



Bentley[®]
Sustaining Infrastructure

AWLRS v1.1 Administration Guide

**25 Jun 2019
Issue: 1.1**

Submitted by:
Bentley Systems (UK) Ltd.,
9th Floor,
20 Gracechurch Street,
London
EC3V 0BG

Document Version History

Version	Owner	Update Source	Date	Description
0.1	Barbara ODriscoll		7-Feb-2019	Initial Draft (based on 1.0.1)
0.2	Barbara ODriscoll		11-Feb-2019	Added NSG.
1.0	Barbara ODriscoll		11-Feb-2019	Final amendments added.
1.1	Mike Huitson		25-Jun-2019	Added section on External Links
For review:				For approval:
Rob May				Pano Voudouris
Mike Huitson				

Table of Contents

1	Introduction.....	2
1.1	Purpose.....	2
1.2	Overview	2
1.3	Dependencies	2
2	Product Options.....	3
3	Map Configuration.....	5
3.1	Defining Background Layers	5
3.2	Defining Map Layers	7
3.3	Custom Tags	8
3.3.1	DisplayedAtStartup.....	9
3.3.2	LegendGroup	9
3.3.3	GeometryTypes	10
3.4	Styling Restrictions.....	13
3.4.1	Color	13
3.4.2	Text.....	13
3.4.3	Line.....	16
3.4.4	Marker.....	19
3.4.5	Advanced.....	20
3.4.6	Theme Styling Rules.....	21
3.4.7	Base Map Theme Properties	22
3.5	Editing zoom level threshold.....	22
3.6	File Based Map Layers.....	23
4	User Interface Configuration.....	25
4.1	Quick Search.....	25
4.1.1	Network Themes	25
4.1.2	exor Asset Themes.....	25
4.1.3	Foreign Table Asset Themes.....	26
4.1.4	Column Priority	27
4.2	External Links.....	28
4.2.1	Examples.....	30
4.3	Modification of deployment options	38
5	Copy/Trace and Attribute Transfer configuration.....	39
6	Street Gazetteer Manager	42
6.1	Enabling NSG	42
6.2	Product Options.....	44
6.3	Custom Map Layers	45
6.4	NSG Configuration.....	46
6.5	Quick Search Configuration.....	48

1 Introduction

1.1 Purpose

This guide is intended for those who will administer the AWLRS application, whether during initial implementation, operational use or subsequent re-configuration.

1.2 Overview

Assetwise ALIM Linear Referencing Services (AWLRS) is an application that allows an organization to manage the network editing and location of their network.

1.3 Dependencies

Exor Core

AWLRS makes extensive use of the Exor Core database & APIs. Network data is referenced from Exor Network Manager. Asset data and spatial geometries are managed using Exor Asset Manager.

Much of the metadata is referenced from Exor Core tables and maintained using Exor Core forms.

Please consult the Exor Core Administration Guide for detailed information regarding the Exor Core forms identified in this document.

NB. Exor Core 4700 and all fixes of Exor Core must have been applied to the schema.

2 Product Options

The following product options should be defined using the exor core "Product Options" form (HIG9310) in the Exor Core application.

Product Options - TRANSINFO@oregon01.WARDEVDB1AJ HIG v4.7.0.1

Product	Option Id*	Name*	Current Value
AWLRS	AWLMAPDEBUG	Mapserver Debug	N
AWLRS	AWLMAPEPSG	AWLRS Map Output Projection	2992
AWLRS	AWLMAPNAME	AWLRS Map Name	AWLRS_MAP
AWLRS	AWLMAPSRID	AWLRS Map Default SRID	2992
AWLRS	AWLRECALHS	AWLRS Recalibrate With History	Y
AWLRS	AWLUIDBUG	UI Debug	Y

Usage Remarks*

When set to Y the generated map file will contain the MS_ERRORFILE paramater that causes a log file to be generated by Mapserver.

Option Id	Description	Default Value
AWLMAPDEBUG	When set to Y the generated map file will contain the MS_ERRORFILE parameter that causes a log file to be generated by MapServer.	N
AWLMAPEPSG	AWLRS Map Output Projection	<PLEASE SET>
AWLMAPNAME	AWLRS Map Name	AWLRS_MAP
AWLMAPSRID	AWLRS Map Default SRID, this value will only be used when the data in the base table of a theme does not have a SRID set in the sdo metadata.	<PLEASE SET>
AWLUIDBUG	When set to Y the full error stack, backtrace and call stack will be returned to the UI for any handled exceptions.	N
AWLRECALHS	When recalibrating a Datum in AWLRS the user has the option to maintain history, this option defines the default position of the switch when the dialog is displayed, when set to Y the switch will default to On otherwise it will default to Off.	Y

Option Id	Description	Default Value
AWLRECALPG	When performing certain network operations, such as Split, Merge and Reshape, on a Datum in AWLRS the user has the option to Rescale all parent groups at the end of the operation, this option defines the default position of the switch when the dialog is displayed, when set to Y the switch will default to On otherwise it will default to Off.	N
AWLPTHMINB	The minimum size (in Metres) of the buffer around the mbr of the two points used to generate a path when building a list of potential candidate network elements.	1000
AWLPTHPERC	The percentage by which to increase the mbr of the two points used to generate a path when building a list of potential candidate network elements.	20

3 Map Configuration

3.1 Defining Background Layers

AWLRS can be configured to use public OSM, WMS or ArcGISRest tile services to be used as background layers. These layers can be configured in the “[AWLRS Web Api]\Scripts\configs\MapBaselayers.json” file on the application server, for example C:\inetpub\wwwroot\AWLRSWebAPI\scripts\configs\MapBaselayers.json

Each entry in the file has following properties:

- **Name:** The Name to be displayed in the map legend.
- **Type:** Should be one of the following
 - “OSM” – OSM tile server.
 - “WMS” – OGC compliant WMS server.
 - “ArcGISRest” – ArcGIS Tile server.
- **Url:** The URL of the service, this may be an array of URLs in the case of a tile server. The service will be called from within the client browser so the URL should be publicly accessible.
- **Parameters:** This property only applies to WMS services and allows the following parameters to be added to the WMS request issued by the client browser:
 - “LAYERS” – A list of layers to be included in the image returned by the service.
 - “BGCOLOR” – The background color to be used by the service when rendering the requested image.
 - “FORMAT” – The format of the image returned, generally this should be “image/png” unless the service does not support this format.
- **Options:**
 - “Attribution” Any acknowledgment or Copyright message that the service provider requires to be displayed in the map.
- **IsActive:**
 - **true** – the layer is displayed in the map legend
 - **false** – the layer is not displayed in the map legend
- **IsShown:**
 - **true** – the layer is displayed at startup
 - **false** – the layer is not displayed at startup
- **MinimumScale:** integer value, the minimum scale at which the base layer is enabled.
- **MaximumScale:** integer value, the maximum scale to which the base layer is enabled. Should be set to 0 if there is no restriction in maximum scale.
- **Projection:** The projection to be requested from the Tile\WMS server.

NB. If AWLRS is deployed to several load balanced servers the contents of this file must be consistent across ALL of them.

The following sample file contains an example of each type of service, **please note that the services used here are for demonstration purposes only and should not be used for any other reason without permission from the service provider.**

```
[
  {
    "Name": "Open Street Map",
    "Type": "OSM",
    "Url": "http://{a-c}.tile.openstreetmap.org/{z}/{x}/{y}.png",
    "Options": {
      "Attribution": "© <a href=\"http://www.openstreetmap.org/copyright\"
target=\"_blank\">OpenStreetMap</a> contributors"
    }, 1
    "IsActive": true,
    "IsShown": true,
    "MinimumScale": 0,
    "MaximumScale": 0,
    "Projection": "EPSG:3857",
  },
  {
    "Name": "OSM-WMS Uni Heidelberg",
    "Type": "WMS",
    "Url": "http://129.206.228.72/cached/osm/service",
    "Parameters": {
      "Layers": ["osm_auto:all"],
      "Format": "image/png",
      "BgColor": "0xA6CAF0"
    },
    "Options": {
      "Attribution": "© <a href=\"http://www.openstreetmap.org/copyright\"
target=\"_blank\">OpenStreetMap</a> contributors",
    },
    "IsActive": true,
    "IsShown": false,
    "MinimumScale": 0,
    "MaximumScale": 0,
    "Projection": "EPSG:3857"
  },
  {
    "Name": "USA ArcGISRestTile",
    "Type": "ArcGISRest",
    "Url": "https://sampleserver6.arcgisonline.com/arcgis/rest/services/USA/MapServer",
    "Options": {
      "Attribution": "This is a sample service hosted by ESRI, powered by ArcGIS Server."
    },
    "IsActive": true,
    "IsShown": false,
    "MinimumScale": 0,
    "MaximumScale": 0,
    "Projection": "EPSG:4326"
  }
]
```


3.2 Defining Map Layers

AWLRS makes use of the standard exor GIS Themes functionality to expose data for mapping purposes.

As with the exor forms application Oracle Map Builder can be used to define a Base Map for the AWLRS application. The AWLRS Base Map can have any name, the Product Option "AWLMAPNAME" is used by the application to identify the AWLRS Base Map, the default value of this Product Option is "AWLRS_MAP" if a different name is used the value of the Product Option should be updated.

NB: Any image files used as markers need to be uploaded to the Application Server(s) using the Write Map Files function therefore any changes to the marker image file or inclusion of themes that make use of an image file not previously used in the map will require use of the Write Map Files function to apply the changes.

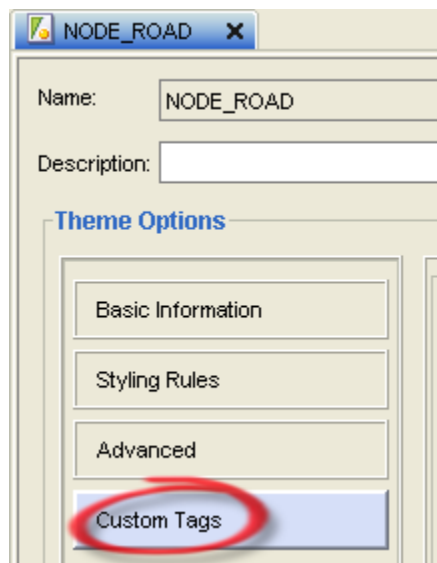
NB. Please refer to the exor core document "Spatial Server Admin" particularly the section on "Configuring Maps with Oracle MapBuilder Tool" for details on how to create a Base Map for AWLRS. Please also refer to the Styling Restrictions section below for details of the layer styling options that are supported by the AWLRS map.

NB. Please also be aware that the names of any themes intended to be used in the AWLRS map should not contain any special characters (except for underscore "_") or begin with a number i.e. the name should only contain the letters A to Z, an underscore or numeric characters 0 to 9. (Theme names are Upper Case)

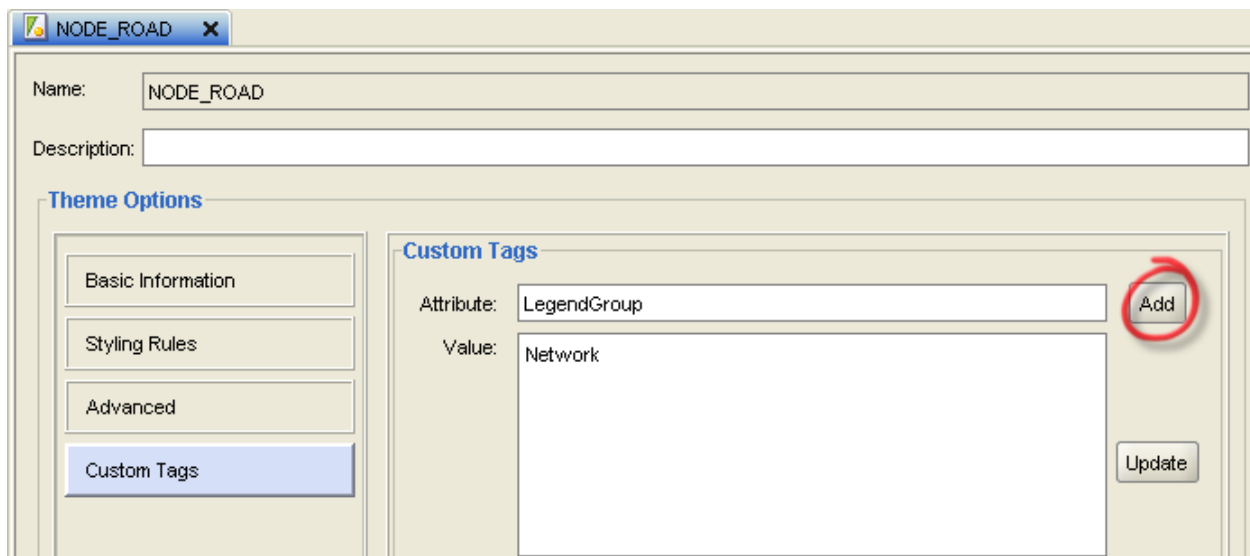
3.3 Custom Tags

The AWLRS map makes use of the Map Builder "Custom Tags" functionality to allow several additional items of metadata to be specified for a Theme.

