pitney bowes

Spectrum™ Technology Platform
Version 12.0 SP2

Spectrum Spatial Administration Guide

Table of Contents

8 - Enterprise Routing Module

9 - Troubleshooting Your
System

1 - Introduction

In this section

What's Included in This Guide

Welcome to the *Spectrum Spatial Administration Guide*. This guide will help you build a web mapping application or embed mapping in an existing application using a variety of web services, capabilities, tools and sample code.

Addressed in this guide are:

- Configuring your system by changing the default port number or repository database; accessing the repository; accessing and uploading resources; configuring web services; and running Spectrum™ Technology Platform as a Linux service
- Managing security using the Management Console, including how to add users and roles, as well as how to apply security entity overrides
- Monitoring your system, including logging, viewing version and license information, using the JMX Console to monitor performance, and monitoring memory usage
- Managing memory and threading, including JVM performance tuning, adjusting pool size, and increasing heap memory
- Load balancing spatial services for resilience or high capacity
- Troubleshooting your system, including rebuilding a corrupt repository index and monitory memory usage of a non-responsive server

Additional Spectrum™ Technology Platform and Location Intelligence Module documentation is located online at **support.pb.com**.

2 - Configuring Your System

In this section

Changing the HTTP Port Number for Spectrum Spatial

The HTTP port is used to access all Spectrum™ Technology Platform web services, whether via REST or SOAP, and for the Welcome page, sample apps and Spatial Manager.

After Spectrum™ Technology Platform is installed, you can change the existing port settings that were assigned during installation by manually editing the global, startup, and individual service configuration files. There are several reasons you may need to change the port number:

- A port conflict occurs after installation.
- You want to try out a new version of Spectrum™ Technology Platform without removing your old one. Since you cannot install them both, you can turn off the existing version and install a Spectrum™ Technology Platform image that uses a different port.
- You need a proxy on port 8080 but have a limited number of ports to expose externally, so you would like to move Spectrum™ Technology Platform without re-creating all your settings and data flows.

  Note: This task is only for experienced administrators who have application server experience changing port numbers, as network port conflicts can result in module components failing to start. One indication that a component has failed to start is if it does not appear in the Management Console. To troubleshoot the problem, look at the Spectrum™ Technology Platform server wrapper log. This log shows which port is causing the problem. You can find the wrapper log in: `<install_folder>\server\app\repository\logs\wrapper.log`.

To make Spectrum™ Technology Platform run under the new HTTP port, a number of entries in properties and configuration files need to be changed. To change the service configurations, you must have WebDAV file editing enabled on the server. WebDAV is available on Windows and Linux servers but may need to be installed.

To change the port number:

1. In spectrum-container.properties change the value of `spectrum.http.port` to the new port number. This file is located in `<install_folder>/server/app/conf`.

2. In the java.properties file change all the `repository.host` ports and `image.webapp.url`. This file is located in `<install_folder>/server/modules/spatial`.

3. In , change the port numbers in these service configurations:

   - Mapping (only necessary when accessing the Mapping Service via SOAP and when the ReturnImage parameter for a RenderMap request is False)
   - WFS
   - WMS
   - WMTS

For instructions, see the "Spatial Manager Guide" in the Utilities section of the *Spectrum Spatial Guide*.

If you are relocating the server so it can use a different port, it is likely that the Spectrum™ Technology Platform server is not running. You will not be able to edit the service configuration files until the server is running. You will need to start the server, edit the configurations and restart the server.

4. Restart Spectrum™ Technology Platform so the ports and property changes can take effect.

Changing Your Repository Database Type

The Location Intelligence Module stores named resources (maps, layers, tables and styles), geographic metadata and configuration in a repository. In the default single server installation an embedded database is used to store these resources on the local server. There are several reasons you may need to use a database other than the embedded Derby database:

• To create a scalable solution that uses a resilient independent database.
• To use an in-house database preferred or dictated by your company.

In this release, the supported repository databases are Oracle, PostGreSQL/PostGIS, and Microsoft SQL Server. For instructions, see **Setting Up a Common Repository Database** on page 80.

Configuring the Web Services

You can, and frequently must, explicitly specify the desired behavior of the Location Intelligence Module web services via settings in each web service's configuration file. The configuration files for web services in the Location Intelligence Module are held in the Location Intelligence Module repository as named configuration.

> **Note:** Named configurations are not like other named resources that are held in the repository. You cannot use the Named Resource Service to access named configurations. Instead, you must use a WebDAV tool such as WebFolders.

Configuration files are pre-loaded in the repository for the Mapping, Feature, Map Tiling, WFS, WMS, and WMTS services. These configuration files are located at `http://hostname:port/RepositoryService/repository/default/Configuration/`.

For information about the name and location of each web service's named configuration in the repository, as well as a list of the configuration parameters for each web service, refer to the "Working With Spatial Services" chapter in the *Spectrum Spatial Developer Guide*.

Controlling Geometry Node Representation

The Location Intelligence and Routing modules provide a property that allows to you control the number of digits that represent geometry nodes returned in a web service response. By default, geometries are returned without placing a limit on the number of digits, which could be as many as 16 digits long. The extra digits add unnecessarily to the payload of a JSON or SOAP response, particular when large polygons or many records are returned. It also has the potential of setting an expectation of accuracy that is not in the data. The difference of one in the least significant digit might be a value of a billionth of a meter. For example, 3989657.014543291 and 3989657.014543292 differ by one billionth of a meter. Spatial data rarely has anything close to that accuracy. By setting the property to true, the values are trimmed typically to 9 or 10 significant digits. Using the previous example, the value would be returned as 3989657.01 which has an accuracy of a centimeter.

To trim the node values, add the following property to %Spectrum%\server\bin\wrapper\wrapper.conf and restart the server.

```
wrapper.java.additional.xx=-Dcom.pb.midev.service.output.geometry.useprecision=true
```

where xx is the number of the next available line in the section.

The coordinate values will be handled the same way for all geometries across services, whether for SOAP or REST calls, including services exposed from a data flow. This includes the Location Intelligence Module's Feature Service, Mapping Service, Geometry Service, Map Tiling Service, WMS, WMTS, and WFS and the Enterprise Routing services.

Applications that are editing polygon data through the web services should not use this property if it is possible that by writing back trimmed geometries, small overlaps or gaps might be created with neighboring geometries.

Disabling Accuracy Files for Datum Transforms

Spectrum Spatial supports conversions between certain datums by using algorithms that help shift coordinates more accurately. A separate jar file that contains these algorithms is installed by default for each datum transform located in the *Spectrum Installation Location*\server\app\types directory:

- midev-core-coordsys-irishtm-*version number*-onprem.jar for Irish Transverse Mercator
- midev-core-coordsys-jgd2000-*version number*-onprem.jar (also enables the updated version, JGD2011) for Japanese datums
- midev-core-coordsys-nadcon-*version number*-onprem.jar for US Nad27-Nad83
- midev-core-coordsys-ntv2-*version number*-onprem.jar for NTV2, which contains multiple conversions for many countries.

> **Note:** An XML file inside this jar controls which conversions are in use. To disable specific conversions within that file, stop the server and extract the XML file from the jar. Use an editor to set the entries to "false" for each conversion you want to disable. Add the edited XML file back into the jar, then restart the server. Similarly, if you want to enable the conversion, set the entries to true. For details see **Enabling NTv2 Transform**.

• midev-core-coordsys-rgf93-*version number*-onprem.jar for French Lambert conversions

By default, all of these jar files are loaded; however, their use can negatively affect the performance of certain operations. These conversions can be disabled in some cases, such as when you do not require a certain type of conversion (for example, if you have no need to convert Japanese datums) or the performance gains outweigh the benefits of accuracy at lower zoom levels.

To disable a specific transform:

1. Stop the server.
2. Remove the jar from the directory. Alternatively, you can rename the jar file to have a different extension (for example, .jar~) which will prevent it from being loaded.
3. Restart the server.

Configuring Request Timeouts

Spectrum Spatial allows you to set a timeout for SOAP and REST operations as part of a request to the Mapping and Feature services. The timeout is enabled by default with a value of 300 seconds (5 minutes).

To apply the timeout, entry and intermediate pointcuts need to be configured. This is done in the aop.xml located under server/modules/spatial/. The file includes several implementations that you can use. The entry point is the point where the timeout starts measuring time. The intermediary points are where the timeout checks if the operation timed out.

Use this, for example, when you want to apply a timeout to SOAP and REST renderMap methods and some intermediary steps (calls to the database, searching tables, retrieving candidates).

To adjust the default timeout value of 300 seconds, edit the timeout property for the Mapping and/or Feature services in the java.properties located under /server/modules/spatial.

```
timeout.mapping.value=300
```

```
timeout.feature.value=300
```

If the specified timeout value is <= 0, then the timeout will be disabled.

After changing the timeout value changes, restart Spectrum™ Technology Platform.

Configuring the Volatile Attribute for Named Tables

Volatility is an indication to Spectrum Spatial that information from a data source can change at any time. The default value for TAB, SAP HANA, and JDBC-based (Oracle, SQL Server and PostGIS) named tables is set to true, meaning that for each data access operation, such as a query or insert, Spectrum Spatial checks with the data source to find out if the table is volatile and if so, whether the data changed. If the data has changed, the cache is flushed and the table is reloaded before the data access operation can proceed. If the table did not change, the query or other operation is carried out on the data in the cache. See **Supported Data Sources** in the Resources and Data section of the *Spectrum Spatial Guide* for more information about for what triggers a change for each data source.

Volatility is enabled (set to true) for named tables that are uploaded from MapInfo Professional using **Map Uploader**. Volatility is also enabled for any named tables created with **Spatial Manager**. Older named tables in the repository are considered to be volatile but will not indicate that when viewed in the Spatial Manager table details page.

Disabling volatility should be done only on tables that do not change. For example, when generating tiles from volatile TAB files, the operation will perform very slowly. If you are using PostGIS, you may also want to consider disabling volatility to avoid encountering connection errors in Spatial Manager (for example, when viewing the sample rows on the table details page).

Volatility can be disabled on the table details page in Spatial Manager. See the Utilities section of the *Spectrum Spatial Guide* for more information on creating and modifying named tables in Spatial Manager.

You must restart the server when you change the volatility setting on any existing named table or when creating a new named table based on a database table that was previously set to false (that is, volatility was disabled).

> **Note:** Do not use the updateNamedResource operation in the Named Resource Service to change this value or manually edit the named table definition that you accessed via WebDAV in a text editor.

Running Spectrum™ Technology Platform as a Linux Service

This tutorial will show you the steps you need to follow to run Spectrum™ Technology Platform as a Linux service.

How to Run Spectrum™ Technology Platform as a Linux Service

These instructions describe how to run the Spectrum™ Technology Platform as a Linux service.

1. Modify the provided `pbspectrum` script which is located here: **PBSpectrum Script** on page 13.

    a) Modify the `chkconfig` parameter at line# 5. By Default this parameter is: `# chkconfig: 35 90 10`

    First value(35) is runlevel. Use 'man init' for more information.

    Second value(90) is start priority

    Third value(10) is stop priority.

    Start and stop priority should be set according to the dependent services. For example, if Oracle Server is running on the same machine and is used by Spectrum™ Technology Platform then the Spectrum™ Technology Platform starting priority should be less than the Oracle Service and stopping priority should be higher than the Oracle service. Use 'man chkconfig' for more information.

    b) Modify SPECTRUM_ROOT variable at line #11 with your Spectrum™ Technology Platform installation directory.

    c) If you are using SUSE Linux, you must change the default preferred user from `su` to `runuser`.

2. Copy the modified `pbspectrum` script to either `/etc/rc.d/init.d` for RedHat Linux or `/etc/init.d` for Suse Linux.

3. Change the mode of the `pbspectrum` script to executable. `/etc/rc.d/init.d` for RedHat Linux or `/etc/init.d` for Suse Linux.

    `cd /etc/init.d` or `cd /etc/rc.d/init.d` depending on your Linux version.

    run `chmod +x pbspectrum`

4. Run `chkconfig --add pbspectrum`

5. Verify the script is working by restarting the machine. Use `shutdown -r now` to reboot from shell.

Once completed, you may also use the following:

- `service pbspectrum start` to start Spatial Server
- `service pbspectrum stop` to stop Spatial Server
- `service pbspectrum restart` to restart Spatial Server

> **Note:** The provided script runs the command 'ulimit -n 8192' which is required to increase the number of open files in Linux.

PBSpectrum Script

The following script is used as the basis for this procedure: **How to Run Spectrum™ Technology Platform as a Linux Service** on page 12.

```bash
   #! /bin/bash
#
#  pbspectrum Bring up/down PB Spectrum platform
#
# chkconfig: 35 90 10
# description: Starts and stops the spectrum
#
# /etc/rc.d/init.d/pbspectrum
# See how we were called.

SPECTRUM_ROOT=/root/PBSpectrum

start() {
        su - spectrum -c ". $SPECTRUM_ROOT/server/bin/setup;
  ulimit -n 8192;
  $SPECTRUM_ROOT/server/bin/server.start"
        RETVAL=$?
        return $RETVAL
}

stop() {
        su - spectrum -c ". $SPECTRUM_ROOT/server/bin/setup;
        $SPECTRUM_ROOT/server/bin/server.stop"
        RETVAL=$?
        return $RETVAL
}

# See how we were called.
case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  restart)
        stop
        start
        ;;
  *)
        echo $"Usage: pbspectrum {start|stop|restart}"
        exit 1
esac
```

```
exit $RETVAL
```

Configuring a Linux Machine for MRR

To use MRR (Multi Resolution Raster) files on Spectrum Spatial in a Linux environment, GCC and LIBC must be upgraded to the proper versions.

To configure a Linux machine for MRR:

1. Install the UUID package, which installs LIBC v.2.17.

   For example, to install UUID on Cent OS:

   - `wget http://ftp.riken.jp/Linux/centos/6/os/x86_64/Packages/libuuid-2.17.2-12.18.el6.x86_64.rpm`
   - `sudo yum -y install libuuid-2.17.2-12.18.el6.x86_64.rpm`
   - `sudo yum -y install libuuid-devel`

2. Install devtoolset-3, which installs GCC v.4.9. For instructions, see **https://www.softwarecollections.org/en/scls/rhscl/devtoolset-3/**.

3. Verify that GCC v.4.9 and LIBC v.2.17 (or higher) are installed.

4. Ensure that all the dependencies were resolved in the above steps. If any dependency is unresolved, install it and then repeat Step 2.

   For example, the following are some of the required dependencies for an OEL 6.5 machine:

   - `wget https://www.softwarecollections.org/en/scls/mizdebsk/maven30-rhel-6/epel-6-x86_64/download/mizdebsk-maven30-rhel-6-epel-6-x86_64.noarch.rpm`
   - `sudo yum -y install mizdebsk-maven30-rhel-6-epel-6-x86_64-1-2.noarch.rpm`

   - `wget https://www.softwarecollections.org/en/scls/rhscl/maven30/epel-6-x86_64/download/rhscl-maven30-epel-6-x86_64.noarch.rpm`
   - `sudo yum -y install rhscl-maven30-epel-6-x86_64-1-2.noarch.rpm`
   - `sudo yum -y install maven30`

   - `wget https://www.softwarecollections.org/en/scls/mbooth/eclipse-luna/fedora-20-x86_64/download/mbooth-eclipse-luna-fedora-20-x86_64.noarch.rpm`
   - `sudo yum -y install mbooth-eclipse-luna-fedora-20-x86_64-1-2.noarch.rpm`
   - `sudo yum -y install --skip-broken eclipse-luna`

Disabling Default HTTP Cache Control Headers

By default, Spectrum™ Technology Platform web services use the following HTTP headers for caching:

```
Cache-Control: no-cache,no-store,no-transform,must-revalidate
Expires: Wed, 07 Jan 2015 15:38:03 GMT //48 hours in the past
Pragma: no-cache
```

These HTTP headers are not appropriate for the Map Tiling Service; however, you can disable these default HTTP headers and instead set the HTTP cache behavior in the headers that are defined in the individual web services.

> **Note:** If you are applying this change to a cluster you must repeat the following procedure on each node in the cluster.

To disable the default HTTP cache control headers:

1. Stop the Spectrum™ Technology Platform server.
2. Open the following file in a text editor:

   SpectrumFolder\server\app\conf\spectrum-advanced.properties

3. Change the following property from true to false:

   ```
   spectrum.cache.control.headers.enable=false
   ```

4. Save and close the properties file.
5. Start the Spectrum™ Technology Platform server.

3 - Managing Security

The Location Intelligence Module uses the same role-based security model that is used for the Spectrum™ Technology Platform. Because security is handled at the platform level, the Management Console can be used to manage all Location Intelligence Module security activities.

In this section

Security for the Spectrum™ Technology Platform

The topics in this section cover the security model and procedures at the platform level that pertain to all modules. See **Security for the Location Intelligence Module** on page 49 for additional security information that is specific to that module.

Security Model

Spectrum™ Technology Platform uses a role-based security model to control access to the system. The following diagram illustrates the key concepts in the Spectrum™ Technology Platform security model:
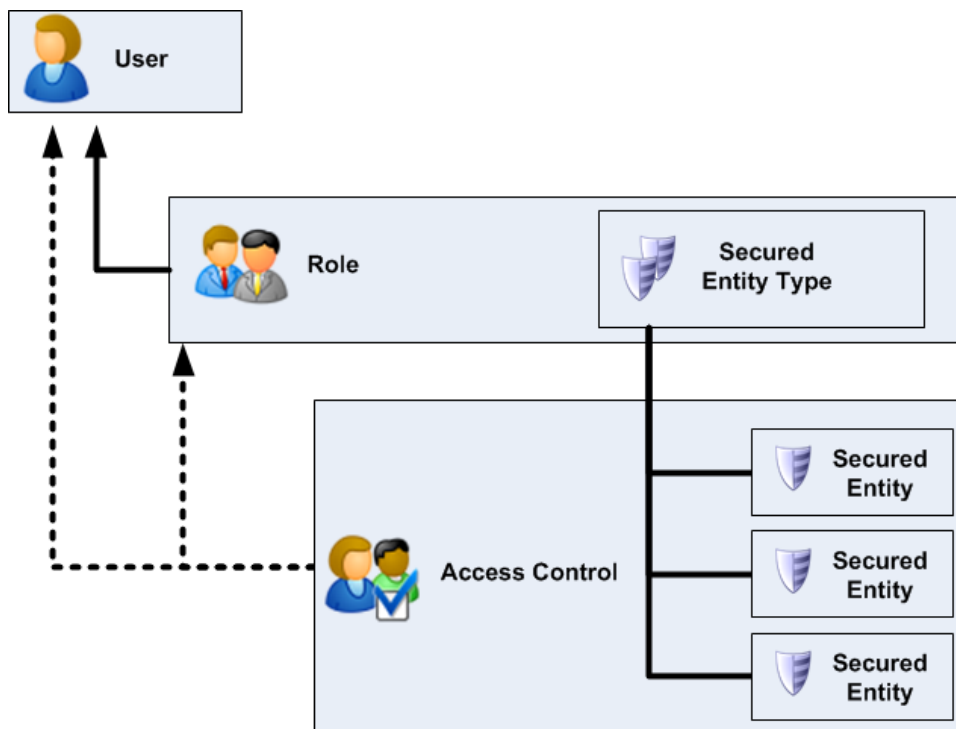
A *user* is an account assigned to an individual person which the person uses to authenticate to Spectrum™ Technology Platform, either to one of the client tools such as Enterprise Designer or Management Console, or when calling a service through web services or the API.

A user has one or more roles assigned to it. A *role* is a collection of permissions that grant or deny access to different parts of the system. Roles typically reflect the kinds of interactions that a particular type of user has with the system. For example, you may have one role for dataflow designers which grants access to create and modify dataflows, and another role for people who only need to process data through existing dataflows.

A role grants permissions to secured entity types. A *secured entity type* is a category of items to which you want to grant or deny access. For example, there is a secured entity type called "Dataflows" which controls the default permissions for all dataflows on the system.

If you need to fine-tune access you can optionally override the settings in the role or user by configuring access control. Access control settings work in conjunction with roles to define the permissions for a user. Roles define the permissions for categories of entities, such as all dataflows or all database resources, and access control settings define the permissions for specific entities, called *secured entities*. Examples of secured entities include specific jobs or specific database connections. Defining access control settings is optional. If you do not define access control settings, the permissions defined in the role will control the user's permissions.

Access control settings work in conjunction with roles to define the permissions for a user. Roles define the permissions for categories of entities, such as all dataflows or all database resources, and access control settings define the permissions for specific entities, called *secured entities*. Examples of secured entities include specific jobs or specific database connections. For example, you may have a role that has granted the Modify permission to the secured entity type "Dataflows", but you may want to prevent users from modifying one specific dataflow. You could accomplish this by using access control to remove the Modify permission for the specific dataflow you do not want modified. You can specify access control settings for users and roles. Access control settings for a user override that specific user's permissions as granted by the user's roles. Access control settings for roles apply to all users who have that role.

Users

Spectrum™ Technology Platform user accounts control the types of actions users can perform on the system. User accounts are required to:

- Use tools like Management Console, Enterprise Designer, Metadata Insights, and command-line tools
- Run jobs on a schedule
- Run jobs from the command line
- Access services through web services or the API

There is an administrative account called **admin** that comes with the system. This account has full access. The initial password is "admin".

Important: You should change the admin password immediately after installing Spectrum™ Technology Platform to prevent unauthorized administrative access to your system.

You can create as many user accounts as you need.

Adding a User

This procedure describes how to create a Spectrum™ Technology Platform user account and assign a role to the account.

1. Open Management Console.
2. Go to **System** > **Security**.
3. Click the Add button ✚ .
4. Leave the **Enabled** switch set to **On** if you want this user account to be available for use.
5. Enter the user name in the **User name** field.

> **Note:** User names can only contain ASCII characters. User names are case sensitive.

6. Enter the user's email address in the **Email address** field. The email address is used by some modules to send notifications to users.
7. Enter a description of the user in the **Description** field.
8. Enter and confirm the user's password.
9. Select the roles you want to give to this user.

    You may create your own roles or use one of the default roles. The default roles are:

admin	This role has full access to all parts of the system.
designer	This role is for users that create dataflows and process flows in Enterprise Designer. It provides the ability to design and run dataflows.
integrator	This role is for users who need to process data through Spectrum™ Technology Platform but do not need to create or modify dataflows. It allows the user to access services through web services and the API, and to run jobs.
user	This is the default role. It provides no access to the system. Users who have this role will only gain access to the system if you grant permission through secured entity overrides.

    For information about creating roles, see **Creating a Role** on page 23.

10. Click **Save**.

Changing a Password

This procedure describes how to change a user's password.

1. Open the Management Console.
2. Go to **System** > **Security**.
3. Select a user then click the Edit button ✎.
4. Click **Change password**.
5. Enter the new password and enter it a second time to confirm it.
6. Click **Save**.

Setting a Minimum Password Length

The minimum password length is enforced when creating or changing a password. Existing passwords that are shorter than the minimum length will continue to be valid.

1. Open a web browser and go to `http://server:port/jmx-console`

   Where:

   server is the IP address or hostname of your Spectrum™ Technology Platform server.

   port is the HTTP port used by Spectrum™ Technology Platform. The default is 8080.

