

# Spectrum™ Technology Platform

バージョン 2019.1.0

Spectrum Spatial™ 管理ガイド



# 目次

## 1 - はじめに

---

このガイドの内容	5
----------	---

## 2 - システムの設定

---

Spectrum Spatial の HTTP ポート番号の変更	7
リポジトリ データベース タイプの変更	8
Web サービスの設定	8
ジオメトリ ノードを表す桁数の制御	9
測地系変換用の精度ファイルの無効化	10
名前付きテーブルの可変属性の設定	11
Spectrum™ Technology Platformを Linux サービスとして実行	12
MRR 用の Linux マシンの設定	14
デフォルトの HTTP キャッシュ制御ヘッダの無効化	15

## 3 - システムの監視

---

システム イベントの表示	18
Spatial のログ	19
メール サーバーの設定	21
有効期限通知を受信する項目の選択	22
バージョン情報の表示	23
ライセンス情報の表示とエクスポート	24
JMX コンソールによるパフォーマンスのモニタリング	24
JMX コンソールによる、ファイルハンドルキャッシュ統計の監視	25
メモリ使用量の監視	26

Spatial モジュールを再起動する	27
MRR キャッシュのクリア	28

## 4 - パフォーマンス チューニング

---

リモート コンポーネントの構成設定	30
データ ソース プーリングの設定	31
距離ベースの操作のパフォーマンスの向上	31
ACL プロパティの管理	32

## 5 - クラスタの管理

---

Spatial モジュールのクラスタ アーキテクチャ	34
クラスタでの Enterprise Designer の使用	36
クラスタの起動	37
クラスタの停止	37
クラスタからのノードの削除	38
Spatial モジュール用のクラスタの管理	39

## 6 - 管理ユーティリティの使用

---

管理ユーティリティを使用する前に	45
管理ユーティリティをスクリプトから使用する	46
Spatial モジュール	47
ルーティング	54

## 7 - ルーティング

---

デフォルト サービス/ステージ オプションの指定	79
サービス/ステージのプレビュー	79

Management Console を用いたルート データの取得 82

## 8 - システムのトラブルシューティング

---

破損したりポジトリ インデックスの再構築 84  
反応がなくなったサーバーのメモリ使用量の監視 84  
空間リモート コンポーネントのヒープ メモリの増加 86

# 1 - はじめに

## このセクションの構成

---

このガイドの内容

5

## このガイドの内容

『*Spectrum Spatial 管理ガイド*』へようこそ。このガイドは、各種 Web サービス、機能、ツール、およびサンプルコードを使用して、Web マッピング アプリケーションを作成したり、既存のアプリケーションにマッピングを組み込むのに役立ちます。

このガイドでは、次の内容について説明します。

- デフォルトのポート番号またはリポジトリ データベースを変更することによるシステムの設定。リポジトリへのアクセス。リソースのアクセスとアップロード。Web サービスの設定。Linux サービスとしての Spectrum™ Technology Platform の実行。
- Management Console を使用したセキュリティの管理。ユーザおよび役割を追加する方法、セキュリティ エンティティのオーバーライドを適用する方法を含みます。
- システムの監視。ログ記録、バージョンとライセンスの情報の表示、JMX コンソールを使用したパフォーマンスの監視、メモリの使用状況の監視などです。
- メモリとスレッドの監視。JVM のパフォーマンスのチューニング、プールサイズの調整、ヒープメモリの追加などです。
- 耐障害性または高いキャパシティのための空間サービスの負荷分散。
- システムのトラブルシューティング。壊れたりリポジトリ インデックスの再構築、応答しないサーバーのメモリ使用状況の注意などです。
- User Management Service (将来のリリースで非推奨) を使ったセキュリティの管理

Spectrum™ Technology Platform および Spatial モジュールの他のドキュメントは、[support.pb.com](https://support.pb.com) からオンラインで入手できます。

# 2 - システムの設定

## このセクションの構成

---

Spectrum Spatial の HTTP ポート番号の変更	7
リポジトリ データベース タイプの変更	8
Web サービスの設定	8
ジオメトリ ノードを表す桁数の制御	9
測地系変換用の精度ファイルの無効化	10
名前付きテーブルの可変属性の設定	11
Spectrum™ Technology Platformを Linux サービスとして実行	12
MRR 用の Linux マシンの設定	14
デフォルトの HTTP キャッシュ制御ヘッダの無効化	15

## Spectrum Spatial の HTTP ポート番号の変更

HTTP ポートは、すべての Spectrum™ Technology Platform Web サービスにアクセスするために使用されます。REST と SOAP のどちらにも対応し、ようこそページ、サンプルアプリ、および Spectrum Spatial™ Manager のアクセスにも使用します。

Spectrum™ Technology Platform のインストール後、グローバル、起動用、または個別のサービス設定ファイルを手動で編集すれば、インストール中に割り当てられた既存のポート設定を変更できます。ポート番号の変更が必要となる理由はいくつかあります。

- インストール後に、ポートの競合が発生した場合。
- 古い Spectrum™ Technology Platform を削除せずに新バージョンを試してみたい場合。両方をインストールすることはできないため、既存のバージョンを無効にして、別のポートを使用する Spectrum™ Technology Platform イメージをインストールします。
- ポート 8080 にプロキシが必要だが、外部に公開するポート数に制限があるため、設定やデータフローを作成し直すことなく Spectrum™ Technology Platform を移動したい場合。

注：この作業は、アプリケーションサーバーのポート番号を変更したことのある経験豊富な管理者のみが実行してください。ネットワークポートに競合があると、モジュールコンポーネントが起動に失敗する恐れがあります。コンポーネントが Management Console に表示されない場合は、コンポーネントが起動に失敗したことを表しています。この問題をトラブルシューティングするには、Spectrum™ Technology Platform サーバーの `wrapper.log` を確認してください。このログには、どのポートが問題を起こしているかが示されています。`wrapper.log` は、`<install_folder>\server\logs\spectrum-server.log` にあります。

ポート番号を変更するには

1. `spectrum-container.properties` で、`spectrum.http.port` の値を新しいポート番号に変更します。このファイルは、`<install_folder>/server/conf` に配置されています。
2. Spectrum Spatial™ Manager で、次のサービス設定のポート番号を変更します。
  - Mapping (SOAP 経由でマッピングサービスにアクセスし、RenderMap 要求の ReturnImage パラメータが False の場合にのみ必要)
  - WFS
  - WMS
  - WMTS

手順については、『Spectrum Spatial ガイド』の「Spatial の管理」にある「Spectrum Spatial™ Manager」を参照してください。

別のポートが使えるようにサーバーを移動させる場合は、Spectrum™ Technology Platform サーバーが実行していない可能性があります。このサーバーを実行しないと、サービス設定ファイルを編集できません。サーバーを起動し、設定を編集し、サーバーを再起動する必要があります。

3. Spectrum™ Technology Platform を再起動してポートとプロパティの変更を適用します。

## リポジトリ データベース タイプの変更

Location Intelligence モジュールは、名前付きリソース (マップ、レイヤ、テーブル、スタイル)、地理メタデータ、構成をリポジトリに保存します。デフォルトのシングルサーバー構成では、組み込みデータベースを使用して、ローカルサーバー上にこれらのリソースを保存します。場合によっては、次のような理由から、組み込みの Derby データベース以外のデータベースを使用することが必要になります。

- 耐障害性を備えた独立したデータベースを使用してスケーラブルなソリューションを構築する場合。
- 会社が推奨または指示する社内データベースを使用する場合。

このリリースではリポジトリ データベースとして Oracle、PostgreSQL/PostGIS、Microsoft SQL Server がサポートされています。手順については、「[共通リポジトリデータベースのセットアップ \(39ページ\)](#)」を参照してください。

## Web サービスの設定

各 Web サービスの設定ファイル内の設定値で、Spatial モジュール モジュールの Web サービスの動作を明示的に指定することができ、あとで変更することも可能です。Location Intelligence モジュールでは Web サービスの設定ファイルが、名前付き設定として Spatial モジュール リポジトリに保持されます。

**注：** 名前付き設定は、リポジトリ内に保持されている他の名前付きリソースとは異なりません。Named Resource Service を使用して名前付き設定にアクセスすることはできません。代わりに WebFolders などの WebDAV ツールを使用する必要があります。

Mapping、Feature、Map Tiling、WFS、WMS および WMTS の各サービスについては、設定ファイルがリポジトリにあらかじめ読み込まれます。これらの設定ファイルは、



`http://hostname:port/RepositoryService/repository/default/Configuration/`に配置されています。

リポジトリ内にある各 Web サービスの名前付き設定の名前と場所、および各 Web サービスの設定パラメータの一覧は、『Spectrum Spatial 開発者ガイド』の「空間サービスの使用」の章を参照してください。

## ジオメトリ ノードを表す桁数の制御

Location Intelligence モジュールと Routing モジュールに新しいプロパティが追加され、Web サービスの応答で返されるジオメトリ ノードを表す桁数が制御できるようになりました。デフォルトでは、返されるジオメトリに桁数の制限はありません。桁数は最大で 16 桁にもなる場合があります。余分な桁数は、特に大きなポリゴンや多くのレコードが返される場合に、JSON または SOAP の応答のペイロードを不要に増加させます。また、元のデータにはない精度を期待させてしまう恐れもあります。最下位桁の 1 の違いは、10 億分の 1 メートルの違いにしかならない場合があります。例えば、3989657.014543291 と 3989657.014543292 の差は、10 億分の 1 メートルです。空間データが、それほどまでの精度を持つことはほとんどありません。プロパティに `true` を設定することで、値は一般的に有効桁数 9 桁または 10 桁に丸められます。先ほどの例を使用すると、返される値は `3989657.01` となり、その精度は 1 センチメートルです。

ノード値を丸めるには、次のプロパティを `%Spectrum%\server\bin\wrapper\wrapper.conf` に追加し、サーバーを再起動します。

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

ここで、`xx` は、セクション内の次の使用可能な行番号です。

座標値は、呼び出しが SOAP または REST のどちらであっても、データ フローから公開されるサービスを含むすべてのサービスにおいて、すべてのジオメトリで同じように扱われます。これには、Location Intelligence モジュールの Feature サービス、Mapping サービス、Geometry サービス、Map Tiling サービス、WMS、WMTS、WFS、および Enterprise Routing サービスが含まれます。

Web サービスによってポリゴンデータを編集するアプリケーションでは、丸められたジオメトリを書き戻すことによって、隣接するジオメトリの間に小さな重なりや隙間ができる可能性がある場合は、このプロパティを使用してはいけません。

## 測地系変換用の精度ファイルの無効化

Spectrum Spatial は、より正確な座標シフトに役立つアルゴリズムを用いることによって、特定の測地系の間の変換をサポートします。Spectrum のインストール場所 `server\app\types` ディレクトリにある測地系変換ごとに、これらのアルゴリズムを含む個別の jar ファイルがデフォルトでインストールされます。

- `midev-core-coordsys-irishtm-バージョン番号-onprem.jar` (アイルランドの横メルカトル図法用)
- `midev-core-coordsys-jgd2000-バージョン番号-onprem.jar` (更新バージョンである JGD2011 も有効にします) (日本の測地系用)
- `midev-core-coordsys-nadcon-バージョン番号-onprem.jar` (米国 Nad27-Nad83 用)
- `midev-core-coordsys-ntv2-バージョン番号-onprem.jar` (NTV2 用、多数の国の複数の変換を含む)

注：この jar に含まれる XML ファイルによって、使用される変換が制御されます。そのファイルの中の特定の変換を無効にするには、サーバーを停止し、jar ファイルからその XML ファイルを取り出します。エディターを使用して、無効にする各変換のエントリに "false" を設定します。編集した XML ファイルを jar ファイルに追加し直して、サーバーを再起動します。

- `midev-core-coordsys-rgf93-バージョン番号-onprem.jar` (フランスのランベルト変換用)

デフォルトで、これらすべての jar ファイルがロードされますが、それらを使用すると一部の操作のパフォーマンスが低下する可能性があります。これらの変換は、場合によって無効にすることができます。例えば、特定のタイプの変換が不要な場合 (日本の測地系を変換する必要がない場合など) や、ズームレベルが低くて精度よりもパフォーマンスが高い方がメリットがある場合などです。

特定の変換を無効にするには

1. サーバーを停止します。
2. ディレクトリから jar ファイルを削除します。または、jar ファイルを別の拡張子 (.jar~ など) に変更して、読み込みから除外することもできます。
3. サーバーを再起動します。

## 名前付きテーブルの可変属性の設定

可変性は、データソースからの情報がいつでも変化できることを Spectrum Spatial に対して示します。TAB、JDBC ベース (Oracle、SQL Server、PostGIS) の名前付きテーブルに対するデフォルト値は true に設定されています。この場合、クエリや挿入のようなデータアクセス操作のたびに、Spectrum Spatial がそのデータソースをチェックして、テーブルが可変かどうか、可変の場合はデータが変化したかどうかを確認します。データが変化している場合、データアクセス操作を続行する前に、キャッシュはフラッシュされ、テーブルは再ロードされます。テーブルが変化していない場合、クエリやその他の操作はキャッシュ内のデータに対して実行されます。各データソースに変化を引き起こす要因の詳細については、『Spectrum Spatial ガイド』の「リソースおよびデータ」セクションの「サポートされるデータソース」を参照してください。

Map Uploader を使用して MapInfo Pro からアップロードされた名前付きテーブルでは、可変性が無効になります (false に設定されます)。Spectrum Spatial™ Manager で作成されたすべての名前付きテーブルでは、可変性は有効になります。リポジトリ内の古い名前付きテーブルは可変であると見なされますが、Spectrum Spatial™ Manager のテーブルの詳細ページに表示される場合にはこの情報が示されません。

変更しないテーブルに対してしか、可変性を無効にはしてはいけません。例えば、可変 TAB ファイルからタイルを生成する操作は、非常に時間がかかります。PostGIS を使用する場合は、可変性を無効にして Spectrum Spatial™ Manager での接続エラーの発生を避けることも検討するとよいでしょう (例えば、テーブルの詳細ページでサンプル行を表示する場合)。

可変性は、Spectrum Spatial™ Manager のテーブルの詳細ページで無効にできます。Spectrum Spatial™ Manager での名前付きテーブルの作成と変更については、『Spectrum Spatial ガイド』の「Spatial の管理」セクションを参照してください。

名前付きテーブルを非可変性に設定した場合は、Spectrum Spatial™ Manager の Spatial モジュールを再起動する必要があります。

注：テキストエディタ内でのこの値の変更、または WebDAV を介してアクセスした名前付きテーブル定義の手動編集には、Named Resource Service の updateNamedResource 操作を使用しないでください。

# Spectrum™ Technology Platformを Linux サービスとして実行

このチュートリアルでは、Spectrum™ Technology Platform を Linux サービスとして実行するために必要な手順を説明します。

## Spectrum™ Technology Platform を Linux サービスとして実行する方法

この手順は、Spectrum™ Technology Platform を Linux サービスとして実行する方法を説明しています。

1. 提供されている pbspectrum スクリプト (**PBSpectrum スクリプト** (13ページ) にあります) を変更します。
  - a) 5 行目の chkconfig パラメータを変更します。このパラメータのデフォルトは #  
chkconfig: 35 90 10 です。

1 番目の値 (35) はランレベルです。詳細については、'man init' を使用してください。

2 番目の値 (90) は起動の優先順位です。

3 番目の値 (10) は停止の優先順位です。

起動と停止の優先順位は、依存関係にあるサービスに応じて設定します。例えば、同じコンピュータで実行される Oracle サーバーを Spectrum™ Technology Platform で使用する  
場合、Spectrum™ Technology Platform の起動の優先順位を Oracle サービスより低く、  
停止の優先順位を Oracle サービスより高くする必要があります。詳細については、'man  
chkconfig' を使用してください。
  - b) 11 行目の SPECTRUM\_ROOT 変数を Spectrum™ Technology Platform のインストール  
ディレクトリに変更します。
  - c) SUSE Linux を使用している場合は、デフォルトの優先ユーザを su から runuser に変更  
する必要があります。
2. 変更した pbspectrum スクリプトを、RedHat Linux の場合は /etc/rc.d/init.d に、Suse  
Linux の場合は /etc/init.d にコピーします。
3. pbspectrum スクリプトのモードを実行可能に変更します。/etc/rc.d/init.d RedHat  
Linux の場合は、Suse Linux の場合は /etc/init.d です。

