Center` policy is deleted when you import the previous Monitoring Configuration. This result is because the Monitoring Configuration import has a “Replaces all” behavior.

Avoid Name Conflicts in Node Groups, Interface Groups, and Assigned UUIDs

NNMi includes a default Node Group called `Routers` with an assigned UUID.

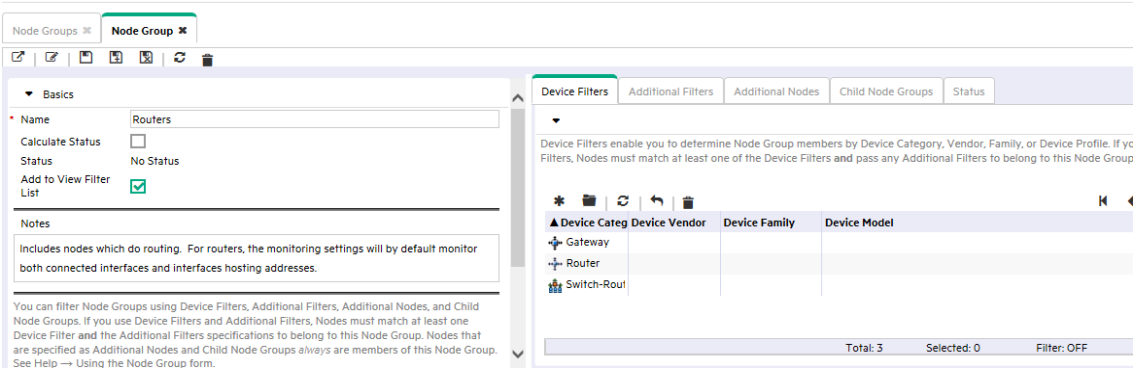

Figure 9: Node Group: Routers

Consider the following scenario. You perform the following steps:

1. Export the original configuration.

2. Delete the default Routers Node Group.
3. Create a new Node Group called Routers.

The newly created Routers Node Group has a different UUID than that of the original Routers Node Group. If you were to attempt to import the configuration that has the original definition of Routers in it, the import will fail because of the conflict between the Node Group name and the UUID.

Tip: Avoid naming customer-built Node Groups with the same names as default Node Groups, in case you later want to revert back to the original product settings.

A similar issue occurs when you try to move configurations from one NNMi management server to another.

Consider the following scenarios. You perform the following steps:

1. Create a Node Group called NetBotz on Server A (Server A assigns a UUID for this Node Group).
2. On Server B, create the same Node Group, NetBotz (Server B assigns a UUID to this Node Group that is different from the one created on Server A).
3. Import the configuration from Server A to Server B.

The import fails because of the conflict between the Node Group name and the UUID.

Tip: To avoid Node Group name and UUID conflicts as the one previously described, rather than creating the same configuration on two different servers, create the configuration on one server and move it to the other server using a configuration import.

Conflict Example

Tip: If you are planning to replace a configuration on a server, then first delete the configuration from that server.

Consider the following example:

1. Export the current default Monitoring Configuration.
2. Delete the Monitoring Configuration based on Routers (Ordering 100).
3. Create a Monitoring Configuration in Node Settings called My Favorite Routers, with Ordering set to 100.

   Assuming you want to revert back to the previous configuration, delete the entire configuration and then re-import:
4. From the workspace navigation panel, select the **Configuration** workspace, expand the **Monitoring** folder, and then click **Monitoring Configuration**.
5. Click the **Node Settings** tab.
6. Select all of the configurations and delete them.
7. Click the **Interface Settings** tab.
8. Select all of the configurations and delete them.

