



DISCOVER > ANALYSE > ACT

# MapLink OGC Services Deployment User Guide

AUM1109|27 August 2021| Version\Status: 1.0\Approved

© **Envitia Ltd. 2019**

North Heath Lane, Horsham, West Sussex, RH12 5UX, United Kingdom

Tel: +44 1403 273 173 Email: [info@envitia.com](mailto:info@envitia.com)

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

### **Commercial in Confidence**

This document contains commercial company information.  
Communication to third parties without written consent from Envitia is forbidden

## AMENDMENT RECORD

ISSUE	PAGES	DESCRIPTION OF CHANGE	DATE
01	-	Initial Release	May 2007
02	16	Remove documentation for unused cache option	June 2009
03	17	Add documentation for JPEG_Compression_Factor option	August 2009
04	-	Raster Cache section removed as it was found to be slower then re-drawing the tile. Rebranded	November 2009
05	-	Corrected a number of errors and added documentation for UseBGCOLORasTransparentColour.	June 2010
06	-	Additions to the Common Problems section. Changed Tomcat Version to 7. Corrections and additional configuration options added.	July 2011
07	-	Updates for MapLink 7 including renaming to OGC Services Deployment Guide and extending to support the MapLink WPS service.	November 2011
08	-	Updates for MapLink 7.1	November 2012
09	-	Updates for MapLink 7.5. The plugins for the WMS are now in a plugins directory.	March 2013
10	-	Changed MapLink version to 8.1 and fixed broken cross references. Licence requirements is made clearer in section 2.4 WPS information updated. WPS Plugin configurations added.	Aug 2015
12	-	Added information for Linux installation	April 2017
14	-	Added Docker information	September 2017

15	-	Configuration of supported map formats added	April 2018
16	-	Clarification of Docker setup	June 2019
17	-	Added information for text anti-aliasing configuration	September 2019

## TABLE OF CONTENTS

<b>1. INTRODUCTION .....</b>	<b>1</b>
1.1. Pre-requisites .....	1
<b>2. THE MAPLINK OGC SERVICES SDK .....</b>	<b>2</b>
2.1. Introduction .....	2
2.2. Using the OGC Services SDK Directly.....	3
2.3. Available APIs .....	3
2.4. Licencing.....	4
<b>3. DEPLOYING MAPLINK OGC SERVICES ON A JAVA WEB SERVER.....</b>	<b>5</b>
3.1. Introduction .....	5
3.2. Configuring the Native Library path.....	5
3.2.1. Apache Tomcat 7.0.22.....	5
3.2.2. Apache Tomcat 7.0.22 Running As A Windows Service .....	6
3.3. Installing the shared MapLinkOGCServices library .....	6
3.3.1. Linux Specific .....	6
3.3.2. Apache Tomcat 7.0.22.....	7
3.4. Configuring a deployment.....	7
3.5. Deploying the OGC Services Servlet.....	8
3.5.1. Apache Tomcat 7.0.22.....	8
3.6. Testing the Deployment.....	8
3.7. Common Problems.....	8
<b>4. DEPLOYING MAPLINK OGC SERVICES ON A .NET WEB SERVER .....</b>	<b>12</b>
4.1. Introduction .....	12
4.2. Creating the Project .....	12
4.3. Building and Deploying .....	12
4.3.1. Reconfiguring to use debug assemblies .....	12
4.3.2. Deploying to IIS .....	13
4.4. Configuring a deployment.....	13
4.4.1. Setting the working directory.....	14
4.5. Configuring the Application Pool .....	14
4.6. Common Problems.....	15
<b>5. THE MAPLINK WMS SERVICE .....</b>	<b>17</b>
5.1. Web Map Service Introduction .....	17
5.2. The Structure of the MapLink WMS.....	17
5.2.1. Introduction .....	17
5.2.2. Plug-Ins to the MapLink WMS.....	18
5.2.3. Data Sources.....	19
5.3. Configuring a MapLink WMS.....	20
5.3.1. Introduction .....	20
5.3.2. Heading 'Server_Data_Sources_Index' .....	20
5.3.3. Heading 'Server_Properties' .....	21
5.3.4. Heading 'Service_Contact' .....	22
5.3.5. Heading 'Service_Addresses' .....	24
5.3.6. Heading 'Root_Layer_Details' .....	27
5.3.7. Heading 'Response_Configuration_Details' .....	27
5.3.8. Heading 'MapLink_Standard_Configuration' .....	30
5.3.9. Example.....	31
5.4. Supplied MapLink WMS Plug-Ins.....	32
5.4.1. Introduction .....	32
5.4.2. The Basic Map Plug-In .....	32
5.4.3. The Historical Map Plug-In .....	33
5.4.4. The Super Map Plug-In .....	33
5.4.4.1. Single Map Mode .....	34

- 5.4.4.2. Multi-Map Mode ..... 36
- 5.4.5. The CDRG Map Plug-In ..... 38
- 5.5. Common Problems ..... 39
- 5.6. Docker ..... 41
  - 5.6.1. Prerequisites: ..... 41
  - 5.6.2. Building The WMS Docker Image ..... 41
  - 5.6.3. Running The WMS Container ..... 41
  - 5.6.4. Stopping The Container ..... 41
  - 5.6.5. Customising Your WMS Configuration ..... 41
  - 5.6.6. Mount Points ..... 42
  - 5.6.7. Generating WMS Config XMLs ..... 42
    - 5.6.7.1. Generating WMS Configuration For A Single Map ..... 42
    - 5.6.7.2. Generating WMS Configuration XMLs For A Directory of Maps ..... 42
- 6. THE MAPLINK WPS SERVICE ..... 43**
  - 6.1. Web Processing Service Introduction ..... 43
  - 6.2. The Structure of the MapLink WPS ..... 43
    - 6.2.1. Introduction ..... 43
  - 6.3. Configuring a MapLink WPS ..... 43
    - 6.3.1. MapLink WPS Configuration File ..... 43
    - 6.3.2. The DataSources element ..... 43
    - 6.3.3. The DataStore element ..... 44
    - 6.3.4. The Options element ..... 44
    - 6.3.5. The DefaultLanguage and LanguageSpecificMetadata elements ..... 44
      - 6.3.5.1. The OperationsMetadata element ..... 44
    - 6.3.6. Example ..... 44
  - 6.4. WPS Router Plugin ..... 48
    - 6.4.1. Deployment ..... 48
    - 6.4.2. Describe Process ..... 48
    - 6.4.3. Execute ..... 49
      - 6.4.3.1. Available Data Input Parameters ..... 49
      - 6.4.3.2. Response Document/ Raw Data Parameters ..... 50
  - 6.5. WPS View Shed Plugin ..... 51
    - 6.5.1. Deployment ..... 51
    - 6.5.2. Describe Process ..... 51
    - 6.5.3. Single View Shed Execute ..... 52
      - 6.5.3.1. Available Data Input Parameters ..... 52
      - 6.5.3.2. Response Document/ Raw Data Parameters ..... 53
    - 6.5.4. Multi View Shed Execute ..... 54
      - 6.5.4.1. Available Data Input Parameters ..... 54
      - 6.5.4.2. Response Document/ Raw Data Parameters ..... 55
    - 6.5.5. Route View Shed Execute ..... 57
      - 6.5.5.1. Available Data Input Parameters ..... 57
      - 6.5.5.2. Response Document/ Raw Data Parameters ..... 58
    - 6.5.6. Route Breakdown Execute ..... 60
      - 6.5.6.1. Available Data Input Parameters ..... 60
      - 6.5.6.2. Response Document/ Raw Data Parameters ..... 61
  - 6.6. WPS Terrain Profile Plugin ..... 61
    - 6.6.1. Deployment ..... 61
    - 6.6.2. Describe Process ..... 62
    - 6.6.3. Execute ..... 62
      - 6.6.3.1. Available Data Input Parameters ..... 62
      - 6.6.3.2. Response Document/ Raw Data Parameters ..... 63
  - 6.7. WPS Import Raster Plugin ..... 64
    - 6.7.1. Deployment ..... 64
    - 6.7.2. Describe Process ..... 64
    - 6.7.3. Execute ..... 65

6.7.3.1.	Available Data Input Parameters .....	65
6.7.3.2.	Response Document/ Raw Data Parameters .....	66
<b>7.</b>	<b>APPENDIX A.....</b>	<b>67</b>
7.1.	WPS Plugin Sample Execute POST Calls .....	67
7.1.1.	Router.....	67
7.1.2.	Single View Shed .....	70
7.1.3.	Multi View Shed .....	74
7.1.4.	Route View Shed .....	76
7.1.5.	Route Breakdown.....	78
7.1.6.	Terrain Profile.....	78

## 1. INTRODUCTION

This document describes how to deploy one of the MapLink Open Geospatial Consortium (OGC) services, on a variety of proprietary web serving software. Currently MapLink supports for following OGC services:

- The MapLink Web Map Service (WMS). A WMS is used to serve up user defined map data, in a standardised format, for use by client software across a network.
- The MapLink Web Processing Service (WPS). A WPS is offers general purpose processing services that can be submitted and the results retrieved across a network.

It is intended that further MapLink OGC services will be release in the future.

It is assumed that users of this guide have a basic understanding of the MapLink SDKs and the OGC produced standard for the OGC service being deployed.

### 1.1. Pre-requisites

In addition to these notes and sample data you will need:

- A server with Envitia MapLink Pro installed.
- Some of the MapLink OGC services require an appropriate licence to have been installed on the machine. These licences can be retrieved from Envitia, either as an evaluation or permanent licence, for either an SDK or standard deployment. An SDK deployment uses debug libraries and is intended to allow development of user created plug-ins for the OGC service, while standard deployments use release libraries and offers the best performance.
- The MapLink WPS Service requires a valid deployment licence. Please contact [sales@envitia.com](mailto:sales@envitia.com) for additional information.
- Third-party Web Server software for deploying the WMS. Examples in this guide cover the following Web Server software:
  - Apache Tomcat 7.0.22
  - Microsoft IIS 5.1/6/6.1

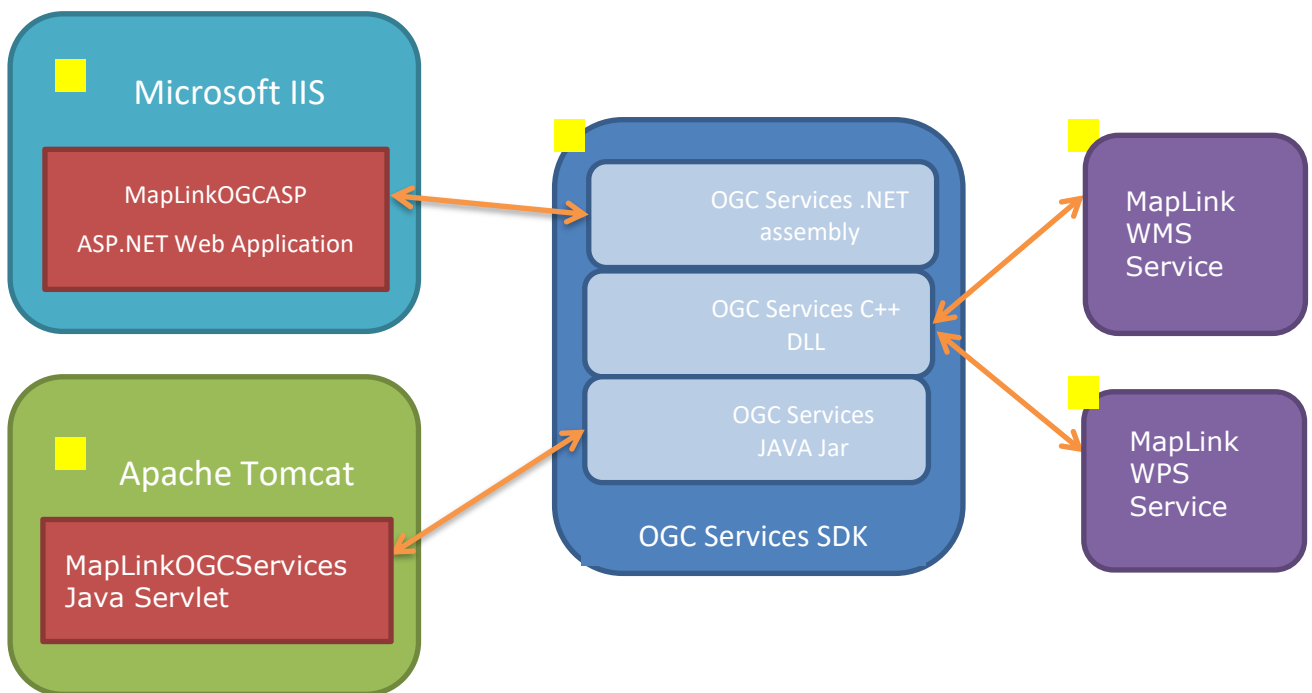
## 2. THE MAPLINK OGC SERVICES SDK

### 2.1. Introduction

The OGC Services SDK provides the ability to construct instances of Envitia provided OGC service implementations, such as the MapLink WMS. This SDK abstracts the actual service from the API used to construct the service. Each OGC Service exists as a plug-in to the OGC Services SDK, while each OGC Service itself may also have its own plug-ins for providing data sources.

When deploying an OGC service, it is usually unnecessary to use the MapLink OGC Services SDK directly, but it may be helpful to have understanding of its use.

The following diagram demonstrates the use of this SDK



**Figure 1 - How the OGC Services SDK is used.**

1. The supplied MapLinkOGCASP ASP.NET project can be used to deploy a MapLink OGC Service using Microsoft IIS.
2. The supplied MapLinkOGCServices Java Servlet can be used to deploy a MapLink OGC Service using a Java Web Server such as Tomcat.
3. The MapLink OGC Services SDK offers three APIs - C++, .NET and Java. Both the .NET and Java use the C++ library internally.
4. Through the OGC Services SDK, a MapLink OGC Service can be constructed, configured and used.



## 2.2. Using the OGC Services SDK Directly

An Envitia OGC service instance can be constructed via the `TSLOGCService1` class' static `create` method, passing in the name of the OGC Service, E.G. "MapLinkWMS" for the MapLink WMS. The service name should not contain any suffixes, such as 'd' to imply debug or '64' to imply constructing a 64-bit service.

Assuming that a service instance is returned, the next step is to configure the service by passing in the location to a service specific configuration file to the `loadConfiguration` method. The format of the configuration file is described later in this document for each of the OGC Service offered.

Once the service has been constructed and configured, it is ready to use. Requests can be made via the `processGetRequest` or `processPostRequest` methods, which will return an instance of the `TSLOGCMIMEResponse` class. Not all service types support both HTTP GET and POST requests - the WMS for instance only supports HTTP GET. The call to the service should include the address that the request was made via and the contents of the request.

The response object principally holds the raw service response data, but also the following information:

- The 'Multipurpose Internet Mail Extensions' (MIME) type of the response. The MIME type denotes how a client should interpret the raw data and its value is dependent upon the service type, the request made and the success/failure of request.
- The encoding of the response, although this may be NULL if it is not applicable. Certain response formats may not be fully described by their MIME type, so this provides additional information.
- The cacheability of the response. This provide a hint as to the duration of validity of the response. Error messages often should not be cached, whereas normal responses can often be cached both locally and remotely to reduce server traffic and workload.
- HTTP code. Some of the newer OGC services require that when an error message is returned the HTTP code of the response should be set to reflect the type of error.

## 2.3. Available APIs

Currently the OGC Services SDK is available in the following APIs:

- A C++ API – All other APIs utilise this API, but it can be used directly by users. Provided through the `MapLinkOGCServices64` library.
- A .NET wrapper API – Provided through the `Envitia.MapLink.OGCServices64` assembly
- A Java API – Provided through the `MapLinkOGCServices64` JAR library

All three APIs are provided in 64-bit debug and 64-bit release forms, with the normal MapLink suffixes used to differentiate between them. The release forms should be used for a final system while the debug form is intended to allow users to create their own plug-ins to a particular OGC Service.

---

<sup>1</sup> Equivalent classes, with similar names and methods, exist in the .NET and Java OGC Services APIs

## 2.4. Licencing

For Deployment you must use do the following:

- Use Release DLLs.
- Obtain a 'MapLink WMS (Deployment)' Licence from Envitia<sup>2</sup>.

For Development you must do the following:

- Use the Debug DLLs.
- Obtain the necessary Development licences from Envitia ('MapLink Pro Developer's Toolkit' and 'MapLink WMS SDK' as a minimum).

---

<sup>2</sup> You will need to have obtained any additional runtime licences as well as the deployment licence. Standard deployments include Core runtime.

## 3. DEPLOYING MAPLINK OGC SERVICES ON A JAVA WEB SERVER

### 3.1. Introduction

To provide access from Java compatible Web Servers, a Java Wrapper library is supplied with the MapLink OGC Services SDK which uses the Java Native Interface to access the OGC Services SDK's C++ libraries. Additionally a pre-built Java Servlet is supplied which handles communication with the JNI wrapper and serves HTTP GET requests made to it.

This section outlines how to deploy this example Servlet to the Tomcat web server, although the steps should be fairly similar when using other Java based web servers

### 3.2. Configuring the Native Library path

The OGC Services SDK requires access to the appropriate MapLink bin directory in order to access the native C++ libraries and any WMS plug-ins. When MapLink is installed on Windows base machines, the `PATH` environment variable is modified to add the `bin64` directory from the MapLink installation.

If the bin directory referenced is not the one being targeted or the `PATH` environment variable has been edited to remove MapLink, then this section should be followed. Otherwise it can be ignored.

Non Windows platforms will also need to follow these instructions, but the `lib64` directory is referenced.

#### 3.2.1. Apache Tomcat

The instructions under this immediate heading are only applicable when Tomcat is not deployed as a service/daemon.

- Navigate to the following directory
- `$CATALINA_HOME/bin`
- For Windows platforms
  - Create a file named  
`setenv.bat`
  - Its contents should be in the form

```
set MY_BIN_DIR=c:\Program files\Envitia\MapLink Pro\X.Y\bin64
set PATH=%PATH%;%MY_BIN_DIR%;%MY_BIN_DIR%\plugins
```
- For other platforms
  - Create a file named  
`setenv.sh`
  - Its contents should be in the form

```
#!/bin/sh
MAPL_HOME=</path/to/maplink>
export MAPL_HOME
LD_LIBRARY_PATH="$LD_LIBRARY_PATH:$MAPL_HOME/lib64:
$MAPL_HOME/lib/plugins"
```

```
export LD_LIBRARY_PATH
```

- When the server starts it will pick up the created file automatically and run it
- Tomcat will need to be restarted for this change to take effect.

### 3.2.2. Apache Tomcat Running As A Windows Service

- Start the "Configure Tomcat" shortcut found under the Tomcat group in the Start Menu.
- Under the "Startup" tab of the application that starts, change the working path setting to point at the appropriate bin directory.
- **Ensure you click the Apply button after making this change.** It would appear that the change does not always get registered if you don't.
- Tomcat will need to be restarted for this change to take effect.

### 3.3. Installing the shared MapLinkOGCServices library

To allow multiple OGC Services or multiple service instances to be deployed on a single Web Server, a MapLink library, `MapLinkOGCServices`, must be added to the servers common class path directory. As this jar file loads the appropriate native C++ library, there are 4 different versions that could be used:

- `MapLinkOGCServices64.jar` - 64-bit Release JAR
- `MapLinkOGCServices64d.jar` - 64-bit Release JAR

For Deployment you must use do the following:

- Use a Release JAR file.
- Obtain a 'MapLink WMS (Deployment)' Licence from Envitia<sup>3</sup>.

For Development you must do the following:

- Use a Debug JAR file.
- Obtain the necessary Development licences from Envitia ('MapLink Pro Developer's Toolkit' and 'MapLink WMS SDK' as a minimum).

It is also important to note that the 64-bit JARs should only be used with a 64-bit Tomcat/Java Runtime Environment (JRE) installation.

The JAR files can usually be located:

```
<MAPLINK_INSTALL_DIR>\java\MapLinkOGC\
```

#### 3.3.1. Linux Specific

For non-Windows platforms the same JAR file is used for all deployment types, `MapLinkOGCServices64.jar`. Debug and Release is not a concept on non-Windows platforms. The `lib64` directory should have been referenced when completing the instructions from section 3.2.