2. Log in using the admin account.
3. Under " Domain: com.pb.spectrum.platform.config", click **com.pb.spectrum.platform.config:manager=AccountConfigurationManager**.
4. In the **updatePasswordPolicy** operation, set the **enableAdvanceControl** option to **True**.
5. In the **minLength** field, enter the minimum password length.
6. Click **Invoke**.
7. Click **Return to MBean View** to go back to the Account Configuration Manager screen.

Changing Your Email Address

The email address associated with your user account is used by some modules to send you notifications. To change your email address, follow these steps.

1. Log in to Management Console.
2. Click the user menu in the top right corner.
3. Select **Profile**.
4. In the **Email** field, enter your new email address.
5. Click **Save**.

Disabling a User Account

You can disable a user account so that it cannot be used to gain access to Spectrum™ Technology Platform. Any jobs that run on a schedule using a disabled user account will not run.

> **Note:** The user account "admin" cannot be disabled.

1. Open the Management Console.
2. Go to **System** > **Security**.
3. Check the box next to the user you want to modify then click the Edit button ✎.
4. Switch the **Enabled** switch to **Off**.
5. Click **Save**.

The user account is now disabled and cannot be used to gain access to Spectrum™ Technology Platform.

Deleting a User

This procedure describes how to permanently delete a Spectrum™ Technology Platform user account.

Tip: User accounts can also be disabled, which prevents the account from being used to access the system without deleting the account.

1. Open the Management Console.
2. Go to **System** > **Security**.
3. Check the box next to the user you want to delete then click the Delete button 🗑.

> **Note:** The user account "admin" cannot be deleted.

User Account Locking

As a security precaution, user accounts are disabled after five unsuccessful authentication attempts in row. This includes unsuccessful authentication attempts to Enterprise Designer, Management Console, web services, and the Client API.

As an administrator, you can re-enable a user account by logging into Management Console, editing the user, and switching the **Enabled** switch to **On**. User accounts can also be re-enabled using the Administration Utility. Users do not have the ability to unlock their own accounts.

> **Note:** If you are using LDAP or Active Directory for authentication, the account locking rules of these services will apply. Your LDAP or Active Directory rules may allow more or fewer unsuccessful login attempts than Spectrum™ Technology Platform.

Unlocking the admin Account

User accounts are locked after several unsuccessful login attempts. Most user accounts can be unlocked through Management Console, but the admin account cannot. Instead, you must run a script on the server to unlock the admin account.

1. Log in to the server running Spectrum™ Technology Platform.

   If you are running Spectrum™ Technology Platform in a cluster, log in to any of the nodes. You only need to run the unlock script on one of the nodes.

2. Open a command prompt and go to the *Spectrum Folder*\server\bin folder.
3. (Unix and Linux only) Run the following command:

   ```
   . ./setup
   ```

4. Run the enableadmin script by typing the following command:

On Windows:

```
enableadmin.bat -h HostAndPort -p AdminPassword [-s]
```

On Unix and Linux:

```
./enableadmin.sh -h HostAndPort -p AdminPassword [-s]
```

Where:

HostAndPort	The hostname and HTTP port used by Spectrum™ Technology Platform. For **example**, `spectrumserver:8080`.
AdminPassword	The password for the admin account. If you do not know the admin account password and the admin account is locked, contact Pitney Bowes Technical Support.
-s	Specify `-s` if Spectrum™ Technology Platform is configured to use HTTPS.

Automatic Logout Due to Inactivity

Users of Enterprise Designer and web clients such as Management Console, the Relationship Analysis Client, Business Steward Portal, and others are automatically logged out after 30 minutes of inactivity.

Roles

A *role* is a collection of permissions that grant or deny access to different parts of the system. Roles typically reflect the kinds of interactions that a particular type of user has with the system. For example, you may have one role for dataflow designers which grants access to create and modify dataflows, and another role for people who only need to process data through existing dataflows.

Spectrum™ Technology Platform comes with these roles already defined:

admin	This role has full access to all parts of the system.
designer	This role is for users that create dataflows and process flows in Enterprise Designer. It provides the ability to design and run dataflows.
integrator	This role is for users who need to process data through Spectrum™ Technology Platform but do not need to create or modify dataflows. It allows the user to access services through web services and the API, and to run jobs.
user	This is the default role. It provides no access to the system. Users who have this role will only gain access to the system if you grant permission through secured entity overrides.

> **Note:** See **Security for the Location Intelligence Module** on page 49 for information about the predefined roles for the Location Intelligence Module.

To view the permissions granted to each of these roles, open Management Console, go to **Security** and click **Roles**. Then select the role you want to view and click **View**.

Tip: You cannot modify the predefined roles. However, you can create new roles using the predefined roles as a starting point.

Creating a Role

A role is a collection of permissions that you assign to a user. If the predefined roles that come with Spectrum™ Technology Platform do not fit your organization's needs, you can create your own roles.

1. Open Management Console.
2. Go to **System** > **Security**.
3. Click **Roles**.
4. Click the Add button ✚ .

   Tip: If you want to create a role that's similar to an existing role, you can make a copy of the existing role by checking the box next to the role you want to copy then clicking the Copy button [⧉]. Then, edit the new role and continue with the following steps.

5. In the **Role name** field, enter the name you want to give to this role. The name can be anything you choose.
6. Optional: Since the list of secured entity types can be long, you may want to display only a certain group of secured entity types. This can be useful if you want to apply the same permissions to all entities in a group. For example, if you want to remove the Modify permission from all database resources, you could filter to show just the Database Resources group. To display and modify only one group:
   a) Check the **Enable group filtering** box.
   b) Click the funnel icon in the header of the **Group** column and select the group you want to display.
   c) Check or clear the box in the column header of the permission you want to apply.
   d) To return to the full list of secured entity types, click the filter icon and select **(All)** then clear the **Enable group filtering** box.
7. Select the permissions you want to grant for each entity type. The permissions are:

   View   Allows the user to view entities contained by the entity type. For example, if you allow the View permission for the JDBC Connection entity type, users with this role would be able to view database connections in Management Console.

   Modify   Allows the user to modify entities contained by the entity type. For example, if you allow the Modify permission for the JDBC Connection entity type, users with this role would be able to modify database connections in Management Console.

   Create   Allows the user to create entities that fall into this entity type's category. For example, if you allow the Create permission for the JDBC Connection entity type,

users with this role would be able to create new database connections in Management Console.

Delete   Allows the user to delete entities contained by the entity type. For example, if you allow the Delete permission for the JDBC Connection entity type, users with this role would be able to delete database connections in Management Console.

Execute   Allows the user to initiate processing of jobs, services, and process flows. For example, if you allow the Execute permission for the Job entity type, users with this role would be able to run batch jobs. If you allow the Execute permission for the Service entity type, users with this role would be able to access services running on Spectrum™ Technology Platform through the API or web services.

8. Click **Save**.

The role is now available to be assigned to a user.

Deleting a Role

A role may be deleted if it is no longer assigned to any users.

> **Note:** The following roles cannot be deleted: admin, user, designer, and integrator.

1. Open Management Console.
2. Go to **System** > **Security**.
3. On the **Users** tab, make sure the role you want to delete is not assigned to any users. You cannot delete a role if it assigned to a user.
4. Click **Roles**.
5. Check the box next to the role you want to delete then click the Delete button 🗑.

Disabling Role-Based Security

Role-based security is enabled by default. This means that the security restrictions assigned to users through roles are enforced. If you want to disable role-based security, the security restrictions assigned to users will not be enforced and all users will be able to access all parts of the system. Note that a valid user account is always required to access services even if you disable role-based security.

This procedure describes how to disable role-based security.

Warning: If you follow this procedure all users will have full access to your Spectrum™ Technology Platform system.

1. Open the Management Console.
2. Go to **System** > **Security**.
3. Switch the **Limit access by role** switch to **Off**.

Secured Entity Types - Platform

An entity type is a category of items to which you want to grant or deny access. For example, there is an entity type called "Dataflows" which controls permissions for all dataflows on the system. Platform entity types apply to all Spectrum™ Technology Platform installations, as compared to module-specific entity types that apply only if you have installed particular modules. The platform-level entity types are:

Audit Log
Controls access to the **System** > **Logs** > **Audit Log** area in Management Console.

Dataflows
Controls access to all dataflow types (jobs, services, and subflows) in Enterprise Designer.

> **Note:** If a user does not have the Edit permission, the user will only see the exposed version and the last saved version in the **Versions** pane in Enterprise Designer.

Dataflows - Expose
Controls the ability to make dataflows available for execution.

> **Note:** In order to expose the latest saved version of the dataflow (the version that's always at the top of the **Versions** pane in Enterprise Designer) the user must have the Edit permission for the **Dataflows** secured entity type in addition to the Edit permission for the **Dataflows - Expose** secured entity type. This is because the latest saved version must first be saved as a version before it can be exposed, which requires the Edit permission for the dataflow.

Flow Defaults - Data Type Conversion
Controls access to the **Flows** > **Defaults** > **Data Type Conversions** area in Management Console. All users have View access to data type conversion options. You cannot remove View access.

Flow Defaults - Malformed Records
Controls access to the **Flows** > **Defaults** > **Malformed Records** area in Management Console. All users have View access to malformed record options. You cannot remove View access.

Flow Defaults - Reports
Controls access to the **Flows** > **Defaults** > **Reports** area in Management Console. All users have View access to report options. You cannot remove View access.

Flow Defaults - Sort Performance
Controls access to the **Flows** > **Defaults** > **Sort Performance** area in Management Console. All users have View access to sort performance options. You cannot remove View access.

Flow History - Jobs
Controls access to view job execution history in Enterprise Designer and Management Console.

Flow History - Process Flows
Controls access to process flow execution history in Management Console and Enterprise Designer.

Flow History - Transactions
Controls access to the **Flows** > **History** > **Transactions** are in Management Console.

Flow Scheduling	Controls access to the **Flow** > **Schedules** area in Management Console.
Jobs	Controls the ability to execute jobs in Enterprise Designer, Management Console, job executor, and the Administration Utility.
Notification - License Expiration	Controls access to configure license expiration notification emails in Management Console.
Notification - SMTP Settings	Controls access to the **System** > **Mail Server** area in Management Console.
Process Flows	Controls access to process flows in Enterprise Designer.
	Note: If a user does not have the Edit permission, the user will only see the exposed version and the last saved version in the **Versions** pane in Enterprise Designer.
Process Flows - Expose	Controls the ability in Enterprise Designer to make process flows available for execution.
	Note: In order to expose the latest saved version of the process flow (the version that's always at the top of the **Versions** pane in Enterprise Designer) the user must have the Edit permission for the **Process Flows** secured entity type in addition to the Edit permission for the **Process Flows - Expose** secured entity type. This is because the latest saved version must first be saved as a version before it can be exposed, which requires the Edit permission for the dataflow.
Resources - Database Connections	Controls the ability to configure database connections in Management Console.
Resources - External Web Services	Controls access to managing external web services in Management Console.
Resources - File Server Connections	Controls the ability to configure file servers in Management Console.
Resources - JDBC Drivers	Controls the ability to configure JDBC drivers in Management Console.
Resources - Remote Server	Controls access to the **Resources** > **Remote Servers** area in Management Console.
Security - Access Control	Controls access to access control settings in the **System** > **Security** > **Access Control** area in Management Console.
Security - Access Token	Controls the ability to view users' tokens and delete tokens. A token facilitates authentication between a client and the server. Read permission allows you to see a list of the active tokens, each of which represent an active session. The Delete permission allows you to delete users' tokens, which ends their session.
Security - Directory Access	Controls the ability to enable or disable restrictions on server directory resources using the **System** > **Security** > **Directory Access** area in Management Console.

Security - Directory paths	Controls the ability to configure server directory resources the **System** > **Security** > **Directory Access** area in Management Console.
Security - Options	Controls access to the ability to turn security on and off in the **System** > **Security** > **Roles** area in Management Console.
Security - Roles	Controls access to role configuration in the **System** > **Security** > **Roles** area in Management Console.
Security - Directory paths	Controls the ability to configure server directory resources the **System** > **Security** > **Directory Access** area in Management Console.
Security - Users	Controls access for managing user accounts in the **System** > **Security** > **Users** area in Management Console.
Services	Controls the ability to execute services through the API and web services.
Stages	Controls whether exposed subflows are available as a stage in dataflows in Enterprise Designer.
System - Licensing	Controls access to the license information displayed in the **System** > **Licensing and Expiration** area in Management Console.
System - Version Information	Controls access to the **System** > **Version** area in Management Console.
System Log	Controls access to the system log in Management Console.

Secured Entity Types - Location Intelligence Module

An entity type is a category of items to which you want to grant or deny access. The Location Intelligence Module has the following module-specific entity types:

Named Resources	Controls permissions to all named resources in the Location Intelligence Module. Users of Location Intelligence Module services must have at least read permissions for the resources they use as well as for any dependent resources. When a named resource is created (using any tool, including Spatial Manager, the Administration Utility, the Named Resource Service, and WebDAV), a new LocationIntelligence.Named Resource secured entity is automatically created for the named resource.
Dataset.DML	Controls permissions to datasets used in the Location Intelligence Module that are associated with named tables. When a named table is created or uploaded (using any tool, including Spatial Manager, the Administration Utility, the Named Resource Service, and WebDAV), a new LocationIntelligence.Dataset secured entity is automatically created for the associated dataset of that named table. A user must have View permissions on a named table *and* Create/Modify/Delete permissions on the dataset in order to perform DML operations on writable (JDBC-based) tables. DML operations include insert, update, and delete operations performed using the Write Spatial Data stage or the Feature Service.

Access Control

Access control settings work in conjunction with roles to define the permissions for a user. Roles define the permissions for categories of entities, such as all dataflows or all database resources, and access control settings define the permissions for specific entities, called *secured entities*. Examples of secured entities include specific jobs or specific database connections. For example, you may have a role that has granted the Modify permission to the secured entity type "Dataflows", but you may want to prevent users from modifying one specific dataflow. You could accomplish this by using access control to remove the Modify permission for the specific dataflow you do not want modified. You can specify access control settings for users and roles. Access control settings for a user override that specific user's permissions as granted by the user's roles. Access control settings for roles apply to all users who have that role.

Configuring Access Control

Access control settings work in conjunction with roles to define the permissions for a user. Roles define the permissions for categories of entities, such as all dataflows or all database resources, and access control settings define the permissions for specific entities, such as specific jobs or specific database connections.

In order to configure access controls you must have View and Modify permissions to these secured entity types:

• Security - Access Control
• Security - Roles
• Security - Users

To configure access control:

1. In Management Console, go to **System** > **Security**.
2. Click the **Access Control** tab.
3. Click the Add button ✚ .
4. Do one of the following:

   • If you want to specify access controls for a role, click **Role**. The access control permissions you specify will affect all users who have the role you choose.
   • If you want to specify access controls for a single user, click **User**. The access control permissions you specify will only affect the user you choose.

5. Select the role or user for which you want to define access controls.
6. Click the Add button ✚ .
7. Select the secured entity type that contains the secured entity you want. For example, if you want to configure access control for a dataflow, choose Platform.Dataflows.

8. Choose the secured entity you want to configure access controls for, then click the **>>** button to add it to the **Selected Entities** list.

9. Click **Add**.

   The secured entities you chose are displayed. The check boxes indicate the permissions in effect for the selected role or user.

10. Specify the permissions that you want to grant for each secured entity. Each secured entity can have one of the following permissions:

⊟	The permission is inherited from the role.
⊟	The permission is inherited from the role and cannot be overridden.
☑	The permission is granted, overriding the permission specified in the user or role.
☐	The permission is denied, overriding the permission specified in the user or role.

Access Control Example

The following shows access control settings for the role RetentionDepartmentDesigner.

In this example, the Platform.Dataflow secured entity type is set to allow the View and Modify permissions but not the Delete permission. So by default, any user that has the RetentionDepartmentDesigner role would have these permissions for all dataflows. However, you want to prevent users with this role from modifying the ExampleJob1 dataflow only. So, you clear the checkbox in the Modify column for

ExampleJob1. Now users with this role will not be able to modify this dataflow but will still be able to modify other dataflows.

Deleting Access Control Settings

When you delete access control settings for a user or role, the permission overrides defined by the access control settings are removed from the user or role. For users, this means that the permissions granted by the user's role will take effect without any overrides. For roles, this means that the permissions defined in the role itself will take effect without overrides.

1. Open Management Console.
2. Go to **System** > **Security**.
3. Click **Access Control**.
4. Check the box next to the user or role for whom you want to remove access control then click the Delete button 🗑 .

Limiting Server Directory Access

Users can browse the Spectrum™ Technology Platform server's folders when performing tasks that require them to select a file. For example, users can browse the server when selecting an input or output file in a source or sink stage in Enterprise Designer. As an administrator, you may want to restrict access so that sensitive portions of the server cannot be browsed or modified.

One way to prevent access to the server's file system by making sure that users do not have the Platform security permission **Security - Directory Paths**. This prevents access to all folders on the server. You can also prevent access to some folders on the server while allowing access to others. When you grant limited access, the folders you allow access to appear as the top-level folders in users' file browse windows. For example, if you allow users to only access a folder on the server named WestRegionCustomers, when users browse the server they would only see that folder, as shown here:
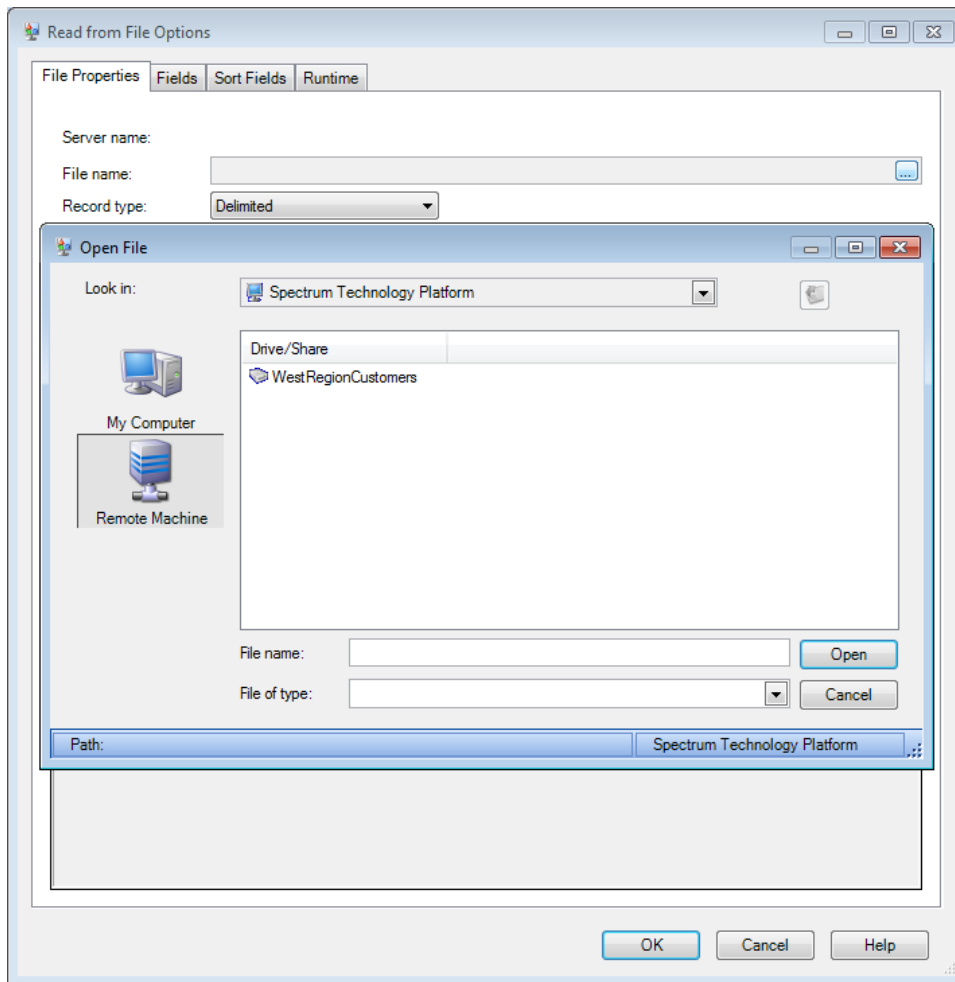
Important: There are two situations where users can view the server's entire file system even if you have granted only limited access:

• When browsing for a database file while creating a Spectrum database in Management Console
• When browsing for a JDBC driver file while creating a driver in Management Console

To prevent users from browsing the server's entire file system, use roles to limit the user's access to Spectrum databases and JDBC drivers.

To provide access to some folders on the server while restricting access to others, follow this procedure.

1. Open Management Console.
2. Go to **System** > **Security**.
3. Click **Directory Access**.
4. Set the **Limit access to server directories** switch to **On**.
5. Click the Add button ✚.
6. In the **Name** field, give a meaningful name for the folder to which you are granting access.

The name you provide here appears as the root name of the directory to users when browsing the server. In the example shown at the beginning of this topic, the name given to the acessible directory is WestRegionCustomers.

7. In the **Path** field, specify the folder to which you want to grant access. Users will be able to access all file and subfolders contained in the folder you specify.

8. Click **Save**.

9. If you want to grant access to additional folders, repeat the previous steps as needed.

Users now have access only to the folders you have specified. Note that users must have the Platform security permission **Security - Directory Paths** in order to access server directories.

> **Note:** If there are any dataflows that had previously accessed files that are no longer available because of file browsing restrictions, those dataflows will fail.

Configuring HTTPS Communication

By default the Spectrum™ Technology Platform server uses HTTP for communication with Enterprise Designer, browser applications such as Management Console and Metadata Insights, as well as for handling web service requests and API calls, and for remote server communication. You can configure Spectrum™ Technology Platform to use HTTPS if you want to secure these network communications.

> **Note:** Spectrum™ Technology Platform uses TLS 1.2 to encrypt communication. Applications that access Spectrum™ Technology Platform web services or the API must support TLS 1.2 in order to connect over HTTPS.

This procedure describes how to enable HTTPS communication on a single-server installation of Spectrum™ Technology Platform. If you want to use HTTPS and you are running Spectrum™ Technology Platform in a cluster, do not follow this procedure. Instead, configure the load balancer to use HTTPS for communication with clients. Communication between the load balancer and the Spectrum™ Technology Platform nodes, and between the nodes themselves, will be unencrypted because Spectrum™ Technology Platform clustering does not support HTTPS. The load balancer and the Spectrum™ Technology Platform servers in the cluster must be behind a firewall to provide a secure environment.

To configure HTTPS communication for a single-server installation of Spectrum™ Technology Platform:

1. Stop the Spectrum™ Technology Platform server.

   - To stop the server on Windows, right-click the Spectrum™ Technology Platform icon in the Windows system tray and select **Stop Spectrum**™. Alternatively, you can use the Windows Services control panel and stop the Pitney Bowes Spectrum™ Technology Platform service.
   - To stop the server on Unix or Linux, source the *SpectrumLocation*`/server/bin/setup` script then execute the *SpectrumLocation*`/server/bin/server.stop` script.

2. Create a certificate signed by a trusted Certificate Authority (CA).

> **Note:** The certificate must meet the requirements for encryption and length for the version of Java used by Spectrum™ Technology Platform. To find out the version of Java, open Management Console and go to **System** > **Version**. For more information, see **java.com/en/jre-jdk-cryptoroadmap.html**.

3. Load the certificate into a JSSE keystore. For more information, see **www.eclipse.org/jetty/documentation/current/configuring-ssl.html#loading-keys-and-certificates**.

