Infor ION Grid Administration Guide for LifeCycle Manager 10

Version 11.1.11.0
Published February 18, 2014

Important Notices

The material contained in this publication (including any supplementary information) constitutes and contains confidential and proprietary information of Infor.

By gaining access to the attached, you acknowledge and agree that the material (including any modification, translation or adaptation of the material) and all copyright, trade secrets and all other right, title and interest therein, are the sole property of Infor and that you shall not gain right, title or interest in the material (including any modification, translation or adaptation of the material) by virtue of your review thereof other than the non-exclusive right to use the material solely in connection with and the furtherance of your license and use of software made available to your company from Infor pursuant to a separate agreement, the terms of which separate agreement shall govern your use of this material and all supplemental related materials ("Purpose").

In addition, by accessing the enclosed material, you acknowledge and agree that you are required to maintain such material in strict confidence and that your use of such material is limited to the Purpose described above. Although Infor has taken due care to ensure that the material included in this publication is accurate and complete, Infor cannot warrant that the information contained in this publication is complete, does not contain typographical or other errors, or will meet your specific requirements. As such, Infor does not assume and hereby disclaims all liability, consequential or otherwise, for any loss or damage to any person or entity which is caused by or relates to errors or omissions in this publication (including any supplementary information), whether such errors or omissions result from negligence, accident or any other cause.

Without limitation, U.S. export control laws and other applicable export and import laws govern your use of this material and you will neither export or re-export, directly or indirectly, this material nor any related materials or supplemental information in violation of such laws, or use such materials for any purpose prohibited by such laws.

Trademark Acknowledgements

Publication Information

Release: 11.1.11.0

Publication date: February 18, 2014

Document Number: GRIDAG_11.1.11.0_UWA_04

Version Log

The version log describes the changes between versions of this document.

Part Number	Release Date	Description
GRIDAG-110111	2013-05	New release, for use with LCM 10. Security information moved to *Infor ION Grid Security Administration Guide*.

Contents

ION Grid Overview

1

The ION Grid

The ION Grid is an application server. It provides a distributed runtime environment to other applications. Those other applications may, at any time, be added (deployed) or removed from a grid. The distributed nature of a grid means that an instance of the Infor ION Grid may span multiple server machines.

The ION Grid consists of several parts:

Host

A host is a server machine that is participating in a grid. The host may be a physical or a virtual machine. Obviously, each grid has at least one host but may have several. A host may be a member of more than one grid.

Hosts operating in the same grid do not have to be running the same operating system as each other. A grid can contain any combination of hosts from the supported platforms.

Bootstrap

The grid bootstrap is a component used to install, upgrade, and launch the grid on a specific host. The bootstrap JVM is not owned by the grid but is in fact executed as a service, registered with the operating system.

Grid Agent

Internally, for administrative purposes, a grid needs to communicate with all hosts that it spans. For this reason, each host needs a grid agent running that takes care of this communication. A host has one grid agent per grid it is a member of.

The grid agent is responsible for managing the grid on a specific host. Responsibilities include deployment of applications on that host and starting of nodes and routers.

Node

A node is a JVM that is registered as being part of a grid where grid applications are running. A grid typically has several nodes running different applications. Each node is running on one of the hosts that are part of the grid.

Nodes operating in the same grid do not have to be running the same Java version as each other. A grid can contain any combination of nodes running the supported Java versions. A typical example of combining Java versions could be using 32-bit and 64-bit editions together depending on the memory requirements for a specific node or application.

Registry

The registry is a special type of node that is needed when new nodes are started. A grid has exactly one registry. However, a second registry host can be defined and in the event of a primary host failure or inability to start the registry node on that host, the secondary host will start a registry to allow the grid to continue to operate without interruption.

The registry is responsible for maintaining the current topology of the grid and is indirectly involved in the use of distributed objects on the grid. Without the registry node, a grid will continue to operate but new nodes will be unable to start and as such the registry is critical to the operation of the grid.

Router

A router acts as a well-defined entry point that client applications can connect to. A router is configured to listen for client requests on a given network interface and with a number of specified ports including HTTP, HTTP(S) and in some cases legacy TCP/IP ports. Normally, the selected port number has to be made accessible through firewalls, when applicable, since it must be reachable from clients.

Administrative Router

The administrative router is a special router used in the grid to manage the administration functions (Configuration Manager) of the grid.

Knowledge Prerequisites

To install this product, you should have the following knowledge and experience:

- Have experience installing and configuring applications.
- Have operating system administrator experience.

ION Grid Concepts

A user of the ION Grid should be aware of the following concepts when installing and administering the grid.

Node Types

A node type defines **what** to run in a specific node. Each node is of exactly one node type. More precisely, the node type defines what application to run in nodes of this type and may also define default values for properties (for example, heap size). Node types are defined by the application developer.

Bindings

A binding defines **where** and **how** to run nodes of a specific node type. It can be seen as an association between a node type and a set of hosts. In order to start a specific node type on a particular host, a binding that associates the node type with the host is needed. Properties needed by the node or the application running in the node may be defined per binding.

Bindings are defined when applications are installed and/or by a grid administrator at runtime.

Applications

A grid application is a logical grouping of one or more application modules. An application may be running in more than one node. It is then said to have more than one application instance. Applications for the grid are packaged in gar files. A gar file is a type of zip file that can be installed in a grid. It contains Java class files (jar files) and any other resources that the application may need.

ION Grid Administration Tools

2

This section describes the administrative tools that are available for a grid and how to access these tools.

ION Grid Administration Tool Overview

A grid includes several administration tools for managing and configuring the grid. These are:

- Grid Management Pages
- Configuration Manager
- Offline Configuration Manager
- Troubleshooting Tools
- The Grid Script Utility

For procedures to access the management and configuration tools, see:

- "Accessing the Grid Management Pages Through Java Web Start" on page 13
- "Accessing HTML-Based Grid Management Pages" on page 13
- "Accessing the Configuration Manager through LifeCycle Manager" on page 14
- "Accessing the Offline Configuration Manager" on page 14

For information about each of these tools, see:

- "Grid Management Pages for Monitoring the Grid" on page 15
- "Configuration Manager Overview" on page 18
- "Offline Configuration Manager" on page 18
- "Troubleshooting Tools" on page 19
- "The Grid Script Utility" on page 20

Accessing the Grid Management Pages through LifeCycle Manager

Use this procedures to access the Grid Management Pages if you are working within the LifeCycle Manager.

Note: You can also access a Java Web Start accessible version of the Grid Management Pages and an HTML-based version of the Grid Management Pages. For more information, see:

- "Accessing the Grid Management Pages Through Java Web Start" on page 13
- "Accessing HTML-Based Grid Management Pages" on page 13

To access the Grid Management Pages through the LifeCycle Manager

1. In the LifeCycle Manager (LCM), select the Applications tab in the left pane and locate your grid.
2. Once you have located the particular grid you want to open the Grid Management Pages for, double-click it.

   This will open the dashboard for this grid in the right pane.
3. On the Tasks tab in the right pane, click the Grid Management Pages link.

   This will open the Management Pages in a new tab.

   Note: If you are presented with a warning about the certificate, see "Importing the Grid Root Certificate" on page 18 for information on how to prevent this warning from appearing.

Accessing the Grid Management Pages Through Java Web Start

You can access the Grid Management Pages through a Java Web Start component.

To access the Grid Management Pages through Java Web Start

1 In a browser, access the Web Start link at http(s)://*server*:*port*/grid/info.html.

where *server* is the name of the server hosting the grid and *port* is the HTTP or HTTPS port for the grid router.

2 In the Grid Information section, click the Web Start link.

3 If you are prompted to run the program to install the Java Web Start component, respond to the prompts.

If your choice is to save the file, click Save and select a location on your computer to save the file.

You can also right-click on the link and select your browser's option for downloading files: "Save target as..." on Internet Explorer, "Save Link As..." on Mozilla Firefox and Google Chrome, and "Download Linked File As..." on Apple Safari.

4 Double-click the downloaded file to run the Java Web Start. The file is a .jnlp file, so if your computer does not recognize the file type, you can browse to the Java **`javaws.exe`** file and use that to open the file.

To access the Grid Management Pages through Java Web Start (alternative method)

1 Open a command window and navigate to *grid_root_installation_path*\grids*yourGridName.*

2 At the command line, type

```
AdminUI.cmd
```

Accessing HTML-Based Grid Management Pages

You can use a browser to access HTML-based Grid Management Pages. The HTML version of the user interface offers the same level of functionality as that found through the Java Web Start component including security.

To access the HTML-based Grid Management Pages, use one of the following browsers:

- Microsoft Internet Explorer, 9.0 and above
- Mozilla Firefox, 12.0 and above
- Google Chrome, 19.0 and above
- Apple Safari, 5.1.7 and above

To access the HTML-based Grid Management Pages

1. Open one of the supported browsers.
2. Navigate to the following URL:

   `http(s)://`*`server`*`:`*`port`***`/grid/ui`**

   where *server* is the name of the server hosting the grid and *port* is the HTTP or HTTPS port for the grid router.

Accessing the Configuration Manager through LifeCycle Manager

Use this procedure to access the Configuration Manager for a grid if you are working with the LifeCycle Manager.

To access the Configuration Manager through the LifeCycle Manager

1. In the LifeCycle Manager (LCM), select the Applications tab in the left pane and locate your grid.
2. Once you have located the particular grid you want to open the Configuration Manager for, double-click it.

   This will open the dashboard for this grid in the right pane.
3. On the Tasks tab in the right pane, click the Configuration Manager link.

   This will open the Configuration Manager in a new tab.

   Note: If you are presented with a warning about the certificate, see "Importing the Grid Root Certificate" on page 18 for information on how to prevent this warning from appearing.

   Tip: It is easy to navigate back and forth between the Grid Management Pages and the Configuration Manager. When you are on a page belonging to either of the two, you will always see a set of icons in the upper right corner. The house icon will take you to the Grid Management Pages and the cogwheels icon will take you to the Configuration Manager.

Accessing the Offline Configuration Manager

Use this procedure to access the Offline Configuration Manager. This is useful when you cannot access the other administrative user interfaces.

To access the Offline Configuration Manager

1. Open a command window and navigate to *grid_root_installation_path*\grids*yourGridName.*
2. At the command line, type

   ```
   OfflineConfigUI.cmd
   ```

3. Enter the connection options for the grid you want to manage. The default is to use the configuration for the grid located where you launched the Offline Configuration Manager from.
4. Click Launch Config Manager.

Grid Management Pages for Monitoring the Grid

When you are running grid applications in a grid, it is important to monitor the state of both the applications and the grid itself. This monitoring enables you to see if an application is having problems that must be addressed or if the grid itself has problems. Typical things to monitor are errors or warnings in the log files. Also, making sure that applications are configured with sufficient memory is very important.

Monitoring of both the grid and its applications is done from the Grid Management Pages. Of the many Grid Management Pages, those that are particularly useful for monitoring the grid are described below.

To access the Grid Management Pages, see "Accessing the Grid Management Pages Through Java Web Start" on page 13 and "Accessing HTML-Based Grid Management Pages" on page 13.

The Topology Overview Page

The Topology Overview page is the initial Grid Management page that you see. It shows a logical view of the hosts running in the grid and all nodes running on them. The page is intended to give information on all individual runtime artifacts: grid agents, the grid registry, grid routers, and application nodes. Technically, each runtime artifact, except hosts, corresponds to a JVM.

The top of the page displays:

- The name of this grid instance
- A stop button that lets you stop the grid or applications in the grid
- A link to the status page
- A link to the logging page

  From the logging page, you can change the log levels and view the content of the log files
- A link to pages showing more advanced details of the grid

  The advanced part contains management of the client connections, dispatchers, and proxies as well as advanced configuration options for the grid internal parts. The information displayed in this part is to be used for troubleshooting purposes and is not to be altered.

- A link to a page with application focus (in contrast to the runtime artifact focus of this page)

  The applications run in the grid are shown here. It is possible to view the status of the applications, the version of the applications, and the number nodes on which they run. It is also possible to load, start, stop, reload and unload the application.

For each runtime artifact, the page displays the following:

Column	Description	Comment
Type	The type of runtime entity displayed on a row, for example, host, grid agent, registry, router, or node	For hosts, this column also includes a start button that allows you to start nodes (bindings) on that particular host.
Name	The name of the runtime entity	For nodes, this is the name of the binding used to start the node.
Status	The general status of the entity. Normally this should be "OK," but it may also be "OFFLINE" or "STOPPING."	When a node is off-line, it will take no new requests while letting already processing request finish. For more information on the off-line status, see "Putting Applications or Parts of the Grid in an Off-Line State" on page 52.
	A stop button for all entities except hosts and grid agents	This is useful when an individual application node needs to be shut down, for example, because it is in an error state.
Id	A unique ID for each runtime artifact.	The ID consists of address, port, and process id.
Log	Link that displays the log file belonging to this runtime artifact	This is the default place to look for information if a node is showing signs of having problems.
Log Status	Number of errors and warnings in the log	View the log for details. If errors or warnings are indicated, the cause should be investigated. When the problems have been taken care of or they have been deemed unimportant or irrelevant, it is possible, and recommended, to reset the loggers using the "Reset Loggers" button at the top of the page. This will reset the error and warning counts to zero so that it is easier to see if new errors appear. It is only the count that is reset. No information is removed from the log files.

Column	Description	Comment
Up Time	How long this runtime artifact been running (time since last started)	A short up time could indicate that a node has been automatically restarted due to some problem. Obviously, the short up time could also have a natural explanation.
CPU%	Approximate CPU usage	Different applications use the CPU in different ways. What is normal depends on the application. This information can indicate if an application does not behave as it usually does.
Heap Usage	Memory usage of the runtime artifact (JVM)	Typically application should not be short of memory in order to perform well (there are exceptions).

You can expand hosts so that more information is displayed. To do so, click on the small plus icon to the far left of each host row in the list or click on the Expand All button. When expanded, routers will list information about external addresses and ports used. Nodes will list individual application modules.

At the bottom of the page are links to all applications. If the applications provide management pages of their own, a link to them is also displayed. These links are useful to get more information about a particular application.

The Status Page

This Status page is accessed from a Status link at the top of the Topology Overview page.

The status page is intended to give a condensed view of the status of the grid itself and of each application. The page is constructed so that it only show problems if they exist. This means that the less information that is shown the better it is. Once an application has some form of problem, this will be indicated on this page and the actual problem will have to be investigated using the other management pages. The application names and problems that are displayed often form a link to a page that is relevant to the problem at hand.

The nature of this page makes it suitable when you want a high level view of how the grid and its applications are performing.

The Applications Page

The Applications page is accessed from an Applications link at the top of the Topology Overview page.

This page displays the same information as the Topology Overview page but the information is rearranged with a focus on applications. Normally, an overview of each application is listed. If a particular application is selected, that application is moved to the top of the page and more information about that application is displayed. The detailed information is essentially the same as in the topology overview but only the runtime artifacts that are related to the application are displayed.

This page also provides convenient ways to access the configuration of the applications as well as starting, stopping, and managing the off-line state of the applications.

Configuration Manager Overview

The Configuration Manager provides access to a variety of aspects of the grid whose configuration you may want to change, such as making changes to affect performance, adding hosts, deploying applications, configuring bindings, and setting up user roles and mappings. You can also use the Configuration Manager to view many types of information about the grid, including audit logs, configuration files, configuration history, defined ports, and certificates.

For a detailed description of the tasks that can be performed through the Configuration Manager, see "Configuration Manager" on page 84.

Offline Configuration Manager

The Offline Configuration Manager is a restricted functionality version of the Configuration Manager that is available through the Grid Management Pages. It enables you to troubleshoot or resolve errors when you cannot do those things through the normal administrative user interfaces.

Some examples of where this might be required are as follows:

- The grid will not start because all hosts have been de-activated.
- You are unable to log in to the grid as grid-admin due to a breakdown in the security process (for example, all administrators have been accidentally removed).

To access the Offline Configuration Manager, see "Accessing the Offline Configuration Manager" on page 14.

Importing the Grid Root Certificate

Sometimes when you are accessing either the Grid Management Pages or the Configuration Manager from within the LifeCycle Manager, you may be presented with a window concerning the certificates. This is because the grid uses a self-signed root certificate as the issuer of grid certificates (client, host, SSL). Certificates issued by the grid root certificates are not by default trusted in browsers and in the Lifecycle Manager client like certificates issued by public certificate authorities such as VeriSign, EnTrust, and Thawte are. In order to permanently trust certificates, the grid root certificate must be imported into the Trusted Root Certificates store in each browser used. This situation is the same for the Lifecycle Manager client.

To import the grid root certificate to the Trusted Root Certificates store

1. In Application tab in the left pane of the LifeCycle Manager, locate the grid whose root certificate you need to import and double-click it.
2. In the right pane on the Tasks tab, click Grid Management Pages.

3 When a window titled "Security Alert" with the message "Revocation information for the security certificate for this site is not available. Do you want to proceed?" is displayed, click the View Certificate button.