使用する Linux のバージョンによっては `cd /etc/init.d` または `cd /etc/rc.d/init.d` の場合もあります。

```
chmod +x pbspectrum
```

4. `chkconfig --add pbspectrum`
5. コンピュータを再起動してスクリプトの動作を確認します。シェルから再起動するには、`shutdown -r now` を使用します。

完了後は、次の操作も利用できます。

- `service pbspectrum start` で **Spatial Server** を起動する
- `service pbspectrum stop` で **Spatial Server** を停止する
- `service pbspectrum restart` で **Spatial Server** を再起動する

注：提供されているスクリプトで実行される '`ulimit -n 8192`' コマンドは、Linux でオープンファイルの数を増加させるために必要なコマンドです。

## PBSpectrum スクリプト

次のスクリプトは、この手順の基盤として使用されます。 [Spectrum™ Technology Platform を Linux サービスとして実行する方法 \(12ページ\)](#)

```
#!/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

```

## MRR 用の Linux マシンの設定

Linux 環境の Spectrum Spatial 上で MRR (Multi Resolution Raster) ファイルを使用するには、GCC と LIBC を適切なバージョンにアップグレードする必要があります。

MRR 用に Linux マシンを設定するには

1. UUID パッケージをインストールします。これによって、LIBC v.2.17 がインストールされます。  
例えば、Cent OS 上に UUID をインストールするには次のようにします。
  - 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. devtoolset-3 をインストールします。これによって、GCC v.4.9 がインストールされます。手順については、「<https://www.softwarecollections.org/en/scls/rhsc/devtoolset-3/>」を参照してください。

3. GCC v.4.9 と LIBC v.2.17 (またはそれ以降) がインストールされていることを確認します。
4. 上記のステップで、すべての依存関係が解決されていることを確認します。解決されていない依存関係がある場合は、それをインストールしてステップ 2 を繰り返します。

例えば以下は、OEL 6.5 コンピュータに対して必要な依存関係の一部です。

- `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`

## デフォルトの HTTP キャッシュ制御ヘッダの無効化

デフォルトで、Spectrum™ Technology Platform Web サービスは、次の HTTP ヘッダをキャッシュに使用します。

```
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
```

これらの HTTP ヘッダは、Map Tiling サービスには適していません。ただし、これらのデフォルトの HTTP ヘッダを無効にして、個々の Web サービスで定義されるヘッダで HTTP キャッシュ動作を設定することができます。

注：この変更をクラスタに適用する場合は、クラスタ内の各ノードに対して以下の手順を繰り返す必要があります。

デフォルトの HTTP キャッシュ制御ヘッダを無効にするには

1. Spectrum™ Technology Platform サーバーを停止します。
2. この設定ファイルをテキスト エディタで開きます。

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

3. 次のプロパティを **true** から **false** に変更します。

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

4. プロパティ ファイルを保存して閉じます。
5. Spectrum™ Technology Platform サーバーを開始します。



# 3 - システムの監視

## このセクションの構成

---

システム イベントの表示	18
<b>Spatial</b> のログ	19
メール サーバーの設定	21
有効期限通知を受信する項目の選択	22
バージョン情報の表示	23
ライセンス情報の表示とエクスポート	24
JMX コンソールによるパフォーマンスのモニタリング	24
JMX コンソールによる、ファイル ハンドル キャッシュ統計の監視	25
メモリ使用量の監視	26
<b>Spatial</b> モジュールを再起動する	27
MRR キャッシュのクリア	28

## システム イベントの表示

システム ログには、Spectrum™ Technology Platform サーバーの `spectrum-server.log` からのメッセージが表示されます。これらのメッセージには、サーバー操作や、API および Web サービスからサービスに対して作成されたリクエストに関する情報が含まれます。トラブルが発生し、考えられる原因についての情報を探す場合、システム ログを表示します。

Spectrum™ Technology Platform をクラスタで実行している場合、使用できるシステム ログはたまたま接続先となったノードで生成されたものです。特定のノードのシステム ログを表示するには、そのノードでファイル `ServerLocation\server\logs\spectrum-server.log` をテキスト エディタに開きます。

1. Management Console を開きます。
2. [システム] > [ログ] を選択します。
3. ダウンロード アイコンをクリックして  システム ログ ファイルをダウンロードします。
4. ダウンロードしたファイルをテキスト エディタで開きます。

## サービスのログ レベルの設定

デフォルトのログレベルや、システム上の各サービスのログレベルを指定することができます。ログレベルを変更しても、変更前に作成されたログ エントリにはその変更は反映されません。

注：サービスに対して指定するログレベルは、監査ログには影響しません。Management Console で表示できるイベント ログのログレベルのみを制御します。現時点では、Web 版の Management Console ではイベント ログを表示できません。

1. Management Console を開きます。
2. [システム] > [ログ] を選択します。
3. [システム デフォルト ログ レベル] フィールドで、システムのサービスに対するデフォルトのイベント ログレベルを選択します。

**無効**           すべてのイベント ログを無効にします。

**致命的**        最小限のログです。致命的なエラーのみがログに記録されます。致命的なエラーとは、システムを使用不可能にするエラーのことです。

エラー	エラーと致命的なエラーがログに記録されます。エラーとは、システムの一部が使用不可能になる単発的な問題を指します。例えば、1つのサービスが機能しなくなる問題によってエラーが生成されます。
警告	イベント警告、エラー、致命的なエラーがログに記録されます。警告とは、システムの動作を停止させることのない問題を指します。例えば、パラメータに無効な値が設定されているサービスをロードすると、警告が発行され、デフォルトのパラメータが使用されます。サービスの使用時に、結果は返ってきたが問題が存在するという場合に、警告がログに記録されます。
情報	抽象度の高いシステム情報がログに記録されます。これは、実稼働に適した最も詳細なログレベルです。通常、情報イベントは、起動や初期化の際に発生し、バージョン情報やロードされたサービスなどの情報を提供します。
デバッグ	システムの問題のデバッグ時に適した、非常に詳細なログレベルです。
トレース	プログラムの実行(メソッドの開始と終了)をトレースする、最も詳細なログレベルです。デバッグに使用する詳細なプログラムフロー情報を提供します。

各ログレベルは、1つ上のログレベルの内容を含みます。つまり、警告ログレベルが選択されている場合、エラーと致命的なエラーも記録されます。情報ログレベルが選択されている場合は、情報メッセージ、警告、エラー、および致命的なエラーが記録されます。

注：最も詳細なログレベルを選択すると、システムのパフォーマンスに影響が生じる恐れがあります。したがって、必要なログ要件を満たす最小レベルの設定を選択する必要があります。

- サービスごとに異なるログレベルを指定することもできます。

## Spatial のログ

logback.xml ファイルを使うと、ログの動作をより細かく制御できます。例えば、デフォルトでは出力はコンソールに送られ、wrapper.log にリダイレクトされますが、別のログファイルに出力を送ることができます。また、ログレベルも設定できるため、たとえばログを完全にオフにしたり、致命的なエラーのみログに記録したりできます。

