AUTODESK **UNIVERSITY** 2014

CV6371 - Driving BIM Forward: Pioneering Object Data Linking on a U.K. Highway Project

Dave Bosworth MRICS, Principal Consultant, Excitech Ltd

CV6371

The U.K. government has mandated that all government-funded projects must be delivered to Level 2 Building Information Modeling (BIM) by 2016. In a breakout from U.K. convention, companies have used AutoCAD Civil 3D software as the highway design tool for the Building Information Modeling (BIM) model and Revit software and Structural Bridge Design software for the major bridge structures. While 3D BIM models have been around for a few years, there usually has not been any attendant data attached (beyond the inherent geometry) to models generated from AutoCAD Civil 3D software.

We will discuss various strategies and methods for creating additional asset data in AutoCAD Civil 3D, and propose and demonstrate a method of attaching additional data to the BIM model and linking this to both the AutoCAD Civil 3D software entities and the Navisworks objects for Field BIM software. As part of this process, we will also discuss how Civil 3D objects may be used to represent real-world assets in the BIM model. Finally we will explore the result of customisation using .NET plug-ins to Civil 3D to further enhance the ability to link data to the BIM model.

Learning Objectives

At the end of this class, you will be able to:

- Gain an understanding of how an infrastructure BIM design can be created by linking AutoCAD Civil 3D and Revit, and how Civil 3D objects can be used to represent different real-world assets.
- Understand the various methods of adding additional asset data to model objects, their pro's and con's, and the implications of each method.
- Understand the methods and limitations of data linking from native AutoCAD Civil 3D software entities to Navisworks software objects in a BIM model
- Discover how additional customisation of Civil 3D can further enhance the data linking process to allow the population of a database with information from Civil 3D.

About the Speaker

Dave Bosworth is a Chartered Surveyor and Principal Consultant for Excitech Ltd, a UK Autodesk Platinum Partner. Dave started his career in construction as an engineering surveyor, working mainly for consulting engineers on construction projects across the UK, before moving into a technical role using Autodesk products. His current role involves working closely with clients to enable them to work effectively with their design tools and guiding them on implementation of advanced workflows for 3D design and delivery of BIM.
dave.bosworth@excitech.co.uk

Introduction

Since the UK government mandated the use of BIM for all publicly funded projects from 2016, there has been a rush to implement BIM – however it is fair to say that the adoption of BIM for infrastructure projects has been somewhat slower than for buildings and architecture. One of the main reasons for this is the perception that the available design software does not provide the necessary functionality. This lecture will focus on how BIM was adopted on a major UK highways project, which was designated as a BIM pilot project by the Highways Agency, and was carried out using AutoCAD Civil 3D, Revit Structure and Navisworks Manage.

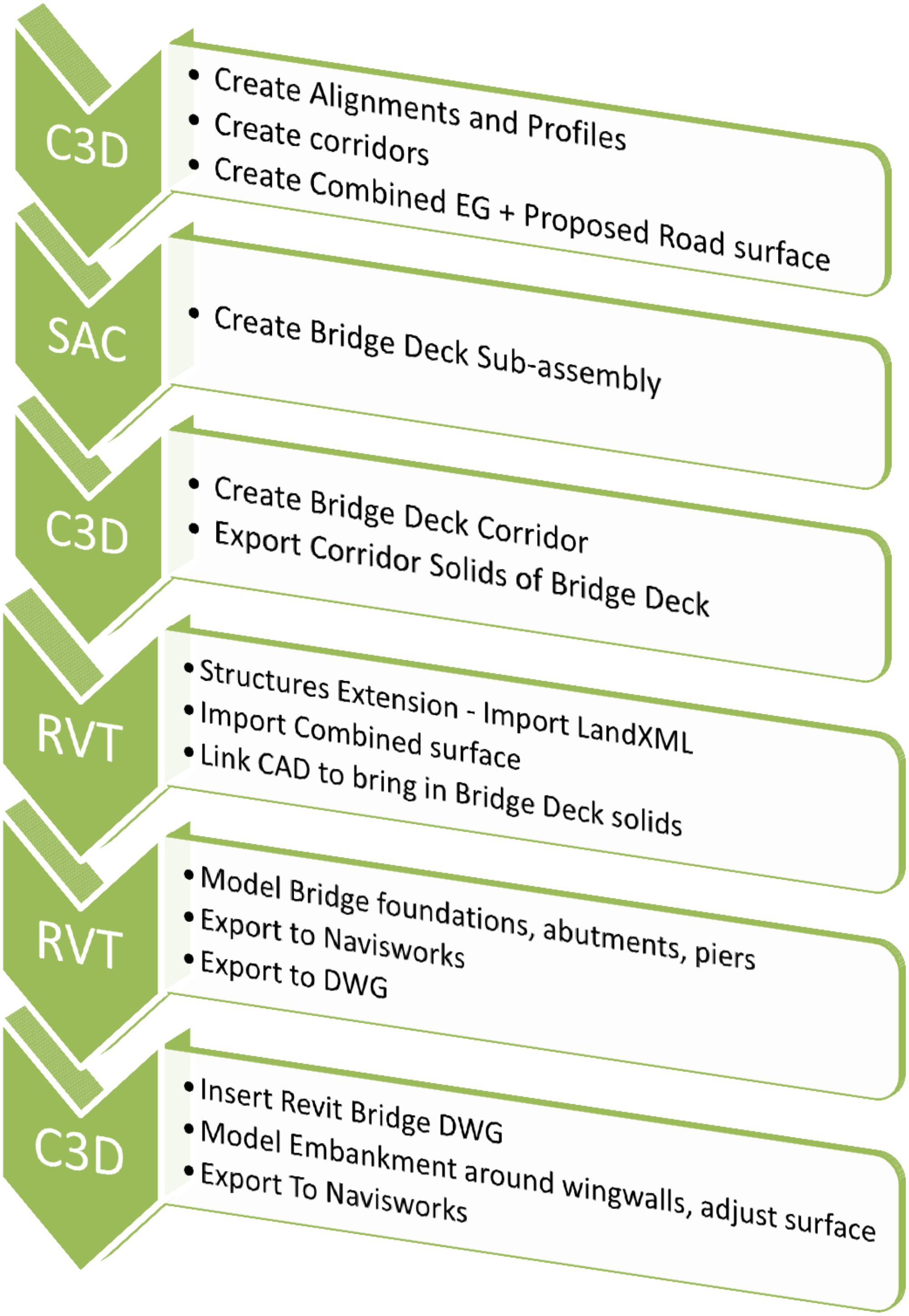
In the BIM pilot project, a number of issues were identified early in the project:-

- Creating realistic model objects of a range of assets for co-ordination in Navisworks using the tools available
- Co-ordinating design data between Civil 3D and Revit Structure for bridges and other structures
- Bringing design and asset data from the design model into Navisworks and beyond as asset data for asset management

Creating the Model

Civil 3D and Revit co-ordination

A Bridge Design toolkit for Civil 3D and Revit has been available for some time, however it is only really suitable for very straightforward bridge designs, and carrying across advanced concepts such as a bridge with a super-elevated deck and a crested profile has proved unsatisfactory. However, the Bridge Design toolkit can be used effectively to bring in a solid bridge deck created in Civil 3D together with design surfaces to enable the modelling of a bridge in Revit. The advantage of using this method is that although there is considerable manual modelling of piers and abutments, etc, the bridge deck will carry the exact geometric properties required from the road design, and the relevant surfaces can be used to determine the foundation design levels in Revit.