---

<sup>3</sup> You will need to have obtained any additional Runtime licences as well as the deployment licence. Standard Deployments include the Core runtime

### 3.3.2. Apache Tomcat

Place the required JAR file in the following directory

```
$CATALINA_HOME/lib
```

Tomcat will need to be restarted for this change to take effect.

## 3.4. Configuring a deployment

Each deployment of a MapLink OGC Service, such as the MapLink WMS, requires a configuration file to load such settings as the plug-ins, spatial data and data source configuration files used. The contents of this configuration file are service specific and are described in later sections of this document.

By default the supplied Java Servlet is configured for use with the MapLink WMS and with the location of a configuration file expected to be at `./MapLinkWMSConfiguration.ini`. If a different service type, multiple instances of the same service type or if this location is not suitable for a particular server configuration, the settings contained in the war file will need to be changed.

Example configuration files can be found in `MAPLINK_INSTALL_DIR\config\ogcservices`. The file paths within these configs will need to be edited, to reflect the location of the MapLink installation.

The supplied Java Servlet, called `MapLinkOGCServices.war`, is normally located:

```
MAPLINK_INSTALL_DIR\java\MapLinkOGC\
```

The following instructions should be followed to edit the servlet's settings:

- Using a zip utility (such as PKZip, WinZip, 7zip or WinRAR) unzip the war file to disk, ensuring that the contained folder structure is maintained. It may be necessary to change the extension of the war file to `.zip` for it to be recognised.
- Under the `WEB-INF` directory of the unzipped files, edit the `web.xml` file using a text editor.
- Roughly halfway through the file there should appear the following snippet:

```
<init-param>
  <param-name>ServicePlugin</param-name>
  <param-value>MapLinkWMS</param-value>
</init-param>
<init-param>
  <param-name>ServiceConfigurationFile</param-name>
  <param-value>./mapLinkwmsconfiguration.ini</param-value>
</init-param>
```

- The `MapLinkWMS` string may be edited to target a different service type.
- The string `'./mapLinkwmsconfiguration.ini'` may be edited to point at a different service configuration file. See 4.4 Configuring a Deployment.
- The files unzipped earlier will need to be either re-added to the war file or zipped into a new archive. The folder structure must be maintained in the archive and the extension may need to be reverted to `.war`.

## 3.5. Deploying the OGC Services Servlet

This section discusses how to deploy the Servlet on the Web Server and specifically how to deploy a second instance in a way that will not conflict with the first.

### 3.5.1. Apache Tomcat

- Log on to the 'Tomcat Web Application Manager', usually accessible from the following URL <http://127.0.0.1:8080/manager/html>. It may be necessary to setup a Tomcat user that has sufficient privileges to access to the Tomcat Manager first.
- Under the heading 'Deploy' and sub-heading 'WAR file to deploy', click on the 'Browse' button and upload the WMS war file
- Click 'Deploy'
- To deploy a second WMS, then simply create a copy of the war file with a different filename and follow the above instructions.

## 3.6. Testing the Deployment

The WMS service can be accessed with a request in the following form:

```
http://.../MapLinkOGCServices/OGC?
```

For instance, a WMS request for the service Capabilities metadata would be made as follows:

```
http://.../MapLinkOGCServices/OGC?service=WMS&request=GetCapabilities
```

## 3.7. Common Problems

Here is a list of some of the common problems that affect deployments to Java and how to resolve them. This list does not include issues that relate to deploying a particular service type which will be covered in later sections.

- When accessing the Servlet's URL you receive a 404 error stating that the Servlet is not available. This is likely to be caused by the shared `MapLinkOGCServices` library not being found. Return to section 3.3 and check that the instructions have been followed correctly.
- When accessing the Servlet's URL you receive a 500 error stating that the Servlet's `init()` for servlet `MapLinkOGCServices` threw an exception. This is usually due to the required C++ DLLs not being located by the runtime.

Return to section 3.2 and check that the instructions have been followed correctly.

This error can also arise when there is a mismatch between the architecture of the JRE being used and the Envitia libraries being loaded. The 64-bit Envitia libraries can only be loaded by a 64-bit JRE.

- For Unix/Linux platforms, Tomcat will use the JRE located in the system path by default. To use a different JRE (for example if the default JRE is 32bit), set the `JRE_HOME` environmental variable before starting the service. E.G. from the tomcat bin directory running `'JRE_HOME=/path/to/64bit_jre ./startup.sh'` will start Tomcat using the JRE installed in `/path/to/64bit_jre`.
- For Unix/Linux platforms, and errors stating that '`GLIBCXX_3.4.11 not found`' or similar messages. Check the X11 Release Notes and ensure that the correct gcc runtime dependencies have been installed.

- Plugins cannot be loaded. The plugins have been moved to a `plugins` directory in the `bin64` folder.
- Evaluation Version on Linux requires a node locked licence. Please ask [support@envitia.com](mailto:support@envitia.com) for help or check the supplied documentation.
- Linux Checklist:

The following needs to be deployed on Linux:

Directories:

- `config`
- `lib64`

The java file:

- `MapLinkOGCServices64.jar`

The WAR file (see 3.3):

- `MapLinkOGCServices.war`

Java/tomcat must be told where to find the MapLink shared libraries and Java files. How this is done will vary between a manual start of tomcat and a `service/daemon`. Please see section 3.2

The following environment variable must be set to point to the base MapLink deployment or installation directory (the directory that contains the `config` and `lib/lib64` directory):

```
MAPL_HOME
```

Check the following environment variables (update if necessary to point to the MapLink directory `lib64` and the plugin directories contained within):

```
PATH
```

```
LD_LIBRARY_PATH
```

The following needs to be edited (found in `MapLinkOGCServices.war`):

- `web.xml`

Please see section 3.4 for more information. Specifically the `'param-value'` for `'ServiceConfigurationFile'` must be updated to point to a configuration file.

It is advised that an absolute path is used to point to the configuration file.

You are reminded that the paths and filenames are case sensitive.

The `'ServiceConfigurationFile'` specifies the WMS plugin to use for a Map and the Map's configuration file.

Newer versions of MapLink Studio will create a WMS `config.xml` for a map. For maps generated using an older version of MapLink Studio you can use the `BMCCreator.exe` on Windows.

- The file that the `'ServiceConfigurationFile'` entry in the `web.xml` needs to be configured for each map (see sections 5.3 and 5.3.9). For Linux the `'Server_Data_Sources_Index'` must be correctly setup with the plugin names and map locations specifically for Linux (paths will be different and the plugin names may be slightly different).

For example the following is a valid entry for the SuperMap plugin on Linux:

```
[Server_Data_Sources_Index]
DatasourceCount=1
Datasource0Location=/projects/TestAndData/output.map
Datasource0Plugin=SuperMapWMS_plugin
Datasource0Configuration=/projects/TestAndData/config.xml
```

For simplicity it is advised to initially use the basicmap plugin before considering to use the SuperMap plugin.

- The `Datasource0Configuration` must point to a valid 'MapLink Pro' Map WMS configuration file.

Newer versions of MapLink Studio will create a WMS `config.xml` for a map. For maps generated using an older version of MapLink Studio you can use the `BMCCreator.exe` on Windows.

It is advised to keep the `config.xml` file alongside the `.map` file.

An example of this file is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<BasicMapConfiguration
  xmlns="http://www.envitia.com/schemas/maplinkwms/basicmapplugin"
  xmlns:wms="http://www.opengis.net/wms"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xlink="http://www.w3.org/1999/xlink"

  xsi:schemaLocation="http://www.envitia.com/schemas/maplinkwms/basicmapplugin
  http://www.envitia.com/schemas/maplinkwms/basicmapplugin/1.0/BasicMapConfiguration.xsd">
  <wms:Name>Map0</wms:Name>
  <wms:Title>Map0</wms:Title>
  <wms:EX_GeographicBoundingBox>
  <wms:westBoundLongitude>-76.000139</wms:westBoundLongitude>
  <wms:eastBoundLongitude>-74.999861</wms:eastBoundLongitude>
  <wms:southBoundLatitude>44.999861</wms:southBoundLatitude>
  <wms:northBoundLatitude>46.000139</wms:northBoundLatitude>
  </wms:EX_GeographicBoundingBox>
  <wms:BoundingBox CRS="EPSG:0" minx="-74.983925" miny="45.997687"
  maxx="-76.011171" maxy="44.999861"/>
  <ImplementedFeaturesList>
  <ImplementedFeature>Rasters</ImplementedFeature>
  </ImplementedFeaturesList>
</BasicMapConfiguration>
```

- If debugging an installation set the following environment variables:

```
export ENV_VERBOSE=verbose,lasterror
export ENV_WMS_DEBUG=1
```



```
export ENV_WMS_DEBUG_STARTUP=1
```

This will enable additional error messages to be output to the console.

## 4. DEPLOYING MAPLINK OGC SERVICES ON A .NET WEB SERVER

### 4.1. Introduction

To allow the MapLink WMS to be deployed on a .NET compatible Web Server the MapLink OGC Services .NET assembly should be used. This section describes how to deploy the supplied ASP.NET C# web application to a Microsoft Internet Information Services (IIS) Web Server.

These instructions are aimed at new users to IIS and should be treated as a way of configuring a basic deployment. For security critical environments, consult your system administrator or the IIS help pages for more complete instructions.

**NOTE:** For users familiar with previous versions of the MapLinkOGCASP sample application, from MapLink Pro 7.0 onwards the main page from `OGC.aspx` to `OGC.ashx`. It should still be possible to access `OGC.aspx` due to redirect logic in the `web.config`.

### 4.2. Creating the Project

The first step to deploying the ASP.NET project is to copy the entire sample directory found at

```
<MAPLINK_INSTALL_DIR>\samples\NT\MapLinkOGCASP
```

to a new location on your hard-drive. Ideally this should be either the root or a sub directory of the root share directory of the Web Server. In a standard IIS setup this will be

```
C:\InetPub\wwwroot\
```

### 4.3. Building and Deploying

Open the `MapLinkOGCASPx.sln` solution using Microsoft Visual Studio 2010<sup>4</sup> SP1. If using Windows Vista, 7 or Server 2008, it is likely that you'll need to run Visual Studio "As Administrator" to allow interaction with IIS.

The MapLinkOGCASP project is dependent upon the `Envitia.MapLink.OGCServices` assembly, but you'll not find it included in the References section of the project. This is down to the fact that Visual Studio does not support the concept of using different assemblies for different build configurations which MapLink requires. Microsoft MSBuild, which is used by all versions of Visual Studio, does support this concept however. The import of the Envitia assembly will therefore be used when building the web application, but cannot be seen through the Visual Studio interface.

#### 4.3.1. Reconfiguring to use debug assemblies

The project is configured to load the Release assemblies, which correspond to the deployment Envitia licence type, when building the project. To redirect the project to use the debug assemblies, which correspond to the SDK Envitia licence type, use the following instructions:<sup>5</sup>

---

<sup>4</sup> The older `MapLinkOGCASP.sln` solution, that targets version 2.0 of the Microsoft .NET framework, could be used instead in either Microsoft Visual Studio 2005 SP1 or upgraded to Microsoft Visual Studio 2008 SP1.

<sup>5</sup> The assembly project name and assembly paths referenced in this section will differ when using Visual Studio 2005/2008.

- Right click on the project node in the Solution Explorer window. Select "Unload Project" from the context menu.
- Right click on the now greyed project node in the Solution Explorer window. Select "Edit MapLinkOGCASP.csproj" from the context menu.
- Approximately halfway down the file there should be the line:

```
<HintPath>c:\Program Files\Envitia\MapLink Pro\X.Y\Bin\  
    Envitia.MapLink.OGCServices.dll</HintPath>
```

This should be changed to

```
<HintPath>c:\Program Files\Envitia\MapLink Pro\X.Y\Bin\  
    Envitia.MapLink.OGCServicesd.dll</HintPath>
```

- Around 10 lines further down there should be the line:

```
<HintPath>c:\Program Files\Envitia\MapLink Pro\X.Y\Bin64\  
    Envitia.MapLink.OGCServices64s.dll</HintPath>
```

This should be changed to

```
<HintPath>c:\Program Files\Envitia\MapLink Pro\X.Y\Bin64\  
    Envitia.MapLink.OGCServices64d.dll</HintPath>
```

- Finally, right click on the greyed project node in the Solution Explorer windows. Select "Reload Project" from the context menu

### 4.3.2. Deploying to IIS

Before deploying the project, build it using the "Build Solution" menu item under the Build menu. It must build successfully before continuing.

The MapLinkOGCASP web application is initially configured to use Visual Studio's built-in Web Server software, so will need to be changed to using IIS:

- Right click on the project node in the Solution Explorer window. Select "Properties" from the context menu.
- Select the "Web" tab.
- Change the radio box select to "Use Local IIS Web Server"<sup>6</sup>
- Alter the "Project Url" setting, if desired, and then click the "Create Virtual Directory" button.<sup>7</sup>

## 4.4. Configuring a deployment

Each deployment of a MapLink OGC Service, such as the MapLink WMS, requires a configuration file to load such settings as the plug-ins, spatial data and data source configuration files used. The contents of this configuration file are service specific and are described in later sections of this document.

---

<sup>6</sup> This label is called "Use IIS Web server" in Visual Studio 2005.

<sup>7</sup> When using Visual Studio 2010, if the required .NET framework has not been registered with IIS you may be prompted to manually register it. Instructions on how to achieve this are included in section 4.6.

By default the supplied MapLinkOGCASP web application is configured for use with the MapLink WMS and with the location of a configuration file expected to be at `./MapLinkWMSConfiguration.ini`. If a different service type, multiple instances of the same service type or if this location is not suitable for a particular server configuration, the settings contained in the `web.config` file will need to be changed. This file is located in the MapLinkOGCASP project.

The following block, which appears at the start of the file, controls these settings:

```
<appSettings>
  <add key="ServicePlugin" value="MapLinkWMS"/>
  <add key="ServiceConfigurationFile" value="./MapLinkWMSConfiguration.ini"/>
</appSettings>
```

#### 4.4.1. Setting the working directory

There is an additional setting that can be used but which is not listed. This controls from where the native dependencies of the .NET project are loaded from. The OGC Services SDK requires access to the appropriate MapLink bin directory in order to access the native C++ libraries and any WMS plug-ins. When MapLink is installed on the machine, the `PATH` environment variable is modified to add the `bin64` directory from the MapLink installation.

If the bin directory referenced is not the one being targeted or the `PATH` environment variable has been edited to remove MapLink, then this section should be followed. Otherwise it can be ignored.

To add the setting that controls the native dependency loading, place the following line inside the `appSettings` XML element

```
<add key="WorkingDirectory" value="c:\Program Files\Envitia\MapLink Pro\X.Y\bin64"/>
```

The value attribute should be changed to the appropriate directory.

## 4.5. Configuring the Application Pool

Microsoft's IIS runs web applications inside "Application Pools" which can be configured via the "Internet Information Services (IIS) Manager". This manager can be accessed from the "Administrative Tools" group in the Control Panel. The application pool assigned to the MapLinkOGCASP web application must be configured appropriately for the settings used in the application. By default, a 64-bit version of Windows will assign web applications to 64-bit application pools, which will cause the application to fail to work if it is built for 32-bit. Similarly an application pool is configured to target a particular .NET Framework version, either version 2.0 or 4.0 currently.

To change these settings on an application pool:

- For Windows Vista, 7 and Server 2008
  - Open the IIS Manager and locate the "Application Pools" node in the tree on the left of the application.
  - Right click on the target application pool and choose "Advance Settings..." from the context menu.
  - The "Enable 32-Bit Applications" setting controls the target architecture whilst the ".NET Framework Version" setting controls the .NET Framework version.
- For Windows XP and Server 2003
  - A 64-bit installation will by default be set to use all 64-bit processed in its application pool. It is not possible to have some 32-bit and some 64-bit

application pools, so all IIS processes need to be changed to running in 32-bit. To make this change run the following command:

```
cscript.exe C:\Inetpub\AdminScripts\adsutil.vbs
set W3SVC/AppPools/Enable32BitAppOnWin64 true
```

To revert IIS back to being 64-bit re-run the above command but changing the `true` to `false`.

- Windows XP and Server 2003 do not allow 32-bit and 64-bit versions of the .NET Framework to be registered at the same time. Therefore you'll need to unregister the 64-bit .NET Framework from IIS and register the 32-bit .NET Framework. When returning to 64-bit mode, you'll need to do the opposite.

To unregister a .NET Framework you'll need to call the appropriate `aspnet_regiis` version passing the `"-c"` parameter. To register a .NET Framework version, you need to do the same passing the `"-i"` parameter. The location of the different versions of `aspnet_regiis` is listed below:

- For the 32-bit version 4.0 .NET Framework:

```
c:\Windows\Microsoft.NET\Framework\v4.0.30319\aspnet_regiis
```

- For the 64-bit version 4.0 .NET Framework:

```
c:\Windows\Microsoft.NET\Framework64\v4.0.30319\aspnet_regiis
```

- For the 32-bit version 2.0 .NET Framework:

```
c:\Windows\Microsoft.NET\Framework\v2.0.50727\aspnet_regiis
```

- For the 64-bit version 2.0 .NET Framework:

```
c:\Windows\Microsoft.NET\Framework64\v2.0.50727\aspnet_regiis
```

- To change the version of the .NET framework that the web application uses, locate the `MapLinkOGCASPX` node in the IIS Manager, usually under the "Default Web Site" under the "Web Sites" node. Right click on it and choose "Properties" from the context menu. Change to the ASP.NET tab and change the version listed as desired.

## 4.6. Common Problems

Here is a list of some of the common problems that affect deployments to IIS and how to resolve them. This list does not include issues that relate to deploying a particular service type which will be covered in later sections.

- The `Inetpub` directory does not exist. This is likely to be caused by the IIS component of Windows not having been installed. This can be rectified using the "Turn Windows features on or off" accessed via the "Add / Remove Programs" feature found in Control Panel of Windows XP and Server 2003 or via the "Programs and Features" feature found in Control Panel of Windows Vista, 7 and Server 2008.
- When accessing the service, a 404.3 error is returned due to the "extension configuration". This is likely to be caused by the appropriate .NET framework not having been registered with IIS, which can happen if IIS is installed after it.

To rectify this one of the following commands should be run, as an administrator on Windows Vista, 7 and Server 2008. The particular command required is based upon the .NET framework and architecture in question.

- For the 32-bit version 4.0 .NET Framework:

```
c:\Windows\Microsoft.NET\Framework\v4.0.30319\aspnet_regiis -i
```

- For the 64-bit version 4.0 .NET Framework:  
`c:\Windows\Microsoft.NET\Framework64\v4.0.30319\aspnet_regiis -i`
- For the 32-bit version 2.0 .NET Framework:  
`c:\Windows\Microsoft.NET\Framework\v2.0.50727\aspnet_regiis -i`
- For the 64-bit version 2.0 .NET Framework:  
`c:\Windows\Microsoft.NET\Framework64\v2.0.50727\aspnet_regiis -i`
- When accessing the service, a Server Error is shown due to an attempt being "made to a load a program with an incorrect format". This is likely to be caused by a mismatch between the architecture of the compiler ASP.NET web application and the process pool running it. Return to section 4.53.3 and check that the instructions have been followed correctly.
- When accessing the server, a Server is shown with the message "External component has thrown an exception" and the top of the stack trace gives the line "TSLOGCService.create". This is likely to be caused by the native dependencies not being found by the runtime. Refer to section 4.4.1 for details.
- On Windows XP and Server 2003, when accessing the `OGC.ashx` page, a "The page cannot be found" error is return, but when attempting to browse the contained folder (E.G. "`http://.../MapLinkOGCASPx/`") an HTTP 403 error is returned stating that the website declined to show the web page. This implies that the web application has been deployed correctly, but is refusing to respond to the request. This may be caused by a number of things, but is most likely due to the ASP.NET "Web Service Extension" not being enabled.

To rectify this, open the IIS Manager and navigate to the "Web Service Extensions" node listed in the tree on the left of the application, at the same level as the "Application Pools" node. Locate the appropriate ASP.NET extension and ensure that it is set to "Allowed".

- On Windows Vista, 7 and Server 2008, when accessing the `OGC.ashx` page, a 404.2 error is returned stating that "The page you are requesting cannot be served because of the ISAPI and CGI Restriction list settings..". This implies that the ASP.NET "Web Service Extension" has not being enabled. The web page should list instructions on how to rectify this issue.

## 5. THE MAPLINK WMS SERVICE

### 5.1. Web Map Service Introduction

A Web Map Service (WMS) produces maps of spatially referenced data dynamically from geographic information. The WMS international standard defines three operations that can be performed on such a server; one returns service-level metadata, 'GetCapabilities'; another returns a map whose geographic and dimensional parameters are well-defined, 'GetMap'; and an optional third operation returns information about particular features shown on a map, 'GetFeatureInfo'.

A WMS is intended to be accessed either programmatically or using a standard web browser by submitting requests in the form of Uniform Resource Locators (URLs). The exact request string is dependent upon the operation being performed and the extra parameters that the operation requires. For instance a 'GetMap' request requires the width, height and geographic location, amongst other parameters, for the WMS to produce the returned image.

The data that a WMS serves is divided into layers, where a single layer can have zero or more sub layers. These layers are advertised via the 'GetCapabilities' operation's response, the service Capabilities. The service Capabilities are an XML document based upon a well a standard defined schema or DTD. The exact schema or DTD followed is dependent upon the version of standard, but are similar in structure across all versions.

Advertised layers can be generally split into two simple categories:

- Unnamed layers - These are often used to categorise their child layers or provide content information about their parent, but cannot be requested as part of a GetMap request.
- Named layers - These can be requested as part of GetMap request.

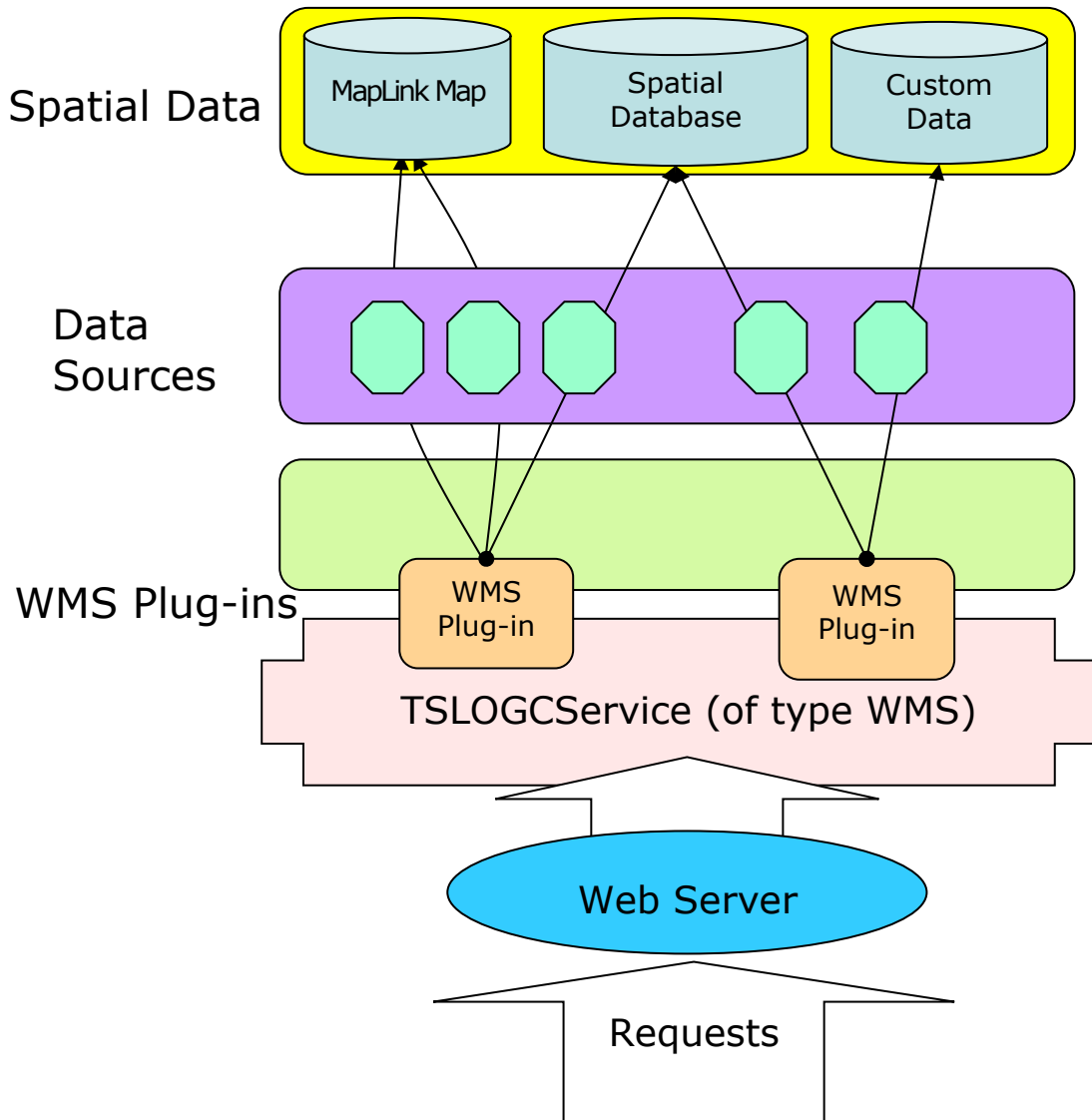
### 5.2. The Structure of the MapLink WMS

#### 5.2.1. Introduction

The MapLink WMS is designed to be as flexible as possible so that it can be used in many different scenarios. The server itself supports version 1.0.0, 1.1.0, 1.1.1 and 1.3.0 of the OGC standard. It uses a plug-in architecture, via a documented API, to load plug-ins that respond to incoming requests.

There are three terms that this section introduces: 'plug-ins', 'spatial data' and 'data sources'. Figure 2 demonstrates how these roles interact within an instance of a WMS. The 'spatial data' role refers to the data that is used to create the WMS response, the 'plug-in' refers to the library that interprets the 'spatial data' and the 'data source' is the combination of a 'spatial data' and 'plug-in' role.

The reason that these three roles are separate is that two different plug-ins may serve the same spatial data differently. Alternatively, even the same plug-in may serve the same spatial data in two different ways, resulting in two separate data sources. The previous example is permissible as each data source, or link between a spatial data role and plug-in role, is defined by the service configuration file.



**Figure 2 – The key roles within a MapLink WMS**

### 5.2.2. Plug-Ins to the MapLink WMS

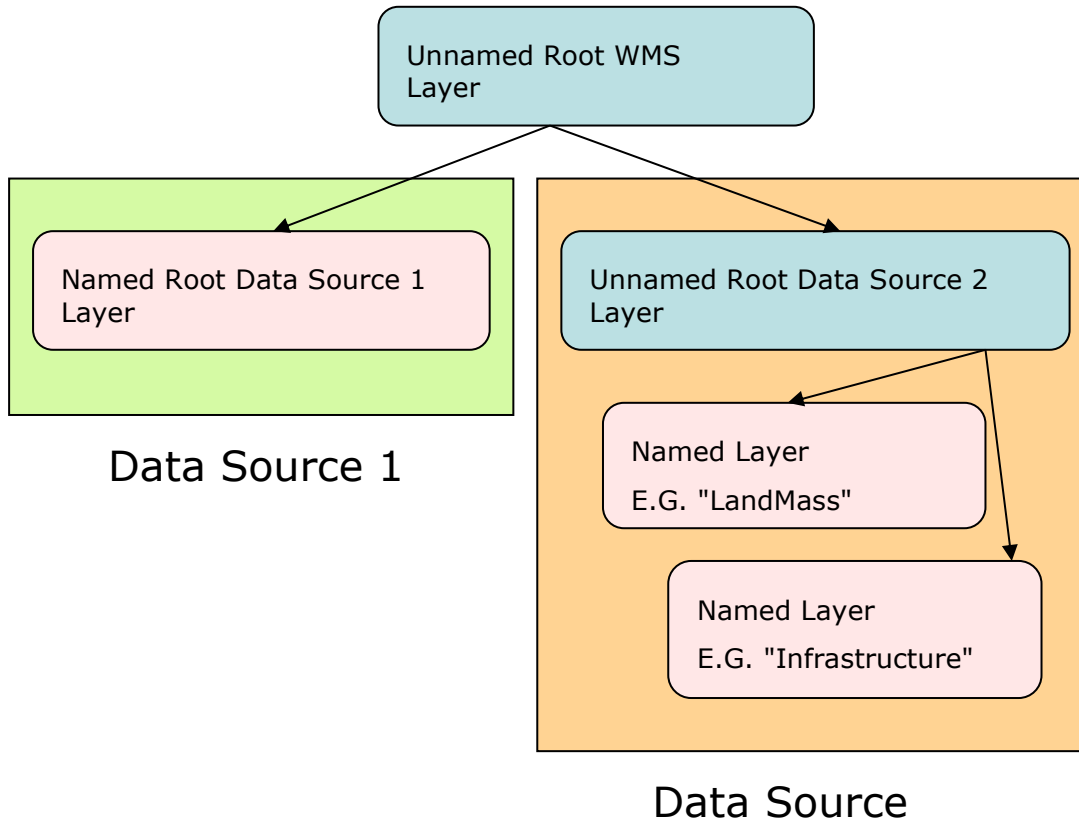
A plug-in to the MapLink WMS dictates both the type of spatial data that it can serve and the format of the configuration file that configures the data source it provides. The WMS service instance only sees the spatial data and configuration file as strings that are passed to the plug-in when creating a data source. For this reason these can refer to absolutely anything, although usually the configuration file will be a file path and the data source either a file path or database address.

A number of pre-built WMS plug-ins are supplied with MapLink which are detailed in section 5.4 of this document. Alternately a MapLink WMS Plug-In SDK is also supplied to allow users to create their own plug-ins using custom data in conjunction with the normal MapLink framework of SDKs. For more information on this SDK please consult the MapLink Developer's Guide.



5.2.3. Data Sources

Each deployed data source provides a child layer to the root unnamed layer in the service Capabilities. It should provide at least one named sub layer to permit serving data. Figure 3 demonstrates this basic layer structure using two data sources.



**Figure 3 – The Layer Structure within a MapLink WMS**

### 5.3. Configuring a MapLink WMS

#### 5.3.1. Introduction

An instance of the MapLink WMS requires a configuration file to define what data sources it should serve up and how. This configuration file also defines much of the service Capabilities as well as certain MapLink specific settings.

This section is intended to serve as a reference, along with the example configuration file at the end, to describe how to configure a server. The format of the configuration file is that of a Windows INI file and as such appears under headings. The sub headings of most of the following sections correspond to the expected headings in the file.

The Required column denotes whether the Key is required. If a required key is not provided a service exception will appear be returned whenever the service is accessed.

#### 5.3.2. Heading 'Server\_Data\_Sources\_Index'

Required	Key	Description
Y	DatasourceCount	Defines the number of data sources that are deployed on the WMS server. For each data source there is expected to be keys under this heading in the form:  DatasourceNLocation DatasourceNPlugin DatasourceNConfiguration where N defines the index of that data source starting from 0.
Y	DatasourceNLocation	Defines the location of the resource used by the data source. This may be a MapLink map or a database connection string. Its value is dependent upon the plug-in that appears as the following key. If this value defines a path then it should be either a fully qualified path or a path relative to the start up locale of the server.
Y	DatasourceNPlugin	The library name of the plug-in used to serve up this data source. This may be a custom defined plug-in using the MapLink WMS Plug-In API or a pre-built library supplied by Envitia.
Y	DatasourceNConfiguration	The location of a configuration file used by the plug-in to define how the resource is to be served up. In certain cases a plug-in may not require such a file, but this key should still be defined and its value should be blank.

5.3.3. Heading ‘Server\_Properties’

Required	Key	Description
Y	ServiceTitle	The WMS title defined in the GetCapabilities. <i>WMS_Capabilities&gt;Service&gt;Title</i>
N	ServiceAbstract	The WMS abstract defined in the GetCapabilities. <i>WMS_Capabilities&gt;Service&gt;Abstract</i>
N	ServiceKeywordCount	The number of service keywords that are expected in this configuration file. If this value is not supplied it is assumed to be 0. When this value is supplied and is non-zero then ServiceKeywordN is expected under this heading where N is the index up to this value starting from 0.  e.g. ServiceKeywordCount=3 ServiceKeyword0=Keyword 1 ServiceKeyword1=Keyword 2 ServiceKeyword2=Keyword 3
N	ServiceKeywordN	The Nth keyword that describes the WMS in the GetCapabilities of the server. <i>WMS_Capabilities&gt;Service&gt;KeywordList&gt;Keyword</i>
N	ServiceFees	A description of the fee for using the WMS. <i>WMS_Capabilities&gt;Service&gt;Fees</i>
N	ServiceAccessConstraints	A description of the access constraints for using the WMS. <i>WMS_Capabilities&gt;Service&gt;AccessConstraints</i>
N	ServiceLayerLimit	The maximum number of layers that can be requested in a single GetMap request. If a user requests more layers than permitted then the request will fail. If this value is not set then there is no limit placed. <i>WMS_Capabilities&gt;Service&gt;LayerLimit</i>
N	ServiceMaxWidth	The maximum width, in pixels, of a GetMap requested. If a user requests a larger width than permitted then the request will fail. If this value is not set then there is no limit placed. <i>WMS_Capabilities&gt;Service&gt;MaxWidth</i>

Required	Key	Description
N	ServiceMaxHeight	The maximum height, in pixels, of a GetMap request. If a user requests a larger height than permitted then the request will fail. If this value is not set then there is no limit placed. <i>WMS_Capabilities&gt;Service&gt;MaxHeight</i>
N	ServiceProvider	The URL of the service provider. <i>WMS_Capabilities&gt;Service&gt;ServiceProvider</i>

### 5.3.4. Heading 'Service\_Contact'

Required	Key	Description
N	ContactPerson	The contact details defined in the GetCapabilities. <i>WMS_Capabilities&gt;Service&gt;ContactInformation&gt;ContactPersonPrimary&gt;ContactPerson</i>
N	ContactOrganisation	The contact details defined in the GetCapabilities. <i>WMS_Capabilities&gt;Service&gt;ContactInformation&gt;ContactPersonPrimary&gt;ContactOrganisation</i>
N	ContactPosition	The contact details defined in the GetCapabilities. <i>WMS_Capabilities&gt;Service&gt;ContactInformation&gt;ContactPosition</i>
N	ContactAddressType	The contact details defined in the GetCapabilities. <i>WMS_Capabilities&gt;Service&gt;ContactInformation&gt;ContactAddress&gt;AddressType</i>
N	ContactAddress	The contact details defined in the GetCapabilities. <i>WMS_Capabilities&gt;Service&gt;ContactInformation&gt;ContactAddress&gt;Address</i>
N	ContactAddressCity	The contact details defined in the GetCapabilities. <i>WMS_Capabilities&gt;Service&gt;ContactInformation&gt;ContactAddress&gt;City</i>
N	ContactAddressStateOrProvince	The contact details defined in the GetCapabilities. <i>WMS_Capabilities&gt;Service&gt;ContactInformation&gt;ContactAddress&gt; StateOrProvince</i>
N	ContactAddressPostalCode	The contact details defined in the GetCapabilities. <i>WMS_Capabilities&gt;Service&gt;ContactInformation&gt;ContactAddress&gt;PostCode</i>

Required	Key	Description
N	ContactAddressCountry	The contact details defined in the GetCapabilities. <i>WMS_Capabilities&gt;Service&gt;ContactInformation&gt;ContactAddress&gt;Country</i>
N	ContactTelephoneNumber	The contact details defined in the GetCapabilities. <i>WMS_Capabilities&gt;Service&gt;ContactInformation&gt;ContactVoiceTelephone</i>
N	ContactFaxNumber	The contact details defined in the GetCapabilities. <i>WMS_Capabilities&gt;Service&gt;ContactInformation&gt;ContactFacsimileTelephone</i>
N	ContactEmailAddress	The contact details defined in the GetCapabilities. <i>WMS_Capabilities&gt;Service&gt;ContactInformation&gt;ContactElectronicMailAddress</i>

5.3.5. Heading 'Service\_Addresses'

Required	Key	Description
Y	GetCapabilitiesAddressesCount	<p>The number of service addresses through which the capabilities of this server can be requested. At the minimum this should be set to 1 for the address of the main server serving this WMS. Usually a value greater than 1 would be to specify either virtual domains of the server or WMS servers that mirror the local configuration.</p> <p>For each index, starting at 0, up to 1 less then this value, the following must be specified (where N is the index)</p> <p>GetCapabilitiesAddressNGet</p> <p>And optionally the following may be specified</p> <p>GetCapabilitiesAddressNPost</p> <p>e.g.</p> <p>GetCapabilitiesAddressesCount=2</p> <p>GetCapabilitiesAddress0Get= http://localhost:8080/wms?</p> <p>GetCapabilitiesAddress1Get= http://anotherhost:80/wms?</p> <p>GetCapabilitiesAddress1Post= http://anotherhost:80/wms?</p>
Y	GetCapabilitiesAddressNGet	<p>The Nth address of this server or mirror server where the server capabilities can be requested via a HTTP Get</p> <p>If this is set to "DYNAMIC", then the value that appears in the capabilities will be set to the address used to access the capabilities.</p> <p><i>WMS_Capabilities&gt;Capability&gt;Request&gt;GetCapabilities&gt; DCPTType&gt;HTTP&gt;Get&gt;OnlineResource</i></p>
N	GetCapabilitiesAddressNPost	<p>The Nth address of this server or mirror server where the server capabilities can be requested via a HTTP Post</p> <p>If this is set to "DYNAMIC", then the value that appears in the capabilities will be set to the address used to access the capabilities.</p> <p><i>WMS_Capabilities&gt;Capability&gt;Request&gt;GetCapabilities&gt; DCPTType&gt;HTTP&gt;Post&gt;OnlineResource</i></p>

Required	Key	Description
Y	GetMapAddressesCount	<p>The number of service addresses through which a GetMap request can be serviced. At the minimum this should be set to 1 for the address of the main server serving this WMS. Usually a value greater than 1 would be to specify either virtual domains of the server or WMS servers that mirror the local configuration.</p> <p>For each index, starting at 0, up to 1 less than this value, the following must be specified (where N is the index)</p> <p>GetMapAddressNGet</p> <p>And optionally the following may be specified</p> <p>GetMapAddressNPost</p> <p>e.g.</p> <p>GetMapAddressesCount=2</p> <p>GetMapAddress0Get= http://localhost:8080/wms?</p> <p>GetMapAddress1Get= http://anotherhost:80/wms?</p> <p>GetMapAddress1Post= http://anotherhost:80/wms?</p>
Y	GetMapAddressNGet	<p>The Nth address of this server or mirror server where a GetMap request can be serviced via a HTTP Get</p> <p>If this is set to "DYNAMIC", then the value that appears in the capabilities will be set to the address used to access the capabilities.</p> <p><i>WMS_Capabilities&gt;Capability&gt;Request&gt;GetMap&gt;DCPType&gt;HTTP&gt;Get&gt;OnlineResource</i></p>
N	GetMapAddressNPost	<p>The Nth address of this server or mirror server where a GetMap request can be serviced via a HTTP Post</p> <p>If this is set to "DYNAMIC", then the value that appears in the capabilities will be set to the address used to access the capabilities.</p> <p><i>WMS_Capabilities&gt;Capability&gt;Request&gt;GetMap&gt;DCPType&gt;HTTP&gt;Post&gt;OnlineResource</i></p>

Required	Key	Description
N	GetFeatureInfoAddressesCount	<p>The number of service addresses through which a GetFeatureInfo request can be serviced. This value can be omitted to denote that GetFeatureInfo is not supported. Usually a value greater than 1 would be to specify either virtual domains of the server or WMS servers that mirror the local configuration.</p> <p>For each index, starting at 0, up to 1 less then this value, the following must be specified (where N is the index)</p> <p>GetFeatureInfoAddressNGet</p> <p>And optionally the following may be specified</p> <p>GetFeatureInfoAddressNPost</p> <p>e.g.</p> <p>GetFeatureInfoAddressCount=2</p> <p>GetFeatureInfoAddress0Get= http://localhost:8080/wms?</p> <p>GetFeatureInfoAddress1Get= http://anotherhost:80/wms?</p> <p>GetFeatureInfoAddress1Post= http://anotherhost:80/wms?</p>
N	GetFeatureInfoAddressNGet	<p>The Nth address of this server or mirror server where a GetFeatureInfo request can be serviced via a HTTP Get</p> <p>If this is set to "DYNAMIC", then the value that appears in the capabilities will be set to the address used to access the capabilities.</p> <p><i>WMS_Capabilities&gt;Capability&gt;Request&gt;GetFeatureInfo &gt; DCPTType&gt;HTTP&gt;Get&gt;OnlineResource</i></p>
N	GetFeatureInfoAddressNPost	<p>The Nth address of this server or mirror server where a GetFeatureInfo request can be serviced via a HTTP Post</p> <p>If this is set to "DYNAMIC", then the value that appears in the capabilities will be set to the address used to access the capabilities.</p> <p><i>WMS_Capabilities&gt;Capability&gt;Request&gt;GetFeatureInfo &gt; DCPTType&gt;HTTP&gt;Post&gt;OnlineResource</i></p>



5.3.6. Heading ‘Root\_Layer\_Details’

Required	Key	Description
Y	RootLayerTitle	The title given to the root layer of the WMS as defined in the GetCapabilities. WMS_Capabilities>Capability>Layer>Title
N	RootLayerCRS	The root coordinate reference system attributed to the root layer of WMS as defined in the GetCapabilities. If this value is set then all data sources must conform to this coordinate system, so it is not usually set. WMS_Capabilities>Capability>Layer>CRS

5.3.7. Heading ‘Response\_Configuration\_Details’

Required	Key	Description
N	DrawingSurfacePoolSize	<p>This numerical value defines the maximum number of MapLink drawing surfaces that are created in a server pool. If none of the plug-ins loaded on the server use a MapLink Drawing Surface then this value will be irrelevant.</p> <p>If a drawing surface is required by a plug-in however, then rather than creating its own one, it should request one from the pool administered by the server. This is to prevent threading issues concerning the use of multiple Drawing Surfaces.</p> <p>The size of the pool must be limited to stop the server running out of resources under heavy load. Higher values permit servicing of more concurrent requests but use more resources; lower values limit the number of requests and should stop the server falling over under heavy load.</p> <p>A maximum pool size of between 10 and 20 is recommended, but for high specification servers it may be possible to increase this value. Experimentation is certainly the best way to find a suitable balance.</p> <p>Please note that the pool is only populated as needed until the maximum pool size is reached. The pool is not allocated on start-up of the WMS.</p> <p>If this value is not specified then a default value of 20 is used.</p>

Required	Key	Description
N	TransparentColourR TransparentColourG TransparentColourB	<p>To enable transparent raster responses to GetMap requests, a global transparent colour is used. This should be defined as a colour that will not appear in the responses of any of the data sources or else that part of the image will also appear transparent.</p> <p>These values represent the 0-255 red, green and blue components of the transparent colour. The default for each colour channel is 0 if they are not specified.</p>
N	UseBGCOLORasTransparentColour	<p>Certain third party WMS viewers do not correctly support image transparency, instead treating a certain colour value as transparent. In most cases, they supply this colour through the BGCOLOR parameter, along with requesting a transparent response.</p> <p>While this is not required by the WMS, it is also not strictly forbidden. This option therefore, allows the BGCOLOR value passed to be used as the transparent colour in responses. If the response format also supports transparency, this colour will be set as the transparent colour in the response image.</p> <p>To enable this feature, set this value to be non-zero. The default is 0, meaning turned off.</p>
N	SupportPNG24 [deprecated]	<p>Certain WMS clients will only handle PNG transparency through the use of an alpha channel, rather than a designated colour value. Performing a GetMap request to a MapLink WMS, using the request format "image/png" and transparency set to true, will use the latter type of PNG transparency.</p> <p>By adding this setting and using the value "1", an additional image request format will be offered called "image/png24" which will return a PNG using the alpha-channel for transparency.</p> <p>By default this additional format is not offered or advertised.</p>

Required	Key	Description
N	SupportedFormat[1..n] SupportedMimeType[1..n] SupportedV100Format[1..n]	<p>Each supported map format should include a set of attributes, each attribute having the same index suffix. A supported format must have a SupportedFormat attribute. SupportedMimeType and SupportedV100Format are optional.</p> <p>If these attributes exist, SupportPNG24 will not be checked. If you would set SupportPNG24, provide a SupportedFormat configuration for PNG24.</p> <p>If these attributes are not provided, the WMS will revert to the legacy built-in supported formats (GIF, PNG, TIFF, JPEG, [if SupportPNG24=1] PNG24).</p> <p>The configuration should look something like this:</p> <pre>SupportedFormat1=image/png SupportedMimeType1=image/png SupportedV100Format1=PNG SupportedFormat2=image/tiff SupportedMimeType2=image/tiff SupportedV100Format2=TIFF ... SupportedFormatn=image/jpeg SupportedMimeTypen=image/jpeg SupportedV100Formatn=JPEG</pre>
N	EnableAntiAliasedFonts	<p>This Boolean value specifies whether anti-aliasing is enabled for the server when rendering TrueType fonts.</p> <p>This setting only applies when running the server on Windows (using the GDI drawing surface).</p> <p>If this setting is not provided text anti-aliasing will be enabled.</p>

5.3.8. Heading ‘MapLink\_Standard\_Configuration’

Required	Key	Description
N	Std_Config_Path	<p>If supplied, this is the path from which the standard MapLink configuration files are to be loaded. If this value is omitted, then the files are assumed to be present in the \config subdirectory of the MapLink installation on the local machine (whose location is found using <code>TSLUtilityFunctions::getMapLinkHome()</code>)</p> <p>All instances of the MapLink WMS within a single process must share the same path. The first instance that is loaded will have this value examined and make the equivalent MapLink SDK call:</p> <p><code>TSLDrawingSurface::loadStandardConfig</code></p> <p>Further instances loaded into the same process will have this value ignored.</p>
N	Colour_List_Location	<p>MapLink Maps loaded into any of the pre-built plug-ins will have their palette file suppressed to ensure thread safety. If any of the maps loaded in a MapLink WMS extend the standard palette, then these entities will not appear correctly unless this value is used.</p> <p>Providing this value will use the palette file found at this location for the loading of all maps and will be set on all MapLink Drawing Surfaces.</p> <p>Omitting this value will mean that all maps loaded and all drawing surfaces will use the standard palette file loaded either from the \config directory of the MapLink installation on the local machine or via providing the <code>Std_Config_Path</code> value.</p> <p>All instances of the MapLink WMS within a single process must share the same path. The first instance that is loaded will have this value examined and make the equivalent MapLink SDK call:</p> <p><code>TSLDrawingSurface::setupColours</code></p> <p>Further instances loaded into the same process will have this value ignored.</p>
N	JPEG_Compression_Factor	<p>Allows control over the compression factor used when a client requests a JPEG image from the MapLink WMS. Valid values are in the range 0,2-255. 0 equates to lossless JPEG, while the values 2-255 give increasingly lossy compression.</p> <p>The default compression factor is 2.</p>

### 5.3.9. Example