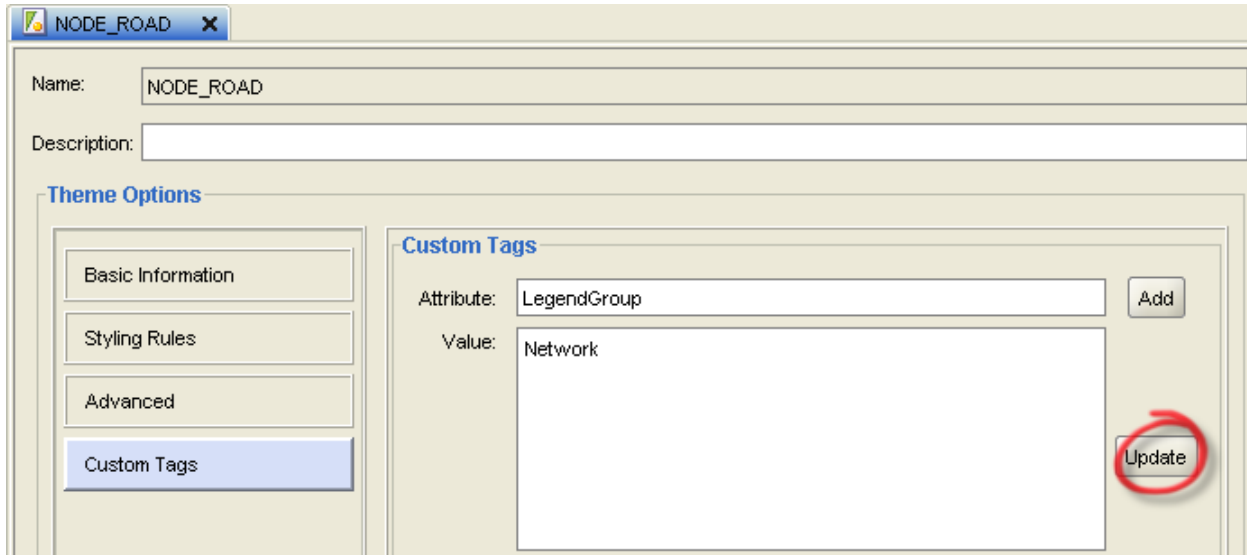
To add a Custom Tag to a theme, open the theme in Map Builder and click on "Custom Tags" in the "Theme Options" panel:



Enter the name of the Custom Tag in the "Attribute:" field and the value in the "Value:" field then click the "Add" button and save the change.



To edit the value of an existing Custom Tag, click on it in the list, change the "Value:" field then click on the "Update" button and save the change.



3.3.1 DisplayedAtStartup

This Custom Tag can be used to specify whether a theme should be displayed when a user first logs into the application.

Attribute: DisplayedAtStartup

Valid Values: "Y" or "N"

If this Custom Tag is not present for a theme its value will be defaulted to "Y" so it only needs to be added to a theme if the value should be "N".

3.3.2 LegendGroup

The map legend in the AWLRS application has the group themes in the Catalog tab.

To place a theme within a group the LegendGroup custom tag should be added to the theme, a group will be created using the value given in the tag and the theme will be added to it.

Where multiple themes have the LegendGroup tag defined with the same value a single group will be created containing all the associated themes.

Attribute: LegendGroup

Value: The name of the group

If this Custom Tag is not present for a theme it will be presented in the Catalog tab in an unnamed group along with any other themes that do not have it.

3.3.3 GeometryTypes

Layers based upon Foreign Table Asset Types or Spatial Tables that are not maintained by the system may contain multiple geometry types. Like many GIS systems Mapserver does not support mixed geometry types within a single layer.

There are a number of ways around this, for example a view could be created for each geometry type with a theme for each view, however this can be confusing for the user and would mean that all the themes would need to be enabled and selectable in the map legend for the user to make a selection based upon a map extent and see the correct outcome regardless of geometry type.

The **GeometryTypes** custom tag has been created to allow the administrator to identify the different geometry types that may be present within a Layer's feature table, this triggers AWLRS to internally create a separate layer for each geometry type identified however the user will only see 1 layer in the legend and will be able to work with all of the layer's data in the same way as any other layer.

Whilst it should not be forgotten that it may be possible to add geometry types that don't currently exist in the table the following SQL query can be used to determine what geometry types currently exist within a spatial table:

```
SELECT S.GEoloc.SDO_GTYPE
       , COUNT (*)
FROM V_NM_NSG_ASD_TP61_SDO S
GROUP
  BY S.GEoloc.SDO_GTYPE
;
```

In this example, an NSG ASD Theme, the results could be something like this:

GEOLOC.SDO_GTYPE	COUNT(*)
2001	7
2006	2
2007	3
2003	101
2002	111107

The SDO_GTYPE value is 4 digits in the format *d/lt*, where:

- *d* identifies the number of dimensions (2, 3, or 4)
- */* identifies the linear referencing measure dimension for a three-dimensional linear referencing system (LRS) geometry, that is, which dimension (3 or 4) contains the measure value. For a non-LRS geometry, or to accept the Spatial default of the last dimension as the measure for an LRS geometry, specify 0.
- *tt* identifies the geometry type (00 through 07, with 08 through 99 reserved for future use).

The following table shows the valid SDO_GTYPE values.

GType Code	GType Descr	Description	Mapserver Layer Type
<i>d/00</i>	UNKNOWN_GEOMETRY	Spatial ignores this geometry.	Not Supported in AWLRS
<i>d/01</i>	POINT	Geometry contains one point.	POINT
<i>d/02</i>	LINE or CURVE	Geometry contains one line string that can contain straight or circular arc segments, or both. (LINE and CURVE are synonymous in this context.)	LINE
<i>d/03</i>	POLYGON	Geometry contains one polygon with or without holes.	POLYGON
<i>d/04</i>	COLLECTION	Geometry is a heterogeneous collection of elements. COLLECTION is a superset that includes all other types.	Not Supported in AWLRS
<i>d/05</i>	MULTIPOINT	Geometry has one or more points. (MULTIPOINT is a superset of POINT.)	POINT
<i>d/06</i>	MULTILINE or MULTICURVE	Geometry has one or more line strings. (MULTILINE and MULTICURVE are synonymous in this context, and each is a superset of both LINE and CURVE.)	LINE
<i>d/07</i>	MULTIPOLYGON	Geometry can have multiple, disjoint polygons (more than one exterior boundary). (MULTIPOLYGON is a superset of POLYGON.)	POLYGON

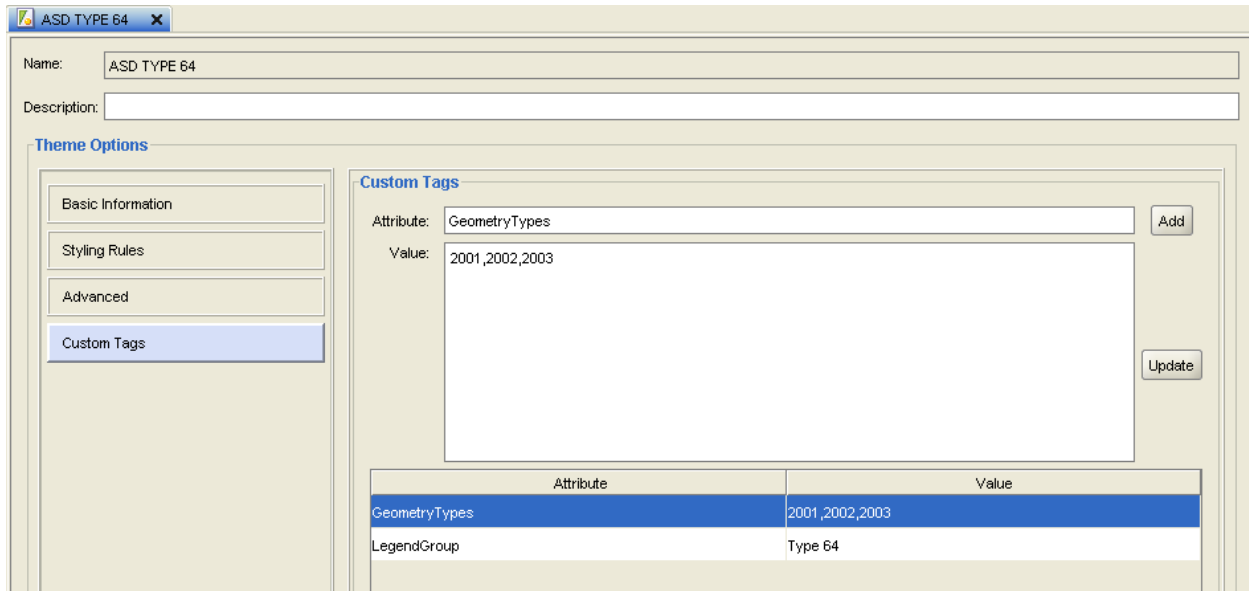
So, in our example we have:

- 2003 and 2007 that both equate to POLYGON in Mapserver.
- 2002 and 2006 that both equate to LINE in Mapserver.
- 2001 that equates to POINT in Mapserver.

It is not necessary to enter all the sdo_gtype codes as values for the **GeometryTypes** tag, 1 for each Mapserver equivalent is enough so in this example the following value, which equates to POINT, LINE and POLYGON, would suffice:

Attribute: GeometryTypes

Value: 2001,2002,2003



The screenshot shows a web-based configuration window titled "ASD TYPE 64". It has a "Name" field with the value "ASD TYPE 64" and an empty "Description" field. On the left, under "Theme Options", there are four tabs: "Basic Information", "Styling Rules", "Advanced", and "Custom Tags" (which is selected). The "Custom Tags" section contains an "Attribute" field with the value "GeometryTypes" and a "Value" field with the value "2001,2002,2003". There are "Add" and "Update" buttons to the right of these fields. Below the input fields is a table with two columns: "Attribute" and "Value".

Attribute	Value
GeometryTypes	2001,2002,2003
LegendGroup	Type 64

3.4 Styling Restrictions

The AWLRS application uses a combination of an OpenLayers Client and MapServer to deliver the map functionality. The styling used in the map can be defined using Oracle Map Builder however as the style data stored by Map Builder is extracted as an SLD (Styled Layer Descriptor) file for use by MapServer there are some types of style that can be created in Map Builder that are not supported by SLD.

Oracle Map Builder allows styles of the following types to be defined and associated with a Theme:

- Area
- Color
- Line
- Marker
- Text
- Advanced

The AWLRS map does not support the use of Area Styles.

Whilst Map Builder allows the specification of multiple styles in the "Styling Rules" of a Theme the AWLRS map only supports the use of a single style for each Theme.

3.4.1 Color

The use of Color Styles for rendering **Polygons** is fully supported.

However, unlike the Oracle Mapviewer client, used in the exor forms application, themes that contain line or multiline geometries will not be rendered in the AWLRS Map if a Color Style is used.

3.4.2 Text

The use of Text Styles within the "Styling Rules" of a theme for Labeling is supported, any other use of Text Styles is not supported.

The following Text Style Options are **not** supported:

- "Sticky"
- Multiline
- Oriented Point
- Path Labeling

When specifying the font size or the width of a Halo only the "px" unit is supported.

The following "Text" options are **not** supported:

- Offset X
- Offset Y
- Extra letter spacing
- Decoration

T.BENCHLABEL

Name: T.BENCHLABEL

Description:

Style Options

Text

☒ Halo

☐ Sticky

☐ Multiline

☐ Oriented Point

☐ Path Labeling

Text

Font: Arial

Size: 12 px

Style: ☒ Italic ☒ Bold

Color: Hex: #0000FF

Opacity: 100%

Offset X: px

Offset Y: px

Extra letter spacing: px

Decoration: NONE

Editor XML

T.BENCHLABEL

Name: T.BENCHLABEL

Description:

Style Options

Text

☒ Halo

☐ Sticky

☐ Multiline

☐ Oriented Point

☐ Path Labeling

Halo

Width: 3 px

Color: Hex: #FF00FF

Opacity: 100%

Editor XML

When selecting the Font for a text style the "Local Fonts" list should be used:



The selected Font should be present on the application server upon which MapServer is installed and the path to the Font file should be included in the MapServer FontSet file which can be found in the MapServer cgi-bin directory:

<MapServer Folder>\cgi-bin\fonts.list

This is a text file and can be edited using any text editor such as Notepad.