4. Create an XML file named `spectrum-override-container-ssl.xml` containing the following:

```
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:util="http://www.springframework.org/schema/util"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
      http://www.springframework.org/schema/beans/spring-beans-3.0.xsd

       http://www.springframework.org/schema/util
      http://www.springframework.org/schema/util/spring-util-3.0.xsd">


    <bean id="defaultWebServerConnector"
class="org.eclipse.jetty.server.ServerConnector">
        <constructor-arg ref="webServer"/>
        <constructor-arg>
          <bean class="org.eclipse.jetty.util.ssl.SslContextFactory">

              <property name="keyStorePath"
value="/SpectrumKeystore"/>
              <property name="keyManagerPassword" value="password"/>

              <property name="keyStorePassword" value="password"/>
          </bean>
        </constructor-arg>
        <property name="host" value="${spectrum.bind.address}"/>
        <property name="port" value="${spectrum.http.port}"/>
        <property name="idleTimeout" value="-1"/>
    </bean>
</beans>
```

5. Modify these lines as needed to reflect your environment:

`<property name="keyStorePath" value="/SpectrumKeystore"/>`	Modify the value to be the full path to the Java keystore.
`<property name="keyManagerpassword" value="password"/>`	Modify the value to be the password to the keystore.
`<property name="keyStorePassword" value="password"/>`	Modify the value to be the password to the key within the keystore.

6. Save the `spectrum-override-container-ssl.xml` file to *SpectrumLocation*`/server/app/conf/spring`.

7. Using a text editor, open the file `spectrum-container.properties` located in *SpectrumLocation*`/server/app/conf`. Uncomment and set these properties:

   ```
   spectrum.http.port=port
   spectrum.runtime.port=port
   spectrum.runtime.hostname=dnsname
   ```

   Where *port* is the network port to use for communication with the clients (for example 8443) and *dnsname* is the hostname of the Spectrum™ Technology Platform server. The port you specify must be the same for both `spectrum.http.port` and `spectrum.runtime.port`.

8. If you are configuring HTTPS communication for the Location Intelligence Module and Spectrum Spatial services, you must perform additional configuration prior to restarting the Spectrum™ Technology Platform server:

   a) Modify the `java.properties` file (*SpectrumLocation*`\server\modules\spatial`) by changing all hostnames and ports to be exactly the same as the ones used for the Spectrum™ Technology Platform server. The hostname must match the DNS name of the server and the CN in the certificate. Set property `repository.useSecureConnection` to true. For example,

   ```
   repository.host=www.spectrum.com
   repository.port=8443
   repository.useSecureConnection=true
   ```

   b) In Spatial Manager, change the URLs in these service configurations to use HTTPS:

   • Mapping (only necessary when accessing the Mapping Service via SOAP and when the ReturnImage parameter for a RenderMap request is False)
   • WFS
   • WMS
   • WMTS

   For instructions, see the "Spatial Manager Guide" in the Utilities section of the *Spectrum Spatial Guide*.

9. Start the Spectrum™ Technology Platform server.

   • To start the server on Windows, right-click the Spectrum™ Technology Platform icon in the Windows system tray and select **Start Spectrum™**. Alternatively, you can use the Windows Services control panel to start the Pitney Bowes Spectrum™ Technology Platform service.
   • To start the server on Unix or Linux, execute the *SpectrumLocation*`/server/bin/server.start` script.

Web Service Authentication

Spectrum™ Technology Platform web services require requesters to authenticate with valid user credentials. There are two methods for authenticating: Basic authentication and token-based authentication.

Basic Authentication

With Basic authentication, the user ID and password are passed to Spectrum™ Technology Platform in the HTTP header of each request to the web service. Basic authentication is allowed by default, but your administrator may choose to disable Basic authentication. If Basic authentication is disabled you must use token-based authentication to access web services.

Token-Based Authentication

With token-based authentication, the requester obtains a token from the Spectrum™ Technology Platform server, then uses the token when sending a request to the web service. Instead of sending user credentials in each request, the token is sent to the server and the server determines if the token is valid.

The following diagram illustrates the process:
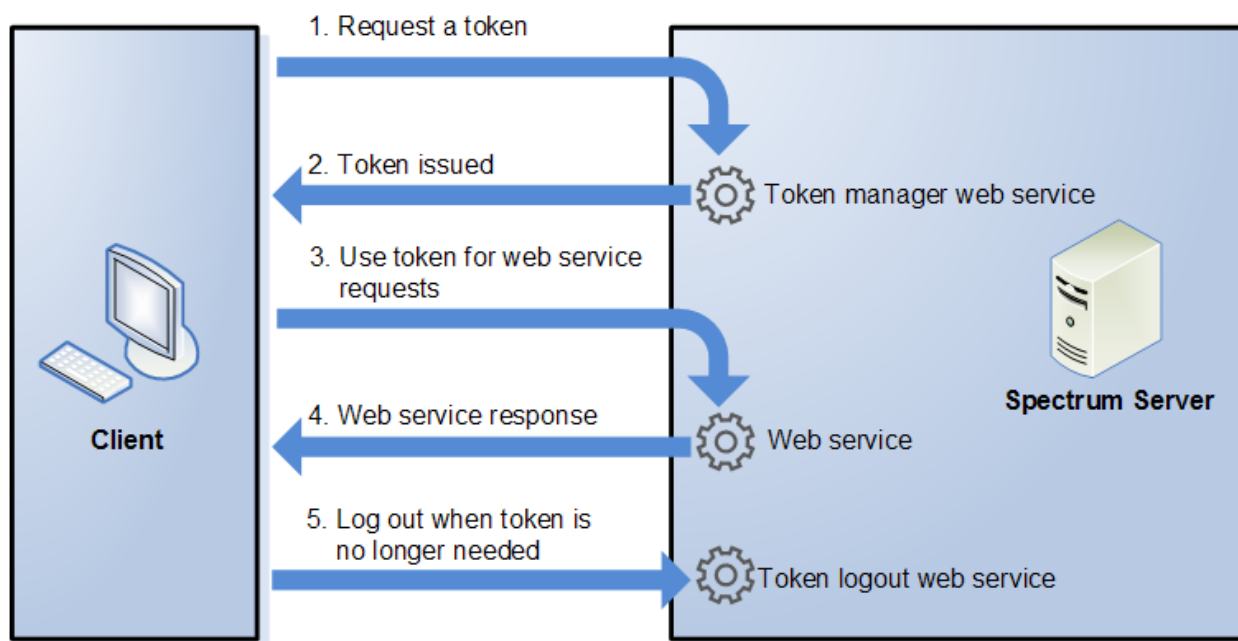
1. Obtain a token from the Spectrum™ Technology Platform server by sending a request to the token manager service.
2. The token manager service issues a token. If you requested a session token it also issues a session ID.

3. Send a request to the desired web service with the token in the HTTP header. For session tokens, include the session ID in the HTTP header.

4. The web service issues a response. You can use the token to make additional web service requests to either the same web service or any other web service on the Spectrum™ Technology Platform server. There is no limit to the number of web service requests you can make with a token, but if the token has an expiration limit (also known as a time-to-live) it will become invalid after the time-to-live has elapsed. If the token is a session token, it will become invalid after 30 minutes of inactivity.

5. When the token is no longer needed you should log out by sending a request to the token logout web service. This will remove the token from the list of valid tokens on the Spectrum™ Technology Platform server.

Disabling Basic Authentication for Web Services

Spectrum™ Technology Platform supports two types of authentication for web service requests: Basic authentication and token authentication. By default, both methods are enabled. If you want to require web service requests to use token authentication instead of Basic authentication, you can disable Basic authentication by following these steps.

> **Note:** Be aware that disabling Basic authentication will cause existing clients to fail. For the Location Intelligence Module, WMS, WMTS, and WFS clients will either be expecting Basic authentication or no authentication. Leaving only token-based authentication will likely cause those clients to fail.

1. Stop the Spectrum™ Technology Platform server.
2. Open this file in a text editor:

   SpectrumLocation/server/app/conf/spectrum-container.properties

3. Set this property to false:

   ```
   spectrum.security.authentication.webservice.basicauth.enabled=false
   ```

4. Start the server.

Disabling Authentication for Web Services

All services and access to resources used by Spectrum™ Technology Platform are configured, by default, with authentication turned on.

Service-level authentication can be disabled for all SOAP or REST web services (or both). This is useful if you have your own high-level authentication built into the solution that is using, for example, the Location Intelligence Module services.

To disable authentication for web services on the Spectrum™ Technology Platform :

1. Stop the Spectrum™ Technology Platform server.
2. Open the following file in a text editor:

SpectrumLocation\server\app\conf\spectrum-container.properties

3. Change the value of each property as needed. For example, to disable authentication for all SOAP services:

```
spectrum.security.authentication.webservice.enabled.REST=true
spectrum.security.authentication.webservice.enabled.SOAP=false
```

> **Note:** For the Location Intelligence Module, REST services also include OGC web services.

4. Save and close the properties file.
5. Start the Spectrum™ Technology Platform server.

Once finished, authentication is turned off for the type of web services that you specified.

Enabling CORS

Cross-Origin Resource Sharing (CORS) is a W3C standard that allows data sharing between domains. CORS enables web applications running in one domain to access data from another domain. By enabling CORS on your Spectrum™ Technology Platform server, you can allow web applications hosted in another domain to access Spectrum™ Technology Platform web services.

For example, say you have a web application hosted at **webapp.example.com**. This web application contains a JavaScript function that calls a Spectrum™ Technology Platform web service hosted at **spectrum.example.com**. Without CORS, you would need to use a proxy server to facilitate this request, which would add complexity to your implementation. With CORS, you do not need to use a proxy server. Instead, you can designate **webapp.example.com** as an "allowed origin", thus permitting Spectrum™ Technology Platform to respond to web service requests that originate from the domain **webapp.example.com**.

To enable CORS on your Spectrum™ Technology Platform server:

1. Stop the Spectrum™ Technology Platform server.
2. Open this file in a text editor:

    SpectrumLocation/server/app/conf/spectrum-advanced.properties

3. Edit the following parameters.

    spectrum.jetty.cors.enabled

    > Set this property to true to enable CORS. The default is false.

    spectrum.jetty.cors.allowedOrigins

    > A comma separated list of origins that are allowed to access resources on the Spectrum™ Technology Platform server. The default value is http://localhost:8080,http://localhost:443, which allows access to resources using the default HTTP port 8080 and the default HTTPS port of 443.

If an allowed origin contains one or more asterisks ("*"), for example http://*.domain.com, then asterisks are converted to .* and dots characters (".") are escaped to "\." and the resulting allowed origin is interpreted as a regular expression. Allowed origins can therefore be more complex expressions such as https?://*.domain.[a-z]{3} that matches http or https, multiple subdomains and any three-letter top-level domain (.com, .net, .org, etc.).

spectrum.jetty.cors.allowedMethods

A comma separated list of HTTP methods that are allowed to be used when accessing resources on the Spectrum™ Technology Platform server. The default value is POST,GET,OPTIONS,PUT,DELETE,HEAD.

spectrum.jetty.cors.allowedHeaders

A comma separated list of HTTP headers that are allowed when accessing resources on the Spectrum™ Technology Platform server. The default value is X-PINGOTHER, Origin, X-Requested-With, Content-Type, Accept. If the value is a single asterisk ("*"), all headers will be accepted.

spectrum.jetty.cors.preflightMaxAge

The number of seconds that preflight requests can be cached by the client. The default value is 1800 seconds, or 30 minutes.

spectrum.jetty.cors.allowCredentials

Indicates whether the resource allows requests with credentials. The default value is true.

4. Save and close the file.
5. Start the Spectrum™ Technology Platform server.

Disabling Host Checks in Token Authentication

In token authentication, the Spectrum™ Technology Platform server examines the token presented by the client before responding to the request. The server checks the token to see if it has expired, if it is encrypted correctly, and if it is from the correct host. For session tokens, the server also checks the session ID. If any of these checks fail, the token is rejected and the server does not respond to the request.

In a clustered environment, it is possible that requests may be redirected in a way that makes the request appear to be coming from a different host than is specified in the token, resulting in "invalid token" errors. For example, say you have a cluster with two nodes as shown here:
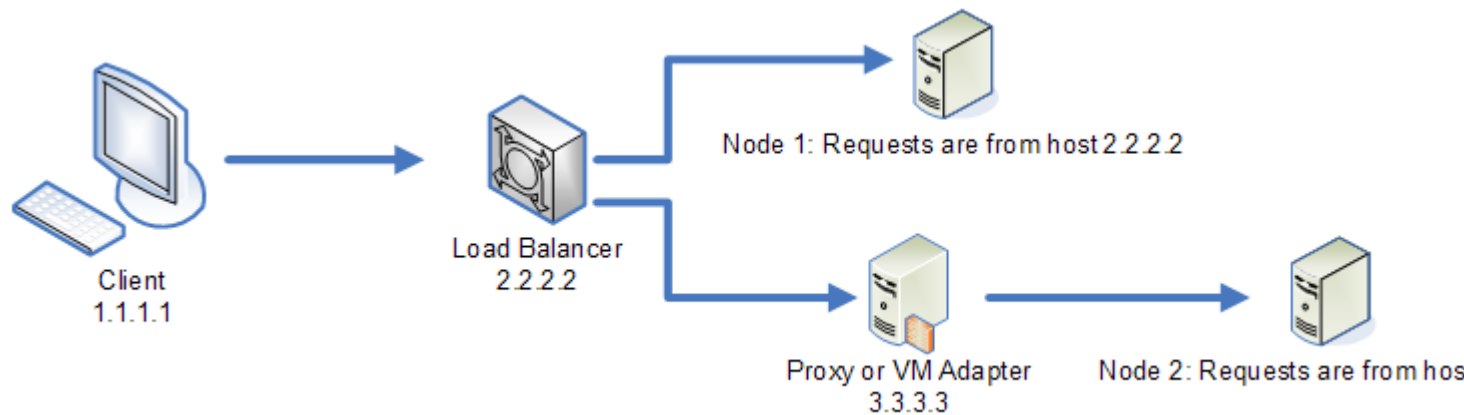
Let's say that the client makes a request and is the request is routed to Node 1. A token is created and tied to host 2.2.2.2 (the load balancer) since the node views the request as coming from the load balancer. If the next request from the client is routed to Node 2, the token will still be tied to host 2.2.2.2 but the request will appear to be coming from the proxy server, 3.3.3.3. In this case the node will reject the token because it appears that it is not associated with the host making the request.

In this situation you must configure the Spectrum™ Technology Platform server to ignore the host information included in the token. This should only be done if you have an environment where there are different network devices between the load balancer and the nodes. If all nodes are behind the same network device, there is no need to disable the host check.

> **Note:** If you follow this procedure, client tokens become "open" tokens, since the host check is disabled. Session tokens will continue to be associated with a specific session ID, but not with a specific host.

1. Open the following properties file on the Spectrum™ Technology Platform server:

   SpectrumLocation/server/app/conf/spectrum-container.properties

2. Set the following property to false.

   ```
   spectrum.security.authentication.token.remoteClientCheck.enabled=false
   ```

3. Save and close the properties file.
4. Repeat this process on all the nodes in the cluster.

Using LDAP or Active Directory for Authentication

Spectrum™ Technology Platform can be configured to use an LDAP or Active Directory server for authentication. When a user logs in to Spectrum™ Technology Platform, the user's credentials are verified using LDAP or AD. The system then checks to see if there is a Spectrum™ Technology

Platform user with the same name. If there is, the user is logged in. If there is not, then a Spectrum™ Technology Platform user account is automatically created for the user and given the role `user`.

The following diagram illustrates this process:
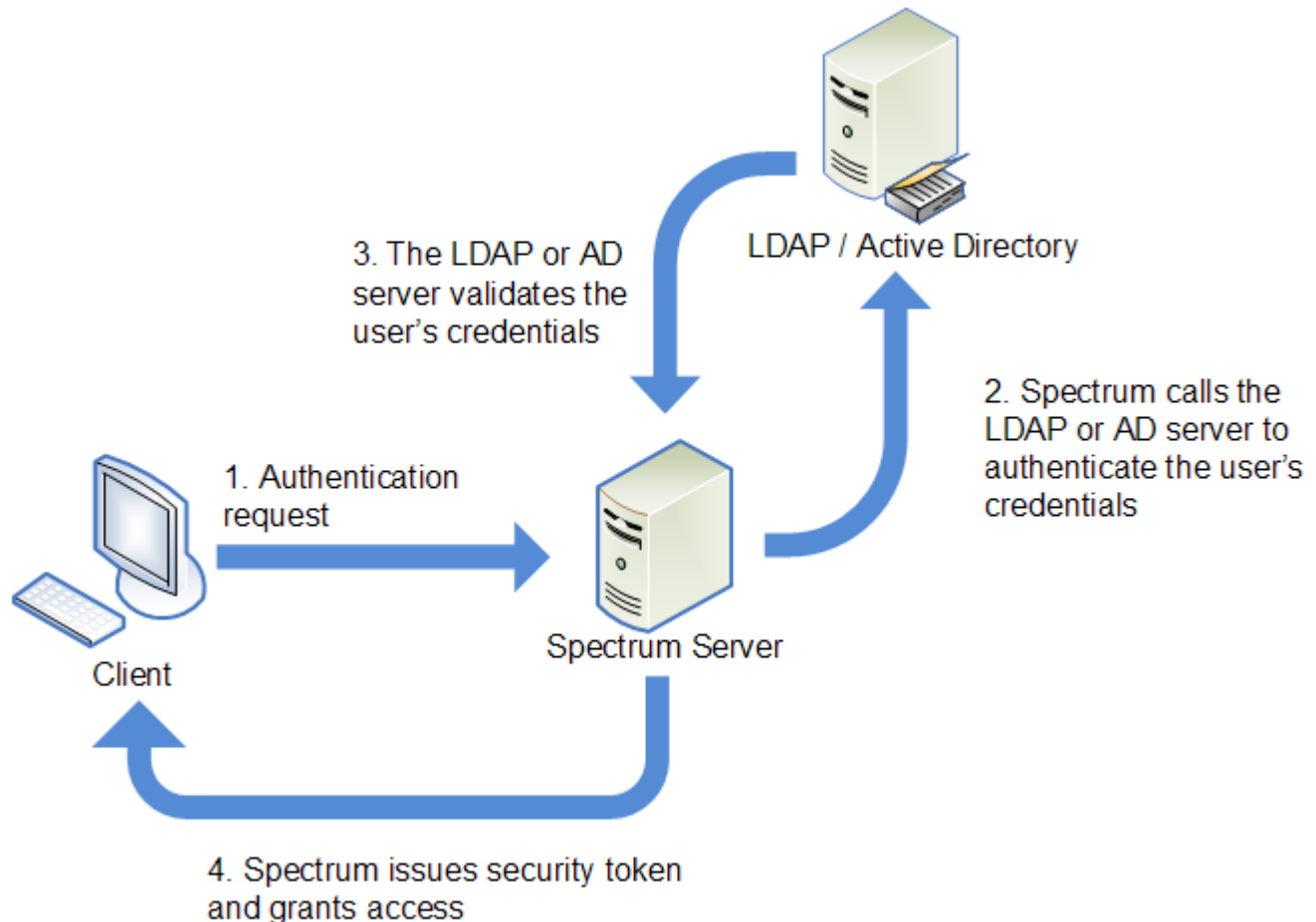
Before configuring Spectrum™ Technology Platform to use a directory service for authentication, confirm that your directory service meets these requirements:

• For LDAP, the directory server must be LDAP Version 3 compliant.
• There are no specific requirements for the Active Directory server.

> **Note:** We recommend that you contact Pitney Bowes Technical Support or Professional Services to guide you through this process.

> **Note:** When setting up Spectrum using LDAP or STS or SSO_STS , If the property is, by default, `spectrum.security.account.createNonExisting=true`, Active Directory users are created automatically in Spectrum™ Technology Platform after their first login to Spectrum. If you turn off the property `spectrum.security.account.createNonExisting=false`, LDAP/Active Directory users will not be authenticated to Spectrum™ Technology Platform until the administrator manually creates users.

1. If there are existing users configured in Management Console and you want to use them after you enable LDAP or Active Directory authentication, create those users in your LDAP or Active Directory system. Be sure to use the same user name as in Spectrum™ Technology Platform.

   Note: You do not need to create the "admin" user in LDAP or Active Directory since this user will continue to use Spectrum™ Technology Platform for authentication after you enable LDAP or Active Directory authentication.

2. Stop the Spectrum™ Technology Platform server.

3. Turn on LDAP or Active Directory authentication:

   a) Open this configuration file in a text editor:

   ```
   server\app\conf\spectrum-container.properties
   ```

   b) Set the property `spectrum.security.authentication.basic.authenticator` to LDAP:

   ```
   spectrum.security.authentication.basic.authenticator=LDAP
   ```

   The setting `LDAP` is used to enable Active Directory as well as LDAP.

   c) Save and close the file.

4. Configure the connection properties:

   a) Open this configuration file in a text editor:

   ```
   server\app\conf\spring\security\spectrum-config-ldap.properties
   ```

   b) Modify these properties.

   spectrum.ldap.url

   The URL, including port, of the LDAP or Active Directory server. For example,

   ```
   spectrum.ldap.url=ldap://ldapserver.example.com:389/
   ```

   spectrum.ldap.dn.format

   The format to use to search for the user account in LDAP or Active Directory. Use the variable `%s` for the user name. For example,

   LDAP:

   ```
   spectrum.ldap.dn.format=uid=%s,ou=users,dc=example,dc=com
   ```

   Active Directory:

   ```
   spectrum.ldap.dn.format=%s@example.com
   ```

   spectrum.ldap.dn.base

   The distinguished name (dn) to search for user accounts in LDAP or Active Directory. For example,

LDAP:

```
spectrum.ldap.dn.base=ou=users,dc=example,dc=com
```

Active Directory:

```
spectrum.ldap.dn.base=cn=Users,dc=example,dc=com
```

spectrum.ldap.search.filter

A search filter to use when searching for attributes such as roles. The search filter can contain these variables:

- `{user}` is the user name logging into Spectrum™ Technology Platform
- `{dn}` is the distinguished name specified in `spectrum.ldap.dn.base`.

For example:

LDAP:

```
spectrum.ldap.search.filter=uid={user}
```

Active Directory:

```
spectrum.ldap.search.filter=userPrincipalName={dn}
```

spectrum.ldap.attribute.roles

Optional. Specifies the LDAP or Active Directory attribute that contains the name of the Spectrum™ Technology Platform roles for the user. The role name you specify in the LDAP or Active Directory attribute must match the name of the role defined in Spectrum™ Technology Platform.

For example, to apply the roles defined in the attribute `spectrumroles` you would specify:

```
spectrum.ldap.attribute.roles=spectrumroles
```

If this attribute contains a role named `designer` then the `designer` role would be granted to the user.

You can only specify one attribute but the attribute may contain multiple roles. To specify multiple roles inside an attribute, separate each with a comma. You can also specify a multi-value attribute, with each instance of the attribute containing a different role. Only the roles specified in this one attribute are used in Spectrum™ Technology Platform. No other LDAP or Active Directory attributes will have any impact on Spectrum™ Technology Platform roles.

If the user has roles assigned to it in Spectrum™ Technology Platform, the user's permissions are the union of the roles from LDAP or Active Directory and the roles from Spectrum™ Technology Platform.