デフォルトの logback ファイルは次のとおりです。

(<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>${gl.server.modules.dir}/spatial/${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>

```

オプション	値
レベル	<ul style="list-style-type: none"> <li>• OFF – ログをオフにします。</li> <li>• ERROR – 実行時エラーや予期せぬエラーをログに記録します。</li> <li>• WARN – 非推奨の API を使用したなどの警告のみをログに記録します。</li> <li>• INFO – 起動やシャットダウンなどの実行時のイベントをログに記録します。[デフォルト]</li> <li>• DEBUG – 詳細なデバッグ情報をログに記録します。</li> </ul>
出力	<ul style="list-style-type: none"> <li>• CONSOLE-SPATIAL – ログ情報をコンソールに送ります。[デフォルト]</li> <li>• FILE-SPATIAL – ログ情報をコンポーネント別にログファイルに記録します (現在の Spectrum Spatial には 1 つのリモートコンポーネントしかないため、このオプションは適用されません)。</li> </ul>

## メールサーバーの設定

Spectrum™ Technology Platform では、電子メールアラートを送信して重要なイベントを通知できます。電子メール通知は、データフローおよびプロセスフロー内の状況に従って送信したり、ライセンスやデータベースなどのアイテムが有効期限切れになりそうなタイミングで送信したりできます。

Spectrum™ Technology Platform にはメールサーバーが組み込まれていないので、電子メール通知を有効にするためには、外部の SMTP サーバーを使用できるように設定を行う必要があります。

1. Management Console を開きます。
2. [システム] > [メールサーバー] を選択します。
3. [ホスト] フィールドに、電子メール通知の送信に使用する SMTP サーバーのホスト名または IP アドレスを入力します。

4. **[ポート]** フィールドに、Spectrum™ Technology Platform サーバーと SMTP サーバーとの間のネットワーク通信で使用するポートの番号または範囲を入力します。  
デフォルトのポート番号は 25 です。
5. **[ユーザ名]** および **[パスワード]** フィールドに、Spectrum™ Technology Platform サーバーが SMTP サーバーでの認証に使用する資格情報を入力します。
6. **[送信者アドレス]** フィールドに、通知の送信元電子メール アドレスを入力します。
7. メールサーバーが正しく設定されていることを確認するためには、テスト用の電子メールを送信できます。テストメールの送信先電子メール アドレスを **[テスト アドレス]** フィールドに入力し、**[テスト]** をクリックします。
8. **[保存]** をクリックします。

これで、Spectrum™ Technology Platform サーバーが SMTP サーバーに接続され、SMTP サーバーを使用して通知メールを送信できます。

#### 例: メールサーバーの設定

mail.example.com という SMTP サーバーがあるとします。このメールサーバーを使用して、Spectrum™ Technology Platform サーバーから送信される電子メール通知を処理したいとします。この SMTP サーバー上には Spectrum123 というアカウントを Example123 というパスワードで作成してあります。このアカウントの電子メール アドレスは spectrum.notification@example.com です。

以上の情報に従って通知を設定するには、各フィールドに次のように入力します。

ホスト	mail.example.com
送信者アドレス	spectrum.notification@example.com
ユーザ名	Spectrum123
パスワード	Example123

## 有効期限通知を受信する項目の選択

通知を希望する項目が選択できるため、気になる項目についてのみ通知を受け取ることができます。

Spectrum™ Technology Platform は、ライセンス、データベース、ソフトウェア コンポーネントの有効期限が切れる前に電子メール通知を送付できます。これによって、期限切れによってビジネス プロセスに支障をきたすことなく、必要な措置を講じることができます。有効期限があるコンポーネントには、以下のようなものがあります。

- ライセンス
  - 電子メール通知は、トランザクションベースのライセンスに対しては提供されていません。ライセンスのトランザクション数が上限に近づいている場合は、**Management Console** のシステム ログにメッセージが表示されます。
  - **Spectrum Spatial™ Manager** に管理者としてログインした際に、ライセンス失効日までの期間が **Management Console** で設定されたライセンス有効期限の範囲内の場合、"Spatial ライセンスは <n> 日後に失効します。" という警告メッセージが表示されます。
- データベース (CASS 処理に使用される米国郵便データベースなど)
- 一部のソフトウェア コンポーネント (**Universal Addressing** モジュールにおいて米国住所の検証に用いられるエンジンなど)

ヒント：有効期限がある項目を参照するには、**Management Console** を開き、**[システム] > [ライセンスと有効期限]** を選択します。

1. **Management Console** を開きます。
2. **[システム] > [ライセンスと有効期限]** を選択します。
3. ある項目に対する有効期限通知メールを受信するには、**[通知を送る]** 列のチェックボックスをオンにします。

デフォルトよりも早く、または遅く通知を受け取りたい場合は、有効期限の何日前に通知を希望するかを、有効期限までの日数で指定します。

## バージョン情報の表示

1. Web ブラウザで次の URL を表示します。

`http://サーバー:ポート/managementconsole`

ここで **server** は、**Spectrum™ Technology Platform** サーバーの名前または IP アドレスで、**port** は **Spectrum™ Technology Platform** が使用する HTTP ポートです。デフォルトでは、HTTP ポートが 8080、HTTPS ポートが 8443 になっています。

2. **[システム] > [バージョン]** をクリックします。

## ライセンス情報の表示とエクスポート

ライセンスに関する情報を XML ファイルにエクスポートできます。この機能は、テクニカル サポートとともにライセンスの問題を解決する際に必要となる場合があります。

1. Web ブラウザで次の URL を表示します。

`http://サーバー:ポート/managementconsole`

ここで *server* は、Spectrum™ Technology Platform サーバーの名前または IP アドレスで、*port* は Spectrum™ Technology Platform が使用する HTTP ポートです。デフォルトでは、HTTP ポートが 8080、HTTPS ポートが 8443 になっています。

2. [システム] > [ライセンスと有効期限] をクリックします。
3. エクスポート アイコンをクリックします。

ライセンス情報は、拡張子が `.lic` である XML ファイルに保存されます。

## JMX コンソールによるパフォーマンスのモニタリング

JMX コンソールはブラウザベースのツールで、データフロー内の各ステージのパフォーマンス統計を記録するパフォーマンス モニタリング ツールを提供します。

1. Web ブラウザを開いてに移動します。 `http://server:port/jmx-console`

説明:

*server* は、Spectrum™ Technology Platform サーバーの IP アドレスまたはホスト名です。

*port* は、Spectrum™ Technology Platform が使用する HTTP ポートです。デフォルト値は 8080 です。

2. 管理者アカウントでログインします。
3. "Domain: com.pb.spectrum.platform.performance" の下にある **com.pb.spectrum.platform.performance:service=PerformanceMonitorManager** をクリックします。
4. **[enable]** の横にある **[Invoke]** ボタンをクリックします。
5. **[Return to MBean View]** をクリックして、**[PerformanceMonitorManager]** 画面に戻ります。



これでパフォーマンス モニタリングが有効になります。データフローを実行すると、パフォーマンス統計が [PerformanceMonitorManager] 画面の上部に表示されます。次のことに注意してください。

- 更新を確認するには、画面を再表示する必要があります。
- カウンタをリセットするには、**[reset]** の横にある **[Invoke]** ボタンをクリックします。
- Spectrum™ Technology Platform サーバーを停止すると、パフォーマンス モニタリングはオフになります。サーバーを再起動したら、パフォーマンス モニタリングを再度オンにする必要があります。

## JMX コンソールによる、ファイル ハンドル キャッシュ 統計の監視

JMX コンソールは、ネイティブ TAB およびシェイプファイルのファイル ハンドル キャッシュ統計を含む、パフォーマンスやレコード統計を監視するブラウザベースのツールです。

1. Web ブラウザを開いてに移動します。 `http://server:port/jmx-console`

説明:

*server* は、Spectrum™ Technology Platform サーバーの IP アドレスまたはホスト名です。

*port* は、Spectrum™ Technology Platform が使用する HTTP ポートです。デフォルト値は 8080 です。

2. 管理者アカウントでログインします。
3. [Domain: Spatial] の下で、**[Spatial:name=TABFileHandlePool,type=Remote Component]** または **[Spatial:name=ShapeFileHandlePool,type=Remote Component]** をクリックして、TAB ファイルまたはシェイプファイルのファイル ハンドル キャッシュ統計を表示します。

注: このページ上で、サーバーを再起動することなく、ファイル ハンドル キャッシュを無効にするか、クリアすることもできます。

4. **[すべての MBean]** をクリックして、メインの JMX コンソールに戻ります。

## メモリ使用量の監視

JMX コンソールでは、空間リモート コンポーネントの JVM ヒープ使用量を監視できます。

### Spectrum Technology Platform JMX Console

**MBean: Spatial:name=Process,type=Remote Feature Component**  
**Description: The Managed Bean of Remote Feature Component containing FeatureService**  
**UserManagementService for process monitoring**

#### Attributes

Name	Value
HeapMemoryUsage	javax.management.openmbean.CompositeDataSupport(compositeType=javax.management.openmbean.CompositeType(name=((itemName=committed,itemType=javax.management.openmbean.SimpleType(name=java.lang.Long)),(itemName=init,itemType=javax.management.openmbean.SimpleType(name=java.lang.Long)),(itemName=max,itemType=javax.management.openmbean.SimpleType(name=java.lang.Long)),(itemName=used,itemType=javax.management.openmbean.SimpleType(name=java.lang.Long)))),contents={committed=994639 used=20588920})
NonHeapMemoryUsage	javax.management.openmbean.CompositeDataSupport(compositeType=javax.management.openmbean.CompositeType(name=((itemName=committed,itemType=javax.management.openmbean.SimpleType(name=java.lang.Long)),(itemName=init,itemType=javax.management.openmbean.SimpleType(name=java.lang.Long)),(itemName=max,itemType=javax.management.openmbean.SimpleType(name=java.lang.Long)),(itemName=used,itemType=javax.management.openmbean.SimpleType(name=java.lang.Long)))),contents={committed=45088 used=44770368})
RuntimeName	1252@spectrum8-sp2

#### Operations

Name	Return type	Description
------	-------------	-------------

メモリ使用量 (HeapMemoryUsage と NonHeapMemoryUsage) は、標準の JVM メモリ MBean をベースとしています。リモート コンポーネントを実行している JVM のメモリ使用量を表します。この中では、init、max、committed、used の各メモリ容量が示されます。

RuntimeName にはプロセス ID が含まれているため、オペレーティングシステムで詳細情報を確認したり (たとえば Windows のタスク マネージャーを使用)、プロセスを停止するために使用できます。

ヒープ セクションの {committed=143130624, init=134217728, max=1908932608, used=23483928} はバイト単位の値です。

init は JVM が最初に割り当てる容量で、-Xms で指定する値に相当します。max は -Xmx で指定する値に相当します。used は JVM がオブジェクトに使用しているメモリ容量です。これらの関連性を表すと、-Xms < committed < -Xmx および used < committed となります。

ヒープメモリを変更するには、spatial フォルダにある java.vmargs ファイル (<インストール先>\Pitney Bowes\Spectrum\server\modules\spatial\java.vmargs) で -Xm を変更します。詳細については、「[ヒープメモリの増加](#)」を参照してください。

## Spatial モジュールを再起動する

Spectrum Spatial™ Manager を使って Spatial モジュールのリモート コンポーネントを再起動すると、キャッシュをクリアしたり、Spectrum™ Technology Platform を再起動せずに新しい設定を有効にしたりすることができます。

### Spatial モジュールを再起動する方法

1. Spectrum Spatial™ Manager を起動します。
2. **[設定]** をクリックしてから **[設定]** サブメニューを開きます。
3. **[Spatial モジュールを再起動します]** の **[再起動]** をクリックします。警告メッセージが表示されます。

注：一部のリクエストが再起動中にエラーになることがあるため、トラフィックがない状態で再起動を行うことをお勧めします。

4. **[はい]** をクリックして再起動します。

### Spatial モジュールの再起動が便利なケース

1. Spectrum Spatial では、頻繁には更新されないテーブル構造やジオメトリ、インデックス関連情報がキャッシュに記録されます。テーブル内のデータはおおむね静的ですが、ときには更新されます。新しいデータを展開するときは、Spatial モジュールを再起動してキャッシュをクリアすることをお勧めします。Spectrum Spatial™ Manager でテーブル変更機能を使うと、テーブルの可変設定を更新できます。

データソースの可変性を理解するには、[ここをクリック](#)してください。

注：

- TAB ファイルやシェイプファイルのようなファイルベースのテーブルの場合、テーブル構造やデータが変更されたときに Spatial モジュールを再起動します。
- データベース テーブルの場合は、テーブル構造が変更されたときにだけ、Spatial モジュールを再起動します。

## MRR キャッシュのクリア

MRR ファイルは、ハンドルがキャッシュ内に開かれている間はロックされるため、ファイルへの更新、削除、または置換の操作は阻止されます。MRR のネイティブ ハンドルを解放するため、開かれたすべてのハンドルを手動で閉じるオプションが JMX コンソールに用意されています。

1. Web ブラウザを開いて に移動します。 `http://server:port/jmx-console`

説明:

`server` は、Spectrum™ Technology Platform サーバーの IP アドレスまたはホスト名です。

`port` は、Spectrum™ Technology Platform が使用する HTTP ポートです。デフォルト値は 8080 です。

2. 管理者アカウントでログインします。
3. [ドメイン: 空間] で、**Spatial:name=MRRCache,type=Remote Component** をクリックします。
4. **closeAll** 操作の横にある **Invoke** ボタンをクリックして、開かれたすべての MRR ハンドルを閉じます。

呼び出しのステータスに関するメッセージが表示されます。

注：ハンドルがレンダリングのために並行して使用されていた場合、強制的に閉じると例外が生成されます。このような例外は無視されますが、ログに情報は残ります。

# 4 - パフォーマンス チューニング

このセクションでは、メモリ管理とスレッド処理によってパフォーマンスを向上する方法について説明します。また、Location Intelligence モジュールのパフォーマンスを最適化するためのベスト プラクティスを紹介します。経験豊富な管理者向けの内容です。

Spectrum では、サーバーのパフォーマンスを最適化するためのいくつかのチューニング オプションが提供されています。どの設定が最適であるかは、展開の性質に依存します。適切にチューニングされたサーバー環境を構築するために、展開済み環境においてパフォーマンス テストを実施して、最適な設定を判断することを推奨します。このセクションでは、パフォーマンス チューニングに関する一般的な指針を示します。

## このセクションの構成

---

リモート コンポーネントの構成設定	30
データ ソース プーリングの設定	31
距離ベースの操作のパフォーマンスの向上	31
ACL プロパティの管理	32

## リモート コンポーネントの構成設定

Spectrum™ Technology Platform のすべての空間サービスは、プラットフォーム ランタイムから独立したリモート コンポーネント (JVM インスタンス) に展開されます。これによって、プラットフォームはそれに含まれるモジュールに依存せず、また、JVM 構成は空間サービスごとに適用可能となるため、メモリ割り当ての柔軟性が高まり、サービスの特性に応じたパフォーマンス チューニングが可能になります。

リモート コンポーネントは、空間サービス (Feature Service や Mapping Service など) とステージ (Spatial Calculator や Query Spatial Data など) に空間機能を提供します。リモート コンポーネントのプール サイズは、コンポーネントが同時に処理できる要求の数です。この設定は、空間サービスと空間ステージの両方のスループットに影響を与えます。

空間リモート コンポーネントの権限を管理するには、他のセキュア エンティティ タイプと同じ方法で Management Console を使用します。空間リモート コンポーネントは、"Spatial Component" セキュア エンティティ タイプとして [データベース リソース] グループに表示されます。空間リモート コンポーネントの権限は、役割の作成時または編集時に設定したり、アクセス制御設定を使用して設定することができます。詳細については、[#unique\\_35](#)を参照してください。

## プール サイズの変更

JVM の調整に加えて、空間リモート コンポーネントのプール サイズも調整できます。リモート コンポーネントのプール サイズは、コンポーネントが同時に処理できる要求の数です。この設定は、Spectrum™ Technology Platform からのサービス要求を待機したり、Location Intelligence モジュールのステージを実行したりするコンポーネント上のスレッド数 (つまり管理対象の接続の最大数) を表します。

すべての Web サービスリクエストは、プラットフォームから Spectrum に入り、コンポーネントへと引き渡されます。デフォルト値の 1 より大きくすることで、高いリクエスト負荷に対応できます。CPU の数に一致するプール サイズが推奨されます。この最大値を、CPU コア数の 2 倍以上に設定しないでください。例えば、CPU が 4 基のマシンでは、全サービスのスレッドの合計数が 8 を超えないようにします。使用状況に対して最適なパフォーマンスが得られるように、さまざまな設定でパフォーマンス テストを実施する必要があります。

空間リモート コンポーネントのプール サイズは Management Console で調整できます。

1. Management Console を開きます。
2. [リソース] > [Location Intelligence] に移動します。

3. 矢印を使うか値を入力して、リモート コンポーネントのプールサイズを変更します。最小値は 1 で、最大値は 64 です。
4. **[保存]** をクリックします。
5. プールサイズを小さくした場合は、サーバーを再起動します。プールサイズを増やした場合はすぐに有効になるので、サーバーを再起動する必要はありません。

## データ ソース プーリングの設定

pooling-datasource-factory.properties ファイル (\server\modules\spatial にあります) を使用して、JDBC ベースのデータ ソース (Oracle、SQL Server など) で使用される接続のプーリングを、最適なパフォーマンスが得られるように設定できます。

多くの場合は、`validateOnBorrow=true` に設定し、接続の検証を有効にすることをお勧めします。これにより、プールから借用する前にオブジェクトを検証できます。検証に失敗した場合は、その接続がプールから破棄され、別の接続の借用を試みます。検証クエリは、カスタムデータプロバイダを使用する場合など、特殊なケースに対しても使用できます。検証クエリが有効になっている場合は、検証クエリが使用されます。

検証を有効にすると、パフォーマンスが若干低下する可能性があります。ただし、このテストクエリにより、Spectrum Spatial と外部データベース間の通信に信頼性がない場合でも、コネクションプール内のすべての接続の整合性が維持されます。検証によるパフォーマンスへの影響が少なくなるように、検証間隔を設定します。ある接続を検証するタイミングになっても、この間隔内に検証が既に済んでいる場合は、その接続が再び検証されることはありません。

## 距離ベースの操作のパフォーマンスの向上

PGD インデックス ファイルは、TAB ファイル セットを補足するファイルであり、ネイティブ、ネイティブ拡張 (NativeX)、およびシームレス TAB で GSB ファイルに匹敵するパフォーマンスを発揮することができます。ラインとポリゴンを含むネイティブ データセットに対する、特定の距離ベースの操作のパフォーマンスを向上させるために、この特殊なインデックス ファイルを作成する、PGD Builder というコマンドライン ユーティリティが用意されています。PGD Builder を使用して作成したインデックスは、ラインとリージョンに基づくデータを検索していて、以下を使用している場合に有効です。

- Point in Polygon ステージで、距離を含める場合
- Find Nearest ステージで、(距離を含めるかどうかにかかわらず) 入力ポイントである場合

- Feature サービスの SearchNearest 操作で、入力ポイントと、ラインまたはポリゴンの検索テーブルを使用する場合

PGD Builder ユーティリティは、ようこそページの [Spectrum Spatial] セクションにある、[ユーティリティ] タブの [PGD Builder] からダウンロードできます。ドキュメントへのリンクも、ようこそページのユーティリティ ダウンロード リンクの横にあります。

注：PGD ファイルは、TAB 用の .MAP ファイルの 5 ～ 6 倍のサイズになります。1 つの TAB ファイルに対して 1 つの PGD ファイルが生成されます。ただし、シームレス TAB の場合は、各サブ TAB に対して PGD ファイルが作成されます。

また、TAB 内のデータを変更すると (行の追加や削除、または、TAB の MAP 部分のジオメトリの変更)、PGD ファイルはシステムによって使用されなくなります。警告が有効である場合は (「[Spatial のログ \(19ページ\)](#)」を参照)、有効期限切れの PGD ファイルに関するメッセージが、wrapper.log、または、Spatial のログ用に設定されたログ ファイルがある場合はそれに表示されます。メッセージが表示された場合は、更新された TAB ファイルに対する PGD を再生成する必要があります。

## ACL プロパティの管理

起動に時間がかかり、アクセス制御リスト (ACL) が不要なチェックを行うのを避けるために、必要に応じて、**spatial.security.acl.enable** プロパティと **spatial.security.acl.check** プロパティを **true** または **false** に設定できます。このようにすると、これらのチェックを実行するかしないかを制御できます。これらのプロパティ

は、%Spectrum%\server\modules\spatial\java.properties に配置されています。

注：プロパティの変更後に、サーバーを再起動する必要があります。

次は、これらの新しいプロパティの説明です。

- **spatial.security.acl.enable**: システム管理者が Spatial リポジトリ内の名前付きリソースに対して権限を使用しない場合は、このプロパティを **false** に設定して ACL チェックを無効にすることができます。すべてのユーザがすべての名前付きリソースに完全な権限を持つので、このプロパティを **false** に設定すると、起動時に行われる Spatial リポジトリとアクセス制御データベースとの一貫性チェックがスキップされます。これにより、処理が速くなります。
- **spatial.security.acl.check**: システムの起動時に Spatial は一貫性チェックを実行します。Spatial リポジトリに含まれる名前付きリソースの数が多いと、このチェックに時間がかかります。管理者は、このプロパティを **false** に設定して、このチェックをスキップし、サーバーの起動時間を縮めることができます。



# 5 - クラスタの管理

## このセクションの構成

---

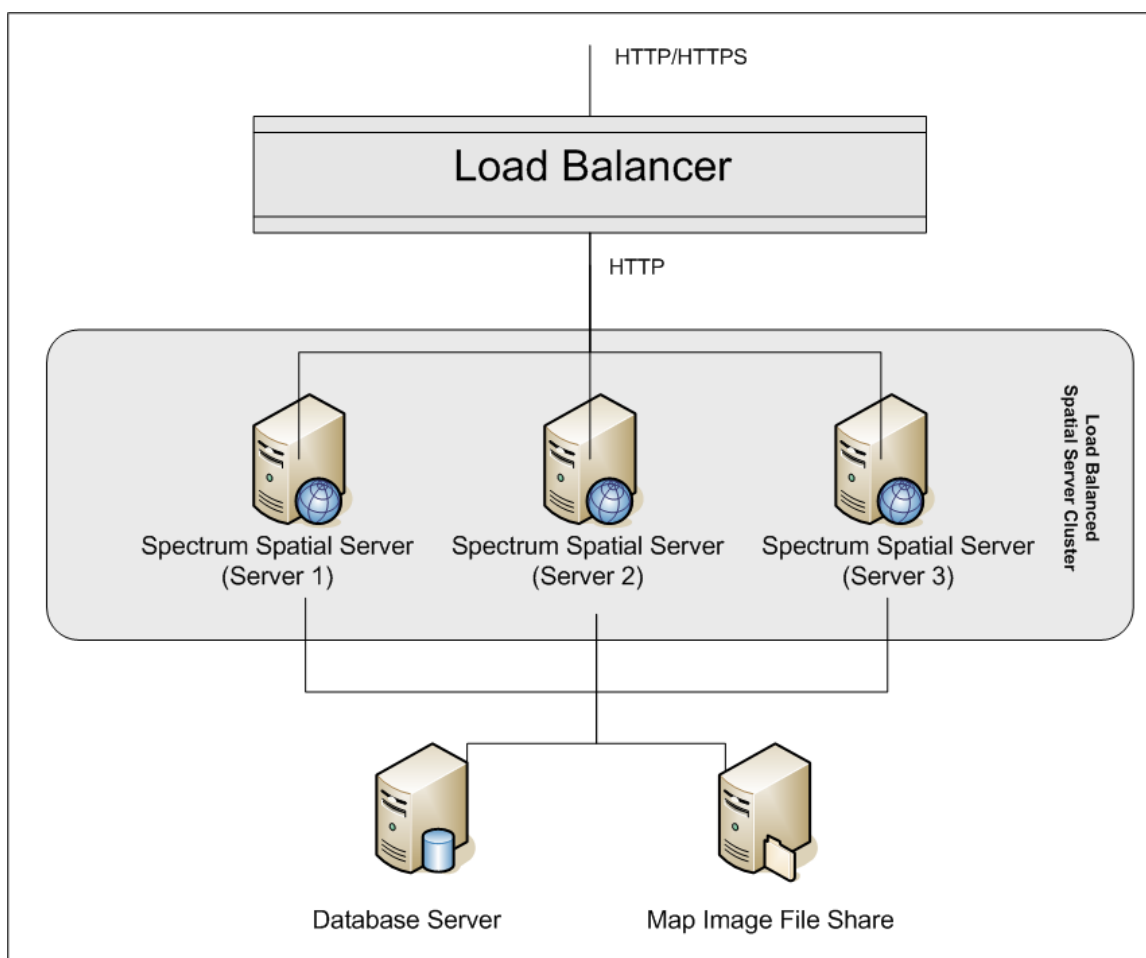
Spatial モジュールのクラスタ アーキテクチャ	34
クラスタでの Enterprise Designer の使用	36
クラスタの起動	37
クラスタの停止	37
クラスタからのノードの削除	38
Spatial モジュール用のクラスタの管理	39

## Spatial モジュールのクラスタ アーキテクチャ

クラスタ環境では、サーバーの複数のインスタンスが処理を共有します。以下の図は、このような構成の展開アーキテクチャを示したものです。負荷分散を利用することによって、高い可用性とスケーリングをサポートすることができます。展開アーキテクチャは、ロード バランサー (負荷分散サーバー)、Spectrum Spatial クラスタ、データベース、ファイル共有で構成されます。このアプローチによって、水平および垂直の両方のスケーリングが可能になります。Spatial モジュールは、プラットフォームをクラスタリングしてもしなくてもクラスタ化できます。

**注：** Spectrum™ Technology Platform クラスタと Location Intelligence モジュール クラスタの両方を設定することを推奨します。これには、以下に示すいくつかの利点があります。

- 名前付きリソースのセキュリティ (ACL) 同期が自動的に行われます。
- 1つのノード上で作成されたデータフロー、ユーザ、役割が、すべてのノードに自動的に同期されます。
- Location Intelligence モジュールのすべてのデモ ページとユーティリティ (Spectrum Spatial™ Manager など) がロード バランサーを参照できます。



### ロード バランサー

ロードバランサーは、Spectrum Spatial インスタンス間にリクエストを分散します。HTTP/HTTPS リクエストの負荷分散をサポートする任意のロード バランサーを使用できます。

### Spectrum Spatial クラスタ

このクラスタは、Spatial モジュール共有管理、名前付きリソース、地理メタデータコンテンツ、および構成設定を伴う Spectrum インスタンスのコレクションです。クラスタにさらにノードを追加することによって、耐障害性を高めたり、さらに高い負荷に対応したりすることができます。各ノードは、ハードウェアリソースを追加するか、または、大規模リソースを伴うハードウェアに対して必要である場合はインスタンスを追加することによって、垂直にスケーリング可能です。Spectrum は、限られた数の CPU を使用するように構成できます。

### データベース

Spectrum は名前付きリソース(マップ、レイヤ、テーブル、スタイル)、地理メタデータ、構成をリポジトリに保存します。デフォルトのシングルサーバー構成では、組み込みデータベースを使

用して、ローカルサーバー上にこれらのリソースを保存します。耐障害性に優れたスケラブルなソリューションを構築するには、この組み込みデータベースを耐障害性を備えた独立したデータベースに置き換える必要があります。Oracle、PostgreSQL/PostGIS、Microsoft SQL Server が、リポジトリ データベースとしてサポートされています。

負荷分散構成において、Spectrum ノードはローカルキャッシュにこれらのリソースをキャッシュし、クラスタの各ノード内のインデックスを検索します。Spectrum ノードはリクエストを受信すると、ローカル キャッシュとインデックスを使用してリソースを検索します。名前付きリソースは、クラスタ内の任意のノードを介して追加できます。各ノードは、ローカル キャッシュと中央データベースとの違いをチェックして、キャッシュを最新の状態に保ちます。このチェックは、デフォルトで2秒ごとに実行されます。チェック時間の間隔は設定可能です。このアーキテクチャによって、サーバーは高いパフォーマンスでトランザクションを処理することができ、リポジトリ データベースの負荷は最小限に抑えられることとなります。新しい Spectrum ノードがクラスタに追加されると、キャッシュとインデックスが自動的に作成されます。新しい Spectrum の追加は、障害が発生したノードを復旧したい場合や展開のキャパシティを拡大したい場合に、行われる可能性があります。

### ファイル共有

ファイル共有は、Spectrum によって生成されたマップ イメージを格納するためのフォルダを提供します。Web サービスを使用してマップをレンダリングする際、サーバーは、URL を介して返されるマップ イメージまたは base 64 エンコード イメージとして返されるマップ イメージをサポートします。URL が返される場合、マップ イメージはファイルとして保存され、その URL がリクエストされた場合に提供されます。任意の Spectrum ノードがマップ イメージを返せるようにするために、イメージの保存にファイル シェアが使用されます。

## クラスタでの Enterprise Designer の使用

1. Enterprise Designer を起動します。
2. **[サーバー名]** フィールドに、ロード バランサーのサーバー名を入力します。
3. **[ポート]** フィールドに、ロード バランサーがリッスンするように設定したポートを入力します。

注：入力ファイル、出力ファイル、およびデータベース リソースは、共有ドライブ、ファイルサーバー、またはその他の共通にアクセスできる場所に置く必要があります。そうしない場合は、Spectrum™ Technology Platform サーバーをホストする各サーバーにすべてのファイルをロードする必要があり、すべてのファイルが同じパスに存在する必要があります。

ログインした後は、普通に Enterprise Designer を使用できます。実行するアクションは、ログインしているクラスタのすべての Spectrum™ Technology Platform インスタンスに対して適用されます。

## クラスタの起動

次の手順は、サーバーが停止されていることを前提にしています。

クラスタ内のすべてのノードが停止した場合は、次の手順に従ってクラスタを安全に起動し、データの損失を回避する必要があります。

最後に停止したノードで、サーバーを起動します。これを、クラスタ内の各ノードで行います。

**警告：**最新のデータを維持するために、最初に起動するノードは、最後に停止したノードでなければなりません。別のノードを最初に起動すると、ジョブ履歴や構成設定などのデータが失われる恐れがあります。最後に停止したノードが不明な場合は、各ノードのログでシャットダウンメッセージのタイムスタンプを確認します。ログは、次の場所にあります。

`SpectrumDirectory\server\logs\spectrum-server.log`

- a) サーバーを開始します。
- b) アップグレードの後に、すべてのノードを続けて起動します。ノード1の起動後数秒以内にノード2を起動します。残りのノードもこのタイミングで起動してください。

Spectrum™ Technology Platform サーバーが完全に起動したかどうかは、次のログファイルで確認できます。`SpectrumDirectory\server\logs\spectrum-server.log`サーバーが完全に起動している場合は、次のメッセージが表示されます。

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

## クラスタの停止

クラスタ全体を停止するには

1. シードノードとして使用されているノードを識別します。これを行うには、`SpectrumDirectory/server/conf/spectrum-container.properties` ファイルを開き、`spectrum.cluster.seeds` プロパティに示される一連のノードに注目します。

2. クラスタに含まれる Spectrum™ Technology Platform サーバーを 1 つずつ停止します。最後に停止するノードが必ずシード ノードになるようにします。
3. 作業ディレクトリを Spectrum™ Technology Platform サーバーの bin ディレクトリに変更し、セットアップファイルのソースを指定して、コマンド `./server.stop` を入力します。

**警告：** クラスタの起動時にデータが失われるのを防ぐため、最後に停止したノードを最初に起動する必要があります。また、そのノードはシード ノードである必要があります。

4. 停止した最後のノードをメモに書き留めておいてください。この情報は、クラスタの起動時に必要になります。
5. Windows システムトレイの Spectrum™ Technology Platform アイコンを右クリックして、**[Spectrum™ を停止する]** を選択します。

**警告：** クラスタの起動時にデータが失われるのを防ぐため、最後に停止したノードを最初に起動する必要があります。また、そのノードはシード ノードである必要があります。

## クラスタからのノードの削除

クラスタからノードを削除するには、Spectrum™ Technology Platform サーバーを停止します。

1. 削除するノードを、次の手順で停止します。
  - Unix または Linux では、作業ディレクトリを Spectrum™ Technology Platform サーバーの bin ディレクトリに変更し、セットアップファイルのソースを指定して、コマンド `./server.stop` を入力します。
  - Windows の場合は、システムトレイの Spectrum™ Technology Platform アイコンを右クリックして、**[Spectrum™ を停止する]** を選択します。
2. ファイル `server/app/conf/spectrum-container.properties` をテキスト エディターで開き、`spectrum.cluster.enabled` を `false` に設定します。
3. クラスタ内の他の各ノード上で、ファイル `spectrum-container.properties` を開き、`spectrum.cluster.seeds` プロパティからノードを削除します。

**Location Intelligence モジュールを使用する場合：** ノードをスタンドアロンのままにして、クラスタの外部で実行できるようにするには、元の `repository.xml` ファイルをコピーして戻し、Spectrum™ Technology Platform の各インスタンスの `/server/modules/spatial/jackrabbit` ディレクトリから、`repository`、`version`、`workspaces` というフォルダを削除します。サーバーを再起動し、リポジトリのコンテンツをインポートします。

## Spatial モジュール用のクラスタの管理

### 共通リポジトリ データベースのセットアップ

クラスタに対して共通リポジトリ データベースを使用するように、Location Intelligence モジュールを設定する必要があります。これによって、名前付きリソース、地理メタデータ、構成設定が、クラスタ全体で管理されるようになります。

リポジトリは、一連の名前付きリソース、地理メタデータ、構成ファイルとともにインストールされています。これらのリソースを共通データベース リポジトリに移行するには、リソースをデフォルトの内部リポジトリ データベースから新しい共有リポジトリ データベースへとエクスポートする必要があります。

リポジトリ コンテンツを一括でエクスポートまたはインポートする場合は `limrepo import`、管理ユーティリティの `limrepo export` コマンドと `limrepo import` コマンドを使用します。これらのコマンドには、権限を維持するオプションがあります (手順については『*Spectrum Spatial ガイド*』の「管理」セクションを参照してください)。

PostgreSQL、Oracle、または Microsoft SQL Server のいずれかの共通データベースにリポジトリをセットアップするには、次の手順を実行します。

1. 管理ユーティリティの `limrepo export` コマンドを使用して、すべてのリポジトリ リソースをローカルフォルダにエクスポートします (手順については『*Spectrum Spatial ガイド*』の「管理」セクションを参照してください)。

インストールしたリポジトリのコンテンツをエクスポートする必要があります。このステップは 1 回だけ実行します。Spectrum™ Technology Platform のすべてのインスタンスに対し、この時点でのリポジトリのコンテンツを同一にするためです。

2. すべてのノード上で Spectrum™ Technology Platform サーバーを停止します (手順については [クラスタの停止](#) (37ページ) を参照してください)。
3. Spectrum™ Technology Platform のすべてのノード上で、共通データベースを指定するように設定を変更します。
  - a) `repository.<databaseType>.xml` の内容を、`repository.xml` にコピーします。このファイルは `server/modules/spatial/jackrabbit` フォルダにあります。ここで `<databaseType>` は、お使いのデータベースに対する適切なタイプです (`postgres`、`oracle`、または `mssql`)。
  - b) `repository.xml` で、次の処理を行います。

- **DataSource** セクションを、サーバーのホスト名、ポート、データベース、ユーザ、パスワードで変更します。
  - **Cluster** セクションを変更して、**Node1** のような個別のクラスタ ID を割り当てます。クラスタ内の後続のすべてのノードに一意的 ID を割り当てます (**Node2**、**Node3** など)。
  - 変更を `repository.xml` に保存します。
- c) `/server/modules/spatial/jackrabbit` フォルダから、`repository`、`version`、`workspaces` というフォルダを削除します。
4. データベースにこれまでにリポジトリ コンテンツが含まれていたことがある場合は、クリーンなリポジトリが作成できるようにデータベースからテーブルを削除しておく必要があります。以下のテーブルを削除する必要があります。

<code>default_binval</code>	<code>security_binval</code>
<code>default_bundle</code>	<code>security_bundle</code>
<code>default_names</code>	<code>security_names</code>
<code>default_refs</code>	<code>security_refs</code>
<code>rep_fsenry</code>	<code>version_binval</code>
<code>rep_global_revision</code>	<code>version_bundle</code>
<code>rep_journal</code>	<code>version_names</code>
<code>rep_local_revisions</code>	<code>version_refs</code>

Oracle を使用している場合は、`version_seq_names_id`、`security_seq_names_id`、`default_seq_names_id` も削除します。

5. シード ノード上にも、バックアップしたリポジトリ コンテンツをインポートします。
- a) Spectrum™ Technology Platform サーバーを起動します (手順については「[クラスタの起動 \(37ページ\)](#)」を参照してください)。
  - b) `limrepo import` コマンドでシード ノードを指定して、コンテンツをインポートします。
6. クラスタ内の残りのノードを起動します (手順については「[クラスタの起動 \(37ページ\)](#)」を参照してください)。



## システムの設定

Spectrum™ Technology Platform をインストールして共通リポジトリを設定した後、別の仮想マシンに複製する前に、インスタンスを構成設定する必要があります。仮想マシン環境を使用していない場合は、インストールされている各 Spectrum™ Technology Platform に対して以下の手順を実行する必要があります。

### マップ ファイル共有の設定

Spectrum™ Technology Platform にマップ ファイル共有 (共有イメージフォルダ) を設定するにはまず、共有マップ イメージ ディレクトリが必要です。マップ ファイル共有の作成については、「[Unix/Linux 上でのマップ イメージ ファイル共有の作成 \(42ページ\)](#)」または「[Windows 上でのマップ イメージ ファイル共有の作成 \(42ページ\)](#)」を参照してください。

マップ イメージ ディレクトリを作成したら、以下の手順でマップ ファイル共有を設定します。

1. マッピングサービス構成を、共有イメージフォルダと負荷分散サーバーを使用するように変更します。ImageCache において、Directory パラメータを共通イメージ ディレクトリに変更し、AccessBaseURL パラメータをロード バランサー マシンのイメージ URL に変更します。

仮想マシン環境を使用している場合は、この IP アドレスを覚えておいてください。ロード バランサー VM をこの IP に設定する必要があるためです。

Unix/Linux インストールの場合:

```
<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>
```

Windows インストールの場合:

```
<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. **Unix/Linux** インストールの場合は、マップ イメージを共有ファイル システムに配置できるようにシンボリック リンクを設定する必要があります。

マウントされた共有フォルダに `images` サブフォルダを作成します。例: `/mnt/<linux mount>/images`。

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

### Unix/Linux 上でのマップ イメージ ファイル共有の作成

ファイル共有は、**Spectrum Spatial** によって生成されたマップ イメージを格納するためのフォルダを提供します。すべての **Spectrum** ノードからアクセス可能な共有フォルダを作成します。マップが **Base64** エンコード イメージとして **Web** サービスから返される場合は、ファイル共有は必要ありません。

Unix/Linux 上でのマップ イメージ ファイル共有を作成するには

1. **Spectrum** をホスティングする各オペレーティングシステム上で、共有フォルダをマウントします。以下のコマンドによって、**Microsoft Windows Server** または **CIFS** をサポートするネットワーク ドライブ上にドライブをマウントします。

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

2. `/etc/fstab` において、起動時にイメージ シェアをロードするように設定します。

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

### Windows 上でのマップ イメージ ファイル共有の作成

ファイル共有は、**Spectrum Spatial** によって生成されたマップ イメージを格納するためのフォルダを提供します。すべての **Spectrum** ノードからアクセス可能な共有フォルダを作成します。マップが **Base64** エンコード イメージとして **Web** サービスから返される場合は、ファイル共有は必要ありません。

Windows 上でのマップ イメージ ファイル共有を作成するには

1. **Windows** エクスプローラで、共有するイメージ フォルダを選択します。
2. 右クリックして **[共有]** をクリックします。

3. イメージフォルダに書き込みを行うユーザを選択します。これらのユーザは読み取り/書き込み権限を持つ必要があります。

### クラスタ用の OGC サービス設定の変更

Spectrum™ Technology Platform クラスタと Location Intelligence モジュール クラスタの両方がある場合のクラスタ環境が正しく動作するように、Spectrum Spatial™ Manager によって OGC サービス設定ファイルを変更する必要があります。WFS、WMS、WMTS の設定ページで、オンラインリソース (サービス) URL をロード バランサーの IP アドレスおよびポートに変更します。詳細については、『Spectrum Spatial ガイド』の「ユーティリティ」セクションにある「Spectrum Spatial™ Manager ガイド」を参照してください。

### 複数の Spectrum インスタンスのポートの構成

単一のマシンに複数の Spectrum™ Technology Platform インスタンスがある場合は、各インスタンスのポート番号を変更する必要があります。<Spectrum root>/server/conf/spectrum-container.properties 内のすべてのポートを、使用されていない新しいポートに変更します。この HTTP ポートは、インストーラで入力されたポート番号を示しています。

### 共有される Spectrum ローカル データ

ファイル システム上の TAB ファイル データを使用する場合は、このデータを負荷分散環境に含まれるすべての Spectrum インスタンスがアクセス可能な共有場所に配置する必要があります。また、ファイル システム上のデータにアクセスするリポジトリ内のすべての名前付きリソースが、この共有場所を指す必要があるということも、重要な点です。

Spectrum をホスティングする各 VM またはマシンは、マウントされた共有ドライブにアクセスできる必要があります。

注：データベース テーブルを指す名前付きリソースを使用するのに、共有ドライブは必要ではありません。リポジトリ内の名前付きリソースは、ファイル パスを使用してデータにアクセスするのではなく、データベース内のデータへの名前付き接続を使用するためです。

# 6 - 管理ユーティリティの使用

## このセクションの構成

---

管理ユーティリティを使用する前に	45
管理ユーティリティをスクリプトから使用する	46
Spatial モジュール	47
ルーティング	54

## 管理ユーティリティを使用する前に

管理ユーティリティでは、管理機能をコマンドラインから実行できます。コマンドの実行は対話的に行うことも、スクリプトで行うこともできます。一部の管理機能は、管理ユーティリティでは利用できません。以下の機能は、**Management Console** を使用して実行できます。

注： The Administration Utility requires Java 8 or later. Verify that Java 8 is in the system's path before running the Administration Utility.