NB: When adding a Font to the file any spaces in its name should be removed.

Several predefined Fonts are included in the file on installation, depending upon where windows is installed the included paths may need to be updated:

Arial	c:\windows\fonts\arial.ttf
Arial-bold	c:\windows\fonts\arialbd.ttf
Arial-italic	c:\windows\fonts\ariali.ttf
Arial-bold-italic	c:\windows\fonts\arialbi.ttf
ArialBlack	c:\windows\fonts\ariblk.ttf
ComicsansMS	c:\windows\fonts\comic.ttf
ComicsansMS-bold	c:\windows\fonts\comicbd.ttf
CourierNew	c:\windows\fonts\cour.ttf
CourierNew-bold	c:\windows\fonts\courbd.ttf
CourierNew-italic	c:\windows\fonts\courl.ttf
CourierNew-bold-italic	c:\windows\fonts\courbi.ttf
Georgia	c:\windows\fonts\georgia.ttf
Georgia-bold	c:\windows\fonts\georgiab.ttf
Georgia-italic	c:\windows\fonts\georgiai.ttf
Georgia-bold-italic	c:\windows\fonts\georgiaz.ttf
Impact	c:\windows\fonts\impact.ttf
MonotypeCorsiva	c:\windows\fonts\mtcorsva.ttf
TimesNewRoman	c:\windows\fonts\times.ttf
TimesNewRoman-bold	c:\windows\fonts\timesbd.ttf
TimesNewRoman-italic	c:\windows\fonts\timesi.ttf
TimesNewRoman-bold-italic	c:\windows\fonts\timesbi.ttf
TrebuchetMS	c:\windows\fonts\trebuc.ttf
TrebuchetMS-bold	c:\windows\fonts\trebucbd.ttf
TrebuchetMS-italic	c:\windows\fonts\trebucit.ttf
TrebuchetMS-bold-italic	c:\windows\fonts\trebucbi.ttf
Verdana	c:\windows\fonts\verdana.ttf
Verdana-bold	c:\windows\fonts\verdanab.ttf
Verdana-italic	c:\windows\fonts\verdanai.ttf
Verdana-bold-italic	c:\windows\fonts\verdanaz.ttf

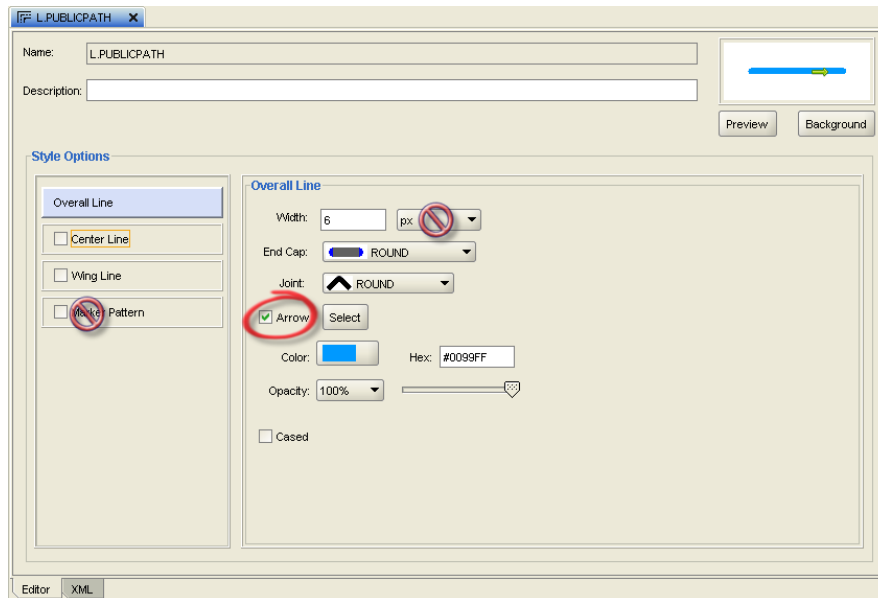
3.4.3 Line

The use of Line Styles is supported for rendering **Lines**.

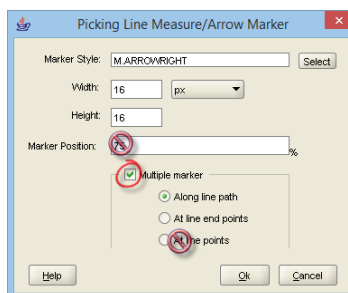
There are however some features of the Map Builder line styles that are not supported:


Marker Patterns are not supported at all.

Only “px” is supported when specifying the Width or a Dash Pattern.



In the "Overall Line" options there are some restriction to be noted with regards to the use of **Arrows**.



The marker used should be an arrow that points from left to right e.g. , markers that point in other directions will not correctly indicate the cardinality of the geometry.

Marker Position is not used and a single marker cannot be defined.

“Along line path” will result in a line with markers repeated at regular intervals:



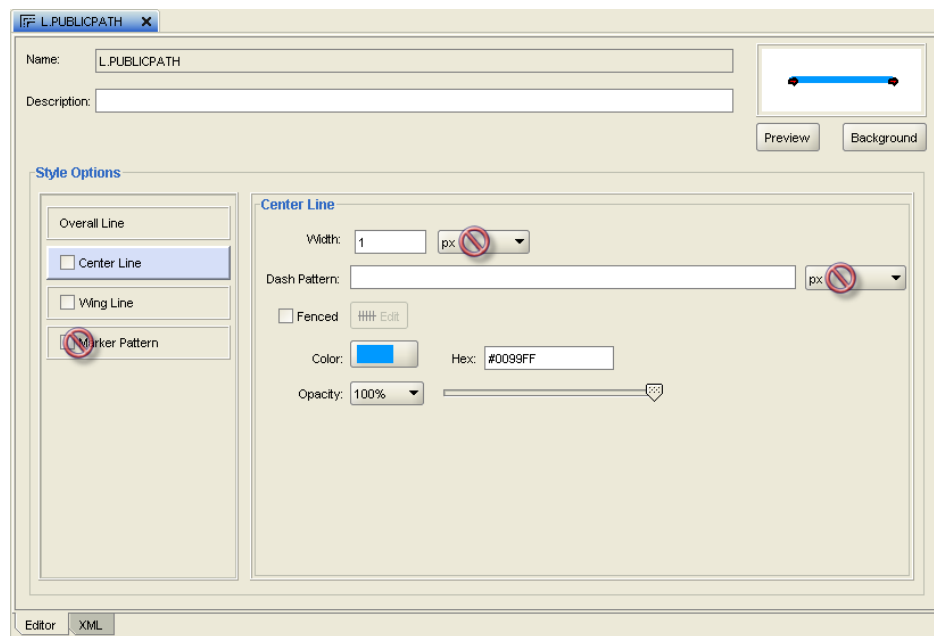
“At line end points” will result in a line with a marker at the start and the end of the line.



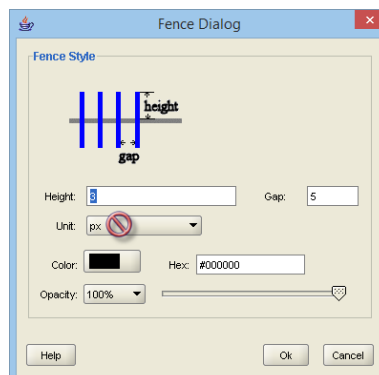
“At line points” is **not supported**, if specified the markers will be drawn on the vertices of the geometry however **their direction will be unreliable**.



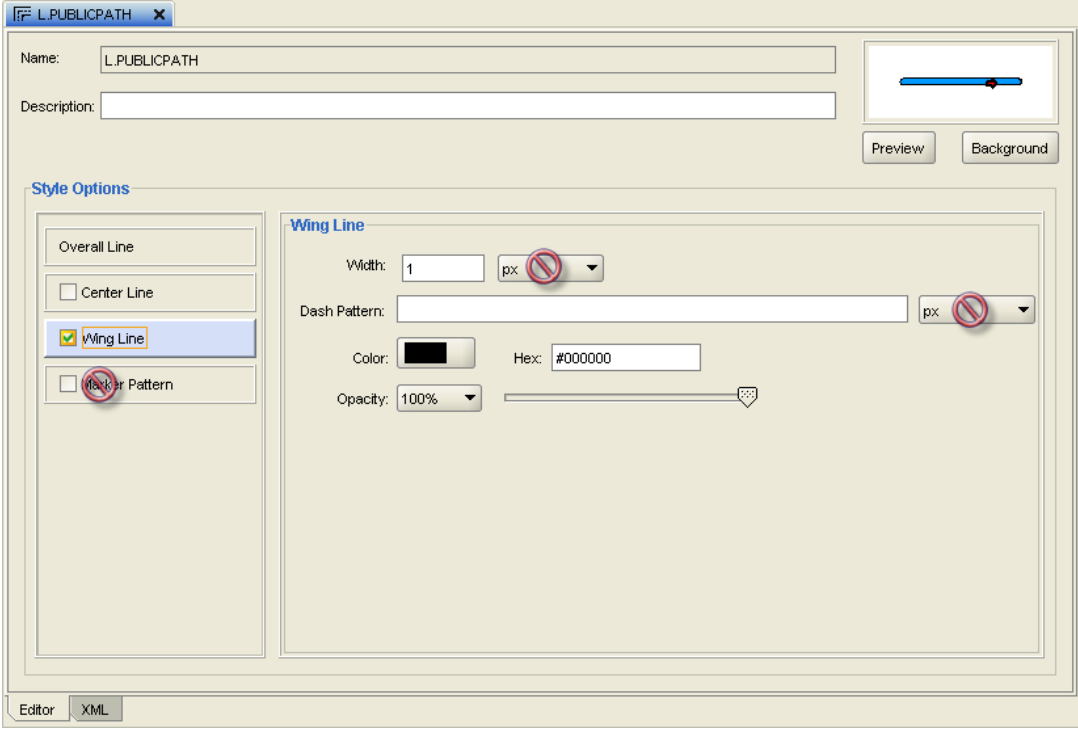
In the "Center Line" options only "px" is supported when specifying the Width or a "Dash Pattern".



Only "px" is supported when specifying the Unit in the edit “Fence” dialog.



In the "Wing Line" options only "px" is supported when specifying the Width or a "Dash Pattern".



The screenshot shows the Bentley Map Configuration dialog box for the layer L.PUBLICPATH. The 'Style Options' section is active, and the 'Wing Line' option is selected. The 'Wing Line' section includes the following fields:

- Width: 1 px (The 'px' dropdown is disabled with a red 'X' mark.)
- Dash Pattern: (The 'px' dropdown is disabled with a red 'X' mark.)
- Color: (Color picker set to black)
- Hex: #000000
- Opacity: 100% (Slider set to 100%)

The 'Overall Line' section on the left has the following options:

- Center Line: ☐
- Wing Line: ☒
- Dash Pattern: ☐ (disabled with a red 'X' mark)

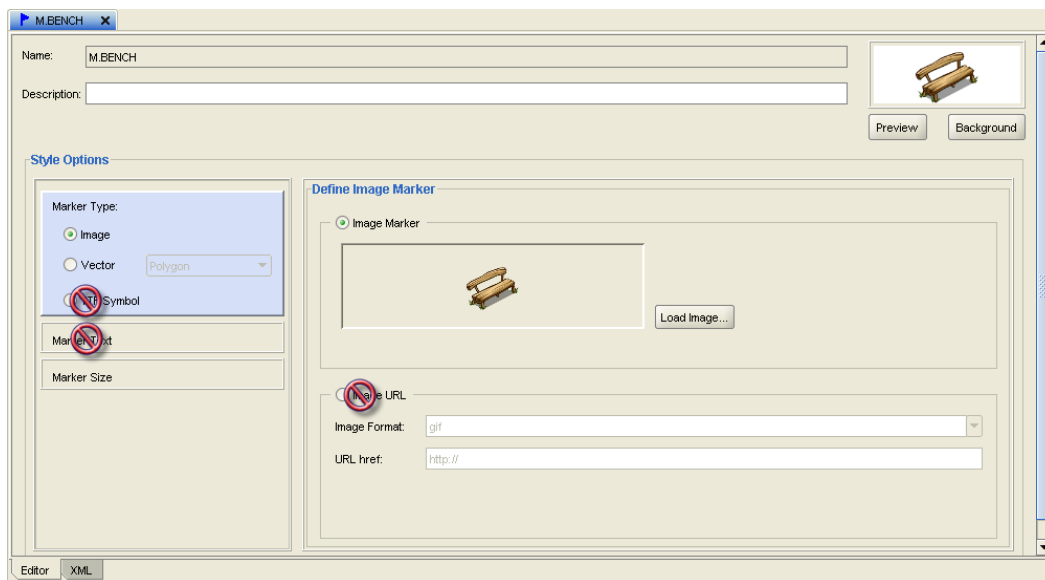
The 'Preview' and 'Background' buttons are located in the top right corner. The 'Editor' and 'XML' tabs are at the bottom left.