A new window, titled "Certificate," appears and shows the details of the host certificate for the grid.

4 Select the Certification Path tab at the top of the window.

5 In the Certificate Path field, select the entry at the top. This should be a certificate icon with the name of the grid next to it. The certificate icon most likely has a little symbol on the bottom left corner indicating that it is not trusted. The symbol is a red circle with a white X in it.

6 After you select the certificate, click View Certificate.

A new window is displayed, titled "Certificate," which shows the details of the root certificate for the grid.

7 Click on the Install Certificate… button.

8 In the Certificate Import wizard welcome screen, click Next.

9 In the Certificate Store screen select "Place all certificates in the following store" and click Browse… to select the store.

10 In the Select Certificate Store window select "Trusted Root Certification Authorities" and click OK.

11 Back in the Certificate Store screen of the Certificate Import wizard, click Next.

12 Click Finish.

13 When a security warning is displayed about importing a certificate from a certificate authority, click Yes to add the certificate.

14 Click OK to close the window that states that the import was successful.

15 Close the Certificate windows for the root and the host certificates.

16 Back at the Security Alert window click Yes.

Troubleshooting Tools

The ION Grid provides many sources of information about the configuration and runtime status of a grid and the applications running within the grid. These include:

- Log files

  Log files are available for each node. For more information, see "Viewing Log Files" on page 21.

- Counters

  Counters are available for each grid node as well as a counter history. For more information, see "Viewing Counters" on page 31.

- Configuration history

You can review a history of changes to the grid configuration so that you can, for example, identify changes that caused a problem or revert to a previous configuration. For more information, see "Viewing Configuration History" on page 35.

- Heap dumps and thread dumps

  You can view heap dumps and thread dumps for individual grid nodes. For more information, see "Getting a Heap Dump or Thread Dump Via the Grid Agent" on page 66.

- Grid Status report

  The Grid Status report is a generated report that shows the overall status of the grid, and also includes log files, configuration files, counter history, and thread dumps. For more information, see "Generating a Grid Status Report" on page 25 and "Viewing a Grid Status Report" on page 26.

For specific troubleshooting procedures and advice, see "Troubleshooting" on page 65.

The Grid Script Utility

The ION Grid includes a built-in utility class that you can help you set up application maintenance. For example, if you need to schedule backups, it is useful to have a script that shuts down an application programmatically, performs the maintenance tasks, and then restarts the application. The built-in utility class enables you to easily include the stop and start of applications in a script.

For more information, see "Using the Grid Script Client to Manage the Grid" on page 62.

Monitoring the Grid

3

Log Files and Reports

Viewing Log Files

Log files are associated with each grid node. Because each application is running in one or more grid nodes, you can obtain valuable information about an application by viewing these log files. You typically access log files from the Topology Overview page, Status page, or Applications page. Once a log file is opened, you can filter and search for relevant information. It is also possible to view log files of old nodes that are no longer running and to download log files in order to, for example, e-mail them to someone.

To view log files

1. Access the Grid Management Pages for the grid whose log files you want to view: Topology Overview page, Status page, or Applications page for an application.
2. Click the appropriate link, depending on which Grid Management page you are on.
   - On the Topology Overview page, click the button in the Log column for the runtime artifact whose log file you want to view.

- On the Topology Overview page, click the Logging link and then click the View/Filter/Download button in order to view the merged system log.
- On the Status page, click the LOG link for the item whose log file you want to view and then click the button in the Log column on the Application Logs dialog box.
- On the Applications page, click the Logs button for the application whose log file you want to view.

To search and view merged system log

Note: This procedure provides a merged view of all in-memory backlogs for all active nodes. It also shows the system warning count (entries in log files which are marked SYSTEM).

1 At the top of the Topology Overview page, click the Logging link.

2 Click the View/Filter/Download button.

3 On the System Logs dialog box, select one of the following options for working with system log files:

- To view the merged in-memory backlogs for all active nodes, click the document icon to the right of Merged Log.
- To reset the SYSTEM warning count for all active nodes, click Reset.
- To show or hide the details for the nodes that are included in a merge, click the Show Details or Hide Details link.
- To reset the SYSTEM warning count for individual nodes, click the Show Details link and then click Reset for the individual node.
- To open the in-memory backlog for an individual node, click the Show Details link and then click the document icon for the individual node.
- To show or hide the filter options, click the Show Filter or Hide Filter link.
- To set the filter options, click the Show Filter link and then select from the following:

Filter Setting	Description
Recent Only	Include only in-memory backlogs.
Active Log Files	Include all active log files (combination of persisted log and in-memory backlog).
Filter Text	Case insensitive text search for word(s) to include.
From	A date and time from which the filtered merge should begin.
To	A date and time to which the filtered merge should end.
View Result	View the merged log based on the selected files and filter.

Filter Setting	Description
Download Result	Download the merged log based on the selected files and filter.

To view log files for old nodes or nodes not currently running

Note: This procedure is useful to access log files for nodes that have crashed. The log files in that case are not accessible by the usual means.

1 At the top of the Topology Overview page, click the Logging link.

2 On the Logging page, click the Log Files link.

3 In the list of log files, click the link in the Filename column for the log file you want to view.

To download log files

Note: You can also download log files from the complete list of log files described in the procedure for viewing log files for old nodes. Simply click on the Download link in the Action column for the log file you want to download. The following procedure provides a more selective list of log files to download.

1 On the Topology Overview page, click the Name link for the application's node.

2 On the local management page for the node, click the Logging link.

3 On the Logging page for the node, click the Log Files link.

4 In the list of log files, locate the log file you want to download.

5 Click on the Download link in the Action column for the log file you want to download.

Configuring Logging Levels

The log level can be configured on multiple levels. The configuration is found in slightly different ways for each level.

Grid-wide Logging Levels

To configure grid-wide logging levels

1. Access the Configuration Manager as a user with the grid-admin role.
2. Click Grid Properties.
3. Find the Node log level property and click on the link in the Value column. The link is either "<undefined>" or a list of log levels such as "ERROR,WARN,INFO,NOTE."
4. Select which log levels to have. Click Save.
5. Click the Save button on the upper left corner.

Router Logging Levels

To configure router logging levels

1. Access the Configuration Manager as a user with the grid-admin role.
2. Click Grid Properties.
3. Find the Node log level property and click the "Node log level" link (not the link in the Value column).
4. To change the log level of a particular router, click on the link in either the "Any host" column or the column for a particular host. If a particular host is selected, the logging levels set will only affect the selected router on that host.
5. Click the Save button on the upper left corner.

Application Logging Levels

To set the application log level it is either possible to follow the instructions for router levels above or to do the following:

To configure application logging levels

1. Access the Configuration Manager as a user with the grid-admin role.
2. Click Applications.
3. Click on the application to configure log levels for.
4. Click on Edit Properties.
5. Find the Node log level property and click on the link in the Value column. The link is either "<undefined>" or a list of log levels such as "ERROR,WARN,INFO,NOTE."
6. Select which log levels to have. Click Save.

7 Click the Save button on the upper left corner.

Application-level Logging for a Specific Host

To configure application-level logging for a specific host

1 Access the Configuration Manager as a user with the grid-admin role.

2 Click Applications.

3 Click on the application to configure log levels for.

4 Click on Edit Properties.

5 Find the Node log level property and click the "Node log level" link (not the link in the Value column).

6 On the Property:Node log level page it is possible to set the log level for all hosts, as well as for individual hosts. To change the log level, click on the link in either the "Any host" column or the column for a particular host. If a particular host is selected, the logging levels set will only affect that host.

7 Select which log levels to have. Click Save.

8 Click the Save button on the upper left corner.

Generating a Grid Status Report

Use this procedure to generate a Grid Status report.

The ION Grid is able to generate a report that includes a variety of information that is useful for determining the state of a grid and its deployed applications. The report will indicate problems that it finds and it will also contain log files, configuration files, and other things that may be of use for tracking down problems. Although you can use the report simply to confirm that the grid is working satisfactorily, the report is very helpful to include when you need to report a bug or problem with the ION Grid or an application running in the grid.

To generate a Grid Status report

1 Navigate to the Topology Overview page in the Grid Management pages.

2 Click the Advanced link.

3 Click the Grid Status Report link.

4 Click the Auto Select Node button.

Note: Any grid node or router may be used to generate the report. Clicking the Auto Select Node button means that the grid will select a node that it thinks is the recommended one to use in this case as indicated in the list of nodes below the Auto Select Node button.

If you have reason to believe that the recommended node is not a good choice because it is experiencing problems, you may manually select a specific node from the list below instead of clicking the Auto Select Node button.

5. Select what you want to be included in the report. Typically you want the default values.
6. Click the Generate Report button and wait for the report to be generated.
7. In the dialog box that opens, click the Download… link.
8. Save the report (zip file) to disk.
9. Close the dialog box.

Viewing a Grid Status Report

Use this procedure to view a Grid Status report that you have generated and downloaded as described in "Generating a Grid Status Report" on page 25.

To view a Grid Status report

1. Unzip the saved report into a separate directory. It needs to be unzipped to work correctly. Viewing the report directly from within the zip file doesn’t work.
2. In the new directory, open the report.html document with a browser. The report.html document provides an overview of the current status. Any problems that were found are typically indicated using strong yellow or red color schemes.
3. View other report files in the directory. These include:
   - Log files for existing and old grid nodes
   - Configuration files for the grid
   - Counter history for active grid nodes
   - Thread dumps for all active grid nodes

Monitoring Tools

- "Monitoring the State of the Grid from a Web Browser" on page 27

- "Monitoring the State of the Grid from the Java Web Start Component" on page 28
- "Viewing Counters" on page 31

Monitoring the State of the Grid from a Web Browser

Use this procedure to view information on the state of the grid by accessing HTML pages through a web browser.

You access the HTML pages via any of the HTTP(S) ports that are defined for the grid routers. For information about how to identify the HTTP and HTTPS ports, see "Listing All Ports Exposed by Routers in a Grid" on page 45.

The general theme of all of these pages is that when the status is good only limited and basic information is displayed. However, if problems are found, more information will be displayed with strong yellow or red warning color schemes for problem areas.

Note that the same information that is used to render the HTML pages is also available in XML format. The XML format is suitable in situations where you programmatically want to monitor the state of the grid.

To access HTML status pages via a web browser

1. Identify the HTTP or HTTPS port for the grid whose status you want to check. For information about how to identify the HTTP and HTTPS ports, see "Listing All Ports Exposed by Routers in a Grid" on page 45.
2. Open a web browser and enter the URL for the status page you want to view. See the sections below for the URL for each page.

info.html

URL: http(s)://*server*:*port*/grid/info.html

This page displays basic grid version information as well as the following:

- A link you can use to launch the Grid Management Pages using Java Web Start
- A listing of the context roots defined by applications installed in the grid
- Grid Status, Host Status, Node Status, and Port Status links to, respectively, the status.html, hosts.html, nodes.html, and ports.html pages

status.html (Grid Overview)

URL: http(s)://*server*:*port*/grid/status.html

This status page displays a brief entry for each deployed application. The intention of this page is to give a quick indication that everything is in order or to indicate if there are problems with one of the applications or the grid itself. In addition, the page shows detailed information if there is a problem. So, if there are no details displayed, it means that things are good.

This page also provides links to detailed information for each application as described below.

status.html (Specific Application)

URL: http(s)://*server*:*port*/grid/application/*applicationName*/status.html

This status page provides more information about a particular application. It shows the deployment status on different hosts (if the application is deployed on more than one host) and it shows information of all running nodes that belong to this application. If the application is badly configured (in a way that is automatically discoverable), that will be indicated here as well.

hosts.html

URL: http(s)://*server*:*port*/grid/hosts.html

This page displays information related to each host in the grid. The information includes such things as the operating system, number of CPU cores, memory usage, and online status. If problems are found, they are indicated as well, for example, if a grid agent is inaccessible on one of the hosts.

nodes.html

URL: http(s)://*server*:*port*/grid/nodes.html

This page is similar to the Topology Overview page of the Grid Management pages but is rendered using html. It shows a list of all grid entities (agents, registry, routers, and nodes) and their status. Links to the active log files are also provided. Problems are highlighted if they exist.

ports.html

URL: http(s)://*server*:*port*/grid/ports.html

This page displays information for each port that is exposed by the grid. Apart from just listing the ports and their addresses, information about how much data that has been read and written on each port is also displayed.

This page can be used to find a suitable port to connect a client to or it may be used to get a view of the total network exposure that the grid has in order to make sure that firewalls are configured accordingly.

Monitoring the State of the Grid from the Java Web Start Component

Use this procedure to monitor the grid using the profiler viewer and log viewer that are available through the Grid Management Pages when these are accessed through the Java Web Start. The tools are accessed through the Launch menu at the top of the window.

Profiler Viewer

This tool enables advanced low-level profiling for the grid application nodes to enable troubleshooting and advanced performance monitoring. The profiler is primarily intended for use by developers and those with a deep understanding of how the grid functions.

When you launch the tool, you will be presented with a list of the grid nodes for which profiling is enabled. That view is structured as follows:

```
Grid
Node:<node_name>/<ip_address>:<port>-<process_id>
  -<module_name>
   -<grid_proxy_name>
    -<method>
     -<counter>
```

For each of the counters you can view the following information which is valid for the duration between you starting then stopping profiling:

- Count - number of iterations of that entity
- Time(ms) - total time spent on iterations in ms
- Size (KB) - total amount of data for all the iterations
- Time/Count - average time spent per iteration
- Size/Count - average amount of data per iteration

Menu options available in the function and descriptions are as follows:

Menu	Option	Description
File Menu		
	Open	Open a previously saved profile data log file.
	Save	Save the current profiling data to a profile data log file.
Actions Menu		
	Reset Profiler	Clear profiling data.
	Stop Profiling	Stop profiling data.
	Start Profiling	Start profiling data.
View Menu		
	Size Unit	Change the displayed size unit for profiled data (Bytes, Kilobytes, Megabytes).
	Time unit	Change the displayed time unit for profiled data (Nanoseconds, Microseconds, Milliseconds, Seconds).

Menu	Option	Description
	Expand/Collapse	Show Methods - expand the profile data tree to only show methods, only applicable when tree is fully collapsed. Expand All - fully expand the profile data tree. Collapse All - fully collapse the profile data tree.
	Aggregation	Change the aggregation levels for the data.
	Refresh	Refresh the panel.

Log Viewer

The log viewer allows multiple log files to be able to be combined and searched in a merged fashion. This has multiple uses including the ability to find out what an entire grid was doing at a particular point in time, or to find all instances of a particular word or phrase.

The viewer consists of four separate panels: a Files panel, a Filter panel, a Logs Time Span panel, and an Output panel.

Files Panel

On the Files panel, you can add files either by clicking on the Add button or by dragging and dropping them directly into the pane. You can select multiple files for adding by using the Shift key and the Ctrl key. Files can be either individual .log files or zip-files. If you add a zip file, the tool will recursively search the zip file directories and add all .log files to the panel. This is useful for loading the contents of a Grid Status Report zip file.

For any files that do not have a .log extension, you will be prompted with the option to include them. Note that adding files which do not have a .log extension may prevent the merging of the log files from occurring properly.

The Files panel has four buttons:

- Add - add files (either .log or .zip containing log files).
- Remove - remove the selected file or if no selection is made, remove all files.
- Details - toggle between showing all file details, including the name, size, modified date, and the file path, and showing only the name. The default is to show all details.
- Sort - sort the list by Added (the order in which the files were added), Name (the file names), Size (the size of the files), or LastModified (the date the files were last modified).

Filter Panel

The Filter panel enables you to set filter criteria for viewing the log files. After you set the criteria, you click the Update button and the results appear in the Output panel (right panel).

There are several filter options:

- Clear - clear the filters
- Include - case insensitive text search for word(s) to include. It is possible to include OR as an operator by inserting a vertical bar (|) between the words to search for, for example, " info OR system" would be constructed as "info|system".
- Exclude - case insensitive text search for word(s) to exclude. I is possible to include OR as an operator by inserting a vertical bar (|) between the words to search for, for example, " info OR system" would be constructed as "info|system".
- From - specify a date and time from which the filtered merge should begin.
- To - specify a date and time at which the filtered merge should end.
- Update - update the merged log output pane based on the selected files and filter. If no files are selected in the file list pane, all files are included.

Logs Time Span Panel

The Logs Time Span panel shows the time span(s) at which the merged entity occurred in an included log file.

To work with this panel:

- Hold the mouse cursor over an area in the panel to see the date and time shown as a tool tip.
- Click in the window to position the merged output pane according to the date and time displayed in the tool tip.
- Right-click in the window on a specific log file to position the merged output pane on the selected log file according to the date and time displayed in the tool tip.

Output Panel

This panel show the merged log output. The data is ordered in the following manner: First, the merged sources (log files) are listed. Second, the results of the merge, including applied filters, are displayed. The result lines contain identifying numbers showing which source they are from.

Viewing Counters

Counters are a mechanism that measures (counts) different things that occur in a grid node and that may be of interest for monitoring and diagnosing the inner workings of an application. Counters are maintained for each grid node and they are not persisted, so the information is lost if a node is stopped.

Good examples of counters are Used Heap, which shows the memory consumption, and Total Requests, which shows how many requests have been handled by a thread pool in a node. However, there are many others.

One very important aspect of counters is that the counter value is sampled periodically and a history of counter values is maintained for each counter. This makes it possible to view things like memory usage over time and even get the information displayed as a graph.

To view counters

1 Navigate to the Topology Overview Page in the Grid Management Pages.

2 Click the name link of the node you want to monitor (remember, counters belong to nodes).

3 Click the Advanced link.

4 Click the Counters link.

A list of counters will be displayed. The displayed list shows the counters with their current value. Some counters define a valid value range. If a counter is outside of the valid range, it is marked with a yellow background. A counter that is outside of its valid range is unusual and is worth investigating.

5 Clicking on the name of a counter will display a graph with the counter values over time. At the top of the page it is possible to select units of measurement and also how often the counter history should be polled and how much of it to keep in memory. The two latter settings may also be permanently configured using the following grid properties, Counters Poll Delay and Counters Keep History.

Changing the Grid Configuration

4

Configuring Memory Given to Applications

This procedure explains how to set a global application value for the Max Heap property. The Max Heap property as defined in the grid configuration controls the maximum memory amount that can be allotted at node start to an application in a particular node. This property is one of the most commonly configured properties. It enables you to accomplish the important task of ensuring that application nodes have sufficient amounts of memory.

As with any grid property, it is possible to configure the Max Heap property differently for different contexts. One example could be that you want an application node to have more memory if it is running on a particular host. Sometimes applications define several node types and you may want to configure the Max Heap property differently for different node types. The same goes for bindings. For more information about configuring a property for different contexts, see "Grid Properties" on page 89.

To configure application memory

1. Navigate to the Configuration Manager.
2. Click the Applications link.
3. In the list, select the application that you would like to change max heap for.

4 Click the Edit Properties link.

5 Locate the Max Heap property in the Grid Defined Properties section. This section is below the application defined properties. You may need to expand it to see the Max Heap property.

6 Edit the max heap for the application by clicking on the value link in the value column, and then specifying the max heap size in the dialog box that appears. Click Save.

   Note: If the max heap has been overridden or configured in several contexts, you will not see a simple dialog box, but rather a matrix that lets you configure the property differently for different contexts. For more information about configuring a property for different contexts, see "Grid Properties" on page 89.

Configuring Bindings

Bindings are needed in order to start applications. A binding is a mapping from a node type, which the application defines, to a set of hosts. So, in order to start an application on a particular host, there has to exist at least one binding that associates the application (and its node type) with that host.

The bindings are also used to govern the minimum and maximum number of application node instances that should be allowed. This means that if a binding is configured with a minimum of 2, the grid will always try to make sure that at least two application node instances referenced by that binding are running in the grid. If the number of nodes becomes fewer than the configured value, the grid will automatically start nodes until the minimum is met.

The maximum works in the same but reverse way. It is impossible to start more application node instances using this binding than the configured maximum.

It is also possible to configure the initial number of application nodes you want for a binding. If configured, the initial value will be considered when the application starts. In fact, starting applications is just a matter of honoring the initial value for all the bindings that belong to the application.

Also, as described in "Grid Properties" on page 89, the bindings are one of the different contexts that you may use when defining property overrides. This enables you to define different property values for your application depending on which binding is used to do the launch. Configuring bindings is done from the Configuration Manager.

To configure bindings

1 Navigate to the Configuration Manager.

2 Click the Applications link.

3 In the list, select the application you want to configure bindings for.

4 To add a binding, click the Add Binding link at the top of the page. To edit existing bindings, click one of the bindings in the list on the page. For more details, see "Edit Bindings to Application Defined Node Types" on page 87.

Changing the Display Name of a Grid

Use this procedure to change the display name for a grid. When you create a grid, you provide a name for the grid that then appears in various management tools. However, at a later date, you may want to change the name, especially if the name no longer matches what you use the grid for.

Note that changing the display name of a grid does not change the name of the directory where grid configuration information is stored. The directory name will still match the original name of the grid.

To change the display name of a grid

1. Navigate to the Configuration Manager.
2. Click on the Advanced Configuration link.
3. Click on the Change Grid Name link.
4. In the dialog box, specify a new name and click Set to close the dialog box.
5. Save the configuration by clicking the Save button in the upper left corner of the page.

Viewing Configuration History

Whenever changes are made in the Configuration Manager, a new version of the runtime.xml file is saved and propagated to all hosts that are part of the grid. You can list older versions of the configuration and see a description of the configuration changes that were made in each version.

To view configuration history

1. Navigate to the Configuration Manager.
2. Click the Advanced Configuration link.
3. Click the Configuration History link.

   A list of old versions of the runtime.xml file will be shown.
4. Click on any of them to get information about what changes were made when that version of the file was saved.

Reverting to an Older Grid Configuration

Use this procedure if you need to revert a grid to an older version of the grid configuration.

Important: This is a feature that should be used with care. If you have saved a new version of the configuration and you realize that it was not want you wanted, it is most likely safe to revert to the previous version.

However, if you want to restore a version of the configuration that is several versions old, you must be very careful. When restoring such a version, any configuration changes done in intermediate versions will be lost even though you may want to keep some of them. It is strongly recommended that you view the changes of all intermediate versions to see what changes were made in each and, based on that review, decide if it is safe to revert the configuration to this version. If you decide to revert, you may have to manually redo some of the configuration changes that you lost but actually wanted to keep.

To revert to an older grid configuration

1. Navigate to the Configuration Manager.
2. Click the Advanced Configuration link.
3. Click the Configuration History link.

   A list of old versions of the runtime.xml file will be shown.
4. Click the version of the configuration you want to restore.
5. Click the Apply button.
6. Commit the changes by clicking the Save button in the upper felt corner of the page.

Changing How JVMs Are Launched on the OS Process Level

Grid nodes are JVMs. When a grid node is started on a host, by default the new node is started using the same JDK or JRE as the grid agent on that host is running on. However, at times you may want to influence how the JVM processes are started in the operating system.

Examples:

- You want a particular application to be launched using a particular JDK or JRE that is different from the one used by the grid agent.
- You have an application that is monopolizing a resource (such as CPU) and you want to constrain the application nodes to a subset of the available CPU cores on a host.
- You want JVM processes belonging to a particular application to be launched using a specific user that is different from the user running the grid agent process.

Regardless of example or needs, the way you change how JVM processes are launched is as follows:

1. Create a host operating system specific script file that launches the JVM process in the way you want.

2. Override the Java Executable grid property so that it points to the script file created above. The scripts are most likely OS- and host-specific, so the recommendation is to define the property overrides in different host contexts.

Requirements for Script Files that Launch New JVMs

Obviously, the script file should in one way or the other launch a JVM since that is the purpose of the script, but there are additional requirements that the script has to comply with

1. The script must be passed all arguments that the grid would normally pass to the JVM during launch. Those arguments define what type of grid node to create. The script is responsible for propagating all those arguments to the JVM that is started by the script.
2. The script may terminate in one of two possible ways:
   - The script may block and not terminate until the launched JVM (grid node) terminates.
   - The script may asynchronously launch the JVM and exit with a return code of zero (0). A return code other than zero will be considered an error by the grid.

Overriding the Java Executable Grid Property

Example scenarios with different scripts are described below. Each of the scenarios requires that you override the Java Executable grid property so that it targets the script of each scenario. A generic description of how to do this is given below and specific details will be given in each scenario.

Working with grid property overrides in general is described in "Grid Properties" on page 89.

To override the Java Executable grid property

1. Navigate to the Configuration Manager.
2. Decide whether the override should be made in the context of an application or be made global? Typically you want to override the property in the context of an application. However, there may be odd scenarios where you want to perform the override regardless of the application. In that case, you do it in the global context.

   Depending on the situation, select one of the two paths below:

   - Override the property in the context of an application.

     a. Click the Applications link.
     b. Select the application you want to reconfigure (click on its link in the list).
     c. Click the Edit Properties link.

   - Override the property in the global context.

- Click the Grid Properties link.

3 In the Grid Defined Properties section, click the Java Executable property link in the Property column (not the Value column).

4 The Property Matrix for this grid property is shown. Using this matrix it is possible to override in various contexts. In the different example scenarios described below, details will be given relevant to each case that will enable you to continue from this point.

5 Given details from the different scenarios described below, identify the property context that you want to override and click the corresponding property value link in the matrix.

6 In the dialog box that appears, enter the path to the script file from the different example scenarios.

7 Enter the path to the script file. Note that regardless of host platform (OS) the path to the script file should be entered using forward slashes ("/") (even on Windows).

8 Click the Save button to close the dialog box.

9 Commit the changes by clicking the Save button at the upper left corner of the page.

Verifying the Scripts Are Using to Launch the Node

If you want to make sure that the script files are used to start the grid nodes in the correct situations and in the correct way, check in the grid agent log. If you have a script file named "C:/script/start.cmd", you should be able to find log entries similar to the following: **`2011-09-28 09:36:12,864 DEBUG NodeLauncher: launching: [C:\script\start.cmd`**. However, the log level DEBUG must be turned on in the grid agent in order for this to be displayed.

Example Scripts

The scripts are executed by the operating system of each grid host. Since the operating system may vary between different hosts the scripts has to be made specifically for each platform. The examples in this document are for the Windows platform but there should be no problem to port them to other platforms using other but equivalent commands.

Multi-Host Grids

The grid you are working on may be a multi-host grid and the application you want to change in the different scenarios may be deployed on many of the hosts in that multi-host grid. In order to keep things simple, the description of each scenario below will operate on one host but you must repeat the process for each host that is relevant for each scenario.

Scenarios

The scenarios described below give detailed information for each case. The detailed information is intended to be used together with the generic instructions above.

Scenario 1: Using a Specific JRE When Launching an Application

In this scenario, assume you want all grid nodes belonging to a particular grid application to be running in JVMs from a specific JRE. The rest of the grid and other applications should not be affected. Typical reasons for wanting to do this may be that the application only works using a specific patch level or a JRE from a specific vendor.

In this scenario, assume that:

- The grid host is named HOST1
- HOST1 has a JRE installed on the following path "C:\Program Files\Java\jre6".
- The grid application APP1 is deployed on HOST1

To use a specific JRE when launching an application

1 Create a script file on the grid host (HOST1), for example, C:\GridNodeStartScript\StartNodeJRE6.cmd, containing the following single line:

```
"C:\Program Files\Java\jre6\bin\java.exe" %*
```

As can be seen, the script line simply targets java.exe in the JRE that was wanted, and, by the use of "%*" at the end of the line, it makes sure to propagate all arguments that were passed to the script.

2 Configure the application to use the script by editing the Java Executable grid property. Use the following values:

- Override in the context of an application (APP1 in this case).
- In the matrix, identify the host column belonging to the host (HOST1) and click on the value link in the first row (the application level row).
- Enter the following path in the dialog box: **`C:/GridNodeStartScript/StartNodeJRE6.cmd`**. Remember to use forward slashes in the path.

The application nodes should now be launched using the specified script on HOST1.

Note: The examples given here are trying to illustrate the generic approach of creating a script file and configuring the Java Executable grid property to point to that script file. This generic approach allows for advanced scripts to be created. However, this particular example is so simple that you actually don’t have to create a script file at all. In this case, you could just enter "C:/Program Files/Java/jre6/bin/java.exe" as the value of the Java Executable grid property and skip the script file completely.

Scenario 2: Constraining an Application to a Subset of Available CPU Cores

In this scenario we have an application that is using the CPU very heavily and thus starving other applications from CPU cycles. Since the grid is intended to run several applications in the same grid, it is important that an application is not allowed to monopolize resources. To solve this problem, you need to constrain the application to a subset of the available CPU cores on the host(s) that the application is running on. This would conserve resources for other applications.

In this scenario, assume that:

- The grid host is named HOST1
- HOST1 has a JRE installed on the following path "C:\Program Files\Java\jre6".
- The grid application APP2 is deployed on HOST1

To constrain an application to a subset of available CPU cores

1 Create a script file on the grid host (HOST1), for example, C:\GridNodeStartScript\StartNodeCore.cmd, containing the following single line:

```
START /B /AFFINITY 3 "" "C:\Program Files\Java\jre6\bin\java.exe" %*
```

As can be seen, the script line uses the Windows Start command to start the JVM (java.exe) and, by the use of "%*" at the end of the line, it makes sure to propagate all arguments that were passed to the script. The Start command allows us to specify the CPU affinity. In the example, the affinity is set to 3, which means that cores 0 and 1 will be used (3 in binary format = 11).

Note the empty quotes "" after the /AFFINITY 3 argument. Due to some strange behavior of the Start command, it seems to be needed if and only if the next argument (path to java.exe) needs quotes. In the example the path contains a space, so quotes are needed, hence the extra empty quote.

2 Configure the application to use the script by editing the Java Executable grid property. Use the following values:

- Override in the context of an application (APP2 in this case).
- In the matrix, identify the host column belonging to the host (HOST1) and click on the value link in the first row (the application level row).
- Enter the following path in the dialog box: **`C:/GridNodeStartScript/StartNodeCore.cmd`**. Remember to use forward slashes in the path.

The application nodes should now be launched using the specified script on HOST1 and the application nodes should be constrained to cores 0 and 1.

This may be verified in Windows Task Manager. Right-click on the process corresponding to a grid node started by the script above. Select Set Affinity… and verify that the correct set of cores is enabled.

Tip: In the menu of Windows Task Manager, select View > Select Columns, and make sure that the Command Line column is selected. Information from the command line will make it easier to find your grid node among all processes.

Scenario 3: Running Grid Nodes Belonging to an Application as a Specific User

Grid nodes are started using the same user as the grid agent is running as. Normally, an application is not concerned about this user. It is a user that was specified when the grid was created and it should be work appropriately. However, if a particular application has to be running as another user, you have a problem. The grid is intended to run multiple applications in the same grid, so a particular application can’t be allowed to change the user that the entire grid is using since that would impact the other applications. In this case, you need to launch nodes belonging to just this particular application using anther user.

Doing this in a safe way can be tricky since it may require that user names and passwords are hard coded in script files. yet that is not something that can be recommended for security reasons.

Due to the problems just mentioned, no concrete example or recommendations will be given in this case. However, several general observations can be made.

On a UNIX platform, you have the sudo command and on Windows you have the runas command that can be used to start processes as another user. Neither of them accepts that passwords are passed to them from the command line. This is good since hard coding passwords in script files should be avoided.

Third-party tools exist that are similar to the Windows runas command and also accept a password on the command line. However, these are not recommended by Infor for the reasons mentioned above, even though they would technically solve the problem. Some of the third-party tools support encryption of the password information. Potentially those tools may provide a secure way of solving this problem but Infor does not give any recommendations in this area.

Giving the user access to the configuration area

If a particular user is used to run all or some of the grid nodes, it is crucial that the user has sufficient security rights in the grid configuration area. The topic, "Changing the Grid Service User" on page 42, describes how to change the directory security. The information is for Windows but the situation is the same on all platforms. The user has to be given access to the grid configuration area.

Changing the JDK for a Grid Node

If a grid is installed using a particular JDK version (see the *Infor ION Grid Installation Guide*) and you later want to change to another JDK version, you may do that by configuring a grid property. As always with grid properties, they may be configured and overridden in different contexts (see "Grid Properties" on page 89). Typically a change of JDK version will be done on all hosts in a grid or on individual hosts of a grid. This documentation describes how to change the JDK version for all nodes running in one host. If you want to change the JDK version for all hosts, you repeat the procedure for each host.

To change the JDK for a grid node

1. Navigate to the Configuration Manager for the grid from within the LifeCycle Manager.
2. On the Configuration Manager page, click Grid Properties.
3. In the Property column list, click Java Executable in the Misc. Node Properties section.
4. In the matrix, locate the column for the host whose JDK you want to change and click on the property value link for that column in the Grid-Wide row (first row).
5. Log in if necessary.
6. In the dialog box, enter the absolute path to the Java executable of the new JDK version (for example, **`C:/JDK1.6/bin/java.exe`**).

7 Click the Save button to close the dialog box and save your changes.

8 Persist the configuration changes by clicking the Save button in the upper left corner of the Configuration Manager.

9 Repeat this procedure for other hosts whose JDK you want to change.

10 For any nodes currently running, you will need to restart them to apply the changes. Any new nodes will use the new JDK when they are started.

Changing the Grid Service User

Use this procedure if you need the grid bootstrap service to run as a different user than the default user.

- Change the user assigned to the grid bootstrap service.

If the grid runs on multiple hosts, you will need to repeat this procedure for each host.

To change the user assigned to the grid service on Windows platforms

1 Access the Windows Server Manager on the server where the grid bootstrap service runs.

2 Under Configuration, select Services.

3 In the list of services, locate the grid bootstrap service. It will have a name in the format: **`Infor ION Grid Bootstrap -`** *`gridName`***`-`***`hostName`*.

4 Double-click the entry to open the Properties dialog box.

5 On the Log On property page, change the default user to the user you want the grid on this host to run as.

To change the user assigned to the grid service on Linux platforms

1 Stop the grid. At the command line, type

```
/grid_installation_dir/grids/gridName/StopHost
```

2 Stop the grid bootstrap service. At the command line, type

```
service infor_ion_grid_bootstrap_gridname_x.x.x.x stop
```

3 Change the service script to launch the bootstrap service as the new user. At the command line, type

nano /etc/init.d/infor_ion_grid_bootstrap_*gridname_x.x.x.x*
"/*grid_installation_dir*/resources/linux-amd64/jsvc" -debug -user *newUserName* -procname "infor_ion_grid_bootstrap_*gridname_x.x.x.x*" -pidfile "/var/run/infor_ion_grid_bootstrap_*gridname_x.x.x.x*.pid" -home "/usr/java/*jdk_Dir*" -cp

"/*grid_installation_dir*/resources/bootstrap-daemon-*x.x.x-x*.jar" com.infor.bootstrap.DaemonWrapper -baseDir "/opt/*grid_installation_dir*/"

4. Change the owner of the grid directory. At the command line, type

   chown -R *newUserName*:*newUserName* /*grid_installation_dir*/

5. Start the grid bootstrap service. At the command line, type

   ```
   service infor_ion_grid_bootstrap_gridname_x.x.x.x start
   ```

6. Start the grid. At the command line, type

   ```
   /grid_installation_dir/grids/gridname/StartHost
   ```

Configuring Network Access to the Grid

5

The section describes how you can examine and control the port usage for your grid, as well as fine tune the access to the grid through the context root URLs and through the use of web servers as proxy servers.

Ports for Client Applications

This section provides a general description of how client applications and other server applications connect to the grid through ports. More specific descriptions of this are in the installation guides for each application that needs to connect to the grid. Use this description to identify the correct ports for client applications to use to connect to the grid.

Grid applications do not expose ports themselves. Clients connect to ports opened in one of the grid routers. This is an important concept that enables grid applications to be moved between hosts and scaled out without confusing the connected clients.

It is possible for applications to bypass the routers and open ports themselves. However, they rarely do and, if they do, the correct way of configuring that application is described in the application's documentation. In that case, what is written here does not apply.

The routers may expose ports of different types:

- Grid - Access to the intrinsic communication protocol of the ION Grid
- HTTP and HTTPS - Access to web applications, web services, and REST
- Connection Dispatchers - Ports opened on behalf of grid applications with proprietary communication

Technically, routers are application neutral. A client may connect to any router in a grid as long as that router exposes the correct type of port on a network interface that is accessible from the client. So, given that an application client needs a particular type of port, it is possible to select any such port as long as the client can reach it. However, this should be described in the documentation of the client application.

Listing All Ports Exposed by Routers in a Grid

Use this procedure to see a list of all ports that are exposed, including their type and their address. A suitable port may be selected with the help of this list, but this should be explained better in the documentation of each application.

To list all ports exposed by routers in a grid

1 Navigate to the Configuration Manager.

2 Click the Advanced Configuration link.

3 Click the Defined Ports link.

Reasons the List of Ports May Be Incorrect

The ION Grid only knows about ports that are part of the grid's configuration and that it manages. Often it is those ports that clients should connect to but not always.

Typical examples:

- A web server may have been placed between the clients and the grid routers. In this case, clients that normally would have connected to an HTTP(S) port of one of the routers should now connect to the web server instead. Traffic will then be routed from the web server to the routers in a transparent manner unknown to both the grid and the client.
- Several grid routers may have been configured on different grid hosts and some form of network load balancer has been placed in front of them in order to achieve fault tolerance. In this case, clients should not connect to the ports that the grid is aware of (the router ports). Instead, clients should connect to the ports associated with the network load balancer.

What commonly occurs in these example cases (and there may be others) is that the ION Grid has no way of knowing of these external ports. Since the grid does not know about them, the grid is not in a position to indicate to clients that the external ports should be used. It is entirely up to the person who installs the client application to know of the existence of the above situations and connect to the correct ports.

Note: Although the ION Grid may be unaware of the correct ports to use, there is a mechanism in the ION Grid that you can use to determine how clients connect. This mechanism, however, must be configured manually. Also, the clients have to be coded to make use of this information. For more information, see "Working With Context Root Discovery URLs" on page 46.

How to Use HTTP and HTTPS Ports

Applications that provide web applications and web services expose those through the HTTP or HTTPS ports of grid routers. The applications and web services are separated by the context roots. The context roots are part of the URL to each resource.

One way to get more information about those services is to point a web browser to one of the defined HTTP(S) ports. See "Listing All Ports Exposed by Routers in a Grid" on page 45 for instructions about how to list the ports. In the list, locate an http(s) port and note the address that is used by the router. Point a web browser to the port using the listed address information, for example, http://<address>:<port>.

The displayed web page lists all web applications and web services. For each web application there is a link to that application. The links for the web services point to another information page that lets you display the WSDL file associated with the web service. This information is important when developing clients to the web services.

Working With Context Root Discovery URLs

The ION Grid provides a service that enables clients to discover the context root URLs for HTTP-based grid applications. This service provides a document in XML or JSON format that shows the accessible web facets of the grid and its applications. Clients can then discover how to connect to grid applications by having only the HTTP address to a grid router.

The following is an example of the raw XML document:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<grid xmlns="http://schemas.lawson.com/grid/http/info" version="10.1.8.0 4" name="ProductionGrid">

  <application typeName="SYSTEM" name="SYSTEM">
    <module accessible="true" name="WebAccess">
      <contextRoot type="Rest Service" href="grid">
        <url authType="none">http://server1.mycompany.com:10080/grid/</url>
      </contextRoot>
    </module>
  </application>
  <application typeName="MI-WS" name="M3-API-WS">
    <module accessible="true" name="MI-WS">
      <contextRoot type="Web Service" href="m3api">
        <url authType="none">http://server1.mycompany.com:10080/m3api/</url>
      </contextRoot>
      <contextRoot type="Rest Service" href="m3api-rest">
        <url authType="none">http://server1.mycompany.com:10080/m3api-rest/</url>
      </contextRoot>
    </module>
  </application>
  <webstart>http://server2.mycompany.com:29904/</webstart>
</grid>
```

This information in the XML file repeats some of the same information you can view in HTML format at **http(s)://***server***:***port***/grid/info.html**. To retrieve the URL of a specific grid application, you need to provide three keys: the application type, module and context root type. If multiple instances of an application have been installed, you must also provide the application name when querying for the URL.

The application type, module, and context root type are pre-configured in the application and cannot be changed. The application name is set at installation. The context root type will be Web Service, Rest Service, or Web Application.

In the info.html file, the values for these are in the Application, Module, and Type columns. In the XML document, the values are in the application, module, and contextRoot elements.

Modifying Context Root Discovery URLs

The information documents, by default, describe the paths to the different applications' context roots through the same router port and scheme as the current request is using (by utilizing the host header of the request). This may not be desirable in all cases. For example, you may want to present different context root discovery URLs when traffic from external sources is to be routed through a web server fronting the grid and traffic from internal sources should connect directly to the grid. Another example where you may want a different context root discovery URL is when an application should be defaulted to https even though the discovery-request is not.

To modify the context root discovery URLs, you add or modify entries for the Context root discovery urls grid property. This property is a map where the key is a contextRoot (optionally together with a router) and the value is a list of addresses (optionally together with SSL authentication mode). Multiple entries in the list are a preferred order that enables fallback addresses in the discovery document.

The format for entries in the Context root discovery urls grid property is as follows:

- Key: **`[RouterName:]contextRootName`**
- Value (list of): **`[sslMode:]httpAddress`**

  Valid sslMode values are:

  - **`none`** - applicable for HTTP only
  - **`server`** - default for https, the server presents certificates to identify itself to the client
  - **`client`** - the server presents certificates to identify itself to the client and the client may choose to identify itself with a certificate
  - **`required`** - the server presents certificates to identify itself to the client and the client is required to identify itself with a certificate

To modify or override context root discovery URLs

1. Navigate to the Configuration Manager.
2. Click the Grid Properties link.
3. Locate the property named Context root discovery urls in the Grid Http Discovery Services section.
4. Click the link in the Value column.
5. In the resulting dialog box, click Add New Entry and add values for the context root (and optional router) and for the addresses (and optional SSL authentication modes).

6 Click Save to close the dialog box, click the Save button on the Grid Properties page, and then click Save to confirm the changes.

Example 1: A Web Server or Two in Front of the Grid

Context root discovery urls value:

```
TestRouter:m3api-rest -> client:https://m3rest.example.com, http://
restportal.example.com/m3
```

This grid property values means that when the discovery service is accessed through the TestRouter grid router, the ordered list of access addresses for the context root m3api-rest will be https://m3rest.example.com (https mode "client") and then http://restportal.example.com/m3.

Example 2: Use of HTTPS as Preferred Scheme

Context root discovery urls value:

```
m3api -> https://server1.mycompany.com:10081/m3api/
```

This grid property values means that no matter which router the discovery request comes from, the context root m3api should be accessed through the TestRouter on https (the TestRouter has that specific address and https port).

Using a Web Server as a Proxy Server to Redirect HTTP Traffic to the Grid

Use this procedure if you want to put a web server in front of the grid so that you expose the HTTP resources of the grid through a DMZ.

The grid can be used with or without SSL offloading. If SSL offloading is used, the client SSL is terminated at the HTTP server traffic between the HTTP server, and the grid is forwarded as HTTP to the grid. If you are not using SSL offloading and have not bought an SSL certificate for your grid host, you need to export a grid certificate (for information on certificates, see the *Infor ION Grid Security Administration Guide*).

To set up IIS as a proxy server

1 Install the Application Request Routing plug-in. This is available at http://www.iis.net/download/ApplicationRequestRouting

2 Create a server farm.

3 Add a server for your grid router with HTTP and HTTPS ports defined to your server farm. Make sure to put the correct FQDN of the SSL certificate for the host in the address field as well as the correct HTTP and HTTPS ports under advanced.

4 On the Server farm/Proxy, make sure "Reverse rewrite host in response headers" is checked.

5 Decide if you want to use SSL offloading and make the correct configuration under Server farm/Routing Rules.

6 If you are not using SSL offloading, you need to import the Grid/Host SSL to make IIS trust the grid root certificate.

Using a Load Balancer with the Grid

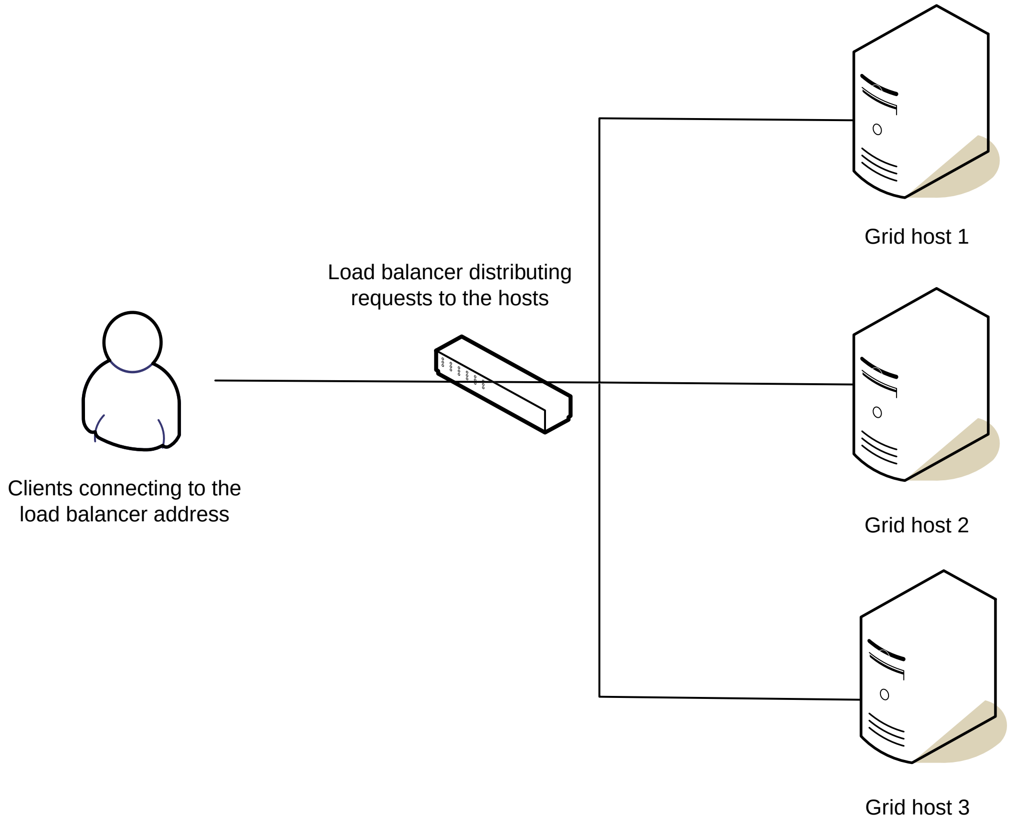
In a grid the primary host always has a configured router (by default) which is named Default Router. All client traffic either via a browser or legacy application will be directed through this one single router.

When utilizing more than one host in the grid, it can make sense, depending on the application(s) used, to spread this client loading amongst the hosts by defining additional routers on other hosts. Similarly, it is feasible to create additional routers on the same host, distributing the load across more nodes (JVMs) and therefore potentially increasing throughput. In either case, you will have more than one entry point which the clients would need to be aware of, and to give a list of server addresses and ports to a client would be difficult to administer and make future changes onerous.

In these instances, the use of a load balancer will isolate the clients from the infrastructure. This means that the clients only have one endpoint to connect to and the load balancer holds the configuration of the entry points in the grid behind and automatically distributes the client load according to rules. Furthermore, the SSL certificate can be installed at the load balancer, meaning no requirement to distribute the SSL certificates of all the grid hosts to the clients. Again, this is a deployment choice and other options exist.

The following diagram aims to illustrate the principle of a load balancer:

Figure 1. Illustration: Basic load balancer setup


Grid host 1
Load balancer distributing requests to the hosts
Grid host 2
Clients connecting to the load balancer address
Grid host 3

Various load balancers exist in the market today and are primarily in two forms:

- Hardware

  This is a physical or virtual appliance with the sole purpose of load balancing. Many of these appliances can handle SSL offloading, meaning that SSL traffic is decrypted by the appliance, easing the load on the grid router.

- Software

  This is software application used to load balance. Apache HTTPD is a good example of such an application and supports creating load balancing clusters with definable rules on how the load should be distributed. As an open source project, there is potentially no cost with this option.

Configuration of the load balancer will depend on the type and product selection. As such, no instructions can be documented, though the following basic steps would be followed to use a load balancer with the grid

Basic Load Balancer Setup Steps

1 Create additional router(s) in the Grid.

These could either be on the same host or in the case of a multi-host Grid on separate hosts though the choice depends on the desired effects.

2 Configure the load balancer, adding each of the Grid routers to the cluster.

3 Configure the clients to utilize the load balancer as their connection point.

One additional consideration is that context root discovery URLs may need to be configured to redirect to the load balancer, depending on the clients and whether discovery is required for any of the applications deployed. For more information about context root discovery URLs, see "Working With Context Root Discovery URLs" on page 46.

Managing the Grid

6

This section describes common administrative tasks, just as those for stopping and starting the grid. Some of these tasks are performed from within the LifeCycle Manager while others can be performed programmatically from outside the LifeCycle Manager.

- "Managing the Grid Through the LifeCycle Manager" on page 52
- "Managing the Grid Programmatically" on page 56

Managing the Grid Through the LifeCycle Manager

- "Putting Applications or Parts of the Grid in an Off-Line State" on page 52
- "Stopping an Individual Grid Node" on page 54
- "Starting New Grid Nodes (Application Instances)" on page 55

Putting Applications or Parts of the Grid in an Off-Line State

In the ION Grid it is possible to prevent new client requests from being accepted by the server applications. This state is called off-line. When some part of the grid is off-line, it will no longer accept new requests but ongoing requests will be allowed to finish. This is ideal for situations when parts of a grid need to be taken down but you do not want to simply kill nodes since that would terminate existing processes performed there. In this situation, you may start by having the grid enter an off-line state. When all ongoing requests are completed, you can stop the nodes or the relevant parts.

A typical example is that you want to stop an application in order to perform some maintenance. Just stopping all application nodes could perhaps result in some ongoing processing being terminated prematurely. The solution is to first put the application in an off-line state. This prevents clients from calling the application with more requests. When all ongoing requests are finished, you may safely stop the application.

An important implication of an application being in an off-line state is that nodes will no longer be automatically started even if the application has bindings that are configured to maintain a minimum number of running nodes. If an application has bindings of this type, it will be impossible to stop the application without also putting it in an off-line state. This is because not doing so would cause the grid to automatically start new nodes to replace the stopped one.

The following entities may be put in an off-line state:

- The entire grid
- Individual hosts within the grid
- Applications
- Individual application nodes

The following procedures describe how to put different parts of the grid in an off-line state. Getting them on-line again is done in the same way.

To put the grid in an off-line state

Note: This will put the entire grid and all running applications in an off-line state. The typical reason for doing this is that you intend to stop the entire grid for maintenance and you want to allow applications to finish executing in a controlled manner.

1 Navigate to the Topology Overview page in the Grid Management pages.

2 Click on the grid link close to the upper left corner. (This link is the one with the grid's name following the word "Grid".)

3 Click the Set Offline link.

To put hosts in an off-line state

Note: This will put all the nodes on this host in an off-line state regardless of what application they belong to. The typical reason for doing this is that you intend to remove this host from the grid and you want to allow application nodes to finish executing in a controlled manner.

1 Navigate to the Topology Overview page in the Grid Management pages.

2 In the list, click on the host name link for the host that should be off-line.

3 Click the Set Offline link.

To put applications in an off-line state

> **Note:** This will put all the nodes belonging to a particular application in an off-line state regardless of where the nodes are running. It will also prevent the grid from automatically restarting nodes even if they have been configured for that. The typical reason for doing this is that you intend to stop an application for maintenance but you want to allow application nodes to finish executing in a controlled manner.

1. Navigate to the Topology Overview page in the Grid Management pages.
2. At the bottom of the page, click on the application link for the application that should be put in an off-line state.
3. Click the Set Offline link.

To put individual application nodes in an off-line state

> **Note:** This will put only one node in an off-line state. The typical reason for doing this is that you intend to stop this node and you want to allow ongoing work in the node to finish executing in a controlled manner.

1. Navigate to the Topology Overview page in the Grid Management pages.
2. In the list, click on the node name link for the node that should be off-line.
3. Click the Set Offline link.

Stopping an Individual Grid Node

Normally, you do not stop individual application nodes. Instead, you stop an entire application and, as a result, all nodes belonging to the application will be stopped. However, there are situations when you want to stop an individual application node. Examples are:

- A node is experiencing problems and you need to stop it, perhaps with the intention of starting a new node to replace the stopped one.
- The application has been started in multiple instances (nodes) but all that capacity is no longer needed, so some of the nodes can be removed in order to free resources.
- You want to clear a particular host in a multi-host grid from all nodes in order to perform some maintenance on that host.

Considerations Before Stopping a Node:

- Should the node be put in an off-line state first?

- If the application node is the only one of its kind, the application may stop working. Should a new similar application node be started first before the node is stopped?
- Applications are implemented differently. Some applications manage their nodes themselves. Consult the documentation of the application. Does the documentation recommend some alternative procedure to stopping the node in this case?

To stop an individual grid node

1 Navigate to the Topology Overview page in the Grid Management pages.

2 In the list, click on the stop button for the node that should be stopped.

3 Click OK in the dialog box.

Starting New Grid Nodes (Application Instances)

Application instances are started by launching a binding that targets the correct node type. Normally you do not start individual application nodes. Instead, you start an entire application and, as a result, all nodes belonging to the application will be started. However, there are situations when you want to start an individual application node. Examples are:

- An application is experiencing increased load and you want to start a new server application instance, perhaps on a new host with spare capacity, in order to cope with the new situation.
- You intend to stop an existing node and you want to start a new alternative node first so that the operations of the application will not be disturbed when you stop the node.
- You want to move the execution of this application from one host to another. This would be done by starting new nodes on the new host combined with stopping the old nodes on the original host. Note that not all applications support this operation. Consult the documentation for the application.

Considerations before Starting a Node:

- Does the application support manually starting new nodes? Some applications manage their own nodes and they should not be started manually.
- Some applications do not support several instances running simultaneously. If that is the case with the application you are starting, existing nodes may have to be stopped first. However, ideally, applications should be written to support this.
- Always consult the application's documentation on what is supported in each case.

To start grid nodes

1 Navigate to the Topology Overview page in the Grid Management pages.

2 In the list, each host has a start button. Click the start button for the host that you want the new node to run on.

3 In the dialog box, select the binding you want to start.

What If the Binding to Start Does Not Appear in the Dialog Box?

The typical reasons for not finding a particular binding when trying to launch it on a host are:

- The binding is not configured to be able to run on that host.
- The binding is configured to allow a maximum number of simultaneously running nodes of this type and that maximum number of nodes is already running in the grid.

  For information on configuring bindings, see "Configuring Bindings" on page 34.

Managing the Grid Programmatically

Programmatically Operating on the Grid using REST

Use this procedure to programmatically operate on a grid. The ability to programmatically act upon a grid depends on the fact that the ION Grid exposes a number of status documents and operations using REST. The REST-enabled APIs are easy to call programmatically and may be used in various scripting scenarios that operate on the grid or on individual parts of it, for example, applications or nodes).

The REST services are defined by a WADL file in the same way that web services are defined in a WSDL file. (For more information on WADL, see http://www.w3.org/Submission/wadl/ and https://wikis.oracle.com/display/Jersey/WADL.)

The WADL file describes the set of operations that is exposed and the schema of the data that is passed as requests and returned as responses (the inclusion of types and schemas is defined as extended wadl in the Jersey documentation link above). The WADL file is typically imported into some external tooling that will generate code (in different languages) that makes it easy to implement calls to the exposed REST services.

The REST services are accessed via any of the HTTP(S) ports that are defined in the grid routers. The WADL file is also accessible via any of those ports. For information about how to identify the HTTP and HTTPS ports, see "Listing All Ports Exposed by Routers in a Grid" on page 45.

Accessing the application.wadl File

The application.wadl file is located at http(s)://*server*:*port*/grid/application.wadl.

Security

Most of the exposed REST services that use the HTTP POST method will require authentication. Authentication may be performed using username/password or certificates. For information about how to generate a client certificate, see the *Infor ION Grid Security Administration Guide*.

REST Services Exposed by the ION Grid

The available REST services are shown in the tables below. Note that the REST services that use the HTTP GET method can easily be tested from any web browser. For example, assume that the router that exposes an HTTP port on the following address: http://server:20005. In this case, you can call the following from any browser:

- http://server:20005/grid/status
- http://server:20005/grid/nodes
- and so on

Grid-wide REST Services

Resource	Type	Description	Sample Path
info	GET	This document contains basic grid version information and context roots defined by applications installed into the grid.	/grid/info
status	GET	The status document contains an entry for each deployed application. The information for each application contains such things as global status, log warnings and error counts, memory status, CPU status, and offline state.	/grid/status
hosts	GET	This returns status information for hosts that are part of the grid. Information examples are OS version, system time, offline state, and memory usage.	/grid/hosts
ports	GET	This returns information about all ports used by the grid and what purpose they serve.	/grid/ports

Resource	Type	Description	Sample Path
bindings	GET	This returns information about all bindings that are defined in the grid. This information may be used when starting nodes using the startnode service: /grid/host/*hostName*/ startnode?binding=*bindingName*	/grid/bindings
nodes	GET	This returns status information for all running nodes, routers, and the registry. The returned information includes, for example, log warning and error counts, memory usage, CPU usage, and offline state.	/grid/nodes
offline	POST	Puts the grid in an offline state.	/grid/offline
online	POST	Puts the grid in an online state.	/grid/online
stop	POST	Stops the grid with the given grace period and reason.	/grid/stop /grid/ stop?graceSeconds=20 &reason=*reasonDescription*
resetLogCounters	POST	Resets the log counters for the grid.	/grid/resetLogCounters

Host-related REST Services

Resource	Type	Description	Sample Path
offline	POST	Puts the host in an offline state.	/grid/host/*hostName*/offline
online	POST	Puts the host in an online state.	/grid/host/*hostName*/online
startnode	POST	Starts the node using a specified binding on the specified host.	/grid/host/*hostName*/ startnode?binding=*bindingName*

Application-related REST Services

Resource	Type	Description	Sample Path
start	POST	Starts the specified application. Optionally, you can specify if the application should be put in an offline state after starting it.	/grid/application/*appName*/ start?offline=false

Resource	Type	Description	Sample Path
stop	POST	Stops the specified application. Optionally, you can specify if the application should be put in an offline state after stopping it. Also, it is possible to say that the call should block until the application has stopped completely.	/grid/application/*appName*/ stop?offline=true &block=true
offline	POST	Puts the specified application in an offline state.	/grid/application/*appName*/ offline
online	POST	Puts the specified application in an online state.	/grid/application/*appName*/ online
status	GET	Return status information for the specified application. The returned information includes the global state and offline state as well as status information for each grid node that belongs to this application and information for each host that the application is deployed on.	/grid/application/*appName*/ status

Node-related REST Services

Resource	Type	Description	Sample Path
log	GET	Returns the log file for the specified node (nodes are identified by their JVM ID). Optionally, the log may be filtered on a supplied string. If the argument **`file=true`** is given, the entire log file will be returned (not only the part that is kept in memory by the node).	/grid/node/*JVMID*/log /grid/node/*JVMID*/ log?filter=ERROR /grid/node/*JVMID*/ log?file=true
stop	POST	Stops the specified node (JVMID). It is possible to specify a grace period before the node is forcefully stopped. A grace of -1 means that it is up to the grid to decide on the grace period. The reason that is given for the stop is logged in the node log before the node is stopped	/grid/node/*JVMID*/ stop?graceSeconds=-1 &reason=*reasonDescription* &block=true
offline	POST	Puts a node (JVMID) in an offline sate.	/grid/node/*JVMID*/offline

Resource	Type	Description	Sample Path
online	POST	Puts a node (JVMID) in an online state.	/grid/node/*JVMID*/online
nodeWeight	GET	Returns the node weight of a node (JVMID).	/grid/node/*JVMID*/ nodeWeight
nodeWeight	POST	Sets the node weight of a node (JVMID).	/grid/node/*JVMID*/ nodeWeight?weight=100
resetLogCounters	POST	Resets the log counters for a node (JVMID).	/grid/node/*JVMID*/ resetLogCounters
status	GET	Returns status information for a node (JVMID). The returned status contains such things as log warning and error counts, memory usage, CPU usage, and offline state.	/grid/node/*JVMID*/status

Configuration-related REST Services

Resource	Type	Description	Sample Path
config/ binding/ update	GET	Updates a binding in a grid. The binding name is required. Specifying a host is optional. You can also list multiple hosts and you can indicate all hosts with an asterisk (*). The initial, max, and min provide values for the initial binding count, maximum binding count, and minimum binding count.	/grid/config/binding/ update?name=*binding_Name* &host=*host_Name* [&host=*host_Name*. ..]&initial=*initial_value*&max= *max_value* &min=*min_value*

Resource	Type	Description	Sample Path
config/host/add	GET	Adds a host. The fully qualified domain name is required. The host name is optional. If you do not supply it, it will be derived from the FQDN. The agent port is optional. If you do not supply it, the primary host agent port is used. If you supply a port, you must ensure it is free. No validation is done. The optional active value indicates the initial state of the host when deployment is complete. Valid values are true and false. The optional failover value indicates whether this host should provide failover capabilities for the registry. If you do not set this, it will be set to true if there is no existing registry failover host. Valid values are true and false.	/grid/config/host/add?fqdn=*FQDN* &port=*agentPort*&host= *host_Name* &active=*active_value*& failover=*failover_value*
config/host/remove	GET	Removes a host. The host name is required.	/grid/config/host/remove?host=*host_Name*
config/host/update	GET	Activates/deactivates a host. The host name and the active value are required. Valid values for the active value are true and false.	/grid/config/host/update?host= *host_Name*&active=*active_value*
config/host/installer	GET	Downloads the installer jar file. The host name is required. It is the name of the host you want to download the bootstrap jar for.	/grid/host/installer?host=*host_Name*

Using the Grid Script Client to Manage the Grid

The ION Grid includes a built-in utility class that can help you control the grid and grid applications as well as query the grid for status information. It is primarily built to be used from the command line or a scripting language

For example, if you need to schedule backups, it is useful to have a script that shuts down an application programmatically, sets the application offline, performs the maintenance tasks, starts the application, and finally queries the application for its status to make sure it is running correctly.

Note: An alternative to using this script utility is to operate programmatically on the grid using REST. For more information, see "Programmatically Operating on the Grid using REST" on page 56.

To use the Grid Script client

1. Copy the grid-core.jar to a directory of your choice. This jar file is located in an installed grid's resources directory.
2. Create a client certificate with the correct credentials. This is needed because the utility is a grid client that will perform administrative tasks in the grid.
   - a In LifeCycle Manager, click Manage Security on the dashboard for the target grid.
   - b Select Generate client keystore and click Next.
   - c Select a name, password, and target directory for your keystore (certificate file). Make sure to specify "grid-admin" in the role list.
   - d Make a note of a grid agent port and address or copy the topology file from the installed grid's config directory.
3. To run a scripting command, use the following command line:

```
java –cp grid-core.jar;grid-ui.jar com.lawson.grid.util.ScriptingClient
agentAddress agentPort -ks keystoreFileName keystorePassword command target
```

Supported Commands

The following commands are supported. The result will be presented on standard out and the return code will be 0 unless there is an ERROR message. In the case of an ERROR, the return code will be 1 and all messages will be presented on standard err. If you get an ERROR result the command has failed to complete

Command	Results and Descriptions
startgrid	SUCCESS - grid successfully started TIMED_OUT - grid not started within timeout period

Command	Results and Descriptions
stopgrid	SUCCESS - grid successfully stopped TIMED_OUT - grid not stopped within timeout period
gridoffline	SUCCESS - grid is set offline
gridonline	SUCCESS - grid is set online
gridstatus	REGISTRY_STOPPED - the registry is stopped ADMINROUTER_STOPPED - the admin router is stopped GRIDAGENT_STOPPED - reported once for each grid agent that is stopped GRIDAGENT_RUNNING - reported once for each grid agent that is running OK - registry, router, and all grid agents are running STOPPED - when only grid agents are running OFFLINE - the grid is offline ONLINE - the grid is online NO_OFFLINE_STATUS - unable to get offline status (the grid might be stopped)
hostoffline *hostName*	SUCCESS - host is set offline
hostonline *hostName*	SUCCESS - host is set offline
hoststatus *hostName*	GRIDAGENT_RUNNING - the grid agent is running GRIDAGENT_STOPPED - the grid agent is stopped OFFLINE - the host is offline ONLINE - the host is online NO_OFFLINE_STATUS - unable to get offline status for this host (the grid might be stopped)

Command	Results and Descriptions
applicationstatus *application*	OK - the application global status returns OK NOT_OK - the application global status returns NOT OK COMMENT - each line of the application status report is reported as a COMMENT line NO_APPLICATION_STATUS - unable to get application status STOPPED - the application is stopped OFFLINE - the application is offline ONLINE - the application is online NO_OFFLINE_STATUS - unable to get offline status for this application (the grid might be stopped)
applicationoffline *application*	SUCCESS - application is set offline
applicationonline *application*	SUCCESS - application is set online
startapplication *application*	SUCCESS - application is started TIMED_OUT - application not started within timeout period
stopapplication *application*	SUCCESS - application is stopped TIMED_OUT - application not stopped within timeout period

Troubleshooting

7

Introduction to Troubleshooting

The first approach to troubleshooting is to use the techniques for monitoring the grid (see "Grid Management Pages for Monitoring the Grid" on page 15). As described there, you typically view log files and change log levels in order to gain more information. When this is not enough, you can consider the state of the hosts (machines) and anything related to the operating system on them. Are they configured optimally and do they have enough resources at hand. For example, are the disks full? Do you need more memory? Are the CPUs stressed? Are there network issues?

Investigating Problems or Issues

- "Viewing Threads" on page 66
- "Getting a Heap Dump or Thread Dump Via the Grid Agent" on page 66
- "Reviewing Each Application's Configuration" on page 67
- "Diagnosing Network Issues" on page 67
- "Gathering Information When Reporting a Problem" on page 68

Viewing Threads

If a grid node is using an unusually high percentage of the CPU for an extended period of time or if a node show signs of being unresponsive, you may consider looking at the threads in the node in order to see what is happening in the node. It could be a situation where a thread is in an endless loop or it could be in a deadlock situation.

To view threads

1. Navigate to the Topology Overview page in the Grid Management pages.
2. In the list, click on the name link for the node with the suspected problems.
3. Click on the Advanced link.
4. Click on the Threads link.
5. Click on the Threads link that forms the root of the tree of threads The resulting window will list all threads and what they are doing at the moment and what state they are in.
6. Refresh the window in order to see what changes over time.

Getting a Heap Dump or Thread Dump Via the Grid Agent

When troubleshooting problems, it may be useful to get a heap dump or a thread dump from individual grid nodes (JVMs). As described in "Viewing Threads" on page 66, you can view threads interactively in the Grid Management UI. However, if the node (JVM) that you want to get the thread dump from is experiencing severe problems, it may not be possible to retrieve it interactively. Described below is an alternative way of retrieving the thread dump via the grid agent. This mechanism also provides you with the possibility to download a heap dump.

Note: This feature is only available for grid nodes running a Sun/Oracle JVM.

To get a heap dump or thread dump via the grid agent

1. Navigate to the Topology Overview page in the Grid Management pages.
2. Each host has a separate grid agent. Click on the Agent link in the Name column for the grid agent that is running on the same host as the grid node you want to get the dump files from.
3. Click the Advanced link.
4. Click the Process Utilities link.

   A list of grid nodes running on this host will be displayed.

Note: If the link is missing, you need to enable the "Show Developer Features" option. Do this by clicking the small icon depicting a screwdriver and wrench in the upper right corner of the management pages. In the dialog box that appears, make sure to check the "Show Developer Features" check box.

5. Identify the node you want to work with and click either the Heap Dump link or the Thread Dump link, depending on what you want.
6. After you click on either of the two links, a new link will be added at the bottom of the page with a link to the dump file that was generated. Clicking on that link to download the dump file and save it locally on disk.
7. To view the thread dump, open it in any text editor.
8. To view the heap dump, you need a tool that can view the specific binary format of the dump files. Sun/Oracle provide a tool called jhat as part of their JDK that can be used to view this kind of file. After downloading the heap dump, use the jhat tool in the JDK to view the heap dump.

Reviewing Each Application's Configuration

Problems may result from applications being incorrectly configured. One way of getting an indication of such problems is to view the configuration of each application.

To review an application's configuration

1. Navigate to the Configuration Manager.
2. Click the Applications link.
3. For each application in the list do the following:

   a. Click on the application name link in the list.
   b. If a suggested action to perform is displayed at the top of the page, you may have a problem. Often the problem can be fixed by clicking on the Fix this problem link next to the suggested action. However, consult the application's documentation in each case.
   c. In addition to looking for a suggested action, you may review the information on this page. Look for something that seems strange and consult the application's documentation in each case.

Diagnosing Network Issues

A multi-host grid is dependent on a fast network in order to perform well. A Gigabit network between grid hosts is strongly recommended. Strange intermittent problems on one host at the same time that application instances on another host have no problems may be an indication that the host is having network problems.

Diagnosing the network may be done with tools external to the ION Grid but the grid provides ways of measuring the network performance and it is always a good idea to perform that test when network problems are suspected.

To diagnose network issues

1. Navigate to the Topology Overview page in the Grid Management pages.
2. Click the Advanced link at the top of the page.
3. Click the Network Diagnostics link.
4. You will be presented with two tests. Run both and look for results that look unusual.

   It is hard to give general guidelines for these tests as far as what is considered good results and what is considered bad results. The recommendation is to perform the tests from time to time in order to learn what is normal in each instance of the grid. By learning what is normal in your case, you will be able to spot when something is out of the ordinary.

Gathering Information When Reporting a Problem

When reporting a problem, it is very helpful if as much information as possible is included. Log files with errors and warnings are particularly helpful. For instructions on how to download log files, see "Viewing Log Files" on page 21. It is also helpful to capture a screenshot of the Topology Overview page in the Grid Management pages. If the problem is related to a specific application, gathering as much information as possible about that application is also helpful.

If the grid has operational problems, but at least parts of it are still running, it is always best to generate a Grid Status Report as described in "Generating a Grid Status Report" on page 25 and include that when reporting the problem.

It the grid is experiencing so many problems that it can’t be started, it will not be possible to generate the Grid Status Report. You will also not be able to view log files and other information sources the normal way. In this case, log files and other information must be retrieved manually from disk. See "The Grid and How It Is Stored on Disk" on page 75 for instructions on how to locate the grid configuration area. In the configuration area, all log files and configuration files may be found.

Disaster Recovery

Disaster Recovery

By "disaster" is typically meant fatal hardware failures such as failed disks, network adapters, or entire machines. Such failures can lead to the loss of major components of a grid.

Recovering from a Missing Registry Host

If the host containing the registry is missing, you have a serious problem. It must be resolved as soon as possible. Without the registry, you cannot start new application nodes, for example. However, existing application nodes will be able to continue executing client requests as long as the clients are able to connect to the grid using a router on another host belonging to the grid.

Note: One way of avoiding this situation is to cluster the registry host.

The administrative router is also located in the same host as the registry, so that router is also missing. The administrative router is not critical in itself, but since you cannot get to it you cannot use it to make any configuration changes that would resolve the problems. The only tool that can resolve this situation is the LifeCycle Manager.

How to recover from this situation depends on the topology of the grid. If the grid consisted of only one host, the entire grid is lost and the way forward is to install a new grid and reinstall the applications that were deployed in that grid. Consult the documentation of each application for backup and restore procedures. The only alternative to reinstalling the grid is to have some form of machine backup, perhaps in the form of a captured image, which can be restored. The rest of the procedures described below are not relevant in this case since you simply recreate the entire grid.

If, however, the grid has several hosts, you need to move the registry to one of the remaining hosts. You can only move the registry to a non-transient host, so the procedure to move the registry includes an initial task to check if all of the remaining hosts are transient. If all are transient, you will then need to reconfigure one of them as non-transient. If at least one remaining host is non-transient, the recommendation is to move the registry there.

The procedure to recover from a missing registry host includes the following main tasks:

1. "Stopping the Grid" on page 69
2. "Moving the Registry" on page 70
3. "Removing the Missing Host" on page 70
4. "Dealing with Deployed Applications or Parts of Applications that Only Existed on the Removed Host" on page 71
5. "Replacing Missing Routers and Connection Dispatchers" on page 71
6. "Removing a Missing Host from the Configuration" on page 72

Stopping the Grid

At this point you are (or should be) in the following situation:

- You have a multi-host grid.
- You have verified that at least one of the remaining hosts is non-transient.

If the above is true, you may proceed to move the registry by first stopping the grid.

To stop the grid

1. From the LifeCycle Manager, select the Applications tab in the left pane.
2. In the tree, locate the grid you have problems with and double-click on it.
3. In the right pane, select the Tasks tab.
4. Click Stop Grid.

   LifeCycle Manager will still try to communicate with the missing hosts, so this operation may take some time but will eventually succeed.

Moving the Registry

After you stop the grid, you can move the registry to another host.

To move the registry

1. From the LifeCycle Manager, double-click on the missing registry host in the tree to the left.
2. In the panel to the right, select Move Registry.
3. At this time LCM may complain about not being able to run scripts and not being able to connect to the missing host. Under the circumstances this is expected. Acknowledge any error dialog box by clicking OK. A Move Registry dialog box will be shown.
4. In the Host Name combo box, select the host you want to move the registry to and click Next. If the Host Name combo box includes no alternative hosts to move the registry to, it is because there are no other hosts in the grid.
5. Accept the default values given for ports and address or change them to something that you want.
6. Click Next (may take some time).
7. Click Finish. At this time LCM may complain about not being able to run scripts and not being able to connect to the missing host. Under the circumstances this is expected. Acknowledge any error dialog box by clicking OK.
8. In the tree to the left, verify that the host you selected is now designated as being the registry host. If so, the registry has been moved successfully.

Removing the Missing Host

After you have moved the registry, you can remove the missing host from the configuration of the grid.

To remove the missing host

1 From the LifeCycle Manager, double-click on the missing host in the tree to the left.

2 In the panel to the right, select Remove Host.

3 A list of applications that was deployed on the missing host is listed. Take note of those applications since you may have to verify that they still are operational after the removal of this host.

4 Click Finish. At this time LCM may complain about not being able to run scripts and not being able to connect to the missing host. Under the circumstances this is expected. Acknowledge any error dialog box by clicking OK.

5 Verify that the missing host is now removed from the tree to the left

Dealing with Deployed Applications or Parts of Applications that Only Existed on the Removed Host

If an application was only deployed on the host that you have removed, that application must be installed again. Depending on the application, there may have been information stored on disk that is now lost. Consult the documentation of each application for backup and restore procedures.

Replacing Missing Routers and Connection Dispatchers

The host that you previously removed may or may not have contained one or more routers in addition to the administrative router that most likely was on the removed host. When you used LifeCycle Manager to move the registry, the administrative router was also moved so that one is already taken care of.

However, if there were other routers on the removed host, you have to consider how to replace them. One important thing to remember is that clients may connect to any router in a grid as long as that router is exposing the right kind of port. So, if you have lost a router that exposes an HTTP port but there is another existing router that exposes an HTTP port, you are not required to start any new routers. In this case, you simply must reconfigure the clients to connect to the other (existing) router instead of the old lost one.

This shows that the goal here is actually not to replace routers, but to replace lost ports. If the lost ports exist in other routers, you do not have to replace them. You just need to reconfigure clients to connect to the existing alternative ports. Whenever you are in doubt about what to replace or not replace, the recommendation is the following procedure:

To replace a router

1 Add a new router as described in "Editing and Adding Routers" on page 92. Give it a name, port, HTTP port, and HTTPS port as well as an external address.

2 Configure certificates related to the HTTPS port as described in "Generating Client Certificates from LifeCycle Manager" and "HTTPS and SSL Certificates" in the *Infor ION Grid Security Administration Guide*.

3. In order to find lost connection dispatchers, start by navigating to the Configuration Manager.
4. Click the Applications link
5. For each application in the list, do the following

   a. Click on the application name link in the list.
   b. If, at the top of the page, a suggested action to perform is displayed saying that a connection dispatcher is missing, click on the Fix this problem link next to the suggested action. Using the displayed dialog box, add the connection dispatcher to the new router and give it a port number. Consult the documentation for each application for details.
   c. Return to the list of applications and repeat the previous steps until all applications have been dealt with.

6. Save the new configuration if changes have been made.

At this point you have in one way or the other replaced all missing ports. Still, clients need to be reconfigured in order to find the new ports. Even if you gave the ports exactly the same port numbers, they are still on a new IP address, so clients need to be reconfigured. Consult the documentation for each client application that is affected.

Note: The word "clients" here refers to clients to the grid. These clients may still be server side products. For example, if a web server is placed between the actual clients and the grid, it is the web server that needs to be reconfigured and not the actual clients. Likewise, if some form of network load balancing is put in front of routers, it is the network load balancer that needs to be reconfigured, not the clients.

Removing a Missing Host from the Configuration

Once you remove the missing host from the configuration, you can push the new configuration to the remaining hosts.

To remove a missing host from the configuration

1. Double-click on the grid host in the Applications tab of the left pane in LifeCycle Manager.
2. In the Host Maintenance tab of the right pane, click Remove Host.
3. A confirmation window is shown. Complete the wizard and click Finish. At this time LCM may complain about not being able to run scripts and not being able to connect to the missing host. Under the circumstances this is expected. Acknowledge any error dialog box by clicking OK
4. Double-click on the grid in the left panel of LifeCycle Manager.
5. In the Grid Maintenance tab in the right pane, select Synchronize Grid.
6. In the same pane, select Grid Management Pages and verify that the host is removed.

Note: At this point you have removed the missing host and the grid itself is back at a fully operational state. However, there may have been some resources that only existed on the missing host and they need to be replaced. As mentioned above, things may be missing that only existed on the removed host. This could be applications or routers with associated connection dispatchers. This has to be considered and if things are missing they must be replaced. For more information on replacing missing items, see "Dealing with Deployed Applications or Parts of Applications that Only Existed on the Removed Host" on page 71 and "Replacing Missing Routers and Connection Dispatchers" on page 71.

Configuring the Registry for Active-Passive Failover Operation

Use this procedure to improve the availability of the registry node. However, note that this procedure is only part of creating a highly available solution.

The registry node is responsible for many key tasks within the grid including holding the current topology of the running grid and managing distributed locks. If the registry node fails the grid will slowly come to a halt and it will not be possible to start new nodes.

The registry is not stateful, so in the event of a failure a replacement node can take its place with no impact to the operation of the grid. Thus, it is possible to configure a secondary host to support the registry so that in the event of the primary host failing, the grid will not fail. In the event of the primary registry host being unable to start the registry node, the secondary host will start up a registry node. Once the primary registry node is able to be started again, the secondary registry node will terminate.

Before you start Prerequisites:

- Grid 11.1.11.0 or greater

To configure the registry failover host in LifeCycle Manager

1. In LifeCycle Manager, expand the Grid Hosts tree for the grid you wish to configure the second registry host for.
2. Right-click on the registry host and select Host Maintenance > Configure Registry Failover.
3. Select the host you want to add as the registry failover host.
4. Click Next and then click Finish.

To remove the registry failover host in LifeCycle Manager

1. In LifeCycle Manager, expand the Grid Hosts tree for the grid you wish to configure the second registry host for.
2. Right-click on the registry host and select Host Maintenance > Configure Registry Failover.

3 Select the blank entry in the list of hosts.

4 Click Next and then click Finish.

Maintenance

Grid Database

The grid uses a database to store all grid data, including but not limited to the following examples:

- Grid runtime binary files
- Grid applications and associated configuration files
- Grid topology and runtime metadata

The content of the grid database is generally considered to be static, that is, non-transactional. The most common trigger of an update to the grid database is when configuration changes are made. The following are some examples of such changes:

- Add, remove or change a host
- Add, remove, upgrade or re-configure an application
- User management changes
- Grid configuration changes including but not limited to bindings and properties

Given the content of the database, it is critical to the operation of the grid. Loss of the database or corruption to the data will result in the inability to operate the grid and therefore consideration must be given to the infrastructure, security, and maintenance of the database server.

It is highly recommended that the database server be prepared for production purposes in that some or all of the following principles are employed. Please note that these are only intended as guidelines to enforce the need to consider the requirements that the grid database be highly available:

- Disks are configured in a RAID array employing redundancy (for example, RAID 1, 5 or 10)
- Database server is clustered either active-active or active-passive to cope with single server failure
- Database server employs multiple network interfaces with a common address
- Regular point-in-time backups are taken and their ability to be restored is validated at each occasion

The Grid and How It Is Stored on Disk

Each instance of a grid has a folder on disk on each host that is part of that grid instance. This folder is called a grid configuration area. It contains the runtime artifacts of the grid, configuration data, and the applications deployed to that host.

The configuration area is located within *grid_root_installation_path*.

When looking within a configuration area, you will find a set of subfolders, some of which are:

- **applications** - All applications that are deployed in this host will reside here in a subfolder of their own
- **config** - Configuration data used by the grid
- **log** - Log files from all application grid nodes running on this host

Backing Up and Restoring Important Grid Files

In a grid, certain files are essential to the communication within and to that grid. It is recommended to perform a back-up of these files in case of data loss or corruption. Backups should be performed for all hosts in a grid.

To locate the grid files area, see "The Grid and How It Is Stored on Disk" on page 75.

Note: When backing up the grid files, the applications will be backed up as well. However, from the applications' perspective, that may not be enough. Consult the documentation for each application regarding backup procedures.

Important: When you back up the grid files, be sure to apply file security to them so that they are protected similarly to how the live production grid files are protected.

What Grid Files Should Be Backed Up?

As an example, assume a grid is called TestGrid, and is installed in the /Infor ION Grid directory of each participating host. Listed below are the directories that should be backed up.

Installation /secure Directory

The contents of the *grid_root_installation_path*/secure directory should be backed up. In the TestGrid example, this would be the **`/Infor ION Grid/secure`** directory.

After restoring the contents of this folder, restart the host.

Installation /config Directory

The contents of the *grid_root_installation_path*/config directory should be backed up. In the TestGrid example, this would be the **`/Infor ION Grid/config`** directory.

After restoring the contents of this folder, restart the host.

Host /secure Directory

The contents of the /secure directory of the grid on each host should be backed up. In the TestGrid example, this would be the **`/Infor ION Grid/grids/TestGrid/secure`** directory.

After restoring the contents of this folder, restart the host.

Note: If the contents of this folder have been damaged beyond repair and there is no backup, it is possible to perform disaster recovery by reinitializing the grid integrity. This procedure is described in the *Infor ION Grid Security Administration Guide*.

Host /config Directory

The contents of the /config directory of the grid on each host should be backed up. In the TestGrid example, this would be the **`/Infor ION Grid/grids/TestGrid/config`** directory.

After restoring the contents of this folder, restart the host.

Changing the Grid Database Connection Properties

Use this procedure to re-configure the connection properties for the grid database. The following changes are supported:

- Changing the database server and instance
- Changing database schema
- Changing database connection user name or password
- Changing JDBC connection URL

The tool also supports migrating data, for example, if you wish to move the grid database to a new server or database instance.

To reconfigure the database connection properties

1 In the Applications tab in the left pane of the LifeCycle Manager, double-click on the grid you want to reconfigure.

2 In the Grid Maintenance tab of the right pane, click Configure Grid Database.

3 Enter the properties:

Database server	Select a database server which is registered in the LifeCycle Manager from the list box.
Database	Change the name for the database instance name if required.
Schema	Enter a name for the database schema name if required.
JDBC user	Enter the name of the database user.

JDBC password	Enter the name of the database user’s password.
JDBC URL	Change the URL if required.
Copy config data to new connection	If you are migrating the database to another, check the box to copy the existing grid database over.

4 Click Next.

5 Verify the summary and click Finish.

Administrative Task Reference

8

This section provides additional detail about the tasks and administration pages that are available for the grid. Use these procedures and reference material to alter the behavior of the grid and its components.

Administering the Grid

This section describes administrative tasks related to the entire grid.

❏ Start the Grid

The grid will start automatically when the host starts. The sequence of events is:

- The host starts and the operating system boots up.
- The grid bootstrap service is started by the operating system.
- The grid bootstrap service checks the activation flag in the grid database and, if the host should be active, it starts the grid agent.

If the grid does not start automatically, take the following steps (these are for Windows):

___**1** Access the Windows Server Manager on the server where the grid bootstrap service runs.

___**2** Under Configuration, select Services.

___**3** In the list of services, locate the grid bootstrap service. It will have a name in the format: Infor ION Grid Bootstrap - *gridName - hostName*.

___**4** Verify the Status column states "Started". If the service is not started, right-click on the service and select Start.

___**5** If the grid still does not start automatically, execute the StartHost.cmd script, which is located in *grid_root_installation_path*\grids\gridName

___**6** If the grid still does not start automatically, follow these steps:

a Launch the offline configuration manager. See "Accessing the Offline Configuration Manager" on page 14.

b Click on the Hosts link

c Check that the host is activated. If it is not, click on the start symbol to activate the host.

❏ Restart the Grid

Sometimes it can be necessary to restart the grid, for example if asked to do so by support personnel. Restarting the grid will stop all running applications immediately and then will stop all grid nodes including the agent. Restarting the grid should only be done in controlled circumstances since it will impact users of the application(s).

___**1** In the Grid Management Pages for the grid, click the Stop button.

___**2** Check that the option to deactivate hosts is not selected since this will prevent the grid from restarting.

___**3** Select the shutdown immediately option if you do not wish to wait for graceful shutdown of application nodes. This option can be useful when certain application components are not responding.

___**4** Click OK.

❏ Stop the Grid

___**1** In the Grid Management Pages for the grid, click the Stop button.

___**2** Choose whether to select the option to deactivate hosts or not The impact of this flag is as follows:

- If selected, the grid will stop and will not restart until you re-activate the hosts using the Offline Configuration Manager.
- If de-selected, the Grid will stop and then immediately restart.

___**3** Select the shutdown immediately option if you do not wish to wait for graceful shutdown of application nodes. This option can be useful when certain application components are not responding.

___**4** Click OK.

Administering Grid Hosts

This section describes administrative tasks related to grid hosts.

❑ Add additional host

Use this procedure to add another host to the grid. By adding a new host to the grid, applications may scale out to this new host or new applications may be installed there.
The new host must meet the following prerequisites:
Supported Platforms:

- Windows Server 2008 R2
- Windows Server 2012
- Red Hat Enterprise Linux 6
- SUSE Linux Enterprise Server 11

Java JDK Prerequisite:

- 1.6SE Update 41 or higher (32-bit and 64-bit editions supported)
- 1.7SE Update 9 or higher (32-bit and 64-bit editions supported)

Service User Requirements:

- Windows: local or domain account with the ability to log on and run as a service with CRUD access to the file system where the grid will be installed
- Linux: local service account with CRUD access to the file system where the grid will be installed

___**1** In the Configuration Manager, click the Hosts link.

___**2** Click Add Host.

___**3** Enter the following information:

Host FQDN	The fully qualified domain name (FQDN) for the host.
Host Name	A name for the host (optional). This is an alias used to identify the host within the grid. By default, the FQDN will be truncated and the first part will become the host name.
Agent Port	Change the agent port if required. Note that the same port will be suggested as used on the primary (registry) host. Verify that this port is free on the new host and, if it is not, select an alternative port.
Initial State	Change the initial state if required. The default is true, which means that when the bootstrap installer package is executed on the new host, it will immediately start the grid on that host.
Set as Secondary Registry Host	Change the "Set as Secondary Registry Host" if required. The default is true is there is not currently a secondary registry configured within the grid.
Applications	Change the applications if required. All applications which can be auto-scaled to new hosts will appear in the list and if chosen by the developer they will be automatically selected.

___**4** Click OK.

___**5** Download the bootstrap installation package by clicking on the save symbol. If you do not carry out this process from the new host directly, transfer the package to the new host.

___**6** Execute the package and follow the prompts.

❑ Change the JDK for a grid host

Use this procedure to change the JDK version used on a particular grid host. Note that any overrides at host, application, or node level to the JDK set in the grid properties will still be in place following this step, and will not be updated.
Changing the JDK will stop all running applications immediately and then will stop all grid nodes including the agent. This procedure should only be done in controlled circumstances since it will impact users of the application(s).

___**1** Execute the grid installation package again.

___**2** Set the installation path to the same path originally set, that is, the existing grid installation location.

___**3** Click Next.

___**4** Set the JDK path to the new JDK version which has been installed.

___**5** Click Next and close the installation window when the installation is completed.

Following change, the grid will restart using the new JDK specified. This step should be repeated for each host in the grid as required.

❑ Remove an additional host

Use this procedure to remove a host from the grid. Before removing a host from the grid you should un-deploy any applications running there. For more information on removing applications, see "Administering Grid Applications" on page 81.

___**1** In the Configuration Manager, click on the Hosts link.

___**2** Click Remove Host *hostName* and confirm that you want to remove it.

___**3** In the file system of the host you are removing, navigate to the grid uninstall folder. This is *grid_root_installation_path*\uninstall.

___**4** Using elevated rights (run as administrator on Windows, root on Linux), execute the uninstall.jar.

Administering Grid Applications

Use the procedures below to administer the applications running in a grid.

❑ Start Application

Use this procedure to start an application.

An application is started based on the information given in the bindings. If a binding is configured with an initial node count greater than zero, that node count will be considered when starting the application. So, if an application has two different bindings and both of them have an initial count of one (1), the grid will recognize this and start one node each for the two bindings

___**1** Access the Grid Management Pages.

___**2** Click the link at the bottom for the application you would like to start.

___**3** Click Start and confirm the operation.

❑ Stop Application

Use this procedure to stop an application. The Stop Application task stops all grid nodes belonging to this application and sets the application in an off-line state. Since the application is put in an off-line state, nodes will not be automatically started even a binding exists with a minimum count greater than zero. The application will remain in an off-line state until it is started again.

Note: It is recommended to first put the application in an off-line state before stopping it. This gives the application time to finish processing while at the same time preventing it from accepting new client requests.

___**1** Access the Grid Management Pages.

___**2** Click the link at the bottom for the application you would like to stop.

___**3** Select the shutdown immediately option if you do not wish to wait for graceful shutdown of application nodes. This option can be useful when certain application components are not responding.

___**4** Click Stop and confirm the operation.

❑ Manage Application

Use this procedure to manage the application. The management pages for an application are displayed if the application is running. The actual content of the pages will differ depending on the application.

___**1** Access the Grid Management Pages.

___**2** Click on the Management Pages link at the bottom for the application you would like to manage.

The management page for this application is now displayed.

❑ Monitor Application

Use this procedure to monitor an application. In order for this page to be shown, the application must be running. The page shows the current state of the application, and it is possible to start and stop the application and its nodes. In addition, this page provides links to the application's configuration and management pages.

___**1** Access the Grid Management Pages.

___**2** Click on the name of the application at the bottom of the page for the application you would like to monitor

The page for monitoring this application is now displayed.

❑ Configure Application

Use this procedure to configure an application. The way the application is running in the grid is set up using the Grid Configuration Manager. Here you define on which hosts to run the application, the number of nodes to start and all other properties needed in order to run the application in this environment.

___**1** Access the Grid Management Pages.

___**2** Click on the name of the application at the bottom of the page for the application you would like to configure.

___**3** Click Configuration.

The configuration manager for this application is now displayed.

❑ Deploy Application on Hosts

If you want to scale out an application to a host that it is not yet deployed to, you may use this procedure. After you have deployed the application on the new hosts, you also must reconfigure bindings for the application so that the bindings allow for the application to start on the new hosts. To configure the bindings, use the procedure "Configure Application" on page 83.
An alternative to scaling out is that you want to move the application to a new host. In that case, you deploy the application to the new host, as described here, and later remove the application from the old host. An alternative to removing the application from the old host is to leave it there but just reconfigure the bindings so that it is not possible for the application to start on that host.

Note: Consult the documentation of each application. Some applications do not support scale-out and there may be restrictions on what each application supports in terms of moving an application.

___**1** Access the Grid Management Pages.

___**2** Click on the name of the application at the bottom of the page for the application you would like to deploy.

___**3** Click Configuration.

___**4** Click the Scale Out link.

___**5** Select the host(s) you would like to scale the application out to and confirm the change.

___**6** Save the changes by clicking on Save at the top of the configuration panel.

❑ Remove Application from Hosts

Use this procedure to remove an application from one or several hosts. Binding references associated with this application and the selected hosts will be removed.
It is recommended to first stop any application nodes on the host that you want to remove the application from.

Note: Note that removing the application from a host will remove all artifacts of that application including configuration and/or data files depending on the application. Further, any grid properties which were configured for that host will be lost. Please consult the documentation for each application for more guidance.

___**1** Access the Grid Management Pages.

___**2** Click on the name of the application at the bottom of the page for the application you would like to remove.

___**3** Click Configuration.

___**4** Click the "X" to the right of the host(s) you want to remove the application from.

___**5** Save the changes by clicking Save at the top of the configuration panel.

Configuration Manager

The Configuration Manager is used to change the runtime aspects of a grid. This includes configuration of installed grid applications and also configuration of the grid itself. The changes made here are kept persistent in the grid configuration area and will survive a restart of the grid and the nodes in it.

Note: Running nodes are not affected by changes made in the Configuration Manager until they are restarted.

The first page of the Configuration Manager consists of links that let you configure different aspects of the grid. These links are described below.

Applications

This page lists all installed grid applications and provides basic information about them.

Column	Description
Name	The configured name of the installed application.
Version	The application version.
Type	The application type. An application may be installed several times in the same grid under different names.
Hosts	A list of hosts that the application in installed on.
Remove button	Uninstalls the application from one or several hosts.

Install New Application

Click Install New Application to install an application. Select a gar file from the application repository using the drop-down list or click the Upload button to add a new gar to the repository, give it a name, and select a set of hosts to deploy the application onto.

Edit Application

Click on one of the applications in the list to edit the configuration for that application.

On the Application configuration page, information about the deploy status is displayed and it is possible to configure the application. This includes:

- Application properties
- Bindings
- Connection dispatchers
- Context roots

Application Properties

Click on the Edit Properties link to view a list of application-defined properties (if any).

Column	Description
Property	The property title if defined (the property name otherwise). Clicking on this link lets you edit the property in different contexts.
Value	The grid-wide value of the property if defined. Clicking this link lets you edit the property value in the global context. Note that if the property has previously been overridden in different contexts, the result of clicking this link is the same as clicking the property link described above.
Unit	The unit of the property, for example, seconds or MB.
Type	The type of the property, for example, Integer or String
Description	A description of the property.
Name	The property name (which may be different from the title shown in the Property column).

Click on one of the listed application properties to edit its runtime value.

Property Contexts

If you have chosen to edit a property in different contexts by clicking the on the property link, you will be shown a matrix. The matrix displays the values of this property in the different contexts and lets you override the values.

By default, contexts that are not used will be hidden. This is done in an attempt to make the configuration of properties less complicated. If no overrides have been done in other contexts, only the global context and the host contexts are shown. If possible, it is recommended to use only those contexts. However, there are situations when you need to use the other contexts.

If you need to use a context that is hidden, you need to change the display complexity at the top of the page in order to see and edit them.

Example:

Consider an application that has a property DBCon that holds the connection string to a database. Typically we would give this property a value in the global context. That means that all application instances on any host will get this value when reading this property. However, if one of the hosts is running on a different operating system and is using a different database driver, we may need a different connection string on that host but on that host only. What we do in that case is to also override the DBCon property in the context of that host. Thus, the property will have the global value on all hosts except this host, but on this host it will have the new overridden value.

The contexts have different priorities, so a property may be defined in several contexts, and the one with the highest priority will be the actual runtime value depending on how and where the application is running.

The matrix displays the different values and how they are resolved. So by overriding in one context it is easy to see what other contexts are overridden (because of lower priority) and what contexts inherit this value.

To help the user to understand how different contexts override other contexts, the matrix is displayed using overlapping boxes. Each context box encompasses all other contexts that it may influence. At the same time, it is easy to see if other context boxes overlap this box and thus have the potential of overriding the first context box.

The contexts are (lowest priority first):

Global (Application & Any Host): This is the base context (lowest priority). Use this context to give an application property a value regardless of other contexts (for example, host, binding). It is good practice to define a value even if you intend to override it in one of the other contexts. In the matrix, the global value is shown in the first row (Application) and first column (Any Host).

Node Type (Node Type & Any Host): Use this context to override a property for all application instances that are running in a node of this node type regardless of host or binding. In the matrix, this value is shown in the row corresponding to the node type and the first column (Any Host).

Specific Host (Application Global & Specific Host): Use this context to override a property on a specific host. Any application instance started on that host will use this value unless overridden by one of the other contexts with higher priority. In the matrix, this value is shown in the first row (Application) and the column corresponding to the specific host.

Binding (Binding & Any Host): Use this context to override a property in the context of a binding. Any application instance started with this binding will use this value unless overridden by a context with higher priority. In the matrix, this value is shown in the row corresponding to the binding and the first column (Any Host).

Node Type & Specific Host: Use this context to override a property in the context of a node type on a specific host. In the matrix, this value is shown in the row corresponding to the node type and the column corresponding to the host.

Binding & Specific Host: This is the context with the highest priority. It overrides a property value for application instances started with a particular binding on a particular host.

At runtime, resolving a property value will be done in the following fashion:

Each application instance is started using exactly one binding on exactly one host, so when resolving a property, the system will first look for a property override for exactly that binding on exactly that host (Binding & Specific Host context). If it finds one, the system has found the applicable runtime value and is finished. If not, the system will continue to look for a property override in contexts with lower priority until it finds a context that has defined an override. The last context to look in is the global context.

If a property has been overridden in a given context, a small black bullet will be displayed in front of the value. To remove an override, edit the value in the context that you want to remove, and click Remove.

Edit Bindings to Application Defined Node Types

The application page displays the bindings that are defined.

Column	Description
Name	Binding name.
Node Type	The node type that the binding is using when starting a node. The node type is one of the node types that this application defines.
Startable	Indicates if the binding targets a node type that is manually startable from the management UI. If it is not manually startable, the binding is only startable programmatically.
Min	The minimum number of nodes (started with this binding) that are allowed. If the number of running nodes are below this value, the system will automatically start new nodes in order to meet this minimum.
Max	The maximum number of nodes allowed (started with this binding).
Initial	How many nodes to start at system and application start (Min <= Initial <= Max). When you start an application, this value will be considered.
Hosts	A set of hosts that this binding may use. This binding may only be used to start nodes on this set of hosts.
Remove button	Removes this binding.

Add a new binding by clicking Add Binding at the top of the page. Give the binding a name, select a node type, give relevant values to the Min, Max and Initial field and select the hosts to use.

Connection Dispatchers

The application page also displays configured connection dispatchers (if any).

Column	Description
Name	The name of the connection dispatcher.
Router	The router to use when opening the port.
Port	The port to use.
Remove button	Removes this connection dispatcher.

Add a new connection dispatcher by clicking Connection Dispatcher at the top of the page.

Context Roots

If the application contains a web application or it exposes web services, those require that context roots are defined. In the Context Root Mappings panel, it is possible to manage the context roots. See the installation guide for each application for details.

Column	Description
Module	The grid application module that is exposing the web application or web server.
Type	Indicates if the context root is for a web application or web services.
Web Application	The name of the web application if any.
Context Root	The name of the context root.
Remove button	Removes this context root.

Application Repository

The application repository is the central store for all applications stored in the Grid database. From here you can:

- View the applications currently in the repository
- Add a new application gar to the repository
- Remove an application from the repository

To add an application to the repository

1 Click Upload Application.

2 Click Browse, navigate to the location of the application gar file, and select it.

3 Click Upload.

To remove an application gar file from the repository

Note: You can only remove an application that is not deployed.

- Click the red "X" at the right end of the application gar line.

Hosts

The Hosts page is used to manage hosts in the grid including adding and removing additional hosts.

Grid Properties

Grid-defined properties are listed.

Column	Description
Property	The property title if defined (the property name otherwise).
Value	The grid-wide value of the property if defined.
Unit	The unit of the property, for example, seconds or MB.
Type	The type of the property, for example, Integer or String
Description	A description of the property.
Name	The property name (which may be different from the title shown in the Property column).

Click on one of the listed properties to edit its runtime value.

When editing a property you will be shown a matrix that lets you define and override the value for this property in different contexts.

Property Contexts

If you have chosen to edit a property in different contexts by clicking on the property link, you will be shown a matrix. The matrix displays the values of this property in the different contexts and lets you override the values.

By default, contexts that are not used will be hidden. This is done in an attempt to make the configuration of properties less complicated. If no overrides have been done in other contexts, only the global context and the host contexts are shown. If possible, it is recommended to use only those contexts. However, there are situations when you need to use the other contexts.

If you need to use a context that is hidden, you need to change the display complexity at the top of the page in order to see and edit them.

Example:

Consider an application that uses lots of memory and you want to override the Max Heap property. Typically you would give this property a value in the global context. That means that all application instances on any host will get this value when reading this property.

However, assume that if the application is started using a particular node type, additional tasks will be assigned to it and you need even more memory. What you do in that case is to also override the Max Heap property in the context of that node type. So the property will have the global value in all application instances except instances belonging to this specific node type.

The contexts have different priorities, so a property may be defined in several contexts, and the one with the highest priority will be the actual runtime value depending on how and where the application is running.

The matrix displays the different values and how they are resolved. So, by overriding in one context it is easy to see what other contexts are overridden (because of lower priority) and what contexts inherit this value.

To help the user to understand how different contexts override other contexts, the matrix is displayed using overlapping boxes. Each context box encompasses all other contexts that it may influence. At the same time, it is easy to see if other context boxes overlap this box and thus have the potential of overriding the first context box.

The contexts are (lowest priority first):

Global (Grid-Wide & Any Host): This is the base context (lowest priority). Use this context to give a property a value regardless of other contexts (for example, host, binding). It is good practice to define a value even if you intend to override it in one of the other contexts. In the matrix the global value is shown in the first row (Grid-Wide) and first column (Any Host).

Node Type (Node Type & Any Host): Use this context to override a property for all application instances that are running in a node of this node type regardless of host or binding. In the matrix this value is shown in the row corresponding to the node type and the first column (Any Host).

Specific Host (Application Global & Specific Host): Use this context to override a property on a specific host. Any node and application instance started on that host will use this value unless overridden by one of the other contexts with higher priority. In the matrix this value is shown the first row (Grid-Wide) and the column corresponding to the specific host.

Binding (Binding & Any Host): Use this context to override a property in the context of a binding. Any node and application instance started with this binding will use this value unless overridden by a context with higher priority. In the matrix this value is shown in the row corresponding to the binding and the first column (Any Host).

Node Type & Specific Host: Use this context to override a property in the context of a node type on a specific host. In the matrix this value is shown in the row corresponding to the node type and the column corresponding to the host.

Binding & Specific Host: This is the context with the highest priority. It overrides a property value for application instances started with a particular binding on a particular host.

In runtime, resolving a property value will be done in the following fashion:

Each node and application instance is started using exactly one binding on exactly one host, so when resolving a property, the system will first look for a property override for exactly that binding on exactly that host (Binding & Specific Host context). If it finds one, the system has found the applicable runtime value and is finished. If not, the system will continue to look for a property override in contexts with lower priority until it finds a context that has defined an override. The last context to look in is the global context.

If a property has been overridden in a given context, a small black bullet will be displayed in front of the value. To remove an override, edit the value in the context that you want to remove, and click Remove.

Routers

Defined routers are listed.

Column	Description
Name	The name of the router.
Running	Indicator if the router is running. Green bullet means that the router is running.
Host	The host that the router is configured to run on. It is possible to configure a router so that it automatically runs on all hosts in the grid by choosing <all hosts>. This can be useful if configuring a load balancer to distribute the load of client connections amongst all the hosts in the grid for either resilience or load balancing.
Port	The port the router is listening on.
Encryption	indicates whether encryption is in use for this router. Note that this only relates to grid client traffic which is connected via the port specified previously.
External Address	Optional external address (IP number). If undefined, the router listens on the same address as the grid agent is using on this host. If the host has several network interfaces, it is possible to select a different one here.
Http Port	The HTTP port of the router if defined. The HTTP port serves web applications and web services.
Https Port	The HTTPS port of the router if defined. The HTTPS port serves web applications and web services.
Dispatcher	A list of connection dispatchers exposed by this router (if any).
Remove button	Removes this router.

Editing and Adding Routers

Routers may be added or existing routers may be reconfigured. Client applications may connect to any router as long as it is exposing the right type of port (for example, HTTP) and that port is exposed on a network interface that is accessible from the client.

One reason for adding a new router is that you want to bind it to another network interface compared to the existing routers. The external address property of the router is used for this. See above. Another reason for adding a router is to achieve high availability. By having two routers that serve the same types of ports but on different hosts, you still have a way for clients to connect to the grid even if one of the routers fails. To make this fully transparent to the clients, some form of network load balancer in front of the routers may be needed.

To add a router, click the Add Router link. Edit a router by clicking on the router name link in the list. In both cases you will be presented with a dialog that lets you configure the router.

Setting WWW Authentication Methods for a Router

The router configuration dialog has tabs for setting WWW authentication methods for HTTP and HTTPS, respectively. For more information on what these settings mean, see "Configuring Router WWW Authentication Methods" in the *Infor ION Grid Security Administration Guide*.

Context Root Mappings

Grid applications may expose web applications and web services via an HTTP or HTTPS port in one of the routers. In order to distinguish between web applications and web services provided by different applications, each web application and web service needs to be given a context root that will be part of the address used by a client when connecting.

It is possible to configure context roots in this page, but it is better to do so from the application configuration page. Consult the documentation of each application.

Bindings

All bindings in the configuration are listed on the page.

Column	Description
Name	Binding name
Running	Indicator if the router is running. A green bullet means that at least one node, started using this binding, is running.
Node Type	The node type that the binding is using when starting a node.
Min	The minimum number of nodes (started with this binding) that are allowed. If the number of running nodes is below this value, the system will automatically start new nodes in order to meet this minimum.
Initial	How many nodes to start at system start (Min <= Initial <= Max).

Column	Description
Actual	The actual number of running nodes that were started using this binding.
Max	The maximum number of nodes allowed (started with this binding).
Hosts	A set of hosts that this binding may use. This binding may only be used to start nodes on this set of hosts. If the binding is configured to run on all hosts, an asterisk (*) will appear in this column.
Remove button	Removes this binding.

Note: Bindings to application-defined node types are best configured on the configuration page of each application even though it is possible to do so here also.

Session Providers

For information on session providers, see the *Infor ION Grid Security Administration Guide*.

All Entities

This page displays a view of what is currently running and where it is running in the grid. A green bullet in one of the host columns means that at least one node of the binding, on this row, is running on that host. A white bullet means that it is possible to start this binding on that host but none is started at present.

Advanced Configuration

The following advanced configuration options are available:

- **Change Grid Name**

  Change the display name for the grid.

- **Change Grid Version**

  Manage the grid versions.

  For available procedures on this page, see "Managing Grid Versions" on page 94.

- **Audit Log**

  This page allows you to view the audit trail for the grid showing when hosts are added, started, stopped and removed.

- **Configuration Files**

Display and compare the topology.xml and the runtime.xml files on all hosts. This option could be useful if there is a problem in the grid, as a way of making sure that the same configuration applies to all hosts (which it should).

- **Configuration History**

  This page allows you to see the history of configuration changes for the grid and if required you may revert to a previous configuration.

  Note that this should be done with great care as it may affect the runtime operation of the grid and/or its applications.

- **Defined Ports**

  Display a list of all ports that this grid defines and on which network interface they are bound.

- **Linked Resources**

  Not used in a customer scenario.

- **Certificates**

  Used to configure certificates. For more information, see the *Infor ION Grid Security Administration Guide*.

Managing Grid Versions

The Manage Grid Versions page references the grid version repository, which is the central store for all grid runtime binary packages stored in the grid database. On this page you can:

- View the grid runtime version currently use
- Change the grid runtime version
- View the grid runtimes currently held in the repository
- Add a new grid runtime to the repository
- Remove a grid runtime from the repository

To change the grid runtime version

Note: Changing the grid version will stop all running applications immediately and then stop all grid nodes including the agent. It should only be done in controlled circumstances since it will impact users of the application(s).

1 Click Change Version.

2 Select the version to change to from the drop-down list

3 Click Upgrade.

To add a grid runtime to the repository

1. Click Add Binaries.
2. Click Browse and select the grid runtime file.
3. Click OK.

To remove a grid runtime version

Note: You can only remove a grid runtime which is not in use.

- Click the red "X" at the end of the grid runtime line.

ION Grid Properties

A

All properties for the ION Grid are defined below including a brief explanation of each property. They can be edited through the Configuration Manager.

Grid Properties

Properties whose names are grayed out on the screen are ones that apply globally only. You cannot override them in another context.

Property	Type	Description	Default value
Node Memory			
Max Heap	integer, MB	Sets the maximum heap space for the JVM.	256 MB
Throttling Memory Threshold	integer, percent	Threshold, expressed as percentage of maximum heap, for when to start throttling the node (putting the node in an off-line state).	The default is 85% of max heap.
Throttling GC Frequency	integer, seconds	When throttling, how often a garbage collect should be performed.	5 seconds
Node Logging			
Node Log Level	list	The logging level of the node: ALL, ERROR, WARN, INFO, NOTE, DEBUG, TRACE	ERROR, WARN, INFO, NOTE

Property	Type	Description	Default value
Node Log Detail Level	list	The logging level of a particular logger, the format is **`[logger-name]=[levels]`**, for example, **`Node=INFO,DEBUG`**.	
Node Log Capture Standard Out	boolean	Whether or not to capture stdout in the log file.	true
Node Log Capture Standard Err	boolean	Whether or not to capture stderr in the log file.	true
Node Log Max File Size	integer	The maximum size of a log file.	5 MB
Node Log Archive Age	integer, days	Log files older than this will be archived into a zip file. Set to **`0`** to continuously archive all logs that are not live. More than 10 files must be eligible in order for an archive to be created.	The default is **`-1`**, which disables this feature.
Node Profiling and Monitoring			
Profiler	enumeration	Turns the profiler on or off.	off
Node JMX Enabled	boolean	Setting to enable or disable a node's JMX server.	true
Counters Poll Delay	integer, seconds	How many seconds to wait between polling counters.	15 seconds
Counters Keep History	integer, minutes	How many minutes of counter history to retain.	60 minutes
CPU Sampling	enumeration	Enables or disables CPU sampling.	The default is on (if the JVM supports it).
CPU Sampling Frequency	integer, seconds	How often to sample CPU usage.	5 seconds
Thread Priority Capping	boolean	Enables or disables thread priority capping; that is, threads with consistently high CPU usage will automatically get their priority lowered temporarily. Note that CPU Sampling must be enabled in order for priority capping to work.	The default is on.

Property	Type	Description	Default value
Thread Priority Capping Threshold	integer, percent	Threshold for thread priority capping based on CPU percentage.	85%
Misc. Node Properties			
Node Weight	integer	Node weights are used as a load balancing mechanism. Each Node has a weight (>= 0).	100
Node Startup Failure Shutdown Grace Period	integer, seconds	How many seconds to wait before automatically shutting down a node when the hosted application failed to load or start. If the node hosts several application modules, it will only shut itself down if all modules fail to start.	60 seconds
Java Executable	path	The path to a Java executable.	The default is the one used to start the grid agent.
Working Directory	path	The path to use as working directory for a node. Use a forward slash (/) as the directory separator.	
Library Path	path list	Adds the specified paths to the library path of the JVM. Use a forward slash (/) as the directory separator.	
Generic JVM Commands	list	A list of generic JVM command line switches.	
Debug Port	integer, port number	Adds JVM arguments to start the node in debug mode (not suspended).	The argument is the port where you attach your debugger.
Debug Port (Suspended) (Deprecated)	integer, port number	Adds JVM arguments to start the node in suspended debug mode.	This property is deprecated. The argument is the port where you attach your debugger.

Property	Type	Description	Default value
Default Session Timeout for Ephemeral Sessions	integer, minutes	Sets the default timeout for ephemeral session, that is, sessions logged on as a result of an HTTP request without grid session header(s).	
Stream Buffer Size	integer, KB	The buffer size for transporting stream data. A higher value will potentially increase throughput speed, but have a larger memory footprint.	64 KB
Http Idle Timeout	integer, seconds	The timeout used for closing idle HTTP connections automatically. Set to 0 to disable timeout completely. This property is only valid in a router context.	3600 seconds
Path Element (sub context root) routing key	string	Configures which path element (sub context root) gets routed to "this" node.	blank
Grid Database Library	string	Grid installation database library on IBM i.	blank
Node Subsystem	string	Used by Java executable script (grid.jvm.executable) to control where to submit JVMs (nodes) on IBM i.	blank
Grid Monitoring			
Unresponsive Node Monitoring Retry Count	integer	The number of times an unresponsive node will get a second chance to respond before being blacklisted (and hence disconnected from the grid) by the registry. The default is 0. That is, it will be blacklisted after 60+ seconds. Each retry count adds another 60 seconds to let the unresponsive node respond before being blacklisted.	The default is 0 (60 seconds).
Unresponsive Node Monitoring Process Termination	boolean	Uncheck to prevent the grid from attempting to forcefully terminate an unresponsive node.	Default is true (unresponsive node will be terminated).

Property	Type	Description	Default value
Grid Shutdown			
Grid Shutdown Wait Timeout	integer, seconds	How long to wait for the grid or a grid host to shutdown before forcefully stopping processes.	The default is 60 seconds.
Module Thread Pool			
Thread Pool Max Queued	integer	Sets the maximum number of queued proxy request threads.	The default is 0 (unlimited).
Thread Pool Max Concurrent	integer	Sets the maximum number of concurrently executing proxy request threads.	The default is 0 (unlimited).
Thread Pool Max Pooled	integer	Sets the maximum number of pooled proxy request threads.	The default is 0 (unlimited).
Thread Pool Min Pooled	integer	Sets the minimum number of pooled proxy request threads.	The default is 5 (although the minimum number of pooled threads can actually be lower if the number of concurrently executing threads never exceeds the minimum).
Thread Pool Warn Delay	integer, seconds	The number of seconds to wait for a proxy request to finish before the system puts a warning in the log.	The default is 0 (off).
Module Web Service			
Web service Chunked Transfer Encoding	boolean	If set to true, chunked transfer encoding will be used for web service responses. This will override the auto buffer setting (Web Service Auto Buffer Size) meaning auto buffering will not be used when chunking is set to true.	The default is true.

Property	Type	Description	Default value
Web Service Auto Buffer Size	integer	When set to a value higher than zero (0), web service responses will be buffered up until the specified size (KB) in an attempt to determine the content length. When successful, the content length will be sent, otherwise connection close will be used. This feature is disabled when chunking (Web service chunked transfer encoding) is used.	The default value is 16 KB
Misc. Module Properties			
Class Path	path list	Sets the specified paths as the class path for the module. Use a forward slash (/) as the directory separator.	
Start Timeout	integer, seconds	The timeout to wait for a node to start before declaring the start as failed.	The default is 120 seconds.
Stop Timeout	integer, seconds	When stopping a module, how many seconds to wait for the module to stop by itself before forcefully shutting it down.	The default is 60 seconds.
Security Properties			
Default Session Provider	string	Sets the name of the preferred session provider, if any, to use when requesting a provider using the SessionUtils getProvider(int type) method.	
Default Session Timeout	minutes, integer	Sets the default timeout for sessions.	
Default Logon Service	string	Sets the name of the default logon service used when using session providers from this module.	
Trusted Routers Http Entrypoints	map to string	Trusted routers HTTP entry point mappings.	
Grid Http Discovery Services			

Property	Type	Description	Default value
Context root discovery urls	map to string list	Configures how context roots are presented when discovered through the grid info http service.	
Grid Internal Properties			
grid.internal.jsm	boolean	Internal option. Only set to false if instructed to by Infor.	true
grid.internal.jna	boolean	Internal option. Used to disable native access functions. Only set to false if instructed to by Infor.	true
grid.internal.dup	boolean	Internal option. Used to disable property updates on an entity. Only set to false if instructed by Infor.	true
grid.internal.diepar	boolean	Internal option. Used to disable NTLM IE pre-auth request interception.	false
grid.internal.rme	boolean	Internal option. Used to disable role mapping resolution. Only set to false if instructed by Infor.	true
grid.internal.wdm	boolean	Internal option. Used to enable web ui tracing for developer purposes. Only set to false if instructed by Infor.	true
grid.internal.dnm	boolean	Internal option. Used to disable the network monitor. Only set to false if instructed by Infor.	true
java.net. preferIPv4Stack	boolean	Internal option. Maps to the Java system property java.net. preferIPv4Stack. Only set to false if instructed by Infor.	true
java.net. preferIPv6Addresses	boolean	Internal option. Maps to the Java system property java.net. preferIPv6Addresses. Only set to true if instructed by Infor.	false

Index