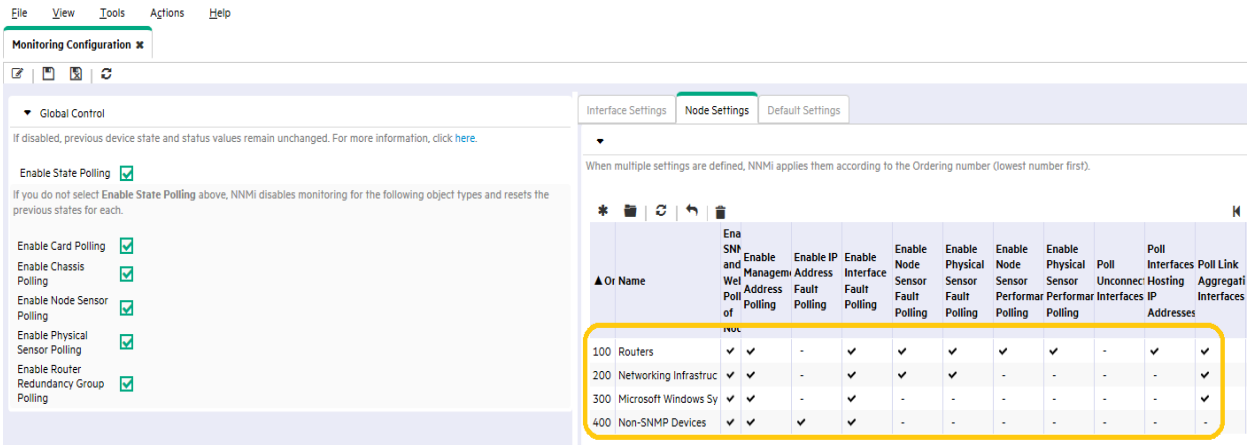

Figure 10: Monitoring Configuration: Delete Node Settings

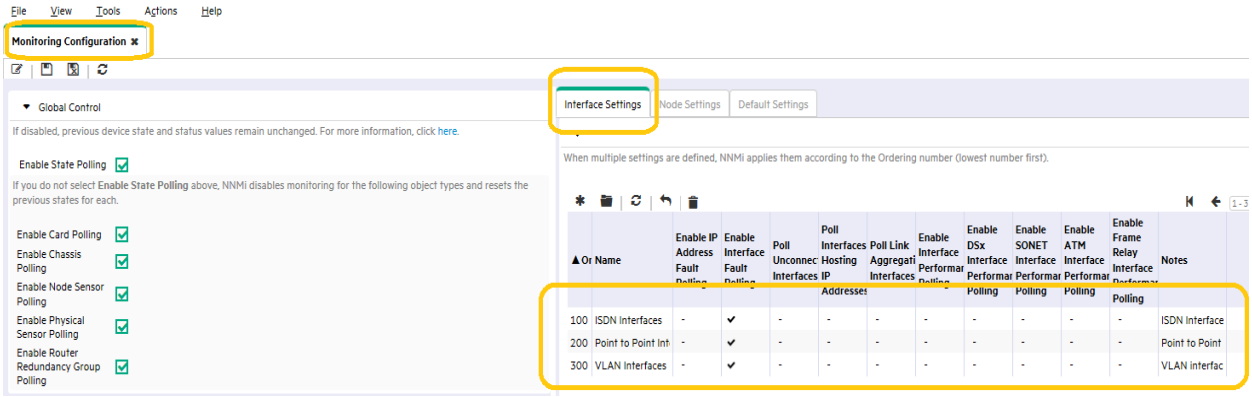

Figure 11: Monitoring Configuration: Delete Interface Settings

Tip: Disable **State Polling** when you delete the current Monitoring Configuration to avoid generating unexpected alarms (because no monitoring policies are in place) between the time that you delete the current configuration and the time the replacement configuration is loaded. This principle is true for other configurations, such as Node Groups, because of the references to Node Groups in Monitoring Configuration.