1. Spectrum™ Technology Platform の **[ホーム]** ページで **[プラットフォーム クライアント ツール]** をクリックします。
2. **[コマンドライン]** をクリックします。
3. **[管理ユーティリティ]** で **[ダウンロード]** をクリックして、管理ユーティリティを使用するコンピュータに zip ファイルをダウンロードします。
4. zip ファイルの内容を解凍します。
5. コマンドライン インターフェイスを起動するには、次のいずれかの操作を実行します。
  - サーバーが Unix または Linux システムで実行されている場合は、`cli.sh` を実行します。
  - サーバーが Windows システムで実行されている場合は、`cli.cmd` を実行します。

注： 必要に応じて、Java のインストールパスが正しく使用されるように `.sh` または `.cmd` ファイルを変更します。

6. 次のコマンドを入力して Spectrum™ Technology Platform サーバーに接続します。

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

例を次に示します。

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

7. 接続が完了すると、コマンドを実行できます。コマンドのヒントを紹介します。
  - 使用できるコマンドの一覧を表示するには、`help` と入力するか、**Tab** キーを押します。
  - コマンドを自動補完するには、最初の数文字を入力してから **Tab** キーを押します。例えば、`us` と入力してから **Tab** キーを押すと、自動補完されて `user` コマンドになります。もう一度 **Tab** キーを押すと、`user` コマンドの全一覧が表示されます。
  - 空白を含むオプション値を指定する場合は、値を二重引用符で囲みます。
8. 作業が終わったら、`exit` コマンドを入力して管理ユーティリティを終了します。

## 管理ユーティリティをスクリプトから使用する

管理ユーティリティのいくつかのコマンドは、スクリプト ファイルから実行できます。これは、管理ユーティリティや **Management Console** でコマンドを手動で実行する代わりに、スクリプトを使用して管理上のアクションを自動化または規格化したい場合に便利です。

1. テキストエディタでスクリプトファイルを作成します。スクリプトファイルには、実行したいコマンドを記述します。

コマンドをスクリプト ファイルに追加するには、コマンドプロンプトでコマンドを入力する場合と同様にコマンドと必要なパラメータを入力します。コマンドは、1行に1つ入力します。

スクリプト ファイルにコメントを挿入する場合は、次の表記方法を使用します。

- `/*`            コメント ブロックの開始を示します。
- `*/`            コメント ブロックの終了を示します。
- `//`            インライン コメントを示します。行の先頭のみを使用します。
- `;`             インライン コメントを示します。行の先頭のみを使用します。

2. スクリプトは、管理ユーティリティを実行するコンピュータに保存するか、管理ユーティリティを実行するコンピュータからアクセスが可能な場所に保存します。ファイル名と拡張子は任意に選択できます。ファイル拡張子 `.cli` の使用を推奨します。
3. スクリプトを実行するには、次のいずれかの操作を行います。

### オプション

### 説明

コマンドラインでスクリプトを実行するには

コマンドライン、またはバッチ スクリプトやシェル スクリプトの中で、以下のように指定します。

```
cli.cmd --cmdfile ScriptFile
```

管理ユーティリティでスクリプトを実行するには

管理ユーティリティを開き、**Spectrum™ Technology Platform** コマンドを使用して `connect` サーバーに接続します。続いて `script` コマンドを使用してスクリプトを実行します。このコマンドの詳細については、[system\\_script.dita](#)を参照してください。

**例: データフローをステージングから実稼働に移動する**

Deduplication、AddressValidation、および DrivingDirections の3つのデータフローがあります。これらのデータフローに修正を加え、テストするためのステージングサーバーと、データフローを実行可能にする実稼働環境があります。これらのデータフローをステージングサーバーから実稼働サーバーに移動する作業に一貫性を与え、自動化する必要があるため、管理ユーティリティ スクリプトを利用することにしました。使用するスクリプトは、次のような内容です。

```
// 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
```

## Spatial モジュール

### limrepo export

注：管理ユーティリティをインストールして実行する手順については、「[管理ユーティリティを使用する前に \(45ページ\)](#)」を参照してください。

limrepo export コマンドは、名前付きリソース (名前付きテーブルなど) を Spectrum Spatial リポジトリからローカル ファイル システムへエクスポートします。このコマンドを使うには、Spatial モジュールがインストールされている必要があります。

リソースはリポジトリのフルパスでターゲットのフォルダにエクスポートされます。例えば、limrepo export --s /Samples/NamedTables --o C:\export を実行すると、NamedTables フォルダ (ディレクトリ) の下の各名前付きテーブルに対して、C:\export\Samples\NamedTables\WorldTable などが作成されます。

注: limrepo export コマンドは必ず、空のフォルダを含むすべてのフォルダを再帰的にエクスポートします。

### 使用方法

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

注: パラメータのリストを表示するには、“help limrepo export” と入力します。

必須	引数	説明
はい	--s or source	エクスポートするリソースまたはフォルダへのパスを指定します。
はい	--o or output	エクスポート先のローカル ファイル システム上のフォルダへのパスを指定します。新規フォルダでも既存フォルダでもかまいませんが、既存フォルダの場合は中身を空にしておく必要があります。空でない場合、エクスポートは失敗します。
いいえ	--q or --quiet	エクスポート中にコピー済みリソースの表示を無効にします (すなわち、クワイエットモードで動作します)。  このフラグを指定した場合のデフォルト値は true です。このフラグを指定しない場合のデフォルト値は false です。
いいえ	--f or --fullpaths	エクスポート元と出力先のフルパスをプリントします。  このフラグを指定した場合のデフォルト値は true です。このフラグを指定しない場合のデフォルト値は false です。



必須	引数	説明
いいえ	<code>--r</code> OR <code>--recursive</code>	<p>サブフォルダ (指定されたエクスポート元フォルダの下にあるフォルダ) を再帰的にエクスポートします。</p> <p><b>true</b> 指定されたフォルダにあるすべてのファイルと、すべてのサブフォルダにあるファイルをエクスポートします。このフラグを指定しない場合、または値を設定しないでこのフラグを指定した場合、この動作がデフォルト設定です。</p> <p><b>false</b> 指定されたフォルダにあるファイルのみをエクスポートします (サブフォルダをエクスポートしません)。</p> <p><b>false</b> を指定すると、エクスポートされていないリソースを参照する名前付きリソースがエクスポート対象に含まれる可能性が高くなります。このフラグを使用するには、極めて注意する必要があります。リポジトリのすべての関連性を把握したユーザが使用してください。</p>
いいえ	<code>--c</code> OR <code>--continueonerror</code>	<p>エラーが発生した場合もエクスポートを続行します。</p> <p>このフラグを指定した場合のデフォルト値は <b>true</b> です。このフラグを指定しない場合のデフォルト値は <b>false</b> です。</p>
いいえ	<code>--a</code> OR <code>--acl</code>	<p>エクスポートするリソースの既存の権限をローカルファイルシステム上のエクスポート先ファイルでも維持します。アクセス制御リスト (ACL) は、ユーザーまたは役割が名前付きリソースに実行できる操作 (作成、表示、編集、削除など) を定義します。</p> <p>このフラグを指定した場合のデフォルト値は <b>true</b> です。このフラグを指定しない場合のデフォルト値は <b>false</b> です。</p>

**例**

この例は、リポジトリの \Samples フォルダにある名前付きリソースをローカルファイルシステムの C:\myrepository\samples にエクスポートします。

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

## limrepo import

**注：**管理ユーティリティをインストールして実行する手順については、「[管理ユーティリティを使用する前に \(45ページ\)](#)」を参照してください。

limrepo import コマンドは、名前付きリソース (名前付きテーブルなど) をローカルファイルシステムから Spectrum Spatial リポジトリへインポートします。このコマンドを使うには、Spatial モジュールがインストールされている必要があります。

リソースをインポートするには、以前にリソースをエクスポートしたフォルダまたはディレクトリを指定する必要があります。例えば、limrepo export --s /Samples/NamedTables --o C:\export を実行すると、C:\export\Samples\NamedTables\WorldTable が、NamedTables フォルダ (ディレクトリ) にある名前付きテーブルごとに作成されます。リソースはリポジトリのフルパスでターゲットのフォルダにエクスポートされます。limrepo import --s C:\export を実行すると、WorldTable が /Samples/NamedTables/WorldTable へインポートされます。

**注：**limrepo import コマンドは必ず、空のフォルダを含むすべてのフォルダを再帰的にインポートします。

インポートを行った後は多くの場合、Spectrum Spatial™ Manager を使用してその新しいパスを指すように名前付き接続を調整する必要があります。例えば、テストインスタンスの C:\myfiles にインストールされていたのと同じネイティブ TAB ファイルが

E:\ApplicationData\Spectrum\Spatial\Q3 にインストールされた場合、インポート後に Spectrum Spatial™ Manager でその接続を修正する必要があります。Spectrum Spatial™ Manager で名前付き接続を編集する手順については、『*Spectrum Spatial ガイド*』の「ユーティリティ」セクションを参照してください。

**注：**Spectrum™ Technology Platform の 12.0 以前のバージョンからエクスポートされたサービス設定ファイルを復元するために limrepo import を使用すると、そのファイルが自動的に修正され、バージョン 12.0 以降と互換性が成立します (例えばリポジトリ URL が削除されます)。

### 使用方法

```
limrepo import --s SourceFilePath
```

注：パラメータのリストを表示するには、“help limrepo import”と入力します。

必須	引数	説明
はい	<code>--s</code> or <code>source</code>	ローカル ファイル システムからインポートするリソースまたはフォルダへのパスを指定します。これは、ローカル ファイル システム上の以前のエクスポートに使用したルートフォルダである必要があります。
いいえ	<code>--q</code> or <code>--quiet</code>	インポート中にコピー済みリソースの表示を無効にします (すなわち、クワイエットモードで動作します)。  このフラグを指定した場合のデフォルト値は <b>true</b> です。このフラグを指定しない場合のデフォルト値は <b>false</b> です。
いいえ	<code>--u</code> or <code>--update</code>	同じ名前のリソースがサーバーに既に存在する場合、この既存のリソースを上書きするかどうかを指定します。  <b>true</b> インポートするリソースと同じ名前のリソースがサーバーに存在する場合、サーバーにあるリソースを上書きされます。このフラグを指定しない場合、または値を設定しないでこのフラグを指定した場合、この動作がデフォルト設定です。  <b>false</b> インポートするリソースと同じ名前のリソースがサーバーに存在する場合、リソースはインポートされません。
いいえ	<code>--f</code> or <code>--fullpaths</code>	エクスポート元と出力先のフルパスをプリントします。  このフラグを指定した場合のデフォルト値は <b>true</b> です。このフラグを指定しない場合のデフォルト値は <b>false</b> です。

必須	引数	説明
いいえ	<code>--c</code> or <code>--continueonerror</code>	<p>エラーが発生した場合もインポートを続行します。</p> <p>このフラグを指定した場合のデフォルト値は <b>true</b> です。このフラグを指定しない場合のデフォルト値は <b>false</b> です。</p>
いいえ	<code>--a</code> or <code>--acl</code>	<p>以前にエクスポートされた権限を維持し、リソースのインポート中にそれらを既存の権限と統合します。アクセス制御リスト (ACL) は、各ユーザーまたは役割が名前付きリソースに実行できる操作 (作成、表示、変更、削除など) を定義します。</p> <p>例えば、エクスポートするリソースに対して、あるユーザーが読み取りと書き込みの権限を持っています。このリソースのインポート時に、このユーザーがそのリソースの読み取り権限しか持っていない場合、インポートが正常に完了した後で書き込み権限が与えられます。</p> <p>競合する権限は統合できず、無視されます。ターゲットのリポジトリに存在しないユーザーや役割の ACL 項目は無視されます。</p> <p>このフラグを指定した場合のデフォルト値は <b>true</b> です。このフラグを指定しない場合のデフォルト値は <b>false</b> です。</p> <p><b>ヒント:</b> このフラグを使用する場合、エクスポート元のサーバー上のユーザが、インポート先のサーバー上にも存在する必要があります。例えば、アクセス制御が設定された "testuser" が存在し、ACL 付きのリソースをあるサーバーからエクスポートして、"testuser" が存在しない別のサーバーに、それらの名前付きリソースをインポートするとします。この場合、名前付きリソースは ACL なしでアップロードされます。</p>

**例**

この例は、名前付きリソースをローカル ファイル システムの `C:\myrepository\samples` からインポートします。

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

## limrepo mwsimport

注：管理ユーティリティをインストールして実行する手順については、「[管理ユーティリティを使用する前に（45ページ）](#)」を参照してください。

limrepo mwsimport 管理ユーティリティの Spectrum™ Technology Platform により、MapInfo Pro または MapXtreme Workspace Manager で作成された MapInfo Workspace (MWS) ファイル内のマップを Spectrum Spatial リポジトリにプロビジョニングすることができます。インポートにより、名前付きマップとそのすべての従属リソース (レイヤ、テーブル、接続) が作成されます。接続には、マップ名に 'Connection' を付加した名前が付けられます。名前付きテーブルと名前付きレイヤは、サブフォルダに作成されます (それぞれ、NamedTables と NamedLayers)。

このコマンドを使うには、Spatial モジュールがインストールされている必要があります。

### 使用方法

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

注：パラメータのリストを表示するには、“help limrepo mwsimport”と入力します。

必須	引数	説明
はい	--s <i>OR</i> source	ローカル ファイル システムからインポートする MWS ファイルへのパスを指定します。
はい	--o <i>OR</i> output	リポジトリ上の名前付きマップへのパスを指定します。すべてのリソースが、名前付きマップとして同じフォルダ内に作成されます。
はい	--p <i>OR</i> path	データが配置されているサーバー上の場所へのファイルパスを指定します。このパスを使用して名前付き接続が作成され、この名前付き接続は、作成されるすべての名前付きテーブルによって参照されます。これらのテーブルは、その名前付き接続への相対ファイルパスを使用します。
いいえ	--l <i>OR</i> local	MWS にサーバー ファイル システム上に存在しないファイルシステムが含まれる場合に、データが配置されているローカル ファイル システム上の場所へのファイルパスを指定します。指定した値が MWS ファイル内に出現すると、それが指定

必須	引数	説明
		したサーバーパスで置き換えられます。MWSファイルの中に部分パスがある場合は、これは不要です。MapXtremeで作成されたものならば通常、MWSファイルの中に部分パスがあります。

**例**

この例では、D: ドライブ上の MWS ファイルをインポートし (サーバー上のデータは C:\mydata に存在します)、名前付きリソースをリポジトリ内の /Europe/Countries に配置します。

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

**結果**

以下の名前付きリソースが作成されます。

```
/Europe/Countries/Europe (名前付きマップ)
/Europe/Countries/EuropeConnection (名前付き接続)
/Europe/Countries/NamedTables/austria (名前付きテーブル)
/Europe/Countries/NamedTables/belgium (名前付きテーブル)
.
/Europe/Countries/NamedLayers/austria (名前付きレイヤ)
/Europe/Countries/NamedLayers/belgium (名前付きレイヤ)
..
```

## ルーティング

### ermdb list

**注:** 管理ユーティリティをインストールして実行する手順については、「[管理ユーティリティを使用する前に \(45ページ\)](#)」を参照してください。

ermdb list コマンドは、サーバー上の既存のルーティング データベース リソースの全一覧を取得します。このコマンドを使うには、Spatial モジュールがインストールされている必要があります。