C3D
- Create Alignments and Profiles
- Create corridors
- Create Combined EG + Proposed Road surface

SAC
- Create Bridge Deck Sub-assembly

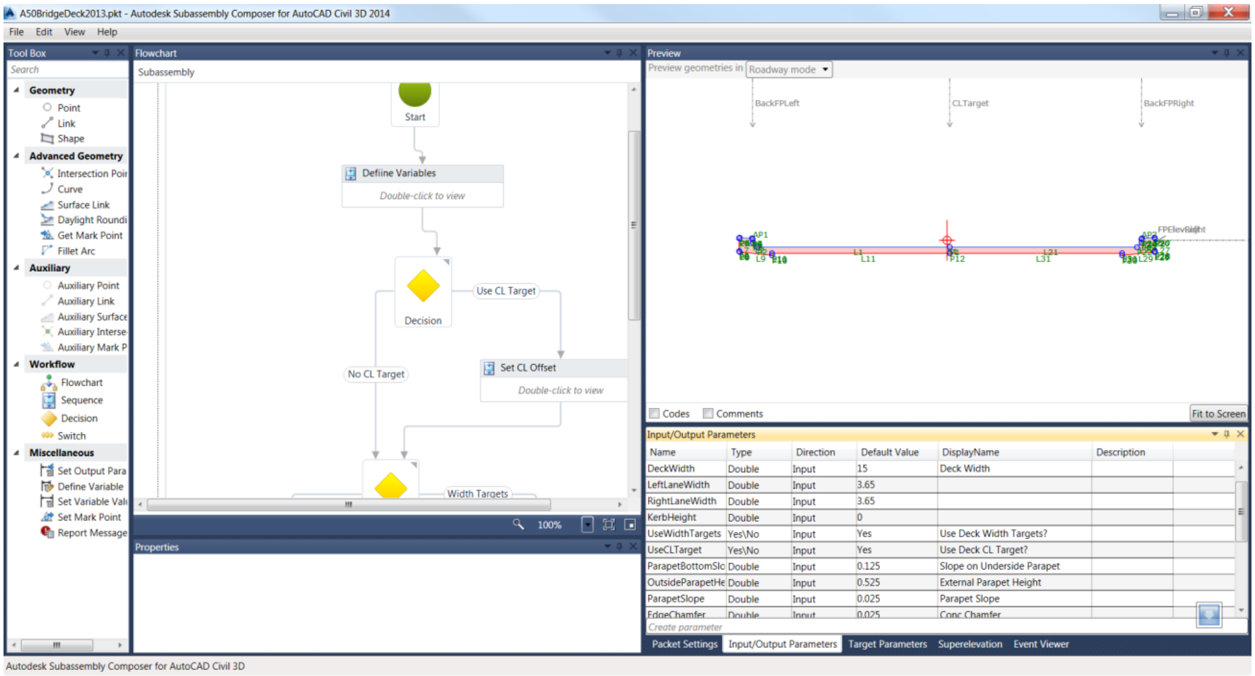
C3D
- Create Bridge Deck Corridor
- Export Corridor Solids of Bridge Deck

RVT
- Structures Extension - Import LandXML
- Import Combined surface
- Link CAD to bring in Bridge Deck solids

RVT
- Model Bridge foundations, abutments, piers
- Export to Navisworks
- Export to DWG

C3D
- Insert Revit Bridge DWG
- Model Embankment around wingwalls, adjust surface
- Export To Navisworks

Figure 1 - Civil 3D and Revit bridge workflow



Name	Type	Direction	Default Value	DisplayName	Description
DeckWidth	Double	Input	15	Deck Width	
LeftLaneWidth	Double	Input	3.65		
RightLaneWidth	Double	Input	3.65		
KerbHeight	Double	Input	0		
UseWidthTargets	Yes\No	Input	Yes	Use Deck Width Targets?	
UseCLTarget	Yes\No	Input	Yes	Use Deck CL Target?	
ParapetBottomSlo	Double	Input	0.125	Slope on Underside Parapet	
OutsideParapetHe	Double	Input	0.525	External Parapet Height	
ParapetSlope	Double	Input	0.025	Parapet Slope	
EdgeChamfer	Double	Input	0.025	Conc Chamfer	