Note: When a user logs in for the first time, if the user does not have a Spectrum™ Technology Platform user account one is created automatically and given the role `user`. The effective permissions for the user are the union of the permissions in the `user` role and the roles specified in the attributes listed in the `spectrum.ldap.attribute.roles` property.

Note: When you view the user's roles in Management Console you will not see the roles assigned to the user by the `spectrum.ldap.attribute.roles` property.

spectrum.ldap.pool.min

The minimum size of the connection pool for connections to the LDAP or Active Directory server.

spectrum.ldap.pool.max

The maximum number of simultaneous connections to the LDAP or Active Directory server.

spectrum.ldap.timeout.connect

Specifies how long to wait to establish a connection to the LDAP or Active Directory server, in milliseconds. The default is 1000 milliseconds.

spectrum.ldap.timeout.response

Specifies how long to wait for a response from the LDAP or Active Directory server after the connection is established, in milliseconds. The default is 5000 milliseconds.

spectrum.ldap.retry.count

The number of times the Spectrum™ Technology Platform server will try connecting to the LDAP or Active Directory server if the initial connection attempt fails. Set this to 0 if you want to allow only one connection attempt.

Tip: If you cluster your LDAP or Active Directory servers, we recommend that you set this value to 1 or more to allow the LDAP or Active Directory load balancer to redirect the connection request to a different server if the one that is initially tried is unavailable.

spectrum.ldap.retry.wait

The number of milliseconds to wait between connection attempts.

spectrum.ldap.retry.backoff

The multiplication factor to use to increase the wait time after each failed retry attempt.

For example,

```
spectrum.ldap.timeout.connect=1000
...
spectrum.ldap.retry.count=5
spectrum.ldap.retry.wait=500
spectrum.ldap.backoff=2
```

In this example, the wait for the initial connection attempt is 1,000 milliseconds, and the wait time for each of the five retry attempts is increased by a factor of two, resulting in these wait times for each retry attempt:

Retry attempt 1: 500 milliseconds
Retry attempt 2: 1,000 milliseconds
Retry attempt 3: 2,000 milliseconds
Retry attempt 4: 4,000 milliseconds
Retry attempt 5: 8,000 milliseconds

   c) Save and close the properties file.

5. If you are using the Location Intelligence Module and plan to map roles (as described in **Mapping LDAP Attribute Values to Roles** on page 44), additional manual configuration of the Jackrabbit configuration file (located in
SpectrumFolder`\server\modules\spatial\jackrabbit\workspaces\default\workspace.xml`)
is needed to make Spectrum Spatial aware of any dynamically assigned LDAP or Active Directory roles. Add the `checkRoles` parameter as shown below:

```
<!--
 Spectrum ACL provider.
-->
<WorkspaceSecurity>
 <AccessControlProvider class="com.mapinfo.repository.jackrabbit.acl

  .AccessControlProviderImpl">
  <param name="checkRoles" value="true"/>
 </AccessControlProvider>
</WorkspaceSecurity>
```

6. Start the Spectrum™ Technology Platform server.

If you are running Spectrum™ Technology Platform in a cluster, you must modify the `spectrum-container.properties` file and the `spectrum-config-ldap.properties` file on each of the servers in the cluster. Stop the server before modifying the file, then start the server after you are done modifying the file. If you mapped an LDAP attribute value to a role, this mapping will replicate to all nodes in the cluster, so you do not need to repeat the mapping procedure in the JMX console.

Mapping LDAP Attribute Values to Roles

Before performing this procedure you must enable LDAP authentication. If you are using the Location Intelligence Module, this also includes modifying the Jackrabbit configuration file. For more information, see **Using LDAP or Active Directory for Authentication** on page 39.

When you configure Spectrum™ Technology Platform to use LDAP or Active Directory for authentication, one of the configuration properties that you configure (the `spectrum.ldap.attribute.roles` property in the file `spectrum-config-ldap.properties`) specifies an LDAP attribute whose values determine the role to grant to a user. By default, the

attribute values must match the Spectrum™ Technology Platform role names exactly in order for the role to be granted. For example to grant the designer role, the attribute you specify must contain the value designer.

If the LDAP attribute value that you want to use does not match the role name in Spectrum™ Technology Platform, you can map the LDAP attribute value to a role name. You can also map an LDAP attribute value that has the same name as a Spectrum™ Technology Platform role to a different role. For example, one of the built-in roles is designer. If you have an LDAP attribute value named designer but you want it to map to another role, you could create a mapping.

1. Open a web browser and go to `http://server:port/jmx-console`

   Where:

   server is the IP address or hostname of your Spectrum™ Technology Platform server.

   port is the HTTP port used by Spectrum™ Technology Platform. The default is 8080.

2. Click this property:

   `com.pb.spectrum.platform.common.security.ldap:mappings`

   > **Note:** This property is only visible after you enable LDAP authentication and the server is fully started. If you have not enabled LDAP authentication, see **Using LDAP or Active Directory for Authentication** on page 39.

3. In the **addMapping** section, in the **ldapValue** field, enter the LDAP attribute value that you want to map to a Spectrum™ Technology Platform role.

4. In the **roleName** field, enter the Spectrum™ Technology Platform role that you want to map to the LDAP attribute value.

5. Click **Invoke**.

Users who have the LDAP attribute will now be granted the role you specified when they long in to Spectrum™ Technology Platform.

To remove a mapping, enter the LDAP attribute you want to un-map in the **ldapValue** field in the **removeMapping** section.

Example

For example, say you want to use a value in the `gecos` attribute to assign a role in Spectrum™ Technology Platform. If `gecos` contains the value data-quality-user, you want the user to be granted the designer role when logging in to Spectrum™ Technology Platform.

To accomplish this, you would specify the `gecos` attribute as the attribute to use assign roles by specifying this in the file `spectrum-config-ldap.properties`:

```
spectrum.ldap.attribute.roles=gecos
```

Then, you would map the data-quality-user value to the designer role in the JMX console:

As a result, any user that has the value data-quality-user in the `gecos` attribute will be granted the role designer.

Enabling SSL Communication with LDAP

Communication between Spectrum™ Technology Platform and an LDAP or Active Directory server uses TCP by default. You can configure Spectrum™ Technology Platform to use LDAP over SSL if you want to secure the communication between the Spectrum™ Technology Platform server and the LDAP or Active Directory server.

1. You may need to add the certificate to the Java TrustStore used by Spectrum™ Technology Platform if:

   • The default Java TrustStore does not contain an entry for the certificate authority you are using.
   • You are using a self-signed certificate. Note that using a self-signed certificate is not recommended in a production environment.

   If either of these situations applies to you, add the certificate to the Java TrustStore by following these steps:

   a) Obtain a copy of the certificate. You can get a copy of the certificate from your LDAP administrator or by using a tool like LDAP Admin to view and save the certificate.
   b) Add the certificate to a new or existing TrustStore using the `keytool` utility included in the JDK.

For example:

```
keytool -import -file X509_certificate_ldap.cer -alias
server.example.com -keystore ldapTrustStore
```

See the Java documentation for more information.

> **Note:** The certificate must meet the requirements for encryption and length for the version of Java used by Spectrum™ Technology Platform. To find out the version of Java, open Management Console and go to **System** > **Version**. For more information, see **java.com/en/jre-jdk-cryptoroadmap.html**.

2. Stop the Spectrum™ Technology Platform server.

   • To stop the server on Windows, right-click the Spectrum™ Technology Platform icon in the Windows system tray and select **Stop Spectrum™**. Alternatively, you can use the Windows Services control panel and stop the Pitney Bowes Spectrum™ Technology Platform service.
   • To stop the server on Unix or Linux, source the *SpectrumLocation*/server/bin/setup script then execute the *SpectrumLocation*/server/bin/server.stop script.

3. Open this file in a text editor:

   SpectrumLocation\server\app\conf\spring\security\spectrum-config-ldap.properties

4. Configure these properties:

   spectrum.ldap.url

   > Specify the URL of the LDAP server. Be sure to specify the SSL port number, which is typically 636. For example:
   >
   > ```
   > spectrum.ldap.url=ldap://server.example.com:636
   > ```
   >
   > > **Note:** Do not include a slash (/) at the end of the URL.

   spectrum.ldap.useSSL

   > Specify true to enable SSL communication with LDAP.

   spectrum.ldap.trustStore

   > Specify the location of the TrustStore containing the certificate to use for SSL communication with LDAP. For example on Windows:,
   >
   > ```
   > spectrum.ldap.trustStore=file:D:\\Certs\\MyTrustStore
   > ```
   >
   > On Linux and Unix:
   >
   > ```
   > spectrum.ldap.trustStore=file://Certs//MyTrustStore
   > ```

   spectrum.ldap.trustStore.password

Specify the TrustStore password.

Important: If you are running Spectrum™ Technology Platform in a cluster, repeat this procedure on each server in the cluster.

Disabling SSL Communication with LDAP

If you have configured Spectrum™ Technology Platform to use SSL communication with LDAP or Active Directory and need to switch back to using TCP, follow this procedure.

1. Stop the Spectrum™ Technology Platform server.

   • To stop the server on Windows, right-click the Spectrum™ Technology Platform icon in the Windows system tray and select **Stop Spectrum™**. Alternatively, you can use the Windows Services control panel and stop the Pitney Bowes Spectrum™ Technology Platform service.
   • To stop the server on Unix or Linux, source the *SpectrumLocation*/server/bin/setup script then execute the *SpectrumLocation*/server/bin/server.stop script.

2. Open this file in a text editor:

   SpectrumLocation\server\app\conf\spring\security\spectrum-config-ldap.properties

3. Configure these properties:

   spectrum.ldap.url

   Change the URL of the LDAP server to use the TCP port rather than the SSL port. The default is 389. For example:

   ```
   spectrum.ldap.url=ldap://ldapserver.example.com:389/
   ```

   Note: You must include a slash (/) at the end of the URL.

   spectrum.ldap.useSSL

   Specify false to disable SSL communication with LDAP.

   spectrum.ldap.trustStore

   Comment out this property.

   spectrum.ldap.trustStore.password

   Comment out this property.

Security for the Location Intelligence Module

The Location Intelligence Module uses the role-based security that is used for the Spectrum™ Technology Platform. Because security is handled at the platform level, Management Console can be used to manage all Location Intelligence Module security activities. This includes setting permissions for named resources in addition to managing user accounts (that is, creating, modifying, and deleting user accounts).

Predefined Spatial Roles

After you install the Location Intelligence Module, three predefined roles are available in Management Console:

spatial-admin
The spatial-admin role provides full permissions (Create/View/Modify/Delete) for all named resources and datasets associated with named tables. These permissions are controlled using the Location Intelligence Module's secured entity types, Location Intelligence.Named Resources and Location Intelligence.Dataset.DML. Users of Location Intelligence Module services must have at least View permissions for the resources they use as well as for any dependent resources. See **Access Control for Datasets** on page 56 for more information on controlling dataset permissions.

spatial-user
The spatial-user role provides View permissions to named resources only. These permissions are controlled using the Location Intelligence Module's secured entity type, Location Intelligence.Named Resources. Users of Location Intelligence Module services must have at least View permissions for the resources they use as well as for any dependent resources.

spatial-dataset-editor
The spatial-dataset-editor role provides full permissions (Create/View/Modify/Delete) on datasets. These permissions are controlled using the Location Intelligence Module's secured entity type, Location Intelligence.Dataset.DML. See **Access Control for Datasets** on page 56 for more information on this role and controlling permissions on datasets.

Dataflow designers who require access to named resources need additional permissions beyond that of the "designer" role. For instructions on creating a spatial dataflow designer, see **Creating a Spatial Dataflow Designer** on page 59.

Custom Spatial Roles and Access Control Settings

You can create custom roles based on the predefined spatial roles, assign them to user accounts, then fine-tune access to named resources for those roles and users by applying access control settings (overrides) to individual named resources, datasets, or to folders or directories. A typical scenario and best practice for setting security for the Location Intelligence Module involves creating a role with no permissions, applying access control settings to that role (for example, allowing modify and delete permissions for named resources in a specific folder), then assigning that custom role as well as one of the predefined spatial roles to a user. Another common scenario involves

establishing override permissions for a single user; for example, creating a user account which has view-only permissions to named resources, then applying access control settings to that user that allow modifying and deleting of named resources in a specific folder.

Folders

Folder permissions are inherited by the resources and folders underneath as long as those resources and folders do not have any specific access control settings that override them. This is useful when you want to set permissions on a set of resources. You can make a folder accessible only to specified users or roles; other users will not see that folder or anything underneath it. For the Location Intelligence.Named Resources entity type, all listed resources that end with a forward slash (/) are folders or directories in the repository.

Permissions at the folder level, however, do not override permissions set at the lower, individual resource level. For example, if a folder has Create permissions for a specific role or user, but a single resource in the folder (such as a named table) has an access control setting to View permissions for that same role or user, the View (read-only) permissions for the single resource take precedence over the Create permissions for the folder.

Understanding ACL

Overview

The Access Control List (ACL) in Spectrum Spatial is a list of permissions attached to named resources or to folders in the Spectrum Spatial repository. Permissions can be granted to enable users to render maps, query or edit features, or to manage folders and resources within the repository. Permissions can be assigned to either users or roles. Users will inherit all of the permissions from the roles to which they belong.

The way permissions are defined has been fully revised in this version. When upgrading previous versions of Spectrum, the permissions will be migrated to the new model. Also new in this release, it is now also possible to grant permissions on repository folders to allow users other than admin to manage specific folders in the repository. Users with these permissions are referred to as sub-admins. A new ACL service API has been provided to facilitate listing, adding, and removing permissions. In future releases of Spectrum, a new User Interface in Spectrum Spatial Manager will be provided to manage permissions, which will replace the UI in the Spectrum Management Console. At this time sub admins will also be able to log into and use Spectrum Spatial Manager. In this release sub admins are only able to manage resources and their ACL via the REST APIs.

The management of users and roles is still undertaken using the Spectrum Management Console. The ACL service does not provide operations for managing users or roles.

ACL and Repository

The ACL permissions that can be granted using the ACL services fall into three categories.

- **Folder ACL:** Grants permissions for managing the content of the repository (including uploading, creating, and deleting named resources, and setting further permission on them). These permissions are granted on repository folders. Users with these permissions are able to view or modify named resources within the folders on which they are granted permission. Users who have permissions on one or more folders are called sub-admins because they can manage a sub-set of the repository. Sub-admins have access to the Named Resource and ACL services and in future releases will be also able to log into the Spatial Manager and Map Uploader to manage resources.
- **Resource ACL:** Grants permissions for rendering specific named tiles, named maps and named layers. These permissions are granted on the resources themselves. Users with these permissions are able to use the mapping and tiling services to render and describe mapping resources. Users who are sub-admins will also inherit resource permissions to render resources that are within their folders.
- **Dataset ACL:** Grants permissions for querying or editing specific named tables (i.e. CRUD operations for create, read, update, delete). These permissions are granted on the named table resources themselves. Users with these permissions are able to query features from the tables or to insert/update/delete features. Users who are sub-admins will also inherit dataset permissions to query any tables.. However, they do not inherit the dataset insert, update or delete permissions. To edit tables, they must be given these permissions in addition to the folder permissions.

The following table summarizes the three categories, the named resources they affect and the specific permissions that can be granted under each category. There are also some named resources which do not have permissions granted on them. These are also listed in the table.

Table 1: Summary of ACL Permissions

Type of Permission	Granted On	Permissions set using ACL Services	**Permissions that are persisted to the Spectrum Platform**	Activities that users can perform
Folder Permission	Repository Folders	READ	NamedResource.EXECUTE NamedResource.VIEW	The user can view folders, subfolders, and their content as sub-admin. The user can render any maps and layers within their folders. The user can query any tables within their folders.
		WRITE	NamedResource.CREATE NamedResource.DELETE NamedResource.MODIFY	The user can create, delete, or modify resources within their folders including uploading resources and setting new ACL permissions on them.

Type of Permission	Granted On	Permissions set using ACL Services	**Permissions that are persisted to the Spectrum Platform**	Activities that users can perform
Resource Permission	Named Tiles, Named Maps, Named Layers, and Named Label Sources	EXECUTE	NamedResource.EXECUTE	The user can render the maps and layers on which they have this permission.
Dataset Permission	Named Tables and Named View Tables	EXECUTE	NamedResource.EXECUTE	The user can query the data from the tables on which they have this permission.
		CREATE	Dataset.DML.CREATE	User can insert new records into the tables on which they have this permission.
		DELETE	Dataset.DML.DELETE	The user can delete records from the tables on which they have this permission.
		MODIFY	Dataset.DML.MODIFY	The user can update records in the tables on which they have this permission.
No permissions required	Named Styles	There is no ACL applied to the Named Styles. Any named style referenced in a layer or WMS can be accessed when rendering the layer.		
	Named Connections	There is no ACL applied to the Named Connections. Any Named connection can be used when querying data from a Named Table. However, a Named Connections can only be seen by the users who are sub-admins (i.e. who have folder permissions) via the Named Resource Service.		
	Metadata Resources	There is no ACL applied to the Named Resource Metadata. Currently these can only be viewed by the admins or the users who are sub-admins (i.e. who have folder permissions) via the Named Resource Service.		

ACL and Accessing Services and Applications

Service and application access is restricted depending on the ACL that has been granted. The following list describes the permissions needed by users. Full details are provided under each service method in REST and SOAP guide for each service.

- **Mapping Service (REST and SOAP):** Users can list, describe and render the maps and layers on which they have resource EXECUTE permission. Permission is not required for underlying resources to render a specific map or layer (but will be needed if a client application also needs to describe or access the underlying resources if they are presented to users).
- **Map Tiling Service (REST and SOAP):** Users can list, describe and render the named tiles on which they have resource EXECUTE permission. Permission is not required for underlying resources to render a specific tile (but will be needed if a client application also needs to describe or access the underlying resources if they are presented to users).
- **Feature Service (REST and SOAP):** Users can list, describe and query features from the named tables and views on which they have dataset EXECUTE permission. Users can insert, update and delete features from the named tables on which they have dataset CREATE, MODIFY or DELETE permission
- **Named Resource Service (SOAP):** In order to use any operation in the Named Resource Service a user must have folder permissions on at least one folder (and they must have READ or WRITE on the folders to see or manage the resources)
- **ACL Service (REST):** The listDatasetPermissions and listFolderPermissions in the ACL service are available to all users. In order to use the other "ACL" operations (to list, add or delete any resource, folder or dataset permissions) a user must have folder permissions on at least one folder (and they must have READ or WRITE on the folders to see or manage the resources).
- **WMTS:** There are no ACL permissions applied to Named WMTS tiles. If a Named WMTS tile is created this implies READ access to it via the WMTS service. ACL permissions are not required for the underlying resources. A user will be able to access the tile via the WMTS service (but not via the other services, unless they have specific resource permissions).
- **WMS:** For the WMS service adding a layer to service implies read access to it via the WMS service. ACL permissions are not required on the underlying Named Layer resource. The layer will be listed in the capabilities file and users will be able to render the map and legend and get feature info via the WMS service (but not via the other services, unless they have specific resource permissions)
- **WFS:** For the WFS service adding a table to service implies read access to it via the WFS. ACL permissions are not required on the underlying Named Table resource. The table will be listed in the capabilities file and users will be able to query features via the WFS service (but not via the other services, unless they have specific resource permissions)
- **Spatial Manager:** In order to manage resources in the Spatial Manager application, a user must have spatial admin permissions. Currently users who are sub-admins can manage resources using the service APIs.
- **Map Uploader:** In order to upload using the Map Uploader a user must have spatial admin permissions. Currently users who are sub-admins can manage resources using the service APIs.
- **Dataflows in Enterprise Designer:** In order to execute dataflows, a user must have admin or spatial admin along with designer role permissions. The user must have EXECUTE permissions

on namedtables and Create/Modify/Delete on the dataset to perform DML operations for the supported writable table.

ACL Management

It is recommended to use the ACL services to add and remove ACL rather than using the Spectrum Management Console. The ACL service APIs are documented in the REST API section of the Spectrum Spatial guide. The services will ensure that the correct combination of permissions is persisted to Spectrum Platform.

The ACL services can also propagate (recurs) permissions to the dependent resources. This is important when using Spectrum Spatial with client applications (such as Spectrum Spatial Analyst) where users need to render maps, render the layers that the maps reference, and also need permissions to query features for the tables that the layers reference.

The Spectrum Management Console can be used to view the permissions that are granted. If the Spectrum Management Console is used to modify permissions then these rules must be followed to ensure the consistency of the permissions granted:

- There should not be any deny permissions granted on any resources or folders. Deny permissions will prevent:

  - users inheriting permissions from roles
  - sub-admins inheriting permissions from folders

- To provide render and query access to named resources only NamedResource.EXECUTE should be granted. Never grant NamedResource.VIEW, NamedResource.CREATE, NamedResource.DELETE or NamedResource.MODIFY to named resource directly (these permissions convey sub-admin privileges and should only be granted on folders).
- To provide dataset edit permissions to named tables, grant any one of Dataset.DML.CREATE, Dataset.DML.DELETE, or Dataset.DML.MODIFY permissions as appropriate.
- To provide READ access to repository folders to sub-admins, grant both NamedResource.EXECUTE, NamedResource.VIEW on the folders. These permissions should always be granted together. Do not grant one without the others.
- To provide WRITE access to repository folders to sub-admins grant all three of the NamedResource.CREATE, NamedResource.DELETE and NamedResource.MODIFY permissions. These three permissions should always be granted together. Do not grant one or two without the others.
- Do not grant any permissions on Named Connections, Named Styles, Named WMTS Tiles, or on any metadata resources.
- If client applications are accessing maps, layers, and tables then permissions need to be set on all of the dependent named resources that are to be used.

Upgrading With ACL

Migration to 12.0 SP2

When upgrading to 12.0 SP2, the ACL model gets upgraded to a new security model. The migration script will run as a part of installation but it also can be run independently of the installation procedure.

> **Note:** If you export CLI, it does not alter the permissions. These permission will need to be migrated. For example the resources from 12.0 SP1 and before come with VIEW permission but when you upgrade to 12.0 SP2, the migration script changes VIEW to EXECUTE.

The migration script will run the first time Spectrum server is started after upgrading to 12.0 SP2. It will only run once. The migration script will perform the following:

- All the permissions on folders will be removed.
- All the permissions on resources will be removed.
- The new EXECUTE permission will be applied if the VIEW permission existed on any resources.
- Any existing Deny permissions will be removed.
- Dataset VIEW permission is removed. Other permissions on Dataset such as CREATE, MODIFY, and DELETE will be preserved if they existed prior to upgrade.
- NamedResourceMetadata resources will have no ACL for now.

> **Note:** Having added the new EXECUTE permission, the Spectrum will start over as a new slate.

Understanding CLI Changes for ACL

With the new ACL security model in place, the CLI has updated permissions. This section describes the changes made to the CLI tool to import and export the new ACL model can be imported and exported.

Working with CLI

Only an Admin or spatial-admin can run the CLI utility. In 12.2, you can export with minimum permissions - just VIEW permission on a resource and import with just CREATE permissions on target folder. However, in general, only admin and sub-admin should be able to import and export within 12.2.

- A user can export the resources if they have the permission to VIEW.
- A user can import the resources if they have the permission to CREATE to the folder in the repository.

Export

The exporter must have VIEW (resource) permissions on all resources to be exported otherwise an Access denied exception is thrown and the export process is stopped.