### 使用方法

```
ermdb list
```

#### 例

この例では、サーバー上のすべてのデータベースリソースの全一覧を取得します。

```
ermdb list
```

## ermdb get

**注：**管理ユーティリティをインストールして実行する手順については、「[管理ユーティリティを使用する前に \(45ページ\)](#)」を参照してください。

ermdb get コマンドでは、サーバー上に設定されているルーティング データベースに関する情報を取得することができます。返される情報は、データベースの名前、ファイル システム (パス) 上のデータベースの場所、データベース用に設定されているプール サイズです。このコマンドを使うには、Spatial モジュールがインストールされている必要があります。

### 使用方法

```
ermdb get --name database_name
```

**注：**パラメータのリストを表示するには、"help ermdb get" と入力します。

必須	引数	説明
はい	<code>--name</code> or <code>--n <i>database_name</i></code>	情報を返すデータベースリソースの名前を指定します。名前は、サーバー上で一意でなければなりません。既存のルーティングデータベースリソースの一覧を表示するには、 <code>ermdb list</code> コマンドを使用します。

#### 例

この例は、データベース リソース **US** の情報をサーバーから取得します。

```
ermdb get --name US
```

## ermdb add

注：管理ユーティリティをインストールして実行する手順については、「[管理ユーティリティを使用する前に（45ページ）](#)」を参照してください。

ermdb add コマンドは、新しいルーティング データベース リソースをサーバー上に作成します。このコマンドを使うには、Spatial モジュールがインストールされている必要があります。

注：ermdb add コマンドでは、追加する各データベースに対して一意の名前を使用する必要があります。

## 使用方法

```
ermdb add --name database_name --poolsize pool_size --path database_path --mn minimum_memory_size --mx maximum_memory_size
```

注：パラメータのリストを表示するには、“help ermdb add” と入力します。

必須	引数	説明
はい	--name or --n <i>database_name</i>	追加するデータベースリソースの名前を指定します。名前は、サーバー上で一意でなければなりません。既存のルーティング データベース リソースの一覧を表示するには、ermdb list コマンドを使用します。
いいえ	--poolsize or --s <i>pool_size</i>	データベースで処理する同時要求の最大数を指定します。指定しない場合のデフォルト値は 4 です。同時要求数としては、1～128 の任意の整数が指定可能です。
はい	--path <i>database_path</i>	ファイル サーバー上のルーティング データベースの場所を指定します。
いいえ	--mn or --minMem <i>minimum_memory_size</i>	このデータベースに割りてられるメモリの最小サイズを定義します。この値は、--mx で設定された値と同じか、それより小さい必要があります。
いいえ	--mx or --maxMem <i>maximum_memory_size</i>	このデータベースに割りてられるメモリの最大サイズを定義します。この値は 0 より大きく、なおかつ 65536 MB より小さい必要があります。



**例**

この例では、データベースリソース **US** を `E:/ERM-US/2019.09/driving/south` からサーバーに追加します。

```
ermdb add --name US --poolsize 10 --path
E:/ERM-US/2019.09/driving/south --mn 1200 --mx 65536
```

## ermdb delete

**注：**管理ユーティリティをインストールして実行する手順については、「[管理ユーティリティを使用する前に \(45ページ\)](#)」を参照してください。

`ermdb delete` コマンドは、サーバー上の既存のルーティング データベース リソースを削除します。このコマンドを使うには、**Spatial** モジュールがインストールされている必要があります。

### 使用方法

```
ermdb delete --name database_name
```

**注：**パラメータのリストを表示するには、“`help ermdb delete`” と入力します。

必須	引数	説明
はい	<code>--name or --n <i>database_name</i></code>	削除するデータベースリソースの名前を指定します。既存のルーティング データベース リソースの一覧を表示するには、 <code>ermdb list</code> コマンドを使用します。

**例**

この例では、データベース リソース **US** をサーバーから削除します。

```
ermdb delete --name US
```

## ermdb modify

**注：**管理ユーティリティをインストールして実行する手順については、「[管理ユーティリティを使用する前に \(45ページ\)](#)」を参照してください。

ermdb modify コマンドは、サーバー上の既存のルーティング データベース リソースを変更します。このコマンドを使うには、Spatial モジュールがインストールされている必要があります。

### 使用方法

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

注：パラメータのリストを表示するには、“help ermdb modify” と入力します。

必須	引数	説明
はい	--name or --n <i>database_name</i>	変更するデータベースリソースの名前を指定します。既存のルーティング データベース リソースの一覧を表示するには、ermdb list コマンドを使用します。
いいえ	--poolsize or --s <i>pool_size</i>	データベースで処理する同時要求の最大数を指定します。同時要求数としては、1～128の任意の整数が指定可能です。新しいプールサイズと新しいデータベースパスのいずれかを指定する必要があります。
いいえ	--path <i>database_path</i>	ファイル サーバー上のルーティング データベースの新しい場所を指定します。新しいプールサイズと新しいデータベースパスのいずれかを指定する必要があります。

#### 例

この例では、新しいバージョンのプール サイズとデータベース パスの両方を変更します。

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

## ermdb import

注：管理ユーティリティをインストールして実行する手順については、「[管理ユーティリティを使用する前に \(45ページ\)](#)」を参照してください。

ermdb import コマンドでは、ルーティング データベース設定を含むファイルをインポートして、データベースリソースをサーバー上に作成できます。インポート ファイルを作成するには、

ermdb template コマンド、または ermdb export コマンドで作成したファイルを使用します。このコマンドを使うには、Spatial モジュールがインストールされている必要があります。

インポート ファイルの形式は以下のとおりです。

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

ここで、product と module はそれぞれ Spatial と routing でなければなりません。name はデータベース名、maxActive はこのデータベースで処理する同時リクエストの最大数 (またはプールサイズ)、DatasetPaths はデータベース リソースのデータ セットへのパスです。

1つのインポート ファイルに複数のデータベースを追加できます (上の例をコピーします)。また、各データベース リソースに対して複数のデータセットをセミコロンで区切って追加できます。

**注:** インポート ファイルで UTF-8 文字を指定するには、CLI コマンド プロンプトの起動時に、値 UTF-8 を設定した JVM パラメータ file.encoding を追加する必要があります。例:  
-Dfile.encoding=UTF-8

## 使用方法

ermdb import --file *file\_name*

**注:** パラメータのリストを表示するには、“help ermdb import”と入力します。

必須	引数	説明
はい	--file or --f <i>file_name</i>	インポート ファイルのディレクトリと名前を指定します。

### 例

この例では、それぞれ複数のデータセットで構成される、US1 と US2 という 2つのデータベースをインポートします。

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

入力ファイルは次のように定義されています。

```
[
{
  "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

注：管理ユーティリティをインストールして実行する手順については、「[管理ユーティリティを使用する前に（45ページ）](#)」を参照してください。

ermdb export コマンドでは、サーバー上に設定されているルーティング データベースをファイルにエクスポートできます。このファイルはその後、バックアップや、あるインスタンスから別のインスタンスへの移行を目的とした、ermdb import コマンドによる別のインスタンスへのインポートに使用できます。このコマンドを使うには、**Spatial** モジュールがインストールされている必要があります。

注：ermdb export コマンドは必ず、ermDbResource.txt という名前のエクスポートファイルを作成します。

### 使用方法

```
ermdb export--directorydirectory_name
```

注：パラメータのリストを表示するには、“help ermdb export”と入力します。

必須	引数	説明
いいえ	<code>--directory</code> or <code>--o</code> <i>directory_name</i>	データベース ファイルのエクスポート先となるファイルシステム上のディレクトリ名を指定します。このエクスポート コマンドは必ず、ermDbResource.txt という名前のエクスポートファイルを作成します。このパラメータを指定しない場合は、このエクスポート コマンドを実行したディレクトリにエクスポート ファイルが作成されます。

#### 例

この例では、E:/ERM-US/export ディレクトリにエクスポート データベースファイルを作成します。

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

## erm getpointdata

注：管理ユーティリティをインストールして実行する手順については、「[管理ユーティリティを使用する前に \(45ページ\)](#)」を参照してください。

erm getpointdata コマンドは、ポイントに対するセグメント情報を返します。指定されたポイントに最も近い 1 つ以上のセグメントが返されます。返される情報のタイプとしては、セグメント ID、道路タイプ、長さ、速度、方向、時間、道路名があります。このコマンドを使うには、Spatial モジュールがインストールされている必要があります。

### 使用方法

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

注：パラメータのリストを表示するには、“help erm getpointdata”と入力します。

必須	引数	説明
はい	<code>--datasource <i>db_resource</i></code>	データを返すデータベースリソースの名前を指定します。既存のルーティングデータベースリソースの一覧を表示するには、 <code>ermdb list</code> コマンドを使用します。
はい	<code>--point "<i>x,y,coordsys</i>"</code>	最も近いセグメントの情報を返すポイントを指定します。ポイントは、" <i>x,y,coordsys</i> " という形式で指定します。ここで、 <i>coordsys</i> はポイントの座標系です。

**例**

この例では、サーバー上に設定されている `US_NE` データベースリソースから、指定されたポイントに最も近いセグメントのデータを返します。

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

## erm getsegmentdata

**注：**管理ユーティリティをインストールして実行する手順については、「[管理ユーティリティを使用する前に \(45ページ\)](#)」を参照してください。

`erm getsegmentdata` コマンドは、指定されたセグメント ID に対するセグメント情報を返します。返される情報のタイプとしては、セグメント ID、道路タイプ、長さ、速度、方向、時間、道路名があります。このコマンドを使うには、`Spatial` モジュールがインストールされている必要があります。

### 使用方法

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

**注：**パラメータのリストを表示するには、“`help erm getsegmentdata`”と入力します。

必須	引数	説明
はい	<code>--datasource <i>db_resource</i></code>	データを返すデータベースリソースの名前を指定します。既存のルーティングデータベースリソースの一覧を表示するには、 <code>ermdb list</code> コマンドを使用します。

必須	引数	説明
はい	<code>--segmentid "segment_id"</code>	情報を返すセグメントを指定します。セグメントは、データ内で指定されている形式で指定します。例: <code>"7e3396fc:6e5251"</code> 。

**例**

この例では、サーバー上に設定されている `US_NE` データベースリソースから、指定されたセグメントのデータを返します。

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

## erm createpointupdate

**注：**管理ユーティリティをインストールして実行する手順については、「[管理ユーティリティを使用する前に \(45ページ\)](#)」を参照してください。

`erm createpointupdate` コマンドは、指定されたポイントに最も近いセグメントのルーティングデータをオーバーライドします。このコマンドにより、速度を設定または変更したり、ルートの子セクションを除外したりできます。このコマンドを使うには、`Spatial` モジュールがインストールされている必要があります。

**注：**永続更新タイプは特定のデータリソースに対してのみ有効で、データ更新後には有効でなくなる場合があります。

### 使用方法

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

**注：**パラメータのリストを表示するには、“`help erm createpointupdate`”と入力します。

必須	引数	説明
はい	<code>--datasource <i>db_resource</i></code>	データをオーバーライドするデータベースリソースの名前を指定します。既存のルーティングデータベースリソースの一覧を表示するには、 <code>ermdb list</code> コマンドを使用します。

必須	引数	説明
はい	<code>--point "x,y,coordsys"</code>	最も近いセグメントの情報をオーバーライドするポイントを指定します。ポイントは、 <code>"x,y,coordsys"</code> という形式で指定します。ここで、 <code>coordsys</code> はポイントの座標系です。
いいえ	<code>--exclude</code>	<code>true</code> に設定された場合、指定されたポイントをすべてのルート計算から除外します。コマンドにこのパラメータを含めることにより、そのポイントを除外するかどうかを指定します。除外しない場合は、 <code>--exclude</code> の後に <code>false</code> を追加します。
いいえ	<code>--velocity velocity_value</code>	速度更新を定義します。新しい速度を指定することにより、ポイントの新しい速度を指定します。 <code>velocityunit</code> パラメータを指定しない限り、デフォルトの単位は <code>mph</code> (マイル/時) です。
いいえ	<code>--velocityunit velocity_unit</code>	<code>velocity</code> または <code>velocityadjustment</code> オーバーライドの速度単位を定義します。デフォルトは <code>mph</code> (マイル/時) です。速度更新の場合、速度単位には <code>kph</code> (キロメートル/時)、 <code>mps</code> (メートル/秒)、 <code>mph</code> (マイル/時) のいずれかの値を指定できます。
いいえ	<code>--velocityadjustment velocity_adjustment_value</code>	速度更新を定義します。速度 (単位と値) の変更を指定することにより、ポイントの速度の変化を定義します。速度値は増加 (正の値) または減少 (負の値) させることができます。 <code>velocityunit</code> パラメータを指定しない限り、デフォルトの単位は <code>mph</code> (マイル/時) です。
いいえ	<code>--velocitypercentage velocity_percentage_value</code>	速度更新を定義します。速度を増加 (正の値) または減少 (負の値) させる割合を指定することにより、ポイントの速度の増加を定義します。

**例**

この例では、サーバー上に設定されている `US_NE` データベースリソースから、ポイントの速度を `15 mph` にオーバーライドします。

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



この例では、サーバー上に設定されている **US\_NE** データベースリソースから、指定されたポイントを除外します。

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

この例では、サーバー上に設定されている **US\_NE** データベースリソースから、ポイントの速度を **45 kph** だけ増加させてオーバーライドします。

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

この例では、サーバー上に設定されている **US\_NE** データベースリソースから、ポイントの速度を **60%** だけ減少させてオーバーライドします。

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

## erm resetpointupdate

注：管理ユーティリティをインストールして実行する手順については、「[管理ユーティリティを使用する前に](#)（45ページ）」を参照してください。

erm resetpointupdate コマンドは、データの元の状態に対するオーバーライドがあれば、それを返します。このコマンドを使うには、**Spatial** モジュールがインストールされている必要があります。

### 使用方法

```
erm
resetpointupdate--datasourcedb_resource--point"x,y,coordsys"--resettyperereset_type
```

注：パラメータのリストを表示するには、“help erm resetpointupdate”と入力します。

必須	引数	説明
はい	--datasource <b>db_resource</b>	オーバーライドを持つデータベースリソースの名前を指定します。既存のルーティングデータベースリソースの一覧を表示するには、ermdb list コマンドを使用します。
はい	--point " <i>x,y,coordsys</i> "	既存のオーバーライドが位置するポイントを指定します。ポイントは、“ <i>x,y,coordsys</i> ”という形式

必須	引数	説明
		で指定します。ここで、 <i>coordsys</i> はポイントの座標系です。
はい	<code>--resettype <i>reset_type</i></code>	削除 (取り消し) するオーバーライドのタイプです。  <b>速度</b> 速度更新を削除します。 <b>除外</b> 除外更新を削除します。

**例**

この例では、指定されたポイントの既存の除外オーバーライドを、サーバー上に設定されている **US\_NE** データベース リソースからリセットします。

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

## erm createsegmentupdate

**注：**管理ユーティリティをインストールして実行する手順については、「[管理ユーティリティを使用する前に \(45ページ\)](#)」を参照してください。

`erm createsegmentupdate` コマンドは、指定されたセグメントのルーティング データをオーバーライドします。このコマンドにより、速度を設定または変更したり、ルートの子セクションを除外したり、道路タイプを変更したりできます。このコマンドを使うには、**Spatial** モジュールがインストールされている必要があります。

**注：**永続更新タイプは特定のデータ リソースに対してのみ有効で、データ更新後には有効でなくなる場合があります。

### 使用方法