9. Disable State Polling by ensuring that the **Enable State Polling** check box is cleared.

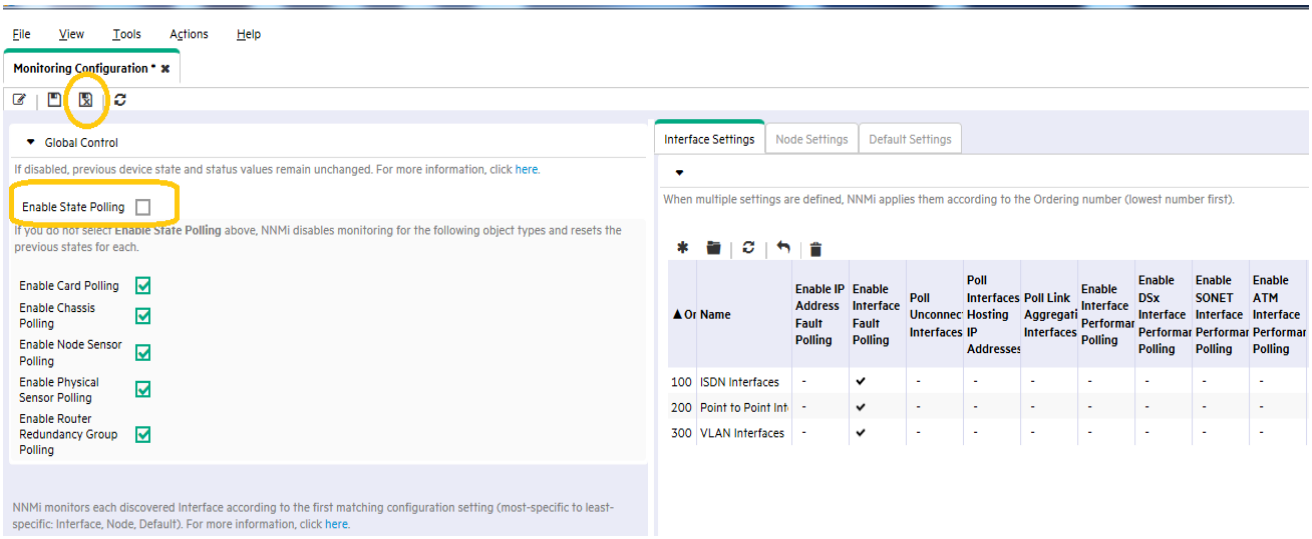

Figure 12: Monitoring Configuration - Disable State Polling

10. Click **Save** and **Close**
11. Perform an import using the following command:

    `nnmconfigimport.ovpl -f monitoring.xml`

    You should see the following message:

    `Successfully imported monitoring.xml.`

The `Routers` Node Setting has been restored with the `Ordering` at 100 and State Polling has been re-enabled.

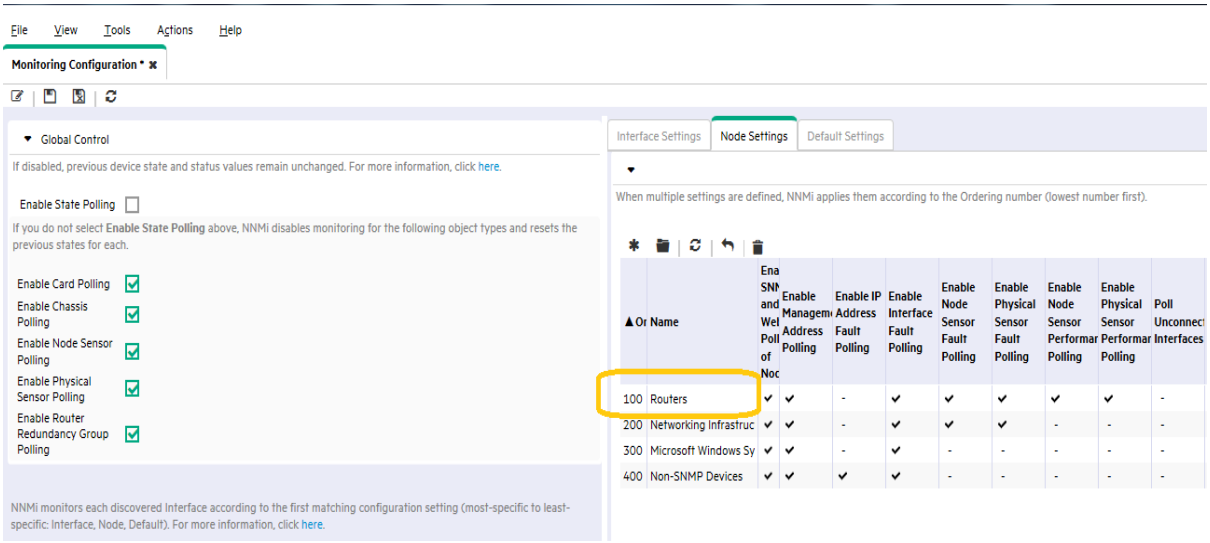

Figure 13: Monitoring Configuration - Routers Stored and State Polling Re-enabled

Import Files in the Correct Order Based on Dependencies

The `nnmconfigimport.ovpl` tool allows you to import an entire set of configuration files in a directory; however, you may also load files one at a time. This enables you to see any errors based on individual files. You can then delete the conflict and re-import the file. You may need to load multiple configuration files to satisfy any dependencies. For example, an Interface Group can reference a Node Group in its configuration. To satisfy this dependency, load the Node Group configuration before loading the Interface Group configuration.