3.4.4 Marker

The use of Marker Styles is supported for rendering Points.

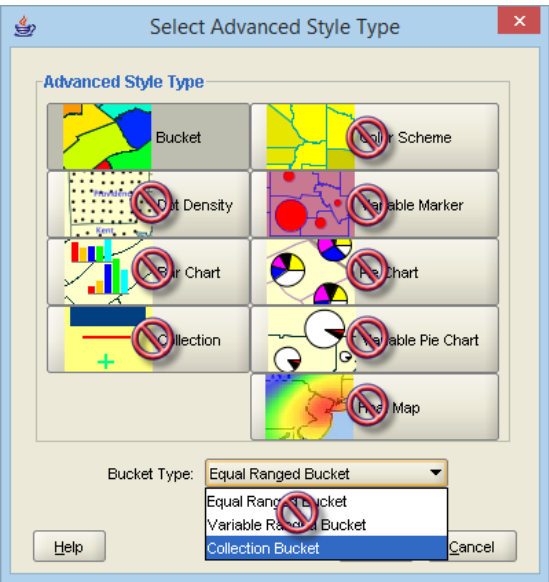
There are however some features of the Map Builder Marker styles that are not supported:

- Marker Text is not supported at all.
- The specification of an image via a URL is not supported.
- The "TTF Symbol" marker type is not supported.



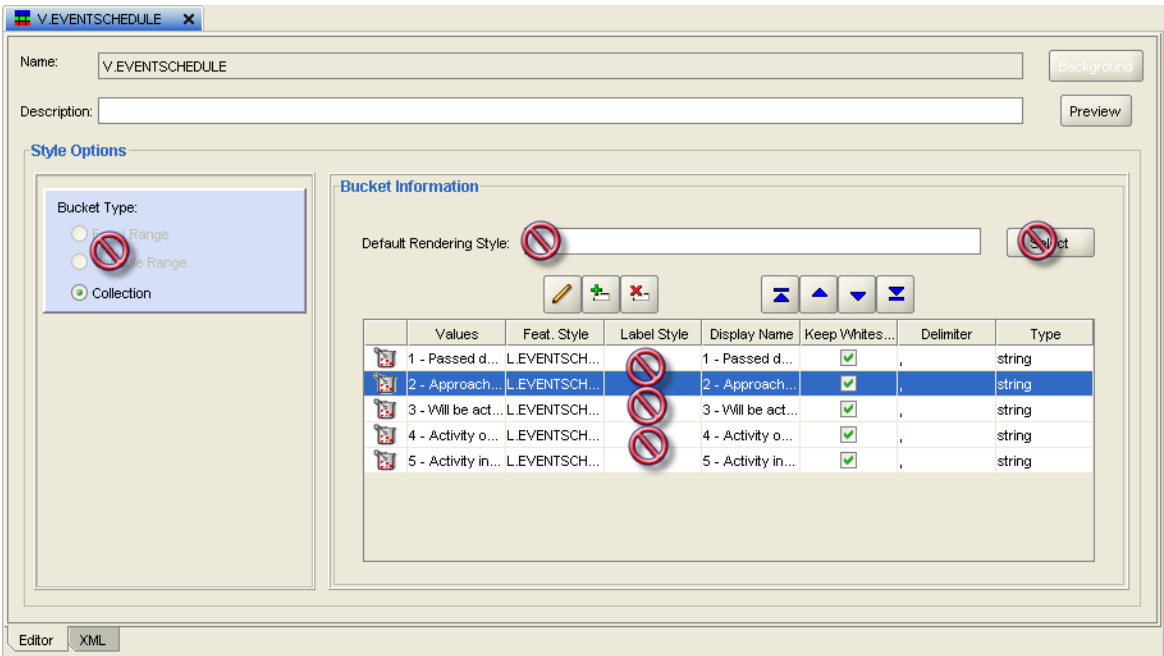
3.4.5 Advanced

The AWLRS Map currently only supports "Collection Bucket" Advanced styles:



For "Collection Bucket" Advanced styles the following options are not supported:

- Default Rendering Style.
- Label Style (A single Label style can be setup on the Theme).



3.4.6 Theme Styling Rules

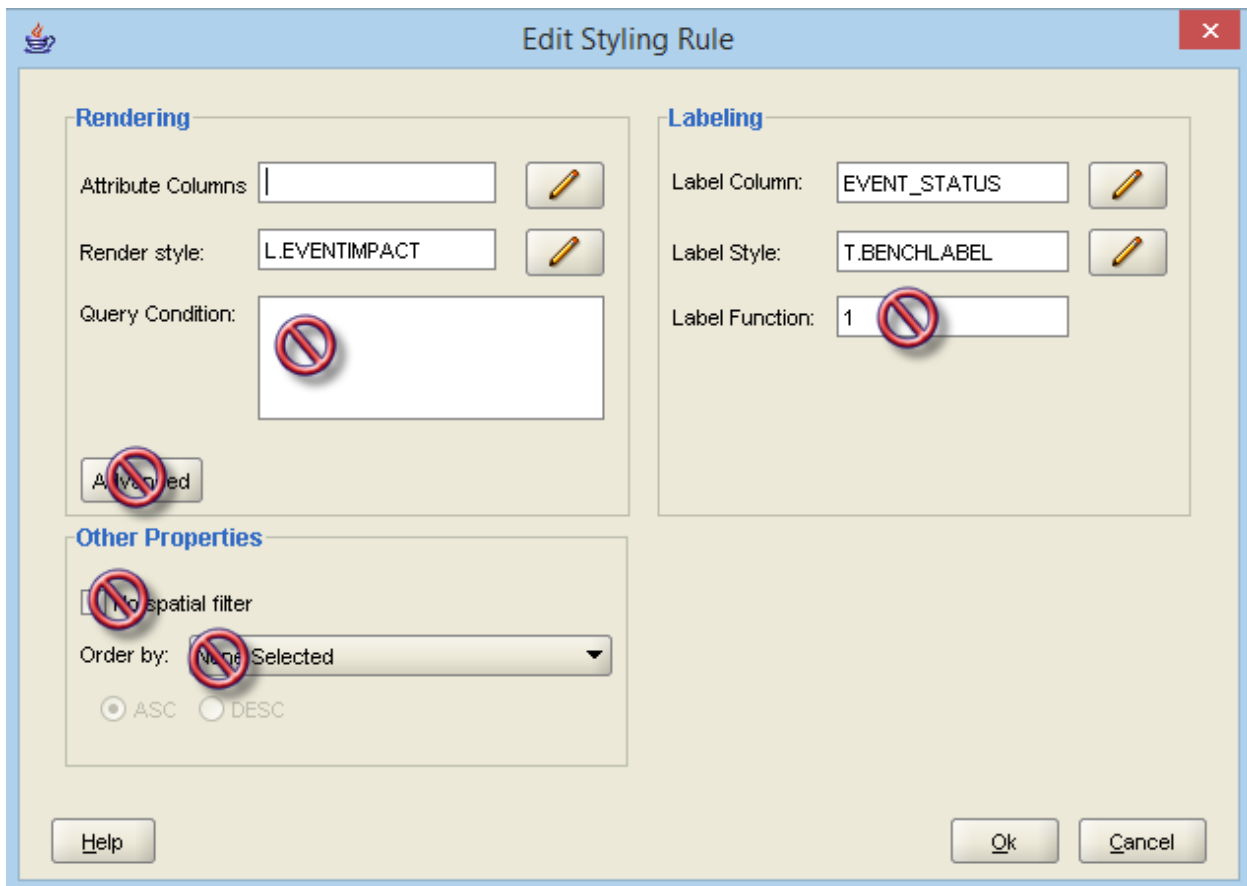
The AWLRS Map does not support the following options in the "Edit Styling Rule" dialog:

- Query Condition (Rendering).
- Advanced (Rendering).
- No spatial filter (Other Properties).
- Order by (Other Properties).

Whilst Map Builder allows the specification of multiple styles in the "Styling Rules" of a Theme the AWLRS map only supports the use of a single style for each Theme.

Label Style - Only Text Styles are supported.

Label Function - The AWLRS map does not support the use of a Function in this field, values of 0 or less result in the label not being displayed, values of 1 or higher result in the label being displayed.



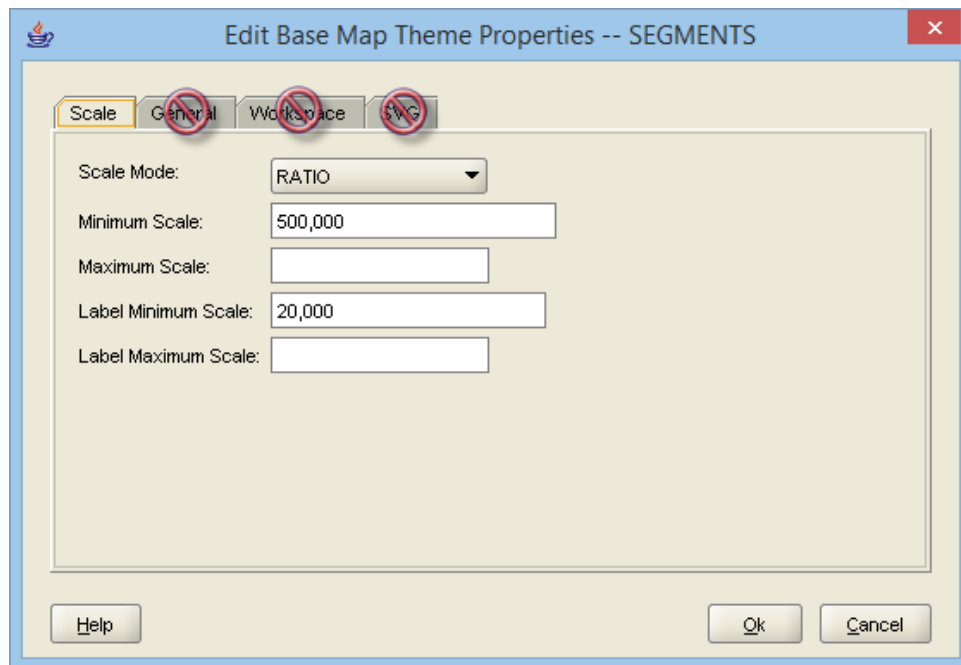
The screenshot shows the "Edit Styling Rule" dialog box. It has a title bar with a close button. The dialog is divided into three main sections: "Rendering", "Labeling", and "Other Properties".

- Rendering Section:**
 - "Attribute Columns": A text box with a pencil icon.
 - "Render style": A text box containing "L.EVENTIMPACT" with a pencil icon.
 - "Query Condition": A large text box with a red prohibition icon.
 - "Advanced": A button with a red prohibition icon.
- Labeling Section:**
 - "Label Column": A text box containing "EVENT_STATUS" with a pencil icon.
 - "Label Style": A text box containing "T.BENCHLABEL" with a pencil icon.
 - "Label Function": A text box containing "1" with a red prohibition icon.
- Other Properties Section:**
 - "No spatial filter": A checkbox with a red prohibition icon.
 - "Order by": A dropdown menu showing "None Selected" with a red prohibition icon.
 - "ASC" and "DESC": Radio buttons.

At the bottom of the dialog are three buttons: "Help", "Ok", and "Cancel".

3.4.7 Base Map Theme Properties

The AWLRS Map supports the Minimum and Maximum Scale properties that can be set for a Theme and its Labels in the Base Map however the other properties that can be defined in the "Edit Base Map Theme Properties" dialog are not supported.



3.5 Editing zoom level threshold

To aid in performance, the editing of network layer will only occur at a map scale less than or equal to that specified in the configuration file [AWLRS deployment]\app\config\config.json.