Figure 2 - Bridge Deck sub-assembly

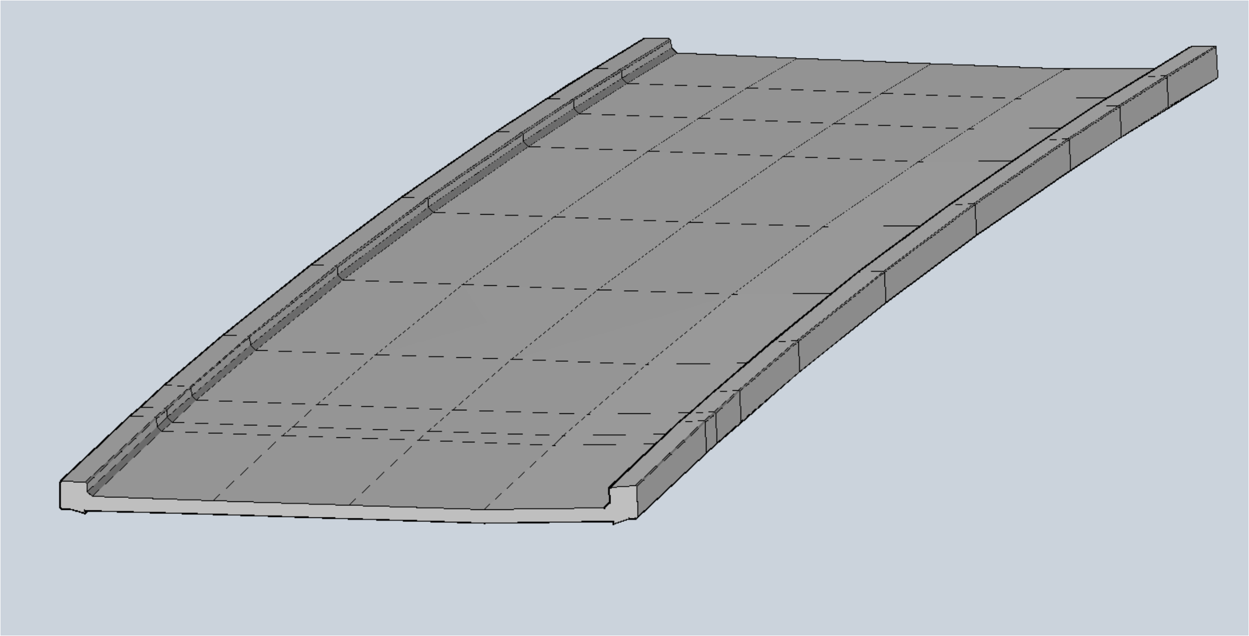


Figure 3 - Bridge Deck corridor solids

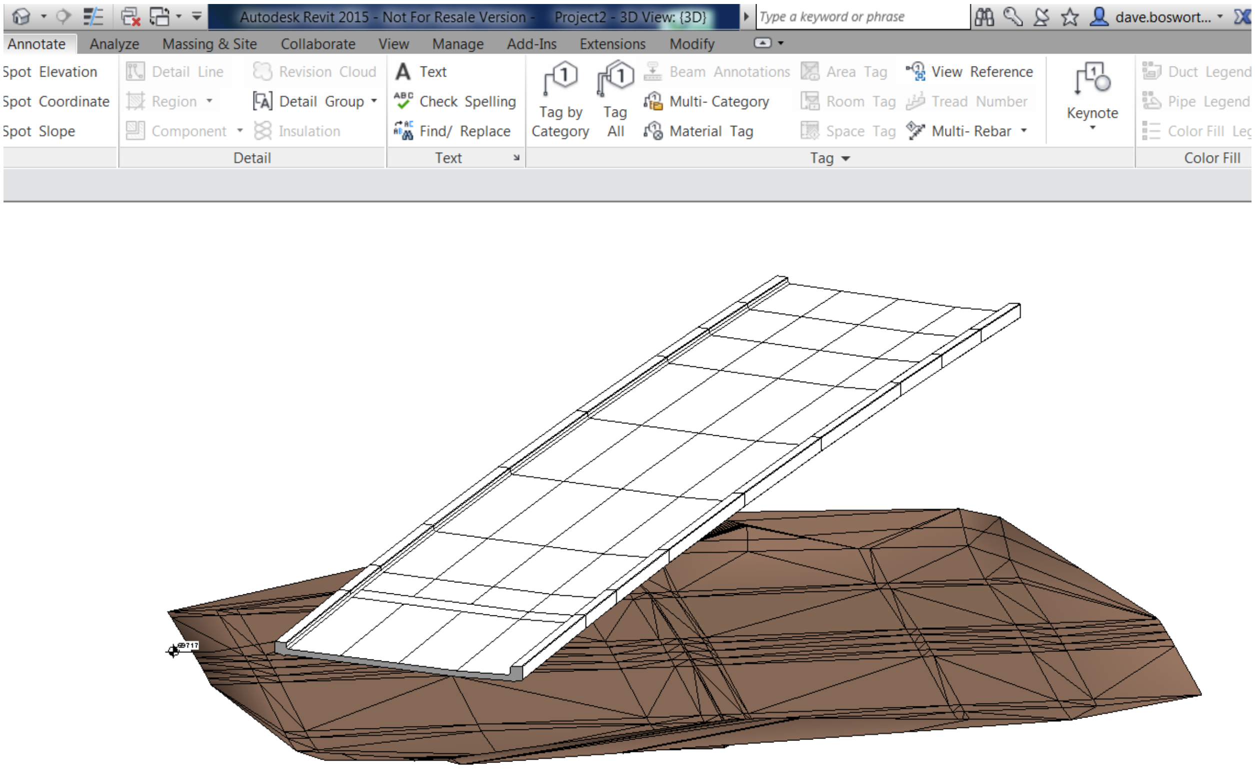


Figure 4 - Bridge Deck in Revit

Asset Modelling

Civil 3D can of course model highways using corridors, and drainage systems and underground utilities using pipe networks or pressure pipes. However, there are other highway assets that need to be modelled efficiently, and these can be broadly split into three types: Point, Linear, and Area.

Point assets might be items such as comms cabinets or light columns. These can be represented as blocks, and Civil 3D points can be used to place them in the model. Using multi-view blocks as the point marker works quite well, and will show as individual 3D objects in Navisworks. Furthermore, additional data can be added using User-defined data fields in Civil 3D, although there are problems in getting these additional fields into Navisworks – but of course it works great for scheduling, and they can be managed effectively using Point Groups. Points were also used to track CDM / HSE information in the project, with hazards identified as a 3D block with data attached which could then be displayed effectively in Navisworks during model reviews. One-off objects and some more complex objects – gantries for example – were either modelled in Revit or using AutoCAD solids and shown as a 3D block.

Linear assets abound on highways projects, and embody assets such as sound barriers, highway fences and safety barriers. These were modelled using corridors, with assemblies being created for different fence types. Sub-assembly Composer was used to create the sub-assemblies for different configurations. The posts were incorporated in the corridor model using point markers in the corridor at the required spacing, where the marker style is a multi-view block of the post. This method proved to be very flexible, as the levels can be controlled by the profile, and additional targets can be employed in the sub-assembly to change the height of the object.

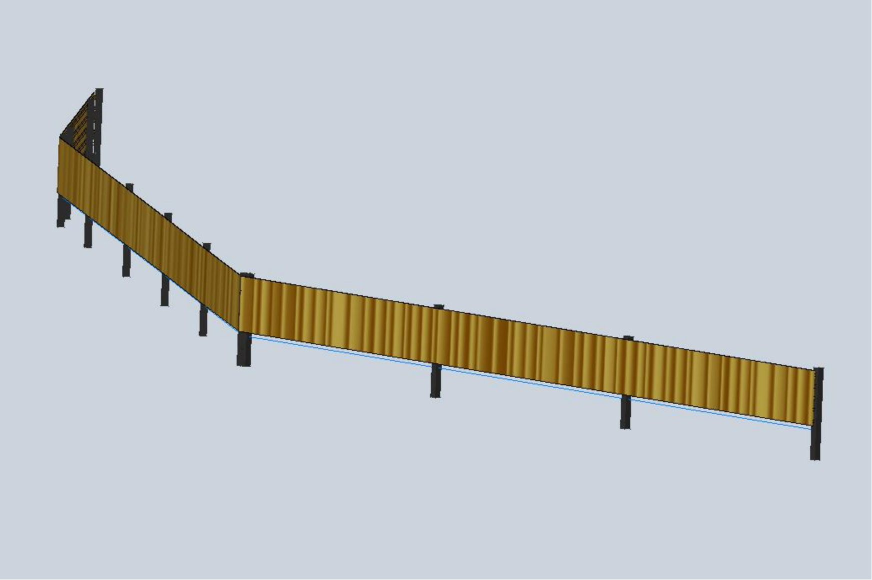


Figure 5 - Fence created using corridors

Sound barriers were created using a special sub-assembly that created what is termed as “false cut”. A target in the sub-assembly allowed the height of the resulting bund to be controlled using a polyline in the drawing, enabling a quick method of transitioning the height of the bund without the need for a 3D polyline or a feature line.

Area assets are of course best created using surfaces. These were used to represent areas of planting, for example. Where necessary the height of the planting could be shown by simply raising the surface and creating a “skirt” down to the proposed ground surface. Parcels were also considered for this purpose, but surfaces offered the possibility of deriving 3D areas automatically and also are much better to view in the context of a 3D model.

Getting the Data In – Developing a BIM Data Strategy

AutoCAD has many options for creating additional metadata that can be linked or attached to the graphical objects that are shown in the drawing. Not all of these methods are appropriate for all types of objects, and particularly when considering asset data for BIM projects, the objects we are creating are not always discrete (unlike, for example, a valve), particularly when looking at Infrastructure objects with items such as roads being represented. This topic describes some of the methods that are available for creating embedded data, that is, data that is stored in the drawing file itself.

Blocks With Attributes

Blocks are an ideal way of grouping elements that representing a real world object when the object has a single insertion co-ordinate. Attributes can be added to the block and the user can be prompted to fill out the required asset data at the time of insertion. These attributes can easily be extracted using existing tools in AutoCAD. However, blocks are best suited when an object is likely to be inserted many times, and only for discrete objects – linear objects such as railings and fences are not good choices for this as blocks would have to be created for each instance, and the insertion point method is not really applicable for this type of object.