```
[Server_Data_Sources_Index]
DatasourceCount=2
Datasource0Location=C:\Maps\BasicMap\BasicMap1.map
Datasource0Plugin=basicmapplugin
Datasource0Configuration=C:\Maps\BasicMap\BasicMap1Config.xml
Datasource1Location=C:\Maps\HistoricalMap\HistoricalMap1.map
Datasource1Plugin=historicalmapplugin
Datasource1Configuration=C:\Maps\HistoricalMap\HistoricalMap1Config.xml
```

```
[Server_Properties]
ServiceTitle=My WMS Service
ServiceAbstract=My abstract
ServiceKeywordCount=1
ServiceKeyword0=My Keyword
ServiceFees=None
ServiceAccessConstraints=None
ServiceLayerLimit=16
ServiceMaxWidth=4000
ServiceMaxHeight=4000
ServiceProvider=http://www.envitia.com
```

```
[Service_Contact]
ContactPerson=Mr. A Person
ContactOrganisation=Envitia
ContactPosition=Engineer
ContactAddressType=Postal
ContactAddress=North Heath Lane
ContactAddressCity=Horsham
ContactAddressStateOrProvince=West Sussex
ContactAddressPostalCode=RH12 5UX
ContactAddressCountry=England
ContactTelephoneNumber=+441403 273 173
ContactFaxNumber=+441403 273 173
ContactEmailAddress=support@envitia.com
```

```
[Service_Addresses]
GetCapabilitiesAddressesCount=1
GetCapabilitiesAddress0Get=DYNAMIC
GetMapAddressesCount=1
```

```
GetMapAddress0Get=DYNAMIC
GetFeatureInfoAddressesCount=1
GetFeatureInfoAddress0Get=DYNAMIC
```

```
[Root_Layer_Details]
RootLayerTitle=Envitia WMS Layers
RootLayerCRS=EPSG:27700
```

```
[Response_Configuration_Details]
DrawingSurfacePoolSize=15
TransparentColourR=255
TransparentColourG=0
TransparentColourB=0
UseBGCOLORasTransparentColour=0
SupportPNG24=1
```

```
[MapLink_Standard_Configuration]
Std_Config_Path=c:\program files\envitia\maplink pro\8.1\config
Colour_List_Location=c:\maps\wms.pal
```

## 5.4. Supplied MapLink WMS Plug-Ins

### 5.4.1. Introduction

Envitia supplies some pre built WMS plug-ins that facilitate serving common spatial data. This section will outline those plug-ins, what they're for and how to configure them.

Note: The plugins are in the `plugins` directory of the `bin64` folder.

### 5.4.2. The Basic Map Plug-In

This plug-in is largely deprecated as using the Super Map Plug-In offers far better performance and reduced memory foot print.

The basic map plug-in is used to serve standard MapLink maps, built using MapLink Studio, in a customisable manner. In this case the spatial data would be the fully qualified path to the MapLink map, the plug-in would be the `'basicmapplugin'` (`BasicMapWMS_plugin` on Linux) and the configuration file would be a fully qualified path to an xml file.

The XML configuration file can be created using the `BMCCreator` utility supplied with MapLink. This utility allows different map features to be associated with WMS layers as well as configuring all the standard WMS layer attributes.

To create a configuration manually, the schema for the configuration file can be used as a reference. It can be access from the Envitia website at the following URL:

<http://www.envitia.com/schemas/maplinkwms/basicmapplugin/1.0/BasicMapConfiguration.xsd>

The Basic Map Plug-In services requests using a pool of MapLink Map Data Layers, assigning one temporarily to each request to allow a draw to occur before returning it the

pool. Each Map Data Layer has a cache of the map files that it most recently accessed, which by default is limited to 32 Megabytes, while the pool of Map Data Layers by default contains 20 layers. The total amount of memory that deployment of the Basic Map Plug-In requires therefore, by default, is at least 640 Megabytes, but in fact it usually uses approximately 50% more than this in practice. These defaults can be configured through the `BMCCreator` however. It is because of this high memory requirement that the Basic Map Plug-In was replaced by the Super Map Plug-In.

### 5.4.3. The Historical Map Plug-In

This plug-in is largely deprecated as using the Super Map Plug-In offers far better performance and reduced memory footprint. Configuring the Super Map Plug-In to use historical data can be difficult however, so for expediency the Historical Map Plug-In is often used.

The historical map plug-in is almost identical to the basic map plug-in explained in the previous section except it is intended to serve MapLink Maps with historical information built using the Seamless Layer Manager. The plug-in requires the archive directory of the seamless layer map to be located in the same directory as the `.map` file.

As with the basic map plug-in, the spatial data of the plug-in is the absolute path to the map and the configuration file is an xml file created using the `BMCCreator` utility. The plug-in string for the historical map plug-in is `'historicalmapplugin'` however.

The historical map plug-in adds a time dimension to the capabilities of the basic map plug-in so that WMS users can rollback the map to previous versions. Although the historical map plug-in can be used to serve a non historical MapLink map, the super map plug-in is better optimised for speedier responses from these maps.

### 5.4.4. The Super Map Plug-In

The Super Map Plug-In is a new addition to MapLink that replaces the existing Basic and Historical Map Plug-Ins, allowing both standard and historical MapLink maps to be served. It offers the best performance and lowest memory footprint of any of the Envitia supplied WMS plug-ins, through the use of the new MapLink Threaded Map Cache SDK.

The Basic and Historical Map Plug-Ins service requests using a pool of MapLink Map Data Layers, assigning one temporarily to each request to allow a draw to occur before returning it the pool. This is because standard MapLink Map Data Layers cannot be shared amongst threads due to thread safety issues. The Threaded Map Cache offers a variant of the Map Data Layer that allows a loaded map to be shared amongst threads. Additionally, each standard Map Data Layer has a cache of the map files that it most recently accessed, whereas the Threaded Map Cache shares a single, lock-free but thread safe, cache amongst all requests. This greatly reduces the memory footprint whilst improving performance thanks to a greater likelihood of locating the required map file in the memory cache, rather than having to load it from disk.

Unlike the Basic and Historical Map plug-ins, the Super Map plug-in supports `GetFeatureInfo` requests for vector layers in MapLink maps. By default this returns an XML document containing information on the selected feature(s), the schema of which is available at the following location:

<http://www.envitia.com/schemas/maplinkwms/supermapplugin/getfeatureinfo/1.0/GetFeatureInfo.xsd>

The following example shows the default output format of `GetFeatureInfo` requests:

```
<FeatureCollection
xmlns="http://www.envitia.com/schemas/maplinkwms/supermapplugin/getfeatureinfo"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xlink="http://www.w3.org/1999/xlink"

xsi:schemaLocation="http://www.envitia.com/schemas/maplinkwms/supermapplugin/ge
tfeatureinfo
http://www.envitia.com/schemas/maplinkwms/supermapplugin/getfeatureinfo/1.0/Get
FeatureInfo.xsd">
  <Polygon sourceLayer="UKFull" id="1000015012413"
    featureType="MasterMap.Building: area">
    <Broken>False</Broken>
    <Theme>Buildings</Theme>
    <Calculated_Area_Value>62.920004</Calculated_Area_Value>
    <Descriptive_Group>Building</Descriptive_Group>
    <Make>Manmade</Make>
    <Physical_Level>50</Physical_Level>
  </Polygon>
</FeatureCollection>
```

In addition, the Super Map plug-in allows extra response formats to be generated through user-supplied XSL transforms that process the default XML document into the desired format.

The Super Map plug-in has two modes of configuration, henceforth referred to as 'single map mode' and 'multi-map mode'. The configuration mode used determines how the MapLink map or maps used as the data sources are served from the WMS. In both cases its plug-in string would be 'supermapplugin' on Windows and 'SuperMapWMS\_plugin' on all other platforms.

#### 5.4.4.1. Single Map Mode

In single map mode the Super Map plug-in operates similarly to the Basic Map plug-in. As the name suggests, this configuration mode should be used to serve a single MapLink map from a data source. Like the basic and historical map plug-ins, the super map plug-in in single map mode takes the fully qualified path to the MapLink map as its Spatial Data parameter and the configuration file would be a fully qualified path to an xml file. It accepts the same format of XML configuration file that is produced by the `BMCCreator` utility, with the following differences:

- The "Number of Map Data Layers" setting from the `BMCCreator`'s Options menu should be set to a much higher level than would be used for the Basic or Historical Map Plug-Ins. Ideally it should match the drawing surface pool size, configured in the service's configuration file, as described in section 5.3.7.
- The "Cache Size Per Data Layer (KB)" setting from the `BMCCreator`'s Options menu has a different meaning. Rather than refer the cache size per pooled data layers, as for the Basic or Historical Map Plug-Ins, it instead refers to the shared cache's size. This should be set fairly high, if possible, preferably in the hundreds of Megabytes range.

The Super Map plug-in also has an extended configuration format for single map mode, the format of which is described by the schema at the following location:

<http://www.envitia.com/schemas/maplinkwms/supermapplugin/1.0/SuperMapConfiguration.xsd>

This format is very similar to that output from the `BMCCreator` utility, but allows configuration of two additional pieces of functionality only offered by the Super Map plug-in. The first of these pieces of functionality is the ability to use MapLink dynamic renderers to implement



specific named WMS styles. These dynamic renders should be built as separate DLLs/shared objects and register themselves with the MapLink `TSLDynamicRendererFactory` on DLL/shared object load. The `dynamicRendererStore` attribute on the `SuperMapConfiguration` element should then be set to the location of the dynamic renderer(s). The name used to register the dynamic renderer with the factory determines the name that the style is advertised as in the server's capabilities document.

The second piece of functionality allows for the user-defined `GetFeatureInfo` formats mentioned in section 5.4.4 to be specified through the optional `GetFeatureInfoResponseList` element. Within this element a list of `ResponseFormat` elements can be provided, each of which defines an additional response format to be advertised by the server. The `transform`, `advertisedFormat` and `mimeType` attributes must be provided for each response format and define the location of the XSL transform that should be run on the normal XML output document before being returned to the client, the value for the Format string that will be listed in the server's `GetFeatureInfo` format list and the MIME type that will be used for the responses respectively. The optional `WMSCapabilities1_0_0Format` attribute is only used when clients issue requests using version 1.0.0 of the WMS standard. This version restricts advertised `GetFeatureInfo` formats to the second part of the MIME type and thus cannot use the same configuration setting. If this attribute is not specified, the response format will not be advertised to clients using version 1.0.0 of the WMS standard.

The following example configuration file shows the extended single map mode configuration format:

```
<?xml version="1.0" encoding="UTF-8"?>
<SuperMapConfiguration
xmlns="http://www.envitia.com/schemas/maplinkwms/supermapplugin"
  xmlns:wms="http://www.opengis.net/wms"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xsi:schemaLocation="http://www.envitia.com/schemas/maplinkwms/supermapplugin
http://www.envitia.com/schemas/maplinkwms/supermapplugin/1.0/SuperMapConfigurat
ion.xsd"
mapDataLayerCount="50" mapDataLayerCacheSize="262144"
symbolTextViewExpansion="15"
dynamicRendererStore="/path/to/dynamicrenderers/folder">
  <wms:Name>UKFull</wms:Name>
  <wms:Title>UKFull</wms:Title>
  <wms:EX_GeographicBoundingBox>
    <wms:westBoundLongitude>-7.557160</wms:westBoundLongitude>
    <wms:eastBoundLongitude>3.632021</wms:eastBoundLongitude>
    <wms:southBoundLatitude>49.766807</wms:southBoundLatitude>
    <wms:northBoundLatitude>61.464590</wms:northBoundLatitude>
  </wms:EX_GeographicBoundingBox>
  <wms:BoundingBox CRS="EPSG:27700" minx="705744.843000" miny="1314615.533000"
    maxx="7162.949000" maxy="2916.550000"/>
  <ImplementedFeaturesList>
    <ImplementedFeature>Rasters</ImplementedFeature>
    <ImplementedFeature>MasterMap</ImplementedFeature>
  </ImplementedFeaturesList>
  <BasicMapLayer>
    <wms:Name>Raster</wms:Name>
    <wms:Title>Raster</wms:Title>
    <wms:BoundingBox CRS="EPSG:27700" minx="705269.607000"
      miny="1300000.000000" maxx="-2083.332000"
      maxy="3186.274000"/>
    <ImplementedFeaturesList>
      <ImplementedFeature>Rasters</ImplementedFeature>
    </ImplementedFeaturesList>
  </BasicMapLayer>
```

