



DISCOVER > ANALYSE > ACT

MapLink Pro 11.0

Deployment of End User Applications

AUM1110|27 August 2021| Version\Status: 2.0\Approved

© Envitia Ltd. 2021

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

Tel: +44 1403 273 173 Email: info@envitia.com

www.envitia.com

Commercial in Confidence

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

TABLE OF CONTENTS

1	INTRODUCTION.....	1
1.1	Supported Platforms	1
1.2	Training, Consultancy and Sub-Contracting	1
1.3	Glossary of Terms and Definitions.....	1
2	LICENCE REQUIREMENTS AND RESTRICTIONS	2
2.1	Restrictions.....	2
2.2	Copyright Statements for Derivative Programs.....	2
2.2.1	Open Source Libraries.....	2
2.3	Sample Map Data and Maps	2
3	DEPLOYMENT OF AN APPLICATION.....	3
3.1	Release Mode DLLs.....	3
3.2	Runtime Unlock	3
3.3	Watermark.....	3
3.4	Configuration Files.....	3
3.5	Directory Naming Conventions	3
4	CONFIGURATION FILES.....	5
4.1	C++	5
4.2	.NET	6
4.3	GDAL/OGR	6
4.3.1	MapLink Runtime Deployment.....	6
4.3.2	GDAL/OGR Executable Deployment	7
4.4	config Directory	8
4.4.1	Standard Resource files	9
4.4.2	OGC Services.....	10
4.4.3	3D	10
4.4.4	Error Message Files.....	11
5	WINDOWS REQUIRED LIBRARIES	13
5.1	MapLink Libraries	14
5.1.1	The MapLink Pro Toolkit.....	14
5.1.2	GeoPackage	21
5.1.3	MapLink Pro Accelerator SDK	21
5.1.4	MapLink Pro Terrain SDK	22
5.1.5	MapLink Pro 3D SDK	23
5.1.6	MapLink Pro Satellite SDK	24
5.1.7	MapLink Pro S-63 SDK	24
5.1.8	MapLink Pro S-52 SDK	24
5.1.9	MapLink Pro Network SDK.....	25
5.1.10	MapLink Pro Editor SDK.....	26
5.1.11	MapLink Pro Spatial Editor SDK (LandLink).....	27
5.1.12	MapLink LandLink DIGM Convertor SDK	27
5.1.13	MapLink Impact Assessor SDK.....	27
5.1.14	MapLink CADRG Exporter SDK.....	28
5.1.15	MapLink ASRP Exporter SDK.....	28
5.1.16	MapLink Track Manager SDK.....	28
5.1.17	MapLink OGC SDKs	29
5.1.18	MapLink OGC Services	30
5.1.19	MapLink Filter DLLs.....	31
5.1.20	GDAL/OGR DLLs	34
5.1.21	PDFTOPPM.EXE.....	34
5.2	.NET Assemblies.....	36
5.3	Raster Handling Libraries.....	37

5.3.1	Required Libraries.....	37
5.3.2	Optional Libraries	38
5.4	Microsoft Runtime Libraries	40
5.5	3 rd Party Libraries/Executables	42
5.5.1	GDAL Executables	43
6	.NET DEPLOYMENT ISSUES	45
6.1	Dependency Loading	45
6.1.1	Locating MapLink assemblies via the application configuration file	45
6.1.2	Locating MapLink assemblies from the Global Assembly Cache	46
6.1.3	Using the PATH environment variable	46
7	MAPLINK STUDIO AUTOMATION	48
7.1	Deploying.....	48
7.1.1	Registry Updates.....	48
7.1.1.1	64bit Studio Automation	48
7.2	Uninstalling Studio Automation	49
7.3	Checking Studio Automation	50
7.3.1	Regenerating the Automation Classes	50
7.4	Studio Automation Deployment Problems.....	51
7.4.1	Licence Dialog is shown	51
7.4.2	Config Directory cannot be found	52
7.4.3	Shipping a Licenced version of MapLink Studio	53
8	LINUX/OPENGL/X11 REQUIRED LIBRARIES.....	54
8.1	MapLink Libraries	54
8.1.1	The MapLink Pro Toolkit.....	54
8.1.2	MapLink GeoPackage	60
8.1.3	MapLink Pro Accelerator SDK.....	60
8.1.4	MapLink Pro Terrain SDK.....	60
8.1.5	MapLink Pro 3D SDK	61
8.1.6	MapLink OGC Services	61
8.2	Raster Handling Libraries.....	63
8.2.1	Required Libraries.....	63
8.3	3 rd Party Libraries.....	63
9	COMMON DEPLOYMENT PROBLEMS	68
9.1	DLL/Shared Library not found	68
9.2	MapLink Studio.....	68

1 INTRODUCTION

This document details the necessary steps for deploying your application, from necessary modifications, required configuration files and a list of all libraries required.

1.1 Supported Platforms

Please refer to the Release Notes for the Supported platforms.

1.2 Training, Consultancy and Sub-Contracting

Envitia provides a range of training options to help you get the best from MapLink Pro and MapLink Studio. These courses greatly help to accelerate your development, produce optimised applications more quickly and to explore alternative ways of achieving your objectives.

Dedicated consultancy can also be provided either on-site or remotely, allowing our experienced developers to guide you towards the most appropriate approach to your application arena. Customers frequently find this useful when adding additional new functionality to their systems.

Envitia can also help accelerate your development by developing the MapLink component of your application for you or by undertaking a more extensive part of your project for you. Envitia has extensive experience of developing applications internally and for external customers.

If you wish to discuss these opportunities, please contact Sales by email sales@envitia.com or by phone: +44 1403 273173.

1.3 Glossary of Terms and Definitions

BOM	UTF-8 byte order mark.
DLL	Shared Library (.so/.dll etc...).
JPEG	JPEG raster format.
MFC	Microsoft Foundation Classes.
PNG	Portable Network Graphic raster format.
SDK	Software Development Kit.
TIFF	Tagged Image File Format raster format.
TMF	Envitia Mapping Format.

2 LICENCE REQUIREMENTS AND RESTRICTIONS

2.1 Restrictions

MapLink Pro software shall not be installed without a valid licence from Envitia.

You shall not distribute any file supplied with MapLink Pro without first obtaining the proper licence from Envitia.

You shall not distribute the header files with your application and you shall not provide software development capabilities in your application.

For further information on Envitia's Maplink Pro Software Terms and Conditions for use and the details of all copyright statements please refer to the 'MapLink Pro Windows Licence.pdf' or the 'MapLink Pro X11 Android & OpenGL Licence.pdf'. The appropriate document can be found on the installation CD and with the documentation installed on your development machine (within the `help` or `docs` directory of your installation).

Please contact Envitia (www.envitia.com) or your supplier for further information.

2.2 Copyright Statements for Derivative Programs

MapLink Pro uses additional software from various sources. MapLink Pro and these additional software portions are copyrighted to their respective owners. Any programs derived from MapLink Pro must contain the appropriate copyright statements.

This section details when each copyright statement must be included in the 'About Box' of derived end user applications.

All applications require the follow statement:

```
Portions hereof copyright © 1991-2017 Envitia Group PLC. All rights reserved;
```

Windows-based applications that use raster maps or raster images require the following statement:

```
Portions hereof copyright © 1991-2009 LEAD Technologies, Inc. All rights reserved;
```

Additional copyright statements may be required for Open Source components. Please see the next section.

Intel® TBB is used by MapLink Pro. The statement for this can be found in the 'MapLink Pro 3rd Party Licences.pdf' file depending on the platform being deployed upon.

2.2.1 Open Source Libraries

Several Open Source libraries are used by MapLink Pro. The copyright and licensing information for all such components may be found in the 'MapLink Pro 3rd Party Licences.pdf' file. The document can be found in either the `docs` or `help` directory of your installation.

Envitia can supply the source code for these libraries upon request.

2.3 Sample Map Data and Maps

Please refer to the 'MapLink Pro Sample Data & Maps.pdf' document for details of the data used to generate the Maps and associated copyright and license information. This can be found within the installation `docs` directory or via the shortcut menu.

3 DEPLOYMENT OF AN APPLICATION

3.1 Release Mode DLLs

Only Release mode DLLs should be deployed with an application. The debug versions are supplied only for development.

The MapLink Pro DLLs are listed in this document to help you work out what DLLs to ship with your application.

3.2 Runtime Unlock

Certain features are runtime locked and can be unlocked via a key. This is achieved using the following method (.NET version exists with a very similar name):

```
TSLUtilityFunctions::unlockSupport (TSLKeyedXXXXXXXX, UNLOCK_CODE);
```

The section 'MapLink Libraries' lists which SDKs require additional licenses or unlock codes.

If you are entitled to an unlock code and have not received it please contact support@envitia.com.

3.3 Watermark

From MapLink 7.1 onwards, maps generated from an Evaluation version of MapLink Studio will always display a Watermark when used in an application. This also applies to the sample maps that are supplied. Customers wishing to use such maps in a deployed application can request an unlock code from Envitia Support. This unlock code should be added to the application:

```
TSLUtilityFunctions::unlockSupport (TSLKeyedEvaluationMap, UNLOCK_CODE);
```

Please contact support@envitia.com for the unlock code.

3.4 Configuration Files

MapLink Pro uses a significant number of configuration files. These need to be shipped with an application and your application code needs to be updated to refer load the configuration files (see the next section).

You do not need to ship all the files. Only the files your application uses needs to be shipped. Please contact support@envitia.com if you require additional guidance.

3.5 Directory Naming Conventions

MapLink expects that the following directory conventions are observed by an application installation:

Architecture	Directory Name
64 bit	bin64
All	config

In the case of the `bin64` directory MapLink stores plugins. Because the plugins may have the same name on different architectures the name of the directory containing the shared libraries and executables is assumed to define the architecture.

The `config` directory is assumed to be in the same directory as the `bin64` directory.

4 CONFIGURATION FILES

MapLink Pro loads all of its necessary configuration files from the '`<MapLink Installation>\config`' directory, usually through a call to `TSLDrawingSurface::loadStandardConfig`. When deploying an application based upon MapLink Pro, a copy of this folder must be shipped along with the application.

The MapLink Pro installer adds a reference to the system registry to allow the MapLink libraries to locate the `config` directory at runtime. Envitia does not recommend that this registry key or any MapLink environment variable is used when deploying applications based upon MapLink Pro.

Therefore as the MapLink libraries will not know the location of this directory on the deployment machine's file system, calls to various MapLink methods will need to be changed to be passed the location of the `config` directory. The following table lists the current method calls which will need to be updated, depending upon the technology being used.

Note: If an application to be deployed does not use a method mentioned, then that method may be ignored.

4.1 C++

Method
<code>TSLDrawingSurface::loadStandardConfig</code>
<code>TSLDrawingSurface::setupColours</code> – Pass the location of the <code>tslcolours.dat</code> file that the <code>config</code> directory contains.
<code>TSLDrawingSurface::setupFillStyles</code> – Pass the location of the <code>tslfillstyles.dat</code> file that the <code>config</code> directory contains.
<code>TSLDrawingSurface::setupFonts</code> – Pass the location of the <code>tslfonts.dat</code> file that the <code>config</code> directory contains.
<code>TSLDrawingSurface::setupLineStyle</code> – Pass the location of the <code>tsllinestyles.dat</code> file that the <code>config</code> directory contains.
<code>TSLDrawingSurface::setupSymbols</code> – Pass the location of the <code>tslsymbols.dat</code> file that the <code>config</code> directory contains.
<code>TSLCoordinateSystem::loadCoordinateSystems</code> – Pass the location of the <code>tsltransforms.dat</code> file that the <code>config</code> directory contains.
<code>TSLAPP6Ahelper::TSLAPP6Ahelper</code>
<code>TSLAPP6Ahelper::setDefaultConfigPath</code>
<code>TSL3DdrawingSurface::loadStandardConfig</code>
<code>TSL3DdrawingSurface::setupModels</code> – Pass the location of the <code>tslmodels.dat</code> file that the <code>config</code> directory contains.
<code>TSLUtilityFunctions::setMapLinkHome</code> – set the directory that contains the MapLink <code>config</code> directory.

4.2 .NET

Method
TSLNDrawingSurface::loadStandardConfig
TSLNDrawingSurface::setupColours - Pass the location of the <code>tslcolours.dat</code> file that the config directory contains.
TSLNDrawingSurface::setupFillStyles - Pass the location of the <code>tslfillstyles.dat</code> file that the config directory contains.
TSLNDrawingSurface::setupFonts - Pass the location of the <code>tslfonts.dat</code> file that the config directory contains.
TSLNDrawingSurface::setupLineStyle - Pass the location of the <code>tsllinestyles.dat</code> file that the config directory contains.
TSLNDrawingSurface::setupSymbols - Pass the location of the <code>tslsymbols.dat</code> file that the config directory contains.
TSLNCoordinateSystem::loadCoordinateSystems - Pass the location of the <code>tsltransforms.dat</code> file that the config directory contains.
TSLNAPP6AHelper::TSLNAPP6AHelper
TSLNAPP6AHelper::setDefaultConfigPath
TSLN3DDrawingSurface::loadStandardConfig
TSLN3DDrawingSurface::setupModels - Pass the location of the <code>tslmodels.dat</code> file that the config directory contains.
TSLNUtilityFunctions::setMapLinkHome - set the directory that contains the MapLink config directory.

4.3 GDAL/OGR

4.3.1 MapLink Runtime Deployment

MapLink Pro uses GDAL/OGR in the CoreSDK. The initialisation of GDAL/OGR occurs when any of the MapLink resource files are loaded.

GDAL/OGR uses the following two environment variables to define where to find its plugins and data:

- GDAL_DRIVER_PATH
- GDAL_DATA

These environment variables are not used by default, because they may have been set by another installation, unless the following environment variables are set:

- TTL_USE_GDAL_DRIVER_PATH
- TTL_USE_GDAL_DATA

The defaults are set by MapLink internally as follows:

```
GDAL_DATA          DIRECTORY/config/gdal
GDAL_DRIVER_PATH   ttlmgm.dll/gdalplugins/GDAL_VERSION_MAJOR.GDAL_VERSION_MINOR
```

Where:

```
DIRECTORY          DIRECTORY can be one of two values:
```

- MAPL_HOME
- The directory containing the resource file being loaded.

ttltgm.dll This is the location of the specified DLL.

GDAL_VERSION_MAJOR GDAL/OGR Major version

GDAL_VERSION_MINOR GDAL/OGR Minor version

4.3.2 GDAL/OGR Executable Deployment

The GDAL/OGR executables are provided with MapLink Pro as a courtesy. There are no restrictions on deployment as long as the GDAL/OGR and associated Third Party licences are complied with.

The functionality we have built into GDAL/OGR is limited to the default and some 3rd Party. The JPEG2000 support is disabled due to licencing restrictions.

The executables depend on the GDAL/OGR DLL/plugins and configuration data supplied with MapLink Pro.

The GDAL/OGR executables do not use the following two environment variables to define where to find thier plugins and data:

- GDAL_DATA
- GDAL_DRIVER_PATH

The default lookup locations are:

GDAL_DATA DIRECTORY/./config/gdal

GDAL_DRIVER_PATH DIRECTORY/gdalplugins/GDAL_VERSION_MAJOR.GDAL_VERSION_MINOR

Where:

DIRECTORY DIRECTORY is the directory containing the executable.

GDAL_VERSION_MAJOR GDAL/OGR Major version

GDAL_VERSION_MINOR GDAL/OGR Minor version

The default locations can be overridden by creating the following files:

- gdaldata.dat
- gdaldriver.dat

These files, if created, shall be placed alongside the executables.

The contents of the files are a single line specifying the location of the directory relative to the executables, for example; in a MapLink installation the contents would be:

File	File Contents
gdaldata.dat	./config/gdal
gdaldriver.dat	gdalplugins/2.1

The string should be UTF-8, though we would recommend limiting the string to ASCII. The file shall not start with a BOM.

4.4 config Directory

The MapLink Pro config directory contains all the configuration files for MapLink Pro. When you deploy an application you do not need to deploy all the files. You only need to deploy the files your application requires. This section explains what the files are required for.

Directory	Explanation
<i>DBIF</i>	DBIF-SDK configuration files.
<i>featurebooks</i>	Featurebooks for MapLink Pro Studio.
<i>FeatureDefinitionMapping</i>	Contains MapLink Studio configuration files. The configuration files provide standard mappings of feature and field data to an actual Feature and description.
<i>geotiff</i>	GeoTIFF filter support files.
<i>gdal</i>	GDAL/OGR Configuration files. Used by: MapLink Pro Studio KML direct import MapLink Pro
<i>palettes</i>	Alternative palettes for MapLink Pro Studio.
<i>s52</i>	S-52 & S-63 SDK configuration files.
<i>directimport</i>	Direct Import driver configurations.
<i>colourramps</i>	Colour-ramps for terrain visualisations (Studio and Direct Import)
<i>renditions</i>	Renditions for specific data products

File/Directory	Explanation
AMLS57Catalogue_v10.xml	S-57 AML Filter configuration file.
AMLS57Catalogue_v10_s52.xml	S-57 AML Filter configuration file for S-52.
2525bConfig.csv	2525B symbology configuration file
2525bUnfilledConfig.csv	2525B unfilled symbology configuration file
app6aConfig.csv	APP6A symbology configuration file
app6aUnfilledConfig.csv	APP6A unfilled symbology configuration file
APP6APalettes.dat	Palette file for the APP6A sample
DAFIFRedirectedAttributeLookup.txt	DAFIF Filter configuration.
DxfFilterEncoding.csv	DXF Filter text encoding conversions configuration file.
IHOS57Catalogue_v31.xml	S-57 v3.1 Filter configuration file.
IHOS57Catalogue_v311.xml	S-57 v3.1.1 Filter configuration file.
IHOS57Catalogue_v31_AMLv21.xml	AML 2.1

File/Directory	Explanation
IHOS57Catalogue_v31_AMLv30.xml	AML 3.0
ImageStudio.ini	Image Studio configuration file.
Jeppesen.ini	Jeppesen Filter configuration file.
jeppesenRedirectAttributeLookup.xml	Jeppesen Filter configuration file.
Landscape.tpt	Print Template
mapl.ini	MapLink Studio configuration file.
MapLink.fbk	MapLink Studio feature book
MapLinkStudioCodePages.csv	MapLink Studio text encoding conversions configuration file.
Mapuserconfig.ini	MapLink Studio user configurable items – this is copied to the users Application directory on settings change.
Metadata.xsl	Metadata xsl
metadatasimple.xsl	Metadata xsl
NitfConfiguration.ini	NITF Filter configuration file.
Osmastermapfilterencoding.csv	OS MasterMap filter text encoding conversions configuration file.
OSMasterMap.ini	OS Mastermap configuration file.
OSVectorMapLocal.ini	OS Vector Map configuration file.
OSVectorMapDistrict.ini	OS Vector Map District configuration file.
Portrait.tpt	Print Template
stdconfig.pth	Default pathlist
mapviewerlayout.dat	MapViewer default panel layout.
Mapviewer.ini	Map Viewer configuration file.
Mifinteroperability.ini	MIF Filter configuration file.
PrintTemplateStudio.ini	Print Template Studio configuration file.
ptlayout.dat	Print Template Studio default layout.
stlayout.dat	Symbol Studio default layout.
SymbolTool.ini	Symbol Studio configuration file.
TerrainViewer.ini	Terrain Viewer configuration file.
vpffield.ini	VPF Filter configuration file.

4.4.1 Standard Resource files

The following directories and files are normally required for a deployment.

There is considerable scope for reducing the number of symbols deployed in the symbols directory to just the ones you use in your application and maps.

File/Directory	Explanation
<i>fillstyles</i>	MapLink Pro Fillstyle configuration files.
<i>Fonts</i>	Additional fonts used by MapLink Pro.
<i>Linestyles</i>	MapLink Pro Linestyle configuration files.
<i>Symbols</i>	MapLink Pro symbol configuration files and symbols.
tslandroidfonts.dat	Android font list
tscolours.dat	Colours list
tsfillstyles.dat	Fillstyle list
tslfonts.dat	Windows font list
tsllinestyles.dat	Linestyle list
tslsymbols.dat	Standard symbol list
tslsymbolsAPP6A.dat	APP6A symbology setup (required for APP6A symbols)
tsltransforms.dat	Projections
tslunixbitmapfonts.dat	X11 bitmap font list
tslunixfonts.dat	X11 xft font list

4.4.2 OGC Services

Example configuration files for WMS Server and WPS Server services.

4.4.3 3D

You only need to deploy the following resources if you use the 3D SDK. The `earth_images` are optional.

File/Directory	Explanation
<i>models</i>	3D-SDK 3DS models.
<i>Earth_images</i>	Alternative 3D backdrop images.
tslmodels.dat	3D models list.
earth.png	Default 3D backdrop image

4.4.4 Error Message Files

The error message files are used to provide additional error information rather than just an error number.

You only need to deploy the message files you need as the runtime will skip any that are missing.

File/Directory	Explanation
allmaplinkerrors.msg	Top level error message file which pulls in all the other error message files. Required if you need a more meaningful error message.
tsl3dsdkerrors.msg	3D SDK
ptmsgs.msg	Print Template Studio interaction messages
stmsgs.msg	Symbol Studio interaction messages
featuresymbologyhelpererrors.msg	Feature Symbology SDK
owscontexterrors.msg	OWSContext SDK
tsl3dsdkerrors.msg	3D SDK
tslacceleratorsdk.msg	Accelerator SDK
tslasrpexportersdkerrors.msg	ASRP Exporter SDK
tslcadrgexportersdkerrors.msg	CADRG Exporter SDK
tsldbdatalayererrors.msg	Database Layer SDK
tsldbifjobqueueerrors.msg	DBIF and Database Layer SDK
tsldbiferrors.msg	DBIF SDK
tslfilterdatalayererrors.msg	Filter Datalayer
tslfiltererrors.msg	Filter's
tsllandlinkdigmconverrors.msg	LandLink DIGM Converter
tslmaplink3dimodeerrors.msg	3D SDK Interaction modes
tslmaplinkddoerrors.msg	Dynamic Data Object SDK
tsltrackmanagererrors.msg	Track Manager SDK
tslmaplinkedterrors.msg	Editor SDK
tslmaplinkentitystoreerrors.msg	Entity Store SDK
tslmaplinkerrors.msg	MapLink: API errors
tslmaplinkgeopackageerrors.msg	GeoPackage SDK
tslmaplinkgmlerrors.msg	GML SDK
tslmaplinkimodeerrors.msg	MapLink: Interaction modes
tslmaplinkopenglsurfaceerrors.msg	OpenGL drawing surface

File/Directory	Explanation
tslmaplinkspatialerrors.msg	LandLink SDK
tslmaplinkwfsclienterrors.msg	WFS Client SDK
tslmetadataerrors.msg	MapLink: Metadata
tslnetworksdkerrors.msg	Network SDK
tslogfiltererrors.msg	OGC Filter SDK
tslopengldataoptimisererrors.msg	OpenGL Data Optimiser SDK
tslasterprojectionerrors.msg	MapLink: Raster reprojection
tsls52errors.msg	S-52 SDK
tsls63errors.msg	S-63 SDK
tslsatellitepropagatorerrors.msg	Satellite Propagator SDK
tslterrainsdkerrors.msg	Terrain SDK
tsltgmerrors.msg	MapLink: TGM
tsltmserrors.msg	MapLink: TMS
tsltransformerrors.msg	MapLink: Transforms (projections)
tsltrasherrors.msg	MapLink: Raster Handling

5 WINDOWS REQUIRED LIBRARIES

This section describes which MapLink Pro libraries should be shipped when deploying an application based upon MapLink Pro. It is important to note however that you may not copy or redistribute any file supplied with MapLink Pro without first having obtained the proper licence. Please refer to section 2 for further details. These files are referred to as «Deployable Components» in the Licence Agreement.

Only the release configuration of the MapLink Pro C++ libraries can be deployed along with your application. Deployed debug libraries will be unusable due to licence key restrictions imposed on them. Debug libraries can usually be identified by the 'd' suffix in their filename. MapLink Pro does not ship with Debug versions of the .NET libraries; therefore this restriction is not applicable.

All MapLink Pro libraries that are to be deployed with an application should be copied from the 'redist' directory of the MapLink Pro installation.

The tables later in this section summarise the files you may need to include in your application to support the MapLink Pro functions. Some of the libraries listed are required for all applications, whilst others are only required if certain functionality or SDKs are used. If a library is not listed, then it is likely that it may not be redistributable. If you are unsure as to whether a library may be redistributed, then please contact Envitia Support.

If any issues are found when running a deployed application on Windows, then the Microsoft supplied '[Dependency Walker](#)' application can be used to determine any missing DLLs or version mismatches.

5.1 MapLink Libraries

NOTE: The Microsoft Visual C/C++ runtime libraries from section 5.4 are required if using any MapLink Library.

If you have a licence for an SDK you can use either the C++ or .NET variants of the SDK.

If you are experiencing difficulties in working out which DLLs your application requires we would recommend using [Dependency Walker](#) and either loading a DLL or profiling your application to obtain a list of DLLs. This however may not pick up all required DLLs as the Leadtools DLLs are loaded on use (see section 5.3.1 for additional details).

5.1.1 The MapLink Pro Toolkit

Library	Description:	Notes
Envitia Core Mapping DLLs (Core SDK)		
TTLTGM64.DLL TTLTMS64.DLL TTLCH64.DLL TTLTXF64.DLL TTLMT64.DLL TTLCOMPRESSION64.DLL MAPLINK64.DLL	Envitia Core Mapping DLLs. Requires the Intel Thread Building Blocks DLLs from section 5.5. Required by every application dependent on MapLink Pro.	
TTLCLSWF64.DLL TTLCLSSTRK64.DLL	Custom Line Style DLLs. Required if using the standard configuration files.	
MAPLINKMETADATA64.DLL	Envitia Metadata SDK DLL. Required if using the Metadata SDK or Direct Import.	
TPLLBCURL64.DLL	cURL library. Used by the TSLRemoteLoader (WMS, WMTS Data-layers) Required by every application dependent on MapLink Pro.	

<p>boost_atomic-vc140-mt-1_62.dll boost_chrono boost_container boost_context boost_coroutine boost_date_time boost_fiber boost_filesystem boost_graph boost_iostreams boost_locale boost_log_setup boost_log boost_math_c99f boost_math_c99l boost_math_c99 boost_math_tr1f boost_math_tr1l boost_math_tr1 boost_prg_exec_monitor boost_program_options boost_python boost_random boost_regex boost_serialization boost_signals boost_system boost_thread boost_timer boost_type_erasure boost_wave</p>	<p>Boost 1.62 libraries.</p> <p>Not all the boost libraries are required. We would recommend using dependency walker to reduce the number of actual libraries you use for your installation. In future we may use more boost libraries.</p> <p>Required by every application dependent on MapLink Pro.</p>	
---	---	--

Library	Description:	Notes
boost_wserialization		
TPLZLIB64.DLL	Zlib library. Required by TTLCOMPRESSION64.DLL Required by every application dependent on MapLink Pro.	
TTLCRYPT64.DLL	Crypt library. Required by TTLTMS64.DLL Required by every application dependent on MapLink Pro.	
TPLXMLEXPAT64.DLL	Expat PP library and Expat Required by TTLTGM64.DLL Required by every application dependent on MapLink Pro.	
TPLGEOGRAPHIC64.DLL	GeographicLib library. Required by TTLTXF64.DLL Required by every application dependent on MapLink Pro.	
TPLGDAL20164.DLL	GDAL/OGR Library Required by TTLTXF64.DLL Required by every application dependent on MapLink Pro.	See section 5.1.20 and 4.3. Note: plugins are not required for basic functionality.
OpenGL Drawing Surface API		
MAPLINKOPENGLSURFACE64.DLL	Envitia OpenGL Drawing Surface. Dependent upon 'Envitia Core Mapping DLLs' Required by every application using the OpenGL Drawing surface.	
TPLFREETYPE64.DLL	Freetype2 library. Required by MAPLINKOPENGLSURFACE64.DLL Required by every application using the OpenGL Drawing surface.	

Library	Description:	Notes
TPLFRIBIDI64.DLL	GNU Fribidi library. Required by MAPLINKOPENGLSURFACE64.DLL Required by every application using the OpenGL Drawing surface.	
TPLHARFBUZZ-NG64.DLL	Harfbuzz library. Required by MAPLINKOPENGLSURFACE64.DLL Required by every application using the OpenGL Drawing surface.	
TPLTESSELATOR64.DLL	OpenGL Tesselator library (1.2.1.0). Required by MAPLINKOPENGLSURFACE64.DLL Required if using the OpenGL Drawing surface, 3D SDK or Accelerator SDK.	
TPLGLEW64.DLL	GLEW library. Required by MAPLINKOPENGLSURFACE64.DLL Required if using the OpenGL Drawing surface, 3D SDK or Accelerator SDK.	
OpenGL Data Optimiser API		
MapLinkOpenGLDataOptimiser64.DLL	MapLink OpenGL Data Optimiser Dependent upon 'Envitia Core Mapping DLLs' Dependent upon 'Envitia OpenGL Drawing Surface DLL'	
TPLLIBSQUISH64.DLL	libsquish Required by MapLinkOpenGLDataOptimiser64.DLL	
OpenGL Track Helper API		
MAPLINKOPENGLTRACKHELPER64.DLL	Envitia OpenGL Track Helper. Dependent upon 'Envitia Core Mapping DLLs' Dependent upon 'Envitia OpenGL Drawing Surface DLL' Required by every application using the OpenGL Track Helper.	
Direct Import SDK		
MapLinkDirectImport64.dll	Envitia Direct Import SDK	

Library	Description:	Notes
MapLinkDirectImport_gdal64.dll	GDAL/OGR Direct Import Plugin	Requires GDAL/OGR. Must be placed in <bin64>/plugins/directimport Requires directimport configuration files
MapLinkDirectImport_filterNITF64.dll	NITF Vector Direct Import Plugin	Requires the MapLink NITF filter. Must be placed in <bin64>/plugins/directimport Requires directimport configuration files
MapLink WMTS Data Layer DLLs		
MAPLINKWMTSDATALAYER64.DLL TTLWMTSCLIENT64.DLL TTLWMTS64.DLL	Envitia Web Map Tile Service (WMTS) DLLs. MAPLINKOWS64.DLL is required as a dependency of the data layer. Required if using the MapLink TSLWMTSDataLayer	While you require MAPLINKOWS64.DLL you do not require a MapLinkOGC SDK license for using the WMTS DataLayer with the Core SDK.
Additional Core Support DLLs (Core SDK)		
MAPLINKIMODE64.DLL	Envitia Interaction Mode DLL. Required if using the MapLink Interaction Modes SDK.	
MAPLINKTHREADEDMAPCACHE64.DLL	Envitia Threaded Map Cache SDK DLL. Required if using the Threaded Map Cache SDK.	
MAPLINKTIME64.DLL	Envitia Time SDK DLL. Required if using the Time SDK.	
RENDERINGATTRIBUTE64.DLL	Envitia Rendering Attribute Panel SDK. Required if using the Rendering Attribute Panel SDK.	

Library	Description:	Notes
MAPLINK2DKML64.DLL TPLKMLDRAWING64.DLL	Envitia 2D KML Data Layer Required if using the MapLink TSLKMLDataLayer	
Dynamic Data Object SDK		
MAPLINKDDO64.DLL	Envitia Dynamic Data Object SDK DLL. Required if using the Dynamic Data Objects SDK.	
Direct Import (Basic) – GeoTIFF, MIF, NITF, NTF, OS MasterMap & Vector Map Local, Raster, Shapefile, S57		
TTLRPJ64.DLL TTLGENERICFILTER64.DLL	Base Envitia Filter DLLs. The Metadata SDK DLL, MAPLINKMETADATA64.DLL, is required. The Terrain SDK DLL, TTLTERRAIN64.DLL, from section 5.1.4 is required. The Raster Handling Libraries from section 0 are required by the RASTERFILTER64.DLL and the NITFFILTER64.DLL. The GDAL libraries from section 5.1.20 are required. Require if using any MapLink Direct Import Capabilities	While you require TTLTERRAIN64.DLL & TTLGTS.DLL you do not require a Terrain SDK license for using Direct Import with the Core SDK.
GEOTIFFFILTER64.DLL	Required for GeoTIFF direct import via the <code>TSLRasterFilterDataLayer</code> The Raster Handling Libraries from section 0 are required by the GEOTIFFFILTER64.DLL	
MIFFILTER64.DLL	Required for MIF direct import via the <code>TSLUtilityFunctions::importData, appendData, exportData</code> .	Runtime locked. Contact support for unlock code.
RASTERFILTER64.DLL	Required for Raster direct import via the <code>TSLRasterFilterDataLayer</code> . The Raster Handling Libraries from section 0 are required by the RASTERFILTER64.DLL	
SHPFILTER64.DLL	Required for Shapefile direct import via: <code>TSLUtilityFunctions::importData, appendData, exportData</code> .	Runtime locked. Contact support code unlock code.
NTFFILTER64.DLL	Required for OS NTF direct import via: <code>TSLUtilityFunctions::importData, appendData, exportData</code>	Runtime locked.

Library	Description:	Notes
NITFFILTER64.DLL	Required for NITF / NSIF direct import via the <code>TSLNITFFilterDataLayer</code> . The Raster Handling Libraries from section 0 are required by the NITFFILTER64.DLL	Additional Licence. Runtime locked.
S57FILTER64.DLL	Required for S57 direct import via: <code>TSLUtilityFunctions::importData, appendData, exportData</code>	Additional License. Runtime locked.
OSMASTERMAPFILTER64.DLL	Required for OS Master Map direct import via: <code>TSLUtilityFunctions::importData, appendData, exportData</code> Required for OS Vector Map Local direct import via: <code>TSLUtilityFunctions::importData, appendData</code>	Runtime locked. Contact support for unlock code.
OGRFILTER64.DLL	Required for KML direct import via: <code>TSLUtilityFunctions::importData, appendData, exportData</code> Requires GDAL Filter DLLs from section XXXXX	
MapLink Application Framework SDK (MAF)		
MAPLINKMAF64.DLL MAF64.DLL MAFBD64.DLL	Envitia MapLink Application Framework SDK DLLs. Required if using the MapLink Application Framework SDK.	
Database Interfaces SDK (DBIF)		
MAPLINKDBIF64.DLL TTLDBIFS64.DLL	Envitia Database Interfaces (DBIF) SDK DLLs. Requires TPLXMLXERCES64.DLL from section 5.5.	
TTLDBIFSORA64.DLL	Envitia Database Interfaces Oracle Database Plug-in. Requires the Oracle 10g Database Client libraries to be installed. Required if using the DBIF SDK to connect to Oracle Databases.	
MAPLINKENTITYSTORE64.DLL	Envitia Entity Store SDK DLL. Requires MAPLINKDBIF64.DLL and TTLDBIFS64.DLL. Required if using the Entity Store SDK.	

Library	Description:	Notes
MAPLINKDATABASELAYER64.DLL MAPLINKDBIFJOBQUEUE64.DLL	Envitia Database Layer SDK. Requires MAPLINKENTITYSTORE64.DLL. Required if using the Database Layer SDK.	

5.1.2 GeoPackage

Library	Description:	Notes
MapLink Pro Toolkit	The SDK requires the MapLink Pro Toolkit.	
MAPLINKGEOPACKAGE64.DLL	Envitia GeoPackage SDK DLL.	
TPLSQLITE64.DLL	SQLite 3.14.2.0	

5.1.3 MapLink Pro Accelerator SDK

This is a separately licensed component.

Library	Description:	Notes
MapLink Pro Toolkit	The SDK requires the MapLink Pro Toolkit.	
MAPLINKACCELERATOR64.DLL	Envitia OpenGL Accelerator SDK DLL. Requires TLTESSELATOR.DLL from section 0. Requires GLEW32.DLL from section 5.5. Required if using the OpenGL Accelerator SDK.	
MAPLINKDXACCELERATOR64.DLL	Envitia DirectX Accelerator SDK DLL. Requires the DirectX 9.0c Runtime. Required if using the DirectX Accelerator SDK and wish to use the original DirectX runtime.	
TPLGLEW64.DLL	GLEW library. Required by MAPLINKACCELERATOR64.DLL Required if using the OpenGL Drawing surface, 3D SDK or Accelerator SDK.	

5.1.4 MapLink Pro Terrain SDK

This is a separately licensed component.

Library	Description:	Notes
MapLink Pro Toolkit	The SDK requires the MapLink Pro Toolkit.	
TTLTERRAIN64.DLL TPLGTS64.DLL TPLGLIB64.DLL	Envitia Terrain SDK DLL. Requires Envitia Core MapLink DLLs.	
Part of Terrain SDK	Direct Import DMED	Additional Licence. Runtime locked.

5.1.5 MapLink Pro 3D SDK

This is a separately licensed component.

Library	Description:	Notes
MapLink Pro Toolkit	The SDK requires the MapLink Pro Toolkit.	
MapLink Pro Terrain SDK	The SDK requires the MapLink Pro Terrain SDK.	
MAPLINK3D64.DLL	Envitia 3D Mapping DLL. Requires GLEW32.DLL from section 5.5. Required if using the MapLink 3D SDK.	
MAPLINK3DIMODE64.DLL	Envitia 3D Interaction Mode DLL. Required if using the MapLink 3D Interaction Modes SDK.	
MAPLINKCAMERAMANAGER64.DLL	Envitia Camera Manager DLL. Requires MAPLINKTIME.DLL from section 5.1.1 and MAPLINK3D64.DLL. Required if using the Camera Manager SDK.	
TPLTESSELATOR64.DLL	OpenGL Tesselator library (1.2.1.0). Required by MAPLINK3D64.DLL Required if using the OpenGL Drawing surface, 3D SDK or Accelerator SDK.	
TPLGLEW64.DLL	GLEW library. Required by MAPLINK3D64.DLL Required if using the OpenGL Drawing surface, 3D SDK or Accelerator SDK.	
TPL3DS64.DLL	Sample 3D Model DLLs that may be redistributed. The source code for this library is provided as a sample. Requires TPLLIB3DS-1_364.DLL from section 5.5. Required if using 3D Models.	

5.1.6 MapLink Pro Satellite SDK

This is a separately licensed component.

Library	Description:	Notes
MapLink Pro Toolkit	The SDK requires the MapLink Pro Toolkit.	
MAPLINKSATELLITEPROPAGATOR64.DLL	Envitia Satellite Propagator SDK DLLs. Requires MAPLINKTIME.DLL from section 5.1.1.	
MAPLINKSATELLITEPROPAGATOR3D64.DLL	Envitia Satellite Propagator 3D SDK DLLs. Requires MAPLINK3D64.DLL from section 0 and MAPLINKSATELLITEPROPAGATOR64.DLL.	3D SDK required.

5.1.7 MapLink Pro S-63 SDK

This is a separately licensed component.

Library	Description:	Notes
MapLink Pro Toolkit	The SDK requires the MapLink Pro Toolkit.	
MAPLINKS6364.DLL	Envitia S-63 SDK DLL. Requires S57FILTER64.DLL from section 5.1.1. Required if using the S63 SDK.	

5.1.8 MapLink Pro S-52 SDK

This is component requires data to have been imported via the S-57 Filter. This can be achieved either by MapLink Studio, S-63 SDK or direct import of S-57. Please refer to the 'MapLink S-63 SDK and S-52 SDK Developers Guide' (UUM1146-XX).

Library	Description:	Notes
MapLink Pro Toolkit	The SDK requires the MapLink Pro Toolkit.	
MAPLINKS5264.DLL	Required if using the S52 Dynamic Renderer SDK.	

5.1.9 MapLink Pro Network SDK

This is a separately licensed component.

Library	Description:	Notes
MapLink Pro Toolkit	The SDK requires the MapLink Pro Toolkit.	
MAPLINKNWK64.DLL	Envitia Network SDK DLL. Required if using the Network SDK.	
MAPLINKOPTNWK64.DLL	Envitia Optimised Network SDK DLL. Requires MAPLINKNWK64.DLL. Required if using the Optimised Network SDK.	
MAPLINKISOCHRONE64.DLL	Envitia Isochrone SDK DLL Requires MAPLINKOPTNWK64.DLL. Required if using the Isochrone SDK.	

5.1.10 MapLink Pro Editor SDK

This is a separately licensed component.

Library	Description:	Notes
MapLink Pro Toolkit	The SDK requires the MapLink Pro Toolkit.	
APP6A64.DLL	Envitia Editor APP-6A SDK DLL. Required if using the APP6A classes from the Editor SDK	
MAPLINKEDT64.DLL	Envitia Editor SDK DLL. Required if using the Editor SDK.	
MAPLINKEDTIMODE64.DLL	Envitia Editor Interaction Mode DLL. Requires MAPLINKEDT64.DLL Required if using the Editor Interaction Modes SDK.	
MAPLINKEDTMAF64.DLL	Envitia Editor MapLink Application Framework DLL. Requires MAPLINKEDT64.DLL, MAPLINKMAF64.DLL, MAF64.DLL, MAFBD64.DLL. Required if using the Editor MapLink Application Framework SDK.	Requires MAF SDK
MAPLINKDATABASELAYEREDITORINTERF ACE64.DLL	Envitia Editor SDK Interface to the Database Layer SDK. Requires MAPLINKEDT64.DLL, MAPLINKDATABASELAYER64.DLL and MAPLINKDBIFJOBQUEUE64.DLL. Required if using the Editor SDK Interface to the Database Layer SDK.	Requires DBIF and Database Layer.

5.1.11 MapLink Pro Spatial Editor SDK (LandLink)

This is a separately licensed component.

Library	Description:	Notes
MapLink Pro Toolkit	The SDK requires the MapLink Pro Toolkit.	
LANDLINK64.DLL	Envitia Spatial Editor SDK DLL. Requires MAPLINKEDT64.DLL from section 5.1.10. Required if using the Spatial Editor SDK.	
LANDLINKQAS64.DLL	Envitia Spatial Editor Quick Address Search SDK DLL. Required if using the Spatial Editor Quick Address Search SDK.	

5.1.12 MapLink LandLink DIGM Convertor SDK

This is a separately licensed component.

Library	Description:	Notes
MapLink Pro Toolkit	The SDK requires the MapLink Pro Toolkit.	
LANDLINKDIGMCONV64.DLL	Envitia Spatial Editor DIGM Converter SDK DLL.	

5.1.13 MapLink Impact Assessor SDK

This is a separately licensed component.

Library	Description:	Notes
MapLink Pro Toolkit	The SDK requires the MapLink Pro Toolkit.	
MapLink Pro Spatial Editor SDK	The SDK requires the MapLink Pro Spatial Editor SDK	
MAPLINKIMPASS64.DLL	Envitia Impact Assessment SDK DLL. Requires MAPLINKDBIF64.DLL and TTLDBIFS64.DLL from section 5.1.1 and LANDLINK64.DLL from section 5.1.11,	

5.1.14 MapLink CADRG Exporter SDK

This is a separately licensed component.

Library	Description:	Notes
MapLink Pro Toolkit	The SDK requires the MapLink Pro Toolkit.	
MAPLINKCADRGEXPORT64.DLL	Envitia CADRG Exporter SDK DLL. Required if using the CADRG Exporter SDK.	

5.1.15 MapLink ASRP Exporter SDK

This is a separately licensed component.

Library	Description:	Notes
MapLink Pro Toolkit	The SDK requires the MapLink Pro Toolkit.	
MAPLINKASRPEXPORT64.DLL TTLISO8211FILE64.DLL	Envitia ASRP Exporter SDK DLL. Requires MAPLINKCADRGEXPORT64.DLL from section 5.1.14. Required if using the ASRP Exporter SDK.	

5.1.16 MapLink Track Manager SDK

This is a separately licensed component.

Library	Description:	Notes
MapLink Pro Toolkit	The SDK requires the MapLink Pro Toolkit.	
MAPLINKTRACKMANAGER64.DLL	Envitia MapLink Track Manager SDK DLL. Required if using the Track Manager SDK.	
TTLTRACKDRAWINGUTILITY_GDI64.DLL	Track drawing utility required if using GDI drawing surfaces. Required if using the Track Manager SDK.	
TTLTRACKDRAWINGUTILITY_GL64.DLL	Track drawing utility required if using OpenGL drawing surfaces. Required if using the Track Manager SDK.	

5.1.17 MapLink OGC SDKs

This is a separately licensed component.

Library	Description:	Notes
MapLink Pro ToolKit	The SDK requires the MapLink Pro Toolkit.	
MAPLINKGML64.DLL	Envitia GML SDK DLL. Requires TPLXMLXERCES64.DLL from section 5.5. Required if using the GML SDK.	Additional Licence. Runtime locked.
MAPLINKWFSCLIENT64.DLL MAPLINKOGCFILTER64.DLL	Envitia WFS Client SDK DLLs. Requires MAPLINKGML64.DLL and MAPLINKOWS64.DLL. Required if using the WFS Client SDK.	Additional Licence. Runtime locked.
MAPLINKOWS64.DLL	Envitia OWS Support Client DLL. Required if using the WFS Client SDK or WMTS Data Layer.	

5.1.18 MapLink OGC Services

Please refer to the 'MapLink OGC Services Deployment Users Guide' (UUM1106-XX) as additional configuration is required over and above deploying the required DLLs.

The MapLink OGC Services require a server license.

Library	Description:	Notes
MapLinkWPS64.DLL	MapLink Web Processing Service (WPS) SDK (See the Developers Guide) Depends on MapLink Core, MAPLINKOWS64.DLL, MAPLINKOGCSERVICES64.DLL	
MAPLINKWMS64.DLL	MapLink Web Map Service (WMS) SDK (See the Developers Guide) Depends on MapLink Core, MAPLINKOGCSERVICES64.DLL	
CADRGMAPPLUGIN64.DLL	MapLink CADRG WMS Plugin See the 'MapLink CADRG WMS Plug-in User Documentation' (UUM1155-xx) Depends on MAPLINKWMS64.DLL	Additional licensed Component
BASICMAPPLUGIN64.DLL	MapLink Basic WMS plugin. Depends on MAPLINKWMS64.DLL	Deprecated, use the SuperMap Plugin.
SUPERMAPPLUGIN64.DLL	MapLink WMS Super Map plugin Depends on MapLink Core, MAPLINKWMS64.DLL, MAPLINKTHREADEDMAPCACHE64.DLL, TTLXALANHELPERS64.DLL, Intel TBB, TPLXMLXERCES64.DLL, XLANA-C_1_11.DLL	
HISTORICALMAPPLUGIN64.DLL	MapLink Historical Map plugin Depends on MapLink Core, MAPLINKWMS64.DLL.	Deprecated, use the SuperMap Plugin.
MAPLINKOGCSERVICES64.DLL	MapLink OGC Services	
TTLXALANHELPERS64.DLL	MapLink helper methods for XALAN	
TPLXMLXERCES64.DLL	Apache Xerces library.	
TPLXALAN-C_1_1164.DLL	Apache Xalan-C library.	
TTLWPSUTILITIES64.DLL	MapLink WPS Plugin support functions/objects Depends on MapLink Core, MAPLINKWPS64.DLL, MAPLINKOWS64.DLL, MAPLINKGML64.DLL, TPLGDAL20164.DLL	Additional licensed Component

Library	Description:	Notes
VIEWSHEDWPSPLUGIN64.DLL	MapLink WPS View Shed Plugin Depends on MapLink Core, MAPLINKWPS64.DLL, MAPLINKOGCSERVICES64.DLL, MAPLINKOWS64.DLL, MAPLINKGML64.DLL, TTLTERRAIN64.DLL, TTLWPSUTILITIES64.DLL, TPLGDAL20164.DLL	Additional licensed Component
ROUTERWPSPLUGIN64.DLL	MapLink WPS Router Plugin Depends on MapLink Core, MAPLINKWPS64.DLL, MAPLINKOGCSERVICES64.DLL, MAPLINKOWS64.DLL, MAPLINKGML64.DLL, TTLWPSUTILITIES64.DLL, TPLGDAL20164.DLL, MAPLINKOPTNWK64.DLL	Additional licensed Component
TERRAINPROFILEWPSPLUGIN64.DLL	MapLink WPS TerrainProfile Plugin Depends on MapLink Core, MAPLINKWPS64.DLL, MAPLINKOGCSERVICES64.DLL, MAPLINKOWS64.DLL, MAPLINKGML64.DLL, TTLTERRAIN64.DLL, TTLWPSUTILITIES64.DLL, TPLGDAL20164.DLL, MAPLINKOPTNWK64.DLL	Additional licensed Component

5.1.19 MapLink Filter DLLs

You do not normally need to deploy the MapLink Filter DLLs unless you have a MapLink Studio OEM license or a license for a specific direct import format (please see section 5.1.1).

The Filter DLLs use the TTLGenericFilter.DLL (see section 5.1.1, Direct Import) which uses the GDAL DLL (see section 5.1.20).

Library	Description:	Notes
OGRFilter64.DLL	OGR Filter	Direct Import (KML)
ADRGFilter64.DLL	ADRG Filter	
ASCIIDEMFilter64.DLL	ASCII DEM Filter	
ASRPFilter64.DLL	ASRP Filter	
CIBFilter64.DLL	CIB/CADRG Filter	

Library	Description:	Notes
CRPFilter64.DLL	CRP Filter	
DAFIFFilter64.DLL	DAFIF Filter	
JeppesenFilter64.DLL	Jeppesen Filter	
ArcsFilter64.DLL	ARCS Filter	
ASCIIFilter64.DLL	Envitia ASCII Filter	
BLFilter64.DLL	Boundary Line Filter	
DBDBVFilter64.DLL	DBDBV versions 1-4 Filter	
DBDBV6Filter64.DLL	DBDBV version 5 Filter	
DBDBV5Filter64.DLL	DBDBV version 6 Filter	
DFADFilter64.DLL	DFAD Filter	
DTEDFilter64.DLL	DTED Filter	
DXFFilter64.DLL	DXF Filter	
MIFFilter64.DLL	MIF Filter	
NTFFilter64.DLL	NTF Filter	
RasterFilter64.DLL	Raster Filter	Direct Import
S57Filter64.DLL	S-57 Filter	Direct Import/Export (Licensed)
SHPFilter64.DLL	Shapefile Filter	Direct Import/Export
VPFFilterPlus64.DLL	VPF plus Filter	
KMLFilter64.DLL	KML Export Filter	Locked
NITFFilter64.DLL	NITF/NSIF 2.1 Filter	Direct Import (Licensed)
ArcGridFilter64.DLL	ArcGrid ASCII/Binary Filter	
OSMasterMapFilter64.DLL	OS Mastermap Filter	Direct Import/Export
GDALFilter64.DLL	GDAL Filter	
OSCARFilter64.DLL	OSCAR Filter	

Library	Description:	Notes
OGRFilter64.DLL / OldOGRFilter64.DLL	OGR Filter	
GeoTiffFilter64.DLL	GeoTIFF Filter	Direct Import
GPKGFilter64.DLL	GeoPackage Filter	
TPLHDF564.DLL	DBDBV version 5 and older Filter dependency	
TPLHDF5-1-864.DLL	DBDBV version 6 Filter dependency	

5.1.20 GDAL/OGR DLLs

The MapLink Core (see section 5.1.1) and GDAL/OGR filters are dependent upon the following DLLs.

GDAL/OGR is dependent upon configuration files, please see section 4.3 for additional deployment information.

Please see the next section 5.1.21 for Geospatial PDF dependencies.

Library	Description:	Notes
TPLGDAL20164.DLL	GDAL version 2.1.2.0 Used by the OGRFilter64.DLL and GDALFilter64.DLL	
TPLPODOFO64.DLL	Podof0 0.9.2.0	
TPLPROJ_464.DLL	Proj.4 4.8.0	
TPLXPAT64.DLL	Expat 2.1.0	
TPLFREETYPE64.DLL	Freetype 2.5.1	
TPLLIBTIFF64.DLL	LibTIFF 4.0.4	
TPLSQLITE6464.DLL	SQLite 3.18.0	
TPLLIBCURL64.DLL	LibcURL 7.50.3	
gdal_MrSID.DLL	The directory hierarchy must be the same as the architecture being deployed, eg. Bin64/gdalplugins/2.1/, for x86. MrSID plugin for GDAL.	End user has to install 3 rd Party Libraries. Version 9.5.1.4427
gdal_MG4Lidar.DLL	The directory hierarchy must be the same as the architecture being deployed, eg. Bin64/gdalplugins/2.1/, for x86. MG4Lidar plugin for GDAL.	End user has to install 3 rd Party Libraries. Version 9.5.1.4427
gdal_ECW_JP2ECW.dll	The directory hierarchy must be the same as the architecture being deployed, eg. Bin64/gdalplugins/2.1/, for x86. ECW plugin for GDAL.	End user has to install 3 rd Party Libraries. Version 5.3.0 SDK.

5.1.21 PDFTOPPM.EXE

The executable is used by the GDAL filter and Direct Import SDK on Windows for converting PDFs to raster.

Library	Description:	Notes
PDFTOPPM.EXE	Poplar: Standard Win32 build – with no source code changes. Depends on the DLLs below	GPL license
TPLOPENJPEG.DLL	OpenJPEG	
TPLFREETYPE64.DLL	Freetype	
TPLLIBJPEG64.DLL	LibJPEG	
TPLLIBTIFF64.DLL	LibTIFF	
TPLLIBPNG64.DLL	LibPNG	

5.2 .NET Assemblies

MapLink supplies .NET assemblies built using the Microsoft .NET version 4.0 Framework.

You will need the MapLink DLLs listed in section 5.1 for the .NET SDKs you are using below.

If you have a licence for an SDK you can use either the C++ or .NET variants of the SDK.

Library	Description:
Envitia.MAPLINK64.DLL Envitia.MapLinkEx64.DLL ttlmaplink.net64.DLL Envitia.MapLink.NativeHelpers64.DLL	Envitia Core .NET Assembly. Required by every application using the MapLink .NET interfaces. If using the accelerated surface: TTLDXAcceleratorShim64.DLL is required in addition to all the dependencies of the Envitia DirectX Accelerator SDK DLL, see 6.2.3.
Envitia.MAPLINK.EDITOR64.DLL	Envitia Editor .NET Assembly. Requires MAPLINKEDT64.DLL and APP6A64.DLL. Required if using the Editor .NET SDK. Additional Licensed Component.
Envitia.MAPLINK.SPATIAL64.DLL	Envitia Spatial Editor .NET Assembly. Requires LANDLINK64.DLL and Envitia.MAPLINK.EDITOR64.DLL. Required if using the Spatial Editor .NET SDK. Additional Licensed Component.
Envitia.MAPLINK.DDO64.DLL	Envitia Dynamic Data Object .NET Assembly. Requires MAPLINKDDO64.DLL. Required if using the Dynamic Data Object .NET SDK.
Envitia.MAPLINK.INTERACTIONMODES64.DLL	Envitia Interaction Modes .NET Assembly. Requires MAPLINKIMODE64.DLL. Required if using the Interaction Modes .NET SDK.
Envitia.MAPLINK.TERRAIN64.DLL Envitia.MapLink.TerrainEx64.DLL ttlmaplinkterrain.net64.DLL	Envitia Terrain .NET Assembly. Requires TTLTERRAIN64.DLL. Required if using the Terrain .NET SDK. Additional Licensed Component.
Envitia.MAPLINK.S5264.DLL	Envitia S52 .NET Assembly. Requires MAPLINKS5264.DLL. Required if using the S52 .NET SDK.
Envitia.MAPLINK.S6364.DLL	Envitia S63 .NET Assembly. Requires MAPLINKS6364.DLL. Required if using the S63 .NET SDK. Additional Licensed Component.
Envitia.MAPLINK.ML3D64.DLL	Envitia 3D .NET Assembly. Requires Envitia.MAPLINK.TERRAIN64.DLL and MAPLINK3D64.DLL. Required if using the 3D .NET SDK. Additional Licensed Component.

Library	Description:
Envitia.MAPLINK.INTERACTIONMODES.ML3D64.DLL	Envitia 3D Interaction Modes .NET Assembly. Requires Envitia.MAPLINK.ML3D64.DLL and MAPLINK3DIMODE64.DLL. Required if using the 3D Interaction Modes .NET SDK.
Envitia.MAPLINK.NATIVEHELPERS.DLL	Envitia Native Helpers .NET Assembly. Required by every application using the MapLink .NET interfaces.
Envitia.MAPLINK.GEOPACKAGE64.DLL ttlgeopackage.net64.DLL	Envitia GeoPackage .NET Assembly. Requires Envitia.MAPLINK.NATIVEHELPERS.DLL and MAPLINKGEOPACKAGE64.DLL Required if using the GeoPackage .NET SDK
Envitia.MapLink.TrackManager64.DLL ttltrackmanager.net64.DLL	Envitia Track Manager .NET Assembly. Requires Core .NET assemblies and MAPLINKTRACKMANAGER64.DLL and (TTLTRACKDRAWINGUTILITY_GDI64.DLL or TTLTRACKDRAWINGUTILITY_GL64.DLL) Required if using the Track Manager .NET SDK
Envitia.MapLink.DirectImport64.DLL ttldirectimport.net64.DLL	Envitia Direct Import .NET Assembly. Requires Core .NET assemblies and MAPLINKDIRECTIMPORT64.DLL Required if using the Direct Import .NET SDK

5.3 Raster Handling Libraries

This section lists the libraries required for loading, manipulating and saving raster images using MapLink Pro. It should be noted that using MapLink maps that were created from raster source data will require these libraries. Additionally, using symbols that are composed from raster images, e.g. icon symbols, will require these images.

NOTE: The Microsoft Visual C/C++ 9.0 runtime libraries from section 5.4 are required if using the Raster Handling libraries.

5.3.1 Required Libraries

The following table lists the libraries that are required for raster handling.

Leadtools version: 16.5 (patched to 16.5.5.39).

Library	Description:
TTLTRASHLT64.DLL	Envitia Raster-handling DLL.
LTKRNX.DLL	LEADTOOLS kernel.
LTDISX.DLL	LEADTOOLS Display support, required for all MapLink programs using Raster images.
LTWVCAX.DLL LTWVCX.DLL	LEADTOOLS Class Library.
LTFILX.DLL	LEADTOOLS File read and write support.
LTEFXX.DLL	LEADTOOLS Paint effects support.
LTCLR.X.DLL	LEADTOOLS Colour Conversion Library.

Library	Description:
LTIMGCLR.DLL LTIMGCOR.DLL LTIMGEF.DLL LTIMGOPT.DLL LTIMGSF.DLL LTIMGUTL.DLL	LEADTOOLS Image processing support.
LTLSTX.DLL	LEADTOOLS ImageList Control.
LTNETX.DLL	LEADTOOLS Internet Support.
LTTWNX.DLL	LEADTOOLS TWAIN Support.

5.3.2 Optional Libraries

The following table lists the libraries that are needed to support certain raster file formats.

Maps generated from MapLink Studio, containing raster data, may be produced in TIFF, PNG, JPEG or JPEG2000 format depending upon the settings used. It is recommended that at least these raster formats are supported. The libraries required to support these formats are listed with bold descriptions.

Library	Description:
LFAFPX.DLL	AFP file support.
LFANIX.DLL	Windows ANI file support.
LFAVIX.DLL	Windows AVI file support (read only).
LFAWDX.DLL AWCODC32.DLL AWDCXC32.DLL AWDENC32.DLL AWRESX32.DLL AWVIEW32.DLL	Windows FAX (AWD) file support. This format works only on WIN32 systems. The DLLs prefixed with 'AW' are Microsoft DLLs for reading and writing this format.
LFBMPX.DLL	Windows and OS/2 BMP file support, and Windows icon and cursor file support.
LFCALX.DLL	CALS raster file support, which also requires LFFAXX.DLL.
LFCGMX.DLL LVKRNX.DLL	Computer Graphics Metafile file support.
LFCMPX.DLL	JPEG and LEAD CMP file support.
LFCMWX.DLL	LEAD Wavelet file support.
LFDRWX.DLL LVKRNX.DLL	Micrografx Drawing support
LFDWFX.DLL LVKRNX.DLL	AutoCAD DWF file support.
LFDWGX.DLL LFACSX.DLL LVKRNX.DLL	AutoCAD DWG file support.

Library	Description:
LFDXFX.DLL LFACSX.DLL LVKRNX.DLL	AutoCAD DXF file support.
LFEPSX.DLL	Encapsulated PostScript file support, which also requires LFTIFX.DLL for embedded TIFF support.
LFFAXX.DLL	Raw FAX file support.
LFFITX.DLL	Flexible Image Transport System support
LFFLCX.DLL	Flic Animation
LFFPXX.DLL LTFPXX.DLL LVKRNX.DLL	Kodak FlashPix File Support.
LFGBRX.DLL LTKDKX.DLL	Gerber file support.
LFGIFX.DLL	GIF file support.
LFICAX.DLL	IOCA file support, which also requires LFFAXX.DLL.
LFIFFX.DLL	Interchange file support.
LFIMGX.DLL	GEM Image file support.
LFITGX.DLL	Intergraph RLE support, which also requires LFFAXX.DLL.
LFJ2KX.DLL LTJP2X.DLL	JPEG 2000 Support.
LFJB2X.DLL	JBIG 2 file support.
LFJBGX.DLL	JBIG file support.
LFMACX.DLL	File read support.
LFMSPX.DLL	New LEAD 1-Bit file support.
LFPCDX.DLL PCDLIB32.DLL	Kodak PhotoCD file support.
LFPCLX.DLL LVKRNX.DLL	Printer Command Language Format support; Hewlett-Packard Graphics Language for plotters.
LPFCTX.DLL	MacPict file support.
LFPCXX.DLL	ZSoft PCX file support.
LPFNGX.DLL	Portable Network Graphics file support.
LPFNMX.DLL	Portable Bitmap Utilities format file support.
LPFSDX.DLL	Adobe Photoshop 3.0 file support.
LFRASX.DLL	Sun Raster file support.
LFRAWX.DLL	Raw uncompressed file support.
LFSCXX.DLL LTCLRXX.DLL	Scitex Continuous Tone Format support.
LFSGIX.DLL	Silicon Graphics Format support.
LFSPXX.DLL	SMP Format support. This format also requires LFFAXX.DLL.

Library	Description:
LFTGAX.DLL	TARGA file support.
LFTIFX.DLL	Tagged Image File Format support (including TIFF, TIFLZW, and CCITT). This format also requires LFFAXX.DLL.
LFWFXX.DLL	Delrina file support, which also requires LFFAXX.DLL.
LFWMFX.DLL LVKRNX.DLL	Windows Metafile support.
LFWMPX.DLL	Wireless Bitmap file support.
LFWPGX.DLL	WordPerfect file support.
LFXBMX.DLL	X BitMap file support.
LFXPMX.DLL	X PixMap file support.
LFXWDX.DLL	X Window Dump file support.

5.4 Microsoft Runtime Libraries

The following table lists the Microsoft supplied runtime libraries on which MapLink Pro depends.

Visual Studio 2015 complicates the installation of an application. The 'redist' from Microsoft should be installed by the application installer as merge modules are not available. The suggested method is using a WiX bundle to wrap the redist and existing installers.

For the older Visual Studio runtime dependencies users should consider redistributing the libraries via Merge Modules rather than as isolated assemblies.

If there is an associated merge module policy, this should also be installed. Merge module policies are not always shipped with an initial release of Visual Studio.

Library	Description:
Visual Studio 2015 Update 3 version 14.0.24215	Visual Studio 2015 Update 3 runtime DLLs Required by all MapLink libraries.
MSVCP100.DLL MSVCR100.DLL	Microsoft Visual C/C++ 10.0 runtime DLLs. Required by all MapLink libraries.
MFC100.dll MFC100U.dll MFCM100.dll MFCM100U.dll	Microsoft MFC 10.0 runtime DLLs. Required by all MapLink user interface libraries.
MSVCM90.dll MSVCP90.dll MSVCR90.dll Microsoft.VC90.CRT.manifest	Microsoft Visual C/C++ 9.0 runtime DLLs and manifest. Required by raster handling libraries

VS2010 Merge Modules:

- Microsoft_VC100_MFC_x86.msm
- Microsoft_VC100_CRT_x86.msm
- Microsoft_VC100_MFC_x64.msm
- Microsoft_VC100_CRT_x64.msm

- Microsoft_VC100_OpenMP_x86.msm
- Microsoft_VC100_OpenMP_x64.msm

VS2008 Merge Modules:

- Microsoft_VC90_CRT_x86.msm
- Microsoft_VC90_MFC_x86.msm
- policy_9_0_Microsoft_VC90_CRT_x86.msm
- policy_9_0_Microsoft_VC90_MFC_x86.msm
- Microsoft_VC90_CRT_x86_x64.msm
- Microsoft_VC90_MFC_x86_x64.msm
- policy_9_0_Microsoft_VC90_CRT_x86_x64.msm
- policy_9_0_Microsoft_VC90_MFC_x86_x64.msm

5.5 3rd Party Libraries/Executables

The following table lists a number of 3rd Party Libraries on which certain MapLink libraries depend. Usually these should only be deployed if an application uses a library listed in a previous section, which notes that it requires the library.

The 3rd Party libraries that Envitia build on Windows are pre-fixed with 'TPL'. This is to minimise the potential for a name clash with user built libraries, please see the following:

- [http://msdn.microsoft.com/en-us/library/windows/desktop/ms682586\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/ms682586(v=vs.85).aspx)

Please refer to the 'MapLink Pro Windows 3rd Party Licences' (ULA1107-XX) for additional information about the licences for these libraries.

Library	Description:
TBB.DLL TBBMALLOC.DLL TBBMALLOC_PROXY.DLL	Intel Intel® Thread Building Blocks DLLs. Version: 2017 Update 3. Required by every application dependent on MapLink Pro.
TPLGLEW64.DLL	The OpenGL Extension Wrangler Library version 1.13.0. Required by 3D SDK, Accelerator SDK and OpenGL Drawing Surface.
TPLLIB3DS-1_364.DLL	LIB3DS-1_3.DLL is the Lib3ds free toolkit Required by 3D SDK.
TPLTESSELATOR64.DLL	OpenGL Tesselator DLL (1.2.1.0) Required by 3D SDK and OpenGL Drawing Surface.
TPLZLIB64.DLL	Zlib 1.2.8
TPLXMLXERCES64.DLL	Apache Xerces 3.1.1
TPLXALAN-C_1_1164.DLL TPLXALANMSGLIB64.DLL	Apache Xalan-C 1.11
TPLGDAL20164.DLL	GDAL 2.1.2
TPLPODOFO64.DLL	Podof0 0.9.2
TPLPROJ_464.DLL	Proj.4 4.8.0
TPLKDU_v64x64R.DLL	Kakadu 6.4.1 JPEG 2000 support This must not be redistributed.
TPLKDUGDAL64.DLL	Kakadu 6.4.1 JPEG 2000 support. GDAL specific build dependencies. This must not be redistributed.
TPLEXPAT64.DLL	Expat 2.1.0
TPLXMLEXPAT64.DLL	ExpatPP
TPLFREETYPE64.DLL	Freetype 2.5.1
TPLLIBTIFF64.DLL	LibTIFF 4.0.4
TPLOPENJPEG.DLL	OpenJPEG 2.1.2
TPLCRYPT.DLL	CryptoPP 5.6.2
TPLFRIBIDI64.DLL	GNU Fribidi 0.19.5
TPLGEOGRAPHIC64.DLL	GeographicLib 1.34

TPLGLIB64.DLL	Glib 2.38.2
TPLGTS64.DLL	GNU Triangulated Surface library 0.7.6
TPLHARFBUZZ-NG64.DLL	Harfbuzz-ng 0.9.25
TPLFRIBIDI64.DLL	FriBidi 0.19.5.0
TPLHDF564.DLL	Hierarchical Data Format 5 1.6
TPLHDF5-1_864.DLL	Hierarchical Data Format 5 1.8.3
TPLKMLDRAWING64.DLL	FalconView KML Drawing Library
TPLLIBCURL64.DLL	cURL 7.50.3.0
TPLLIBPNG64.DLL	Libpng 1.6.17
TPLSQLLITE64.DLL	SQLite 3.18.0
TPLSQUISH64.DLL	libsquish 0.1.11.0

The following executable is used by GDAL/OGR to convert PDFs to raster:

Executable	Description:
PDFTOPPM.EXE	Poplar: Standard Win32 build – with no source code changes. See section 5.1.21 for dependencies.

5.5.1 GDAL Executables

The following GDAL/OGR executables are shipped as a courtesy with MapLink Pro and can be deployed with your application, please also see section 4.3:

Executable	Description:
dumpoverviews	Dump overviews to external files
gdalflattenmask	Merges regular data bands with the mask bands, for applications not being able to use the mask band concept
gdalbuildvrt	Builds a VRT from a list of datasets
teststepsg	Test executable for translating EPSG to WKT
ogr2ogr	Converts simple features between file formats
ogrtindex	Creates a tile index
ogrinfo	Lists information about an OGR supported data source/set
gdal_grid	Create a raster from scattered data
gdal_rasterize	Convert vector data into a raster
gdal_contour	Create contours from a DEM
gdalsrsinfo	Report an SRS in different formats
gdaltindex	Build a MapServer raster tile index
gdaldem	Tool to analyse and visualise DEMs
gdallocationinfo	Query raster location information
gdaltransform	Transform a list of coordinates

gdalenhance	Prototype
nearblack	Convert nearly black/white borders to exact value
gdalwarp	Warp an image into a new coordinate system (projection)
gdaladdo	Add overviews to a file
gdalinfo	Report information about a file
gdalservice	GDAL API Proxy server
gdal_translate	Copy a raster, with control of the output format

6 .NET DEPLOYMENT ISSUES

6.1 Dependency Loading

Deploying managed applications that reference 'mixed-mode' assemblies, such as the MapLink .NET assemblies, is often found to be difficult. This is most often due to the difference between how dependent .NET assemblies and native DLLs are located by the runtime.

All of the MapLink .NET assemblies depend upon at least some native C++ libraries. If any of these cannot be located, an error will be raised and the application likely terminated.

Furthermore MapLink will attempt, when required, to load other DLLs during the execution of an application, such as the raster handling library TTLTRASHLT.

The easiest and least problematic way around this issue is to place the application and all dependencies in the same directory, although this is not always desirable or possible. In this section we'll outline some of the ways MapLink can be deployed in that scenario, using deployment methods suggested by Microsoft.

The following are the methods we will discuss in subsequent sections:

- Via the `<codeBase>` entries in the application configuration file.
- From the Global Assembly Cache (GAC)
- Using the PATH environment variable

6.1.1 Locating MapLink assemblies via the application configuration file

Given the scenario where the application is located in one place on the file system and the MapLink libraries in another, the MapLink assemblies can be resolved using the application's configuration file, e.g.

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <runtime>
    <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
      <dependentAssembly>
        <assemblyIdentity name="Envitia.MapLink"
                          publicKeyToken="f6586908b95b8151"
                          culture="neutral"/>
        <codeBase version="*.*.*.*"
                  href="file:///C:/Somefolder/Envitia.MapLink64.dll"/>
      </dependentAssembly>
    </assemblyBinding>
  </runtime>
</configuration>
```

Note: The assembly version will need filling in

Assuming all of the dependent MapLink libraries (such as TTLTGM, TTLTMS etc) can be found in the same folder as the assembly or exist in a directory in the system PATH variable, the application should load the assembly correctly.

Assuming the MapLink libraries don't exist in a directory in the system PATH variable, further issues may be encountered when a call to `TSLNDDrawingSurface::loadStandardConfig` is made and whenever the raster handling library is required, such as when loading a raster map.

An example workaround for this issue would be to temporarily change the working folder of the process, call `loadStandardConfig`, force load the raster handling library and then restore the working folder. The new working folder should contain the MapLink runtime DLLs and its value could be read from the application configuration or a registry key for example:

```
String existingWorkingDirectory = Directory.GetCurrentDirectory();
Directory.SetCurrentDirectory("C:\\SomeOtherfolder");
TSLNDrawingSurface.loadStandardConfig(configDirPath);
TSLNRasterUtilityFunctions.loadRasterLibrary("ttltrashlt");
Directory.SetCurrentDirectory(existingWorkingDirectory);
```

Fortunately none of the other MapLink .NET wrappers have any run-time dependencies that aren't loaded when the assembly is loaded.

6.1.2 Locating MapLink assemblies from the Global Assembly Cache

The MapLink .NET assemblies may be added to the Global Assembly Cache (GAC) of the deployment machine, but when they are loaded they may fail to locate their run-time dependencies. When this occurs Visual Studio doesn't provide much feedback as to the module that was not found so it can be difficult to diagnose the source of the issue.

If these dependencies can be found in the directory containing the application or in a directory appearing in the system `PATH` variable, then the application should have no problem locating them. If however they cannot be located the application may crash.

One way around this issue is to again change the working folder of the process and then force load the assemblies. This is quite tricky as it must occur before any MapLink classes are referenced and any procedure called which also includes the procedure that does this action. The best way to achieve this is to add this assembly load to the Main procedure of the application, e.g.

```
static void Main()
{
    Assembly thisAssembly = Assembly.GetExecutingAssembly();
    String existingWorkingDirectory = Directory.GetCurrentDirectory();
    Directory.SetCurrentDirectory("C:\\Somewhere");
    AssemblyName[] refrncdAssmbs =
        thisAssembly.GetReferencedAssemblies();
    foreach ( AssemblyName ass in refrncdAssmbs )
    {
        Assembly.Load(ass);
    }
    Directory.SetCurrentDirectory(existingWorkingDirectory);
    //...
```

It would also be preferable to filter the assemblies loaded via this method to limit them to the MapLink specific ones. This could be achieved by examining the assembly name and checking for the 'Envitia' prefix.

A user should then, later in the loading of the program, utilise the workaround suggested in the previous section for the loading of the raster handling library and for when calling `loadStandardConfig`.

6.1.3 Using the PATH environment variable

The MapLink Pro installation will add an entry in the `PATH` environment variable to the 64-bit bin directory, depending upon the MapLink installation type. This allows both the managed and native dependencies to be located at runtime.

This approach is rarely preferred however as it can cause library load time of the entire system to slow down if more and more applications add their installation location to the PATH variable. Additionally it can cause issues with name clashes, especially with third party libraries.

7 MAPLINK STUDIO AUTOMATION

MapLink Studio Automation is an optional component that provides access at runtime to most of the map processing capabilities of MapLink Studio.

Note: The paths in this section should be updated to reflect the version of MapLink you are using.

7.1 Deploying

Note: The Assumption is you do not have MapLink installed as part of your deployment. If you have MapLink installed then you will be overriding the standard MapLink Studio Automation, which is not the OEM version.

The OEM version of MapLink Studio does not include any licence keying. You need to request the OEM version as it is not shipped by default.

These are the files in the 64-bit MapLink Studio Automation:

```
MapLinkStudioOEM64.exe
MapLinkStudioOEM64.reg
MapLinkStudioOEM64.tlb
```

Copy the set of MapLink Studio Automation files relevant to your application to the applications `bin64` directory.

You will need to edit the `.reg` files, see below.

You will need the contents of the `config` and `projects` directory and the filter DLLs and dependencies. The `redist` directory is unlikely to have all the necessary DLLs as deployment of the Studio Automation is normally limited to OEM distribution.

7.1.1 Registry Updates

Reference: <http://support.microsoft.com/kb/310516>

Edit the registry file `MapLinkStudioOEM64.reg`.

The path denoted by `<INSTALLPATH>` needs to be updated to point to the top level of the installation, not the `bin` directory.

Points to remember:

- Use a double backslash as the path separator.
- On a 64bit machine you need to populate the 32bit part of the registry for 32bit MapLink Studio installation (Change the version number as necessary).
 - `HKEY_LOCAL_MACHINE\SOFTWARE\Envitia\MapLink Pro\X.Y`
 - `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Envitia\MapLink Pro\X.Y`

7.1.1.1 64bit Studio Automation

The 64bit `regedit` version can be found here:

- `C:\Windows\regedit.exe`

You can also override the Studio Automation used on the command line using the following:

Start a `cmd` window 'As Administrator'. In this window run the following:

```
C:\Windows\system32>cd C:\MapLink\version\bin64
C:\MapLink\version\bin64>MapLinkStudioOEM64 -register
```

This will not setup everything that is required so it is preferable to use the supplied registry files (see the registry files).

7.2 Uninstalling Studio Automation

Double click the `RemoveMapLinkStudioOEM.reg` file to remove the MapLink Studio Automation registry settings. This will also remove all the MapLink Studio Automation registry settings so you will need to rerun the registry files or MapLink Studio `-register` command to re-register MapLink Studio Automation.

The deployment registry files also set up the path to the configuration files required by MapLink Studio. The registry commands to remove these entries are commented out in the `RemoveMapLinkStudioOEM.reg` file.

You can also unregister the Studio Automation interface on the command line using the following:

Start a `cmd` window 'As Administrator'. In this window run the following:

```
C:\Windows\system32>cd C:\MapLink\version\bin64
C:\MapLink\version\bin64>MapLinkStudioOEM64 -unregister
```

7.3 Checking Studio Automation

If you have MapLink Pro installed you can check that Studio Automation is working by using the MapLink Studio Automation sample.

- Run the MapLink Studio Automation Sample.
- Loaded a map.
- Click the rightmost button on the toolbar (looks like a whiteboard on a stand).
- Selected the Shapefile icon from the dialog shown and clicked Ok.

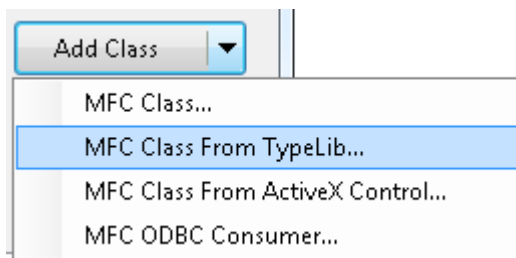
The next dialog (titled "Map Processing Control") should show without error – if it does not then the COM registration was unsuccessful.

You can also see which MapLink Studio is being used by using the Task Manager or a Process Explorer tool.

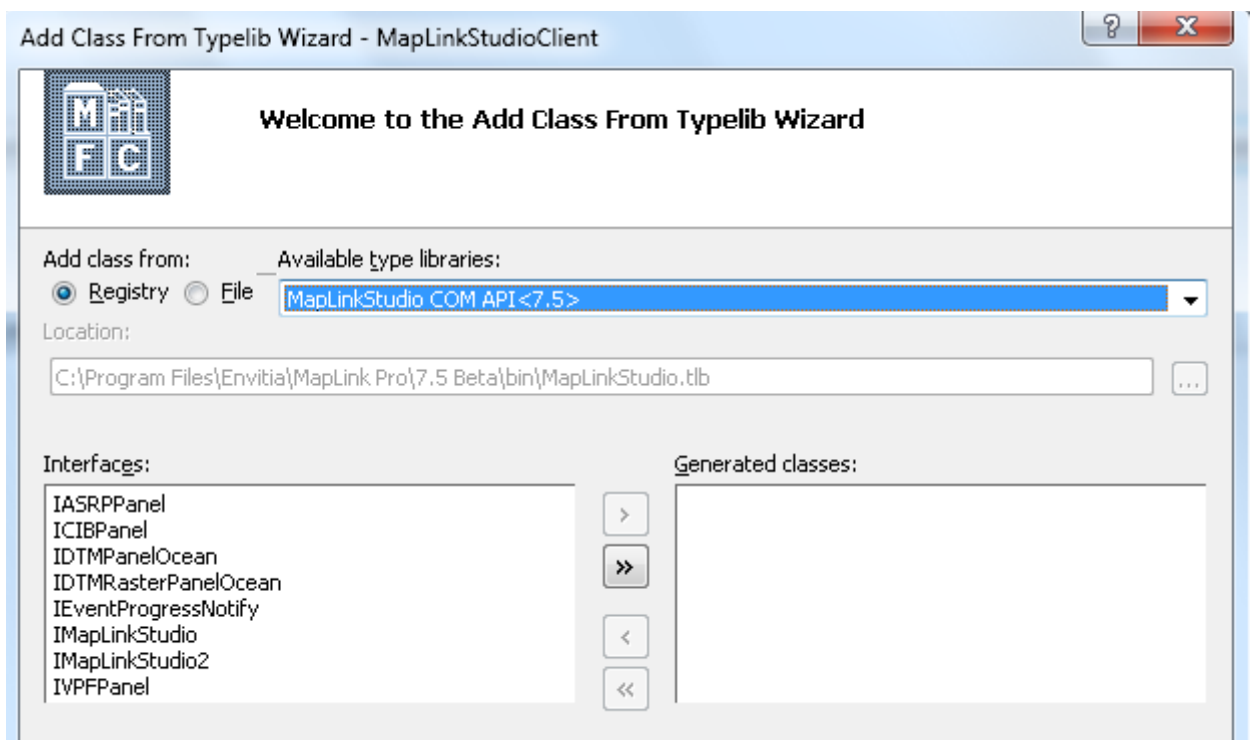
7.3.1 Regenerating the Automation Classes

The sample is shipped pre-built so should not need to be modified.

To regenerate the classes that wrap the MapLink Studio Automation delete the `MapLinkStudio.h/.cpp` (you may need to delete additional files as well if you have regenerated the sample). From the project start the 'MFC Class Wizard' and select 'Add Class'



Select the 'MapLinkStudio COM API' from the drop down menu:



Select the classes you wish to generate/re-generate (you may need to update the code as the Wizard for Microsoft Visual Studio may not generate the same class and filenames as used by the sample as it depends on which version you are using).

7.4 Studio Automation Deployment Problems

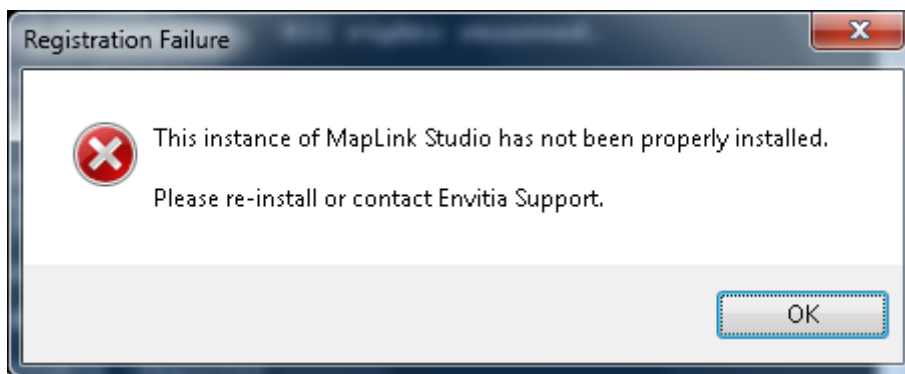
A number of problems can occur from not setting up the registry correctly to running the wrong MapLink Studio for registering the Automation interfaces.

You can use `regedit` to check the registry keys are correctly setup (see the reg files mentioned above for the keys used).

If you have Visual Studio installed you can also use 'OLE COM Object viewer'. This can be found under the Visual Studio start menu in 'SDK Tools'.

7.4.1 Licence Dialog is shown

If you see a dialog like the following:

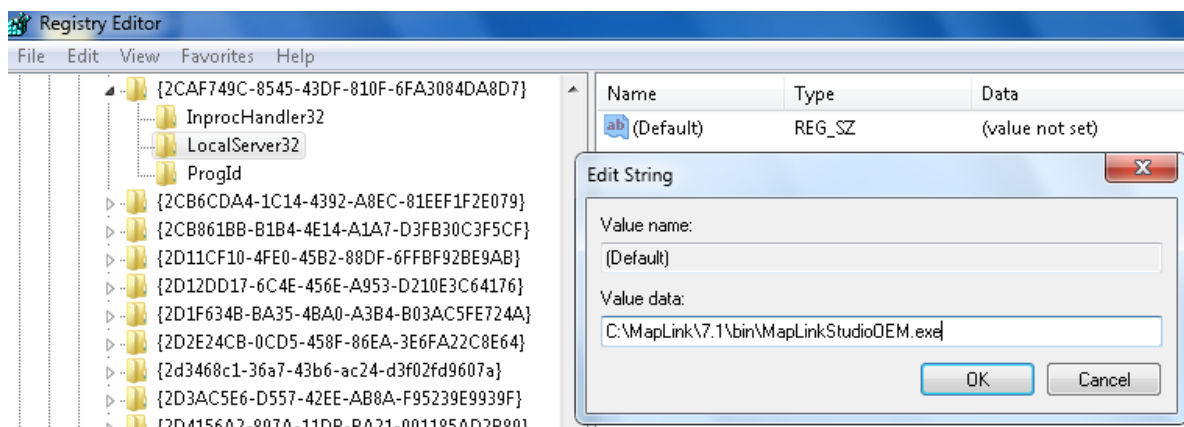


You have just registered the wrong MapLink Studio executable. You can check which one is running by starting the Windows Task Manager.

To fix this run the following:

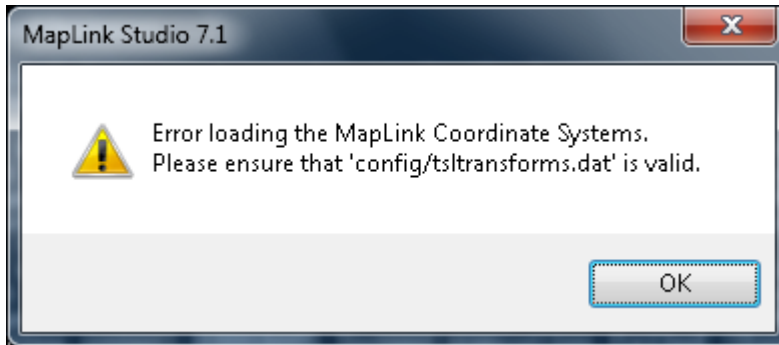
```
C:\Windows\system32>cd C:\MapLink\version\bin64
C:\MapLink\version\bin64>maplinkstudio64 -unregister
C:\MapLink\version\bin64>maplinkstudioOEM64 -register
```

You may need to edit the registry to fix the path to the MapLinkStudioOEM64.exe as shown below:



7.4.2 Config Directory cannot be found

In this case the following dialog box is displayed:



Check the registry keys :

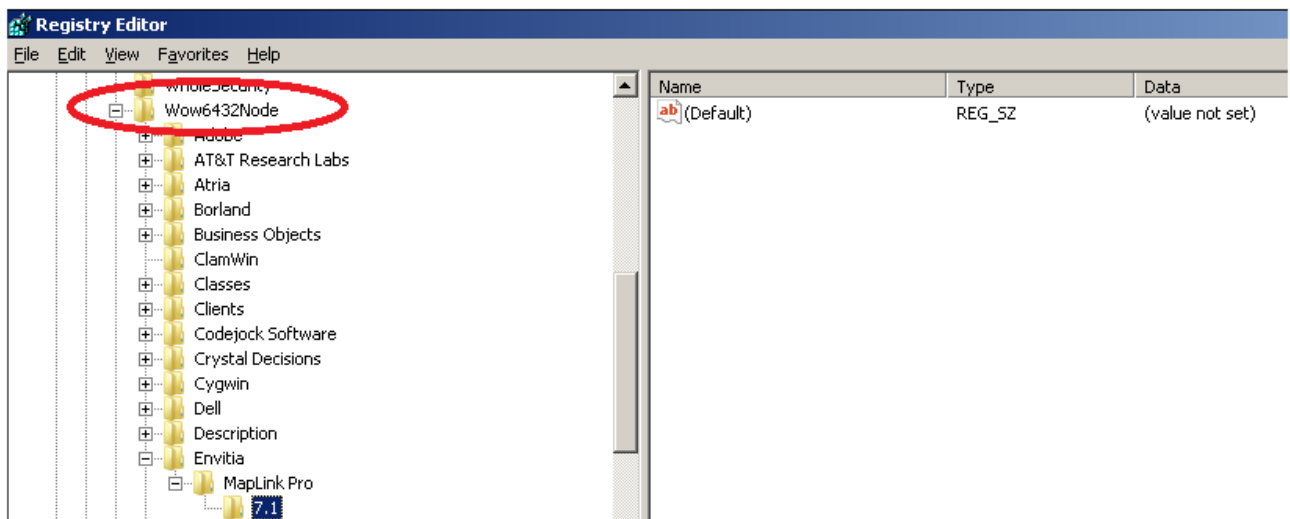
- HKEY_LOCAL_MACHINE\SOFTWARE\Envitia\MapLink Pro\X.Y
- HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Envitia\MapLink Pro\X.Y

There should be a key similar to the one show below (this should point to the top level installation directory of MapLink Pro or your Application Deployment):

```
"MAPL_HOME"="C:\MapLink\X.Y\bin64"
```

If this is missing edit your reg file and update the <INSTALLPATH> to have double backslashes \ as a path separator and rerun the reg file.

If this does not work you are probably on a 64bit machine and therefore need to also edit the 32bit part of the registry or run the appropriate version of regedit:



The above assumes that you are installing MapLink Studio in a default installation directory. This will preclude anyone else installing MapLink Studio or the MapLink tools.

The MapLink tools can be passed the location of the MapLink installation directory on the command line and thus can be run without MapLink having been installed.

Additional information about passing parameters to an automation server can be found here:

[http://msdn.microsoft.com/en-gb/library/windows/desktop/ms683844\(v=vs.85\).aspx](http://msdn.microsoft.com/en-gb/library/windows/desktop/ms683844(v=vs.85).aspx)

7.4.3 Shipping a Licenced version of MapLink Studio

Normally an application will use the OEM version of MapLink Studio and only ship the required filters.

If a licenced version of MapLink Studio is to be shipped then the MapLink Studio installers should be used.

If the MapLink Studio installers cannot be used then a discussion with support@envitia.com needs to be undertaken.

8 LINUX/OPENGL/X11 REQUIRED LIBRARIES

8.1 MapLink Libraries

NOTE: Linux Platform: glibc 2.5 or newer and libstdc++ from GCC 4.7 or newer are required when using any MapLink library.

8.1.1 The MapLink Pro Toolkit

Library	Description:	Notes
Envitia Core Mapping DLLs (Core SDK)		
libMapLink.so	Envitia Core Mapping libraries. Requires the Intel Thread Building Blocks libraries from section 8.3.	
libttltgm.so	Requires the Cryptopp library from section 8.3.	
libttltms.so	Required by every application dependent on MapLink Pro.	
libttlch.so		
libttltxf.so		
libttlmt.so		
libttlexpat		
libttlalpha.so		
libttlFeatureCatalogHelper.so		MapLink Feature Catalogue Helper – required by core
ttlclsstrk.so	Custom Line Style libraries.	
ttlclswf.so	Required if using the standard configuration files.	
libMapLinkMetadata.so	Envitia Metadata SDK DLL. Required if using the Metadata SDK.	
OpenGL Drawing Surface (not all platforms)		
libMapLinkOpenGLSurface.so	Envitia MapLink OpenGL Drawing Surface.	
libtess.so	Required if using the OpenGL Drawing Surface.	
Additional Core Support DLLs (Core SDK)		

Library	Description:	Notes
libMapLinkIModes.so	Envitia Interaction Mode library. Required if using the MapLink Interaction Modes SDK.	
libMapLinkThreadedMapCache.so	Envitia Threaded Map Cache SDK library. Required if using the Threaded Map Cache SDK.	
libMapLinkCADRGDL.so	Envitia CADRG Data Layer. Required if using the TSLCADRGDataLayer.	
libMapLinkRemoteLoader.so	Envitia Remote Loader. Required if using the TSLRemoteLoader or TSLWMSDataLayer.	
libMapLinkWMSDL.so	Envitia WMS Data Layer. Required if using the TSLWMSDataLayer.	
libMapLinkWMTSDL.so	Envitia WMTS Data Layer. Required if using the TSLWMTSDataLayer.	
libttlwmmts.so libMapLinkOWS.so		
libMapLinkS52.so	Envitia S52 Data Layer	
libMapLinkS63.so	Envitia S63 Data Layer (Only S-57 layer support)	Separate Licence required

Library	Description:	Notes
libMapLink2DKML.so	Envitia KML Data Layer	
libkmlbase.so		
libkmlbase.so.1		
libkmlbase.so.1.3.0		
libkmlconvenience.so		
libkmlconvenience.so.1		
libkmlconvenience.so.1.3.0		
libkmlDOM.so		
libkmlDOM.so.1		
libkmlDOM.so.1.3.0		
libKMLDrawingLibrary.so		
libkmlengine.so		
libkmlengine.so.1		
libkmlengine.so.1.3.0		
libkmlregionator.so		
libkmlregionator.so.1		
libkmlregionator.so.1.3.0		
libkmlxsd.so		
libkmlxsd.so.1		
libkmlxsd.so.1.3.0		
libttlcompression.so	MapLink Compression Support	
Filters		
libGenericFilter.so	MapLink Base Filter Support	
libttlrpj.so	MapLink Raster Reprojector – Required by Filters and CoreSDK	
libMapLinkS57Filter.so	MapLink S57 Filter	
libttlFeatureSymbologyHelper.so		
libGDALFilter.so	MapLink GDAL Filter	

Library	Description:	Notes
libASRPFilter.so	MapLink ASRP Filter	
libttlISO8211File.so	MapLink ISO8211 Reader – Required by ASRP filter	
libDafifFilter.so	MapLink DAFIF Filter	
libJeppesenFilter.so	MapLink Jeppesen Filter	
libRasterFilter.so	MapLink Raster Filter	
libShapeFilter.so	MapLink Shapefile Filter	
libNITFFilter.so	MapLink NITF Filter	
libOGRFilter.so	MapLink OGR Filter	
libVPFPlusFilter.so	MapLink VPF Filter	
Dynamic Data Object SDK		
libMapLinkDDO.so	Envitia Dynamic Data Object SDK library. Required if using the Dynamic Data Objects SDK.	
Database Interfaces SDK (DBIF)		
libMapLinkDBIF.so	Envitia Database Interfaces (DBIF) SDK libraries. Requires TPLXMLXERCES64.DLL from section 8.35.5.	
ttldbifsora.so	Envitia Database Interfaces Oracle Database Plug-in. Requires the Oracle 10g Database Client libraries to be installed. Required if using the DBIF SDK to connect to Oracle Databases.	
libMapLinkEntityStore.so	Envitia Entity Store SDK library. Requires libmaplinkdbif.so. Required if using the Entity Store SDK.	
OGC Filter SDK		
libMapLinkOGCFilter.so	Envitia OGC Filter SDK	
OpenGL Map Optimiser SDK		
libMapLinkOpenGLDataOptimiser.so	Envitia OpenGL Data Optimiser SDK	

Library	Description:	Notes
Direct Import SDK		
libMapLinkDirectImport.so	Envitia Direct Import SDK	
libMapLinkDirectImport_filterNITF.so	NITF Vector Direct Import – Requires the MapLink NITF Filter	
libMapLinkDirectImport_gdal.so	MapLink GDAL/OGR Direct Import Plugin – Requires GDAL/OGR	
Track Manager SDK		
libMapLinkTrackManager.so	Envitia Track Manager SDK	
libTTLTrackDrawingUtility_X11.so	Track drawing utility required if using X11 drawing surfaces – Requires Envitia Track Manager SDK	
libTTLTrackDrawingUtility_GL.so	Track drawing utility required if using OpenGL drawing surfaces – Requires Envitia Track Manager SDK	
GML SDK		
libMapLinkGML.so	Envitia GML SDK	
libOWCGMLOffering.so	MapLink OWS GML Context Offering	
libOWCWMSOffering.so	MapLink OWS WMS Offering	
libOWCWMTSOffering.so	MapLink OWS WMTS Offering	
OpenGL Track Helper SDK		
libMapLinkOpenGLTrackHelper.so	Envitia OpenGL Track Helper SDK	
OWS SDK		
libMapLinkOWSContext.so	Envitia OWS SDK	
Third Party Interop SDK (OGR)		
libMapLinkThirdPartyInterop.so	Envitia Third Party Interop with OGR	
GDAL Plugins		
gdal_ECW_JP2ECW.so	GDAL Plugin for JP2 & ECW – Requires Third Party Libraries	

Library	Description:	Notes
gdal_MrSID.so	GDAL Plugin for MrSID Format – Requires Third Party Libraries	

8.1.2 MapLink GeoPackage

Library	Description:	Notes
MapLink Pro Toolkit	The SDK requires the MapLink Pro Toolkit.	
libMapLinkGeoPackage.so	Envitia MapLink GeoPackage SDK DLL.	
libsqlite3.so	SQLite 3.18.0	
libgdal.so	GDAL/OGR 2.1.2	

8.1.3 MapLink Pro Accelerator SDK

This is a separately licensed component.

Library	Description:	Notes
MapLink Pro Toolkit	The SDK requires the MapLink Pro Toolkit.	
libMapLinkAccelerator.so	Envitia OpenGL Accelerator SDK DLL. Requires libGLEW from section 8.3. Required if using the OpenGL Accelerator SDK.	

8.1.4 MapLink Pro Terrain SDK

This is a separately licensed component.

Library	Description:	Notes
MapLink Pro Toolkit	The SDK requires the MapLink Pro Toolkit.	
libMapLinkTerrain.so	Envitia Terrain SDK library. Requires Envitia Core MapLink libraries.	
	Direct Import DMED	Additional Licence. Runtime locked.

8.1.5 MapLink Pro 3D SDK

This is a separately licensed component.

Library	Description:	Notes
MapLink Pro Toolkit	The SDK requires the MapLink Pro Toolkit.	
MapLink Pro Terrain SDK	The SDK requires the MapLink Pro Terrain SDK.	
libMapLink3D.so	Envitia 3D Mapping library. Requires libGLEW from section 8.3. Required if using the MapLink 3D SDK.	
libMapLink3DIModes.so	Envitia 3D Interaction Mode library. Required if using the MapLink 3D Interaction Modes SDK.	
libtess.so	Envitia Tesselator library. Required if using the 3D SDK.	
ttl3ds.so	Sample 3D Model library that may be redistributed. The source code for this library is provided as a sample. Requires lib3ds-1_3.so from section 8.3. Required if using 3D Models.	

8.1.6 MapLink OGC Services

Please refer to the 'MapLink OGC Services Deployment Users Guide' (UUM1106-XX) as additional configuration is required over and above deploying the required shared libraries.

The MapLink OGC Services require a server license.

Library	Description:	Notes
libMapLinkWMS.so	MapLink Web Map Service (WMS) SDK (See the Developers Guide) Depends on MapLink Core, libMapLinkOGCServices.so	
CADRGMapWMS_plugin.so	MapLink CADRG WMS Plugin See the 'MapLink CADRG WMS Plug-in User Documentation' (UUM1155-xx) Depends on libMapLinkWMS.so	Additional licensed Component

Library	Description:	Notes
BasicMapWMS_plugin.so	MapLink Basic WMS plugin. Depends on libMapLinkWMS.so	Use the SuperMap Plugin.
SuperMapWMS_plugin.so	MapLink WMS Super Map plugin Depends on MapLink Core, libMapLinkWMS.so, libMapLinkThreadedMapCache.so, libttlXalanhelpers.so, Intel TBB, libxerces-c, libxalan-c.so	
HistoricalMapWMS_plugin.so	MapLink Historical Map plugin Depends on MapLink Core, libMapLinkWMS.so.	Use the SuperMap Plugin.
libMapLinkOGCServices.so	MapLink OGC Services	
libttlXalanhelpers.so	MapLink helper methods for Xalan-C	
libxalan-c.so.111	Apaxe Xalan-C	

8.2 Raster Handling Libraries

This section lists the libraries required for loading, manipulating and saving raster images using MapLink Pro. It should be noted that using MapLink maps that were created from raster source data will require these libraries.

8.2.1 Required Libraries

The following table lists the libraries that are required for raster handling.

Note: JBIG support is disabled.

Library	Description:
libttrashx.so	Envitia Raster-handling DLL.
libjpeg.so libjpeg.so.9 libjpeg.so.9.2.0	JPEG support (internal).
libopenjp2.so libopenjp2.so.7 libopenjp2.so.2.1.2	OpenJpeg 2.1.2
libpng16.so libpng16.so.16 libpng16.so.16.25.0	PNG support (internal).
libtiff.so libtiff.so.5 libtiff.so.5.2.4	TIFF support (internal).
libttdgdk.so	Modified GDK library (internal).
libGraphicsMagick.so libGraphicsMagick.so.3 libGraphicsMagick.so.3.14.1 libGraphicsMagick++.so libGraphicsMagick++.so.12 libGraphicsMagick++.so.12.0.1 libGraphicsMagickWand.so libGraphicsMagickWand.so.2 libGraphicsMagickWand.so.2.7.3	Graphics Magick 1.3.25
libfreeimage.so	FreeImage 3.17.0

8.3 3rd Party Libraries

The following table lists a number of 3rd Party Libraries on which certain MapLink libraries depend. Usually these should only be deployed if an application uses a library listed in a previous section notes that it requires it.

The relevant symbolic links should also be preserved and deployed.

Library	Description:
libtbb.so.2 libtbbmalloc.so.2	Intel Intel® Thread Building Blocks DLLs. Version: 2017 Update 3 Required by every application dependent on MapLink Pro on Linux x86/x64 only.
libGLEW.so libGLEW.so.1.13 libGLEW.so.1.13.0	The OpenGL Extension Wrangler Library version 1.13.0. Required by 3D SDK, Accelerator SDK and OpenGL Drawing Surface.
lib3ds-1.so.3	LIB3DS-1_3.DLL is the Lib3ds free toolkit Required by 3D SDK.
libtess.so	OpenGL Tesselator DLL (1.2.1.0) Required by 3D SDK and OpenGL Drawing Surface.
libz.so libz.so.1 libz.so.1.2.8	Zlib 1.2.8
libxerces-c-3.1.so	Apache Xerces 3.1.1
libxalan-c.so.111 libxalan-c.so.111.0	Apache Xalan-C 1.11
libexpat.so libexpat.so.1 libexpat.so.1.6.0	Expat 2.1.0
libttlexpat.so	ExpatPP
libfreetype.so libfreetype.so.6 libfreetype.so.6.11.0	Freetype 2.5.1
libtiff.so libtiff.so.5 libtiff.so.5.2.4	LibTIFF 4.0.6
libfribidi.so libfribidi.so.0 libfribidi.so.0.3.4	GNU Fribidi 0.19.4
libGeographic.so libGeographic.so.10 libGeographic.so.10.1.1	GeographicLib 1.34
libglib-2.0.so libglib-2.0.so.0 libglib-2.0.so.0.3800.2 libgio-2.0.so libgio-2.0.so.0 libgio-2.0.so.0.3800.2 libgmodule-2.0.so libgmodule-2.0.so.0 libgmodule-2.0.so.0.3800.2	Glib 2.38.2

libgobject-2.0.so libgobject-2.0.so.0 libgobject-2.0.so.0.3800.2 libgthread-2.0.so libgthread-2.0.so.0 libgthread-2.0.so.0.3800.2	
libgts-0.7.so.5	GNU Triangulated Surface library 0.7.6
libharfbuzz.so libharfbuzz.so.0 libharfbuzz.so.0.925.0	Harfbuzz-ng 0.9.25
libboost_atomic.so libboost_atomic.so.1.62.0 libboost_chrono.so libboost_chrono.so.1.62.0 libboost_date_time.so libboost_date_time.so.1.62.0 libboost_filesystem.so libboost_filesystem.so.1.62.0 libboost_regex.so libboost_regex.so.1.62.0 libboost_thread.so libboost_thread.so.1.62.0	Boost 1.62.0
libcurl.so libcurl.so.4 libcurl.so.4.4.0	cURL 7.50.3
libcryptopp.so	Crypto++ 5.6.5
libpng16.so.16	Libpng 1.6.25
libfontconfig.so libfontconfig.so.1 libfontconfig.so.1.8.0	FontConfig 2.11
libgmp.so libgmp.so.10 libgmp.so.10.2.0	GNU MP 6.0.0a
libgnutls.so libgnutls.so.28 libgnutls.so.28.41.8	GnuTLS 3.3.16 (3.1.9 on Solaris)
libnettle.so libnettle.so.6 libnettle.so.6.1 libhogweed.so libhogweed.so.54 libhogweed.so.54.1	Nettle 3.1.1 (2.5 on Solaris)
libpango-1.0.so libpango-1.0.so.0 libpango-1.0.so.0.3600.1	Pango 1.36.1 (1.20.5 on Solaris)

libpangoft2-1.0.so libpangoft2-1.0.so.0 libpangoft2-1.0.so.0.3600.1 libpangoxft-1.0.so libpangoxft-1.0.so.0 libpangoxft-1.0.so.0.3600.1	
libxft.so.2.1.13	Xft 2.1.13 (Solaris x86 only)
libiconv.so.2.5.1 libcharset.so.1.0.0	Iconv 1.14 (Solaris, Embedded Linux, Android only)
libsqlite3.so libsqlite3.so.0 libsqlite3.so.0.8.6	SQLite 3.18
libminizip.so libminizip.so.1 libminizip.so.1.0.0	Minizip 1.0.0
libpcrecpp.so libpcrecpp.so.0 libpcrecpp.so.0.0.1 libpcre.so libpcre.so.1 libpcre.so.1.2.7	PCRE 8.39
liburiparser.so liburiparser.so.1 liburiparser.so.1.0.20	uriparser 0.8.4
libgdal.so libgdal.so.2 libgdal.so.2.1.2	GDAL 2.1.2
libtidsdk.so libtidsdk.so.9	MrSID (See Third Party license)
alibexpat.so libexpat.so.1 libexpat.so.1.6.0	The Expat XML Parser
libffi.so libffi.so.6 libffi.so.0.1	Foreign Function Interface
libicudata.so libicudata.so.54 libicudata.so.54.1 libicui18n.so libicui18n.so.54 libicui18n.so.54.1 libicuio.so libicuio.so.54 libicuio.so.54.1	ICU 54 – International Components for Unicode

libicule.so libicule.so.54 libicule.so.54.1 libiculx.so libiculx.so.54 libiculx.so.54.1 libicutest.so libicutest.so.54 libicutest.so.54.1 libicutu.so libicutu.so.54 libicutu.so.54.1 libicuuc.so libicuuc.so.54 libicuuc.so.54.1	
libsquish.so libsquish.so.0.0	libSquish 0.1.11
libproj.so libproj.so.12 libproj.so.12.0.0	Proj4 4.9.3
libxml2.so libxml2.so.2 libxml2.so.2.9.1	libxml2 2.0.0

9 COMMON DEPLOYMENT PROBLEMS

9.1 DLL/Shared Library not found

On Windows use dependency walker to see what DLLs are loaded and if they can be found.

On Linux use `ldd` to see what libraries are being used by a shared library or executable.

.NET may require the use of a `ResolveEventHandler`.

Make sure that the MapLink DLLs are in the path environment variable or that you run your application in the directory containing the MapLink DLLs.

If you could end up deploying multiple versions of your application to a single machine you should avoid setting the path in the environment variable and run the application in the `bin64` directory. If you do this then be careful that the application or GUI toolkit does not change the working directory.

9.2 MapLink Studio

Only one version of the Automation interface can be registered at any-one time.

Please refer to section 7. If this section does not help please contact support.