The option 'editzoomthreshold' is shipped with a default value of 20000, meaning edit will be allowed when the map scale 1:20000 or lower.

3.6 File Based Map Layers

AWLRS can display file-based layers for all users to view, currently the following file formats are supported:

- ESRI Shapefile
- Bentley DGN

NB: The current MapServer documentation states that “DGN files from Microstation versions predating version 8.0 are supported for reading.” This effectively means that DGN files intended for use with AWLRS need to be saved as DGN version 7.

The file to be displayed will need to be uploaded to the application server, it is suggested that the file be placed in folder under the <Mapserver>\cgi-bin to ensure that MapServer has access to it, for example C:\mapserver\cgi-bin\Shapes

The file-based layers are configured on the application server in the file [AWLRS Web API]\app_data\MapCustomLayers.xml, for example:
C:\inetpub\wwwroot\AWLRSWebAPI\app_data\MapCustomLayers.xml

Each layer defined in the file has following properties:

- **Name:** The Name to be displayed in the map legend.
- **Feature Id:** The name of the file attribute that contains the unique key for the feature.
- **Projection:** The projection used in the file.
- **Displayed At Startup:**
 - **true** – the layer is displayed when the user logs in.
 - **false** – the layer is not displayed when the user logs in.
- **Displayed In Legend:**
 - **true** – the layer is displayed in the legend.
 - **false** – the layer is not displayed in the legend (This can be used to remove a file from the map without removing it from the configuration file).
- **Legend Group:** The group in which to display the layer in the Catalog.
- **Type:** The type of geometry stored in the file, valid values are:
 - **LINE**
 - **POINT**
 - **POLYGON**
- **Connection Type:**
 - **NULL for ESRI Shapefiles**
 - **“OGR” for DGN files.**
- **Connection: (DGN Files only)** The path to the file.
- **Data: (ESRI Shapefiles only)** The path to the file.
- **Style:** The style to be used for file formats that do not contain style information, when displaying the geometries in the file, this is provided in MapServer mapfile format, for further details see section 4.1.22 (STYLE) in the MapServer documentation available at <http://mapserver.org/pdf/>

NB. If AWLRS is deployed to several load balanced servers the contents of this file must be consistent across ALL of them. The file containing the data will also need to be uploaded to ALL the servers.

Example File:

```
<?xml version="1.0" encoding="utf-8" ?>
<MapCustomLayers xmlns="http://tempuri.org/MapCustomLayers.xsd">
  <!--<ErrorLogFilePath>C:\MapServer\cgi-bin\error.log</ErrorLogFilePath>-->
  <!--<ShapePath>C:\MapServer\cgi-bin\Shapes</ShapePath>-->
  <Layer name="EXAMPLE SHAPEFILE">
    <featureid>LRS_KEY</featureid>
    <projection>EPSG:2992</projection>
    <displayedatstartup>>false</displayedatstartup>
    <displayedinlegend>>true</displayedinlegend>
    <legendgroup>File Layers</legendgroup>
    <Type>LINE</Type>
    <Data>C:\MapServer\cgi-bin\Shapes\example_shapefile.shp</Data>
    <Style>
      WIDTH 2
      COLOR 128 128 0
    </Style>
  </Layer>
  <Layer name="EXAMPLE DGN FILE">
    <featureid>NE_ID</featureid>
    <projection>EPSG:2992</projection>
    <displayedatstartup>>false</displayedatstartup>
    <displayedinlegend>>true</displayedinlegend>
    <legendgroup>File Layers</legendgroup>
    <Type>LINE</Type>
    <ConnectionType>OGR</ConnectionType>
    <Connection>C:\MapServer\cgi-bin\Shapes\example_dgn_v7.dgn</Connection>
  </Layer>
</MapCustomLayers>
```

Once a layer has been added the next time a user logs in, the layer will be available for display.

4 User Interface Configuration

4.1 Quick Search

By default, Quick Search will operate on specific predefined columns depending upon the theme selected by the user however the columns searched can be overridden by adding rows to the metadata table AWLRS_QUICK_SEARCH_COLUMNS.

NB. Currently there is no UI to maintain the contents of this table so configuration would be a case of using insert/update/delete statements via SQL*Plus or using a tool such as SQL Developer or TOAD.

4.1.1 Network Themes

For themes that represent a Network Type the following columns are search by default:

- NE_UNIQUE
- NE_DESCR

By way of an example, consider a Datum Network Type that is configured for Auto Inclusion into a Parent Group Type, one of the datum attributes is the Unique of the Group that it belongs to, this attribute is stored in the column NE_NUMBER.

It would potentially be useful for quick search to match against the parent group's Unique as well as the datum's Unique and Description.

Inserting the rows shown below into awlrs_quick_search_columns would achieve this:

	AQSC_THEME_NAME	AQSC_COLUMN	AQSC_PRIORITY
▶	SEGMENTS	NE_UNIQUE	1
	SEGMENTS	NE_DESCR	2
	SEGMENTS	NE_NUMBER	3

NB. Any column names added to the awlrs_quick_search_columns table for a network theme must exist on the table nm_elements_all.

4.1.2 exor Asset Themes

For themes that represent an exor Asset Type the following columns are search by default:

- IIT_PRIMARY_KEY
- IIT_DESCR

By way of an example, consider an Asset Type that is configured such that the value of IIT_PRIMARY_KEY is the internal id (IIT_NE_ID) and the IIT_DESCR column is not populated, this is quite a common situation and renders the default Quick Search configuration above useless.

In this situation, there is normally an attribute of the asset type that contains the data that a user would recognize the asset by, for the example below such an attribute, “Sign Bridge Id” is stored in the column IIT_CHR_ATTRIB27.

Inserting the row shown below into awlrs_quick_search_columns would mean that when a quick search is performed against the “SIGN BRIDGE” theme the “Sign Bridge Id” attribute would be matched against the search string entered by the user.

	AQSC_THEME_NAME	AQSC_COLUMN	AQSC_PRIORITY
▶	SIGN BRIDGE	IIT_CHR_ATTRIB27	1

NB. Any column names added to the awlrs_quick_search_columns table for an exor asset theme must exist on the table nm_inv_items_all.

4.1.3 Foreign Table Asset Themes

For themes that represent a Foreign Table Asset Type the column identified as the Primary Key is search by default.

By way of an example, consider a Foreign Table Asset where the identified Primary Key is an internal id from the source, again this would render the default Quick Search configuration useless.

In this situation, it is conceivable that an attribute of the foreign table asset type contains the data that a user would recognize the asset by, for the example below such an attribute, “Culvert Id” is stored in the column CULV_ID. The users also want to be able to quick search on the “Year Built” attribute stored in the column “YEAR_BUILT”.

Inserting the rows shown below into awlrs_quick_search_columns would mean that when a quick search is performed against the “DFMS CULVERTS” theme the “Culvert Id” and “Year Built” attributes would be matched against the search string entered by the user.

	AQSC_THEME_NAME	AQSC_COLUMN	AQSC_PRIORITY
▶	DFMS CULVERTS	CULV_ID	1
	DFMS CULVERTS	YEAR_BUILT	2

NB. Any column names added to the awlrs_quick_search_columns table for a foreign table asset theme must exist in the table or view that the foreign table asset type is based on.

4.1.4 Column Priority

When performing a Quick Search AWLRS will order the results as follows:

1. Records where the value of one of the columns being matched exactly matches the search string.
2. Records where the value of one of the columns being matched starts with the search string.
3. Records where the value of one of the columns being matched contains the search string.

The Column Priority is used to further refine this ordering, using the above example for a Network Theme:

	AQSC_THEME_NAME	AQSC_COLUMN	AQSC_PRIORITY
▶	SEGMENTS	NE_UNIQUE	1
	SEGMENTS	NE_DESCR	2
	SEGMENTS	NE_NUMBER	3

If the user were to enter the search string “1234” the results would be displayed in the following order:

1. Records where the value of the NE_UNIQUE column is 1234
2. Records where the value of the NE_DESCR column is 1234
3. Records where the value of the NE_NUMBER column is 1234
4. Records where the value of the NE_UNIQUE column starts with 1234
5. Records where the value of the NE_DESCR column starts with 1234
6. Records where the value of the NE_NUMBER column starts with 1234
7. Records where the value of the NE_UNIQUE column contains 1234
8. Records where the value of the NE_DESCR column contains 1234
9. Records where the value of the NE_NUMBER column contains 1234

4.2 External Links

AWLRS allows Network Elements and Assets to be linked to data in any external system that allows access to a record via a URL.

Once configured an external link will be displayed in the Action menu in the Search Results and Selection Results dialogs (if a single record is ticked) and in the Details dialog for a Network Element or Asset of a type that has been configured.

At time of writing there is no User Interface to allow the maintenance of this metadata so direct manipulation by an administrator using a sql client is required.

Table awlrs_external_links

This is the top-level table, rows in this table act as parent records to the rows in the parameters tables, this allows where required, the creation of multiple external link definitions that can be configured for any collection of Network and Asset Types.

The table contains a unique identifier, a name and the template URL.

	AEL_ID	AEL_NAME	AEL_URL_TEMPLATE
	1	ProjectWise	pw:\\Alpo.bentley.com:alpo-bentleygeospatial\\Documents\\AWLRS_Test\\{1}\\{2}\\

New rows can be added using an insert statement similar to the following:

```
INSERT
  INTO awlrs_external_links
    (ael_id
    , ael_name
    , ael_url_template)
VALUES (ael_id_seq.NEXTVAL
    , 'ProjectWise'
    , ' pw:\\Alpo.bentley.com:alpo-bentleygeospatial\\Documents\\AWLRS_Test\\{1}\\{2}\\' )
/
```

The template URL can contain multiple substitution variables that take the form {n} in the example above there are 2 substitution variables defined {1} and {2} (See below for details how to configure the source of these variables) these allow flexibility with regards to the URL that is generated for a specific Network Element or Asset.

The above example shows how External Links could be used to integrate with Bentley ProjectWise which allows access to folders and files via a URN which is displayed in the Address Bar of ProjectWise Explorer:

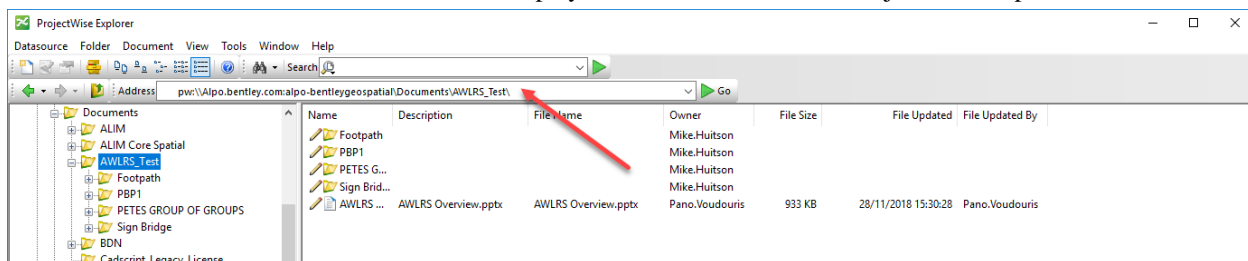


Table awlrs_external_link_params

The metadata contained within this table has 2 purposes:

- To determine which Network or Asset Type(s) the parent external link is valid for.
- Details of how to populate any substitution variables defined within the URL Template on the parent table

AELP_ID	AELP_AEL_ID	AELP_ENTITY_TYPE	AELP_ENTITY_TYPE_TYPE	AELP_SEQUENCE	AELP_SOURCE_TYPE	AELP_SOURCE	AELP_DEFAULT_VALUE
5	1	NETWORK	PAGT	1	VALUE	Footpath\Footpath Groups	
6	1	NETWORK	PAGT	2	COLUMN	NE_NAME_1	Default
7	1	NETWORK	PBNT	1	VALUE	PETES GROUP OF GROUPS	
8	1	NETWORK	PBNT	2	COLUMN	NE_NAME_1	Default
1	1	ASSET	SNBR	1	VALUE	Sign Bridge	
2	1	ASSET	SNBR	2	COLUMN	IIT_NUM_ATTRIB100	Default
3	1	NETWORK	PATH	1	VALUE	Footpath	
4	1	NETWORK	PATH	2	COLUMN	NE_OWNER	Default
17	1	ASSET	PBP1	1	VALUE	PBP1	
18	1	ASSET	PBP1	2	COLUMN	IIT_CHR_ATTRIB27	Default

In the screenshot above the Network Types “PATH”, “PAGT” and “PBNT” and Asset Types “SNBR” and “PBP1” have been configured for the ProjectWise parent record in the awlrs_external_links table (aelp_ael_id = 1).