```
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
```

**注：**パラメータのリストを表示するには、“`help erm createsegmentupdate`” と入力します。

必須	引数	説明
はい	--datasource <i>db_resource</i>	データをオーバーライドするデータベースリソースの名前を指定します。既存のルーティングデータベースリソースの一覧を表示するには、 <code>ermdb list</code> コマンドを使用します。
はい	--segmentid " <i>segment_id</i> "	オーバーライドするセグメントを指定します。セグメントは、データ内で指定されている形式で指定します。例: " <code>7e3396fc:6e5251</code> "。
いいえ	--exclude	<code>true</code> に設定されている場合、指定されたセグメントをすべてのルート計算から除外します。コマンドにこのパラメータを含めることにより、そのセグメントを除外するかどうかを指定します。除外しない場合は、 <code>--exclude</code> の後に <code>false</code> を追加します。
いいえ	--velocity <i>velocity_value</i>	速度更新を定義します。新しい速度を指定することにより、セグメントの新しい速度を指定します。velocityunit パラメータを指定しない限り、デフォルトの単位は mph (マイル/時) です。
いいえ	--velocityunit <i>velocity_unit</i>	velocity または velocityadjustment オーバーライドの速度単位を定義します。値は mph (マイル/時) です。速度更新の場合、速度単位には kph (キロメートル/時)、mps (メートル/秒)、mph (マイル/時) のいずれかの値を指定できます。
いいえ	--velocityadjustment <i>velocity_adjustment_value</i>	速度更新を定義します。速度(単位と値)の変更を指定することにより、セグメントの速度の変化を定義します。速度値は増加(正の値)または減少(負の値)させることができます。velocityunit パラメータを指定しない限り、デフォルトの単位は mph (マイル/時) です。
いいえ	--velocitypercentage <i>velocity_percentage_value</i>	速度更新を定義します。速度を増加(正の値)または減少(負の値)させる割合を指定することにより、セグメントの速度の増加を定義します。
いいえ	--roadtype <i>road_type</i>	セグメントの新しい道路タイプを定義します。

**例**

この例では、サーバー上に設定されている **US\_NE** データベースリソースから、セグメントの速度を **15 mph** にオーバーライドします。

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

この例では、サーバー上に設定されている **US\_NE** データベースリソースから、指定されたセグメントを除外します。

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

この例では、サーバー上に設定されている **US\_NE** データベースリソースから、セグメントの速度を **45 kph** だけ増加させてオーバーライドします。

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

この例では、サーバー上に設定されている **US\_NE** データベースリソースから、セグメントの速度を **60%** だけ減少させてオーバーライドします。

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

この例では、サーバー上に設定されている **US\_NE** データベースリソースから、セグメントの道路タイプをフェリーにオーバーライドします。

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

## erm resetsegmentupdate

**注：**管理ユーティリティをインストールして実行する手順については、「[管理ユーティリティを使用する前に \(45ページ\)](#)」を参照してください。

erm resetsegmentupdate コマンドは、データの元の状態に対するオーバーライドがあれば、それを返します。このコマンドを使うには、**Spatial** モジュールがインストールされている必要があります。

### 使用方法

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

**注：**パラメータのリストを表示するには、“help erm resetsegmentupdate” と入力します。

必須	引数	説明						
はい	<code>--datasource <i>db_resource</i></code>	オーバーライドを持つデータベースリソースの名前を指定します。既存のルーティングデータベースリソースの一覧を表示するには、 <code>ermdb list</code> コマンドを使用します。						
はい	<code>--segment "<i>segment_id</i>"</code>	既存のオーバーライドが位置するセグメントを指定します。セグメントは、データ内で指定されている形式で指定します。例: <code>"7e3396fc:6e5251"</code> 。						
はい	<code>--resettype <i>reset_type</i></code>	削除 (取り消し) するオーバーライドのタイプです。 <table border="0" style="margin-left: 20px;"> <tr> <td><b>速度</b></td> <td>速度更新を削除します。</td> </tr> <tr> <td><b>除外</b></td> <td>除外更新を削除します。</td> </tr> <tr> <td><b>roadType</b></td> <td>道路タイプ更新を削除します。</td> </tr> </table>	<b>速度</b>	速度更新を削除します。	<b>除外</b>	除外更新を削除します。	<b>roadType</b>	道路タイプ更新を削除します。
<b>速度</b>	速度更新を削除します。							
<b>除外</b>	除外更新を削除します。							
<b>roadType</b>	道路タイプ更新を削除します。							

**例**

この例では、指定されたセグメントの既存の道路タイプ オーバーライドを、サーバー上に設定されている `US_NE` データベース リソースからリセットします。

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

## erm getsegmentupdates

**注：**管理ユーティリティをインストールして実行する手順については、「[管理ユーティリティを使用する前に \(45ページ\)](#)」を参照してください。

`erm getsegmentupdates` コマンドは、指定されたセグメントのルーティング データにおけるオーバーライドのリストを返します。このコマンドを使うには、**Spatial** モジュールがインストールされている必要があります。

**注：**`segmentids` はオプションのパラメータです。セグメント ID を指定しない場合は、使用可能なすべてのセグメントに対するオーバーライドが返されます。

### 使用方法

erm

```
getsegmentupdates--datasourceddb_resource--segmentids"segment_ids"--velocityunitvelocityunit
```

注：パラメータのリストを表示するには、“help erm getsegmentupdates”と入力します。

必須	引数	説明
はい	--datasource <i>db_resource</i>	オーバーライドを持つデータベースリソースの名前を指定します。既存のルーティングデータベースリソースの一覧を表示するには、 <code>ermdb list</code> コマンドを使用します。
いいえ	--segmentids " <i>segment_ids</i> "	オーバーライド情報を返すセグメント ID のカンマ区切りリスト。セグメントは、データ内で指定されている形式で指定します。例: " <b>7e3396fc:6e5251</b> "。
いいえ	--velocityunit <i>velocityunit</i>	レスポンスに表示される速度単位を指定します (mph - マイル/時、kph - キロメートル/時、mtps - メートル/秒、mtpm - メートル/分)。デフォルトは mph です。

#### 例

この例では、サーバー上に設定されている `US_NE` データベースリソースから、セグメントのオーバーライドを返します。

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

## erm createroadtypeupdate

注：管理ユーティリティをインストールして実行する手順については、「[管理ユーティリティを使用する前に \(45ページ\)](#)」を参照してください。

`erm createroadtypeupdate` コマンドは、指定された道路タイプのルーティングデータをオーバーライドします。このコマンドにより、特定の道路タイプのルートの設定または変更できます。このコマンドを使うには、**Spatial** モジュールがインストールされている必要があります。

注：永続更新タイプは特定のデータリソースに対してのみ有効で、データ更新後には有効でなくなる場合があります。

## 使用方法

erm

```

createtype -datasource db_resource -roadtype "road_type" -velocity db_velocity -velocityunit db_velocity_unit -width db_width -velocityacttag db_velocity_act_tag -roadtype db_roadtype

```

注：パラメータのリストを表示するには、“help erm createroadtypeupdate”と入力します。

必須	引数	説明
はい	--datasource <b>db_resource</b>	データをオーバーライドするデータベースリソースの名前を指定します。既存のルーティングデータベースリソースの一覧を表示するには、ermdb list コマンドを使用します。
はい	--roadtype <b>"road_type"</b>	オーバーロードする道路タイプ <ul style="list-style-type: none"> <li>• 進入路</li> <li>• 裏道路</li> <li>• 接続道路</li> <li>• フェリー</li> <li>• 小道</li> <li>• 侵入制限道路 (密集都市部)</li> <li>• 侵入制限道路 (農村部)</li> <li>• 侵入制限道路 (郊外)</li> <li>• 侵入制限道路 (都市部)</li> <li>• 地方道路 (密集都市部)</li> <li>• 地方道路 (農村部)</li> <li>• 地方道路 (郊外)</li> <li>• 地方道路 (都市部)</li> <li>• 主要地方道路 (密集都市部)</li> <li>• 主要地方道路 (農村部)</li> <li>• 主要地方道路 (郊外)</li> <li>• 主要地方道路 (都市部)</li> <li>• 幹線道路 (密集都市部)</li> <li>• 幹線道路 (農村部)</li> <li>• 幹線道路 (郊外)</li> <li>• 幹線道路 (都市部)</li> <li>• 補助地方道路 (密集都市部)</li> <li>• 補助地方道路 (農村部)</li> <li>• 補助地方道路 (郊外)</li> </ul>

必須	引数	説明
		<ul style="list-style-type: none"> <li>• 補助地方道路 (都市部)</li> <li>• 一般道路 (密集都市部)</li> <li>• 一般道路 (農村部)</li> <li>• 一般道路 (農村部)</li> <li>• 一般道路 (都市部)</li> <li>• 主要高速道路 (密集都市部)</li> <li>• 主要高速道路 (農村部)</li> <li>• 主要高速道路 (郊外)</li> <li>• 主要高速道路 (都市部)</li> <li>• 出入路 (密集都市部)</li> <li>• 出入路 (侵入制限道路)</li> <li>• 出入路 (幹線道路)</li> <li>• 出入路 (主要高速道路)</li> <li>• 出入路 (農村部)</li> <li>• 出入路 (一般高速道路)</li> <li>• 出入路 (都市部)</li> <li>• 出入路 (郊外)</li> <li>• 一般高速道路 (密集都市部)</li> <li>• 一般高速道路 (農村部)</li> <li>• 一般高速道路 (郊外)</li> <li>• 一般高速道路 (都市部)</li> </ul>
いいえ	<code>--velocity <i>velocity_value</i></code>	速度更新を定義します。新しい速度を指定することにより、道路タイプの新しい速度を指定します。 <code>velocityunit</code> パラメータを指定しない限り、デフォルトの単位は <b>mph</b> (マイル/時) です。
いいえ	<code>--velocityunit <i>velocity_unit</i></code>	<code>velocity</code> または <code>velocityadjustment</code> オーバーライドの速度単位を定義します。デフォルトは <b>mph</b> (マイル/時) です。速度更新の場合、速度単位には <b>kph</b> (キロメートル/時)、 <b>mps</b> (メートル/秒)、 <b>mph</b> (マイル/時) のいずれかの値を指定できます。
いいえ	<code>--velocityadjustment <i>velocity_adjustment_value</i></code>	速度更新を定義します。速度 (単位と値) の変更を指定することにより、道路タイプの速度の変化を定義します。速度値は増加 (正の値) または減少 (負の値) させることができます。 <code>velocityunit</code>



必須	引数	説明
		パラメータを指定しない限り、デフォルトの単位は mph (マイル/時) です。
いいえ	<code>--velocitypercentage</code> <code>velocity_percentage_value</code>	速度更新を定義します。速度を増加 (正の値) または減少 (負の値) させる割合を指定することにより、道路タイプの速度の増加を定義します。

**例**

この例では、サーバー上に設定されている **US\_NE** データベースリソースから、指定された道路タイプの速度を **25 mph** にオーバーライドします。

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

この例では、サーバー上に設定されている **US\_NE** データベースリソースから、指定された道路タイプの速度を **50 kph** だけ増加します。

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

この例では、サーバー上に設定されている **US\_NE** データベースリソースから、指定された道路タイプの速度を **65%** だけ減少させることによってオーバーライドします。

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

## erm resetroadtypeupdate

注：管理ユーティリティをインストールして実行する手順については、「[管理ユーティリティを使用する前に \(45ページ\)](#)」を参照してください。

erm resetroadtypeupdate コマンドは、データの元の状態に対するオーバーライドがあれば、それを返します。このコマンドを使うには、**Spatial** モジュールがインストールされている必要があります。

### 使用方法

```
erm resetroadtypeupdate--datasourcedb_resource--roadtype"road_type"
```

注：パラメータのリストを表示するには、“help erm resetroadtypeupdate” と入力します。

必須	引数	説明
はい	<code>--datasource <i>db_resource</i></code>	オーバーライドを持つデータベースリソースの名前を指定します。既存のルーティングデータベースリソースの一覧を表示するには、 <code>ermdb list</code> コマンドを使用します。
はい	<code>--roadtype "<i>road_type</i>"</code>	既存のオーバーライドを持つ道路タイプを指定します。道路タイプの一覧については、「 <a href="#">erm createroadtypeupdate (70ページ)</a> 」を参照してください。

**例**

この例では、"normal road suburban" (一般道路 (郊外)) の道路タイプオーバーライドを、サーバー上に設定されている US\_NE データベースリソースからリセットします。

```
erm resetroadtypeupdate --datasource US_NE --roadtype "normal road suburban"
```

## erm getroadtypeupdates

**注：**管理ユーティリティをインストールして実行する手順については、「[管理ユーティリティを使用する前に \(45ページ\)](#)」を参照してください。

`erm getroadtypeupdates` コマンドは、指定された道路タイプのルーティング データにおけるオーバーライドのリストを返します。このコマンドを使うには、**Spatial** モジュールがインストールされている必要があります。

**注：**`roadtypes` はオプションのパラメータです。道路タイプを指定しない場合は、使用可能なすべての道路タイプに対するオーバーライドが返されます。

### 使用方法

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

**注：**パラメータのリストを表示するには、「`help erm getroadtypeupdates`」と入力します。

必須	引数	説明
はい	<code>--datasource <i>db_resource</i></code>	オーバーライドを持つデータベースリソースの名前を指定します。既存のルーティングデータベースリソースの一覧を表示するには、 <code>ermdb list</code> コマンドを使用します。
いいえ	<code>--roadtypes "<i>road_types</i>"</code>	オーバーライド情報を返す道路タイプのカンマ区切りリスト。道路タイプの一覧については、「 <a href="#">erm createroadtypeupdate (70ページ)</a> 」を参照してください。
いいえ	<code>--velocityunit <i>velocityunit</i></code>	レスポンスに表示される速度単位を指定します (mph - マイル/時、kph - キロメートル/時、mtps - メートル/秒、mtpm - メートル/分)。デフォルトは mph です。

**例**

この例では、"normal road urban" (一般道路 (都市部)) の道路タイプのオーバーライドを、サーバー上に設定されている US\_NE データベースリソースから返します。

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

## erm getallupdates

**注：**管理ユーティリティをインストールして実行する手順については、「[管理ユーティリティを使用する前に \(45ページ\)](#)」を参照してください。

`erm getallupdates` コマンドは、指定されたルーティング データベース リソースに対するすべてのオーバーライドのリストを返します。このコマンドを使うには、**Spatial** モジュールがインストールされている必要があります。

### 使用方法

```
erm getallupdates--datasourcedb_resource"segment_ids"--velocityunitvelocityunit
```

**注：**パラメータのリストを表示するには、「`help erm getallupdates`」と入力します。

必須	引数	説明
はい	<code>--datasource <i>db_resource</i></code>	オーバーライドを持つデータベースリソースの名前を指定します。既存のルーティングデータベースリソースの一覧を表示するには、 <code>ermdb list</code> コマンドを使用します。
いいえ	<code>--velocityunit <i>velocityunit</i></code>	レスポンスに表示される速度単位を指定します (mph - マイル/時、kph - キロメートル/時、mtps - メートル/秒、mtpm - メートル/分)。デフォルトは mph です。

**例**

この例では、サーバー上に設定されている `US_NE` データベースリソースからのすべてのオーバーライドを返します。

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

## erm resetallupdates

**注：**管理ユーティリティをインストールして実行する手順については、「[管理ユーティリティを使用する前に \(45ページ\)](#)」を参照してください。

`erm resetallupdates` コマンドは、データの元の状態に対するすべてのオーバーライドを返します。このコマンドを使うには、**Spatial** モジュールがインストールされている必要があります。

### 使用方法

```
erm resetallupdates--datasourcedb_resource
```

**注：**パラメータのリストを表示するには、“`help erm resetallupdates`” と入力します。

必須	引数	説明
はい	<code>--datasource <i>db_resource</i></code>	オーバーライドを持つデータベースリソースの名前を指定します。既存のルーティングデータベースリソースの一覧を表示するには、 <code>ermdb list</code> コマンドを使用します。

**例**

この例では、サーバー上に設定されている US\_NE データベースリソースからのすべてのオーバーライドをリセットします。