```

<BasicMapLayer queryable="1">
  <wms:Name>Vector</wms:Name>
  <wms:Title>Vector</wms:Title>
  <wms:BoundingBox CRS="EPSG:27700" minx="702083.332000"
    miny="1296813.725000" maxx="-2083.332000"
    maxy="6372.549000"/>
  <ImplementedFeaturesList>
    <ImplementedFeature>MasterMap</ImplementedFeature>
  </ImplementedFeaturesList>
</BasicMapLayer>
<GetFeatureInfoResponseList>
  <ResponseFormat transform="/path/to/xml_html.xsl"
    advertisedFormat="text/html" mimeType="text/html"
    WMSCapabilities1_0_0Format="HTML"/>
</GetFeatureInfoResponseList>
</SuperMapConfiguration>

```

#### 5.4.4.2. Multi-Map Mode

In multi-map mode the Super Map plug-in operates similarly to the Historical Map plug-in but offers additional functionality. In addition to serving a single MapLink map containing historical information, the Super map plug-in in this mode is capable of taking multiple separate MapLink maps that may or may not contain history and present them as a single set of layers with the combined history of all the maps.

In this mode both the Spatial Data parameter and the configuration file would be fully qualified paths to xml files. The format of the Spatial Data configuration file is described by the schema at the following location:

<http://www.envitia.com/schemas/maplinkwms/supermapplugin/datasource/1.0/SuperMapDataSource.xsd>

The element type used to contain all of the source layers defines how the MapLink maps will be advertised by the WMS. Currently the only supported element type is `HistoricalMap`, which indicates all data source child elements should be advertised as a single set of unified layers. It is an error to list more than one `HistoricalMap` element within the same Spatial Data configuration file.

Within this element is a list of the MapLink maps to use as data sources. If a map has historical information it should be listed using the `HistoricalMapMultiVersion` element, with the `archiveDirectory` attribute set to the location of the archive directory of the map. Unlike the Historical Map plug-in, this directory does not have to be in the same folder as the `.map` file. If the map does not have historical information it should be listed using the `HistoricalMapVersion` element with the `timestamp` attribute set to the time that the MapLink map represents.

Regardless of which element is used to identify the MapLink map location, the `id` attribute should be set to a unique value. This identifier is used to match the data sources listed within the Spatial Data configuration file to the WMS layer configurations within the data source's configuration file.

The following example demonstrates a Spatial Data configuration file that includes MapLink maps with and without historical information:

```

<?xml version="1.0" encoding="UTF-8"?>
<SuperMapDataSource
  xmlns="http://www.envitia.com/schemas/maplinkwms/supermapplugin/datasource"
  xmlns:wms="http://www.opengis.net/wms"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xsi:schemaLocation="http://www.envitia.com/schemas/maplinkwms/supermapplugin/datasource

```

```

http://www.envitia.com/schemas/maplinkwms/supermapplugin/datasource/1.0/SuperMapDataSource.xsd">
  <HistoricalMap>
    <HistoricalMapMultiVersion sourceMap="/path/to/historical_map_1/map.map"
      archiveDirectory="/path/to/historical_map_1/archive" id="1"/>
    <HistoricalMapMultiVersion sourceMap="/path/to/historical_map_2/map.map"
      archiveDirectory="/path/to/historical_map_2/archive" id="2"/>
    <HistoricalMapVersion sourceMap="/path/to/normal_map/map0.map"
      timestamp="2010-11-25T11:45:09Z" id="3"/>
  </HistoricalMap>
</SuperMapDataSource>

```

In multi-map mode the WMS data source configuration file uses the `SuperMultiMapConfiguration` element as its root node. This element has a list of `SuperMapConfiguration` child elements that define the WMS layer configurations for each of the data sources defined within the Spatial Data configuration file. The contents of each `SuperMapConfiguration` element is the same as if the data source was being used in single map mode, with the following exceptions:

- The additional `id` attribute must be set to the same value as used in the attribute of the same name for the data source.
- The `mapDataLayerCount`, `mapDataLayerCacheSize`, `symbolTextViewExpansion` and `dynamicRendererStore` attributes are no longer valid at this level. These should be specified on the `SuperMultiMapConfiguration` element.
- Any additional `GetFeatureInfo` response formats defined in the `GetFeatureInfoResponseList` at this level are ignored. These should be listed as a child of the `SuperMultiMapConfiguration` element.

Additionally, the following restrictions apply when using multiple MapLink maps within a single WMS data source through the `HistoricalMap` Spatial Data configuration element:

- Each `SuperMapConfiguration` object must define the same set of WMS layers. Any layers that are not present in all `SuperMapConfiguration` objects will not be advertised from the WMS. Note that these layers do not have to have the same `ImplementedFeature` list.
- The bounding boxes of a WMS layer should be consistent across all definitions of that layer.
- The value of the `queryable` attribute of a WMS layer should be consistent across all definitions of that layer.

The following example demonstrates the corresponding configuration for the above Spatial Data example configuration:

```

<?xml version="1.0" encoding="UTF-8"?>
<SuperMultiMapConfiguration
xmlns="http://www.envitia.com/schemas/maplinkwms/supermapplugin"
  xmlns:wms="http://www.opengis.net/wms"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xsi:schemaLocation="http://www.envitia.com/schemas/maplinkwms/supermapplugin
http://www.envitia.com/schemas/maplinkwms/supermapplugin/1.0/SuperMapConfiguration.xsd"
mapDataLayerCount="50" mapDataLayerCacheSize="262144"
symbolTextViewExpansion="15"
dynamicRendererStore="/path/to/dynamicrenderers/folder">
  <SuperMapConfiguration id="1" queryable="1">
    <wms:Name>IsleOfWight</wms:Name>

```

```

<wms:Title>IsleOfWight</wms:Title>
<wms:EX_GeographicBoundingBox>
  <wms:westBoundLongitude>-7.557160</wms:westBoundLongitude>
  <wms:eastBoundLongitude>3.632021</wms:eastBoundLongitude>
  <wms:southBoundLatitude>49.766807</wms:southBoundLatitude>
  <wms:northBoundLatitude>61.464590</wms:northBoundLatitude>
</wms:EX_GeographicBoundingBox>
<wms:BoundingBox CRS="EPSG:27700" minx="498161.764000" miny="121078.431000"
  maxx="408946.078000" maxy="57352.941000"/>
<ImplementedFeaturesList>
  <ImplementedFeature>Rasters</ImplementedFeature>
  <ImplementedFeature>MasterMap</ImplementedFeature>
</ImplementedFeaturesList>
</SuperMapConfiguration>
<SuperMapConfiguration id="2" queryable="1">
  <wms:Name>IsleOfWight</wms:Name>
  <wms:Title>IsleOfWight</wms:Title>
  <wms:EX_GeographicBoundingBox>
    <wms:westBoundLongitude>-7.557160</wms:westBoundLongitude>
    <wms:eastBoundLongitude>3.632021</wms:eastBoundLongitude>
    <wms:southBoundLatitude>49.766807</wms:southBoundLatitude>
    <wms:northBoundLatitude>61.464590</wms:northBoundLatitude>
  </wms:EX_GeographicBoundingBox>
  <wms:BoundingBox CRS="EPSG:27700" minx="498161.764000" miny="121078.431000"
    maxx="408946.078000" maxy="57352.941000"/>
  <ImplementedFeaturesList>
    <ImplementedFeature>Rasters</ImplementedFeature>
    <ImplementedFeature>MasterMap</ImplementedFeature>
  </ImplementedFeaturesList>
</SuperMapConfiguration>
<SuperMapConfiguration id="3" queryable="1">
  <wms:Name>IsleOfWight</wms:Name>
  <wms:Title>IsleOfWight</wms:Title>
  <wms:EX_GeographicBoundingBox>
    <wms:westBoundLongitude>-7.557160</wms:westBoundLongitude>
    <wms:eastBoundLongitude>3.632021</wms:eastBoundLongitude>
    <wms:southBoundLatitude>49.766807</wms:southBoundLatitude>
    <wms:northBoundLatitude>61.464590</wms:northBoundLatitude>
  </wms:EX_GeographicBoundingBox>
  <wms:BoundingBox CRS="EPSG:27700" minx="498161.764000" miny="121078.431000"
    maxx="408946.078000" maxy="57352.941000"/>
  <ImplementedFeaturesList>
    <ImplementedFeature>Rasters</ImplementedFeature>
    <ImplementedFeature>Vector</ImplementedFeature>
  </ImplementedFeaturesList>
</SuperMapConfiguration>
<GetFeatureInfoResponseList>
  <ResponseFormat transform="/path/to/xml_html.xsl"
    advertisedFormat="text/html" mimeType="text/html"
    WMSCapabilities1_0_0Format="HTML"/>
</GetFeatureInfoResponseList>
</SuperMultiMapConfiguration>

```

#### 5.4.5. The CADRG Map Plug-In

Please see the document 'MapLink CADRG WMS Plug-In User Guide.'

## 5.5. Common Problems

- When making requests an exception is returned. This normally indicates that a configuration error has been made or a `GetMap` request string is invalid. It is recommended that the user reads the content of the `ServiceException` element to amend the problem.
- After deployment, a service exception is shown when using the service stating that the WMS cannot find or access the MapLink WMS configuration file. This is usually caused by either the configuration file not being where the WMS is configured to expect it or that it doesn't have sufficient permission to access it.

IIS based deployments most often suffer from the latter issue as the ISS worker process runs in a reduced privileged user account. Granting the read privilege on the file to the "NETWORK SERVICE" user account is usually the best way to rectify this issue.

Please remember when deploying to non-Windows platforms that the case of the configuration file path is important.

- After deployment, a service exception is shown when using the service stating that the WMS is not licensed, yet the appropriate licence has been installed correctly. This can sometimes be caused by another Envitia process, usually the Licence Key Administrator, locking out the WMS from check whether a licence is installed. Shut down other Envitia applications, restart the web server and try again.
- After deployment, a service exception is shown when using the service stating that it failed to load the standard MapLink configuration. This is usually caused by "Std\_Config\_Path" setting, documented in section 5.3.8, having been incorrectly configured.
- After deployment, a service exception is shown when using the service stating that it failed to load a plug-in library. This may be caused by one of the following:
  - The plug-in was not found by the runtime. Ensure that it is accessible to the runtime either via the working path or PATH environment variable on Windows.  
On non-windows platforms, ensure that the case of the plug-in name contained in the configuration file matches that of the filename.
  - On Windows only, the filename of the DLL was not suffixed correctly to match the WMS service. For instance if running in 64-bit, the name of the plug-in that appears in the configuration file will be appended with "64.dll". 64-bit debug will append "64d.dll".
  - The runtime does not have the required permissions to access the library. IIS based deployments most often suffer from this issue as the ISS worker process runs in a reduced privileged user account. Granting the read privilege on the library and any dependencies to the "NETWORK SERVICE" user account is to rectify this issue.
- The deployment of the service was successful and the Capabilities can be retrieved, but when attempting to use the service in a WMS client nothing is displayed.

The most common cause of this error is that the `GetMap` address that appears in the service Capabilities does not match the address that the service is deployed on. Refer to section 5.3.5 for details of how to configure the address that appear.

When using one of the Envitia supplied service plug-ins, this issue may be caused by the geographic area configured to be advertised for a data source not containing any data.

- The service works when accessed from the server it is deployed upon, but not from another machine on the same network.

This issue is mainly beyond the scope of this document, but two common causes are that the server's firewall is blocking access to the web server's port and the second is that a loopback address has been used when configuring the Capabilities of the server (E.G. "localhost" or "127.0.0.1") when completing section 5.3.5 of the service's configuration file.

- When using a `TSLWMSDataLayer` with any WMS data source a situation can occur where tiles are constantly loaded and unloaded which causes a flickering effect. This issue is not at the server end, but arises because the layer's tile cache does not have enough memory to store all of the requested tiles. To resolve this issue the user should increase the size of the tile cache using the `cacheSize()` method on the data layer.

## 5.6. Docker

### 5.6.1. Prerequisites:

Make sure you have Docker 1.12+ installed on your Linux box.

### 5.6.2. Building The WMS Docker Image

Build the WMS image using the following command:

```
docker build -t maplink-wms:10.2.8.0 -f redist64/docker/wms/Dockerfile .
```

### 5.6.3. Running The WMS Container

Run the container using the following command:

```
docker run -p 8022:8080 -v /home/user/wms/maps:/opt/wms/maps --name  
maplink-wms -d maplink-wms:10.2.8.0
```

Where:

- /home/user/wms/maps is the root of your maps folder.
- 8022 is the port where the container gets mapped on your Docker host.

To check that the WMS is running correctly, visit the following URL:

<http://host:8022/MapLinkOGCServices/OGC?SERVICE=WMS&Request=GetCapabilities>

Replace the **host** with your linux host name/ip.

### 5.6.4. Stopping The Container

```
docker stop maplink-wms  
docker rm maplink-wms
```

NOTE: the container has to be removed before starting it again.

### 5.6.5. Customising Your WMS Configuration

Create a folder on your host for your base configuration.

E.g. `mkdir -p /home/user/wms/baseconfig`

Copy the example base config from redist64/docker/wms/baseconfig.ini to the baseconfig folder.

Edit the configuration file, as outlined in 5.3.

Run the container using:

```
docker run -p 8022:8080 -v /home/user/wms/maps:/opt/wms/maps -v  
/home/user/wms/baseconfig:/opt/wms/baseconfig --name maplink-wms -d  
maplink-wms:10.2.8.0
```

Where:

/home/user/wms/maps - Is the root of your maps folder.

/home/user/wms/baseconfig - Contains the base configuration file called baseconfig.ini

### 5.6.6. Mount Points

<code>/opt/wms/maps</code>	You must mount this folder as this is the location where the container will search for maps
<code>/opt/wms/config</code>	If you mount this folder you will see the generated WMS configuration file
<code>/opt/wms/baseconfig</code>	This folder stores the base configuration file

### 5.6.7. Generating WMS Config XMLs

Ensure that you have `xsltproc` installed. E.G.:

```
apt-get install xsltproc
```

Make sure that the MapLink environment variables are set and the bin folder is added to the path.

```
source mapl_init.bash
export PATH=$MAPL_HOME/bin/x86_64:$PATH
```

Use the following scripts to generate XMLs for the map files you wish to add to your WMS as data sources.

Once generated, copy the map files and the corresponding XML files into your `"/home/user/wms/maps"` folder.

#### 5.6.7.1. Generating WMS Configuration For A Single Map

Use the following command:

```
genwmsconf.sh ./maps/NaturalEarthBasic/NaturalEarthBasic.map
```

#### 5.6.7.2. Generating WMS Configuration XMLs For A Directory of Maps

Navigate to the directory of maps that you want to generate config files for and execute the following command:

```
genwmsdir.sh ./maps/
```



## 6. THE MAPLINK WPS SERVICE

### 6.1. Web Processing Service Introduction

The Web Processing Service (WPS) can provide a set of "processes" that receive zero or more inputs and return one or more outputs. The WPS standard describes a process as "any algorithm, calculation or model that operates on spatially referenced data," although its interface is not limited to geospatial operations.

The MapLink WPS Plug-In SDK can be used to create user plug-ins. For more information on this SDK please consult the MapLink Developer's Guide.

### 6.2. The Structure of the MapLink WPS

#### 6.2.1. Introduction

The WPS itself supports version 1.0.0 of the standard, currently the only published version. It is intended that when further versions are released, the MapLink WPS will be extended to support them.

Each WPS plugin can provide one or more WPS processes

On start-up the WPS service will read its configuration file and determine what WPS plugins it needs to load.

### 6.3. Configuring a MapLink WPS

#### 6.3.1. MapLink WPS Configuration File

The MapLink WPS configuration file defines what data sources ('processes') it should serve up and how. This configuration file also defines much of the service Capabilities as well as certain MapLink specific settings.

The format of the configuration file is that of an XML file the schema of which is available at the following URL:

<http://www.envitia.com/schemas/maplinkwps/1.0/MaplinkWPS.xsd>

#### 6.3.2. The DataSources element

In the WPS Service configuration file is a 'DataSources' xml element which contains a number of 'DataSource' xml elements.

Each 'DataSource' xml element:

- Represents a WPS plugin.
- Has a 'Plugin' xml element - this defines the DLL name of the data source plugin to load. This is the name of the DLL, without the configuration-specific suffix.
- Has a 'DataPath' xml element - this is an extra parameter that can be passed into the Data Source plugin at start up
- Has a 'ConfigPath' xml element - this is an extra parameter that can be passed into the Data Source plugin at start up, this parameter tends to be used to define the configuration file

### 6.3.3. The DataStore element

This optional element defines the data store of the service. This allows for WPS requests to ask the WPS service to store the result, so it can be picked up later.

Note: it is optional for a WPS plugin to support data store usage. This is defined in the implementation of the WPS plugin itself.

The DataStore element is defined in the schema as abstract with currently only one derivate supported; 'FileStore'. This type of store will use a file system directory to store asynchronous responses and referenced outputs, which is provided via the "Directory" attribute. The File Store can optionally be configured with a purge strategy which indicates how older items should be removed. If a purge strategy is not configured, then the directory can grow large over time and will need to be manually purged.

### 6.3.4. The Options element

The Options element is used to configure various MapLink settings, although currently there is only one offered by the MapLink WPS. It is expected that the configurable options will be expanded upon in patches and future releases.

Each Option sub-element must be provided with the "name" attribute to indicate what is being configured and therefore the type of content to be expected inside the element. The "name" attribute is an XML union of an enumeration and the string type. The enumeration defines the current build-in names, with "StandardConfigPath" with being the only current value to configure the location of the MapLink configuration directory, whilst the string allows future additions and undocumented values.

### 6.3.5. The DefaultLanguage and LanguageSpecificMetadata elements

The WPS standard is one of the first OGC standards to offer the service Capabilities in more than one language, specifically the parts that are for consumption by a user rather than the client software. The MapLink WPS offers this functionality in the service capabilities and other service responses.

A LanguageSpecificMetadata element is used to define the parts of the Capabilities document for a specific language, including the ServiceIdentification, ServiceProvider, OperationsMetadata and WSDL elements. When the Capabilities of the service are requested for that language, what appears in the LanguageSpecificMetadata will be included in exactly the same way as it appears in the configuration file. To support multiple languages, simply include a LanguageSpecificMetadata element for each of them.

The service must also be told which of the supported languages it should treat as the default, for when a request does not specify which language it would like the response in. The language type that appears in the DefaultLanguage element must reference one of the languages used in an included LanguageSpecificMetadata element.

#### 6.3.5.1. The OperationsMetadata element

The OperationsMetadata element is used to describe the Operations that the service supports. This is of importance to the MapLink WMS as it also describes the addresses that these services are available at. These addresses can either be full, qualified, or, like offered in the MapLink WMS, use the DYNAMIC keyword. The DYNAMIC keyword will be replaced in the Capabilities returned during a "GetCapabilities" request with the address that the request was made to.

### 6.3.6. Example

The following example demonstrates a configuration file which has the following settings:

- It deploys two data sources; one that uses "MyPlugin" and the other uses "MyOtherPlugin". Both take a TXT file for their data and an INI file for their configuration.
- It is configured to offer a file based data store, that uses c:\temp\WPSStore to store asynchronous and referenced resources, and perform a purge job everyday which will remove items which were created at least 2 days ago.
- A single option is defined, which specifies MapLink's standard configuration path.
- "en-GB", which is the abbreviation of British English, is defined as the default language.
- LanguageSpecificMetadata is included for the default language, "en-GB", and for "en-US" is the abbreviation of US English.
- Both defined LanguageSpecificMetadata elements include the same Service Identification and Service Provider sub elements.
- Both defined LanguageSpecificMetadata elements include OperationsMetadata sub element which describe the service's end-points using the DYNAMIC keyword.