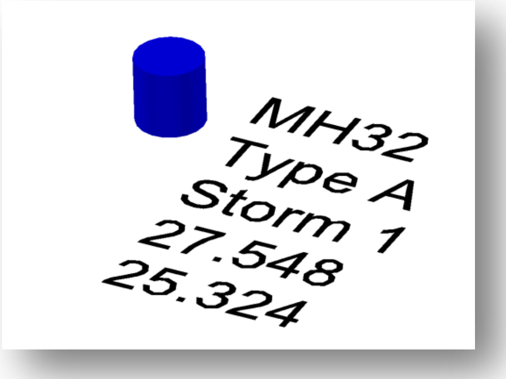
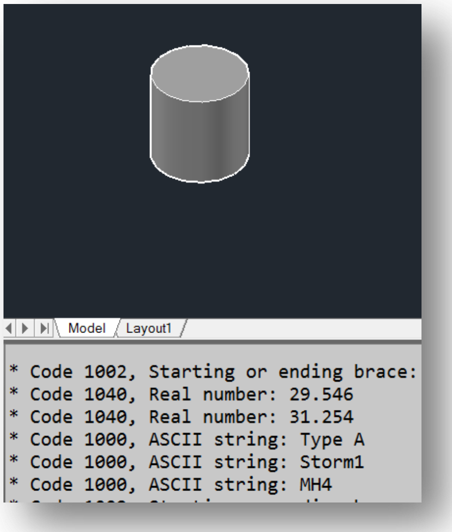


Figure 6 - Block with attributes

XData

AutoCAD includes a method of attaching metadata to objects which is not immediately visible in the drawing model space. This method is called Extended Entity Data or XData for short. The data structures are created in advance, and are stored in the drawing file in a “dictionary” area of the drawing. The data itself is not actually attached to the object, but each line of data residing in the dictionary is tagged with the object ID of the drawing entity it belongs to. As a result of the way this data is managed, it is difficult to get a drawing wide view of all the data that is added. There is no dedicated interface for creating and managing this data, with much of the work being entered at the command line, and it can be awkward to extract. However, Xdata can be used to attach information to any AutoCAD or Civil 3D object or entity.

Xdata is intended primarily for use by programmers who need to add linking information to graphical objects which can be read by their own code to manipulate and manage AutoCAD entities.



```
* Code 1002, Starting or ending brace:
* Code 1040, Real number: 29.546
* Code 1040, Real number: 31.254
* Code 1000, ASCII string: Type A
* Code 1000, ASCII string: Storm1
* Code 1000, ASCII string: MH4
```

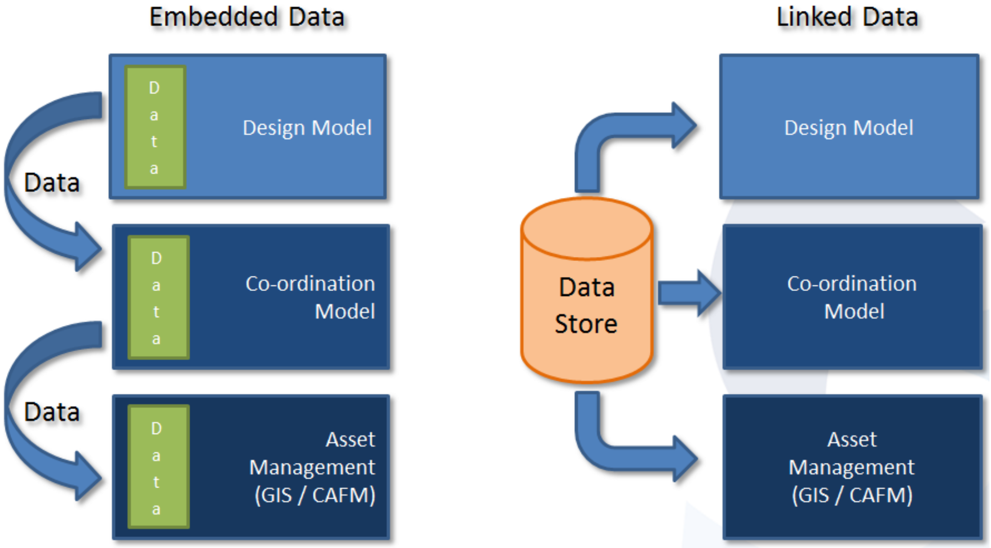
Figure 7 - Xdata example

Map 3D Object Data

Map 3D has additional functionality over AutoCAD, called Object Data. This is a customised form of Xdata and is intended to provide GIS type metadata structures for AutoCAD users. The data is still stored in the drawing in the same way that Xdata is, but an interface has been added in Map 3D to allow easy creation of the required data structures, with additional tools to attach and edit the data. It is still difficult to get a “drawing wide” view of all the object data that exists in a particular drawing, but it is much easier to manage than Xdata. In addition, tools are provided to export this type of data to other external formats such as GIS file formats or databases.

Since Civil 3D inherits most of the Map 3D functionality, Object Data is a viable method for adding additional data to Civil 3D objects.

Linked Data in Civil 3D / Map 3D



Linked data is where the metadata is stored in an external data source, such as a spreadsheet, a file based database, or an enterprise database such as SQL Server or Oracle. The advantage of using linked data is that data can be created and maintained outside of AutoCAD by non-AutoCAD users if necessary. The main disadvantage is that to pass the data on, the database or data store must be made available in addition to the drawing files.

Linked Data in AutoCAD

AutoCAD has the ability to create links between a database and the graphical drawing objects. This technology is called CAO (Connectivity Automation Objects) and can be invoked on the command line by typing DBCONNECT.

The principle is that connections are made between the objects in the drawing and a row in a database table. Some information is stored in the drawing to enable the linking to take place – a “Link Template” is created in the drawing file that records how the data in the datasource is linked to objects in the drawing. For example, a unique “key” column in the database table is mapped to an object ID in the drawing. The information required to re-establish these links is saved in a “dictionary” in the drawing file – essentially a piece of Xdata that knows the value of the key field in the database that it links to. This method works very well, with the ability to see all of the data in a single “table view” from inside AutoCAD. However, one of the main drawbacks with this method is that links have to be created manually, by selecting a row in a table and attaching this to an item in the drawing. Furthermore, if the “link template” stored in the drawing is deleted or corrupted, the links cannot automatically be re-established, and if a drawing object is deleted the link is deleted. This can be quite a major drawback as often we want to delete an object and re-create it, in which case we must know how to re-create the link to the database.

Linked Data with Map 3D / Civil 3D

Map 3D also uses CAO, together with link templates, and it works the same way as with AutoCAD. However, the process is taken a step further with additional interface dialogs that allow some manipulation of the link templates, and with tools that allow automatic linking of Object Data with a data source using CAO. The additional visibility of the process that Map 3D provides makes this a much better prospect for handling BIM data.

Proposed method for Data Linking

Combining Object Data and Linked Data

The proposal for creating linked BIM data using Map 3D and Civil 3D involves a process of creating Object Data which is attached to drawing objects, and then establishing links automatically between the Object Data and the linked datasource. The advantage of combining these two methods means that with some careful planning, links can be re-established at any time between the drawing objects and the data source. With some automation using Lisp or VB.Net, this process can be repeatable even if the Object Data is deleted, or the link template becomes corrupted, or if an object needs to be deleted and re-created.

This makes the process much more robust.

The same datasource can be re-used to link data back to objects in Navisworks or other model review tools.

Data Strategy for Creating the Object Data

The process relies on creating some unique identifier in the drawing (as Object Data) that can be automatically matched to the identical data that is stored in the data source (spreadsheet or database).