The first substitution variable {1} has been configured as a “VALUE” and each Type has its own folder defined as the value (aelp_source).

The second substitution variable {2} has been configured as a “COLUMN” and the name of the column is given as the value of aelp_source.

New rows can be added using an insert statement similar to the following:

```
INSERT
  INTO awlrs_external_link_params
    (aelp_id, aelp_ael_id, aelp_entity_type, aelp_entity_type_type, aelp_sequence
    ,aelp_source_type, aelp_source, aelp_default_value)
VALUES(aelp_id_seq.NEXTVAL
      ,1
      ,'ASSET'
      ,'SNBR'
      ,1
      ,'VALUE'
      ,'Sign Bridge'
      ,NULL)
/
```

aelp_ael_id is the id of the parent record in awlrs_external_links.

aelp_entity_type can be either ‘NETWORK’ or ‘ASSET’

aelp_entity_type_type can be any Network Type or Asset Type (depending upon the value of **aelp_entity_type**)

aelp_sequence determines the substitution variable that the row refers to i.e. a value of 1 would point to “{1}” in the template URL.

aelp_source_type can one of the following:

- “VALUE
- “COLUMN”
- “FUNCTION”

aelp_source Depending upon the value used for aelp_source_type this can be a hard-coded value (“VALUE”), a column that exists upon the relevant Network/Asset table (“COLUMN”) or a function that conforms to a specific signature (“FUNCTION”).

Note: When “COLUMN” is used as the value aelp_source_type of the value given in aelp_source must be:

- A column from the table nm_inv_items_all for an exor Asset Type
- A column from the Foreign Table for a Foreign Table Asset Type
- A column from either nm_elements_all for a Network Type (or nm_inv_items_all if the Network Type has a Primary AD Asset Type defined).

aelp_default_value – can be used as a default in case the nominated column or function NULL.

4.2.1 Examples

The combination of a template URI and the substitution variables makes the configuration of External Links very flexible. The following examples should help give an idea of what can be achieved:

4.2.1.1 URI as template with substitution variables

The example used in the above section shows how this can be used to integrate with ProjectWise via a URN, have a separate folder for each Network\Asset Type and an Attribute that stores a specific Sub Folder and/or Filename for each Network Element or Asset. The example shows how the part of the URN that is common to all entities can be placed in the URL Template column with substitution variables supplying just the part of the URN that is specific to each entity type and individual entity. A similar configuration could be used for any external system that has a common URI with variations for a specific entity, for example the Oracle Forms application that has a form that accepts the entity id as a parameter:

In this example the exor Oracle Forms application has a form for displaying and editing Network Datum Elements, the form accepts a parameter containing the Network Element Id and will automatically query the given Network Element when the form is called and a value is provided for this parameter:

Table awlrs_external_links

	AEL_ID	AEL_NAME	AEL_URL_TEMPLATE
	2	Exor Forms	http://wardevas1ef.online.local:9001/forms/frmservlet?config=core_4700{1}{2}

This row would be added using the following insert statement:

```
INSERT
  INTO awlrs_external_links
    (ael_id
    , ael_name
    , ael_url_template)
VALUES (ael_id_seq.NEXTVAL
    , 'Exor Forms'
    , 'http://wardevas1ef.online.local:9001/forms/frmservlet?config=core_4700{1}{2}')
/
```

Table awlrs_external_link_params

	AELP_ID	AELP_AEL_ID	AELP_ENTITY_TYPE	AELP_ENTITY_TYPE_TYPE	AELP_SEQUENCE	AELP_SOURCE_TYPE	AELP_SOURCE	AELP_DEFAULT_VALUE
▶	9	2	NETWORK	PATH	1	VALUE	&Form=nm0105&otherparams=ne_id_1=	
	10	2	NETWORK	PATH	2	COLUMN	NE_ID	

In the screenshot above the Network Type “PATH” has been configured for the Exor Forms parent record (aelp_ael_id = 2).

The first substitution variable {1} has been configured as a “VALUE” that names the form to be called and the parameter to be populated.

The second substitution variable {2} has been configured as a “COLUMN” and names the primary key (NE_ID) as the column to be used.

By using 2 substitution variables this External Link could be configured to call different forms using a different parameter name and the relevant primary key column for any Entity Type, for example for a Network Element that is a Road Group the Road Groups form could be called, for and Asset the Asset Maintenance form could be called.

The rows shown above would be added using the following insert statements:

```
INSERT
  INTO awlrs_external_link_params
    (aelp_id, aelp_ael_id, aelp_entity_type, aelp_entity_type_type, aelp_sequence
    , aelp_source_type, aelp_source, aelp_default_value)
VALUES (aelp_id_seq.NEXTVAL
    , 2 -- The relevant id for the parent record in awlrs_external_links
    , 'NETWORK'
    , 'PATH'
    , 1
    , 'VALUE'
    , '&Form=nm0105&otherparams=ne_id_1='
    , NULL)
/
```

```

INSERT
  INTO awlrs_external_link_params
    (aelp_id, aelp_ael_id, aelp_entity_type, aelp_entity_type_type, aelp_sequence
    ,aelp_source_type, aelp_source, aelp_default_value)
VALUES (aelp_id_seq.NEXTVAL
    ,2 -- The relevant id for the parent record in awlrs_external_links
    , 'NETWORK'
    , 'PATH'
    ,2
    , 'COLUMN'
    , 'NE_ID'
    , NULL)
/

```

4.2.1.2 URI with no substitution variables

There may be a situation in which the requirement is simply to take the user to a predefined URI that is the same for every instance of a Network or Asset Type, this may for example point to a web page that details some relevant legislation relating to the operation of assets of the configured type.

For simplicity the example below uses the home page of the Bentley Web Site, but this could be any valid URI.

Table awlrs_external_links

	AEL_ID	AEL_NAME	AEL_URL_TEMPLATE
▶	3	Bentley	https://www.bentley.com/

This row would be added using the following insert statement:

```

INSERT
  INTO awlrs_external_links
    (ael_id
    , ael_name
    , ael_url_template)
VALUES (ael_id_seq.NEXTVAL
    , 'Bentley'
    , ' https://www.bentley.com/')
/

```

Table awlrs_external_link_params

	AELP_ID	AELP_AEL_ID	AELP_ENTITY_TYPE	AELP_ENTITY_TYPE_TYPE	AELP_SEQUENCE	AELP_SOURCE_TYPE	AELP_SOURCE	AELP_DEFAULT_VALUE
▶	19	3	ASSET	PBP4	1	VALUE	any value	

In the screenshot above the Asset Type “PBP4” has been configured for the “Bentley” parent record (aelp_ael_id = 3). As no substitution variable has been defined in the Template URL of the parent record the VALUE given in

aelp_soure can be anything as it will not be used when constructing the URI to be called.

The row shown above would be added using the following insert statement:

```
INSERT
  INTO awlrs_external_link_params
    (aelp_id, aelp_ael_id, aelp_entity_type, aelp_entity_type_type, aelp_sequence
    ,aelp_source_type, aelp_source, aelp_default_value)
VALUES(aelp_id_seq.NEXTVAL
  ,3 -- The relevant id for the parent record in awlrs_external_links
  , 'ASSET'
  , 'PBP4'
  ,1
  , 'VALUE'
  , 'any value'
  , NULL)
/
```

4.2.1.3 Entire URI stored as an attribute value

There may be a situation in which a customer wants to be able to just paste a URI into an attribute and have an Action to launch the given URI. This would allow the user complete flexibility to determine what is called for each individual instance of a Network Element or Asset.

The following shows how an External Link could be configured for integration with ProjectWise where the user can simply cut and paste the entire URN from the ProjectWise Explorer Address Bar into an attribute of the Network Element or Asset.

Table awlrs_external_links

	AEL_ID	AEL_NAME	AEL_URL_TEMPLATE
	6	ProjectWise Full URN	{1}

This row would be added using the following insert statement:

```
INSERT
  INTO awlrs_external_links
    (ael_id
    , ael_name
    , ael_url_template)
VALUES(ael_id_seq.NEXTVAL
  , 'ProjectWise Full URN'
  , ' {1} ')
/
```

Table awlrs_external_link_params

AELP_ID	AELP_AEL_ID	AELP_ENTITY_TYPE	AELP_ENTITY_TYPE_TYPE	AELP_SEQUENCE	AELP_SOURCE_TYPE	AELP_SOURCE	AELP_DEFAULT_VALUE
20	6	ASSET	SNBR	1	COLUMN	IIT_CHR_ATTRIB75	

In the screenshot above the Asset Type “SNBR” has been configured for the “ProjectWise Full URN” parent record (aelp_ael_id = 6). As the Template URL is defined purely as a single substitution variable the value of the attribute stored in the column IIT_CHR_ATTRIB75 will be used as the URI to be called.

The row shown above would be added using the following insert statement:

```
INSERT
  INTO awlrs_external_link_params
    (aelp_id, aelp_ael_id, aelp_entity_type, aelp_entity_type_type, aelp_sequence
    ,aelp_source_type, aelp_source, aelp_default_value)
VALUES(aelp_id_seq.NEXTVAL
    ,6 -- The relevant id for the parent record in awlrs_external_links
    ,'ASSET'
    ,'SNBR'
    ,1
    ,'COLUMN'
    ,'IIT_CHR_ATTRIB75'
    ,NULL)
/
```

4.2.1.4 Using a function to derive the value of a substitution variable

There may be situations where the value required for a substitution variable cannot be known upfront (VALUE) and is not stored as an attribute (COLUMN). If so then a custom function may be able to derive the required value.

Any function written for this purpose must conform with the following signature:

```
FUNCTION <function name>(pi_entity_type      IN  awlrs_external_link_params.aelp_entity_type%TYPE
                        ,pi_entity_type_type IN
awlrs_external_link_params.aelp_entity_type_type%TYPE
                        ,pi_entity_id       IN  NUMBER)
  RETURN VARCHAR2
```

The body of the function can contain any valid PLSQL that can be used to derive the required value to be returned as a character string.

When the system calls the Function as part of deriving the URL for a specific entity the parameters will be populated as follows:

- pi_entity_type - “NETWORK” or “ASSET” (awlrs_external_link_params.aelp_entity_type)
- pi_entity_type_type - The Network or Asset Type (awlrs_external_link_params.aelp_entity_type_type)
- pi_entity_id - The primary key of the Network Element or Asset

So as long as the Network or Asset Type and the primary key is enough information to derive the desired value a Function can be used.

The following example show how the x and y co-ordinates at the midpoint of a Network Element can be found and converted to the Lat – Lon co-ordinate system and returned in a format that can be used as part of a Google Street View URL so that Street View is opened at the given co-ordinates.

The custom function for this example has been coded to work with Network Elements only and to return a default set of co-ordinates should the values passed represent an Asset Type or something goes wrong. This function could be extended to support Asset Types if required, the important point being that the author of the function can do whatever they want to in order to derive the required value.

```
CREATE OR REPLACE FUNCTION get_sv_lat_lon(pi_entity_type      IN
awlrs_external_link_params.aelp_entity_type%TYPE
                                ,pi_entity_type_type IN
awlrs_external_link_params.aelp_entity_type_type%TYPE
                                ,pi_entity_id      IN NUMBER)
RETURN VARCHAR2 IS
--
lv_retval  nm3type.max_varchar2;
lv_default nm3type.max_varchar2 := '48.857832%2C2.295226'; -- default to Eiffel Tower
lv_point   mdsys.sdo_geometry;
--
BEGIN
--
CASE pi_entity_type
WHEN 'NETWORK'
THEN
lv_point :=
sdo_cs.transform(nm3sdo.get_xy_from_measure(pi_entity_id,nm3net.get_ne_length(pi_entity_id)/2),83
07);
lv_retval :=
nm3sdo.get_y_from_pt_geometry(lv_point)||'%2C'||nm3sdo.get_x_from_pt_geometry(lv_point);
ELSE
lv_retval := lv_default;
END CASE;
--
RETURN lv_retval;
--
EXCEPTION
WHEN others
THEN
RETURN lv_default;
END get_sv_lat_lon;
/
```

Table awlrs_external_links

	AEL_ID	AEL_NAME	AEL_URL_TEMPLATE
	4	Street View	https://www.google.com/maps/@?api=1&map_action=pano&viewpoint={1}

This row would be added using the following insert statement:

```
INSERT
  INTO awlrs_external_links
    (ael_id
    , ael_name
    , ael_url_template)
VALUES (ael_id_seq.NEXTVAL
    , 'Street View'
    , 'https://www.google.com/maps/@?api=1&map_action=pano&viewpoint={1}')
/
```

Table awlrs_external_link_params

	AELP_ID	AELP_AEL_ID	AELP_ENTITY_TYPE	AELP_ENTITY_TYPE_TYPE	AELP_SEQUENCE	AELP_SOURCE_TYPE	AELP_SOURCE	AELP_DEFAULT_VALUE
▶	15	4	NETWORK	HWY	1	FUNCTION	get_sv_lat_lon	
	12	4	NETWORK	PATH	1	FUNCTION	get_sv_lat_lon	
	13	4	NETWORK	SEGM	1	FUNCTION	get_sv_lat_lon	

In the screenshot above the Network Types “HWY”, “PATH” and “SEGM” have been configured for the “Street View” parent record (aelp_ael_id = 4).

The first substitution variable {1} has been configured as a “FUNCTION” with the source being get_sv_lat_lon so this function will be called and it’s return value used when deriving the URI for a specific entity.

Notes:

- The function used does not have to be a standalone function, if the function belongs to a package then the standard notation of <package name>.<function name> can be used as the value of aelp_source.
- If required different functions can be specified for different Network or Asset Types.

The rows shown above would be added using the following insert statements:

```
INSERT
  INTO awlrs_external_link_params
    (aelp_id, aelp_ael_id, aelp_entity_type, aelp_entity_type_type, aelp_sequence
    , aelp_source_type, aelp_source, aelp_default_value)
VALUES (aelp_id_seq.NEXTVAL
    , 4 -- The relevant id for the parent record in awlrs_external_links
    , 'NETWORK'
    , 'HWY'
    , 1
    , 'FUNCTION'
    , 'get_sv_lat_lon'
    , NULL)
/
```

```
INSERT
  INTO awlrs_external_link_params
    (aelp_id, aelp_ael_id, aelp_entity_type, aelp_entity_type_type, aelp_sequence
    ,aelp_source_type, aelp_source, aelp_default_value)
VALUES(aelp_id_seq.NEXTVAL
  ,4 -- The relevant id for the parent record in awlrs_external_links
  ,'NETWORK'
  ,'PATH'
  ,1
  ,'FUNCTION'
  ,'get_sv_lat_lon'
  ,NULL)
/
```

```
INSERT
  INTO awlrs_external_link_params
    (aelp_id, aelp_ael_id, aelp_entity_type, aelp_entity_type_type, aelp_sequence
    ,aelp_source_type, aelp_source, aelp_default_value)
VALUES(aelp_id_seq.NEXTVAL
  ,4 -- The relevant id for the parent record in awlrs_external_links
  ,'NETWORK'
  ,'SEGM'
  ,1
  ,'FUNCTION'
  ,'get_sv_lat_lon'
  ,NULL)
/
```

4.3 Modification of deployment options

AWLRS has built in options set when the application is deployed. Some of these can be amended, if needed, in the following file:

[AWLRS UI application]\app\config\config.json

For example: C:\inetpub\wwwroot\AWLRS\app\config\config.json

The options that can be changed include:

- **webApiUrl** – The URL of the deployed AWLRS Web API.
- **KendoUICulture** - The browser displayed culture type for example 'en-GB', 'en-US'
- **keepAliveInterval** - Number of milli-seconds after which a ping request is sent to WebApi to keep the session alive
- **editZoomThreshold** - The minimum zoom level before editing is allowed
- **includeEndDatedForQuickSearch**:
 - **"true"** – End dated data will be included in quick search results.
 - **"false"** - End dated data will NOT be included in quick search results.
- **maxRowsForQuickSearch** - The max number of returned rows on quick search
- **snapToTolerance** – The pixel tolerance applied when snapping to vertices and edges of geometries when drawing new features.

The contents of the file will look something like this:

```
"appConfig": {  
  "sysConfig": {  
    "webApiUrl": "http://[redacted]/AWLRSWebApi/api",  
    "kendoUICulture": "en-GB",  
    "projectTitle": "AssetWise ALIM Linear Referencing Services",  
    "version": "01.01.06.01",  
    "keepAliveInterval": 300000,  
    "externalIdP": false,  
    "decimalFormat": "#.####",  
    "decimalPlaces": 4,  
    "mapSMConfig": {  
      "editZoomThreshold": 20000,  
      "includeEndDatedForQuickSearch": false,  
      "maxRowsForQuickSearch": 100,  
      "snapTolerance": 5,  
      "recalibrateWithHistory": false,  
      "maxRecordsForExport": 150,  
      "searchExtentMinChar": 3  
    }  
  },  
}
```


5 Copy/Trace and Attribute Transfer configuration

Metadata mapping shape file feature attributes to Datum and Group attributes is needed to allow the attribute values of a selected feature in a shape file to be used:

- as default values when performing a Copy/Trace operation
- to overwrite the attribute values of one or more existing Datum elements (Attribute Transfer)

At time of writing there is no User Interface to allow the maintenance of this metadata so direct manipulation by an administrator using a sql client is required.

Table awlrs_file_feature_maps

This is the top-level table, rows in this table act as parent records to the rows in the Datum and Group attribute mapping tables, this allows where required, the creation of multiple attribute mappings to cover any format of shapefile that is likely to be encountered

The table contains a unique identifier, a description and the Target Datum Network Type.

	AFFM_ID	AFFM_FILE_DESCR	AFFM_DATUM_NT
I	1	Example Shapefile	SEGM

New rows can be added using an insert statement similar to the following:

```
INSERT
  INTO awlrs_file_feature_maps
    (affm_id
    ,affm_file_descr
    ,affm_datum_nt)
VALUES (affm_id_seq.NEXTVAL
    , 'Example Shapefile'
    , 'SEGM')
/
```

NB: At time of writing only one file format should be added to the table, a future enhancement will allow users to pick from a list of formats.

Table awlrs_file_grp_attr_map

This table allows attributes from the shapefile to be mapped to the Network Type attributes of any auto inclusion Group Type(s) configured for the Target Datum Network Type.

The purpose of this mapping is primarily to establish the Unique reference of the Parent Group allowing the system to check for the groups existence before displaying the Create Datum dialog.

Note: If the Unique is to be entered by the user and it exists as an attribute in the shapefile it can be specified in this table with the target attribute being NE_UNIQUE.

	AFGAM_ID	AFGAM_AFFM_ID	AFGAM_TARGET_NT	AFGAM_TARGET_GTY	AFGAM_FILE_ATTRIB	AFGAM_TARGET_ATTRIB
▶	1	1	HWY	HWY	HWYNUMB	NE_OWNER
	2	1	HWY	HWY	ST_HWY_SFX	NE_SUB_TYPE
	3	1	HWY	HWY	RDWY_ID	NE_PREFIX
	4	1	HWY	HWY	MLGE_TYP	NE_NAME_1
	5	1	HWY	HWY	OVLP_CD	NE_NAME_2
	6	1	HWY	HWY	HWYNAME	NE_DESCR

New rows can be added using an insert statement similar to the following:

NB: the column afgam_affm_id is the id of the record that was inserted into awlrs_file_grp_attrib_map (above).

```
INSERT
  INTO awlrs_file_grp_attrib_map
    (afgam_id
    ,afgam_affm_id
    ,afgam_target_nt
    ,afgam_target_gty
    ,afgam_file_attrib
    ,afgam_target_attrib)
VALUES (AFGAM_ID_SEQ.NEXTVAL
    ,1
    ,'HWY'
    ,'HWY'
    ,'HWYNUMB'
    ,'NE_OWNER')
/
```

NB: If auto inclusion is being used and the Parent Group does not exist then it will need to be manually created before using Copy/Trace to create the datum.

Table awlrs_file_datum_attrib_map

This table allows attributes from the shapefile to be mapped to the attributes of the Target Datum Network Type allowing the values of the selected feature in the shapefile to be used as the default values in the Create Datum dialog or to be used to update the attributes of 1 or more existing Datums.

Note: Any Target Attributes that form part of the Datum Unique or are the Child Column in an Auto Inclusion configuration will not be included in any Attribute Transfer performed.

AFDAM_ID	AFDAM_AFFM_ID	AFDAM_FILE_ATTRIB	AFDAM_TARGET_ATTRIB
3	1	HWYNUMB	NE_PREFIX
2	1	HWYNAME	NE_DESCR

New rows can be added using an insert statement similar to the following:

NB: the column afdam_affm_id is the id of the record that was inserted into awlrs_file_feature_maps (above).

```

INSERT
  INTO AWLRS_FILE_DATUM_ATTRIB_MAP
    (AFDAM_ID
    ,AFDAM_AFFM_ID
    ,AFDAM_FILE_ATTRIB
    ,AFDAM_TARGET_ATTRIB)
VALUES (AFDAM_ID_SEQ.NEXTVAL
    ,1
    , 'HWYNUMB '
    , 'NE_PREFIX')
/

```

6 Street Gazetteer Manager

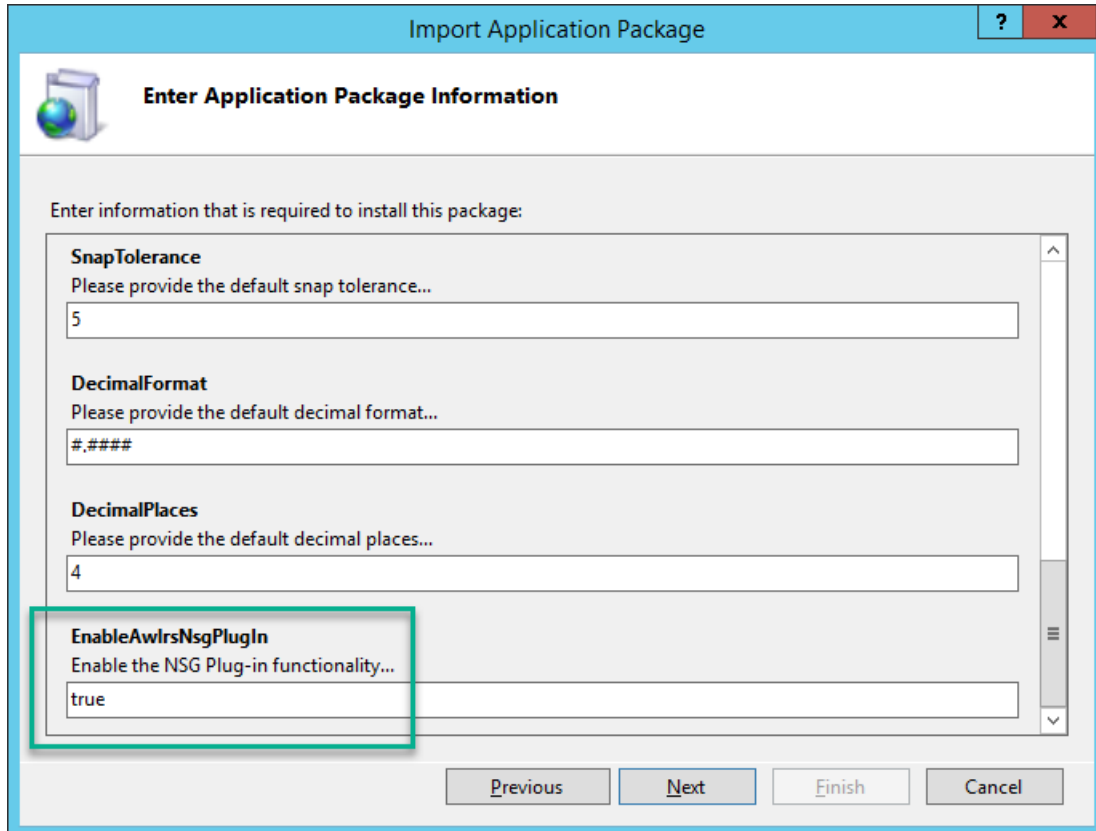
Included in the AWLRS 1.1.6 release is our Street Gazetteer Manager product. For the purpose of this document, and to avoid any confusion, we will refer to this product with its more commonly known name of NSG.

6.1 Enabling NSG

To be able to use NSG on the AWLRS platform, the following prerequisites need to be undertaken:

- NSG Database patches to be applied
 - exnsg04070038en_updt17
 - exnsg04070039en_updt18
- Enable NSG plugin

To enable the AWLRS NSG Plug-in, when installing AWLRS you can simply update the option EnableAwlrNSGPlugin to true as shown in the example below.
Please refer to section 4.4 of the AWLRS Product and Installation Guide v1.1.6 should you require further information.



To enable NSG post installation of AWLRS 1.1.6, simply locate the config file for the deployed application, for example:

... \AWLRS\app\config\config.json

Then locate the section:

```
{  
  "name": "awlrs.nsg",  
  "version": "01.01.06.01",  
  "enabled": false,  
  "stylesheets": [  
    "
```

And change the option “enabled”: from false to true.

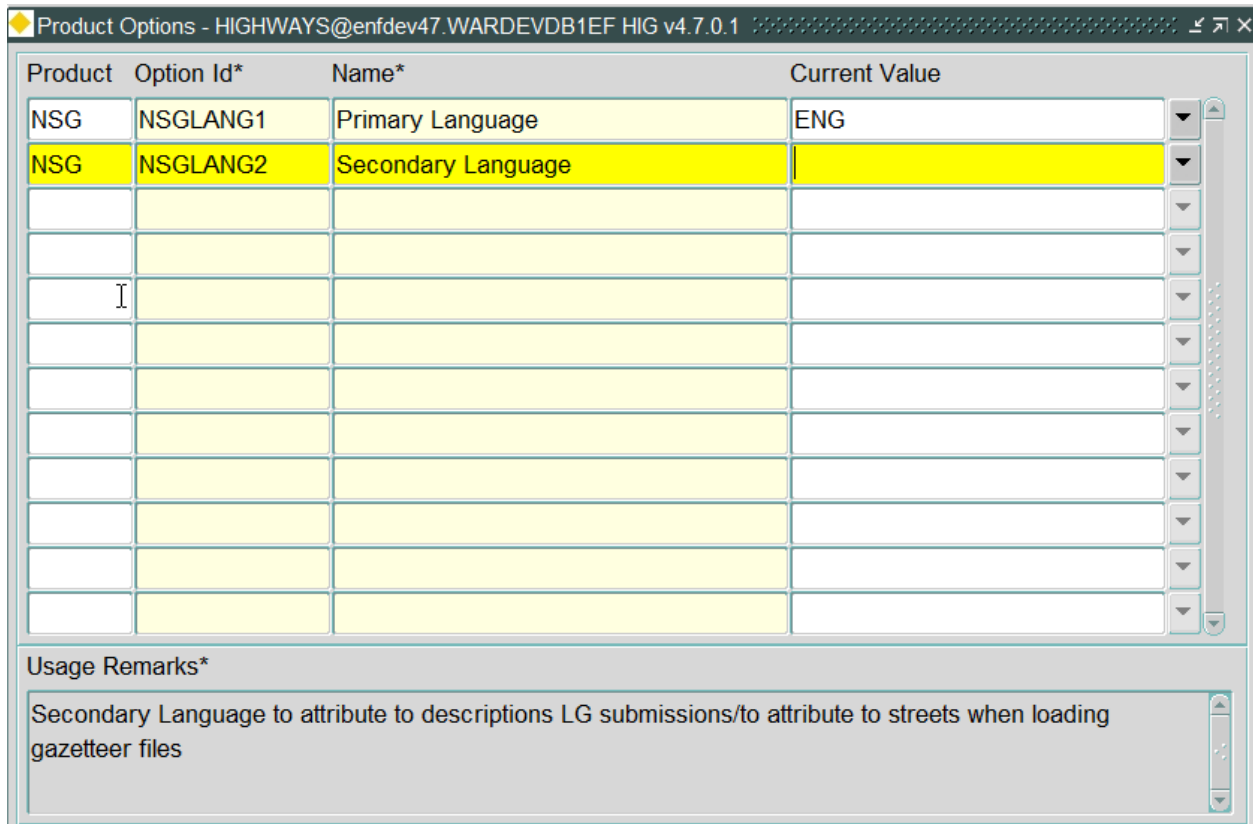
You can use this same method to disable NSG, the only difference being here is that you would be setting the enabled option to false.

6.2 Product Options

Here are some product options that can/should be set when implementing AWLRS NSG.

Many will be set up in NSG implementations already as they are used in the Exor Forms implementation. Options can be set at product level and User level as needed.

- NSGLANG1 – ENG
- NSGLANG2 – Currently not set. However, if working with Welsh streets this should be set to CYM to allow this secondary language to be captured. Please note once set, it will be mandatory for all streets to have street aliases present so setting this product option at User level may be your preferred choice.



Product	Option Id*	Name*	Current Value
NSG	NSGLANG1	Primary Language	ENG
NSG	NSGLANG2	Secondary Language	

Usage Remarks*

Secondary Language to attribute to descriptions LG submissions/to attribute to streets when loading gazetteer files

For users whose responsibility is to Create Streets, we recommend the following product options to be set appropriately:

- DFLT_AU
- DFLT_CNTY
- DFLT_NAUTH

- DFLT_TOWN
- DFLT_TRNCE
- DFLT_TYPE

6.3 Custom Map Layers

AWLRS installations by default assume that a map configuration exists and by default is named "AWLRS_MAP" (a configurable product option). We refer to the Map Configuration section within this document together with the exor core document "Spatial Server Admin" for details on how to use Oracle Map Builder to configure a Base Map for AWLRS.

AWLRS NSG makes use of the AWLRS map to surface NSG Street, ESU and ASD data visually. This functionality relies on a set of map themes existing within the base AWLRS map. The NSG capabilities within AWLRS NSG are made aware of the details of the map themes to use through configuration. This allows for adjustments to the themes to be made and alterations to the way the NSG data is presented to the user.

The default configuration is set with the following themes:

- ESU
Elementary Street Unit
- ASDTYPE61
A collection of up to three geometry types (for each of point, line and polygon) for ASD Type 61 (interest) details
- ASDTYPE62
A collection of up to three geometry types (for each of point, line and polygon) for ASD Type 62 (construction) details
- ASDTYPE63
A collection of up to three geometry types (for each of point, line and polygon) for ASD Type 63 (special designation) details
- ASDTYPE64
A collection of up to three geometry types (for each of point, line and polygon) for ASD Type 64 (restrictions) details
- ASDTYPE66
A collection of up to three geometry types (for each of point, line and polygon) for ASD Type 66 (PRoW) details
- TYPE1AND2STREETS
- TYPE3STREETS
- TYPE4STREETS

For some of the themes noted above a collection is provided so that different geometry types (point, line, polygon) can all be referenced in the same layer, meaning they can be shown on screen together and searched across together in the AWLRS user interface.

Individual point, line, and polygon themes can also be used to identify the specific spatial data for an individual geometry type if so required.

The grouped theme collects the individual geometry types together by referencing the different types in a custom tag set up in map builder. Please refer to the discussion of the **GeometryTypes** custom tag in the Custom Tags section (3.3) of this Administration Guide for more detail on how this is used.

6.4 NSG Configuration

Whilst the provided themes are not suitable for all production purposes they serve as an example to assist in getting started with AWLRS NSG and as an introduction to the way the themes should be set up.

When altering the themes to suit a specific deployment the following should be noted.

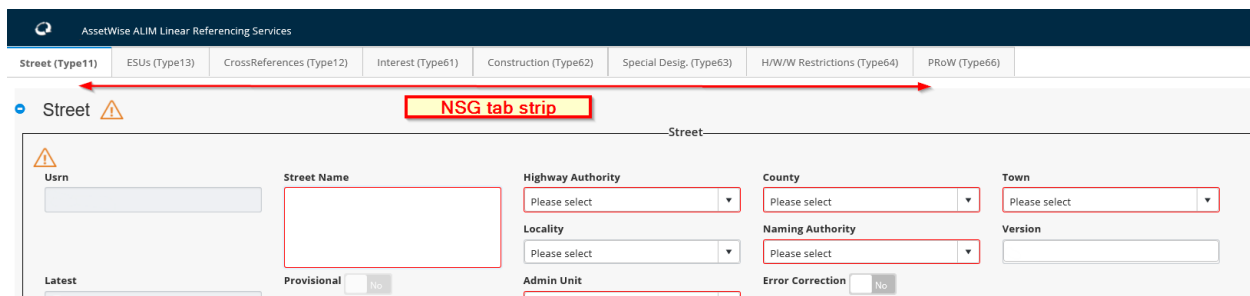
- If you choose to use a theme with a different name, then the relevant configuration in AWLRS NSG must be updated to reflect the changed name (see below).
- If you do not require a particular type of data to be available to your users, then the relevant section can be removed from the NSG configuration so that it does not appear in the UI (see below).

The AWLRS NSG specific configuration related to themes and layers is managed in a configuration file named **template.json** which is located in the following folder

- [AWLRS UI application deployment path]\app\config\nsg
 For example, C:\inetpub\wwwroot\AWLRS\app\config\nsg

This **template.json** file is crucial to the function of AWLRS NSG and so alterations must be undertaken with care. You are advised to make a backup copy of the file prior to making any changes. You are also asked to only alter the items documented here. Uncontrolled change could result in the failure of AWLRS NSG.

The content of the is JSON configuration, so is human and machine-readable text and can be edited with a simple text editor. The arrangement of the contents is a series of tab objects relating to the tabs shown in the NSG tab strip of the AWLRS NSG browser tab:



The order of the tab objects in the configuration file determine the order of the tabs with the default order being that shown above, starting with “Street (Type11)” and ending with “PRoW (Type66)”.

The title in each tab is set in the JSON configuration. Consider for example the fourth tab along from the left “Interest (Type61)” the configuration section in **template.json** will be:

```
{
  "text": "Interest (Type61)",
```



```
"templateId": 104,
"preLoadContent": true,
"url": "awlrs.nsg/templates/type61.html",
"contentUrl": "awlrs.nsg/templates/type61.html?templateId=104",
"additionalData": {
  "asdLayerNames": {
    "group": "ASDTYPE61",
    "line": null,
    "polygon": null,
    "point": null
  },
  "streetIdAttr": "STREET_ID",
  "mapId": "ASDListMap_61",
  "controllerName": "type61.controller"
},
"tabProperties": {
  "hidden": false,
  "disable": false
}
},
```

In the above snippet from the configuration file the following items are of interest and suitable for user configuration:

- **"text"**
The text set for this item will appear as the title on the tab itself.
- **"group"**
The value of this configuration item is the theme name of the group theme set in map builder. The relevance of this setting is that it controls the default layer turned on for the tab in question and hence the information that will be shown on the map within the tab. The **group** item is a property of the **asdLayerNames** object. The **group** value should be set when group theme has been configured (the recommended approach). Within the **asdLayerNames** object there are also properties for **point**, **line** and **polygon**. Each of these can be used in isolation or in combination with **group** to allow for the tab to visualize map data for that geometry type, if the relevant named theme has been created with map builder and associated with the AWLRS map.
- **"streetIdAttr"**
Each tab needs to be able to identify the street related to the data shown in the tab. To do this we identify the attribute for the theme data that represents the street id.

Separate to the tab strip there are some other items that could be configured if theme names are changed:

```
"layersByGroupNumber": [ { "groupNumber": "1", "layerName": "TYPE1AND2STREETS" }, { "groupNumber": "2", "layerName": "TYPE1AND2STREETS" }, { "groupNumber": "3", "layerName": "TYPE3STREETS" }, { "groupNumber": "4", "layerName": "TYPE4STREETS" } ]
```

- **"layersByGroupNumber"**

The tab strip configuration detailed above is about ASD record data the **layersByGroupNumber** identifies the themes that represent actual streets on the map and are used to show the streets and to allow the searching for streets in AWLRS. Layers are provided for:

- TYPE1AND2STREETS
- TYPE3STREETS
- TYPE4STREETS

Each street type has its own record type and the record types must be associated with the correct theme. The record type is referred to as the groupNumber in the configuration and the configuration by default asserts that TYPE1AND2STREETS is represented both by groupNumber 1 and 2, TYPE3STREETS with groupNumber 3 and TYPE4STREETS with groupNumber 4. If the theme names or record types are altered, then the changes must be reflected in **layersByGroupNumber**.

6.5 Quick Search Configuration

To support street name and USRN searches in AWLRS from the quick search tool you will need to ensure it is suitably configured in the Exor database.

For example, for Type 1 and 2 streets and Type 3 streets:

AQSC_THEME_NAME	AQSC_COLUMN	AQSC_PRIORITY
▶ TYPE 1 AND 2 STREETS	NE_NUMBER	1
TYPE 1 AND 2 STREETS	NE_DESCR	2
TYPE 3 STREETS	NE_NUMBER	1
TYPE 3 STREETS	NE_DESCR	2

For more information regarding the configuration of Quick Search capabilities in AWLRS, we refer you to section 4.1 in this administrative guide.