When you export with --a option, after all the resources are exported successfully, the ACL of all exported resources will also be exported (including entities of all other Users/Roles).

Import

A user must have CREATE (resource) permissions on the target folder in the repository otherwise an Access denied exception is thrown and import process is stopped.

When you import with --a option, all ACL will be merged into existing ACL registered in the system (same as in earlier versions).

When you import resources exported from an older version with --a specified, the ACL will be upgraded based on the following rules before merge into existing ACL in the system:

• All permissions on folders will be ignored.
• All permissions on resources will be ignored.
• The EXECUTE permission will be added if a resource had VIEW permission.
• All DENY permissions will be ignored.
• All dataset permissions will be merged as before.

Access Control for Datasets

What is a Dataset?

A dataset is a collection of data values in a tabular form that typically consists of rows (or records) and columns. In the Location Intelligence Module, a dataset can take the form of a .TAB file, a shapefile, a GeoPackage file, or a JDBC-based table such as an MS SQL Server table.

Benefits of Dataset Access Control

Dataset access control allows administrators to disassociate the permissions of a named table from the editing permissions of the dataset that the named table points to. For example, as an administrator you can grant full editing (Create/Modify/Delete) permissions to a dataset while keeping read-only (Execute) permissions on the named table. When a user attempts to perform a data manipulation language (DML) operation (an insert, update, or delete operation using the Feature service or the Write Spatial Data stage), the user's permissions will be verified not only against the specified named table in the Location Intelligence.Named Resources entity type but also against the Location Intelligence.Dataset.DML entity type. If Execute permissions are denied, the named table will not appear in the user's repository.

What is a Dataset Secured Entity?

The LocationIntelligence.Dataset.DML secured entity is one of the two types of secured entities for the Location Intelligence Module. It controls DML permissions to datasets that are associated with

named tables. When a named table is created or uploaded (using any tool, including Spatial Manager, the Administration Utility, the Named Resource Service, and WebDAV), a new LocationIntelligence.Dataset.DML secured entity is automatically created for the associated dataset of that named table. A user must have Execute permissions on a named table *and* Create/Modify/Delete permissions on the dataset in order to perform DML operations on writable (JDBC-based) tables. DML operations include insert, update, and delete operations performed using the Write Spatial Data stage or the Feature Service.

> **Note:** Although you can set Create/Modify/Delete permissions on dataset secured entities for non-writable datasets such as .TAB files or shapefiles, you still cannot perform DML operations on these datasets.

Tip: The Execute permission on the secured entity for the dataset has no impact on its permissions. If you turn the Execute permission off on a dataset secured entity you will still be able to view the data in the table. If you do not want a user to see a table, remove Execute permissions on the secured entity for the named resource instead.

When a named table is renamed, moved, or deleted, Spectrum Spatial will rename or delete the associated secured entity for the dataset.

Spatial Roles and Dataset Access

Roles are used to grant or deny access to different parts of the system and help make permissions management easier. Three predefined roles for users of the Location Intelligence Module are available in Management Console:

spatial-admin	The spatial-admin role provides full permissions (Execute/Create/Modify/Delete) for all named resources and datasets. A user with a spatial-admin role can view named resources as well as edit datasets.

> **Note:** Additional file-server access is required to create or edit the source folder for named connections that are file-system based as well as certain settings in service configuration files (such as the image directory for the Mapping Service). For more information, see **Creating a Named Resources Administrator** on page 58.

spatial-user	The spatial-user role provides the Execute permissions to named resources only. A user with a spatial-user role can view resources but cannot edit datasets.
spatial-dataset-editor	The spatial-dataset-editor role provides full permissions (Execute/Create/Modify/Delete) on datasets. For example, an administrator can easily grant full permissions to datasets by adding the spatial-dataset-editor role to a user who currently has the spatial-user role.

These predefined roles cannot be modified. You can, however, create custom roles based on the predefined spatial roles, assign them to user accounts, then fine-tune access on those roles and users by applying access control settings (overrides) to datasets, individual named resources, or

folders containing named resources. See **Configuring Access Control** on page 28 for more information.

Creating a Named Resources Administrator

To manage named resources in the repository using Spatial Manager and Management Console, a user must have an assigned role that allows full access to those resources in addition to the access that is provided by the predefined spatial roles. The predefined spatial roles cannot be modified and a predefined "Named Resources Administrator" role is not provided by the Spectrum™ Technology Platform; however, you can create such a role using a predefined spatial role as a base.

1. Open Management Console.
2. Go to **System** > **Security**.
3. Click **Roles**.
4. Check the box next to the spatial-admin role to use as a starting point then click the Copy button
    . The spatial-admin role provides View, Modify, Create, and Delete permissions for the Location Intelligence Module.Named Resources and Location Intelligence Module.Dataset secured entity types.
5. In the **Role name** field, enter the name you want to give to this role (for example, "resource-admin").
6. Set additional permissions as follows for these secured entity types:

    Database Resources:

    • **Centrus Database Resources** to View/Modify/Create/Delete/Execute (if required)
    • **Enterprise Routing** to View/Modify/Create/Delete/Execute (if required)

    Platform:

    • **Services** to View/Modify/Execute
    • **System - Version Information** to View

    Resource Connection:

    • **Resources - File Server Connections** to View
    • **Resources - JDBC Drivers** to View

7. Click **Save** to save the new resource-admin role.
8. Click **Users**.
9. Either select an existing user and click the Edit button   to modify it, or click the Add button
     to create a new user.
10. Assign the new "resource-admin" role to the user account to allow it to manage named resources

The user now has the access required to manage named resources in Spatial Manager and Management Console.

Creating a Spatial Dataflow Designer

To create dataflows for Location Intelligence Module stages and services, a user must have both the designer and spatial-user roles assigned. The spatial-user role provides View access to named resources under the Location Intelligence.Named Resources secured entity type. The designer role provides the necessary access to Platform secured entity types such as Dataflows.

1. In Management Console, go to **System** > **Security**.

2. Either select an existing user and click the Edit ✎ button, or click the Add button ✚ to create a new user.

3. In the Roles section, assign both the designer and spatial-user roles to the user account.

The user now has permission to view named resources and design dataflows using those resources for Location Intelligence Module stages and services.

Limiting WebDAV Access to the Repository

WebDAV is used as a protocol to access resources within the Spectrum Spatial repository. By default, accessing the repository using WebDAV is not restricted to a particular server, rather open to all servers that can access the repository. You can restrict access to particular servers by modifying the spatial java property file. You can do this by adding the following property that includes a list of hostnames (IPs) that WebDAV is open to (comma separated). A Spectrum™ Technology Platform server restart is required after the change.
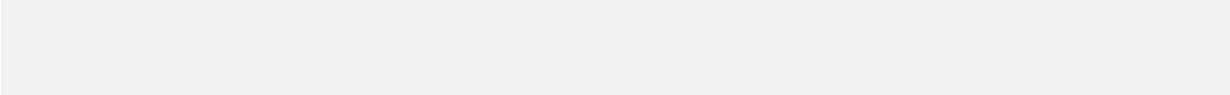
To limit repository access using WebDAV:

1. Open the `modules/spatial/java.properties` file in an editor.

2. Add the following property to the file.

```
repository.accesscontrol.allows=
```

3. Include a list of IP addresses that you want to allow WebDAV access. Multiple servers can be added using a comma separated list of IP addresses. Leaving the property empty disables all access using WebDAV for all servers except the machine where Spectrum™ Technology Platform is installed.

```
repository.accesscontrol.allows=192.168.2.1,192.168.2.2
```

4. Restart the server.

Once finished, WebDAV access is limited for the repository.

Using WebDAV with HTTPS

When communicating to the server over HTTPS to map a drive to the repository, a WebDAV client is required to use the TLS v1.2 protocol. For client machines running on Windows 7 SP1, Windows Server 2008 R2 SP1, and Windows Server 2012, you must apply a security patch and registry update to leverage this protocol.

1. On the client machine, apply the appropriate patch for the operating system from the Microsoft Knowledge Base: **https://support.microsoft.com/en-us/kb/3140245**
2. Follow the instructions in the KB article to update the registry to include support for TLS v1.2. The DefaultSecureProtocols value must be at least 0x00000800.
3. Restart the client machine after changing the registry entry.

4 - Monitoring Your System

In this section

Viewing System Events

The system log displays messages from the Spectrum™ Technology Platform server's wrapper log. These messages include information about server operations as well as requests made to services from the API and through web services. View the system log when you experience trouble and are looking for information about possible causes.

If you are running Spectrum™ Technology Platform in a cluster, the system log that you will get will be the one from the node you happen to be connected to. You can view the system log for a specific node by using a text editor to open this file on the node you want:
ServerLocation\server\app\repository\logs\wrapper.log.

1. Open the Management Console.
2. Go to **System** > **Logs**.
3. Click the Download icon ⬇ to download the system log file.
4. Open the downloaded file in a text editor.

Setting Logging Levels for Services

You can specify the default logging level as well as logging levels for each service on your system. When you change logging levels the change will not be reflected in the log entries made before the change.

> **Note:** The logging levels you specify for services do not affect the audit log. They only control the level of logging for the event log which you can view in Management Console. At this time you cannot view the event log in the web version of Management Console.

1. Open the Management Console.
2. Go to **System** > **Logs**.
3. In the **System default logging level** field, select a default event logging level for services on your system.

Disabled	No event logging enabled.
Fatal	Minimal logging. Only fatal errors are logged. Fatal errors are those that make the system unusable.
Error	Errors and fatal errors are logged. Errors indicate an isolated problem that causes part of the system to become unusable. For example, a problem that causes a single service to not work would generate an error.

Warn	Event warnings, errors, and fatal errors are logged. Warnings indicate problems that do not stop the system from working. For example, when loading a service where a parameter has an invalid value, a warning is issued and the default parameter is used. During the use of a service, if results are returned but there is a problem, a warning will be logged.
Info	High-level system information is logged. This is the most detailed logging level suitable for production. Info events are typically seen during startup and initialization, providing information such as version information and which services were loaded.
Debug	A highly detailed level of logging, suitable for debugging problems with the system.
Trace	The most detailed level of logging, tracing program execution (method entry and exit). It provides detailed program flow information for debugging.

Each logging level includes the ones above it on the list. In other words, if Warning is selected as the logging level, errors and fatal errors will also be logged. If Info is selected, informational messages, warnings, errors, and fatal errors will be logged.

> **Note:** Selecting the most intensive logging level can affect system performance. Therefore, you should select the least intensive setting that meets your particular logging requirements.

4. If you want to specify different logging levels for each service choose the logging level you want.

Spatial Logging

The logback.xml file allows you to control on logging behavior, such as sending output to a log file instead of by default sending it to the console which redirects to the wrapper.log. You can also set the log level to turn off logging altogether or log only fatal errors, for example.

Default logback file
(<Installed>\Pitney Bowes\Spectrum\server\modules\spatial\logback.xml)

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
=========================================================================
 -->
<!-- Logger configuration for remote components
                 -->
<!--
                 -->
```

```
<!-- log to console, redirected to Platform log
(server\app\repository\logs\wrapper.log) -->
<!-- log to files, redirected to (server\modules\spatial\spatial.XXX.log)
                -->
<!--
                    -->
<!-- for general information about the configuration file, check out
the logback manual  -->
<!-- at http://logback.qos.ch/manual/configuration.html
                    -->
<!--
==================================================================================
 -->
<configuration>
 <appender name="CONSOLE-SPATIAL"
class="ch.qos.logback.core.ConsoleAppender">
  <encoder>
   <pattern>[Spatial] - [%thread] %-5level %logger{35} - %msg%n</pattern>

  </encoder>
 </appender>
 <!--appender name="FILE-SPATIAL"
class="ch.qos.logback.core.rolling.RollingFileAppender">
  <file>${g1.server.modules.dir}/spatial/${component.name}.log</file>
  <encoder>
   <pattern>%d [%thread] %-5level %logger{35} - %msg%n</pattern>
  </encoder>
  <append>true</append>
  <triggeringPolicy
class="ch.qos.logback.core.rolling.SizeBasedTriggeringPolicy">
   <maxFileSize>10MB</maxFileSize>
  </triggeringPolicy>
  <rollingPolicy
class="ch.qos.logback.core.rolling.FixedWindowRollingPolicy">
   <fileNamePattern>${component.name}.log.%i</fileNamePattern>
   <maxIndex>1</maxIndex>
  </rollingPolicy>
 </appender-->
 <!-- Level: OFF, ERROR, WARN, INFO, DEBUG -->
 <logger name="com.mapinfo.midev" level="INFO" additivity="false">
  <appender-ref ref="CONSOLE-SPATIAL"/>
  <!-- appender-ref ref="FILE-SPATIAL"/ -->
 </logger>
</configuration>
```

Option	Values
Level	• OFF–turn off logging • ERROR–log runtime or unexpected errors • WARN–log warnings only; for example, using a deprecated API • INFO–log runtime events such as startup or shutdown [default] • DEBUG–log detailed debugging information
Output	• CONSOLE-SPATIAL –sends log information to the console [default] • FILE-SPATIAL–sends log information to a log file based on component (no longer applicable - Spectrum Spatial has a single remote component)

Configuring a Mail Server

Spectrum™ Technology Platform can send email alerts to notify you of important events. Email notifications can be sent as a result of conditions within dataflows and process flows, and when time-based licenses, databases, and other items are about to expire.

Spectrum™ Technology Platform does not have a built-in mail server, so in order to enable email notification you must configure it to use an external SMTP server.

1. Open the Management Console.
2. Go to **System** > **Mail Server**.
3. In the **Host** field, enter the host name or IP address of the SMTP server you want to use to send email notifications.
4. In the **Port** field, enter a port number or range to use for network communication between the Spectrum™ Technology Platform server and the SMTP server.

   The default port is 25.

5. In the **User name** and **Password** fields, enter the credentials that the Spectrum™ Technology Platform server should use to authenticate with the SMTP server.
6. In the **From address** field, enter the email address from which notification e-mail will be sent.
7. To confirm that you have correctly configured a mail server, you can send a test email. Enter the email address you want to send the test to in the **Test address** field then click **Test**.
8. Click **Save**.

The Spectrum™ Technology Platform server is now connected to an SMTP server and can use that server to send notification email.

Example: Configuring a Mail Server

You have an SMTP server named mail.example.com. You want to use this mail server to handle email notifications sent from the Spectrum™ Technology Platform server. You have created an account on the SMTP server called Spectrum123 with a password of Example123, and the email address for this account is spectrum.notification@example.com.

To configure notification with this information, you would complete the fields as follows:

Host	mail.example.com
From address	spectrum.notification@example.com
User name	Spectrum123
Password	Example123

Selecting Items for Expiration Notification

Spectrum™ Technology Platform can send an email notification when a license, database, or software component is about to expire. This allows you to take the necessary action to ensure that your business processes are not disrupted by an expiration. Some of the components that have expiration dates include:

• Licenses

> **Note:** Email notifications are not available for transaction-based licenses. If you are approaching the maximum number of transactions for a license, a message appears in the system log in Management Console.

• Databases, such as U.S. postal databases used for CASS processing
• Certain software components, such as the engine used to validate U.S. addresses in the Universal Addressing Module

Tip: To view the items that have expiration dates, open Management Console and go to **System** > **Licensing and Expiration**.

You can choose which items you want to be notified about so that you only receive notifications for those items that concern you.

1. Open the Management Console.
2. Go to **System** > **Licensing and Expiration**.
3. To receive an expiration notification email for an item, check the box in the **Send Notification** column. If you want to be notified earlier or later than the default, specify the number of days in advance of the expiration that you want to be notified.

Viewing Version Information

1. In a web browser go to this URL:

   http://*server*:*port*/managementconsole

   Where *server* is the server name or IP address of your Spectrum™ Technology Platform server and *port* is the HTTP port used bySpectrum™ Technology Platform. By default, the HTTP port is 8080.

2. Click **System** > **Version**.

Viewing and Exporting License Information

You can export information about your license to an XML file. This may be necessary when resolving license issues with technical support.

1. In a web browser go to this URL:

   http://*server*:*port*/managementconsole

   Where *server* is the server name or IP address of your Spectrum™ Technology Platform server and *port* is the HTTP port used bySpectrum™ Technology Platform. By default, the HTTP port is 8080.

2. Click **System** > **Licensing and Expiration**.
3. Click the export icon.

Your license information is saved to an XML file with a `.lic` extension.

Monitoring Performance with the JMX Console

The JMX console is browser-based tool that provides a performance monitoring tool that records performance statistics for each stage in a dataflow.

1. Open a web browser and go to `http://`*server*`:`*port*`/jmx-console`

   Where:

   server is the IP address or hostname of your Spectrum™ Technology Platform server.

port is the HTTP port used by Spectrum™ Technology Platform. The default is 8080.

2. Log in using the admin account.
3. Under " Domain: com.pb.spectrum.platform.performance", click
   com.pb.spectrum.platform.performance:service=PerformanceMonitorManager.
4. Click the **Invoke** button next to **enable**.
5. Click **Return to MBean View** to go back to the PerformanceMonitorManager screen.

Performance monitoring is now enabled. When a dataflow runs, the performance statistics will display at the top of the PerformanceMonitorManager screen. Note the following:

• You must refresh the screen to see updates.
• To reset the counters, click the **Invoke** button next to **reset**.
• If you stop the Spectrum™ Technology Platform server, performance monitoring will be turned off. You will have to turn it back on when you start the server again.

Monitoring File Handle Caching Statistics with the JMX Console

The JMX console is browser-based tool that monitors performance and records statistics, including file handle caching statistics for native TAB and shapefiles.

1. Open a web browser and go to `http://server:port/jmx-console`

   Where:

   server is the IP address or hostname of your Spectrum™ Technology Platform server.

   port is the HTTP port used by Spectrum™ Technology Platform. The default is 8080.

2. Log in using the admin account.
3. Under " Domain: Spatial", click **Spatial:name=TABFileHandlePool,type=Remote Component** or **Spatial:name=ShapeFileHandlePool,type=Remote Component** to view the file handle caching statistics for TAB files or shapefiles.

   > **Note:** You can also disable the file handle cache or clear it on this page without needing to restart the server.

4. Click **All MBeans** to return to the main JMX console.

Monitoring Memory Usage

The JMX Console allows you to monitor the JVM heap usage of the spatial remote component.

Memory usage (HeapMemoryUsage and NonHeapMemoryUsage) is based on the standard JVM memory MBean. It shows the memory usage of the JVM that the remote component running on. It includes the amount of init, max, committed and used memory.

RuntimeName includes the process ID that you can use to find more information from the operating system (for example, by using the Windows Task Manager), or even kill the process.

In the heap sections, `={committed=143130624, init=134217728, max=1908932608, used=23483928}` are shown in bytes.

Init is the initial amount JVM allocated (-Xms); max is the one specified by –Xmx. Used is the amount of memory that used by JVM for objects. The relationship is like this: –Xms < committed < -Xmx, and used < committed.

You can modify the heap memory by modifying the -Xm in the java.vmargs file under the spatial folder (<Installed>\Pitney Bowes\Spectrum\server\modules\spatial\java.vmargs). See **Increasing Heap Memory for more instructions.**

5 - Performance Tuning

This section describes approaches for improving performance by managing memory and threading, and also relates best practices for optimizing the performance of the Location Intelligence Module. It is intended for experienced administrators.

Spectrum provides several tuning options to optimize performance of the server. The optimal selection of settings is dependent on the nature of the deployment. To create a well-tuned server environment, it is recommended that performance tests should be executed in the deployed environment to determine optimal settings. This section provides some general guidance on performance tuning.

In this section

Remote Component Configuration

All spatial services in the Spectrum™ Technology Platform are deployed into a remote component (JVM instance) that is separate from the platform runtime. This ensures the platform is independent of the modules within it and that JVM configuration can be applied to the spatial services, allowing flexibility of memory allocation and tuning for performance based on the characteristics of those services.

The remote component supplies spatial functions to spatial services (such as the Feature Service and Mapping Service) and stages (such as the Spatial Calculator and Query Spatial Data). The pool size for a remote component is the number of requests the component can handle concurrently. This affects the throughput of both spatial services and spatial stages.

To manage permissions for the spatial remote component, use the Management Console as you would for any other secured entity type. The spatial remote component is listed as the "Spatial Component" secured entity type under the **Databases Resources** group. You can set permissions for the spatial remote component when creating or editing roles or by using access control settings. See **Managing Security** on page 16 for more information.

Modifying the Pool Size

In addition to JVM tuning, you can also adjust the pool size of the spatial remote component. The pool size for a remote component is the number of requests the component can handle concurrently. This setting represents the number of threads on the components that are listening for service requests from the Spectrum™ Technology Platform or executing a Location Intelligence Module stage (that is, the maximum number of managed connections).

Every web service request enters Spectrum from the platform and is passed to the component. The default value of 1 can be increased to accommodate greater request loads. A pool size that matches the number of CPUs is recommended. The maximum setting should not go above twice the number of the CPU core; for example, on a 4 CPU machine the combined number of threads for all services should not exceed 8. Performance tests should be run with various settings until optimal performance is achieved for the usage.

You have the ability to adjust the pool size in Management Console for the spatial remote component:

1. Open the Management Console.
2. Go to **Resources** > **Location Intelligence**.
3. Change the pool size for the remote component using the arrows or by typing in a value. The minimum value is 1 and the maximum value is 64.
4. Click **Save**.

5. If you decreased the pool size, restart the server. Increasing the pool size takes effect immediately and does not require a server restart.

Data Source Pooling Configuration

The `pooling-datasource-factory.properties` file (located under `\server\modules\spatial`) may be used to configure the pooling of connections used by JDBC-based data sources (such as Oracle and SQL Server) to optimize performance.

In most cases, we recommend enabling the validator class. This allows objects to be validated before being borrowed from the pool. If the validation fails, the connection will be dropped from the pool and an attempt will be made to borrow another. A validation query is also available for special cases, such as when using a custom data provider. If both the validation query and the validator class are enabled, the validator class will be used.

Enabling validation may have a slight negative performance impact; however, the test query maintains the integrity of all the connections in the connection pool in cases where communication between Spectrum Spatial and an external database is not reliable. Set a validation interval to mitigate the performance impact of validation. If a connection is due for validation but has been validated previously within this interval, it will not be validated again.

Improving Performance for Distance-Based Operations

A PGD index file is a supplemental file to the TAB file set that can make performance for native, native extended (NativeX), and seamless TABs comparable to that of GSB files. The PGD Builder, a command-line utility, is available to generate these specialized index files to improve performance of certain distance-based operations for native datasets containing lines and polygons. An index built using the PGD Builder is helpful when the data you are searching is based on lines and regions and you are using:

• the Point in Polygon stage, when you are including distance
• the Find Nearest stage, when the input is a point (whether or not you are including distance)
• SearchNearest operations in the Feature Service, with an input point and a line or polygon search table

The PGD Builder utility can be downloaded from the Spectrum Spatial section of the Welcome Page, under **PGD Builder** on the Utilities tab. A link to the is also available on the Welcome Page next to the download link for the utility.

Note: A PGD file is 5-6 times larger than the .MAP file for the TAB. One PGD file is generated per TAB file, except in the case of a seamless TAB which will have PGD files created for each sub-TAB.

Also, a PGD file will no longer be used by the system if you change the data in the TAB (that is, if rows have been added or deleted or a geometry has been changed in the MAP portion of the TAB). If warnings are enabled (see **Spatial Logging** on page 63), a message about the out-of-date PGD file will appear in the `wrapper.log` or if applicable, in the log file that has been configured for Spatial logging. You must then regenerate the PGD for the updated TAB file.

# 6 - Managing a Cluster

In this section

Clustered Architecture for the Location Intelligence Module

In a clustered environment, processing is shared among two or more instances of the server. The diagram below illustrates the deployment architecture of such a configuration. Load balancing can be used to support high availability and scaling. The deployment architecture includes a load balancer, a Spectrum Spatial cluster, a database, and a file share. With this approach it is possible to scale both horizontally and vertically. You can cluster the Location Intelligence Module with or without platform clustering.

> **Note:** Setting up both a Spectrum™ Technology Platform cluster and a Location Intelligence Module cluster is recommended and has several benefits:
>
> • Security (ACL) synchronization happens automatically for named resources .
> • Dataflows, users, and roles created on one node will automatically synchronize to all nodes.
> • All Location Intelligence Module demo pages and utilities (such as Spatial Manager) can and should point to the load balancer.
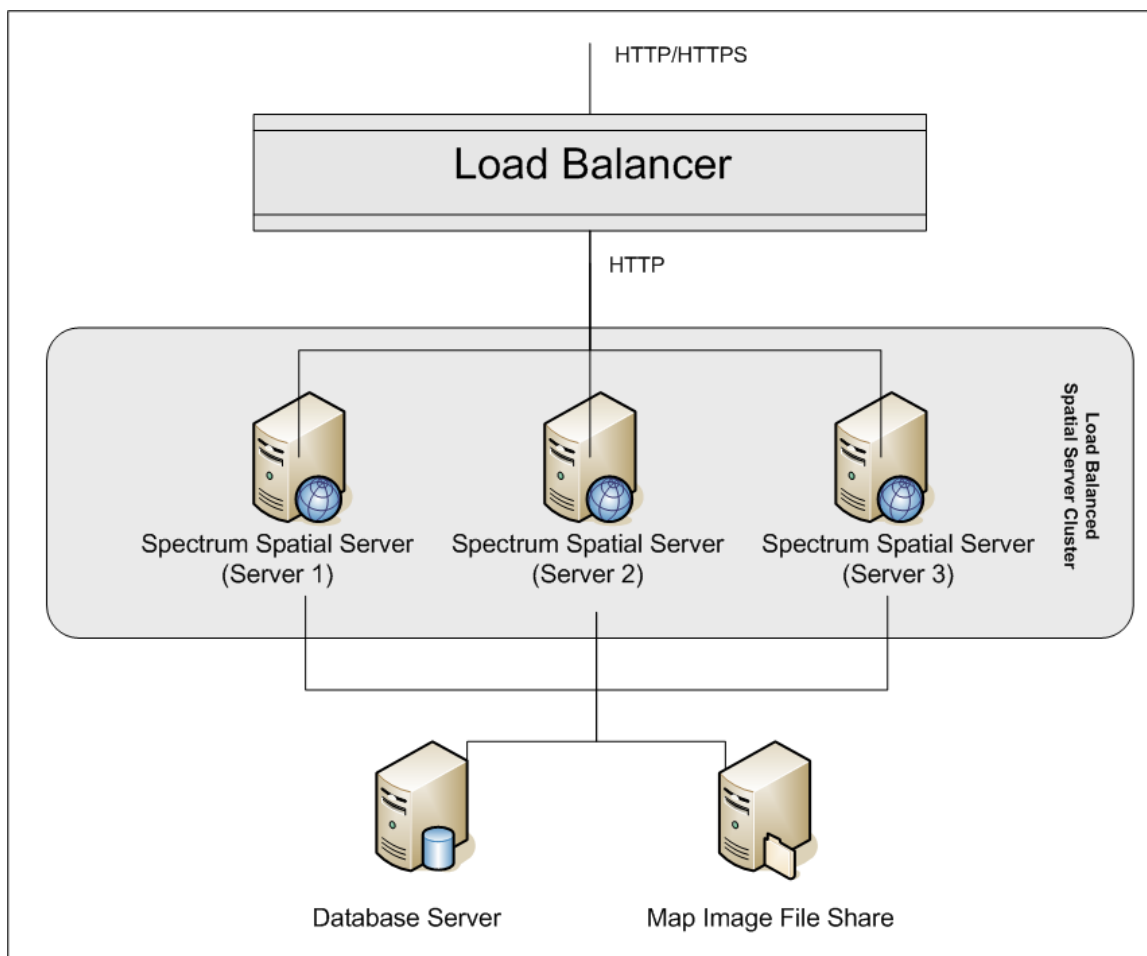
Load Balancer

The load balancer spreads requests between the Spectrum Spatial instances. Any load balancer that supports load balancing HTTP/HTTPs requests can be used.

Spectrum Spatial Cluster

The cluster is a collection of Spectrum instances with LIM sharing administration, named resources, geographical metadata content and configuration settings. Additional nodes can be added to the cluster for resilience or to deliver support for greater loads. Each node can be scaled vertically through additional hardware resources and/or additional instances should this be required for hardware with massive resources. Spectrum can be configured to use restricted numbers of CPUs.

Database

Spectrum stores named resources (maps, layers, tables and styles), geographic metadata and configuration in a repository. In the default single server installation an embedded database is used to store these resources on the local server. To create a resilient scalable solution this embedded database should be replaced with a resilient independent database. Oracle, PostGreSQL/PostGIS and Microsoft SQL Server are the supported repository databases.

In the load balanced configuration, Spectrum nodes cache these resources in a local cache and search index in each node in the cluster. When a Spectrum node receives a request it uses the local cache and index to find resources. Named resources can be added through any node in the cluster. Each node keeps its cache current by checking for differences between its local cache and the central database. This check occurs every 2 seconds by default. Time frequency can be configured. This architecture ensures the server delivers high performance transactions and the load on the repository database is kept to a minimum. If a new Spectrum node is added to the cluster the cache and index are created automatically. Such a scenario can occur to remedy a node failure or grow the capability of the deployment.

File Share

The file share provides a folder to hold map images generated by Spectrum. When maps are rendered using the web services the server supports the map images being returned through URLs or returned as a base 64 encoded image. When a URL is returned the map image is stored as a file and served on request of the URL. To ensure any Spectrum node can return the map image a file share is used to store the images.

Using Enterprise Designer with a Cluster

1. Launch Enterprise Designer.
2. In the **Server name** field, enter the server name of the load balancer.
3. In the **Port** field, enter the port that you have configured the load balancer to listen on.

   Note: Input files, output files and database resources must be on a shared drive, or file server, or some commonly-accessible location. Otherwise, all files must be loaded on each server that hosts a Spectrum™ Technology Platform server and must be located in the same path.

Once you have logged in you can use Enterprise Designer as normal. The actions you take will apply to all Spectrum™ Technology Platform instances in the cluster where you are logged in.

Starting a Cluster

If all the nodes in a cluster are stopped, you must follow this procedure to start the cluster safely and avoid data loss.

1. On the last node that was stopped last, remove the seed nodes and start the server.

Warning: The first node that you start must be the last node that was stopped, and that node must be a seed node. Starting another node first may result in loss of data such as job history and configuration settings. If you do not know which node was stopped last, look in each node's wrapper log for the time stamp of the shutdown message. You can find the wrapper log in: *Spectrum Location*\server\app\repository\logs\wrapper.log.

a) Open this file in a text editor:

```
server/app/conf/spectrum-container.properties
```

b) In the `spectrum.cluster.seeds` property, remove all host names and IP addresses except for the one for this server. Save the host names and IP addresses so that you can re-add them later.

c) Save the file.

d) Start the server.

e) Wait for the Spectrum™ Technology Platform server to *completely* start.

You can tell when the Spectrum™ Technology Platform server has completely started by looking in the wrapper log: *Spectrum Location*\server\app\repository\logs\wrapper.log. This message is displayed when the server is completely started:

```
Pitney Bowes Spectrum(TM) Technology Platform (Version Version
Number) Started.
```

f) In the properties file `spectrum-container.properties`, in the `spectrum.cluster.seeds` property, add the host names or IP addresses that you had removed, separating each with a comma.

g) Save and close the file. You do not need to restart the server.

2. Start the other nodes in the cluster.

Warning: Be sure to wait for the first node to start *completely* before starting additional nodes. Starting additional nodes before the first one is started may result in loss of data.

Stopping a Cluster

To stop an entire cluster:

1. Identify which nodes are seed nodes. To do this, open the file *SpectrumFolder*/server/app/conf/spectrum-container.properties and look at the nodes listed in the `spectrum.cluser.seeds` property.

2. Stop each Spectrum™ Technology Platform server in the cluster, making sure that the last node you stop is a seed node. Change the working directory to the Spectrum™ Technology Platform server's `bin` directory, source the setup file, then type the following command: `./server.stop.`

   **Warning:** To prevent loss of data, the last node you stop must be a seed node.

3. Make a note of the last node you stopped. You will need this information when starting up the cluster. Right-click the Spectrum™ Technology Platform icon in the Windows system tray and select **Stop Spectrum**™.

   Warning: To prevent loss of data when starting the cluster, the first node you start must be the last node that was stopped, and that node must be a seed node.

Removing a Node from a Cluster

To remove a node from a cluster, stop the Spectrum™ Technology Platform server.

1. To stop the server, right-click the Spectrum™ Technology Platform icon in the Windows system tray (shown below) and select **Stop Spectrum**™.
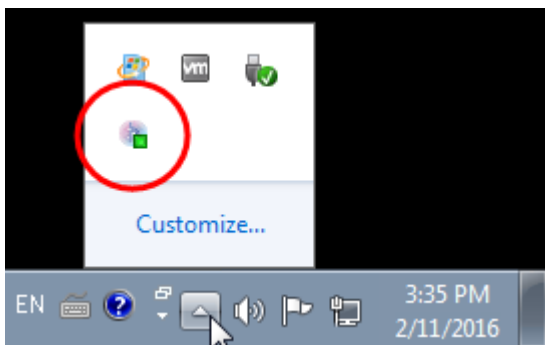
2. Stop the node you want to remove:

   change the working directory to the Spectrum™ Technology Platform server's `bin` directory, source the setup file, then type the following command: `./server.stop` .

   On Windows, right-click the Spectrum™ Technology Platform icon in the system tray and select **Stop Spectrum**™.

3. Open the file `server/app/conf/spectrum-container.properties` in a text editor and set `spectrum.cluster.enabled` to `false`.

4. On each of the other nodes in the cluster, open the `spectrum-container.properties` file and remove the node from the `spectrum.cluster.seeds` property.

For Location Intelligence Module users: If you want to keep the node standalone and able to run outside the cluster, copy back the original `repository.xml` file and remove the following folders from the `/server/modules/spatial/jackrabbit` directory for each instance of Spectrum™

Technology Platform: repository, version, workspaces. Restart the server and import the repository content.

Managing a Cluster for the Location Intelligence Module

Setting Up a Common Repository Database

You must configure the Location Intelligence Module to use a common repository database for the cluster. This ensures that named resources, geographic metadata and configuration settings are managed across the cluster.

The repository is installed with a set of named resources, geographic metadata and configuration files. To migrate these resources to the common database repository the resources need to be exported from the default internal repository database and reimported into the new shared repository database.

For bulk export and import of repository content, use the `limrepo import` and `limrepo export` commands in the Administration Utility. These commands give you the option of preserving permissions (see the Administration section of the *Spectrum Spatial Guide* for instructions.)

These steps describe how to set up your repository on a common database, either PostgreSQL, Oracle, or Microsoft SQL Server:

1. Export all repository resources to a local folder using the `limrepo export` command in the Administration Utility (see the Administration section of the *Spectrum Spatial Guide* for instructions).

   The contents of the installed repository must be exported. This step only needs to be performed once, as the contents of the repository should be the same at this point for all instances of Spectrum™ Technology Platform.

2. Stop the Spectrum™ Technology Platform server on all nodes (for instructions, see **Stopping a Cluster** on page 78.)

3. On all nodes of Spectrum™ Technology Platform modify the configuration to specify the common database.

   a) Copy the contents of `repository.<databaseType>.xml` to `repository.xml` located under the `server/modules/spatial/jackrabbit` folder where `<databaseType>` is the appropriate type for your database (postgres, oracle, or mssql).

   b) In `repository.xml`:

   • Modify the DataSource section with the server host name, port, database, user, and password.

- Modify the Cluster section to assign a distinct cluster ID, like Node1. Ensure unique IDs are assigned to every subsequent node in the cluster (for example, Node2, Node3).
- Save the changes to `repository.xml`.

c) Remove these folders from the `/server/modules/spatial/jackrabbit` folder: repository, version, workspaces.

4. If your database has previously contained any repository content, you must remove these tables to create a clean repository:

- default_binval
- default_bundle
- default_names
- default_refs
- rep_fsentry
- rep_global_revision
- rep_journal
- rep_local_revisions
- security_binval
- security_bundle
- security_names
- security_refs
- version_binval
- version_bundle
- version_names
- version_refs

If using Oracle, then also delete `version_seq_names_id`, `security_seq_names_id`, and `default_seq_names_id`.

5. On the seed node only, import the backed up repository content.

a) Start the Spectrum™ Technology Platform server (for instructions, see **Starting a Cluster** on page 77).

b) Import the contents using the `limrepo import` command, pointing to the seed node.

6. Start the remaining nodes in the cluster (for instructions, see **Starting a Cluster** on page 77).

Configuring Your System

Once the Spectrum™ Technology Platform is installed and you have configure a common repository, you need to configure your instance before you can replicate it to another virtual machine. If you are not using a virtual machine environment, you will need to perform these steps on each of your Spectrum™ Technology Platform installations.

Configure the Map File Share

To configure the map file share (a shared image folder) to Spectrum™ Technology Platform, you first need a shared map image directory.

> **Note:** To create a Unix/Linux map file share, see **Creating a Map Image File Share on Unix/Linux** on page 83.

> **Note:** To create a Windows map file share, see **Creating a Map Image File Share on Windows** on page 83.

Once a map image directory has been created, configure the map file share:

1. Modify the Mapping Service configuration by pointing to a shared image folder and load balance server. In the ImageCache change the Directory parameter to a common image directory, and change the `AccessBaseURL` parameter to the load balancer machine image URL.

   If you are using a virtual machine environment, remember this IP address, as you must set the load balancer VM to this IP address.

   For Unix/Linux installations:

   ```
   <ImageCache>
   <Directory>/<spatial server
   root>/server/modules/spatial/images</Directory>
   <AccessBaseURL>http://<loadbalance_IP_address>/rest/Spatial/
   MappingService/internal/imageCache</AccessBaseURL>
    <FileExpire>30</FileExpire>
    <ScanInterval>30</ScanInterval>
   </ImageCache>
   ```

   For Windows installations:

   ```
   <ImageCache>
   <Directory>\\server\Share\images</Directory>
   <AccessBaseURL>http://<loadbalance_IP_address>/rest/Spatial/MappingService/
   internal/imageCache
   </AccessBaseURL>
    <FileExpire>30</FileExpire>
    <ScanInterval>30</ScanInterval>
   </ImageCache>
   ```

2. For Unix/Linux installations, you must set up a symbolic link to enable map images to go to the shared file system.

   Create an `images` subfolder in the mounted share folder, e.g., `/mnt/<linux mount>/images`

   ```
   cd /<spatial server root>/server/modules/spatial
   rm -Rf images
   ln -s /mnt/<linux mount>/images ./images
   ```

Creating a Map Image File Share on Unix/Linux

The file share provides a folder to hold map images generated by Spectrum Spatial. Create a shared folder accessible to all Spectrum nodes. The file share is not required if maps are returned from the web services as Base64-encoded images.

To create a map image file share on Unix/Linux:

1. Mount a shared folder on each operating system hosting Spectrum. The commands below mount a drive on a Microsoft Windows Server or network drive supporting CIFS.

```
mkdir /mnt/<linux mount>
mount -t cifs //<windows host>/<windows share> /mnt/<linux mount>-o
username=shareuser,password=sharepassword,domain=pbi
```

2. Set the image share to load at startup in `/etc/fstab`.

```
//<windows ip address for share>/share /path_to/mount cifs
username=server_user,password=secret,_netdev 0 0
```

Creating a Map Image File Share on Windows

The file share provides a folder to hold map images generated by Spectrum Spatial. Create a shared folder accessible to all Spectrum nodes. The file share is not required if maps are returned from the web services as Base64-encoded images.

To create a map image file share on Windows:

1. In Windows Explorer, select the image folder you want to share.
2. Right-click, and then click **Share** or **Share with**.
3. Select the users who will be writing to the image folder. These users must have read/write privileges.

Modifying OGC Service Configurations for Clustering

To ensure clustering works when you have both a Spectrum™ Technology Platform cluster and a Location Intelligence Module cluster, changes are required to the Open Geospatial Consortium (OGC) services configuration files using Spatial Manager: From the WFS, WMS, and WMTS settings pages, change the online resource (service) URL to the IP address and port of the load balancer. See the *Spatial Manager Guide* in the Utilities section of the *Spectrum Spatial Guide* for more information.

Modifying the Java Properties Files in All Nodes

You must change the java property file in all nodes of the cluster. To modify the java properties for Spectrum™ Technology Platform:

1. Modify the java.properties file, located in
   `<spectrum>/server/modules/spatial/java.properties`, to point repository.host to
   localhost.
2. Change the images.webapp.url and all of the service host and port numbers to point to the load
   balance server.

Configuring Ports for Multiple Spectrum Instances

If you have multiple Spectrum™ Technology Platform instances on a single machine, you must
change the port numbers for each instance. Change all ports in `<Spectrum
root>/server/app/conf/spectrum-container.properties` to new port values that are
not in use. The http port reflects the port number entered in the installer.

Shared Spectrum Local Data

If you are using TAB file data on the file system, this data needs to be in a shared location accessible
by all instances of Spectrum in the load balanced environment. It is also important to note that all
named resources in the repository accessing data on the file system should point to this shared
location.

Each VM or machine hosting Spectrum needs to have access to the mounted shared drive.

> **Note:** Using named resources that point to database tables do not require a shared drive,
> as the named resources in the repository do not access the data using a file path; rather they
> use a named connection to the data in the database.

7 - Using the Administration Utility

In this section

Getting Started with the Administration Utility

The Administration Utility provides command line access to administrative functions. You can use it in a script, allowing you to automate certain administrative tasks. You can also use it interactively. Not all administrative functions are available in the Administration Utility. Use Management Console to access the functions that are not available in the Administration Utility.

> **Note:** The Administration Utility requires Java 8 or later. Verify that you are referencing a supported version of Java, in the system's path, before running the Administration Utility.

1. Open a web browser and go to the Spectrum™ Technology Platform Welcome Page at:

   http://<servername>:<port>

   For example, if you installed Spectrum™ Technology Platform on a computer named "myspectrumplatform" and it is using the default HTTP port 8080, you would go to:

   http://myspectrumplatform:8080

2. Click **Platform Client Tools**.
3. Click **Command Line Clients**.
4. Under **Administration Utility**, click **Download** and download the zip file to the computer where you want to use the Administration Utility.
5. Extract the contents of the zip file.
6. To launch the command line interface, do one of the following:

   - If you are running the server on a Unix or Linux system, execute `cli.sh`.
   - If you are running the server on a Windows system, execute `cli.cmd`.

     > **Note:** If necessary, modify the `.sh` or `.cmd` file to use the path to your Java installation.

7. Connect to the Spectrum™ Technology Platform server by typing this command:

   `connect` --h *servername*:*port* --u *username* --p *password* --s *SSLTrueOrFalse*

   For example,

   `connect` --h myserver:8080 --u admin --p myPassword1 --s true

8. Once you are connected you can run commands. Some tips:

   - For a list of available commands, type `help` or press the tab key.
   - To auto-complete a command, type the first few characters then press the tab key. For example, typing `us` then pressing the tab key automatically completes the command `user`. Pressing the tab key again will display a list of all the `user` commands.
   - If you specify an option value that contains a space, enclose the value in double quotes.

9. When you are done, type `exit` to exit the Administration Utility.

Using a Script with the Administration Utility

The Administration Utility can execute a series of commands from a script file. This is useful if you want to automate or standardize administrative actions through the use of a script instead of manually executing commands through the Administration Utility or by using the Management Console.

1. Using a text editor, create a script file. A script file contains the commands that you want to execute.

   To add a command to a script file, type the command and the necessary parameters as you would if you were entering the command at the command prompt. Enter one command per line.

   To insert comments into a script file, use the following notation:

`/*`	Indicates the start of a block comment.
`*/`	Indicates the end of a block comment.
`//`	Indicates an inline comment. Use at the start of a line only.
`;`	Indicates an inline comment. Use at the start of a line only.

2. Save the script either on the computer where you run the Administration Utility or in a location that is accessible from the computer where you run the Administration Utility. You can use any file name and extension you choose. The recommend file extension is `.cli`.

3. To execute the script, do one of the following:

Option	Description
To execute the script at the command line	Specify the following at the command line or in a batch or shell script: `cli.cmd --cmdfile ScriptFile`
To execute the script form the Administration Utility	Open the Administration Utility and connect to the Spectrum™ Technology Platform server using the `connect` command. Then, use the `script` command to execute the script. For more information on this command, see **script**.

Example: Moving Dataflows from Staging to Production

You have three dataflows: Deduplication, AddressValidation, and DrivingDirections.
You have a staging server where you make changes to these dataflows and test
them, and a production environment where the dataflows are made available for
execution. You want to have a consistent and automated way to move these dataflows
from your staging server to your production server so you decide to use an
Administration Utility script to accomplish this. The script might look like this:

```
// Connect to the staging server
connect --h stagingserver:8080 --u allan12 --p something123

// Export from staging
dataflow export --d "Deduplication" --e true --o exported
dataflow export --d "AddressValidation" --e true --o exported
dataflow export --d "DrivingDirections" --e true --o exported

// Close connection to the staging server
close

// Connect to the production server
connect --h productionserver:8080 --u allan12 --p something123

// Import to production
dataflow import --f exported\Deduplication.df
dataflow import --f exported\AddressValidation.df
dataflow import --f exported\DrivingDirections.df

// Close the connection to the production server
close
```

Location Intelligence Module

limrepo export

Note: For instructions on installing and running the Administration Utility, see **Getting Started with the Administration Utility** on page 86.

The `limrepo export` command exports named resources (such as named tables) from the
Spectrum Spatial repository to a local file system. You must have the Location Intelligence Module
installed to use this command.

Resources are exported with their full repository paths in the target folder. For example, if you run `limrepo export` --s /Samples/NamedTables --o C:\export, the tool creates `C:\export\Samples\NamedTables\WorldTable`, and so on for each named table under the NamedTables folder or directory.

> **Note:** The `limrepo export` command will always recursively export all folders, including empty ones.

Usage

`limrepo export` --s *SourceRepositoryPath* --o *OutputFilePath*

> **Note:** To see a list of parameters, type `help limrepo export`.

Required	Argument	Description
Yes	`--s` *SourceRepositoryPath*	Specifies the path to the resource or a folder to be exported.
Yes	`--o` *OutputFilePath*	Specifies the path to a folder on the local file system where you want to export. This can be a new folder or an existing folder; however, an existing folder must be empty otherwise the export will fail.
No	`--q` or `--quiet`	Disables the display of the resources copied during the export; that is, operates in quiet mode. If the flag is specified, the default value is true. If the flag is not specified, the default value is false.
No	`--f` or `--fullpaths`	Prints the full source and output paths. If the flag is specified, the default value is true. If the flag is not specified, the default value is false.
No	`--r` or `--recursive`	Recursively exports subfolders (children of the specified source). If the flag is specified, the default value is true. If the flag is not specified, the default value is true.
No	`--c` or `--continueonerror`	Continues with the export if an error occurs. If the flag is specified, the default value is true. If the flag is not specified, the default value is false.
No	`--a` or `--acl`	Preserves existing permissions for the exported resources in the export folder on the local file system. An access control list (ACL) indicates the operations each user or role can perform on a

Required	Argument	Description
		named resource, such as create, view, edit, or delete.
		If the flag is specified, the default value is true. If the flag is not specified, the default value is false.

Example

This example exports the named resources in the repository's \Samples folder to C:\myrepository\samples on your local file system.