Entity Handles

For example, when dealing with AutoCAD entities, each item in the drawing database is given an Object ID and a “Handle”. The Object ID used in AutoCAD is not persistent, in other words it is recreated for each drawing session. For this reason it is not a good choice for creating database links. The object “Handle”, on the other hand, is assigned when an entity is created, and remains with that item until it is deleted, and it persists in the drawing file. This makes it a good choice for using as a unique ID. (Note that it is possible that running an audit or recover on a drawing, or “wblocking” a drawing might result in new handles being assigned). The only issue with using entity handles for this process is that they cannot easily be seen by the user, and there is no automatic way of pushing the object handles into Object Data. This is where customisation is useful. Deleting and recreating objects might still be an issue when using handles as unique ID’s, as they are destroyed when an object is deleted and a new one assigned when an object is created. It is not possible for the user to assign a handle manually.

Object Names

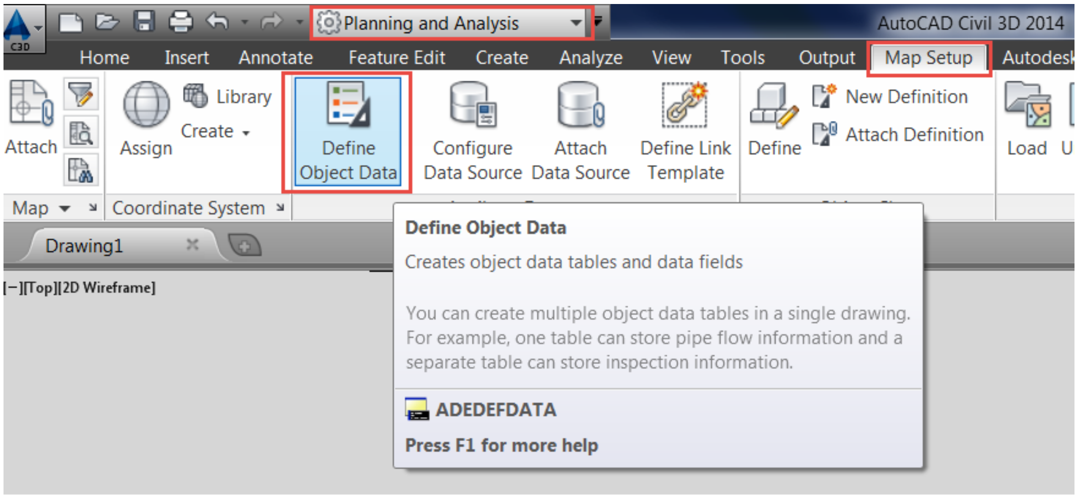
When creating Civil 3D objects, there is an alternative to using Entity Handles. Each Civil 3D object in a drawing can be given a name. For most Civil 3D objects (there are some issues with featurelines), Civil 3D will ensure it is uniquely named within that drawing. This means that if the object name is copied and attached to the object as object data, then the object name can also be used in the database, and a link automatically created between the object in the drawing and the database. In some instances there are two pieces of data that are needed – for example with Pipe Networks. In a Pipe Network, the pipe name is unique within that network (lets say "Pipe1" is assigned). But "Pipe1" could also be used in a different Pipe Network. So to create a unique ID is a combination of the Network Name ("Storm Drain 1" for example) and the Pipe Name ("Pipe1").

The advantage of using this method when dealing with Civil 3D objects is that if we need to delete a pipe and then recreate it, we can manually re-assign the same name and automatically re-link it to the database information.

Data Links – In Practice

Creating the Object Data

Object Data is created in the drawing using the command ADEDEFDATA.



Define Object Data

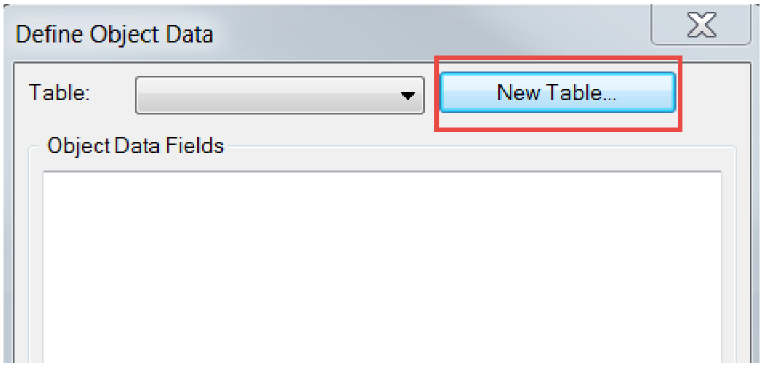
Creates object data tables and data fields

You can create multiple object data tables in a single drawing. For example, one table can store pipe flow information and a separate table can store inspection information.

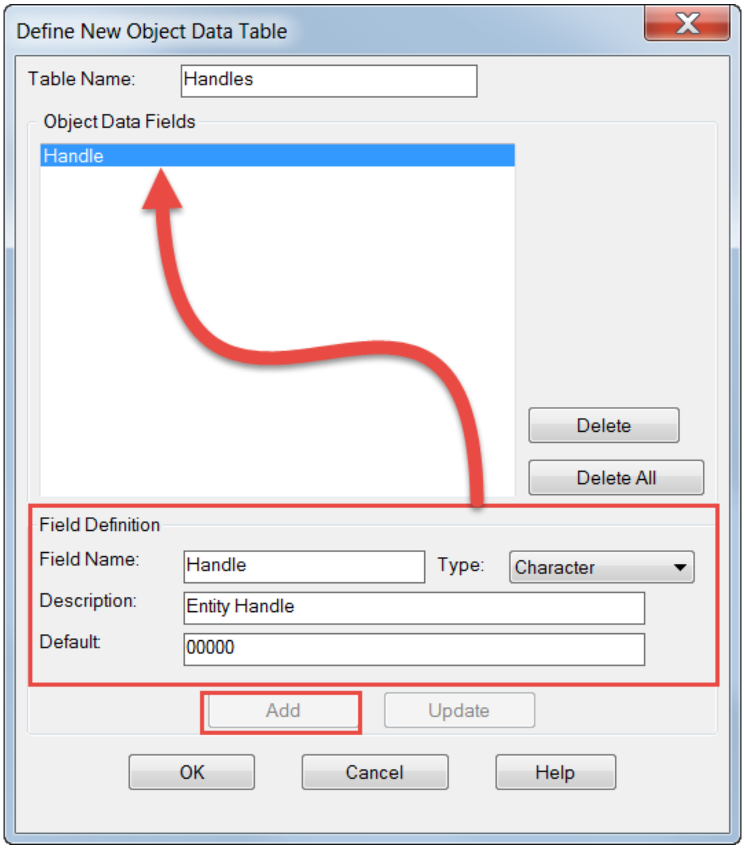
ADEDEFDATA

Press F1 for more help

In the following example, we are using the entity handle as the unique ID, so we create an object data table for this…



And assign the data fields. In this case only one data field is required.