```
erm resetallupdates --datasource US_NE
```

# 7 - ルーティング

## このセクションの構成

---

デフォルト サービス/ステージ オプションの指定	79
サービス/ステージのプレビュー	79
Management Console を用いたルート データの取得	82

## デフォルト サービス/ステージ オプションの指定

デフォルト オプションは、システム上の各サービスまたはステージのデフォルトの動作を制御します。各オプションに対して、デフォルト値を指定できます。デフォルト オプションは、要求においてオプションの値が明示的に定義されていない場合に適用されます。また、デフォルト オプションは、**Enterprise Designer** でこのサービスを使用してデータフローを作成する場合に、デフォルトとして使用される設定でもあります。

オプションの詳細については、『*Spectrum Spatial* ガイド』の **Spatial** モジュールに該当する「ステージ」セクションと「リソースおよびデータ」セクションを参照してください。

注：永続更新は、**Management Console** では管理されません。永続更新を行うには、管理ユーティリティの **Spectrum** コマンドライン機能を使用します。

注：**Management Console** の **Get Route Data** サービスでは、デフォルト オプションは設定されません。このサービスはインタラクティブに動作して、セグメントに対するルーティング データを返します。**Get Route Data** の詳細については、「**Management Console** を用いたルート データの取得 (82ページ)」を参照してください。

1. **Management Console** を開きます。
2. **[サービス]** メニューに移動し、**[ルーティング]** を選択します。
3. 左側のリストで設定するサービスをクリックします。
4. サービスのオプションを設定します。ほとんどのサービスにさまざまな種類のオプションがあり、それらが複数のタブに表示されます。
5. **[保存]** をクリックします。

## サービス/ステージのプレビュー

**Management Console** でサービスの **[プレビュー]** タブを使うと、サービスの結果をプレビューできます。プレビューは、どのオプションを指定するかを決める際に便利です。異なるオプションが、サービスまたはステージから返されるデータに与える影響を直ちに確認できるためです。

1. **Management Console** を開きます。
2. **[サービス]** メニューを開き、プレビューするサービスを選択します。
3. **[プレビュー]** タブをクリックします。

#### 4. テスト用のデータを各フィールドに入力します。

プレビューを使用する際には、次の点に注意してください。



- すべてのフィールドにデータを入力する必要はありません。フィールドを空のままにすると、空の文字列がプレビューに使用されます。
- フィールドに null 値を引き渡した場合の影響をプレビューするには、そのフィールドの横の無効アイコンをクリックします。

##### 入力レコード



##### ▼ 入力レコード 1



- 複数のレコードを同時にプレビューできます。レコードを追加するには、追加ボタン  をクリックします。
- テスト用のデータはファイルからインポートできます。データをインポートするには、インポート ボタン  をクリックします。[ファイル名] と [フィールド区切り文字] を選択します。次のことに注意してください。
  - ファイルの最初の行は、ヘッダーレコードでなければなりません。ヘッダー内のフィールド名は、サービスで必要とされるフィールド名と一致する必要があります。
  - インポートできるレコードは、最大 5 個です。
  - フィールドの区切り文字がスペースの場合、フィールド値を引用符で囲む必要があります。スペースをフィールドの区切り文字に使うファイルの例を以下に示します。

```
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" ""
```

- すべてのレコードを削除するには、プレビュー エリアの上部にある [削除] ボタン





をクリックします。

- 特定のレコードを削除するには、入力レコード名 ("Input Record 1" など) の上にカーソルを合わせてレコード名の横の [削除] ボタン



をクリックします。

- サービスが階層化されたデータを入力として受け取る場合
  - 子レコードを追加するには、親レコードの上にカーソルを合わせて追加ボタンをクリックします。
  - 親からすべての子レコードを削除するには、親レコードの上にカーソルを合わせて削除ボタンをクリックします。
  - 特定の子レコードを削除するには、そのレコードの上にカーソルを合わせて削除ボタンをクリックします。

##### 5. [プレビューを実行] をクリックします。

サービスは入力レコードを処理し、結果を表示します。

- 出力データをプレビューして、サービスやステージから適切な結果が得られることを確認します。必要に応じて、オプションに変更を加えて、**[プレビューを実行]** を再度クリックすることができます(データを再度入力する必要はありません)。

## Management Console を用いたルート データの取得

Management Console を使用して、最も近いポイントまたはセグメント ID からのセグメント情報をプレビューおよび保存することができます。GetRouteData サービスは、ポイントまたはセグメント ID に対するセグメント情報を返します。ポイントを指定すると、最も近いルート セグメントが返されます。セグメント ID を指定すると、指定されたルート セグメントのルート データが返されます。

ルート データをプレビューまたは保存するには

- Management Console を開きます。
- [サービス]** メニューに移動し、**[ルーティング]** を選択します。
- サービス リストから **[Get Route Data]** を選択します。
- [入力タイプ]** フィールドで、**[ポイント データ]** または **[セグメント データ]** を選択します。
- [データベース]** フィールドから、ルーティング データベース リソースを選択します。

新しいルーティング データベース リソースを追加する必要がある場合は、「[ルーティング データベース リソースの作成](#)」を参照してください。

- 選択した入力タイプに必要な情報を入力します。

ポイント データを選択した場合は、ポイント座標と座標系を入力します。セグメント データを選択した場合は、セグメント ID を入力します。

- [プレビュー]** をクリックします。

**[出力データ]** セクションに、ルート セグメント データが返されます。入力に関連するセグメントが複数存在する場合は、複数のセグメントが **[セグメント詳細 1]**、**[セグメント詳細 2]** などのように表示されます。

- [保存]** ボタンをクリックしてルーティング データ結果をテキスト ファイルとして保存するか、**[クリア]** ボタンをクリックして **[出力データ]** セクションから結果を削除します。

# 8 - システムのトラブルシューティング

## このセクションの構成

---

破損したリポジトリ インデックスの再構築	84
反応がなくなったサーバーのメモリ使用量の監視	84
空間リモート コンポーネントのヒープ メモリの増加	86

## 破損したりポジトリ インデックスの再構築

手動操作または停電によって、サーバーが突然シャットダウンした場合や Java プロセスが停止した場合には、リポジトリが破損することがあります。この場合、以前は検索可能だったリソースを取得できなくなることがありますが、ログにエラーや警告は記録されません。権限の変更が原因でないことを確認したら、次の手順でインデックスを再構築してこの問題を修正します。

1. サーバーをシャットダウンします。
2. 次の場所のインデックス ディレクトリを削除します。
  - <Spectrum>\server\modules\spatial\jackrabbit\workspaces\default
  - <Spectrum>\server\modules\spatial\jackrabbit\workspaces\security
  - <Spectrum>\server\modules\spatial\jackrabbit\repository
3. サーバーを再起動します。  
起動時に Jackrabbit が上記の場所にインデックスを再作成します。

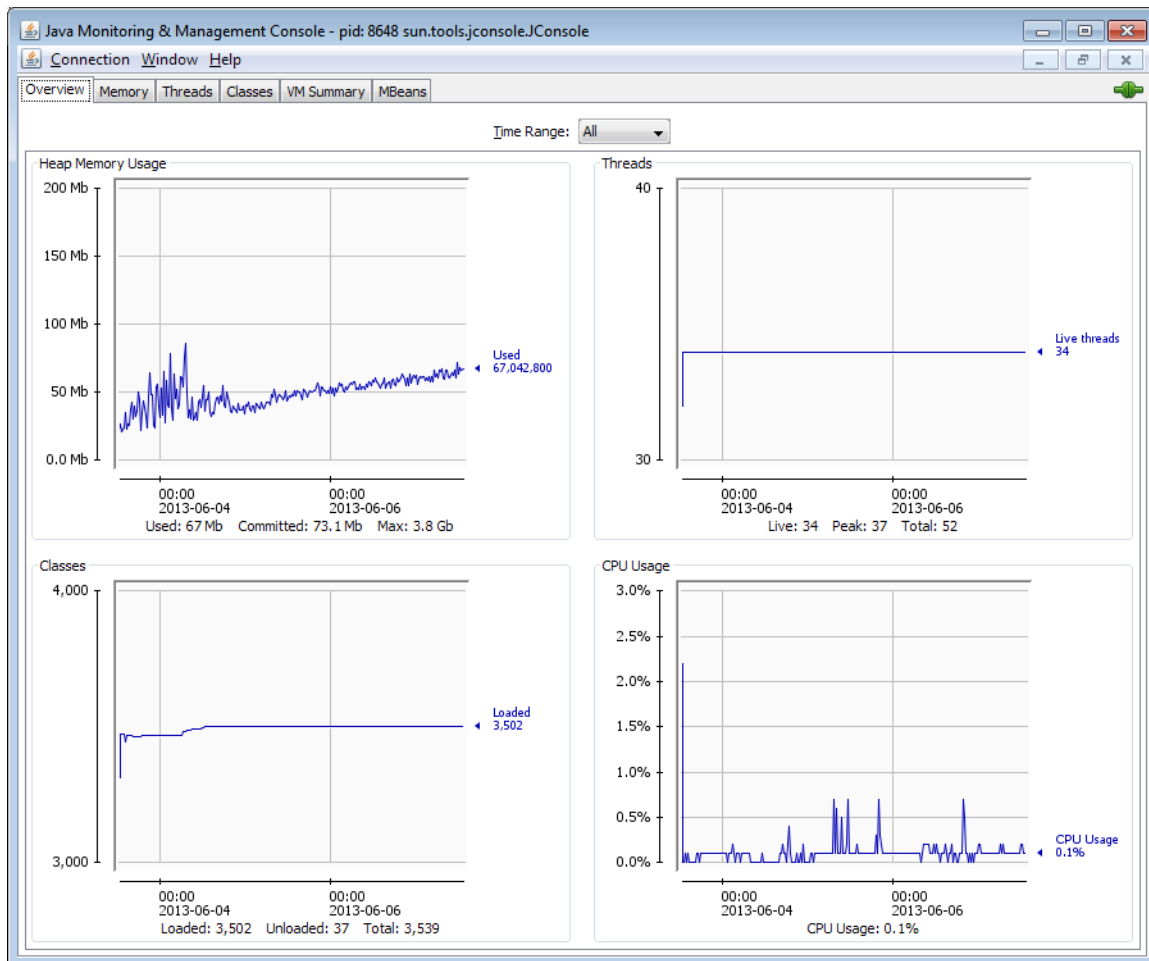
インデックスを再構築した後は、検索は正常な動作に戻ります。

## 反応がなくなったサーバーのメモリ使用量の監視

Spectrum サーバーの反応が停止した場合は、次の手順でパフォーマンスとリソース消費を監視できます。この監視で得られる情報を使用して、メモリとスレッドの使用を調整できます。

1. Mapping Service 以外のサービスが動作しているかどうか確認します。例えば、デモ ページ `http://<サーバー名>:<ポート>/Spatial/FeatureService//DemoPage.html` で Feature Service を開始します。これにより、サーバー全体が停止しているのか、Mapping Service だけが停止しているのかを判断できます。
2. 次の構成ファイルに基づいて、Mapping と Map Tiling の両方のイメージを格納する十分なディスク領域があることを確認します。
  - **Mapping:** `http://<サーバー>:<ポート>/RepositoryService/repository/default/Configuration/MappingConfiguration` の "`<Directory> C:\Program Files\Pitney Bowes\Spectrum/server/modules/spatial/images </Directory>`"
  - **Map Tiling:** "`<Property name="diskPath" value="C:/Program Files/Pitney Bowes/Spectrum/server/modules/spatial/TileCache"/>`" の "`http://<server>:<port>/RepositoryService/repository/default/Configuration/MapTilingConfiguration`"

3. Spectrum サーバーを停止します。
4. テキスト エディタで、`java.vargs` ファイルを `<Installed>\Pitney Bowes\Spectrum\server\modules\spatial\java.vargs` から開きます。
5. `vargs` の値を変更します。デフォルトは **2GB (2048MB)** です。例えば、リモート コンポーネントのメモリを **4GB** に増やすには、`vargs` をデフォルトの `-Xmx2048m` から `-Xmx4096m` に変更します。お使いのオペレーティング システムで使用可能な最大メモリ容量を超えないようにし、またオペレーティング システム自体の処理のための適切な容量を残すようにしてください。
6. `java.vargs` ファイルを保存します。
7. 次の手順でサーバー ラッパーを開始します。
  - a) 管理者としてコマンド プロンプトを開きます。
  - b) `<Installed>\Pitney Bowes\Spectrum\server\bin\wrapper` ディレクトリに移動し、「**wrapper.exe -c**」と入力します。Spectrum サーバーが数分で起動します。
8. サーバーが起動したら、デモ ページから次の手順で要求を実行します。
  - a) `http://<サーバー名>:<ポート>/Spatial/MappingService/DemoPage.html` を開き、名前付きマップ リストのリクエストを実行します。
  - b) `http://<サーバー名>:<ポート>/Spatial/FeatureService/DemoPage.html` を開き、テーブル名リストのリクエストを実行します。
9. `<インストール先>\Pitney Bowes\Spectrum\java64\bin` に移動し、`jconsole.exe` を実行します。
10. ローカル プロセスの中からラッパー プロセスを選択します。
11. Jconsole で、新しいセッションを追加し、Feature Service プロセスを選択します。
12. Jconsole で、新しいセッションを追加し、Mapping Service プロセスを選択します。
13. Jconsole を実行したままにして、Spectrum プラットフォーム ラッパーの Feature Service と Mapping Service のメモリ、CPU、スレッドなどを監視します。



## 空間リモート コンポーネントのヒープメモリの増加

空間サービス要求を処理する Location Intelligence モジュールのリモート コンポーネントで使用されるヒープメモリを増やすには

1. Spectrum™ Technology Platform サーバーを停止します。
2. テキスト エディタで、`java.vargs` ファイルを `<Installed>\Pitney Bowes\Spectrum\server\modules\spatial\java.vargs` から開きます。
3. `vargs` の値を変更します。デフォルトは 2GB (2048MB) です。例えば、リモート コンポーネントのメモリを 4GB に増やすには、`vargs` をデフォルトの `-Xmx2048m` から `-Xmx4096m` に変更します。お使いのオペレーティングシステムで使用可能な最大メモリ容量を超えない

ようにし、またオペレーティング システム自体の処理のための適切な容量を残すようにしてください。

4. `java.vargs` ファイルを保存します。
5. Spectrum™ Technology Platform サーバーを再起動します。

# 著作権に関する通知



© 2019 Pitney Bowes. All rights reserved. MapInfo および Group 1 Software は Pitney Bowes Software Inc. の商標です。その他のマークおよび商標はすべて、それぞれの所有者の資産です。

### USPS® 情報

Pitney Bowes Inc. は、ZIP + 4® データベースを光学および磁気媒体に発行および販売する非独占的ライセンスを所有しています。CASS、CASS 認定、DPV、eLOT、FASTforward、First-Class Mail、Intelligent Mail、LACS<sup>Link</sup>、NCOA<sup>Link</sup>、PAVE、PLANET Code、Postal Service、POSTNET、Post Office、RDI、Suite<sup>Link</sup>、United States Postal Service、Standard Mail、United States Post Office、USPS、ZIP Code、および ZIP + 4 の各商標は United States Postal Service が所有します。United States Postal Service に帰属する商標はこれに限りません。

Pitney Bowes Inc. は、NCOA<sup>Link</sup>® 処理に対する USPS® の非独占的ライセンスを所有しています。

Pitney Bowes Software の製品、オプション、およびサービスの価格は、USPS® または米国政府によって規定、制御、または承認されるものではありません。RDI™ データを利用して郵便送料を判定する場合に、使用する郵便配送業者の選定に関するビジネス上の意思決定が USPS® または米国政府によって行われることはありません。

### データ プロバイダおよび関連情報

このメディアに含まれて、Pitney Bowes Software アプリケーション内で使用されるデータ製品は、各種商標によって、および次の 1 つ以上の著作権によって保護されています。

© Copyright United States Postal Service. All rights reserved.

© 2014 TomTom. All rights reserved. TomTom および TomTom ロゴは TomTom N.V の登録商標です。

© 2016 HERE

Fuente: INEGI (Instituto Nacional de Estadística y Geografía)

電子データに基づいています。© National Land Survey Sweden.

© Copyright United States Census Bureau

© Copyright Nova Marketing Group, Inc.

このプログラムの一部は著作権で保護されています。© Copyright 1993-2007 by Nova Marketing Group Inc. All Rights Reserved

© Copyright Second Decimal, LLC

© Copyright Canada Post Corporation

この CD-ROM には、Canada Post Corporation が著作権を所有している編集物からのデータが収録されています。

© 2007 Claritas, Inc.

Geocode Address World データ セットには、  
<http://creativecommons.org/licenses/by/3.0/legalcode> に存在するクリエイティブ コモンズ アトリビューション ライセンス (「アトリビューション ライセンス」) の下に提供されている GeoNames Project ([www.geonames.org](http://www.geonames.org)) からライセンス供与されたデータが含まれています。お客様による GeoNames データ (Spectrum™ Technology Platform ユーザ マニュアルに記載) の使用は、アトリビューションライセンスの条件に従う必要があります。お客様と Pitney Bowes Software, Inc. との契約と、アトリビューション ライセンスの間に矛盾が生じる場合は、アトリビューションライセンスのみに基づいてそれを解決する必要があります。お客様による GeoNames データの使用に関しては、アトリビューション ライセンスが適用されるためです。



3001 Summer Street  
Stamford CT 06926-0700  
USA

[www.pitneybowes.com](http://www.pitneybowes.com)