```
limrepo export --s /Samples --o C:\myrepository\samples
```

---

limrepo import

> **Note:** For instructions on installing and running the Administration Utility, see **Getting Started with the Administration Utility** on page 86.

The `limrepo import` command imports named resources (such as named tables) from a local file system into the Spectrum Spatial repository. You must have the Location Intelligence Module installed to use this command.

When importing, you must point to the same folder or directory you exported to previously. For example, if you run `limrepo export` --s /Samples/NamedTables --o C:\export, the tool creates `C:\export\Samples\NamedTables\WorldTable`, and so on for each named table under the NamedTables folder or directory. Resources are exported with their full repository paths in the target folder. Running `limrepo import` --s C:\export then imports WorldTable back to `/Samples/NamedTables/WorldTable`.

> **Note:** The `limrepo import` command will always recursively import all folders, including empty ones.

After performing an import, in many cases, you will need to adjust the named connections to point to their new path using Spatial Manager. For example, if your Native TAB files were installed on "C:\myfiles" in your test instance and the same files are installed on "E:\ApplicationData\Spectrum\Spatial\Spring2016" then that connection would have to be corrected in Spatial Manager after import. See the Utilities section of the *Spectrum Spatial Guide* for instructions on using Spatial Manager to edit a named connection.

> **Note:** If you are using `limrepo import` to restore service configuration files that you exported from a pre-12.0 version of Spectrum™ Technology Platform, the files will automatically be modified to be compliant with version 12.0 and later (for example, the repository URLs will be removed).

Usage

`limrepo import --s` *SourceFilePath*

> **Note:** To see a list of parameters, type `help limrepo import`.

Required	Argument	Description
Yes	`--s` *SourceFilePath*	Specifies the path to the resource or a folder on the local file system that is to be imported. This must be the root folder of a previous export on the local file system.
No	`--q` or `--quiet`	Disables the display of the resources copied during the import; that is, operates in quiet mode.
		If the flag is specified, the default value is true. If the flag is not specified, the default value is false.
No	`--u` or `--update`	Specifies whether to overwrite existing resources if resources with the same name are already on the server.
		true If there is a resource on the server with the same name as a resource you are importing, the resource on the server will be overwritten. This is the default setting if the flag is not specified or if the flag is specified without a value.
		false If there is a resource on the server with the same name as a resource you are importing, the resource will not be imported.
No	`--f` or `--fullpaths`	Prints the full source and output paths.
		If the flag is specified, the default value is true. If the flag is not specified, the default value is false.
No	`--c` or `--continueonerror`	Continues with the import if an error occurs.
		If the flag is specified, the default value is true. If the flag is not specified, the default value is false.
No	`--a` or `--acl`	Preserves any previously exported permissions and merges them with existing permissions when importing resources. An access control list (ACL) indicates the operations each user or role can perform on a named resource, such as create, view, modify, or delete.

Required Argument	Description
	For example, a user has read and write permissions on a resource when exporting. If the user only has read permissions on the resource when importing, write permission will be granted again after the import finishes successfully.
	Conflicting permissions cannot be merged and will be ignored. ACL entries for users and roles that do not exist in the target repository are also ignored.
	If the flag is specified, the default value is true. If the flag is not specified, the default value is false.
	Tip: When using this flag, the user on the server you exported from should also exist on the server to which you are importing. For example, you have "testuser" with access control settings and export the resources with ACL from one server, then import those named resources to another server that does not have "testuser". In this case, named resources will be uploaded but not the ACL.

Example

This example imports the named resources from C:\myrepository\samples on your local file system.

```
limrepo import --s C:\myrepository\samples
```

limrepo mwsimport

Note: For instructions on installing and running the Administration Utility, see **Getting Started with the Administration Utility** on page 86.

The `limrepo mwsimport` command in the Spectrum™ Technology Platform Administration Utility allows you to provision a map from a MapInfo Workspace (MWS) file that has been created either by MapInfo Pro or the MapXtreme Workspace Manager into the Spectrum Spatial repository. The import will create the named map and all its dependent resources (layers, tables and connections). The connection is named by appending 'Connection' to the map name. The named tables and named layers are created in subfolders (NamedTables and NamedLayers, respectively).

You must have the Location Intelligence Module installed to use this command.

Usage

```
limrepo mwsimport --s MWSFilePath --o Output --p ServerPath
```

> **Note:** To see a list of parameters, type `help limrepo mwsimport`.

Required	Argument	Description
Yes	`--s` *MWSFilePath*	Specifies the path to an MWS file on the local file system that is to be imported.
Yes	`--o` *Output*	Specifies the path to the named map on the repository. All resources will be created within the same folder as the named map.
Yes	`--p` *ServerPath*	Specifies the file path to the location of the data on the server. This path is used to create a named connection which is then referenced by all the named tables that are created. These tables will use file paths relative to that named connection.
No	`--l` *LocalPath*	Specifies the file path to the location of the data on the local file system, if the MWS contains file paths that do not exist on the server file system. Any occurrences of the specified value in the MWS file will be substituted with the specified server path. If you have partial paths in the MWS file, this is not required; this is usually the case with anything created from MapXtreme.

Example

This example imports an MWS file on the D: drive (where the data on the server exists at C:\mydata) and provisions the named resources into /Europe/Countries in the repository.

```
limrepo mwsimport --s D:\europe.mws --o /Europe/Countries --p
C:\mydata
```

Result

The following named resources are created:

/Europe/Countries/Europe *(named map)*
/Europe/Countries/EuropeConnection *(named connection)*
/Europe/Countries/NamedTables/austria *(named table)*
/Europe/Countries/NamedTables/belgium *(named table)*
.
/Europe/Countries/NamedLayers/austria *(named layer)*

/Europe/Countries/NamedLayers/belgium *(named layer)*
..

Enterprise Routing Module

ermdb list

Note: For instructions on installing and running the Administration Utility, see **Getting Started with the Administration Utility** on page 86.

The `ermdb list` command retrieves a list of all the existing routing database resource on the server. You must have the Enterprise Routing Module installed to use this command.

Usage
```
ermdb list
```

Example

This example returns all the database resources on the server.

```
ermdb list
```

---

ermdb get

Note: For instructions on installing and running the Administration Utility, see **Getting Started with the Administration Utility** on page 86.

The `ermdb get` command allows you to return information on the routing databases configured on the server. Information returned is the name of the database, location of the database on the file system (path), and the pool size configured for the database. You must have the Enterprise Routing Module installed to use this command.

Usage
```
ermdb get --name database_name
```
Note: To see a list of parameters, type `help ermdb get`.

Required	Argument	Description
Yes	`--name` or `--n` *database_name*	Specifies the name of the database resource to return information. The name must be a unique name on the server. For a list of existing routing database resources, use the `ermdb list` command.

> **Example**
>
> This example returns the information for the database resources US from the server.
>
> `ermdb get --name US`

ermdb add

Note: For instructions on installing and running the Administration Utility, see **Getting Started with the Administration Utility** on page 86.

The `ermdb add` command creates a new routing database resource on the server. You must have the Enterprise Routing Module installed to use this command.

Note: The `ermdb add` command requires a unique name be used for each of the databases being added.

Usage

`ermdb add --name` *database_name* `--poolsize` *pool_size* `--path` *database_path*

Note: To see a list of parameters, type `help ermdb add`.

Required	Argument	Description
Yes	`--name` or `--n` *database_name*	Specifies the name of the database resource to be added. The name must be a unique name on the server. For a list of existing routing database resources, use the `ermdb list` command.
No	`--poolsize` or `--s` *pool_size*	Indicates the maximum number of concurrent requests the database should handle. The default if not specified is 4. The accepted range for concurrent requests is any integer between 1 and 128.
YES	`--path` *database_path*	Specifies the location of the routing database on the file server.

> **Example**
>
> This example adds the database resources US from
> `E:/ERM-US/2014.09/driving/south` into the server.
>
> `ermdb add --name US --poolsize 10 --path`
> `E:/ERM-US/2014.09/driving/south`

ermdb delete

> **Note:** For instructions on installing and running the Administration Utility, see **Getting Started with the Administration Utility** on page 86.

The `ermdb delete` command removes an existing routing database resource from the server. You must have the Enterprise Routing Module installed to use this command.

Usage
`ermdb delete --name` *database_name*

> **Note:** To see a list of parameters, type `help ermdb delete`.

Required	Argument	Description
Yes	`--name` or `--n` *database_name*	Specifies the name of the database resource to be deleted. For a list of existing routing database resources, use the `ermdb list` command.

> **Example**
>
> This example removes the database resources US from the server.
>
> `ermdb delete --name US`

ermdb modify

> **Note:** For instructions on installing and running the Administration Utility, see **Getting Started with the Administration Utility** on page 86.

The `ermdb modify` command changes an existing routing database resource on the server. You must have the Enterprise Routing Module installed to use this command.

Usage

`ermdb modify --name` *database_name* `--poolsize` *pool_size* `--path` *database_path*

> **Note:** To see a list of parameters, type `help ermdb modify`.

Required	Argument	Description
Yes	`--name` or `--n` *database_name*	Specifies the name of the database resource to be modified. For a list of existing routing database resources, use the `ermdb list` command.
No	`--poolsize` or `--s` *pool_size*	Indicates the maximum number of concurrent requests the database should handle. The accepted range for concurrent requests is any integer between 1 and 128. You must specify either a new pool size or a new database path.
No	`--path` *database_path*	Specifies the new location of the routing database on the file server. You must specify either a new pool size or a new database path.

Example

This example modifies both the pool size and the database path for a new vintage.

```
ermdb modify --name US --poolsize 20 --path
E:/ERM-US/2015.03/driving/south
```

---

ermdb import

> **Note:** For instructions on installing and running the Administration Utility, see **Getting Started with the Administration Utility** on page 86.

The `ermdb import` command allows you to import a file consisting of routing database configurations and creates the database resources on the server. You can either create the import file, or use the file created by the `ermdb export` command. You must have the Enterprise Routing Module installed to use this command.

The import file format is as follows:

[{ "product": "Spatial", "module": "routing", "name": "US", "maxActive": 4, "properties": { "DatasetPaths": "E:/ERM-US/2014.09/driving/northeast" } }]

Where `product` and `module` must be Spatial and routing, `name` is the name of the database, `maxActive` is the maximum number of concurrent requests you want this database to handle (or the pool size), and `DatasetPaths` is the path to the data sets for the database resource.

You can add multiple databases in an import file (duplicate the example above), and add multiple datasets for each database resource separating them using semi colons.

Note: If you want to specify UTF-8 characters in import file, you must add the JVM parameter file.encoding to the value UTF-8 in the startup of the CLI command prompt. E.g., -Dfile.encoding=UTF-8

Usage

```
ermdb import --file file_name
```

Note: To see a list of parameters, type `help ermdb import`.

Required	Argument	Description
YES	`--file` or `--f` *file_name*	Specifies the directory and name of the import file.

Example

This example imports two databases US1 and US2 each consisting of multiple datasets.

```
ermdb import --file E:/ERM-US/export/ermDbResource.txt
```

The input file is defined as the following:

[{ "product": "Spatial", "module": "routing", "name": "US1", "maxActive": 4, "properties": { "DatasetPaths": "E:/ERM-US/2014.09/driving/northeast;E:/ERM-US/2014.09/driving/south" } }, { "product": "Spatial", "module": "routing", "name": "US2", "maxActive": 4, "properties": { "DatasetPaths": "E:/ERM-US/2014.09/driving/northeast;E:/ERM-US/2014.09/driving/central" } }]

---

ermdb export

Note: For instructions on installing and running the Administration Utility, see **Getting Started with the Administration Utility** on page 86.

The `ermdb export` command allows you to export the routing databases configured on the server to a file. This file can then be used to import into another instance using the `ermdb import` command, either as a backup, or for migration from one instance to another. You must have the Enterprise Routing Module installed to use this command.

Note: The `ermdb export` command will always create an export filename name `ermDbResource.txt`

Usage

```
ermdb export --directory directory_name
```

> **Note:** To see a list of parameters, type `help ermdb export`.

Required	Argument	Description
No	`--directory` or `--o` *directory_name*	Specifies the name of the directory on the file system where to export the database file. The export command will always create an export filename name `ermDbResource.txt`. If this parameter is not specified, the export file will be created in the directory where the export command is being executed.

Example

This example creates an export database file in the `E:/ERM-US/export` directory.

```
ermdb export --directory E:/ERM-US/export
```

---

erm getpointdata

> **Note:** For instructions on installing and running the Administration Utility, see **Getting Started with the Administration Utility** on page 86.

The `erm getpointdata` command returns segments information for a point. The closest segment(s) is returned to the specified point. Types of information returned are; segment ID, road type, length, speed, direction, time, road name, etc. You must have the Enterprise Routing Module installed to use this command.

Usage

```
erm getpointdata --datasource db_resource --point "x,y,coordsys"
```

> **Note:** To see a list of parameters, type `help erm getpointdata`.

Required	Argument	Description
Yes	`--datasource` *db_resource*	Specifies the name of the database resource to return data. For a list of existing routing database resources, use the `ermdb list` command.
Yes	`--point` *"x,y,coordsys"*	Indicates the point to return the closest segment information. The point is specified in the format

Required	Argument	Description
		"x,y,coordsys", where *coordsys* is the coordinate system of the point.

Example

This example returns the closest segment data to the specified point from the US_NE database resources configured on the server.

```
erm getpointdata --datasource US_NE --point "-72,40,epsg:4326"
```

erm getsegmentdata

Note: For instructions on installing and running the Administration Utility, see **Getting Started with the Administration Utility** on page 86.

The `erm getsegmentdata` command returns segments information for a given segment ID. Types of information returned are; segment ID, road type, length, speed, direction, time, road name, etc. You must have the Enterprise Routing Module installed to use this command.

Usage

```
erm getsegmentdata --datasource db_resource --segmentid "segment_id"
```

Note: To see a list of parameters, type `help erm getsegmentdata`.

Required	Argument	Description
Yes	`--datasource` *db_resource*	Specifies the name of the database resource to return data. For a list of existing routing database resources, use the `ermdb list` command.
Yes	`--segmentid` *"segment_id"*	Indicates the segment to return the information. The segment is specified in the format specified in the data. For example, *"7e3396fc:6e5251"*.

Example

This example returns data for the specified segment from the US_NE database resources configured on the server.

```
erm getsegmentdata --datasource US_NE --segmentid
"7e3396fc:6e5251"
```

erm createpointupdate

Note: For instructions on installing and running the Administration Utility, see **Getting Started with the Administration Utility** on page 86.

The `erm createpointupdate` command overrides the routing data of the closest segment for a given point. This command allows you to set or change the speed, or exclude a section of the route. You must have the Enterprise Routing Module installed to use this command.

Note: The type of persistent update is valid only for the specified data resource and may not be valid after a data update.

Usage

```
erm createpointupdate --datasource db_resource --point "x,y,coordsys" --exclude
--velocity velocity_value --velocityunit velocity_unit --velocityadjustment
velocity_adjustment_value --velocitypercentage velocity_percentage_value
```

Note: To see a list of parameters, type `help erm createpointupdate`.

Required	Argument	Description
Yes	`--datasource` *db_resource*	Specifies the name of the database resource to override the data. For a list of existing routing database resources, use the `ermdb list` command.
Yes	`--point` *"x,y,coordsys"*	Indicates the point to override the closest segment information. The point is specified in the format *"x,y,coordsys"*, where *coordsys* is the coordinate system of the point.
No	`--exclude`	Excludes the specified point from all route calculations when set as `true`. Having this parameter in the command specifies whether to exclude the point. To avoid the exclusion, add `false` after --exclude.
No	`--velocity` *velocity_value*	Defines a speed update where you specify the new speed of the point by specifying the new velocity. The default unit is mph(miles per hour) unless you specify the `velocityunit` parameter.
No	`--velocityunit` *velocity_unit*	Defines a unit of speed for the `velocity` or `velocityadjustment` overrides. The default value is mph(miles per hour). For speed updates,

Required	Argument	Description
		the velocity unit can have one of the following values: kph (kilometers per hour), mps(meters per second), or mph (miles per hour).
No	`--velocityadjustment` *velocity_adjustment_value*	Defines a speed update where you define a change in the speed of the point by specifying the change in velocity (unit and value). Speed values can be increased (positive value) or decreased(negative value). The default unit is mph(miles per hour) unless you specify the `velocityunit` parameter.
No	`--velocitypercentage` *velocity_percentage_value*	Defines a speed update where you define an increase in the speed of the point by specifying a percentage to increase(positive value) or decrease(negative value) the speed.

Examples

This example overrides the speed of the point to 15 mph, from the US_NE database resources configured on the server.

```
erm createpointupdate --datasource US_NE --point
"-72,40,epsg:4326" --velocity 15 --velocityunit mph
```

This example excludes the specified point from the US_NE database resources configured on the server.

```
erm createpointupdate --datasource US_NE --point
"-72,40,epsg:4326" --exclude true
```

This example overrides the speed of the point by increasing the speed by 45 kph, from the US_NE database resources configured on the server.

```
erm createpointupdate --datasource US_NE --point
"-72,40,epsg:4326" --velocityadjustment 45 --velocityunit kph
```

This example overrides the speed of the point by decreasing the speed by 60 percent, from the US_NE database resources configured on the server.

```
erm createpointupdate --datasource US_NE --point
"-72,40,epsg:4326" --velocitypercentage -60
```

erm resetpointupdate

Note: For instructions on installing and running the Administration Utility, see **Getting Started with the Administration Utility** on page 86.

The `erm resetpointupdate` command returns any overrides to the original state of the data. You must have the Enterprise Routing Module installed to use this command.

Usage

`erm resetpointupdate --datasource` *db_resource* `--point` *"x,y,coordsys"* `--resettype` *reset_type*

> **Note:** To see a list of parameters, type `help erm resetpointupdate`.

Required	Argument	Description
Yes	`--datasource` *db_resource*	Specifies the name of the database resource that has the overrides. For a list of existing routing database resources, use the `ermdb list` command.
Yes	`--point` *"x,y,coordsys"*	Indicates the point where the existing overrides are located. The point is specified in the format *"x,y,coordsys"*, where *coordsys* is the coordinate system of the point.
Yes	`--resettype` *reset_type*	The type of override to remove (undo).
		speed Removes a speed update.
		exclude Removes an exclude update.

Example

This example resets an existing exclude override for the given point, from the US_NE database resources configured on the server.

```
erm resetpointupdate --datasource US_NE --point
"-72,40,epsg:4326" --resettype exclude
```

---

erm createsegmentupdate

> **Note:** For instructions on installing and running the Administration Utility, see **Getting Started with the Administration Utility** on page 86.

The `erm createsegmentupdate` command overrides the routing data of the specified segment. This command allows you to set or change the speed, exclude a section of the route, or change the road type. You must have the Enterprise Routing Module installed to use this command.

> **Note:** The type of persistent update is valid only for the specified data resource and may not be valid after a data update.

```
erm createsegmentupdate --datasource db_resource --segmentid "segment_id"
--exclude --velocity velocity_value --velocityunit velocity_unit --velocityadjustment
velocity_adjustment_value --velocitypercentage velocity_percentage_value --roadtype
road_type
```

> **Note:** To see a list of parameters, type `help erm createsegmentupdate`.

Required	Argument	Description
Yes	`--datasource` *db_resource*	Specifies the name of the database resource to override the data. For a list of existing routing database resources, use the `ermdb list` command.
Yes	`--segmentid` *"segment_id"*	Indicates the segment to override. The segment is specified in the format specified in the data. For example, *"7e3396fc:6e5251"*.
No	`--exclude`	Excludes the specified segment from all route calculations when set to true. Having this parameter in the command specifies whether to exclude the segment. To avoid the exclusion, add `false` after --exclude.
No	`--velocity` *velocity_value*	Defines a speed update where you specify the new speed of the segment by specifying the new velocity. The default unit is mph(miles per hour) unless you specify the `velocityunit` parameter.
No	`--velocityunit` *velocity_unit*	Defines a unit of speed for the `velocity` or `velocityadjustment` overrides. The default value is mph(miles per hour). For speed updates, the velocity unit can have one of the following values: kph (kilometers per hour), mps(meters per second), or mph (miles per hour).
No	`--velocityadjustment` *velocity_adjustment_value*	Defines a speed update where you define a change in the speed of the segment by specifying the change in velocity (unit and value). Speed values can be increased (positive value) or decreased(negative value). The default unit is mph(miles per hour) unless you specify the `velocityunit` parameter.
No	`--velocitypercentage` *velocity_percentage_value*	Defines a speed update where you define an increase in the speed of the segment by specifying

Required	Argument	Description
		a percentage to increase(positive value) or decrease(negative value) the speed.
No	`--roadtype` *road_type*	Defines the new road type for the segment.

Examples

This example overrides the speed of the segment to 15 mph, from the US_NE database resources configured on the server.

```
erm createsegmentupdate --datasource US_NE --segmentid
"7e3396fc:6e5251" --velocity 15 --velocityunit mph
```

This example excludes the specified segment from the US_NE database resources configured on the server.

```
erm createsegmentupdate --datasource US_NE --segmentid
"7e3396fc:6e5251" --exclude true
```

This example overrides the speed of the segment by increasing the speed by 45 kph, from the US_NE database resources configured on the server.

```
erm createsegmentupdate --datasource US_NE --segmentid
"7e3396fc:6e5251" --velocityadjustment 45 --velocityunit kph
```

This example overrides the speed of the segment by decreasing the speed by 60 percent, from the US_NE database resources configured on the server.

```
erm createsegmentupdate --datasource US_NE --segmentid
"7e3396fc:6e5251" --velocitypercentage -60
```

This example overrides the road type of the segment to ferry, from the US_NE database resources configured on the server.

```
erm createsegmentupdate --datasource US_NE --segmentid
"7e3396fc:6e5251" --roadtype ferry
```

---

erm resetsegmentupdate

Note: For instructions on installing and running the Administration Utility, see **Getting Started with the Administration Utility** on page 86.

The `erm resetsegmentupdate` command returns any overrides to the original state of the data. You must have the Enterprise Routing Module installed to use this command.

Usage

```
erm resetsegmentupdate --datasource db_resource --segmentid "segment_id"
--resettype reset_type
```

Note: To see a list of parameters, type `help erm resetsegmentupdate`.

Required	Argument	Description
Yes	`--datasource` *db_resource*	Specifies the name of the database resource that has the overrides. For a list of existing routing database resources, use the `ermdb list` command.
Yes	`--segment` *"segment_id"*	Indicates the segment where the existing overrides are located. The segment is specified in the format specified in the data. For example, *"7e3396fc:6e5251"*.
Yes	`--resettype` *reset_type*	The type of override to remove (undo).
		speed Removes a speed update.
		exclude Removes an exclude update.
		roadtype Removes a road type update.

Example

This example resets an existing road type override for the given segment, from the US_NE database resources configured on the server.