Define New Object Data Table

Table Name: Handles

Object Data Fields

Handle

Field Definition

Field Name: Handle

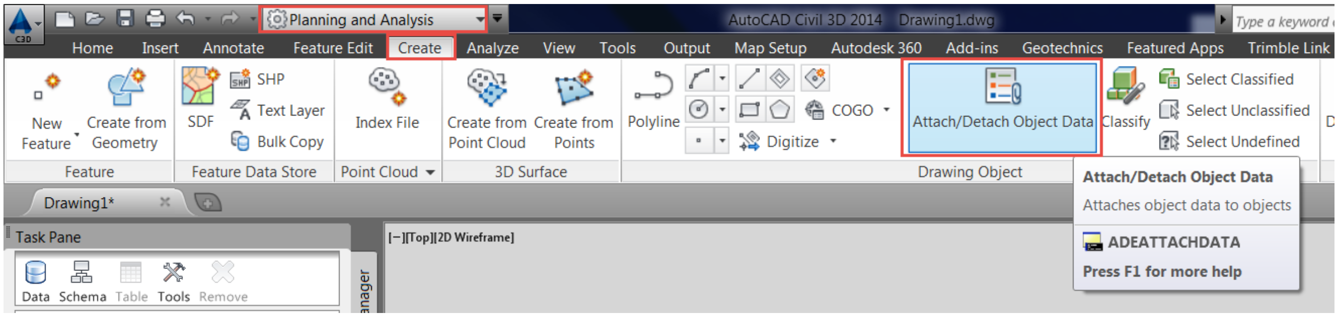
Type: Character

Description: Entity Handle

Default: 00000

Attaching the Object Data

The command to attach Object Data to an object is ADEATTACHDATA.



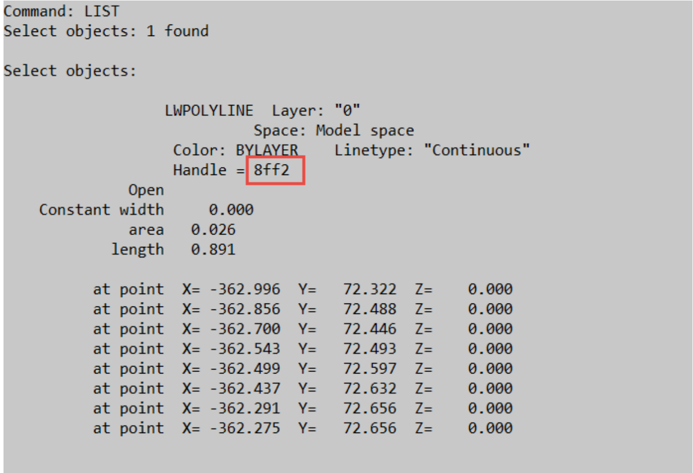
Attach/Detach Object Data

Attaches object data to objects

ADEATTACHDATA

Press F1 for more help

Note: the LIST command can be used to manually copy the handle information….



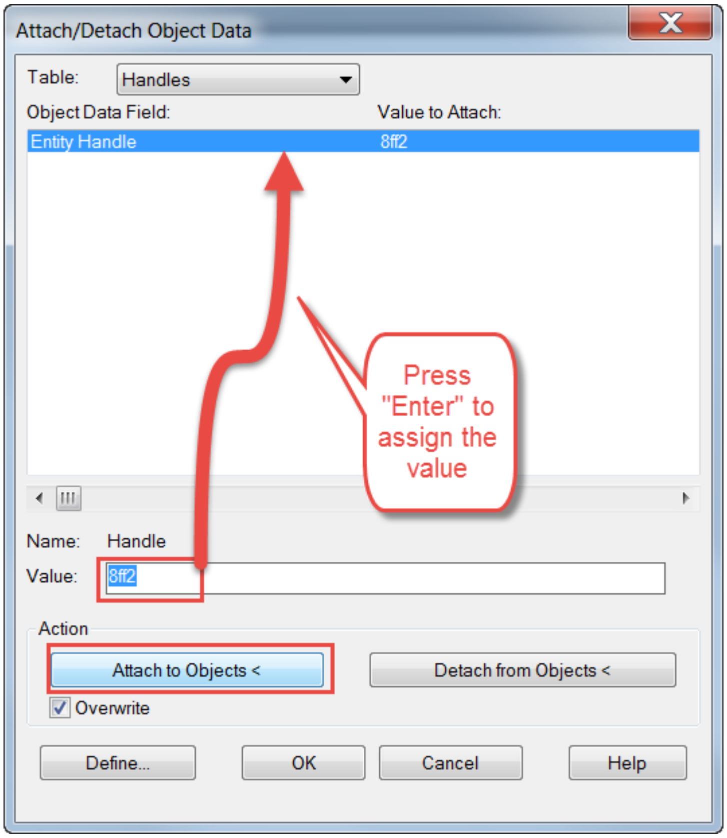
```
Command: LIST
Select objects: 1 found

Select objects:

                  LWPOLYLINE  Layer: "0"
                            Space: Model space
                   Color: BYLAYER    Linetype: "Continuous"
                   Handle = 8ff2
              Open
    Constant width     0.000
              area    0.026
            length    0.891

          at point  X= -362.996  Y=    72.322  Z=    0.000
          at point  X= -362.856  Y=    72.488  Z=    0.000
          at point  X= -362.700  Y=    72.446  Z=    0.000
          at point  X= -362.543  Y=    72.493  Z=    0.000
          at point  X= -362.499  Y=    72.597  Z=    0.000
          at point  X= -362.437  Y=    72.632  Z=    0.000
          at point  X= -362.291  Y=    72.656  Z=    0.000
          at point  X= -362.275  Y=    72.656  Z=    0.000
```

To attach the data, select the object data table to use, pick the data field from the list, and copy the handle into the field. Press the Enter key on the keyboard to apply the change to the data field, then choose “Attach to objects” and select the item in the drawing.



Attach/Detach Object Data

Table: Handles

Object Data Field:	Value to Attach:
Entity Handle	8ff2

Name: Handle

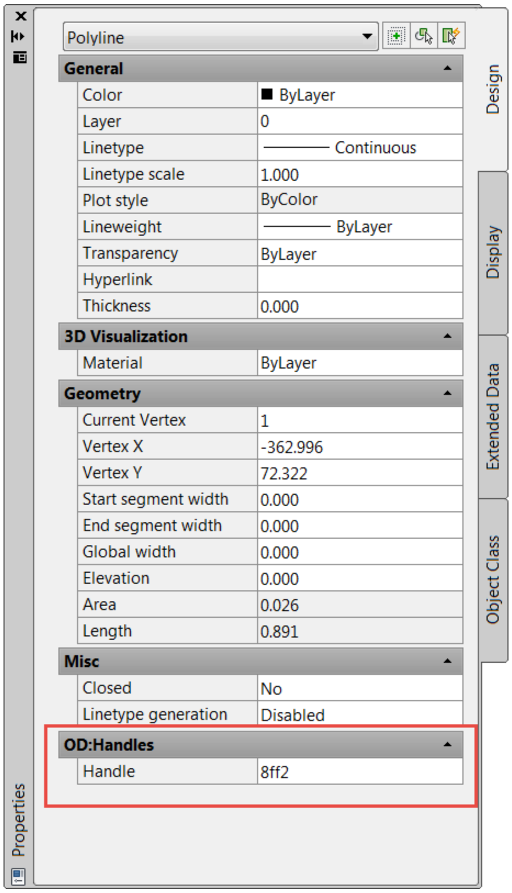
Value: 8ff2

Action

☑ Overwrite

Press "Enter" to assign the value

Check the data has been added by looking at the object properties….