```
<?xml version="1.0" encoding="UTF-8"?>
<mwps:WPSConfiguration
  xmlns:mwps="http://www.envitia.com/schemas/maplinkwps"
  xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xsi:schemaLocation="http://www.envitia.com/schemas/maplinkwps
    http://www.envitia.com/schemas/maplinkwps/1.0/MaplinkWPS.xsd">
  <mwps:DataSources>
    <mwps:DataSource>
      <mwps:Plugin>MyPlugin</mwps:Plugin>
      <mwps:DataPath>c:\mydata\data.txt</mwps:DataPath>
      <mwps:ConfigPath>C:\myconfiguration\config.ini</mwps:ConfigPath>
    </mwps:DataSource>
    <mwps:DataSource>
      <mwps:Plugin>MyOtherPlugin</mwps:Plugin>
      <mwps:DataPath>c:\mydata\data2.txt</mwps:DataPath>
      <mwps:ConfigPath>C:\myconfiguration\config2.ini</mwps:ConfigPath>
    </mwps:DataSource>
  </mwps:DataSources>
  <mwps:DataStore xsi:type="mwps:FileStore" Directory="C:\Temp\WPSStore">
    <mwps:PurgingStrategy>
      <mwps:FrequencyOfOperation>P1D</mwps:FrequencyOfOperation>
      <mwps:PurgeAge>P2D</mwps:PurgeAge>
      <mwps:AgeCalculationMethod>FromCreation</mwps:AgeCalculationMethod>
    </mwps:PurgingStrategy>
  </mwps:DataStore>
  <mwps:Options>
    <mwps:Option name="StandardConfigPath">c:\Program Files\Envitia\MapLink
    Pro\8.1\config</mwps:Option>
  </mwps:Options>
  <mwps:DefaultLanguage>en-GB</mwps:DefaultLanguage>
  <mwps:LanguageSpecificMetadata xml:lang="en-GB">
    <ows:ServiceIdentification>
      <ows:Title>Envitia MapLink WPS Server</ows:Title>
      <ows:Abstract>Envitia MapLink WPS Server</ows:Abstract>
      <ows:Keywords>
        <ows:Keyword>Envitia</ows:Keyword>
        <ows:Keyword>MapLink</ows:Keyword>
      </ows:Keywords>
    </ows:ServiceIdentification>
  </mwps:LanguageSpecificMetadata>
</mwps:WPSConfiguration>
```

```

    <ows:ServiceType>WPS</ows:ServiceType>
    <ows:ServiceTypeVersion>1.0.0</ows:ServiceTypeVersion>
    <ows:Fees>NONE</ows:Fees>
    <ows:AccessConstraints>NONE</ows:AccessConstraints>
  </ows:ServiceIdentification>
  <ows:ServiceProvider>
    <ows:ProviderName>Envitia Group Ltd.</ows:ProviderName>
    <ows:ProviderSite xlink:href="http://www.envitia.com/"></ows:ProviderSite>
    <ows:ServiceContact>
      <ows:IndividualName>Mr. A Person</ows:IndividualName>
      <ows:PositionName>Engineer</ows:PositionName>
      <ows:ContactInfo>
        <ows:Phone>
          <ows:Voice>+44 1403 273 173</ows:Voice>
          <ows:Facsimile>+44 1403 273 123</ows:Facsimile>
        </ows:Phone>
        <ows:Address>
          <ows:DeliveryPoint>North Heath Lane</ows:DeliveryPoint>
          <ows:City>Horsham</ows:City>
          <ows:AdministrativeArea>West Sussex</ows:AdministrativeArea>
          <ows:PostalCode>RH12 5UX</ows:PostalCode>
          <ows:Country>England</ows:Country>
        </ows:Address>
      </ows:ContactInfo>
    </ows:ServiceContact>
    <ows:ElectronicMailAddress>support@envitia.com</ows:ElectronicMailAddress>
  </ows:ServiceProvider>
  <ows:OperationsMetadata>
    <ows:Operation name="GetCapabilities">
      <ows:DCP>
        <ows:HTTP>
          <ows:Get xlink:href="DYNAMIC"/>
          <ows:Post xlink:href="DYNAMIC"/>
        </ows:HTTP>
      </ows:DCP>
    </ows:Operation>
    <ows:Operation name="DescribeProcess">
      <ows:DCP>
        <ows:HTTP>
          <ows:Get xlink:href="DYNAMIC"/>
          <ows:Post xlink:href="DYNAMIC"/>
        </ows:HTTP>
      </ows:DCP>
    </ows:Operation>
    <ows:Operation name="Execute">
      <ows:DCP>
        <ows:HTTP>
          <ows:Get xlink:href="DYNAMIC"/>
          <ows:Post xlink:href="DYNAMIC"/>
        </ows:HTTP>
      </ows:DCP>
    </ows:Operation>
  </ows:OperationsMetadata>
</mwps:LanguageSpecificMetadata>
<mwps:LanguageSpecificMetadata xml:lang="en-US">
  <ows:ServiceIdentification>
    <ows:Title>Envitia MapLink WPS Server</ows:Title>
    <ows:Abstract>Envitia MapLink WPS Server</ows:Abstract>
    <ows:Keywords>
      <ows:Keyword>Envitia</ows:Keyword>
    </ows:Keywords>
  </ows:ServiceIdentification>

```

```

    <ows:Keyword>MapLink</ows:Keyword>
  </ows:Keywords>
  <ows:ServiceType>WPS</ows:ServiceType>
  <ows:ServiceTypeVersion>1.0.0</ows:ServiceTypeVersion>
  <ows:Fees>NONE</ows:Fees>
  <ows:AccessConstraints>NONE</ows:AccessConstraints>
</ows:ServiceIdentification>
<ows:ServiceProvider>
  <ows:ProviderName>Envitia Group Ltd.</ows:ProviderName>
  <ows:ProviderSite xlink:href="http://www.envitia.com/"></ows:ProviderSite>
  <ows:ServiceContact>
    <ows:IndividualName>Mr. A Person</ows:IndividualName>
    <ows:PositionName>Engineer</ows:PositionName>
    <ows:ContactInfo>
      <ows:Phone>
        <ows:Voice>+44 1403 273 173</ows:Voice>
        <ows:Facsimile>+44 1403 273 123</ows:Facsimile>
      </ows:Phone>
      <ows:Address>
        <ows:DeliveryPoint>North Heath Lane</ows:DeliveryPoint>
        <ows:City>Horsham</ows:City>
        <ows:AdministrativeArea>West Sussex</ows:AdministrativeArea>
        <ows:PostalCode>RH12 5UX</ows:PostalCode>
        <ows:Country>England</ows:Country>
      </ows:Address>
    </ows:ContactInfo>
  </ows:ServiceContact>
  <ows:ElectronicMailAddress>support@envitia.com</ows:ElectronicMailAddress>
</ows:ServiceProvider>
<ows:OperationsMetadata>
  <ows:Operation name="GetCapabilities">
    <ows:DCP>
      <ows:HTTP>
        <ows:Get xlink:href="DYNAMIC"/>
        <ows:Post xlink:href="DYNAMIC"/>
      </ows:HTTP>
    </ows:DCP>
  </ows:Operation>
  <ows:Operation name="DescribeProcess">
    <ows:DCP>
      <ows:HTTP>
        <ows:Get xlink:href="DYNAMIC"/>
        <ows:Post xlink:href="DYNAMIC"/>
      </ows:HTTP>
    </ows:DCP>
  </ows:Operation>
  <ows:Operation name="Execute">
    <ows:DCP>
      <ows:HTTP>
        <ows:Get xlink:href="DYNAMIC"/>
        <ows:Post xlink:href="DYNAMIC"/>
      </ows:HTTP>
    </ows:DCP>
  </ows:Operation>
</ows:OperationsMetadata>
</mwps:LanguageSpecificMetadata>
</mwps:WPSConfiguration>

```

## 6.4. WPS Router Plugin

### 6.4.1. Deployment

Before the deployment of a WPS Plugin can take place it is assumed all WPS deployment steps have already taken place.

- Add a new 'DataSource' xml element to the MapLink WPS Configuration File.
- Set the 'Plugin' element to 'RouterWPSPlugin'
- Set the 'ConfigPath' element to point to the RouterWPSplugin.ini file (a copy of which is located in the config/ogcservices folder).
- Leave the DataPath xml element empty as it is not used.

Sample:

```
<mwps:DataSource>
  <mwps:Plugin>RouterWPSplugin</mwps:Plugin>
  <mwps:DataPath></mwps:DataPath>
  <mwps:ConfigPath>
    config\plugins\RouterWPSplugin.ini
  </mwps:ConfigPath>
</mwps:DataSource>
```

Check through the RouterWPSplugin.ini file. The configuration file's comments will instruct on what the various values mean. Take note that the following entries will require attention:

- The 'transformsDatFile' value at the top of the configuration file.
- Each 'Network' setup must be scrutinised.

### 6.4.2. Describe Process

#### Sample GET Call

<http://localhost:8080/MapLinkOGCServices/OGC?&service=WPS&request=DescribeProcess&version=1.0.0&identifier=RouteWPS>

#### Sample POST Call

<http://localhost:8080/MapLinkOGCServices/OGC?>

#### POST data:

```
<?xml version="1.0" encoding="UTF-8"?>
  <wps:DescribeProcess service="WPS" version="1.0.0"
  xmlns:wps="http://www.opengis.net/wps/1.0.0"
  xmlns:ows="http://www.opengis.net/ows/1.1" >
    <ows:Identifier>RouteWPS</ows:Identifier>
  </wps:DescribeProcess >
```

The response to this request will provide:

- A list of network definitions available
- A list of available Route Algorithms (short or quick)
- A list of available Cost Algorithms (simple, heuristic etc.)

- A list of available Vehicle Types
- Details of the start the end locations
- A list of available outputs (GML and Directions)

### 6.4.3. Execute

#### Sample GET Call

[http://localhost:8080/MapLinkOGCServices/OGC?&service=WPS&request=Execute&version=1.0.0&identifier=RouteWPS&datainputs=network=63843-SZ1085;routeAlgorithm=shortest;start\\_lon=-1.7786362;start\\_lat=50.7356394;end\\_lon=-1.7804886;end\\_lat=50.7355128;costAlgorithm=simple;vehicleType=car&responseDocument=gml;directions](http://localhost:8080/MapLinkOGCServices/OGC?&service=WPS&request=Execute&version=1.0.0&identifier=RouteWPS&datainputs=network=63843-SZ1085;routeAlgorithm=shortest;start_lon=-1.7786362;start_lat=50.7356394;end_lon=-1.7804886;end_lat=50.7355128;costAlgorithm=simple;vehicleType=car&responseDocument=gml;directions)

#### Sample POST Call

See Section 7.1.1.

#### 6.4.3.1. Available Data Input Parameters

##### network (Network Identifier)

- This specifies which network map to plan a route against.
- The available networks are defined in the 'RouterWPSplugin.ini' file.
- The value for this parameter must match the configuration file's 'identifier' value within any of the 'Network' sections.

##### routeAlgorithm (Route Algorithm)

- This specifies whether the quickest or shortest route should be built.
- The available options for this are determined by implementation, these cannot be added to without development.
- The shortest option will only make choices based on distance.
- The quickest option will make choices based on the speed the vehicle type can travel on each of the different road types.
- The vehicle's speed for each road type is defined in the 'RouterWPSplugin.ini' file, in the appropriate 'Vehicle' section.

##### costAlgorithm (Cost Algorithm)

- This specifies the type of algorithm to use while building the route.
- The available options for this are determined by implementation, these cannot be added to without development.
- Each algorithm uses different methodology to try and speed up large route calculations
- A simple description of each algorithm:
  - Simple – uses the Network SDK's fundamental route crawling algorithm to build the route

- Heuristic – uses a bounding box around the start and end locations to limit the smaller roads used
- Multiple Heuristic – uses a series of bounding boxes around the start and end points each will limit certain types of roads used.

### **vehicleType (Vehicle Type)**

- This specifies the type of vehicle to plan a route for
- The available vehicle types are defined in the 'RouterWPSplugin.ini' file.
- The value for this parameter must match the configuration file's 'vehicleType' value within any of the 'Vehicle' sections.

### **start\_lon/start\_lat Start Location**

- This specifies the start location of the route to calculate
- The coordinate system these points must be in are WGS84 (lat/lon)
- The actual start point of the route will be the nearest point found on the Network Map to this start location

### **end\_lon/end\_lat Start Location**

- This specifies the end location of the route to calculate
- The coordinate system these points must be in are WGS84 (lat/lon)
- The actual end point of the route will be the nearest point found on the Network Map to this end location

### **6.4.3.2. Response Document/ Raw Data Parameters**

There are two available output formats:

- gml
- directions

Note: A response document can return both of these formats if needed.

Note: The Raw Data output can only return one of the two available formats.

#### **GML**

- This is a GML string of the calculated route.
- The coordinates system for each point will be in WGS84 (lat/lon)

#### **Directions**

- This is a list of string instructions the user needs to follow
- The distance in brackets specifies the distance required for the next instruction



## 6.5. WPS View Shed Plugin

### 6.5.1. Deployment

Before the deployment of a WPS Plugin can take place it is assumed all WPS deployment steps have already taken place.

- Add a new 'DataSource' xml element to the MapLink WPS Configuration File.
- Set the 'Plugin' element to 'ViewShedWPSPlugin'
- Set the 'ConfigPath' element to point to the ViewShedWPSplugin.ini file (a copy of which is located in the config/ogcservices folder).
- Leave the DataPath xml element empty as it is not used.

#### Sample:

```
<mwps:DataSource>
  <mwps:Plugin>ViewShedWPSplugin</mwps:Plugin>
  <mwps:DataPath></mwps:DataPath>
  <mwps:ConfigPath>
    config\plugins\ViewShedWPSplugin.ini
  </mwps:ConfigPath>
</mwps:DataSource>
```

Check through the ViewShedWPSplugin.ini file. The configuration file's comments will instruct on what the various values mean. Take note that each 'SourceData' section must be scrutinised.

### 6.5.2. Describe Process

#### Sample GET Call

<http://localhost:8080/MapLinkOGCServices/OGC?&service=WPS&request=DescribeProcess&version=1.0.0&identifier=ViewShedWPS>

#### Sample POST Call

<http://localhost:8080/MapLinkOGCServices/OGC?>

*POST data:*

```
<?xml version="1.0" encoding="UTF-8"?>
  <wps:DescribeProcess service="WPS" version="1.0.0"
  xmlns:wps="http://www.opengis.net/wps/1.0.0"
  xmlns:ows="http://www.opengis.net/ows/1.1" >
    <ows:Identifier>ViewShedWPS</ows:Identifier>
  </wps:DescribeProcess >
```

#### Available Options

In the sample above the 'ViewShedWPS' value represents the standard Single View Shed service. The available options for services are:

- ViewShedWPS
- MultiViewShedWPS
- RouteViewShedWPS

- RouteBreakdownWPS

### Response Description

The response to this request will provide details of the parameters and outputs each service provides.

## 6.5.3. Single View Shed Execute

### Sample GET Call

[http://localhost:8080/MapLinkOGCServices/OGC?&service=WPS&request=Execute&version=1.0.0&identifier=ViewShedWPS&datainputs=source=sanfran;view\\_lat=37.711949;view\\_lon=-122.308167;view\\_height=0;view\\_htype=groundHeight;view\\_minRadius=0;view\\_maxRadius=10000;target\\_height=0;target\\_htype=groundHeight;requiredDisplayWidth=800;requiredDisplayHeight=600;requiredDisplayExtent=-122.384258,37.716004,-122.357822,37.734605,EPSSG:4326;displayStyle=redGreen&RawDataOutput=image=@mime\\_type=image/png](http://localhost:8080/MapLinkOGCServices/OGC?&service=WPS&request=Execute&version=1.0.0&identifier=ViewShedWPS&datainputs=source=sanfran;view_lat=37.711949;view_lon=-122.308167;view_height=0;view_htype=groundHeight;view_minRadius=0;view_maxRadius=10000;target_height=0;target_htype=groundHeight;requiredDisplayWidth=800;requiredDisplayHeight=600;requiredDisplayExtent=-122.384258,37.716004,-122.357822,37.734605,EPSSG:4326;displayStyle=redGreen&RawDataOutput=image=@mime_type=image/png)

### Sample POST Call

See Section 7.1.2.

#### 6.5.3.1. Available Data Input Parameters

##### source (Source Data)

- This specifies which source data to perform a View Shed against.
- The available source data are defined in the 'ViewShedWPSplugin.ini' file.
- The value for this parameter must match the configuration file's 'identifier' value within any of the 'SourceData' sections.

##### view\_lat/view\_lon (Viewing Location)

- This specifies the viewing location for the View Shed.
- The coordinate system these points must be in WGS84 (lat/lon).

##### view\_height (Viewing Height)

- This specifies the viewing height in meters.
- The maximum height for any particular source data is defined in the 'ViewShedWPSplugin.ini' file.
- This parameter is used in conjunction with the 'view\_htype' parameter.

##### view\_htype (Viewing Height Type)

- This specifies where the 'view\_height' originates from.
- There are two options:
  - Surface Height
  - Absolute Height
- Available height types are defined in code.

##### view\_maxRadius (Viewing Maximum Radius)

- This specifies the maximum radius of the view shed.
- The maximum radius for any particular source data is defined in the 'ViewShedWSPplugin.ini' file.

**target\_height (Target Height)**

- This specifies the target height in meters.
- This means the resulting image will only show consider a point visible if it is visible at the height specified by both this parameter and the 'target\_htype' parameter.
- The maximum height for any particular source data is defined in the 'ViewShedWSPplugin.ini' file.
- This parameter is used in conjunction with the 'target\_htype' parameter.

**target\_htype (Target Height Type)**

- This specifies where the 'target\_height' originates from.
- There are two options:
  - Surface Height
  - Absolute Height
- Available height types are defined in code.

**requiredDisplayWidth/requiredDisplayHeight (Required Display)**

- These two parameters define the resulting image's size in pixels.
- The image is first generated based on the source data's resolution, it is then scaled to fit the required size.

**requiredDisplayExtent (Required Display Extent)**

- This specifies the lat/lon bounding box the resulting image will cover.

**displayStyle (Display Style)**

- This specifies the display style to use when generating the View Shed image.
- The available display styles are defined in the 'ViewShedWSPplugin.ini' file.
- The value for this parameter must match the configuration file's 'identifier' value within any of the 'Colour' sections.

### 6.5.3.2. Response Document/ Raw Data Parameters

There is only one type of output, which is the resulting View Shed image.

**Available Formats**

There are two available image formats:

- png
- tiff - this is not always recognised by internet browsers, so can force a download instead of being able to view directly in the browser.

Note: The resulting format will not have an alpha channel.

**Distribution**

There are two ways of distributing the image:

- Raw Data Format - This will simply return the binary image directly in the response without any XML.
- Response Document as reference - This will store the resulting image in the WPS store and will return a url via the XML response. The caller can use this url to access the image. The length of time the image will be stored is defined in the 'MapLinkWPSConfiguration.xml' file.

Note: Complex binary data (image data) cannot be returned through the response document directly, only as a reference (this is as per the WPS specifications).

## 6.5.4. Multi View Shed Execute

### Sample GET Call

[http://localhost:8080/MapLinkOGCServices/OGC?&service=WPS&request=Execute&version=1.0.0&identifier=MultiViewShedWPS&datainputs=source=sanfran;viewPoints=-122.308167,37.711949,0;view\\_htype=groundHeight;view\\_maxRadius=10000;target\\_height=0;target\\_htype=groundHeight;requiredDisplayWidth=800;requiredDisplayHeight=600;requiredDisplayExtent=-122.384258,37.716004,-122.357822,37.734605,EPSSG:4326;displayStyle=redGreen&RawDataOutput=image=@mime-type=image/png](http://localhost:8080/MapLinkOGCServices/OGC?&service=WPS&request=Execute&version=1.0.0&identifier=MultiViewShedWPS&datainputs=source=sanfran;viewPoints=-122.308167,37.711949,0;view_htype=groundHeight;view_maxRadius=10000;target_height=0;target_htype=groundHeight;requiredDisplayWidth=800;requiredDisplayHeight=600;requiredDisplayExtent=-122.384258,37.716004,-122.357822,37.734605,EPSSG:4326;displayStyle=redGreen&RawDataOutput=image=@mime-type=image/png)

### Sample POST Call

See Section 7.1.3.

#### 6.5.4.1. Available Data Input Parameters

##### source (Source Data)

- This specifies which source data to perform a View Shed against.
- The available source data are defined in the 'ViewShedWPSplugin.ini' file.
- The value for this parameter must match the configuration file's 'identifier' value within any of the 'SourceData' sections.

##### viewPoints (Viewing Location)

- This specifies a set of viewing locations, each which will generate a View Shed
- The XY points of the viewing location must be in WGS84 (lat/lon)
- Each location will also specify a height in meters.
- The height will either be the absolute height or the surface height (this is defined by the 'view\_htype' parameter)
- The maximum height for any particular source data is defined in the 'ViewShedWPSplugin.ini' file.

##### view\_htype (Viewing Height Type)

- This specifies where the viewing height points originate from.
- There are two options:
  - Surface Height
  - Absolute Height
- Available height types are defined in code

##### view\_maxRadius (Viewing Maximum Radius)

- This specifies the maximum radius of the view shed.
- The maximum radius for any particular source data is defined in the 'ViewShedWSPplugin.ini' file.

**target\_height (Target Height)**

- This specifies the target height in meters.
- This means the resulting image will only show consider a point visible if it is visible at the height specified by both this parameter and the 'target\_htype' parameter.
- The maximum height for any particular source data is defined in the 'ViewShedWSPplugin.ini' file.
- This parameter is used in conjunction with the 'target\_htype' parameter.

**target\_htype (Target Height Type)**

- This specifies where the 'target\_height' originates from.
- There are two options:
  - Surface Height
  - Absolute Height
- Available height types are defined in code

**requiredDisplayWidth/requiredDisplayHeight (Required Display)**

- These two parameters define the resulting image's size in pixels.
- The image is first generated based on the source data's resolution, it is then scaled to fit the required size.

**requiredDisplayExtent (Required Display Extent)**

- This specifies the lat/lon bounding box the resulting image will cover.

**displayStyle (Display Style)**

- This specifies the display style to use when generating the View Shed image.
- The available display styles are defined in the 'ViewShedWSPplugin.ini' file.
- The value for this parameter must match the configuration file's 'identifier' value within any of the 'Colour' sections.

### 6.5.4.2. Response Document/ Raw Data Parameters

There is only one type of output, which is the resulting View Shed image.

**Available Formats**

There are two available image formats:

- png
- tiff - this is not always recognised by internet browsers, so can force a download instead of being able to view directly in the browser.

Note: The resulting format will not have an alpha channel.

**Distribution**

There are two ways of distributing the image:

- Raw Data Format - This will simply return the binary image directly in the response without any XML.
- Response Document as reference - This will store the resulting image in the WPS store and will return a url via the XML response. The caller can use this url to access the image. The length of time the image will be stored is defined in the 'MapLinkWPSConfiguration.xml' file.

Note: Complex binary data (image data) cannot be returned through the response document directly, only as a reference (this is as per the WPS specifications).

## 6.5.5. Route View Shed Execute

### Sample GET Call

[http://localhost:8080/MapLinkOGCServices/OGC?&service=WPS&request=Execute&version=1.0.0&identifier=RouteViewShedWPS&datainputs=source=britsouth;routePoints=51.1244288,-1.8908794,51.1244876,-1.8914126,51.1245858,-1.8921404,51.1245918,-1.892175;viewShedPointSpacing=300;view\\_height=0;view\\_htype=groundHeight;view\\_maxRadius=3000;target\\_height=0;target\\_htype=groundHeight;requiredDisplayWidth=800;requiredDisplayHeight=600;requiredDisplayExtent=-1.98,51.096,-1.7925,51.2125,EPSG:4326;displayStyle=redGreen&RawDataOutput=image=@mimetype=image/png](http://localhost:8080/MapLinkOGCServices/OGC?&service=WPS&request=Execute&version=1.0.0&identifier=RouteViewShedWPS&datainputs=source=britsouth;routePoints=51.1244288,-1.8908794,51.1244876,-1.8914126,51.1245858,-1.8921404,51.1245918,-1.892175;viewShedPointSpacing=300;view_height=0;view_htype=groundHeight;view_maxRadius=3000;target_height=0;target_htype=groundHeight;requiredDisplayWidth=800;requiredDisplayHeight=600;requiredDisplayExtent=-1.98,51.096,-1.7925,51.2125,EPSG:4326;displayStyle=redGreen&RawDataOutput=image=@mimetype=image/png)

### Sample POST Call

See Section 7.1.4.

#### 6.5.5.1. Available Data Input Parameters

##### source (Source Data)

- This specifies which source data to perform a View Shed against.
- The available source data are defined in the 'ViewShedWPSplugin.ini' file.
- The value for this parameter must match the configuration file's 'identifier' value within any of the 'SourceData' sections.

##### routePoints (Viewing Location)

- This is a GML string.
- This specifies a set of viewing locations, each which will generate a View Shed
- The XY points of the viewing location must be in WGS84 (lat/lon)

##### viewShedPointSpacing

- This defines the number of meters to exist between each point of the new version of the route.

##### view\_height (Viewing Height)

- This specifies the viewing height in meters.
- The maximum height for any particular source data is defined in the 'ViewShedWPSplugin.ini' file.
- This parameter is used in conjunction with the 'view\_htype' parameter.

##### view\_htype (Viewing Height Type)

- This specifies where the viewing height points originate from.
- There are two options:
  - Surface Height
  - Absolute Height
- Available height types are defined in code

##### view\_maxRadius (Viewing Maximum Radius)

- This specifies the maximum radius of the view shed.

- The maximum radius for any particular source data is defined in the 'ViewShedWPSplugin.ini' file.

**target\_height (Target Height)**

- This specifies the target height in meters.
- This means the resulting image will only show consider a point visible if it is visible at the height specified by both this parameter and the 'target\_htype' parameter.
- The maximum height for any particular source data is defined in the 'ViewShedWPSplugin.ini' file.
- This parameter is used in conjunction with the 'target\_htype' parameter.

**target\_htype (Target Height Type)**

- This specifies where the 'target\_height' originates from.
- There are two options:
  - Surface Height
  - Absolute Height
- Available height types are defined in code

**requiredDisplayWidth/requiredDisplayHeight (Required Display)**

- These two parameters define the resulting image's size in pixels.
- The image is first generated based on the source data's resolution, it is then scaled to fit the required size.

**requiredDisplayExtent (Required Display Extent)**

- This specifies the lat/lon bounding box the resulting image will cover.

**displayStyle (Display Style)**

- This specifies the display style to use when generating the View Shed image.
- The available display styles are defined in the 'ViewShedWPSplugin.ini' file.
- The value for this parameter must match the configuration file's 'identifier' value within any of the 'Colour' sections.

### 6.5.5.2. Response Document/ Raw Data Parameters

There is only one type of output, which is the resulting View Shed image.

**Available Formats**

There are two available image formats:

- png
- tiff - this is not always recognised by internet browsers, so can force a download instead of being able to view directly in the browser.

Note: The resulting format will not have an alpha channel.

**Distribution**

There are two ways of distributing the image:

- Raw Data Format - This will simply return the binary image directly in the response without any XML.



- Response Document as reference - This will store the resulting image in the WPS store and will return a url via the XML response. The caller can use this url to access the image. The length of time the image will be stored is defined in the 'MapLinkWPSConfiguration.xml' file.

Note: Complex binary data (image data) cannot be returned through the response document directly, only as a reference (this is as per the WPS specifications).

## 6.5.6. Route Breakdown Execute

### Sample GET Call

<http://localhost:8080/MapLinkOGCServices/OGC?&service=WPS&request=Execute&version=1.0.0&identifier=RouteBreakdownWPS&datainputs=routePoints=51.1244288,-1.8908794,51.1247762,-1.8931956;viewShedPointSpacing=100&RawDataOutput=gmlDoc=@mimetype=text/xml>

### Sample POST Call

See Section 7.1.5.

### 6.5.6.1. Available Data Input Parameters

#### routePoints (Viewing Location)

- This is a GML string
- This specifies a set of viewing locations, each which will generate a View Shed
- The XY points of the viewing location must be in WGS84 (lat/lon)

#### viewShedPointSpacing

- This defines the number of meters to exist between each point of the new version of the route.

#### view\_height (Viewing Height)

- This specifies the viewing height in meters.
- The maximum height for any particular source data is defined in the 'ViewShedWPSplugin.ini' file.
- This parameter is used in conjunction with the 'view\_htype' parameter.

#### view\_htype (Viewing Height Type)

- This specifies where the viewing height points originate from.
- There are two options:
  - Surface Height
  - Absolute Height
- Available height types are defined in code

#### view\_maxRadius (Viewing Maximum Radius)

- This specifies the maximum radius of the view shed.
- The maximum radius for any particular source data is defined in the 'ViewShedWPSplugin.ini' file.

#### target\_height (Target Height)

- This specifies the target height in meters.
- This means the resulting image will only show consider a point visible if it is visible at the height specified by both this parameter and the 'target\_htype' parameter.
- The maximum height for any particular source data is defined in the 'ViewShedWPSplugin.ini' file.
- This parameter is used in conjunction with the 'target\_htype' parameter.

**target\_htype (Target Height Type)**

- This specifies where the 'target\_height' originates from.
- There are two options:
  - Surface Height
  - Absolute Height
- Available height types are defined in code

**requiredDisplayWidth/requiredDisplayHeight (Required Display)**

- These two parameters define the resulting image's size in pixels.
- The image is first generated based on the source data's resolution, it is then scaled to fit the required size.

**requiredDisplayExtent (Required Display Extent)**

- This specifies the lat/lon bounding box the resulting image will cover.

**displayStyle (Display Style)**

- This specifies the display style to use when generating the View Shed image.
- The available display styles are defined in the 'ViewShedWPSplugin.ini' file.
- The value for this parameter must match the configuration file's 'identifier' value within any of the 'Colour' sections.

### 6.5.6.2. Response Document/ Raw Data Parameters

There is only one type of output, which is the resulting GML String of the broken down route.

**Distribution**

There are two ways of distributing the image:

- Raw Data Format - This will simply return the GML string directly in the response without any XML.
- Response Document as reference - This will store the resulting GML string the WPS store and will return a url via the XML response. The caller can use this url to access the GML string. The length of time the GML string will be stored is defined in the 'MapLinkWPSConfiguration.xml' file.
- Response Document - This will return the GML string in a response document. The GML string will be encoded.

## 6.6. WPS Terrain Profile Plugin

### 6.6.1. Deployment

Before the deployment of a WPS Plugin can take place it is assumed all WPS deployment steps have already taken place.

- Add a new 'DataSource' xml element to the MapLink WPS Configuration File.
- Set the 'Plugin' element to 'TerrainProfileWPSPlugin'
- Set the 'ConfigPath' element to point to the TerrainProfileWPSplugin.ini file (a copy of which is located in the config/plugins folder).

- Leave the DataPath xml element empty as it is not used.

**Sample:**

```
<mwps:DataSource>
  <mwps:Plugin>TerrainProfileWPSPlugin</mwps:Plugin>
  <mwps:DataPath></mwps:DataPath>
  <mwps:ConfigPath>
    config\plugins\ViewShedWPSplugin.ini
  </mwps:ConfigPath>
</mwps:DataSource>
```

Check through the TerrainProfileWPSplugin.ini file. The configuration file's comments will instruct on what the various values mean. Take note that each 'SourceData' section must be scrutinised.

## 6.6.2. Describe Process

**Sample GET Call**

<http://localhost:8080/MapLinkOGCServices/OGC?&service=WPS&request=DescribeProcess&version=1.0.0&identifier=TerrainProfileWPS>

**Sample POST Call**

<http://localhost:8080/MapLinkOGCServices/OGC?>

*POST data:*

```
<?xml version="1.0" encoding="UTF-8"?>
  <wps:DescribeProcess service="WPS" version="1.0.0"
  xmlns:wps="http://www.opengis.net/wps/1.0.0"
  xmlns:ows="http://www.opengis.net/ows/1.1" >
    <ows:Identifier>TerrainProfileWPS</ows:Identifier>
  </wps:DescribeProcess >
```

**Response Description**

The response to this request will provide details of the parameters and outputs each service provides.

## 6.6.3. Execute

**Sample GET Call**

<http://localhost:8080/MapLinkOGCServices/OGC?&service=WPS&request=Execute&version=1.0.0&identifier=TerrainProfileWPS&datainputs=source=sanfran;profilePoints=37.726258,-122.111137,37.724840,-122.108669;viewShedPointSpacing=100;requiredDisplayWidth=800;requiredDisplayHeight=600;displayStyle=clearVis&RawDataOutput=image=@mimetype=image/png>

**Sample POST Call**

See Section 7.1.6.

### 6.6.3.1. Available Data Input Parameters

**source (Source Data)**

- This specifies which source data to perform a Terrain Profile against.
- The available source data are defined in the 'TerrainProfileWPSplugin.ini' file.

- The value for this parameter must match the configuration file's 'identifier' value within any of the 'SourceData' sections.

**profilePoints**

- This is a GML string (for GET requests it is a comma delimited list of coordinates)
- This specifies a route which will be simplified, the result of which is used to generate a Terrain Profile.
- The XY points of the viewing location must be in WGS84 (lat/lon)

**breakdownPointSpacing**

- This defines the number of meters to exist between each profile point above.

**scaleOutput**

- This sets whether the resulting Terrain Profile image is scaled vertically
- This parameter is optional
- By default this is flag off

**lowestDisplayHeight**

- This specifies the lowest height to display in the resulting Terrain Profile image.
- This parameter is optional
- If not provided then the Lowest Display Height will be defined by the lowest Height in the Terrain Profile itself.

**highestDisplayHeight**

- This specifies the highest height to display in the resulting Terrain Profile image.
- This parameter is optional
- If not provided then the Highest Display Height will be defined by the highest Height in the Terrain Profile itself.

**requiredDisplayWidth/requiredDisplayHeight (Required Display)**

- These two parameters define the resulting image's size in pixels.

**displayStyle (Display Style)**

- This specifies the display style to use when generating the Terrain Profile image.
- The available display styles are defined in the 'TerrainProfileWPSplugin.ini' file.
- The value for this parameter must match the configuration file's 'identifier' value within any of the 'Colour' sections.

### 6.6.3.2. Response Document/ Raw Data Parameters

There is only one type of output, which is the resulting Terrain Profile image.

**Available Formats**

There are two available image formats:

- png
- tiff - this is not always recognised by internet browsers, so can force a download instead of being able to view directly in the browser.

Note: The resulting format will not have an alpha channel.

### Distribution

There are two ways of distributing the image:

- Raw Data Format - This will simply return the binary image directly in the response without any XML.
- Response Document as reference - This will store the resulting image in the WPS store and will return a url via the XML response. The caller can use this url to access the image. The length of time the image will be stored is defined in the 'MapLinkWPSConfiguration.xml' file.

Note: Complex binary data (image data) cannot be returned through the response document directly, only as a reference (this is as per the WPS specifications).

## 6.7. WPS Import Raster Plugin

### 6.7.1. Deployment

Before the deployment of a WPS Plugin can take place it is assumed all WPS deployment steps have already taken place.

- Add a new 'DataSource' xml element to the MapLink WPS Configuration File.
- Set the 'Plugin' element to 'TerrainProfileWPSPlugin'
- Set the 'ConfigPath' element to point to the ImportRasterWPSplugin.ini file (a copy of which is located in the config/plugins folder).
- Leave the DataPath xml element empty as it is not used.

Sample:

```
<mwps:DataSource>
  <mwps:Plugin>ImportRasterWPSPlugin</mwps:Plugin>
  <mwps:DataPath></mwps:DataPath>
  <mwps:ConfigPath>
    config\plugins\ImportRasterWPSplugin.ini
  </mwps:ConfigPath>
</mwps:DataSource>
```

Check through the ImportRasterWPSplugin.ini file. The configuration file's comments will instruct on what the various values mean.

### 6.7.2. Describe Process

#### Sample GET Call

<http://localhost:8080/MapLinkOGCServices/OGC?&service=WPS&request=DescribeProcess&version=1.0.0&identifier=ImportRasterWPS>

#### Sample POST Call

<http://localhost:8080/MapLinkOGCServices/OGC?>

*POST data:*

```
<?xml version="1.0" encoding="UTF-8"?>
  <wps:DescribeProcess service="WPS" version="1.0.0"
  xmlns:wps="http://www.opengis.net/wps/1.0.0"
  xmlns:ows="http://www.opengis.net/ows/1.1" >
    <ows:Identifier>ImportRasterWPS</ows:Identifier>
  </wps:DescribeProcess >
```

### Response Description

The response to this request will provide details of the parameters and outputs each service provides.

## 6.7.3. Execute

### Sample GET Call

<http://localhost:8080/MapLinkOGCServices/OGC?&service=WPS&request=Execute&version=1.0.0&identifier=TerrainProfileWPS&datainputs=source=sanfran;profilePoints=37.726258,-122.111137,37.724840,-122.108669;viewShedPointSpacing=100;requiredDisplayWidth=800;requiredDisplayHeight=600;displayStyle=clearVis&RawDataOutput=image=@mimetype=image/png>

### Sample POST Call

See Section 7.1.6.

#### 6.7.3.1. Available Data Input Parameters

##### source (Source Data)

- This specifies which source data to perform a Terrain Profile against.
- The available source data are defined in the 'TerrainProfileWPSplugin.ini' file.
- The value for this parameter must match the configuration file's 'identifier' value within any of the 'SourceData' sections.

##### profilePoints

- This is a GML string (for GET requests it is a comma delimited list of coordinates)
- This specifies a route which will be simplified, the result of which is used to generate a Terrain Profile.
- The XY points of the viewing location must be in WGS84 (lat/lon)

##### breakdownPointSpacing

- This defines the number of meters to exist between each profile point above.

##### scaleOutput

- This sets whether the resulting Terrain Profile image is scaled vertically
- This parameter is optional
- By default this is flag off

##### lowestDisplayHeight

- This specifies the lowest height to display in the resulting Terrain Profile image.
- This parameter is optional
- If not provided then the Lowest Display Height will be defined by the lowest Height in the Terrain Profile itself.

**highestDisplayHeight**

- This specifies the highest height to display in the resulting Terrain Profile image.
- This parameter is optional
- If not provided then the Highest Display Height will be defined by the highest Height in the Terrain Profile itself.

**requiredDisplayWidth/requiredDisplayHeight (Required Display)**

- These two parameters define the resulting image's size in pixels.

**displayStyle (Display Style)**

- This specifies the display style to use when generating the Terrain Profile image.
- The available display styles are defined in the 'TerrainProfileWPSplugin.ini' file.
- The value for this parameter must match the configuration file's 'identifier' value within any of the 'Colour' sections.

### 6.7.3.2. Response Document/ Raw Data Parameters

There is only one type of output, which is the resulting Terrain Profile image.

**Available Formats**

There are two available image formats:

- png
- tiff - this is not always recognised by internet browsers, so can force a download instead of being able to view directly in the browser.

Note: The resulting format will not have an alpha channel.

**Distribution**

There are two ways of distributing the image:

- Raw Data Format - This will simply return the binary image directly in the response without any XML.
- Response Document as reference - This will store the resulting image in the WPS store and will return a url via the XML response. The caller can use this url to access the image. The length of time the image will be stored is defined in the 'MapLinkWPSConfiguration.xml' file.

Note: Complex binary data (image data) cannot be returned through the response document directly, only as a reference (this is as per the WPS specifications).



## 7. APPENDIX A

### 7.1. WPS Plugin Sample Execute POST Calls

#### 7.1.1. Router

Calculate a route with a response document returned

```
<?xml version="1.0" encoding="UTF-8"?>
<wps:Execute service="WPS" version="1.0.0" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" >
  <ows:Identifier>RouteWPS</ows:Identifier>
  <wps>DataInputs>
    <wps:Input>
      <ows:Identifier>network</ows:Identifier>
      <ows:Title>Network Identifier</ows:Title>
      <wps>Data>
        <wps:LiteralData>63843-SZ1085</wps:LiteralData>
      </wps>Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>routeAlgorithm</ows:Identifier>
      <ows:Title>Route Algorithm</ows:Title>
      <wps>Data>
        <wps:LiteralData>shortest</wps:LiteralData>
      </wps>Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>costAlgorithm</ows:Identifier>
      <ows:Title>Cost Algorithm</ows:Title>
      <wps>Data>
        <wps:LiteralData>simple</wps:LiteralData>
      </wps>Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>vehicleType</ows:Identifier>
      <ows:Title>Vehicle Type</ows:Title>
      <wps>Data>
        <wps:LiteralData>car</wps:LiteralData>
      </wps>Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>start_lat</ows:Identifier>
      <ows:Title>Start Latitude</ows:Title>
      <wps>Data>
        <wps:LiteralData>50.719920462855171</wps:LiteralData>
      </wps>Data>
  </wps>DataInputs>
</wps:Execute>
```

```

</wps:Input>
<wps:Input>
  <ows:Identifier>start_lon</ows:Identifier>
  <ows:Title>Start Longitude</ows:Title>
  <wps:Data>
    <wps:LiteralData>-1.7880410120157533</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>end_lat</ows:Identifier>
  <ows:Title>End Latitude</ows:Title>
  <wps:Data>
    <wps:LiteralData>50.771014202442693</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>end_lon</ows:Identifier>
  <ows:Title>End Longitude</ows:Title>
  <wps:Data>
    <wps:LiteralData>-1.8326848043536101</wps:LiteralData>
  </wps:Data>
</wps:Input>
</wps>DataInputs>
<wps:ResponseForm>
  <wps:ResponseDocument>
    <wps:Output asReference="false" >
      <ows:Identifier>gml</ows:Identifier>
    </wps:Output>
    <wps:Output asReference="false" >
      <ows:Identifier>directions</ows:Identifier>
    </wps:Output>
  </wps:ResponseDocument>
</wps:ResponseForm>
</wps:Execute>

```

### Calculate a route with a raw output returned

```

<?xml version="1.0" encoding="UTF-8"?>
<wps:Execute service="WPS" version="1.0.0" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" >
  <ows:Identifier>RouteWPS</ows:Identifier>
  <wps>DataInputs>
    <wps:Input>
      <ows:Identifier>network</ows:Identifier>
      <ows:Title>Network Identifier</ows:Title>
      <wps:Data>
        <wps:LiteralData>63843-SZ1085</wps:LiteralData>
      </wps:Data>
    </wps:Input>
  </wps>DataInputs>
</wps:Execute>

```

```
</wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>routeAlgorithm</ows:Identifier>
  <ows:Title>Route Algorithm</ows:Title>
  <wps:Data>
    <wps:LiteralData>shortest</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>costAlgorithm</ows:Identifier>
  <ows:Title>Cost Algorithm</ows:Title>
  <wps:Data>
    <wps:LiteralData>simple</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>vehicleType</ows:Identifier>
  <ows:Title>Vehicle Type</ows:Title>
  <wps:Data>
    <wps:LiteralData>car</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>start_lat</ows:Identifier>
  <ows:Title>Start Latitude</ows:Title>
  <wps:Data>
    <wps:LiteralData>50.719920462855171</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>start_lon</ows:Identifier>
  <ows:Title>Start Longitude</ows:Title>
  <wps:Data>
    <wps:LiteralData>-1.7880410120157533</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>end_lat</ows:Identifier>
  <ows:Title>End Latitude</ows:Title>
  <wps:Data>
    <wps:LiteralData>50.771014202442693</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>end_lon</ows:Identifier>
```

```

    <ows:Title>End Longitude</ows:Title>
    <wps:Data>
      <wps:LiteralData>-1.8326848043536101</wps:LiteralData>
    </wps:Data>
  </wps:Input>
</wps>DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml">
    <ows:Identifier>gml</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

### 7.1.2. Single View Shed

Calculate a route with a response document returned

```

<?xml version="1.0" encoding="utf-8"?>
<wps:Execute service="WPS" version="1.0.0" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" >
  <ows:Identifier>${WPS_valid_identifier}</ows:Identifier>
  source=sanfran
  view_lat=37.711949
  view_lon=-122.308167
  view_height=0
  view_htype=groundHeight
  view_minRadius=0
  view_maxRadius=10000
  target_height=0
  target_htype=groundHeight
  requiredDisplayWidth=800
  requiredDisplayHeight=600
  requiredDisplayExtent=-122.420041,37.612468,-122.167013,37.822128,EPSSG:4326
  displayStyle=redGreen
  <wps>DataInputs>
    <wps:Input>
      <ows:Identifier>source</ows:Identifier>
      <ows:Title>Data Source</ows:Title>
      <wps:Data>
        <wps:LiteralData>sanfran</wps:LiteralData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>view_lat</ows:Identifier>
      <ows:Title>Viewing Latitude</ows:Title>
      <wps:Data>
        <wps:LiteralData>37.711949</wps:LiteralData>
      </wps:Data>
    </wps:Input>
  </wps>DataInputs>

```

```
</wps:Input>
<wps:Input>
  <ows:Identifier>view_lon</ows:Identifier>
  <ows:Title>Viewing Longitude</ows:Title>
  <wps:Data>
    <wps:LiteralData>-122.308167</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>view_height</ows:Identifier>
  <ows:Title>Viewing Height</ows:Title>
  <wps:Data>
    <wps:LiteralData>0</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>view_htype</ows:Identifier>
  <ows:Title>Viewing Height Type</ows:Title>
  <wps:Data>
    <wps:LiteralData>groundHeight</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>view_maxRadius</ows:Identifier>
  <ows:Title>Viewing Maximum Radius</ows:Title>
  <wps:Data>
    <wps:LiteralData>10000</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>requiredDisplayWidth</ows:Identifier>
  <ows:Title>Required Display Width</ows:Title>
  <wps:Data>
    <wps:LiteralData>800</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>requiredDisplayHeight</ows:Identifier>
  <ows:Title>Required Display Height</ows:Title>
  <wps:Data>
    <wps:LiteralData>600</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>requiredDisplayExtent</ows:Identifier>
  <ows:Title>Required Display Extent</ows:Title>
```

```

    <wps:Data>
      <wps:BoundingBoxData ows:crs="EPSG:4326" >
        <ows:LowerCorner>-122.420041 37.612468</ows:LowerCorner>
        <ows:UpperCorner>-122.167013 37.822128</ows:UpperCorner>
      </wps:BoundingBoxData>
    </wps:Data>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>displayStyle</ows:Identifier>
    <ows:Title>Display Style</ows:Title>
    <wps:Data>
      <wps:LiteralData>redGreen</wps:LiteralData>
    </wps:Data>
  </wps:Input>
</wps>DataInputs>
<wps:ResponseForm>
  <wps:ResponseDocument>
    <wps:Output mimeType="image/png" asReference="true" >
      <ows:Identifier>image</ows:Identifier>
    </wps:Output>
  </wps:ResponseDocument>
</wps:ResponseForm>
</wps:Execute>

```

### Calculate a route with a raw output returned

```

<?xml version="1.0" encoding="utf-8"?>
<wps:Execute service="WPS" version="1.0.0" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" >
  <ows:Identifier>${WPS_valid_identifier}</ows:Identifier>
  source=sanfran
  view_lat=37.711949
  view_lon=-122.308167
  view_height=0
  view_htype=groundHeight
  view_minRadius=0
  view_maxRadius=10000
  target_height=0
  target_htype=groundHeight
  requiredDisplayWidth=800
  requiredDisplayHeight=600
  requiredDisplayExtent=-122.420041,37.612468,-122.167013,37.822128,EPSSG:4326
  displayStyle=redGreen
  <wps>DataInputs>
    <wps:Input>
      <ows:Identifier>source</ows:Identifier>
      <ows:Title>Data Source</ows:Title>
      <wps:Data>
        <wps:LiteralData>sanfran</wps:LiteralData>

```

```
</wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>view_lat</ows:Identifier>
  <ows:Title>Viewing Latitude</ows:Title>
  <wps:Data>
    <wps:LiteralData>37.711949</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>view_lon</ows:Identifier>
  <ows:Title>Viewing Longitude</ows:Title>
  <wps:Data>
    <wps:LiteralData>-122.308167</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>view_height</ows:Identifier>
  <ows:Title>Viewing Height</ows:Title>
  <wps:Data>
    <wps:LiteralData>0</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>view_htype</ows:Identifier>
  <ows:Title>Viewing Height Type</ows:Title>
  <wps:Data>
    <wps:LiteralData>groundHeight</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>view_maxRadius</ows:Identifier>
  <ows:Title>Viewing Maximum Radius</ows:Title>
  <wps:Data>
    <wps:LiteralData>10000</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>requiredDisplayWidth</ows:Identifier>
  <ows:Title>Required Display Width</ows:Title>
  <wps:Data>
    <wps:LiteralData>800</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>requiredDisplayHeight</ows:Identifier>
```

```

<ows:Title>Required Display Height</ows:Title>
<wps:Data>
  <wps:LiteralData>600</wps:LiteralData>
</wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>requiredDisplayExtent</ows:Identifier>
  <ows:Title>Required Display Extent</ows:Title>
  <wps:Data>
    <wps:BoundingBoxData ows:crs="EPSG:4326" >
      <ows:LowerCorner>-122.420041 37.612468</ows:LowerCorner>
      <ows:UpperCorner>-122.167013 37.822128</ows:UpperCorner>
    </wps:BoundingBoxData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>displayStyle</ows:Identifier>
  <ows:Title>Display Style</ows:Title>
  <wps:Data>
    <wps:LiteralData>redGreen</wps:LiteralData>
  </wps:Data>
</wps:Input>
</wps>DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="image/png">
    <ows:Identifier>image</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

### 7.1.3. Multi View Shed

```

<?xml version="1.0" encoding="utf-8"?>
<wps:Execute service="WPS" version="1.0.0" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" >
  <ows:Identifier>${WPS_Multi_Service_Identifier}</ows:Identifier> source=sanfran
view_lat=37.711949 view_lon=-122.308167 view_height=0 view_htype=groundHeight
view_minRadius=0 view_maxRadius=10000 target_height=0 target_htype=groundHeight
requiredDisplayWidth=800 requiredDisplayHeight=600 requiredDisplayExtent=-
122.420041,37.612468,-122.167013,37.822128,EPSSG:4326 displayStyle=redGreen
<wps>DataInputs>
  <wps:Input>
    <ows:Identifier>source</ows:Identifier>
    <ows:Title>Data Source</ows:Title>
    <wps:Data>
      <wps:LiteralData>sanfran</wps:LiteralData>
    </wps:Data>
  </wps:Input>
  <wps:Input>

```



```

    <ows:Identifier>viewPoints</ows:Identifier>
    <ows:Title>Viewing Points</ows:Title>
    <wps:Data>
      <wps:LiteralData>&lt;gml:LineString srsName=&quot;EPSG:4326&quot;&gt;
&lt;gml:posList srsDimension=&quot;3&quot;&gt;-122.308167 37.711949 0.0&lt;/gml:posList&gt;
&lt;/gml:LineString&gt;</wps:LiteralData>
    </wps:Data>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>view_hatype</ows:Identifier>
    <ows:Title>Viewing Height Type</ows:Title>
    <wps:Data>
      <wps:LiteralData>groundHeight</wps:LiteralData>
    </wps:Data>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>view_maxRadius</ows:Identifier>
    <ows:Title>Viewing Maximum Radius</ows:Title>
    <wps:Data>
      <wps:LiteralData>10000</wps:LiteralData>
    </wps:Data>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>requiredDisplayWidth</ows:Identifier>
    <ows:Title>Required Display Width</ows:Title>
    <wps:Data>
      <wps:LiteralData>800</wps:LiteralData>
    </wps:Data>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>requiredDisplayHeight</ows:Identifier>
    <ows:Title>Required Display Height</ows:Title>
    <wps:Data>
      <wps:LiteralData>600</wps:LiteralData>
    </wps:Data>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>requiredDisplayExtent</ows:Identifier>
    <ows:Title>Required Display Extent</ows:Title>
    <wps:Data>
      <wps:BoundingBoxData ows:crs="EPSG:4326" >
        <ows:LowerCorner>-122.420041 37.612468</ows:LowerCorner>
        <ows:UpperCorner>-122.167013 37.822128</ows:UpperCorner>
      </wps:BoundingBoxData>
    </wps:Data>
  </wps:Input>
  <wps:Input>

```

```

    <ows:Identifier>displayStyle</ows:Identifier>
    <ows:Title>Display Style</ows:Title>
    <wps:Data>
      <wps:LiteralData>redGreen</wps:LiteralData>
    </wps:Data>
  </wps:Input>
</wps>DataInputs>  <wps:ResponseForm>
  <wps:RawDataOutput mimeType="image/png">
    <ows:Identifier>image</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

#### 7.1.4. Route View Shed

```

<?xml version="1.0" encoding="utf-8"?>
<wps:Execute service="WPS" version="1.0.0" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" >
  <ows:Identifier>${WPS_Route_Service_Identifier}</ows:Identifier>
  <wps>DataInputs>
    <wps:Input>
      <ows:Identifier>source</ows:Identifier>
      <ows:Title>Data Source</ows:Title>
      <wps:Data>
        <wps:LiteralData>britsouth</wps:LiteralData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>routePoints</ows:Identifier>
      <ows:Title>Route Points</ows:Title>
      <wps:Data>
        <wps:LiteralData>&lt;gml:LineString&gt; &lt;gml:posList&gt; 51.1244288 -
1.8908794 51.1244876 -1.8914126 51.1245858 -1.8921404 51.1245918 -
1.892175&lt;/gml:posList&gt; &lt;/gml:LineString&gt;</wps:LiteralData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>viewShedPointSpacing</ows:Identifier>
      <ows:Title>View Shed Point Spacing</ows:Title>
      <wps:Data>
        <wps:LiteralData>300</wps:LiteralData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>view_height</ows:Identifier>
      <ows:Title>Viewing Height</ows:Title>
      <wps:Data>

```

```

    <wps:LiteralData>0</wps:LiteralData>
  </wps>Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>view_hType</ows:Identifier>
  <ows:Title>Viewing Height Type</ows:Title>
  <wps>Data>
    <wps:LiteralData>groundHeight</wps:LiteralData>
  </wps>Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>view_maxRadius</ows:Identifier>
  <ows:Title>Viewing Maximum Radius</ows:Title>
  <wps>Data>
    <wps:LiteralData>3000</wps:LiteralData>
  </wps>Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>requiredDisplayWidth</ows:Identifier>
  <ows:Title>Required Display Width</ows:Title>
  <wps>Data>
    <wps:LiteralData>800</wps:LiteralData>
  </wps>Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>requiredDisplayHeight</ows:Identifier>
  <ows:Title>Required Display Height</ows:Title>
  <wps>Data>
    <wps:LiteralData>600</wps:LiteralData>
  </wps>Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>requiredDisplayExtent</ows:Identifier>
  <ows:Title>Required Display Extent</ows:Title>
  <wps>Data>
    <wps:BoundingBoxData ows:crs="EPSG:4326" >
      <ows:LowerCorner>-1.98 51.096</ows:LowerCorner>
      <ows:UpperCorner>-1.7925 51.2125</ows:UpperCorner>
    </wps:BoundingBoxData>
  </wps>Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>displayStyle</ows:Identifier>
  <ows:Title>Display Style</ows:Title>
  <wps>Data>
    <wps:LiteralData>redGreen</wps:LiteralData>

```

```

    </wps:Data>
  </wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="image/png">
    <ows:Identifier>image</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

### 7.1.5. Route Breakdown

```

<?xml version="1.0" encoding="utf-8"?>
<wps:Execute service="WPS" version="1.0.0" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" >
  <ows:Identifier>${WPS_RouteBreakdown_Service_Identifier}</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>routePoints</ows:Identifier>
      <ows:Title>Route Points</ows:Title>
      <wps:Data>
        <wps:LiteralData>&lt;gml:LineString&gt; &lt;gml:posList&gt; 51.1244288 -
1.8908794 51.1244876 -1.8914126 51.1245858 -1.8921404 51.1245918 -
1.892175&lt;/gml:posList&gt; &lt;/gml:LineString&gt;</wps:LiteralData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>viewShedPointSpacing</ows:Identifier>
      <ows:Title>View Shed Point Spacing</ows:Title>
      <wps:Data>
        <wps:LiteralData>300</wps:LiteralData>
      </wps:Data>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput asReference="false" mimeType="text/xml" >
      <ows:Identifier>gmlDoc</ows:Identifier>
    </wps:RawDataOutput>
  </wps:ResponseForm>
</wps:Execute>

```

### 7.1.6. Terrain Profile

```

<?xml version="1.0" encoding="utf-8"?>
<wps:Execute service="WPS" version="1.0.0" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" >
  <ows:Identifier>TerrainProfileWPS</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>

```

```

<ows:Identifier>source</ows:Identifier>
<ows:Title>Data Source</ows:Title>
<wps:Data>
  <wps:LiteralData>britsouth</wps:LiteralData>
</wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>profilePoints</ows:Identifier>
  <ows:Title>Profile Points</ows:Title>
  <wps:Data>
    <wps:LiteralData>&lt;gml:LineString&gt; &lt;gml:posList&gt; 51.1244288 -
1.8908794 51.1244876 -1.8914126 51.1245858 -1.8921404 51.1245918 -1.892175
51.1247118 -1.8928772 51.124728 -1.892973 51.1247406 -1.893031 51.1247762
-1.8931956 51.1248092 -1.8933476 51.1248818 -1.8936022 51.1248852 -1.8936134
51.1249268 -1.8937474 51.1249416 -1.8937954 51.12503 -1.8940084 51.125082
-1.8941212 51.1251134 -1.8941892 51.125116 -1.8941948 51.1251326 -1.894223
51.1251936 -1.8943274 51.1252106 -1.8943562 51.1253438 -1.8945474 51.125419
-1.8946498 51.1254276 -1.8946614 51.1254478 -1.8946888 51.1256586 -1.8949432
51.1258912 -1.895222 51.1259956 -1.8953472 51.12614 -1.8955296 51.1262442
-1.8956614 51.1263964 -1.8958544 51.1266578 -1.896192 51.1268586 -1.8965088
51.1268666 -1.8965214 51.1268742 -1.8965354 51.1269822 -1.8967344 51.1270778
-1.896944 51.1272112 -1.897212 51.1273004 -1.8973834 51.1274032 -1.8975514
51.1275492 -1.8977826 51.1276584 -1.8979274 51.1277392 -1.898022 51.127824
-1.8981002 51.1280224 -1.898268 51.1282762 -1.898464 51.1284588 -1.8985836
51.1286034 -1.8986632 51.1287272 -1.8987162 51.1288718 -1.8987592 51.1289998
-1.898774 51.1290436 -1.898772 51.1290884 -1.89877 51.1291518 -1.898767
51.1291792 -1.8987602 51.1292108 -1.8987522 51.1292786 -1.898735 51.1294754
-1.8986568 51.1295982 -1.8986078 51.129833 -1.8985172 51.1300374 -1.8984566
51.1300876 -1.8984418 51.1304084 -1.8983778 51.130617 -1.898349 51.1308706
-1.898325 51.1310268 -1.8983264 51.1312532 -1.898356 51.1314168 -1.8983938
51.131572 -1.8984436 51.131767 -1.898523 51.131872 -1.8985812 51.1320218 -
1.8985624 51.1320952 -1.8985606 51.1321928 -1.8985472 51.1322766 -1.8984786
51.132437 -1.898395 51.1325562 -1.8983314 51.13277 -1.8981944 51.1329388 -
1.8981222 51.133052 -1.8981138 51.1331966 -1.89815 51.1332836 -1.8982216
51.133419 -1.8983562 51.1335554 -1.8985192 51.133695 -
1.8986754&lt;/gml:posList&gt; &lt;/gml:LineString&gt;</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>viewShedPointSpacing</ows:Identifier>
  <ows:Title>View Shed Point Spacing</ows:Title>
  <wps:Data>
    <wps:LiteralData>300</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>lowestDisplayHeight</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>-100</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>highestDisplayHeight</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>50</wps:LiteralData>
  </wps:Data>

```

```
</wps:Input>
<wps:Input>
  <ows:Identifier>requiredDisplayWidth</ows:Identifier>
  <ows:Title>Required Display Width</ows:Title>
  <wps:Data>
    <wps:LiteralData>1600</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>requiredDisplayHeight</ows:Identifier>
  <ows:Title>Required Display Height</ows:Title>
  <wps:Data>
    <wps:LiteralData>600</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>displayStyle</ows:Identifier>
  <ows:Title>Display Style</ows:Title>
  <wps:Data>
    <wps:LiteralData>redGreen</wps:LiteralData>
  </wps:Data>
</wps:Input>
</wps>DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="image/png">
    <ows:Identifier>image</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>
```