```
erm resetsegmentupdate --datasource US --segmentid
"7e3396fc:6e5251" --resettype roadtype
```

---

erm getsegmentupdates

Note: For instructions on installing and running the Administration Utility, see **Getting Started with the Administration Utility** on page 86.

The `erm getsegmentupdates` command returns a list of overrides in the routing data for the specified segments. You must have the Enterprise Routing Module installed to use this command.

Note: `segmentids` is an optional parameter. If no segment ids are specified, then overrides for all available segments are returned.

Usage

```
erm getsegmentupdates --datasource db_resource --segmentids "segment_ids"
--velocityunit velocityunit
```

> **Note:** To see a list of parameters, type `help erm getsegmentupdates`.

Required	Argument	Description
Yes	`--datasource` *db_resource*	Specifies the name of the database resource that has overrides. For a list of existing routing database resources, use the `ermdb list` command.
No	`--segmentids` *"segment_ids"*	A comma separated list of segment ids to return override information. Segments are specified in the format specified in the data. For example, *"7e3396fc:6e5251"*.
No	`--velocityunit` *velocityunit*	Specifies the velocity unit to appear in the response (mph - miles per hour, kph - kilometers per hour, mtps - meters per second, and mtpm - meters per minute). The default is mph.

Example

This example returns the overrides for a segment, from the US_NE database resources configured on the server.

```
erm getsegmentupdates --datasource US_NE --segmentids
"7e3396fc:6e5251" --velocityunit kph
```

---

erm createroadtypeupdate

> **Note:** For instructions on installing and running the Administration Utility, see **Getting Started with the Administration Utility** on page 86.

The `erm createroadtypeupdate` command overrides the routing data of the specified road type. This command allows you to set or change the speed of the route for the particular road type. You must have the Enterprise Routing Module installed to use this command.

> **Note:** The type of persistent update is valid only for the specified data resource and may not be valid after a data update.

Usage

```
erm createroadtypeupdate --datasource db_resource --roadtype "road_type"
--velocity velocity_value --velocityunit velocity_unit --velocityadjustment
```

velocity_adjustment_value `--velocitypercentage` *velocity_percentage_value* `--roadtype`
road_type

> **Note:** To see a list of parameters, type `help erm createroadtypeupdate`.

Required	Argument	Description
Yes	`--datasource` *db_resource*	Specifies the name of the database resource to override the data. For a list of existing routing database resources, use the `ermdb list` command.
Yes	`--roadtype` *"road_type"*	Indicates the road type to override. The road type can be one of the following:

- access way
- back road
- connector
- ferry
- footpath
- limited access dense urban
- limited access rural
- limited access suburban
- limited access urban
- local road dense urban
- local road rural
- local road suburban
- local road urban
- major local road dense urban
- major local road rural
- major local road suburban
- major local road urban
- major road dense urban
- major road rural
- major road suburban
- major road urban
- minor local road dense Urban
- minor local road rural
- minor local road suburban
- minor local road urban
- normal road dense urban
- normal road rural
- normal road rural
- normal road urban

Required	Argument	Description
		• primary highway dense urban • primary highway rural • primary highway suburban • primary highway urban • ramp dense urban • ramp limited access • ramp major road • ramp primary highway • ramp rural • ramp secondary highway • ramp urban • ramp suburban • secondary highway dense urban • secondary highway rural • secondary highway suburban • secondary highway urban
No	`--velocity` *velocity_value*	Defines a speed update where you specify the new speed of the road type by specifying the new velocity. The default unit is mph(miles per hour) unless you specify the `velocityunit` parameter.
No	`--velocityunit` *velocity_unit*	Defines a unit of speed for the `velocity` or `velocityadjustment` overrides. The default value is mph(miles per hour). For speed updates, the velocity unit can have one of the following values: kph (kilometers per hour), mps(meters per second), or mph (miles per hour).
No	`--velocityadjustment` *velocity_adjustment_value*	Defines a speed update where you define a change in the speed of the road type by specifying the change in velocity (unit and value). Speed values can be increased (positive value) or decreased(negative value). The default unit is mph(miles per hour) unless you specify the `velocityunit` parameter.
No	`--velocitypercentage` *velocity_percentage_value*	Defines a speed update where you define an increase in the speed of the road type by specifying a percentage to increase(positive value) or decrease(negative value) the speed.

Examples

This example overrides the speed of a road type to 25 kph, from the US_NE database resources configured on the server.

```
erm createroadtypeupdate --datasource US_NE --roadtype "normal
road suburban" --velocity 25 --velocityunit kph
```

This example increases the speed of the specified road type by 50 kph, from the US_NE database resources configured on the server.

```
erm createroadtypeupdate --datasource US_NE --roadtype "normal
road suburban" --velocityadjustment 50 --velocityunit mph
```

This example overrides the speed of the road type by decreasing the speed by 65 percent, from the US_NE database resources configured on the server.

```
erm createroadtypeupdate --datasource US_NE --roadtype "normal
road suburban" --velocitypercentage -65
```

erm resetroadtypeupdate

Note: For instructions on installing and running the Administration Utility, see **Getting Started with the Administration Utility** on page 86.

The `erm resetroadtypeupdate` command returns any overrides to the original state of the data. You must have the Enterprise Routing Module installed to use this command.

Usage

`erm resetroadtypeupdate --datasource` *db_resource* `--roadtype` *"road_type"*

Note: To see a list of parameters, type `help erm resetroadtypeupdate`.

Required	Argument	Description
Yes	`--datasource` *db_resource*	Specifies the name of the database resource that has the overrides. For a list of existing routing database resources, use the `ermdb list` command.
Yes	`--roadtype` *"road_type"*	Indicates the road type that has the existing overrides. For a list of road types, see **erm createroadtypeupdate** on page 107.

> **Example**
>
> This example resets the "normal road suburban" road type override, from the US_NE database resources configured on the server.
>
> ```
> erm resetroadtypeupdate --datasource US_NE --roadtype "normal
> road suburban"
> ```

erm getroadtypeupdates

Note: For instructions on installing and running the Administration Utility, see **Getting Started with the Administration Utility** on page 86.

The `erm getroadtypeupdates` command returns a list of overrides in the routing data for the specified road types. You must have the Enterprise Routing Module installed to use this command.

Note: `roadtypes` is an optional parameter. If no road types are specified, then overrides for all available road types are returned.

Usage

```
erm getroadtypeupdates --datasource db_resource --roadtypes "road_types"
--velocityunit velocityunit
```

Note: To see a list of parameters, type `help erm getroadtypeupdates`.

Required	Argument	Description
Yes	`--datasource` *db_resource*	Specifies the name of the database resource that has overrides. For a list of existing routing database resources, use the `ermdb list` command.
No	`--roadtypes` *"road_types"*	A comma separated list of road types to return override information. For a list of road types, see **erm createroadtypeupdate** on page 107.
No	`--velocityunit` *velocityunit*	Specifies the velocity unit to appear in the response (mph - miles per hour, kph - kilometers per hour, mtps - meters per second, and mtpm - meters per minute). The default is mph.

> **Example**
>
> This example returns the overrides for the "normal road urban" road type, from the US_NE database resources configured on the server.

```
erm getroadtypeupdates --datasource US_NE --roadtypes "normal
road urban" --velocityunit kph
```

erm getallupdates

Note: For instructions on installing and running the Administration Utility, see **Getting Started with the Administration Utility** on page 86.

The `erm getallupdates` command returns a list of all overrides for a specified routing database resource. You must have the Enterprise Routing Module installed to use this command.

Usage
`erm getallupdates --datasource` *db_resource "segment_ids"* `--velocityunit` *velocityunit*

Note: To see a list of parameters, type `help erm getallupdates`.

Required	Argument	Description
Yes	`--datasource` *db_resource*	Specifies the name of the database resource that has the overrides. For a list of existing routing database resources, use the `ermdb list` command.
No	`--velocityunit` *velocityunit*	Specifies the velocity unit to appear in the response (mph - miles per hour, kph - kilometers per hour, mtps - meters per second, and mtpm - meters per minute). The default is mph.

Example

This example returns all the overrides from the US_NE database resources configured on the server.

```
erm getallupdates --datasource US_NE --velocityunit kph
```

---

erm resetallupdates

Note: For instructions on installing and running the Administration Utility, see **Getting Started with the Administration Utility** on page 86.

The `erm resetallupdates` command returns all overrides to the original state of the data. You must have the Enterprise Routing Module installed to use this command.

Usage

`erm resetallupdates --datasource` *db_resource*

> **Note:** To see a list of parameters, type `help erm resetallupdates`.

Required	Argument	Description
Yes	`--datasource` *db_resource*	Specifies the name of the database resource that has the overrides. For a list of existing routing database resources, use the `ermdb list` command.

Example

This example resets all overrides from the US_NE database resources configured on the server.

`erm resetallupdates --datasource US_NE`

---

8 - Enterprise Routing Module

In this section

Specifying Default Service/Stage Options

Default options control the default behavior of each service or stage on your system. You can specify a default value for each option. The default option takes effect when a request does not explicitly define a value for a given option. These default options are also the settings used by default when you create a dataflow in Enterprise Designer using this service.

For information about the options, see the Stages and Resources and Data sections in the *Spectrum Spatial Guide* that apply to the Enterprise Routing Module.

> **Note:** Persistent Updates are not managed using the Management Console. To make persistent updates, use the spectrum command line functionality in the Administration Utility.

> **Note:** The Get Route Data service in the Management Console does not set default options, rather it is an interactive way to return routing data for segments. For more information on Get Route Data, see **Getting Route Data using Management Console** on page 117.

1. Open Management Console.
2. Click **Services**.
3. Click the module you want (Enterprise Routing Module).
4. Click the service you want to configure from the list on the left.
5. Set the options for the service. Most services have various types of options that appear on different tabs.
6. Click **Save**.

Previewing a Service/Stage

You can preview the results of a service in Management Console using the service's Preview tab. Preview can be useful in helping you decide what options to specify because you can immediately see the effect that different options have on the data returned by the service or stage.

1. Open Management Console.
2. Go to the **Services** menu and select the service you want to preview.
3. Click the **Preview** tab.
4. Enter the test data into each field.

   Here are some tips for using preview:

- You do not have to enter data in every field. Leaving a field empty results in an empty string being used for preview.
- If you want to preview the effect of passing a null value in a field, click the Disable icon next to the field:
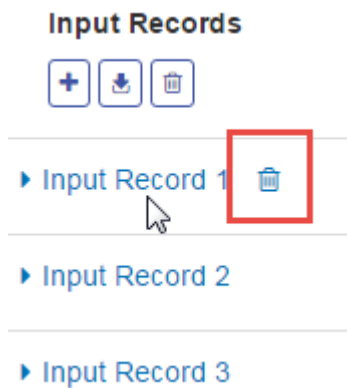
Input Records

[+] [↓] [🗑]

▼ Input Record 1

AddressLine1 ⊘

AddressLine2
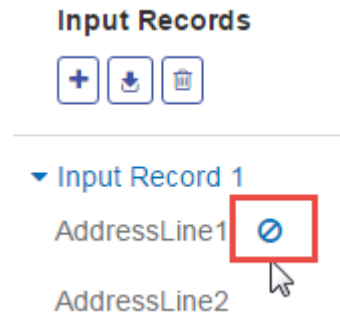
- You can preview multiple records at once. To add a record, click the Add button ✚ .
- You can import test data from a file. To import data, click the Import button [↓]. Select the **File name** and the **Field separator**. Note the following:

  - The first row in the file must be a header record. The field names in the header must match the field names required by the service.
  - The maximum number of records that can be imported is five.
  - If the file uses a space as the field separator, field values must be surrounded by quotes. Here is an example of a file that uses a space as the field separator:

```
AddressLine1 AddressLine2 City StateProvince PostalCode
"One Global View" "" "Troy" "NY" "12180"
"3001 Summer St" "" "Stamford" "CT" "06926"
"224 N Michigan Ave" "Suite 300" "Chicago" "IL" ""
```

- To delete all records, click the Delete button at the top of the preview area:

Input Records

[+] [↓] [🗑]

▶ Input Record 1

▶ Input Record 2
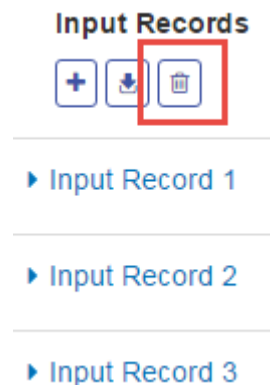
▶ Input Record 3

- To delete an individual record, hover over the input record name (for example, "Input Record 1") and click the Delete button next to the record name:

- If the service takes hierarchical input data:

  - To add child records, hover over the parent record and click the Add button.
  - To delete all children of a parent, hover over the parent record and click the Delete button.
  - To delete individual child records, hover over the child record and click the Delete button.

5. Click **Run Preview**.

   The service processes the input records and displays the results

6. Review your output data, making sure the results are what you intended to get from the service or stage. If necessary you can make changes to the option and click **Run Preview** again. (You do not need to input the data again.)

Getting Route Data using Management Console

Using the Management Console, you can preview and save segment information either from a closest point or segment ID. The GetRouteData service returns segment information for a point or segment ID. When a point is specified, the closest route segments are returned. When a segment ID is specified, the route data for that specified route segment is returned.

To preview and/or save route data:

1. Open Management Console.
2. Go to the **Services** menu and select Enterprise Routing Module.
3. Select **Get Route Data** from the services list.
4. Select either Point Data or Segment Data from the **Input Type** field.
5. Select the routing database resource from the **Database** field.

   If you need to add a new routing database resource, see **Adding a Routing Database Resource**.

6. Enter the required information for the Input Type you selected.

If you selected Point Data, enter the point coordinates and the coordinate system. If you selected Segment Data, enter the segment ID.

7. Click **Preview**.

   The route segment data is returned in the **Output Data** section. When there are more than one segments associated with the input, the multiple segments will be listed with Segment Details 1, Segment Details 2, etc.

8. Click either the **Save** button to save the routing data results as a text file, or the **Clear** button to remove the results from the Output Data.

9 - Troubleshooting Your System

In this section

Rebuilding a Corrupt Repository Index

Sometimes the repository can become corrupt if the server is shut down abruptly or the Java process is killed (manually or due to a power outage). As a result, you may be unable to get resources that were previously searchable, and there will be no errors or warnings in the logs. Once you verify that permission changes are not the cause, rebuild the index to fix this issue:

1. Shut down the server.
2. Delete the index directory at the following locations:

   • <Spectrum>\server\modules\spatial\jackrabbit\workspaces\default
   • <Spectrum>\server\modules\spatial\jackrabbit\workspaces\security
   • <Spectrum>\server\modules\spatial\jackrabbit\repository

3. Restart the server.
   Jackrabbit re-creates the index at the above locations while booting.

After rebuilding the index, the search works correctly again.

Monitoring Memory Usage of a Non-Responsive Server

If your Spectrum server stops responding, you can follow the steps below to monitor its performance and resource consumption. This monitoring provides information you can use to adjust memory and threading usage.

1. Check whether a service other than the Mapping Service is working. For example, start the Feature Service on the demo page:
   http://<servername>:<port>/Spatial/FeatureService//DemoPage.html. This determines whether the whole server is down or just the Mapping Service.
2. Verify you have enough disk space for both Mapping and Map Tiling images to be stored by inspecting the configuration files:

   • **Mapping:**
     http://<server>:<port>/RepositoryService/repository/default/Configuration/MappingConfiguration
     under "`<Directory> C:\Program Files\Pitney`
     `Bowes\Spectrum/server/modules/spatial/images </Directory>`"
   • **Map Tiling:**
     "http://<server>:<port>/RepositoryService/repository/default/Configuration/MapTilingConfiguration"
     under "`<Property name="diskPath" value="C:/Program Files/Pitney`
     `Bowes/Spectrum/server/modules/spatial/TileCache"/>`"

3. Stop the Spectrum server.

4. In a text editor, open the java.vmargs files from `<Installed>\Pitney Bowes\Spectrum\server\modules\spatial\java.vmargs`.

5. Change the vmargs default of 2GB (2048MB). For example, to increase the memory of the remote component to 4GB, change the vmargs from the default of `-Xmx2048m` to `-Xmx4096m`. Do not exceed the maximum memory available to your operating system and leave a suitable space for the operating system to do its work.

6. Save the java.vmargs file.

7. Start the server wrapper:

    a) Open a command prompt as Administrator.

    b) Go to `<Installed>\Pitney Bowes\Spectrum\server\bin\wrapper` directory and type **wrapper.exe -c**.

    This Spectrum server will start in a few minutes.

8. When the server is started, run the following requests from the demo pages:

    a) Open http://<servername>:<port>/Spatial/MappingService/DemoPage.html and run the List Named Maps request.

    b) Open http://<servername>:<port>/Spatial/FeatureService/DemoPage.html and run the List Table Names request.

9. Go to <Installed>\Pitney Bowes\Spectrum\java64\bin and run jconsole.exe.

10. Under Local Process, select the wrapper process.

11. In Jconsole, add a new session and select the Feature Service process.

12. In Jconsole, add a new session and select the Mapping Service process.

13. Leave Jconsole running to monitor the memory, CPU, threads, and so on for the Spectrum Platform wrapper for Feature Service and Mapping Service.
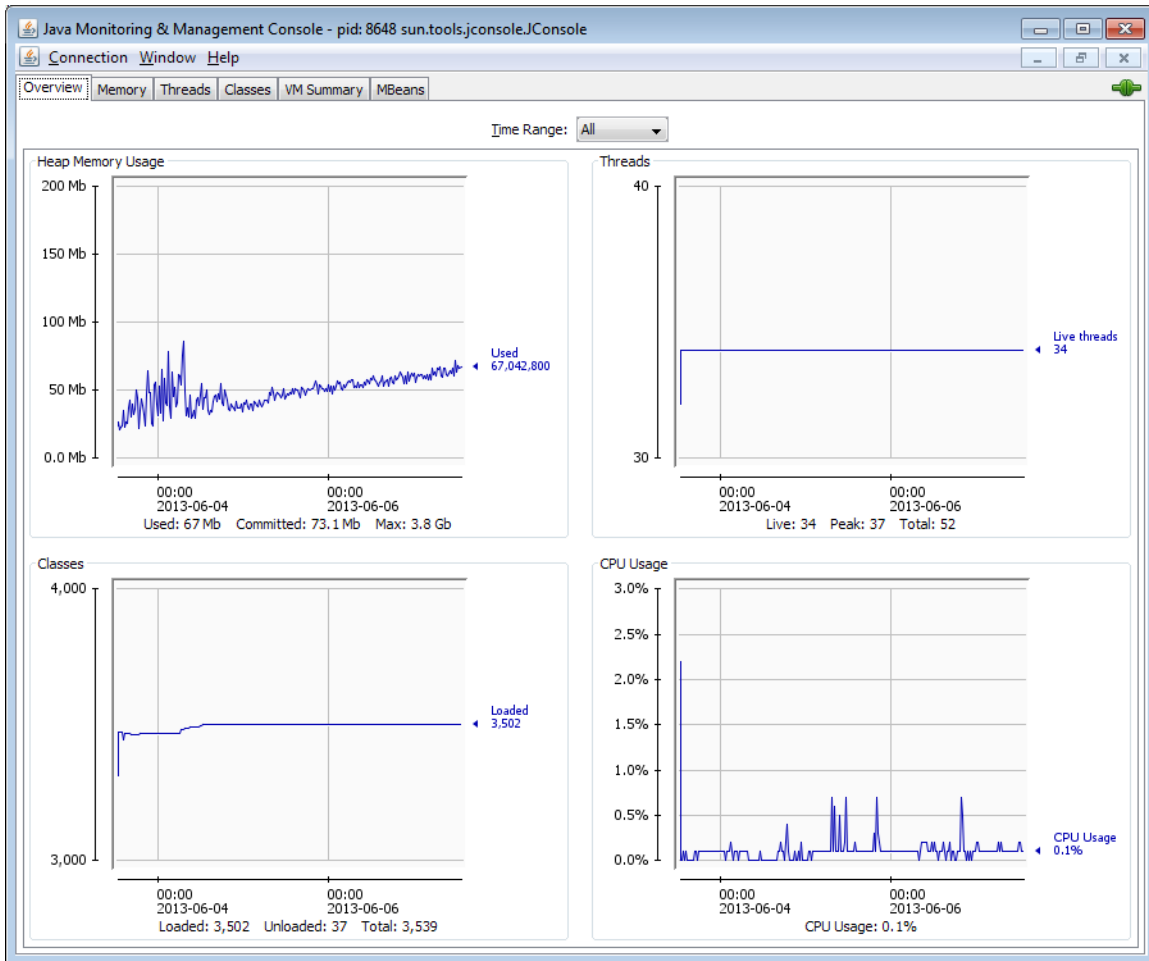
Notices

USPS® Notices

Pitney Bowes Inc. holds a non-exclusive license to publish and sell ZIP + 4® databases on optical and magnetic media. The following trademarks are owned by the United States Postal Service: CASS, CASS Certified, DPV, eLOT, FASTforward, First-Class Mail, Intelligent Mail, LACS^Link, NCOA^Link, PAVE, PLANET Code, Postal Service, POSTNET, Post Office, RDI, Suite^Link , United States Postal Service, Standard Mail, United States Post Office, USPS, ZIP Code, and ZIP + 4. This list is not exhaustive of the trademarks belonging to the Postal Service.

Pitney Bowes Inc. is a non-exclusive licensee of USPS® for NCOA^Link® processing.

Prices for Pitney Bowes Software's products, options, and services are not established, controlled, or approved by USPS® or United States Government. When utilizing RDI™ data to determine parcel-shipping costs, the business decision on which parcel delivery company to use is not made by the USPS® or United States Government.

Data Provider and Related Notices

Data Products contained on this media and used within Pitney Bowes Software applications are protected by various trademarks and by one or more of the following copyrights:

© Copyright United States Postal Service. All rights reserved.

© 2014 TomTom. All rights reserved. TomTom and the TomTom logo are registered trademarks of TomTom N.V.

© 2016 HERE

Fuente: INEGI (Instituto Nacional de Estadística y Geografía)

Based upon electronic data © National Land Survey Sweden.

© Copyright United States Census Bureau

© Copyright Nova Marketing Group, Inc.

Portions of this program are © Copyright 1993-2007 by Nova Marketing Group Inc. All Rights Reserved

© Copyright Second Decimal, LLC

© Copyright Canada Post Corporation

This CD-ROM contains data from a compilation in which Canada Post Corporation is the copyright owner.

© 2007 Claritas, Inc.

The Geocode Address World data set contains data licensed from the GeoNames Project (**www.geonames.org**) provided under the Creative Commons Attribution License ("Attribution

License") located at **http://creativecommons.org/licenses/by/3.0/legalcode**. Your use of the GeoNames data (described in the Spectrum™ Technology Platform User Manual) is governed by the terms of the Attribution License, and any conflict between your agreement with Pitney Bowes Software, Inc. and the Attribution License will be resolved in favor of the Attribution License solely as it relates to your use of the GeoNames data.