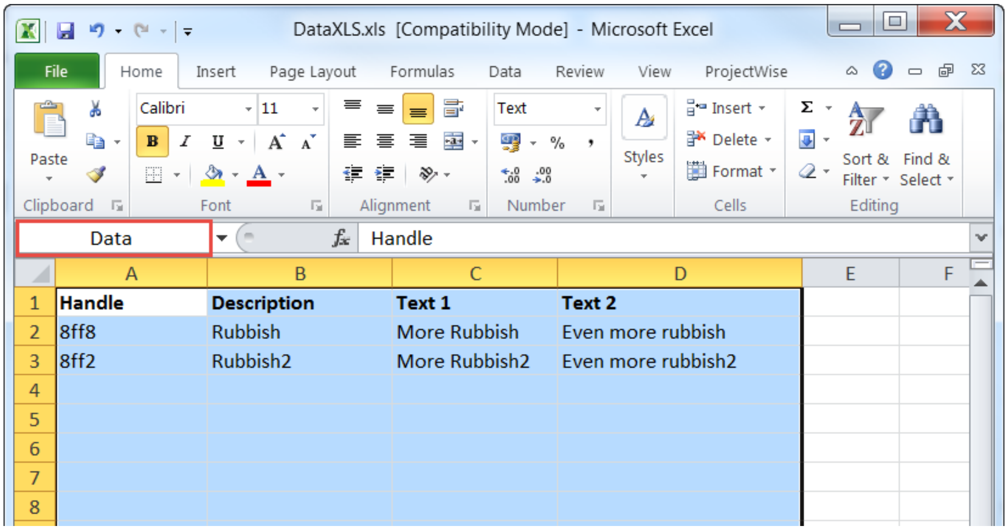


General	
Color	■ ByLayer
Layer	0
Linetype	——— Continuous
Linetype scale	1.000
Plot style	ByColor
Lineweight	——— ByLayer
Transparency	ByLayer
Hyperlink	
Thickness	0.000
3D Visualization	
Material	ByLayer
Geometry	
Current Vertex	1
Vertex X	-362.996
Vertex Y	72.322
Start segment width	0.000
End segment width	0.000
Global width	0.000
Elevation	0.000
Area	0.026
Length	0.891
Misc	
Closed	No
Linetype generation	Disabled
OD:Handles	
Handle	8ff2

Attaching an External Data Source

In this example we are using an Excel Spreadsheet. The disadvantage of using Excel is that the spreadsheet is read-only when added to Map 3D, which means that although links can be created to the drawing, it is not possible to automatically populate the spreadsheet from the Object Data in the drawing – in this example the data rows in the spreadsheet have already been added. In order for a spreadsheet to be used like a database with Map 3D, a “named range” must be created in the sheet before it is added to Map 3D.

To create a “named range” in Excel, create a new workbook and add the required data. Field names are added in row 1. Select the required area of the spreadsheet, and then with the selection still highlighted, type a name in the name box as shown.

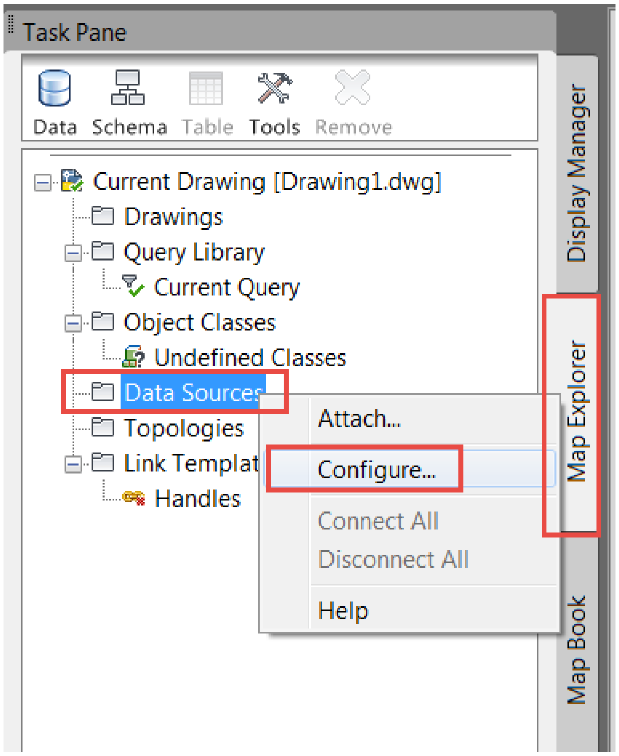


Data | Handle

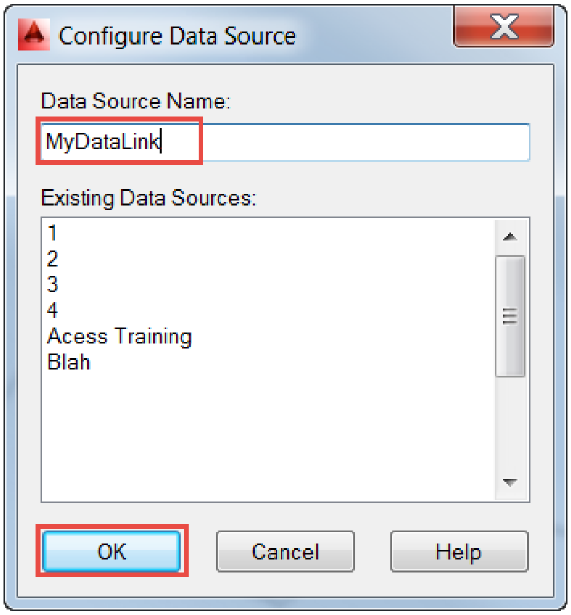
	A	B	C	D	E	F
1	**Handle**	**Description**	**Text 1**	**Text 2**		
2	8ff8	Rubbish	More Rubbish	Even more rubbish		
3	8ff2	Rubbish2	More Rubbish2	Even more rubbish2		
4						
5						
6						
7						
8						

Preparing the Data source

To attach a datasource to Map 3D, Map needs a UDL file to reference the correct data source and driver to use. The UDL can be created from the Map 3D task pane. The taskpane is opened using the command MAPWSPACE.



From the Task Pane, on the Map Explorer tab, click on Data Sources. Right click and choose Configure.



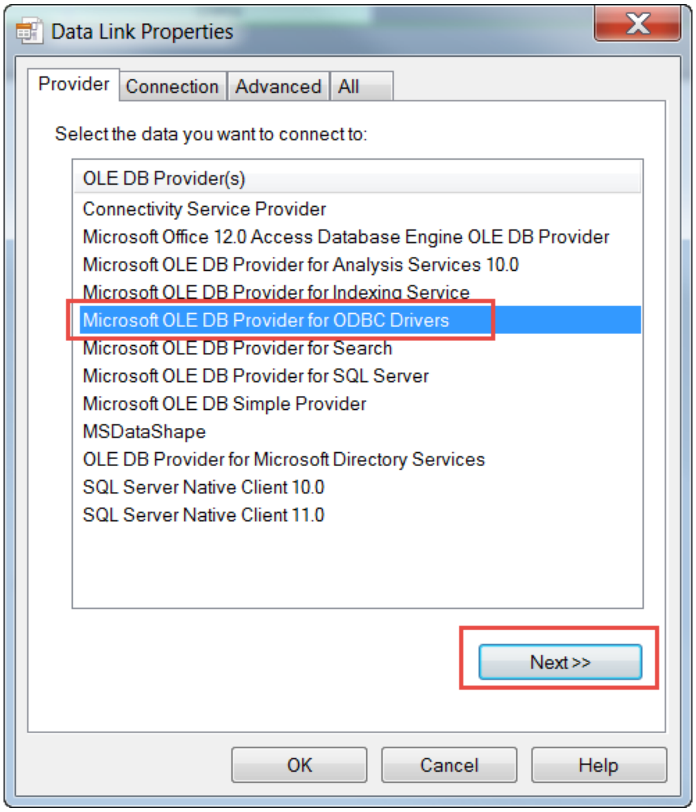
Data Source Name: MyDataLink

Existing Data Sources:

1
2
3
4
Acess Training
Blah

Type in a new name for the UDL file, and click OK. The UDL file enables Map 3D to use the correct connection settings whenever the data source is re-attached.