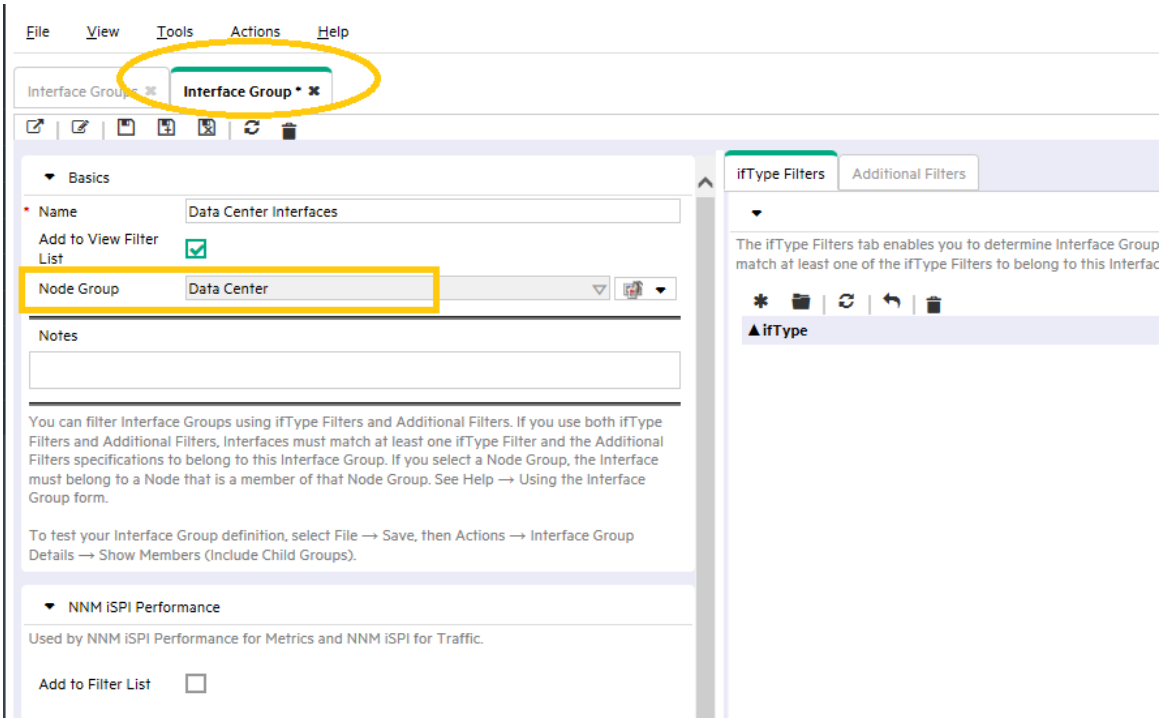

Figure 14: Interface Group: Referenced Node Group

Merger Two Server Configurations

To merge two server configurations (to keep both servers synchronized) use any of the following three methods:

Method 1:

1. Elect one of the servers to be the “master” server where all configuration changes are made.
2. Make a full backup of the master server (using `nnmbackup.ovpl`) and restore it to the “slave” server (using `nnmrestore.ovpl`) to initialize all the UUIDs to the same values.
3. Export your configurations from the master server and then import them onto the slave server.

Method 2:

1. Use the graphical user interface to manually delete configurations on the slave server that may likely cause a conflict (for example, configurations that you have changed).
2. Make sure that you disable State Polling during this period to avoid unexpected alarms.
3. Export your configurations from the master server and then import them onto the slave server.

Method 3:

1. Import configurations one at a time from the master server without deleting anything on the slave server.
2. If an import fails, then delete the conflict and re-import.

Caution

You must import configurations within the same NNMi version and patch level. For example, you cannot export a configuration in NNMi 8.x and import it into NNMi 9.x.

Tips

- Determine which method, import/export or backup/restore, is your best option for moving data based on your goals and configuration.
- Study the NNMi help to determine if the import for a given configuration merges or replaces the current configuration.
- Avoid naming customer-built configurations with the same names as default configurations (such as Node Groups). This will help you avoid conflicts. Remember that you can assign custom configurations a higher priority (lower number) than the default configurations, thereby effectively disabling the default settings.
- Delete configurations that will clash before attempting the import to avoid UUID conflicts.
- Consider completely deleting configurations to ensure a successful import and consider disabling State Polling temporarily.
- Avoid merging between NNMi management servers. If you must use this approach, then elect one of the servers to be the master server where all configuration changes are made. Prepare the servers in advance by making a backup of the master and restoring the backup to the slave server to synchronize UUIDs.
- For simple configurations, consider using the graphical user interface to visually compare servers and re-enter data rather than using the import and export or backup and restore tools.

Conclusion

This document has presented usage scenarios and best practices for NNMi’s export and import tools.

We appreciate your feedback!

If an email client is configured on this system, by default an email window opens when you click here.

If no email client is available, copy the information below to a new message in a web mail client, and then send this message to **network-management-doc-feedback@hpe.com**.

Product name and version: NNMi 10.30

Document title: Step-by-Step Guide to Using NNMi Import and Export Tools

Feedback: