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

The word and design marks set forth herein are trademarks and/or registered trademarks of Infor and/or related affiliates and subsidiaries. All rights reserved. All other company, product, trade or service names referenced may be registered trademarks or trademarks of their respective owners.

Publication Information

Release: 10.1.x
Publication date: February 10, 2014
Document Number: GRIDAG_10.1.x_UWA_08
# Version Log

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

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Release Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRIDAG-9160</td>
<td>2010-10</td>
<td>Rearranged properties and added new properties to appendix. Added new procedure on changing JDK for hosts.</td>
</tr>
<tr>
<td>GRIDAG-9170</td>
<td>2011-04</td>
<td>Added numerous topics on administering, configuring, and troubleshooting the grid.</td>
</tr>
<tr>
<td>GRIDAG-1010</td>
<td>2011-11</td>
<td>Added numerous topics on administering, configuring, and troubleshooting the grid.</td>
</tr>
<tr>
<td>GRIDAG-1010</td>
<td>2012-06</td>
<td>Added information on additional monitoring procedures and the SAML session provider.</td>
</tr>
<tr>
<td>GRIDAG-1010</td>
<td>2012-08</td>
<td>Revised procedure on use of Grid Script Client.</td>
</tr>
<tr>
<td>GRIDAG-1010</td>
<td>2012-11</td>
<td>Added procedure on registry failover.</td>
</tr>
<tr>
<td>GRIDAG-1010</td>
<td>2013-01</td>
<td>Minor session provider clarifications.</td>
</tr>
<tr>
<td>GRIDAG-1010</td>
<td>2013-05</td>
<td>Updates for Lawson Grid 10.1.11.0.</td>
</tr>
</tbody>
</table>
Contents

Chapter 1: The Lawson Grid Overview.................................................................8
  The Lawson Grid..............................................................................................8
  Knowledge Prerequisites................................................................................9
  Lawson Grid Concepts..................................................................................9

Chapter 2: Grid Administration Tools............................................................11
  Navigating between the Grid Administration Tools........................................11
  Grid Management Pages for Monitoring the Grid...........................................12
  Accessing HTML-Based Grid Management Pages.......................................15
  Accessing the Grid Management Pages Through Java Web Start.................16
  Troubleshooting Tools..................................................................................16
  The Grid Script Utility..................................................................................17

Chapter 3: Monitoring the Grid.......................................................................18
  Log Files and Reports....................................................................................18
    Viewing Log Files.......................................................................................18
    Changing Logging Levels...........................................................................20
    Generating a Grid Status Report.................................................................23
    Viewing a Grid Status Report....................................................................23
  Monitoring Tools..........................................................................................24
    Monitoring the State of the Grid from a Web Browser...............................24
    Monitoring the State of the Grid from the Java Web Start Component.........26
    Viewing Counters.....................................................................................29
| Chapter 4: Changing the Grid Configuration | 30 |
| Configuring Memory Given to Applications | 30 |
| Configuring Bindings | 31 |
| Changing the Display Name of a Grid | 32 |
| Viewing Configuration History | 32 |
| Reverting to an Older Grid Configuration | 32 |
| Changing How JVMs Are Launched on the OS Process Level | 33 |
| Changing the Grid Agent Service User on Windows | 38 |

| Chapter 5: Configuring Network Access to the Grid | 40 |
| Ports for Client Applications | 40 |
| Listing All Ports Exposed by Routers in a Grid | 41 |
| How to Use HTTP and HTTPS Ports | 42 |
| Working With Context Root Discovery URLs | 42 |
| Using a Web Server as a Proxy Server to Redirect HTTP Traffic to the Grid | 44 |

| Chapter 6: Managing the Grid | 46 |
| Managing the Grid Through the LifeCycle Manager | 46 |
| Putting Applications or Parts of the Grid in an Off-Line State | 46 |
| Stopping an Individual Grid Node | 48 |
| Starting New Grid Nodes (Application Instances) | 49 |
| Managing the Grid Programmatically | 50 |
| Programantically Operating on the Grid using REST | 50 |
| Using the Grid Script Client to Manage the Grid | 54 |

<p>| Chapter 7: Administering Grid Security | 58 |
| Session Providers | 58 |</p>
<table>
<thead>
<tr>
<th>Chapter 9: Administrative Task Reference</th>
<th>88</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administering the Grid</td>
<td>88</td>
</tr>
<tr>
<td>Administering Grid Hosts</td>
<td>90</td>
</tr>
<tr>
<td>Administering Grid Applications</td>
<td>93</td>
</tr>
<tr>
<td>Grid Management Pages</td>
<td>97</td>
</tr>
<tr>
<td>Configuration Manager</td>
<td>97</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Appendix A: Lawson Grid Properties</th>
<th>107</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid Properties</td>
<td>107</td>
</tr>
</tbody>
</table>

| Index | 114 |
The Lawson Grid

The Lawson 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 Lawson Grid may span multiple server machines.

The Lawson 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.

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.

Normally, a grid is only considered to be fully operational when all grid agents are running on every host in that grid. However, a host may be configured to be transient. A transient host is not required to be running in order for the grid to be considered fully operational.

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.
Registry

The registry is a special type of node that is needed when new nodes are started. A grid has exactly one registry.

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 port number. Normally, the selected port number has to be made accessible through firewalls, when applicable, since it must be reachable from clients.

Administrative Router

A special router used by LifeCycle Manager for administrative purposes.

Knowledge Prerequisites

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

• Have experience installing and configuring applications using LifeCycle Manager.
• Have operating system administrator experience.

Lawson Grid Concepts

A user of the Lawson 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.
**Grid Administration Tools**

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

- "Navigating between the Grid Administration Tools" on page 11
- "Grid Management Pages for Monitoring the Grid" on page 12
- "Accessing HTML-Based Grid Management Pages" on page 15
- "Accessing the Grid Management Pages Through Java Web Start" on page 16
- "Troubleshooting Tools" on page 16
- "The Grid Script Utility" on page 17

**Navigating between the Grid Administration Tools**

Throughout the *Lawson Grid Administration Guide* you will frequently be asked to navigate to the Grid Management Pages, the Configuration Manager, or other pages. How to do that is described below.

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

- "Accessing HTML-Based Grid Management Pages" on page 15
- "Accessing the Grid Management Pages Through Java Web Start" on page 16

**To access the Grid Management Pages**

1. In the LifeCycle Manager (LCM), select the Applications tab in the left pane.
   
   This should list all your instances of the Lawson Grid. The individual grids are organized under their respective grid version in the tree.

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 Under Tasks in the right pane, select the Lawson Grid tab and click the Grid Management Pages link.

This will open the Management Pages in a new tab.

To access the Configuration Manager

1 In the LifeCycle Manager (LCM), select the Applications tab in the left pane.

This should list all your instances of the Lawson Grid. The individual grids are organized under their respective grid version in the tree.

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 Under Tasks in the right pane, select the Lawson Grid tab and click the Configuration Manager link.

This will open the Configuration Manager in a new tab.

**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.

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 "Navigating between the Grid Administration Tools" on page 11.

**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 instance of the Lawson Grid
- 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
- A link to pages showing more advanced details of the grid
- A link to a page with application focus (in contrast to the runtime artifact focus of this page)

For each runtime artifact, the page displays the following:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>The type of runtime entity displayed on a row, for example, host, grid agent, registry, router, or node</td>
<td>For hosts, this column also includes a start button that allows you to start nodes (bindings) on that particular host.</td>
</tr>
<tr>
<td>Name</td>
<td>The name of the runtime entity</td>
<td>For nodes, this is the name of the binding used to start the node.</td>
</tr>
<tr>
<td>Status</td>
<td>The general status of the entity. Normally this should be &quot;OK,&quot; but it may also be &quot;OFFLINE&quot; or &quot;STOPPING.&quot;</td>
<td>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 &quot;Putting Applications or Parts of the Grid in an Off-Line State&quot; on page 46.</td>
</tr>
<tr>
<td>A stop button for all entities except hosts and grid agents</td>
<td>This is useful when an individual application node needs to be shut down, for example, because it is in an error state.</td>
<td></td>
</tr>
<tr>
<td>Id</td>
<td>A unique ID for each runtime artifact.</td>
<td>The ID consists of address, port, and process id.</td>
</tr>
<tr>
<td>Log</td>
<td>Link that displays the log file belonging to this runtime artifact</td>
<td>This is the default place to look for information if a node is showing signs of having problems.</td>
</tr>
</tbody>
</table>
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.

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 (which is the same component served through LifeCycle Manager) 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

**Important:** The HTML Grid Management Pages should not be used as an alternative to LifeCycle Manager for installation, upgrade, or deployment.

To access the HTML-based Grid Management Pages

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

   \[
   \text{http(s)}://server:port/grid/ui
   \]

   where \textit{server} is the name of the server hosting the grid and \textit{port} is the HTTP or HTTPS port for the grid router.
Accessing the Grid Management Pages Through Java Web Start

You can access the Grid Management Pages through a Java Web Start component. The component is the same that is in the LifeCycle Manager client and so can be used to monitor and administer a grid.

In addition to the features this version of the Grid Management Pages has in common with the LifeCycle Manager accessible version, this version offers a profiler viewer and a log viewer. For more information, see "Monitoring the State of the Grid from the Java Web Start Component" on page 26.

**Important:** The Grid Management Pages accessed through Java Web Start should not be used as an alternative to LifeCycle Manager for installation, upgrade, or deployment.

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.

Troubleshooting Tools

The Lawson 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 18.

- Counters
  Counters are available for each grid node as well as a counter history. For more information, see "Viewing Counters" on page 29.

- 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 32.

- 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 77.

- 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 23 and "Viewing a Grid Status Report" on page 23.

For specific troubleshooting procedures and advice, see "Troubleshooting" on page 76.

The Grid Script Utility

The Lawson 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 54.
Monitoring the Grid

Log Files and Reports

• "Viewing Log Files" on page 18
• "Changing Logging Levels" on page 20
• "Generating a Grid Status Report" on page 23
• "Viewing a Grid Status Report" on page 23

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:

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

Changing Logging Levels

Changing the log level for an application is a way of changing the amount and type of information that is added to the log files.

Log Levels

Applications may log information under different categories referred to as levels. There are six levels: INFO, NOTE, WARN, ERROR, DEBUG and TRACE. The levels may all be enabled or disabled individually except for the ERROR level, which is always enabled.
• **ERROR** - A log entry that is made under the ERROR level is obviously important to react to. Log entries in this level are marked as errors in the management pages.

• **WARN** - Similar to the ERROR level, the WARN level is marked as warnings in the management pages. Warnings are less critical than errors but should still be investigated.

• **INFO and NOTE** - The INFO and the NOTE level represent general information and information that is slightly more noteworthy than general information but still not a warning.

• **DEBUG and TRACE** - The two remaining levels, DEBUG and TRACE, are not enabled by default but may be enabled in order to get more fine-grained information. This may be useful in a situation where you are trying to find out the cause of a problem.

**Loggers**

Applications create log entries using one or more named loggers. The name of the logger typically reflects different parts of an application or different concepts within the application. The log levels may be changed for an application as a whole or they may be changed for individual loggers. In a log file you may see which logger was used to produce a particular log entry. It is the name written after the log level in the file.

**Why Change Log Levels?**

The most common reason to change the log level is that you are having some problem with an application and you want to get more information in the logs in order to find the cause. In this case, you typically enable the TRACE and/or the DEBUG level.

**Changing Log Levels**

Once you have decided to change a log level, you may do it in two different ways. You may change it in existing (running) nodes only or you may change it permanently as part of the configuration. If done as part of the configuration, new nodes will get the new log level when they are started.

**Changing Log Levels in Running Application Nodes**

Note that changing log levels in this way overrides log levels defined in the configuration of the grid, but the change is not persisted as the configuration is. So, any changes made will be lost as soon as a node is stopped.

**To change log levels in running application nodes**

1. Navigate to the Topology Overview page in the Grid Management pages and click on the Logging link at the top of the page.

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

   You will be presented with a hierarchical list of loggers organized by application. For each logger you may also see the currently enabled log levels.

3. Locate your application in the list and possibly the individual logger you would like to change.
To permanently change log levels in the grid configuration

1. Navigate to the Configuration Manager.
2. Click the Applications link.
3. In the list, select the application that you would like to change log levels for.
4. Click Edit Properties.
5. Locate the property named Node Log Level in the Grid Defined Properties section. This section is below the application defined properties. You may need to expand it to see the Node Log Level property.
6. Edit the log level for the application by clicking on the value link in the value column.
7. In the resulting dialog box, select the log levels that you want.

   **Note:** If the node log level 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 102.

8. Be sure to save the configuration when you are done editing by clicking the save button at the top of the page.

More advanced configuration alternatives to the one described above exist. As described in "Property Contexts" on page 102, it is also possible to set the log levels in different contexts. You do that by clicking on the property name rather than the property value in the list described above.

Yet another more advanced way of configuring log levels is to configure different log levels for different loggers. This is done by another grid property called Node Log Detail Level. This property is found next to the Node Log Level property. The Node Log Detail Level property is a map from the logger name to a set of log levels. Add new entries as needed and specify the name of the logger as the key combined with the desired log levels.

**Note:** Changing the log level in the configuration does not affect existing application nodes. The changes only take effect when you start new nodes.
Generating a Grid Status Report

Use this procedure to generate a Grid Status report.

The Lawson 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 Lawson 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 Lawson 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 23.

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 24
• "Monitoring the State of the Grid from the Java Web Start Component” on page 26
• "Viewing Counters” on page 29

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 41.

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 41.
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:

<table>
<thead>
<tr>
<th>Menu</th>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Menu</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Menu

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open</td>
<td>Open a previously saved profile data log file.</td>
</tr>
<tr>
<td>Save</td>
<td>Save the current profiling data to a profile data log file.</td>
</tr>
</tbody>
</table>

### Actions Menu

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reset Profiler</td>
<td>Clear profiling data.</td>
</tr>
<tr>
<td>Stop Profiling</td>
<td>Stop profiling data.</td>
</tr>
<tr>
<td>Start Profiling</td>
<td>Start profiling data.</td>
</tr>
</tbody>
</table>

### View Menu

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size Unit</td>
<td>Change the displayed size unit for profiled data (Bytes, Kilobytes, Megabytes).</td>
</tr>
<tr>
<td>Time unit</td>
<td>Change the displayed time unit for profiled data (Nanoseconds, Microseconds, Milliseconds, Seconds).</td>
</tr>
<tr>
<td>Expand/Collapse</td>
<td>Show Methods - expand the profile data tree to only show methods, only applicable when tree is fully collapsed.</td>
</tr>
<tr>
<td></td>
<td>Expand All - fully expand the profile data tree.</td>
</tr>
<tr>
<td></td>
<td>Collapse All - fully collapse the profile data tree.</td>
</tr>
<tr>
<td>Aggregation</td>
<td>Change the aggregation levels for the data.</td>
</tr>
<tr>
<td>Refresh</td>
<td>Refresh the panel.</td>
</tr>
</tbody>
</table>

### 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. 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".
• 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 shows 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

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

- "Configuring Memory Given to Applications" on page 30
- "Configuring Bindings" on page 31
- "Changing the Display Name of a Grid" on page 32
- "Viewing Configuration History" on page 32
- "Reverting to an Older Grid Configuration" on page 32
- "Changing How JVMs Are Launched on the OS Process Level" on page 33
- "Changing the Grid Agent Service User on Windows" on page 38

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 102.

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 102.

### 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 102, 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 100.
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 right corner of the page.

Changing How JVMs Are Launched on the OS Process Level

Grid nodes are JVMs. When a Lawson 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 Lawson 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 Lawson 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 102.

**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.
3. Click the Grid Properties link.

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

5. 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.

6. 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.

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

8. 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).

9. Click the Save button to close the dialog box.

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 Lawson 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 Lawson 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:
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:

   \texttt{START \textbackslash B \textbackslash 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 Lawson 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 Lawson 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 another 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 Lawson 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 Lawson 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 Agent Service User on Windows" on page 38, 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 Grid Agent Service User on Windows

Use this procedure if you need the grid agent service to run as a different user than the default user (which is generally Network Service). This procedure has two main parts:

• Change the user assigned to the grid agent service.
• Add the required user to the local group "Lawson Grid - <Grid Name>". This group was created by the installation to simplify the process of running the grid as another user than the default (Network Service) on Windows. This group is used to secure the file system resources used by the grid.

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 agent service

1 Access the Windows Server Manager on the server where the grid agent service runs.
2 Under Configuration, select Services.
3 In the list of services, locate the grid agent service. It will have a name in the format: GridAgent- 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.
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" on page 40
- "Listing All Ports Exposed by Routers in a Grid" on page 41
- "How to Use HTTP and HTTPS Ports" on page 42
- "Working With Context Root Discovery URLs" on page 42
- "Using a Web Server as a Proxy Server to Redirect HTTP Traffic to the Grid" on page 44

## 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 Lawson 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 Lawson 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 Lawson 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 Lawson Grid may be unaware of the correct ports to use, there is a mechanism in the Lawson 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 42.
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 41 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 Lawson 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
<?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>
    </module>
    <contextRoot type="Rest Service" href="m3api-rest">
      <url authType="none">http://server1.mycompany.com:10080/m3api-rest/</url>
    </contextRoot>
  </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. Currently, the grid supports setting up Internet Information Server, IBM HTTP Server, and Apache HTTP Server as proxy servers.

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 using LCM (see "HTTPS and SSL Certificates" on page 66).

**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](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.

**To set up IBM HTTP Server or Apache HTTP Server as a proxy server**

1 In the LifeCycle Manager Applications tab, right-click the grid you want to work with and select Lawson Grid \textit{<version>} > Grid Maintenance > Configure Web Server.

2 In the Select http router field, select the router you want the HTTP server to forward the traffic to.

3 Select the context roots you want to expose in the web server and click Next.

4 The result of this task is a basic configuration file for the HTTP server. You can either save it or copy and paste the result into your httpd.conf file. The configuration can be applied on the general level or on a virtual host. The commented part is for non-SSL offloading situations where that configuration goes into your HTTPS-configured virtual host.

5 If you are not using SSL offloading, you need to transform your certificate to the crt format and make sure the SSLProxyCACertificateFile directive references the correct file.
Managing the Grid

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 46
- "Managing the Grid Programmatically" on page 50

Managing the Grid Through the LifeCycle Manager

- "Putting Applications or Parts of the Grid in an Off-Line State" on page 46
- "Stopping an Individual Grid Node" on page 48
- "Starting New Grid Nodes (Application Instances)" on page 49

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

In the Lawson 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 31.

Managing the Grid Programmatically

• "Programmatically Operating on the Grid using REST" on page 50

• "Using the Grid Script Client to Manage the Grid" on page 54

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 Lawson 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 operates on the Lawson 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 41.
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 "Generating Client Certificates from LifeCycle Manager" on page 65.

REST Services Exposed by the Lawson 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

<table>
<thead>
<tr>
<th>Resource</th>
<th>Type</th>
<th>Description</th>
<th>Sample Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>info</td>
<td>GET</td>
<td>This document contains basic grid version information and context roots defined by applications installed into the grid.</td>
<td>/grid/info</td>
</tr>
<tr>
<td>status</td>
<td>GET</td>
<td>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.</td>
<td>/grid/status</td>
</tr>
<tr>
<td>hosts</td>
<td>GET</td>
<td>This returns status information for hosts that are part of the grid. Information examples are OS version, system time, offline state, and memory usage.</td>
<td>/grid/hosts</td>
</tr>
<tr>
<td>ports</td>
<td>GET</td>
<td>This returns information about all ports used by the grid and what purpose they serve.</td>
<td>/grid/ports</td>
</tr>
</tbody>
</table>
### Resource Descriptions

<table>
<thead>
<tr>
<th>Resource</th>
<th>Type</th>
<th>Description</th>
<th>Sample Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>bindings</td>
<td>GET</td>
<td>This returns information about all bindings that are defined in the grid. This information may be used when starting nodes using the startnode service: <code>/grid/host/hostName/startnode?binding=bindingName</code></td>
<td><code>/grid/bindings</code></td>
</tr>
<tr>
<td>nodes</td>
<td>GET</td>
<td>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.</td>
<td><code>/grid/nodes</code></td>
</tr>
<tr>
<td>offline</td>
<td>POST</td>
<td>Puts the grid in an offline state.</td>
<td><code>/grid/offline</code></td>
</tr>
<tr>
<td>online</td>
<td>POST</td>
<td>Puts the grid in an online state.</td>
<td><code>/grid/online</code></td>
</tr>
<tr>
<td>stop</td>
<td>POST</td>
<td>Stops the grid with the given grace period and reason.</td>
<td><code>/grid/stop</code></td>
</tr>
<tr>
<td>resetLogCounters</td>
<td>POST</td>
<td>Resets the log counters for the grid.</td>
<td><code>/grid/resetLogCounters</code></td>
</tr>
</tbody>
</table>

### Host-related REST Services

<table>
<thead>
<tr>
<th>Resource</th>
<th>Type</th>
<th>Description</th>
<th>Sample Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>offline</td>
<td>POST</td>
<td>Puts the host in an offline state.</td>
<td><code>/grid/host/hostName/offline</code></td>
</tr>
<tr>
<td>online</td>
<td>POST</td>
<td>Puts the host in an online state.</td>
<td><code>/grid/host/hostName/online</code></td>
</tr>
<tr>
<td>startnode</td>
<td>POST</td>
<td>Starts the node using a specified binding on the specified host.</td>
<td><code>/grid/host/hostName/startnode?binding=bindingName</code></td>
</tr>
</tbody>
</table>

### Application-related REST Services

<table>
<thead>
<tr>
<th>Resource</th>
<th>Type</th>
<th>Description</th>
<th>Sample Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>start</td>
<td>POST</td>
<td>Starts the specified application. Optionally, you can specify if the application should be put in an offline state after starting it.</td>
<td><code>/grid/application/appName/start?offline=false</code></td>
</tr>
</tbody>
</table>
### Sample Path

**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?filter=ERROR&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`  
`/grid/node/JVMID/stop?graceSeconds=-1&reason=reasonDescription&block=true`  
`offline` | POST | Puts a node (JVMID) in an offline state. | `/grid/node/JVMID/offline`
<table>
<thead>
<tr>
<th>Resource</th>
<th>Type</th>
<th>Description</th>
<th>Sample Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>online</td>
<td>POST</td>
<td>Puts a node (JVMID) in an online state.</td>
<td>/grid/node/JVMID/online</td>
</tr>
<tr>
<td>nodeWeight</td>
<td>GET</td>
<td>Returns the node weight of a node (JVMID).</td>
<td>/grid/node/JVMID/nodeWeight</td>
</tr>
<tr>
<td>nodeWeight</td>
<td>POST</td>
<td>Sets the node weight of a node (JVMID).</td>
<td>/grid/node/JVMID/nodeWeight?weight=100</td>
</tr>
<tr>
<td>resetLogCounters</td>
<td>POST</td>
<td>Resets the log counters for a node (JVMID).</td>
<td>/grid/node/JVMID/resetLogCounters</td>
</tr>
<tr>
<td>status</td>
<td>GET</td>
<td>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.</td>
<td>/grid/node/JVMID/status</td>
</tr>
</tbody>
</table>

### Configuration-related REST Services

<table>
<thead>
<tr>
<th>Resource</th>
<th>Type</th>
<th>Description</th>
<th>Sample Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>config/binding/update</td>
<td>POST</td>
<td>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.</td>
<td>/grid/config/binding/update/?name=binding_Name&amp;host=host_Name [&amp;host=host_Name...] &amp;initial=initial_value&amp;max=max_value &amp;min=min_value</td>
</tr>
</tbody>
</table>

### Using the Grid Script Client to Manage the Grid

The Lawson 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 50.

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;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

<table>
<thead>
<tr>
<th>Command</th>
<th>Results and Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>startgrid</td>
<td>SUCCESS - grid successfully started</td>
</tr>
<tr>
<td></td>
<td>TIMED_OUT - grid not started within timeout period</td>
</tr>
<tr>
<td>stopgrid</td>
<td>SUCCESS - grid successfully stopped</td>
</tr>
<tr>
<td></td>
<td>TIMED_OUT - grid not stopped within timeout period</td>
</tr>
<tr>
<td>gridoffline</td>
<td>SUCCESS - grid is set offline</td>
</tr>
<tr>
<td>gridonline</td>
<td>SUCCESS - grid is set online</td>
</tr>
<tr>
<td>Command</td>
<td>Results and Descriptions</td>
</tr>
<tr>
<td>---------------</td>
<td>----------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>gridstatus</td>
<td>REGISTRY_STOPPED - the registry is stopped</td>
</tr>
<tr>
<td></td>
<td>ADMINROUTER_STOPPED - the admin router is stopped</td>
</tr>
<tr>
<td></td>
<td>GRIDAGENT_STOPPED - reported once for each grid agent that is stopped</td>
</tr>
<tr>
<td></td>
<td>GRIDAGENT_RUNNING - reported once for each grid agent that is running</td>
</tr>
<tr>
<td></td>
<td>OK - registry, router, and all grid agents are running</td>
</tr>
<tr>
<td></td>
<td>STOPPED - when only grid agents are running</td>
</tr>
<tr>
<td></td>
<td>OFFLINE - the grid is offline</td>
</tr>
<tr>
<td></td>
<td>ONLINE - the grid is online</td>
</tr>
<tr>
<td></td>
<td>NO_OFFLINE_STATUS - unable to get offline status (the grid might be stopped)</td>
</tr>
<tr>
<td>hostoffline</td>
<td>SUCCESS - host is set offline</td>
</tr>
<tr>
<td>hostonline</td>
<td>SUCCESS - host is set offline</td>
</tr>
<tr>
<td>hoststatus</td>
<td>GRIDAGENT_RUNNING - the grid agent is running</td>
</tr>
<tr>
<td></td>
<td>GRIDAGENT_STOPPED - the grid agent is stopped</td>
</tr>
<tr>
<td></td>
<td>OFFLINE - the host is offline</td>
</tr>
<tr>
<td></td>
<td>ONLINE - the host is online</td>
</tr>
<tr>
<td></td>
<td>NO_OFFLINE_STATUS - unable to get offline status for this host (the grid might be stopped)</td>
</tr>
<tr>
<td>applicationstatus</td>
<td>OK - the application global status returns OK</td>
</tr>
<tr>
<td></td>
<td>NOT_OK - the application global status returns NOT OK</td>
</tr>
<tr>
<td></td>
<td>COMMENT - each line of the application status report is reported as a COMMENT line</td>
</tr>
<tr>
<td></td>
<td>NO_APPLICATION_STATUS - unable to get application status</td>
</tr>
<tr>
<td></td>
<td>STOPPED - the application is stopped</td>
</tr>
<tr>
<td></td>
<td>OFFLINE - the application is offline</td>
</tr>
<tr>
<td></td>
<td>ONLINE - the application is online</td>
</tr>
<tr>
<td></td>
<td>NO_OFFLINE_STATUS - unable to get offline status for this application (the grid might be stopped)</td>
</tr>
<tr>
<td>applicationoffline</td>
<td>SUCCESS - application is set offline</td>
</tr>
</tbody>
</table>

Managing the Grid
<table>
<thead>
<tr>
<th>Command</th>
<th>Results and Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>applicationonline application</code></td>
<td>SUCCESS - application is set online</td>
</tr>
<tr>
<td><code>startapplication application</code></td>
<td>SUCCESS - application is started</td>
</tr>
<tr>
<td></td>
<td>TIMED_OUT - application not started within timeout period</td>
</tr>
<tr>
<td><code>stopapplication application</code></td>
<td>SUCCESS - application is stopped</td>
</tr>
<tr>
<td></td>
<td>TIMED_OUT - application not stopped within timeout period</td>
</tr>
</tbody>
</table>
Session Providers

Some applications in the grid require users to be authenticated and have an established grid session to allow access to their published services. There are three ways to establish a grid session: log on using a user name and password, log on using a Lawson SSO token or using the public key infrastructure of a grid, and log on using the SSL handshake.

The SSL handshake method is built in to the framework of the grid and is used internally to ensure grid integrity and security. It is possible to provide external grid clients with the necessary key material so that a connection can be made as an authenticated user. Since this is an intrinsic mechanism, no session provider needs to be configured for clients to log on to the grid. This method of connection is used in a bootstrap situation, for example, when the grid is initially installed.

The other two methods are delegated to a pluggable architecture called the session provider. A session provider is a grid application that has been given the special right to define who is allowed to log on and to establish some sort of security context for a user. A session provider offers its services through
a grid-provided interface (the SessionProvider interface hierarchy) so that other applications can use the services without having to know the implementation details.

A configured session provider is typically needed when you have applications in the grid that require users to be authenticated.

The security context contains information about the identity of the user and a list of associated roles. How the session provider assigns roles to the user is implementation dependent, and different session providers may choose different mechanisms. For example, an LDAP-aware session provider might look up the list of groups from a specified directory entry. A session provider will always add two default roles in addition to the looked-up ones, the default role (the empty string) and the user name itself.

All session providers give a grid administrator a last say in the assignment process through role mapping. Through role mapping, the session provider roles are mapped to grid roles. For more information on role mapping, see "Role Mapping" on page 71.

The reason for having pluggable authentication architecture is flexibility. The grid framework can never foresee all possible flavors of user storage and credential mechanisms, so it makes sense to de-couple that functionality from the grid release cycle. This introduces a dilemma: the application programming interfaces used internally to create a session in the grid must be reachable for every application since a session provider is an ordinary grid application, and as an administrator you would like to have some control over how users are allowed to log in to the grid (or at least you would like to know who is allowed to perform this gatekeeper task). The grid framework solves this by requiring an application that should be allowed to create grid sessions to be registered as a session provider in that particular grid. Only an administrator can grant session provider privileges to an application, thus making this procedure secure. An application that tries to create sessions without being registered as a session provider will fail with an error report.

Session Provider Requirements and Selection

There are four different session providers available as grid extensions for production scenarios. The purpose of each of these session providers is to provide an authentication or validation mechanism for the Grid. To determine the most appropriate one to use, consider the following information:

Session Provider Types

**Windows Session Provider**

This session provider uses the same authentication mechanisms as Windows itself and provides support for NTLM and Kerberos authentication. It must be installed on a Windows 2008 R2 or Windows 2012 server belonging to the Windows domain against which it will authenticate. The Windows Session Provider supports the following authentication methods: basic authentication, NTLM, and Negotiate.

**LDAP Session Provider**

This session provider supports complex authentication options, including multiple domains, server fail-over options, and authentication against standalone LDAP servers. The LDAP Session Provider can be used for authenticating users to any LDAP server, including Active Directory. The LDAP
Session Provider supports basic authentication using the LDAP authentication method Simple Authentication only. This session provider requires configuration for basic setup and to take advantage of its more powerful features.

The LDAP Session Provider should not be configured to directly connect to a Lawson Security LDAP server (such as the Infor Lawson System Foundation LDAP). When authenticating against a Lawson Security system, use the DSSO Session Provider.

**DSSO Session Provider**

The DSSO Session Provider can authenticate against Lawson Security (in Infor Lawson System Foundation or Landmark runtime environments) using a DSSO base component. It is also used within Landmark itself, where the DSSO Session Provider communicates directly with the main Lawson Security installation. This session provider is needed if you are running Lawson Smart Office in a grid, and you want to authenticate to your Infor Lawson System Foundation or Landmark runtime environment. In this scenario, the DSSO Session Provider requires the DSSO base component. For more information, see the *Distributed Single Sign-on Solution for Lawson Smart Office Installation Guide*. The DSSO Session Provider supports basic authentication.

**SAML Session Provider**

The SAML Session Provider authenticates users using SAML to communicate with AD FS 2.0. User credentials are stored in AD but also synchronized to Infor Federation Services (IFS) for extended attributes (Claims) and also security role assignment (which happens in IFS). The session provider supports the following authentication methods: basic authentication and SAML 2. The SAML Session Provider implements basic authentication using WS-Trust to authenticate users to AD FS 2.0 (for active, non-browser based clients). The SAML 2 authentication method uses WS-Federation (for browser clients that can be automatically redirected).

**Requirements and Selection**

There are four different deployment scenarios for the session providers:

- **AD FS 2.0 and Infor Federation Services**
  
  In this scenario, the users are authenticated to AD FS 2.0 using the SAML protocol. Infor Federation Services is installed on the AD FS server for the extra attributes it provides and also for automating configurations. This scenario applies when Infor Ming.le™ is used with AD FS 2.0.

- **Active Directory**

  In this scenario, Active Directory is used as the user information storage, but AD FS is not used. Users are authenticated directly to the AD.

- **Lawson Security**

  In this scenario, Lawson Security is used for user authentication. The session provider used in this scenario will relay the authentication request to the configured Lawson Security System (Infor Lawson System Foundation or Landmark runtime environment). Lawson Security may store the user credentials in any LDAP or even Active Directory, but this is irrelevant from the session provider’s point of view.

- **Other LDAP**
This scenario is for all other scenarios where users are stored in LDAP. The session provider authenticates the users directly to the LDAP server.

Choosing a Session Provider Based on the Scenario

An X in the matrix below means that the session provider supports the given scenario.

<table>
<thead>
<tr>
<th>Session Provider</th>
<th>AD FS 2.0/IFS</th>
<th>Active Directory</th>
<th>Lawson Security</th>
<th>Other LDAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows</td>
<td>-- X --</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LDAP</td>
<td>-- X --</td>
<td>-- X --</td>
<td>-- X --</td>
<td></td>
</tr>
<tr>
<td>DSSO</td>
<td></td>
<td>-- X --</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAML</td>
<td>-- X --</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Choosing between the LDAP Session Provider and the Windows Session Provider

If your choice is between the LDAP Session Provider and the Windows Session Provider, consider the following information:

<table>
<thead>
<tr>
<th>Platform Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP Session Provider 1.9</td>
</tr>
<tr>
<td>Windows Session Provider 1.9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fail-over Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP Session Provider 1.9</td>
</tr>
<tr>
<td>Windows Session Provider 1.9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Multiple Domain Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP Session Provider 1.9</td>
</tr>
<tr>
<td>Windows Session Provider 1.9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Configuration Differences</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP Session Provider 1.9</td>
</tr>
<tr>
<td>Windows Session Provider 1.9</td>
</tr>
</tbody>
</table>

If you meet the requirements for the Windows Session Provider and do not need the explicit fail-over support or multiple domain support, use the Windows Session Provider for its simplicity.
Secondary LDAP Servers and Multiple Domains for LDAP Session Providers

Secondary LDAP Servers

In the case of the LDAP Session Provider, you can add secondary LDAP servers to your grid configuration. The secondary server then serve as fail-over mechanisms.

The implementation checks each call to the LDAP server (that fails) and looks for some specific exceptions/errors. When one of the known errors is seen, it is interpreted as a failed server and the session provider switches to the next server in the list. The switch is done in a round-robin fashion, and the state is not saved between restarts of the session provider. Therefore, you must make sure to keep the primary server first in the list.

Note that if a fail-over occurs during an attempted logon, that logon will fail. The new server will be used by the next logon attempt. When a switch happens, an INFO message similar to the following is logged in the SessionProvider log:

```
2011-11-04 08:12:38,525  INFO SessionProvider   SessionProvider: Switching server from sestw426.corpnet.lawson.com to ldapemea.corpnet.lawson.com
```

The configured servers, as well as the currently active server, can be seen by selecting the session provider in the LifeCycle Manager, and selecting Manage Application. If fail-over is configured, a list of the servers is shown, with an asterisk (“*”) next to the currently active server.

For procedures on including secondary LDAP servers in the session provider configuration, see the Lawson Grid Extensions Installation and Administration Guide.

Multiple Domains

If you have users in multiple domains, you can add those domains to the configuration for the LDAP Session Provider. The session provider will look for users in all domains simultaneously. Should a username exist in more than one domain, the logon will fail. The reason for this is that the session provider cannot know if the two users are identical, or if they should be treated differently.

For procedures on adding domains for the LDAP session provider configuration, see the Lawson Grid Extensions Installation and Administration Guide.

Changing the Session Provider

When a grid is initially installed, no application is allowed to create grid sessions. The only possible way to gain administrative privileges is to connect to such a grid through the LifeCycle Manager client and the SSL handshake. This means the initial task of defining a session provider must be performed from the LCM client.
To define a session provider

1. Install an application that can act as a session provider, that is, provides services according to the SessionProvider interface hierarchy. In a developer scenario this step is optional.

The grid comes with a built-in session provider that accepts all users and passwords. Do not use this session provider in a production scenario.

2. Grant session provider rights to the selected application from the grid Configuration Manager.

   a. On the Configuration Manager page, click the Session Providers link.

      The Session Providers page will appear. It displays information about the registered session provider and contains a drop-down list of eligible applications. Of these applications, two are pre-populated entries: "No Session Provider" and "The Developer Session Provider, intrinsic to the grid."

   b. Select the preferred application and click the Grant button.

   c. Save the configuration changes by clicking the diskette icon at the top left corner of the page.

      Note that the changes are applied immediately, and the Session Providers page is updated to reflect the new configuration.

Note: There might be a perceived discrepancy between the Registered Session Provider field and the Application granted Session Provider rights field on the page. The latter can contain more entries than the first. This is because when you register a session provider, you are actually registering an application type (the name in between the parentheses in the drop-down list), not a deployed application instance. You will notice this behavior if you have an application type that is deployed multiple times under different names. The result is that all the instances will be granted session provider rights as indicated under the Application granted Session Provider rights field.

Configuring Router WWW Authentication Methods

Each router in a grid can be configured to provide specific authentication methods for its HTTP and HTTPS ports, respectively. The reason for this is to allow different authentication methods via different entry points in the grid. It can also be desirable to force use of basic authentication, for example, to go over the HTTPS port.

The WWW authentication methods configured for a router define the methods accepted by the router when accessed via the HTTP or HTTPS ports. When accessed by a grid client, the intersection of these methods and the methods supported by the configured session provider determines how authentication is performed in practice. For information on the supported authentication methods for each session provider, see "Session Provider Requirements and Selection" on page 59. Basic authentication is supported by all session providers.
If SAML 2 authentication is configured for a router, it will override the other settings. The SAML Session Provider must be the configured session provider in order for this setting to take effect. Note that when the SAML Session Provider is installed, a router (SAML Router) configured to use SAML 2 authentication is automatically installed. The SAML 2 authentication method in the SAML Session Provider uses WS-Federation to authenticate to AD FS 2.0 (for browser clients that can be automatically redirected). The SAML Session Provider also implements basic authentication using WS-Trust to authenticate users to AD FS 2.0 (for active, non-browser based clients).

It is possible that the configured session provider does not support any of the configured WWW authentication methods. In this case, the user will not be able to log on.

**To configure WWW authentication methods for a router**

1. Navigate to the Configuration Manager for the grid you are working with.
2. Click Routers.
3. Click on a router and select the WWW Authentication Methods tab for the HTTP or HTTPS port.
4. Select the appropriate authentication methods and click Update. (Note that the router will automatically restart if these fields are modified.)
5. Click Save and then Save again.

**Certificates**

- "Certificates in the Grid" on page 64
- "Generating Client Certificates from LifeCycle Manager" on page 65
- "Importing Trusted Certificates via the Configuration Manager" on page 66
- "HTTPS and SSL Certificates" on page 66

**Certificates in the Grid**

Certificates contain information about the server the clients are connecting to. That information is guaranteed to be correct by a trusted third party, that is, an authority both the grid and the clients trust. Certificates are part of the public key infrastructure defined by a grid and are mainly used for two purposes:

- To provide a means for grid clients to authenticate themselves without having to pass through the more complex gate of session provider authentication. This mechanism is used internally by the LifeCycle Manager client and is set up automatically by the Grid LifeCycle Manager installer.
• To ensure the server identity for clients connecting through the various communication endpoints in the grid, that is, the HTTPS endpoints and the legacy security-enabled sockets.

Generating Client Certificates from LifeCycle Manager

Grid clients can use certificates to connect to a grid. A typical use would be for a user who is running a grid management console and wishes to be able to log on with administrative rights whether there is a session provider registered or not.

In this procedure, the LifeCycle Manager administrator accomplishes two tasks:

1. Generates an RSA key pair.
2. Signs the private key with the grid private key and inserts the grid certificate into the client certificate chain.

The key store that is created is a Java key store file (JKS), usable by any Java application. Please note that the key store contains secret information and any user who knows the password to the key store and has access to the key store file itself can connect to this particular grid as the user which the key store was created for. This is especially important for key stores created with the grid-admin role.

To create necessary key material for certificates

1. In the left pane of the LifeCycle Manager, navigate to the specific grid you want to work with.
2. Right-click on the grid name and select Lawson Grid 10.1.8.0 > Grid Maintenance > Manage Security.
3. Select the Generate client keystore radio button and click Next.
4. Enter values for the following:
   - **Keystore name**: The keystore name will be used for two things. The first is as a name of the generated key store and the second is the name of the user the client will be authenticated as in the grid (that is, all the actions the client will perform will be performed in the name of this user).
   - **Keystore password**: The password is the password for protecting the generated key store.
   - **Role list separated by**: The role list is a list of security roles the client will be granted, for example, grid-admin or FND-ADMIN.
The format in which the keystore will be exported. You can select one of the following values:

- **JKS** - Java Keystore, the native keystore type for Java applications. File extension (.ks)
- **BKS** - Bouncy Castle Keystore, provided by the Bouncy Castle crypto provider. Works with Java and especially with applications developed for Android devices. File extension (.bks)
- **PKCS12** - A standard format, published by RSA Laboratories and usable in a Windows environment. File extension (.p12)

The directory is where the generated keystore file will be saved.

5. Click Next and then Finish.

Importing Trusted Certificates via the Configuration Manager

The certificate of an external CA can be imported into the grid host SSL truststores to ensure that the grid hosts trust that Certificate Authority. Imported CA certificates must be DER or Base64 encoded.

To import trusted certificates via the Configuration Manager

1. In the Configuration Manager, click Advanced Configuration and then Certificates.
2. For each host you wish to add the certificate to, click Manage Certificates and perform the following steps.
3. Click Import Signed Certificate.
4. Browse for the signed certificate.
5. Click View certificate.
6. Click Import.

HTTPS and SSL Certificates

When a client connects to one of the HTTPS or security-enabled socket communication endpoints, the server will present itself with an SSL certificate. The default certificate is signed with the grid certificate, so in order for clients to trust those endpoints, the grid certificate will have to be imported into the client's trust store.
To export grid certificate and import to client

1. In the left pane of the LifeCycle Manager, navigate to the specific grid you want to work with.
2. Right-click on the grid name and select Lawson Grid 10.1.8.0 > Grid Maintenance > Manage Security.
3. Select the Export Grid SSL certificate radio button and click Next.
4. Select the folder to store the certificate in.
5. Click Next and then Finish.

The name of the file will be `gridName.der` and the format will be a DER-encoded X.509 certificate. This file can be imported into Windows "Trusted Root Certification Authorities" trust store or any Java-based trust store mechanism.

If you want the server to present itself using a certificate from another Certification Authority, you need to create a certificate signing request, which is a standard way to apply for a signed digital certificate.

To create a certificate signing request

1. Navigate to the Configuration Manager for the grid you are working with. For information on how to access the Configuration Manager, see "Navigating between the Grid Administration Tools" on page 11.
2. Click the Advanced Configuration link.
3. On the Advanced Configuration page, click the Certificates link.
4. Click the Manage Certificates link for the host for which you want to create the certificate signing request.
5. Click the Create Certificate Signing Request (CSR) button.
6. On the CSR Info tab, consider the following fields:

   Host FQDN (CN)    The fully qualified domain name of the host. This is the name clients can reach this host by. Case is not important.

   This field is referred to as CN, common name, and is the bare minimum of information to be able to create a CSR. Many commercial certificate authorities require you to submit more information.

   Organization Unit (OU)   Some type of section name, expected by the certificate authority.

   Organization (O)         For example, your company name.

   Locality Name (L)        For example, city.

   State or Province (ST)   The full state or province name.

   Country (C)              Select the ISO 3166-1, alpha 2 code for your country from the drop-down list.
It is possible to add additional names this host is known by as well, although not all CAs will consider such a request.

Note that there is an Advanced tab of the Create Certificate Signing Request dialog box as well. If you need to change the cryptographic parameters, (for example, the key length or signature algorithm), you can enter values in their respective fields on this tab. Use names according to http://download.oracle.com/javase/6/docs/technotes/guides/security/StandardNames.html. These values are not checked for validity until the request is created. The Advanced tab includes a free-format country (C) field if your country is not in the list on the CSR Info tab.

7 When done, click the Create Request button. Read the warning and click yes (or no). The request will be presented in the text field as a base-64 encoded PKCS #10 certificate request.

8 Copy the text or download the text file and submit it to your certification authority. Make sure that you receive the full certificate chain from the certification authority, in a format that can be read by Java, for example, pem-format (base-64 encoded binary certificate surrounded by -----BEGIN CERTIFICATE----- and -----END CERTIFICATE-----, or in a PKCS #7 container, for example, any file with a p7b suffix.

9 Click OK when ready.

10 When you receive the signed certificate chain from the certification authority, return to the same grid Configuration Manager page and click the Import Certificate button. Browse to where you stored your certificate file and click View Certificate. Make sure that the full certificate chain is present and click Import.

11 Restart all router nodes on that host.

Role Mappings

This section describes the types of role domains and how to add or modify role mappings.

- "Roles and Role Domains" on page 68
- "Role Mapping" on page 71
- "Defining Role Mappings in the Configuration Manager" on page 73
- "Adding a New Role" on page 75

Roles and Role Domains

Role domains are the types of roles available in the Lawson Grid. The domains differ in what they apply to, ranging from global roles to application-specific roles. When you configure security and role mappings, it is important to keep track of the different role domains and what the purposes of the different role domains are.
In the Lawson Grid, there are three role domains to consider:

- Session provider defined roles
- Grid application defined roles
- Global Lawson Grid defined roles

**Session Provider Defined Roles**

Each session provider can have different defined roles than other session providers. When a user authenticates through a grid's particular session provider, the session provider can then communicate to the grid what roles the user belongs to. The list of roles that the session provider claims a user belongs to is unknown to the grid and if you change the session provider, the list of roles will most likely change.

The roles a session provider has are dependent on how the session provider operates and how it is configured. For example, an LDAP-aware session provider may look up the list of roles from a specified directory entry. Another session provider implementation may look up roles from a database. Session provider defined roles may be related to organization aspects or geographical aspects of a company. An authenticated user, for example, may belong to such roles as:

- Germany-Office (geographical example)
- Sales-Department (organizational example)
- Manager (organizational example)
- Database-Admin (example relevant for that company)
- GRID-ADMINISTRATOR (users intended to administer a Lawson Grid)
- M3-ADMINISTRATOR (users intended to administer an M3/BE)

**Grid Application Defined Roles**

Grid application defined roles are roles that are defined by each grid application. The Lawson Grid gives each grid application the following implicit roles:

- `applicationName/app-admin`

  This role grants access to all configuration and operational tasks for a particular application from the grid perspective. It is up to an application architect to use or not use this role for application private administrative tasks, such as tasks and operations defined on the application-provided management pages. The following list shows what an app-admin can access from the application-related grid management and configuration pages:

  - starting and stopping a specific application
  - creating and changing bindings for a specific application
  - creating and changing context roots for a specific application
  - creating and changing properties for a specific application
• undeploying a specific application
• performing any of the functions available to an `applicationName/app-poweruser` (see below)

• `applicationName/app-poweruser`
  This role grants access to a limited set of configuration and operational tasks for a particular application from the grid perspective. It is up to an application architect to use or not use this role for application private administrative tasks, such as tasks and operations defined on the application-provided management pages. The following list shows an app-poweruser can access from the application related grid management and configuration pages:
  • setting log levels for a specific application
  • resetting the error/warning count for a specific application
  • creating and deleting role mappings for a specific application
  • setting a specific application on or off-line

• `applicationName/app-user`
  This role is not used internally in the grid but is defined for completeness and is meant to allow applications to protect operational tasks with a global role that is not app-admin or app-poweruser.

**Note:** The implicit application roles are used by the generic APIs of the Lawson Grid. For example, users who are mapped to the role grid-admin may start or stop any application from the Management UI, but if a user isn’t mapped to grid-admin, he may still start and stop a particular application if he is mapped to that application’s `applicationName/app-admin` role.

An application developer may define more roles that are relevant for that application in addition to the roles mentioned above. For example, the developer may define a database administrator or a reviewer role.

When designing a grid application, the developer may associate functions in the different application APIs relative to those grid application defined roles. For example, if the application has a feature that will perform a backup of the database (belonging to the application), the developer may assign a subset of the available application defined roles to that feature. The subset of roles from the example above could be:
  • Database-admin
  • `applicationName/app-admin`
  • grid-admin

If the backup feature of the application is designed with the set of roles above, then authenticated users who belong (are mapped to) any of the application defined roles in this set would be allowed to perform the action. So, users who are mapped to Database-admin will be able to perform the action, but also users who are mapped to `applicationName/app-admin` or grid-admin will be able to perform the action.

**Global Lawson Grid Defined Roles**

There are also some roles belonging to the Lawson Grid itself. These can be viewed as application defined roles where the application in this case is the Lawson Grid itself.
The Lawson Grid defines the following roles:

- **grid-admin**
  This role grants full access to all configuration and operational tasks for all applications and the grid itself.

- **grid-poweruser**
  This role grants access to a limited set of operational tasks for all applications and the grid itself, including:
  - setting log levels
  - resetting the error/warning count
  - starting/stopping applications
  - setting applications on or off-line
  - starting/Stopping nodes
  - setting nodes on or off-line
  - controlling profiler settings
  - controlling counters history and sweep delay
  - controlling thread pool settings

- **grid-user**
  This role is not used internally in the grid but is defined for completeness and is meant to allow applications to protect operational tasks with a global role that is not grid-admin or grid-poweruser.

**Role Mapping**

To implement security, you must perform a mapping between the session provider defined roles (which the grid is initially unaware of) to the application defined roles and global grid defined roles (which the grid is aware of). Until this mapping exists, the grid has no way to control what features or functions an authenticated user has access to.

Mapping basically involves defining a set of session provider defined roles that map to each application defined role. However, to perform this mapping, you must understand two aspects of roles: synthetic session provider roles and how the grid resolves role mapping for a user.

**Synthetic Session Provider Defined Roles**

When a user is authenticated by a session provider, the session provider will associate the user with a set of session provider defined roles. When the Lawson Grid asks the session provider for this list of associated roles, it adds two synthetic roles: **authenticated** and the username of the user. This means that all authenticated users will always be associated to at least two groups (even if the session provider actually doesn’t do any association). The purpose of this is that even if the session provider isn’t configured to associate a user with any roles, you at least get the user name and the fact that the
user is authenticated. This enables you either to map specific users to application defined roles or to simply say that all authenticated users should be mapped to a particular application defined role. For example, all authenticated users can be mapped to the applicationName/app-user role, meaning that all users may use that particular application.

How Role Mappings Are Resolved

Once role mappings have been set up, the grid will have a set of application defined roles that are linked to each user's roles, either the synthetic session provider defined roles or the other session provider defined roles. How the grid resolves the mapping of these roles depends on the specific mappings, of course, but also on whether the roles are configured to exclude or include access.

Exclusion Versus Inclusion

Role mappings may be including or excluding. Including role mappings are the normal intuitive ones where you say such things as: everyone in the session provider role A should belong to (be included in) the application defined role B. Excluding role mappings are the opposite. In this case, you explicitly say that if a user is a member of role C, that user is excluded from role D.

Including role mappings are always resolved first. When all including role mappings have been examined and a set of roles have been formed, a final examination of excluding role mappings is performed. If matched roles are found, the mapped roles will be removed from the set. Normally you work with including role mappings but at times it is helpful to be able to express exceptions in the form of excluding role mappings. For example, if you want to achieve the result that everyone working at the Germany office should be able to use and administer an application except if they work for the sales organization, you may do that by defining two mappings:

1. First, an including mapping saying that all members of the session provider group "Germany-Office" should be included in the application defined role applicationName/app-admin
2. Second, an excluding mapping saying that all members of the session provider group "Sales-Department" should be excluded from the application defined role applicationName/app-admin

The Role Mapping Process

The process of finding the set of application roles is as follows:

1. The session provider reports a set of session provider defined roles for an authenticated user.
2. The Lawson Grid adds the synthetic roles described above to that list.
3. Given this list of roles, the defined (including) role mappings are examined to find other (application defined) roles that are mapped from any of the existing roles in the list. Roles that have a matching mapping are added to the list and a new pass over the list will try to find even more matching mappings (based on the added roles). This process repeats until no new roles are found and added.
4. At this point, the excluding role mappings are examined and roles are removed from the list if there is a matching mapping.
Defining Role Mappings in the Configuration Manager

Use this procedure to configure role mappings for a grid.

You can configure role mappings belonging to a particular application on a configuration manager page belonging to that application (a page with an application centric focus) or you may configure role mappings on a global page enabling you to operate on all application defined roles including the ones defined by the Lawson Grid itself (grid-admin, grid-poweruser & grid-user). Which is most convenient for you depends on the situation.

Navigation to Role Mapping Pages

How you navigate to the pages for editing role mapping depends on whether you want to work with the global role mappings page (where the roles for all applications are available) or with an application-centric role mappings page (where only the roles for one application are available).

To navigate to the global role mapping page

1. Navigate to the Configuration Manager.
2. Click the Users and Role Mapping link.
   
   A page with all application defined roles will be shown, with the possibility to edit both the including and the excluding mappings.

To navigate to an application-centric role mapping page

1. Navigate to the Configuration Manager.
2. Click the Applications link.
3. From the list of deployed applications, click on the application you would like to configure.
4. Click the Edit Role Mappings link at the top of the page.
   
   A page with all application defined roles for this application will be shown, with the possibility to edit both the including and the excluding mappings.

Regardless of whether you used the global or the application-centric role mapping page, you will now be presented with a page that lets you define and edit role mappings.

Editing Role Mappings

To edit role mappings

1. After navigating to the role mappings pages as described above, identify an application defined role that you want to edit mappings for from the list.
2 Click the Edit link in the Included Members column in order to create including mappings, or click on the Edit link in the Excluded Members column in order to create an excluding mapping.

After you click the Edit link, you are presented with a dialog box that shows all roles already mapped to this application role. You can use this dialog box to remove some or all of the mappings, or to access the Add Role Mappings dialog box that allows you to select new roles to be mapped. For important information about the Add Role Mappings dialog box, see “Add Role Mappings Dialog Box” on page 74.

3 If you want to add role mappings, click Add to access the Add Role Mappings dialog box.

4 Select the type of domain whose roles you want to map to the role you selected above.

5 Select the role you want to map to the role you selected above and click Add. (You must click Add after you select the role, or else the role mapping will not be formed.)

6 Repeat the previous two steps until you have completed the role mapping.

Add Role Mappings Dialog Box

This dialog box always has a particular (application defined) role in focus. The role in focus is the role belonging to the Edit link you clicked in order to get to this dialog box. So, this dialog box is used to map other roles to the role in focus. If you want to map roles to another role, you should close this dialog box and select the different role. This dialog box has three parts belonging to the three different role domains.

1 The first is the global domain. Here you may select from the global Lawson Grid defined roles. This enables you to specify such things as: If you are global grid-admin, you should also be app-admin in application A (provided that it is that application defined role you are currently working with). The custom field in this domain section lets you specify individual uses.

2 The second domain is application defined roles. Here you may select from all application defined roles of all applications deployed in the grid. This enables you to specify such things as: If you are app-admin in application A, you should also be app-admin in application B. The custom field in this domain enables you to form mappings from application defined roles that aren’t properly declared by the application but that you actually know exist. This situation is unusual and should not be used unless so instructed by that application’s manual.

3 The final domain is the session provider domain. Here you may select session provider defined roles and map them to the application defined role that you are currently working with. The list of session provider defined roles may be long, so the dialog provides a filter mechanism that enables you to sort through the roles before adding them.

Regardless of which domain you pick your roles from, a corresponding Add button to the right is used to actually create the mapping. Note that the Add button has to be clicked to actually form the mapping. Selecting a role and closing this dialog box without clicking the Add button will not form a mapping. The dialog box is designed so that you may select and add many roles from different domains before finally closing the dialog box. As you add roles in this dialog box, the list of roles that will be added when closing the dialog box is displayed at the bottom.
Adding a New Role

Use this procedure to add new roles or groups of roles in addition to those defined for the grid.

Creating new roles or groups of roles can make it easier to set up mapping. The groups of roles can be managed together and used to map to different target roles. For example, assume the session provider defined three roles that are related to the geographical structure of a company:

- Germany-Office
- UK-Office
- US-Office

Each user is assigned to exactly one of the three roles but you would like to map all of the three roles to many different application defined roles. It would be cumbersome to add all three roles to each of the target application roles. Furthermore, if the organization grows with a new location of Sweden-Office, you would need to edit all the mappings and add the new role to all target roles.

In this situation, it is better to create a group of all the three roles called Any-Office and use that group when mapping to the application defined target roles. In this case, when the new Sweden-Office location appears, you only need to add it to the Any-Office group and all existing mappings will be "updated" as a consequence.

You may work with role mappings on a global page or on an application-centric page targeting a specific application. Both enable you to create new roles but the domain the new roles belong to will depend on the page used to create the role.

Thus, a role created on the global page will belong to the global role domain and a role created for a specific application will be in the application role domain. This distinction has no runtime significance but you need to understand in which domain a new role will end up so that you can find it when creating mappings using the role.

To add a new role or group or roles

1. Navigate to the global Users and Role Mappings page or the corresponding application-centric application Users and Role Mappings page, depending on your needs. For more information, see "Role Mapping" on page 71.
2. Click the Add new... link.
3. In the dialog box, provide a name for the role, click Create, and then click the Save button.
   The new role should now be listed in the list of roles and you may start adding roles to this role in the same way as if it was any of the other roles in the list.
4. After mapping some roles to the role, you may start using this role when mapping to other roles.
Troubleshooting

- "Introduction to Troubleshooting" on page 76
- "Investigating Problems or Issues" on page 76
- "Disaster Recovery" on page 79
- "Maintenance" on page 86

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 12). 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 77
- "Getting a Heap Dump or Thread Dump Via the Grid Agent" on page 77
- "Reviewing Each Application's Configuration" on page 78
- "Diagnosing Network Issues" on page 78
- "Gathering Information When Reporting a Problem" on page 79
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 77, 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 Lawson 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 18. 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 23 and include that when reporting the problem.

If 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 Lawson Grid and How It Is Stored on Disk" on page 86 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" on page 80
- "Configuring the Registry for Active-Passive Failover Operation" on page 85
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. "Creating a Non-Transient Host" on page 81
2. "Stopping the Grid" on page 81
3. "Moving the Registry" on page 82
4. "Removing the Missing Host" on page 82
5. "Synchronizing the Grid After Removing a Missing Host" on page 83
6. "Dealing with Deployed Applications or Parts of Applications that Only Existed on the Removed Host" on page 83
7. "Replacing Missing Routers and Connection Dispatchers" on page 83
8. "Removing a Missing Host from the Configuration" on page 84
Creating a Non-Transient Host

If you are uncertain whether or not there exist non-transient hosts, you can use the following procedure:

To reconfigure a host to be non-transient

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 expand the tree under the grid.
3. Expand the grid host node in the tree so that all hosts are shown.
4. Double-click the host you intend to move the registry to. In the information panel to the right (expand if necessary), information about the host is displayed in the Grid Host tab. If it is listed as Transient the host needs to be reconfigured.
5. If it was transient, click the Reconfigure Host link in the Tasks panel.
6. Uncheck the Transient check box. Click Next and Finish.

Note: Although you may be inclined to add a new host to the grid and have that new host replace the missing registry host, this is not recommended. The recommended procedure is to first follow the instructions below that describe how to move the registry to an existing grid host. When that is completed and you have an operational grid, a new host may be added and the registry may again be moved to that one.

Important: The recommendation is to make sure that the transient hosts, if any, are started when moving the registry. When a transient host is started, it looks for the registry where it was the last time the transient host was running. This means that if the registry is reconfigured to a new address or port while some transient hosts are not running, those transient hosts will be confused when they start since they will not be able to find the registry in the old place. If this recommendation can’t be followed, a manual synchronization of the grid will be needed when the transient hosts are started. See “Synchronize the Grid” on page 89, “Synchronizing the Grid After Removing a Missing Host” on page 83, and "Synchronizing the Configuration Between Hosts" on page 86.

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 Lawson Grid tab.
4  Select 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 or they are all transient. See "Creating a Non-Transient Host" on page 81 for instructions to solve this.
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

**Synchronizing the Grid After Removing a Missing Host**

The next thing to do is to synchronize the grid configuration between the remaining hosts so that they become identical and contain the correct information. This enables you to finally start the grid again.

1 In the LifeCycle Manager, in the tree to the left, double-click on the grid you are working with

2 In the panel to the right, select the Lawson Grid tab.

3 Select Synchronize Grid. The synchronization may take some time.

4 In the same panel, select Start Grid.

5 When the grid has finished starting, verify that the grid is operational by clicking on Grid Management Pages in the same panel. The management pages should be displayed.

   At this point, the basic operational status of the grid is restored. However, things may still be missing that resided on the missing host. The typical things to consider if they have to be replaced are:

   - Deployed applications or parts of applications that only existed on the removed host.
   - Routers and attached connection dispatchers. Those things will be discussed below.

**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 104. 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" on page 65 and "HTTPS and SSL Certificates" on page 66.

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 Lawson Grid 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 Lawson Grid 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 83 and "Replacing Missing Routers and Connection Dispatchers" on page 83.

### 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 10.1.9.0.14 or greater
- At least two hosts which are not configured to be transient

**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 Lawson Grid 10.1.x.x > 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 Lawson Grid 10.1.x.x > Host Maintenance > Configure Registry Failover.
3 Select the blank entry in the list of hosts.
4 Click Next and then click Finish.

**Synchronizing the Configuration Between Hosts**

If hardware problems occur, the grid configuration files may become out of sync. In this case, a manual synchronization should be performed as described in "Synchronizing the Grid After Removing a Missing Host" on page 83.

**Maintenance**

- "The Lawson Grid and How It Is Stored on Disk" on page 86
- "Performing Disk Backups of the Grid" on page 87

**The Lawson 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.

To find out where a configuration area is stored on disk, for a particular host, do as follows:

**To locate grid configuration areas on disk**

1 Start LifeCycle Manager.
2 Select the Hosts tab of the left pane.

3 In the list, double-click on the host that contains the configuration area you are interested in.

4 In the Information panel to the right, information about the host is displayed. Look for the path entry. It contains the path on the host where LCM is storing things related to this host. For example, the path could be \texttt{C:/LifeCycle/HostName1}.

5 There will be a subfolder to the path above that is named "grid". Within this folder, each grid installed on this host has its own configuration area. This configuration area folder has the same name as the grid.

6 So, if we have a path as in the example above and an instance of a grid named Grid1, the full path to the configuration area will be \texttt{C:/LifeCycle/HostName1/grid/Grid1}.

When looking within a configuration area, you will find a set of subfolders. The most important of these are:

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

---

### Performing Disk Backups of the Grid

It is recommended to regularly back up all configuration areas on all hosts participating in a grid. To locate the configuration area, see "The Lawson Grid and How It Is Stored on Disk" on page 86.

**Note:** When backing up the configuration areas, 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.
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" on page 88
- "Administering Grid Hosts" on page 90
- "Administering Grid Applications" on page 93
- "Grid Management Pages" on page 97
- "Configuration Manager" on page 97

Administering the Grid

This section describes administrative tasks related to the entire grid.

- **Start the Grid**
  
  Use this task if you want to start the grid. The registry and all routers will be started. Also, as a consequence of starting the grid, all applications that are configured to automatically start when the grid starts will be started.

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

  2. In the Lawson Grid tab of the right pane, click Start Grid.

  When the task is finished, the grid is started, which is indicated by a green arrow on the icon in the Applications list.

- **Stop the Grid**

  Use this task if you want to stop all applications that run in the grid together with all routers and the registry. The grid agents will be left running.

  **Note:** Before stopping the grid, it is recommended to put the grid in an off-line state.
In the Applications tab in the left pane in the LifeCycle Manager, double-click on the grid you want to stop.

In the Lawson Grid tab of the right pane, click Stop Grid.

When the task is finished, the grid is stopped, which is indicated by a red square on the icon in the Applications list.

Synchronize the Grid

Use this procedure to synchronize the grid and force the redistribution of grid configuration files to all hosts. Internally the grid keeps track of grid hosts, various network addresses and so on in a set of configuration files. These configuration files have to be identical on all hosts that are part of a grid. Normally the grid tooling ensures that this is the case. If, for some reason, this fails to be the case, it is possible to manually synchronize the configuration files from the LifeCycle Manager. It is always safe to perform this synchronization. If the grid already was in sync, no harm will be done.

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

In the Lawson Grid tab of the right pane, click Synchronize Grid.

When the task is finished a window is displayed. Click OK or Click View Log.

Apply New Grid Version

Use this procedure to apply a new minor or fix pack version of the grid to the hosts in the grid. The procedure for upgrading from one major version of the grid to another major grid is described in the Grid Installation Guide.

Important: Only perform the upgrade if all applications that are deployed in the grid are compatible with the new Grid version.

Note: Before you apply the new grid version to the hosts in the grid, you must have installed the new grid version in LifeCycle Manager. Also, you must stop the grid agents and the grid.

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

In the Lawson Grid tab of the right pane, click Apply New Grid Version.

On the Summary window, verify the properties provided.

Click Finish.

Start the grid and the grid agents again.

Remove a Grid

Remove all hosts one at a time, as described in "Administering Grid Hosts" on page 90.
Administering Grid Hosts

This section describes administrative tasks related to grid hosts.

- **Stop Grid Agent**
  Normally grid agents should be running since they are needed when starting new nodes on the hosts.
  Use this procedure in specific cases, such as an upgrade of the grid, to temporarily stop the grid agent.
  
  1. Double-click on the grid host in the left pane in LifeCycle Manager.
  2. In the Lawson Grid tab of the right pane, click Stop Grid Agent.
  3. When the task is finished, a window is displayed. Click OK or Click View Log.

  Stopping a grid agent does not affect existing application nodes. The existing application nodes will continue to be operational.

- **Start Grid Agent**
  Use this procedure to start the Grid Agent that was previously stopped.
  
  1. Double-click on the grid host in the left pane in LifeCycle Manager.
  2. In the Lawson Grid tab of the right pane, click Start Grid Agent.
  3. When the task is finished, a window is displayed. Click OK or Click View Log.

- **Add 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.
  It is safe to add hosts to a grid while the grid and existing applications are running and in use.

  **Note:** In order to be able to add a host, the host has to be running an LCM Service so that LifeCycle Manager is able to communicate with it.
  
  1. Double-click on the Grid Hosts node in the left pane in LifeCycle Manager.
  2. In the Lawson Grid tab of the right pane, click Add Host.
  3. On the Add a New Host to Grid window, enter the following:

     - **Host**: Select a new host to add to the grid.
Select the IP address of the host. This address will be used by the grid agent, registry, and nodes on this host.

**Host Address**
Select the IP address of the host. This address will be used by the grid agent, registry, and nodes on this host.

**Host DNS Name**
Type the DNS name for the host.

**Host Description**
Enter a description for this host.

**JDK**
Select the JDK to use on this host. If “Default” is selected, the same JDK as the LifeCycle Manager Service is using will be used.

**Transient**
Specify if the host should be transient or not. Transient hosts can be inaccessible without the grid treating this as an error.
See below for more information about transient hosts.

**Grid Agent Port**
Enter the port number for the grid agent on this host. This port is used for configuration purposes. It must be accessible from the LifeCycle Manager server.

Click Next.

On the Summary window, verify the properties provided.

Click Finish.

**Transient Hosts**
Normally, the grid will consider a host that is inaccessible as an error. In most cases, this is what you want. You typically have a set of hosts and they all have applications installed that are vital to your operation so if one of them is inaccessible, you want this to be treated as an error.

However, you may have a host that is only intended to be used at certain situations. A good example is that you may have an application installed on a non-transient host and that host is capable of handling the load except at certain peak situations. At those peak situations, you want to add an extra host that can share the load. When the load is lessened, you want to be able to stop the extra host. This could be solved, for example, by creating a virtual machine and adding that as a transient host to the grid. By doing this, the extra transient host may be started when needed and later stopped without the grid considering this to be an error.

**Reconfigure Registry**
Use this procedure to move the Grid Registry from one host to another host in the grid or to reconfigure the ports of the administrative router or registry.

**Note:** If the reason for moving the registry is that the registry host is missing, due to, for example, a hardware failure, you should use the procedure described in “Disaster Recovery” on page 80.
**Important:** The recommendation is to make sure that the transient hosts, if any, are started when reconfiguring the registry. When a transient host is started, it looks for the registry where it was the last time the transient host was running. This means that if the registry is reconfigured to a new address or port while some transient hosts are not running, those transient hosts will be confused when they start since they will not be able to find the registry in the old place. If this recommendation can’t be followed, a manual synchronization of the grid will be needed when the transient hosts are started. See "Synchronizing the Configuration Between Hosts" on page 86, "Synchronizing the Grid After Removing a Missing Host" on page 83, and "Synchronize the Grid" on page 89.

**__1__** Double-click on the host containing the registry in the left pane in LifeCycle Manager.

**__2__** In the Lawson Grid tab of the right pane, click Move Registry.

**__3__** When the task is finished a window is displayed. Click OK or Click View Log.

**Reconfigure Host**

Use this procedure to change the IP address or port number associated with a host.

**Note:** Before you reconfigure the host, stop the grid agents and the grid.

**__1__** Double-click on the grid host in the left pane in LifeCycle Manager.

**__2__** In the Lawson Grid tab of the right pane, click Reconfigure Host.

**__3__** Modify the properties and click Next.

**__4__** Click Finish.

**__5__** When the task is finished, a window is displayed. Click OK or Click View Log.

**__6__** Start the grid and the grid agents again.

**Change the JDK/JRE for Grid Hosts**

If a Lawson Grid is installed using a particular JDK version (see the Grid Installation Guide) and you later want to change to another JDK version, you may do that by configuring a Lawson Grid property. As always with grid properties, they may be configured and overridden in different contexts (see "Grid Properties" on page 102). 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 may just repeat the procedure for each host.

**__1__** Navigate to the Configuration Manager for the Lawson 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.
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).

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

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

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

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.

Remove Host

Use this procedure to remove a host in the grid that is not needed any longer or is replaced by another.

Obviously, before removing a host, all application nodes on that host should be stopped. It is strongly recommended to put the host in an off-line state before stopping nodes. For more information on putting a host in an off-line state, see "Putting Applications or Parts of the Grid in an Off-Line State" on page 46.

Note: If the reason for removing the host is that the host is missing, due to, for example, a hardware failure, you should use the procedure described in "Disaster Recovery" on page 80.

Double-click on the grid host in the left pane in LifeCycle Manager.

In the Lawson Grid tab of the right pane, click Remove Host.

A confirmation window is shown. Complete the wizard and click Finish to remove the host.

When the task is finished, a window is displayed. Click OK or Click View Log.

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.

Double-click on the application in the left pane in LifeCycle Manager.

In the Lawson Grid tab of the right pane, click Start Application.

When the task is finished, a window is displayed. Click OK or Click View Log.
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. Double-click on the application in the left pane in LifeCycle Manager.
2. In the Lawson Grid tab of the right pane, click Stop Application.
3. When the task is finished, a window is displayed. Click OK or Click View Log.

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. Double-click on the application in the left pane in LifeCycle Manager.
2. In the Lawson Grid tab of the right pane, click Manage Application.

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. Double-click on the application in the left pane in LifeCycle Manager.
2. In the Lawson Grid tab of the right pane, click Monitor Application.

The Grid monitor for 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. Double-click on the application in the left pane in LifeCycle Manager.
2. In the Lawson Grid tab of the right pane, click Configure Application.

The configuration page for the 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 94.

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. Double-click on the application in the left pane in LifeCycle Manager.
2. In the Lawson Grid tab of the right pane, click Deploy Application on Hosts. The application details, name, type, version and gar-file name are displayed.
3. Select the check box next to the name of the hosts on which you want to deploy the application. Click Next.
4. Click Finish.
5. When the task is finished, a window is displayed. Click OK or Click View Log.

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.

1. Double-click on the application in the left pane in LifeCycle Manager.
2. In the Lawson Grid tab of the right pane, click Remove Application from Hosts. The application details, name, type, version, and gar-file name are displayed.
3. Select the check box next to the name of the host on which you want to remove the application. Click Next.
4. If the application is running on the host, a confirmation window is shown where you must confirm the shutdown of the nodes before removing the application. Select the check box **Confirm Automatic Shutdown of Nodes** and click Next.
5. Click Finish.

When the task is finished, a window is displayed. Click OK or Click View Log.
Upgrade Application

Use this procedure to upgrade an already existing application in the grid with a new gar-file. The new version will be used when deploying this application to new hosts. It is also possible to downgrade an application using this wizard.

When doing the upgrade, running application nodes will be shut down. However, in order give you better control of this process, it is better and recommended to first put the application in an off-line state. When the application is done processing ongoing requests, the application should finally be stopped before proceeding with the upgrade.

While you perform the upgrade, all hosts (including transient hosts) that have the application installed must be accessible.

1. Double-click on the application in the left pane in LifeCycle Manager.
2. In the Lawson Grid tab of the right pane, click Upgrade Application.
   The application details, name, type, version, and gar-file name are displayed.
3. Select the new gar-file that you have previously downloaded.
   Click Next.
4. Select the hosts on which you want to upgrade the application.
5. If the application is running, a confirmation window is shown where you must confirm shutdown of the nodes before upgrading the application.
   Select the check box **Confirm Automatic Shutdown of Nodes** and click Next.
6. Click Finish.
7. When the task is finished, a window is displayed. Click OK or Click View Log.

Uninstall Application

Use this procedure to uninstall an application running in the grid. All configuration details for this application are removed when the application is uninstalled.

When uninstalling the application, running application nodes will be shut down. However, in order give you better control of this process, it is better and recommended to first put the application in an off-line state. When the application is done processing ongoing requests, the application should finally be stopped before proceeding with uninstalling.

1. Double-click on the application in the left pane in LifeCycle Manager.
2. In the Lawson Grid tab of the right pane, click Uninstall Application.
   The application details, name, type, version, and gar-file name are displayed as well as the hosts on which the application is currently deployed.
   Click Next.
3. If the application is running on a node, a confirmation window is shown where you must shut down the node before uninstalling
   Select the check box **Confirm automatic shutdown of nodes** and click Next.
4. Click Finish.
Grid Management Pages

The Grid Management Pages are used to monitor and manage the behavior of the grid. An overview is available on the first page where the state of the grid and all its hosts are shown.

Overview

The topology shows a logical view of the hosts running in the grid and all nodes running on them. The status of the hosts and nodes, the port numbers used, memory usage, errors, warnings, CPU usage and up time are shown. It is possible to start new nodes using the start button on each host. It is also possible to shut down a node using the stop (square) button.

Logging

Under the Logging heading, it is possible to change the log levels and view the content of the log files.

Applications

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.

Advanced

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.

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.

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The configured name of the installed application.</td>
</tr>
<tr>
<td>Version</td>
<td>The application version.</td>
</tr>
<tr>
<td>Type</td>
<td>The application type. An application may be installed several times in the same grid under different names.</td>
</tr>
<tr>
<td>Hosts</td>
<td>A list of hosts that the application is installed on.</td>
</tr>
<tr>
<td>Remove button</td>
<td>Uninstalls the application from one or several hosts.</td>
</tr>
</tbody>
</table>

**Note:** Avoid removing applications from the Configuration Manager if the grid is managed by LifeCycle Manager. In that case, use the uninstall function in LifeCycle Manager instead.

**Deploy Application**

*Note:* Avoid doing this from the Configuration Manager if the grid is managed by LifeCycle Manager. In that case, use the deploy function in LifeCycle Manager instead.

Click Deploy Application to install an application. Select a gar file, 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).
<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property</td>
<td>The property title if defined (the property name otherwise). Clicking on this link lets you edit the property in different contexts.</td>
</tr>
<tr>
<td>Value</td>
<td>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.</td>
</tr>
<tr>
<td>Unit</td>
<td>The unit of the property, for example, seconds or MB.</td>
</tr>
<tr>
<td>Type</td>
<td>The type of the property, for example, Integer or StringType</td>
</tr>
<tr>
<td>Description</td>
<td>A description of the property.</td>
</tr>
<tr>
<td>Name</td>
<td>The property name (which may be different from the title shown in the Property column).</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module</td>
<td>The grid application module that is exposing the web application or web server.</td>
</tr>
<tr>
<td>Type</td>
<td>Indicates if the context root is for a web application or web services.</td>
</tr>
<tr>
<td>Web Application</td>
<td>The name of the web application if any.</td>
</tr>
<tr>
<td>Context Root</td>
<td>The name of the context root.</td>
</tr>
<tr>
<td>Remove button</td>
<td>Removes this context root.</td>
</tr>
</tbody>
</table>

**Grid Properties**

Grid-defined properties are listed.

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

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.

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of the router.</td>
</tr>
<tr>
<td>Running</td>
<td>Indicator if the router is running. Green bullet means that the router is running.</td>
</tr>
<tr>
<td>Host</td>
<td>The host that the router is configured to run on.</td>
</tr>
<tr>
<td>Port</td>
<td>The port the router is listening on.</td>
</tr>
<tr>
<td>External Address</td>
<td>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.</td>
</tr>
<tr>
<td>Http Port</td>
<td>The HTTP port of the router if defined. The HTTP port serves web applications and web services.</td>
</tr>
<tr>
<td>Https Port</td>
<td>The HTTPS port of the router if defined. The HTTPS port serves web applications and web services.</td>
</tr>
<tr>
<td>Dispatcher</td>
<td>A list of connection dispatchers exposed by this router (if any).</td>
</tr>
<tr>
<td>Remove button</td>
<td>Removes this router.</td>
</tr>
</tbody>
</table>

**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" on page 63.

**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.

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Binding name</td>
</tr>
<tr>
<td>Running</td>
<td>Indicator if the router is running. A green bullet means that at least one node, started using this binding, is running.</td>
</tr>
<tr>
<td>Node Type</td>
<td>The node type that the binding is using when starting a node.</td>
</tr>
<tr>
<td>Min</td>
<td>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.</td>
</tr>
<tr>
<td>Initial</td>
<td>How many nodes to start at system start (Min &lt;= Initial &lt;= Max).</td>
</tr>
<tr>
<td>Actual</td>
<td>The actual number of running nodes that were started using this binding.</td>
</tr>
<tr>
<td>Max</td>
<td>The maximum number of nodes allowed (started with this binding).</td>
</tr>
</tbody>
</table>
### Column Description

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hosts</td>
<td>A set of hosts that this binding may use. This binding may only be used to start nodes on this set of hosts.</td>
</tr>
<tr>
<td>Remove button</td>
<td>Removes this binding.</td>
</tr>
</tbody>
</table>

**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

- "Session Providers" on page 58
- "Changing the Session Provider" on page 62

### 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:

- **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).

- **Property Overrides**
  Display a view of all property overrides grouped by override context. This is a very complex view and is only included for support reasons.

- **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 "Certificates in the Grid" on page 64, and "HTTPS and SSL Certificates" on page 66.
All properties for the Lawson Grid are defined below including a brief explanation of each property. The properties can be edited through LifeCycle Manager > Lawson Grid > Configuration Manager.

- "Grid Properties" on page 107

## Grid Properties

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

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Node Memory</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max Heap</td>
<td>integer, MB</td>
<td>Sets the maximum heap space for the JVM.</td>
<td>256 MB</td>
</tr>
<tr>
<td>Throttling Memory</td>
<td>integer, percent</td>
<td>Threshold, expressed as percentage of maximum heap, for when to start throttling the node (putting the node in an off-line state).</td>
<td>The default is 85% of max heap.</td>
</tr>
<tr>
<td>Throttling GC Frequency</td>
<td>integer, seconds</td>
<td>When throttling, how often a garbage collect should be performed.</td>
<td>5 seconds</td>
</tr>
<tr>
<td><strong>Node Logging</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Node Log Level</td>
<td>list</td>
<td>The logging level of the node: ALL, ERROR, WARN, INFO, NOTE, DEBUG, TRACE</td>
<td>ERROR, WARN, INFO, NOTE</td>
</tr>
<tr>
<td>Property</td>
<td>Type</td>
<td>Description</td>
<td>Default value</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------------------------------------</td>
</tr>
<tr>
<td>Node Log Detail Level</td>
<td>list</td>
<td>The logging level of a particular logger, the format is ([logger-name]=[levels]), for example, Node=INFO, DEBUG.</td>
<td></td>
</tr>
<tr>
<td>Node Log Capture Standard Out</td>
<td>boolean</td>
<td>Whether or not to capture stdout in the log file.</td>
<td>true</td>
</tr>
<tr>
<td>Node Log Capture Standard Err</td>
<td>boolean</td>
<td>Whether or not to capture stderr in the log file.</td>
<td>true</td>
</tr>
<tr>
<td>Node Log Max File Size</td>
<td>integer</td>
<td>The maximum size of a log file.</td>
<td>5 MB</td>
</tr>
<tr>
<td>Node Log Archive Age</td>
<td>integer, days</td>
<td>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.</td>
<td>The default is –1, which disables this feature.</td>
</tr>
<tr>
<td><strong>Node Profiling and Monitoring</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Profiler</td>
<td>enumeration</td>
<td>Turns the profiler on or off.</td>
<td>off</td>
</tr>
<tr>
<td>Node JMX Enabled</td>
<td>boolean</td>
<td>Setting to enable or disable a node’s JMX server.</td>
<td>true</td>
</tr>
<tr>
<td>Counters Poll Delay</td>
<td>integer, seconds</td>
<td>How many seconds to wait between polling counters.</td>
<td>15 seconds</td>
</tr>
<tr>
<td>Counters Keep History</td>
<td>integer, minutes</td>
<td>How many minutes of counter history to retain.</td>
<td>60 minutes</td>
</tr>
<tr>
<td>CPU Sampling</td>
<td>enumeration</td>
<td>Enables or disables CPU sampling.</td>
<td>The default is on (if the JVM supports it).</td>
</tr>
<tr>
<td>CPU Sampling Frequency</td>
<td>integer, seconds</td>
<td>How often to sample CPU usage.</td>
<td>5 seconds</td>
</tr>
</tbody>
</table>
| 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. }
<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thread Priority Capping Threshold</td>
<td>integer, percent</td>
<td>Threshold for thread priority capping based on CPU percentage.</td>
<td>85%</td>
</tr>
<tr>
<td><strong>Misc. Node Properties</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Node Weight</td>
<td>integer</td>
<td>Node weights are used as a load balancing mechanism. Each Node has a weight (&gt;= 0).</td>
<td>100</td>
</tr>
<tr>
<td>Node Startup Failure Shutdown Grace Period</td>
<td>integer, seconds</td>
<td>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.</td>
<td>60 seconds</td>
</tr>
<tr>
<td>Java Executable</td>
<td>path</td>
<td>The path to a Java executable.</td>
<td>The default is the one used to start the grid agent.</td>
</tr>
<tr>
<td>Working Directory</td>
<td>path</td>
<td>The path to use as working directory for a node. Use a forward slash (/) as the directory separator.</td>
<td></td>
</tr>
<tr>
<td>Library Path</td>
<td>path list</td>
<td>Adds the specified paths to the library path of the JVM. Use a forward slash (/) as the directory separator.</td>
<td></td>
</tr>
<tr>
<td>Generic JVM Commands</td>
<td>list</td>
<td>A list of generic JVM command line switches.</td>
<td></td>
</tr>
<tr>
<td>Debug Port</td>
<td>integer, port number</td>
<td>Adds JVM arguments to start the node in debug mode (not suspended).</td>
<td>The argument is the port where you attach your debugger.</td>
</tr>
<tr>
<td>Debug Port (Suspended)</td>
<td>integer, port number</td>
<td>Adds JVM arguments to start the node in suspended debug mode.</td>
<td>This property is deprecated. The argument is the port where you attach your debugger.</td>
</tr>
<tr>
<td>Property</td>
<td>Type</td>
<td>Description</td>
<td>Default value</td>
</tr>
<tr>
<td>----------</td>
<td>------</td>
<td>-------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Default Session Timeout for Ephemeral Sessions</td>
<td>integer, minutes</td>
<td>Sets the default timeout for ephemeral session, that is, sessions logged on as a result of an HTTP request without grid session header(s).</td>
<td></td>
</tr>
<tr>
<td>Stream Buffer Size</td>
<td>integer, KB</td>
<td>The buffer size for transporting stream data. A higher value will potentially increase throughput speed, but have a larger memory footprint.</td>
<td>64 KB</td>
</tr>
<tr>
<td>Http Idle Timeout</td>
<td>integer, seconds</td>
<td>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.</td>
<td>3600 seconds</td>
</tr>
</tbody>
</table>

**Module Thread Pool**

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thread Pool Max Queued</td>
<td>integer</td>
<td>Sets the maximum number of queued proxy request threads.</td>
<td>The default is 0 (unlimited).</td>
</tr>
<tr>
<td>Thread Pool Max Concurrent</td>
<td>integer</td>
<td>Sets the maximum number of concurrently executing proxy request threads.</td>
<td>The default is 0 (unlimited).</td>
</tr>
<tr>
<td>Thread Pool Max Pooled</td>
<td>integer</td>
<td>Sets the maximum number of pooled proxy request threads.</td>
<td>The default is 0 (unlimited).</td>
</tr>
<tr>
<td>Thread Pool Min Pooled</td>
<td>integer</td>
<td>Sets the minimum number of pooled proxy request threads.</td>
<td>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).</td>
</tr>
<tr>
<td>Thread Pool Warn Delay</td>
<td>integer, seconds</td>
<td>The number of seconds to wait for a proxy request to finish before the system puts a warning in the log.</td>
<td>The default is 0 (off).</td>
</tr>
</tbody>
</table>

**Module Web Service**
<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web service Chunked Transfer Encoding</td>
<td>boolean</td>
<td>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.</td>
<td>The default is true.</td>
</tr>
<tr>
<td>Web Service Auto Buffer Size</td>
<td>integer</td>
<td>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.</td>
<td>The default value is 16 KB</td>
</tr>
<tr>
<td>Misc. Module Properties</td>
<td></td>
<td>څ</td>
<td></td>
</tr>
<tr>
<td>Class Path</td>
<td>path list</td>
<td>Sets the specified paths as the class path for the module. Use a forward slash (/) as the directory separator.</td>
<td></td>
</tr>
<tr>
<td>Start Timeout</td>
<td>integer,</td>
<td>The timeout to wait for a node to start before declaring the start as failed.</td>
<td>The default is 120 seconds.</td>
</tr>
<tr>
<td></td>
<td>seconds</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stop Timeout</td>
<td>integer,</td>
<td>When stopping a module, how many seconds to wait for the module to stop by itself before forcefully shutting it down.</td>
<td>The default is 60 seconds.</td>
</tr>
<tr>
<td></td>
<td>seconds</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Security Properties</td>
<td></td>
<td>څ</td>
<td></td>
</tr>
<tr>
<td>Default Session Provider</td>
<td>string</td>
<td>Sets the name of the preferred session provider, if any, to use when requesting a provider using the SessionUtils getProvider(int type) method.</td>
<td></td>
</tr>
<tr>
<td>Default Session Timeout</td>
<td>minutes,</td>
<td>Sets the default timeout for sessions.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>integer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td>Type</td>
<td>Description</td>
<td>Default value</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Default Logon Service</td>
<td>string</td>
<td>Sets the name of the default logon service used when using session providers from this module.</td>
<td></td>
</tr>
<tr>
<td>Trusted Routers Http Entrypoints</td>
<td>map to string</td>
<td>Trusted routers HTTP entry point mappings.</td>
<td></td>
</tr>
<tr>
<td><strong>Grid Http Discovery Services</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Context root discovery urls</td>
<td>map to string list</td>
<td>Configures how context roots are presented when discovered through the grid info http service.</td>
<td></td>
</tr>
<tr>
<td><strong>Grid Internal Properties</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>grid.internal.jsm</td>
<td>boolean</td>
<td>Internal option. Only set to false if instructed by Lawson.</td>
<td>true</td>
</tr>
<tr>
<td>grid.internal.jna</td>
<td>boolean</td>
<td>Internal option. Used to disable native access functions. Only set to false if instructed by Lawson.</td>
<td>true</td>
</tr>
<tr>
<td>grid.internal.dup</td>
<td>boolean</td>
<td>Internal option. Used to disable property updates on an entity. Only set to false if instructed by Infor.</td>
<td>true</td>
</tr>
<tr>
<td>grid.internal.diepar</td>
<td>boolean</td>
<td>Internal option. Used to disable NTLM IE pre-auth request interception.</td>
<td>false</td>
</tr>
<tr>
<td>grid.internal.rme</td>
<td>boolean</td>
<td>Internal option. Used to disable role mapping resolution. Only set to false if instructed by Infor.</td>
<td>true</td>
</tr>
<tr>
<td>grid.internal.wdm</td>
<td>boolean</td>
<td>Internal option. Used to enable web ui tracing for developer purposes. Only set to false if instructed by Infor.</td>
<td>true</td>
</tr>
<tr>
<td>grid.internal.dnm</td>
<td>boolean</td>
<td>Internal option. Used to disable the network monitor. Only set to false if instructed by Infor.</td>
<td>true</td>
</tr>
<tr>
<td>Property</td>
<td>Type</td>
<td>Description</td>
<td>Default value</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>java.net. preferIPv4Stack</td>
<td>boolean</td>
<td>Internal option. Maps to the Java system property java.net. preferIPv4Stack. Only set to false if instructed by Infor.</td>
<td>true</td>
</tr>
<tr>
<td>java.net. preferIPv6Addresses</td>
<td>boolean</td>
<td>Internal option. Maps to the Java system property java.net. preferIPv6Addresses. Only set to true if instructed by Infor.</td>
<td>false</td>
</tr>
</tbody>
</table>
Index

A

administering
  grid applications, 93
  grid hosts, 90
applications
  configuration review, 78
  configuring, 93
  deploying, 93, 97
  ports, 40
  putting in off-line state, 46
  removing, 93
  stopping/starting, 93
application server, 8
Applications page, 15

B

bindings, 9
  configuring, 31

C

certificates, 64, 66
  generating, 65
  importing, 66
  trusted, 66
changing
  grid display name, 32
command line
  grid management, 54
configuration
  bindings, 31
  grid, 32
  grid JVMs, 33
  memory, 30
  properties, 107
  reviewing, 78
configuration area, 86
Configuration Manager, 11, 97
context root, 42
counters, 29

deploying
configuration area, 86
disk storage location, 86
domains
  multiple for grid, 62
downloading
  log files, 18
DSSO session provider, 59

failover, 85

gar files, 9
grid, 8
  administering, 88
  applications, 9
  configuration, 97
  configuration area, 86
  display name, 32
  failover, 85
  log viewer, 26
  monitoring, 12, 24
operating REST with, 50
ports, 40
profiler viewer, 26
properties, 107
putting in off-line state, 46
removing, 88
reverting configuration, 32
status pages, 24
status report, 23, 23
synchronizing, 80
upgrading, 88
URLs, 42
viewing configuration history, 32
grid agent
changing user, 38
grid agents
stopping/starting, 90
grid applications
administering, 93, 93
configuring, 93
deploying, 93, 97
monitoring, 97
removing, 93
stopping/starting, 93
grid hosts
adding, 90
administering, 90
removing, 90
grid JVMs
configuring, 33
Grid Management Pages, 11, 12, 97
HTML-based, 15
Java Web Start, 16
grid registry
moving, 90
Grid Script client, 54
grid user
changing for grid agent, 38

I
issues
reporting, 79

J
JVMs
collection, 33

K
key store, 65

L
Lawson Grid
administering, 88
components, 8
concepts, 9
LDAP
secondary servers, 62
LDAP session provider, 59
log files
downloading, 18
viewing, 18
loggers, 21
logging
changing log levels, 97
settings, 107
viewing log files, 97
logging levels, 20

H
heap dump
viewing, 77
hosts
non-transient, 80
putting in off-line state, 46
replacing, 80

M
mapping
roles, 71, 73
Max Heap property, 30
memory
configuration, 30
missing registry host, 80
monitoring
  grid, 12
  grid state, 24
moving
  registry, 80

N

networks
  diagnosing issues, 78
nodes
  properties, 107
  starting, 49
  stopping, 48
node types, 9

O

off-line state, 46
on-line state, 46

P

ports, 40
  analyzing, 42
  viewing, 41
properties
  grid, 107
proxy server
  web server as, 44

R

registry
  failover, 85
  moving, 80
registry host
  missing, 80
removing
  grid, 88

reporting
  problems, 79
reports
  grid status, 23, 23
REST
  programmatically operating grid with, 50
role domains, 68
  application, 69
  global, 70
session provider, 69
role mapping, 71, 73
roles
  adding for grid, 75
  application, 69
  global, 70
  grid, 68
  mapping, 71
  session provider, 69
routers
  authentication methods, 63

S

SAML session provider, 59
security context, 58
session provider
  DSSO, 59
  LDAP, 59
  SAML, 59
  Windows, 59
session providers, 58
  changing, 62
  requirements, 59
  selection, 59
types, 59
SSL, 66
starting
  applications, 93
  grid, 88
  grid agent, 90
  nodes, 49
status
  grid pages, 24
Status page, 14
stopping
  applications, 93
  grid, 88
grid agent, 90
individual node, 48
synchronizing
grid, 80
hosts, 86

T

thread dump
viewing, 77
thread pool properties, 107
threads, 77
Topology Overview page, 12
troubleshooting
introduction, 76
networks, 78
threads, 77

U

upgrading
grid, 88
user roles, 58

V

viewing
counters, 29
grid configuration history, 32
grid ports, 41
heap dump, 77
log files, 18
thread dump, 77

W

web server
as proxy server for grid, 44
Windows session provider, 59