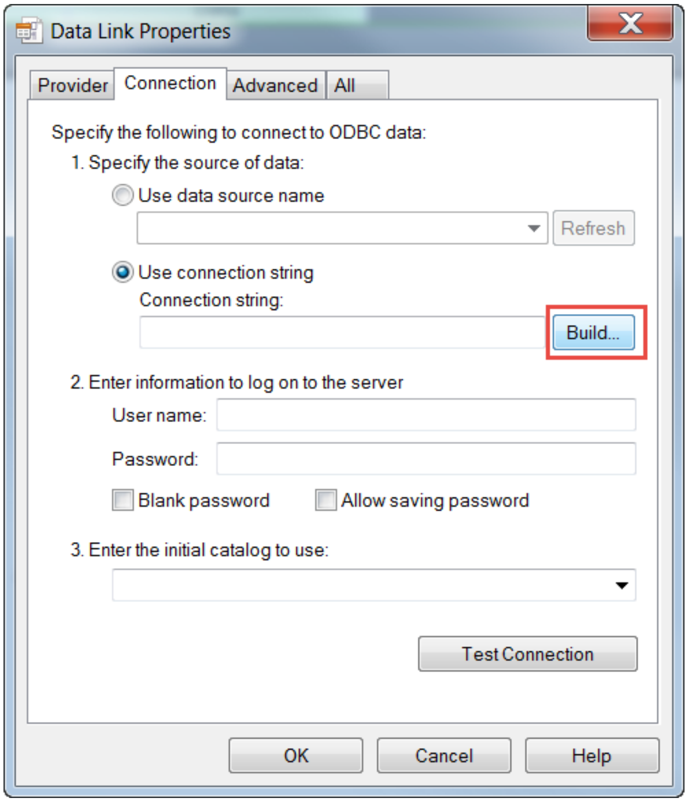
In this example we are using the ODBC driver…



Select the data you want to connect to:

OLE DB Provider(s)
Connectivity Service Provider
Microsoft Office 12.0 Access Database Engine OLE DB Provider
Microsoft OLE DB Provider for Analysis Services 10.0
Microsoft OLE DB Provider for Indexing Service
Microsoft OLE DB Provider for ODBC Drivers
Microsoft OLE DB Provider for Search
Microsoft OLE DB Provider for SQL Server
Microsoft OLE DB Simple Provider
MSDataShape
OLE DB Provider for Microsoft Directory Services
SQL Server Native Client 10.0
SQL Server Native Client 11.0

If a DSN has not already been created, the use the connection string option and choose Build.



Specify the following to connect to ODBC data:

1. Specify the source of data:

○ Use data source name

● Use connection string

Connection string:

2. Enter information to log on to the server

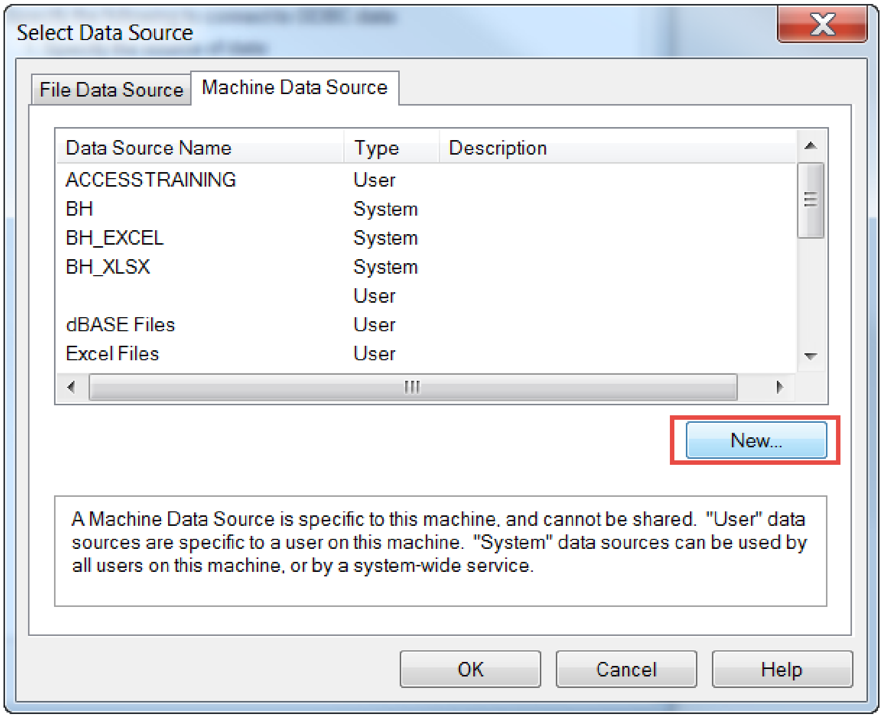
User name:

Password:

□ Blank password □ Allow saving password

3. Enter the initial catalog to use:

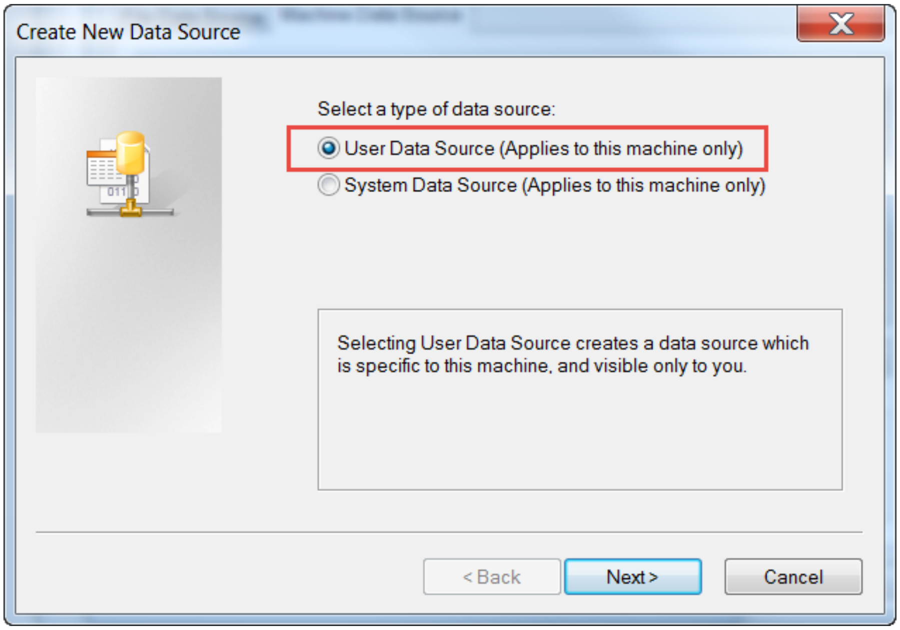
In this example we will configure a Machine Data Source.



Data Source Name	Type	Description
ACCESSTRAINING	User	
BH	System	
BH_EXCEL	System	
BH_XLSX	System	
	User	
dBASE Files	User	
Excel Files	User	

A Machine Data Source is specific to this machine, and cannot be shared. "User" data sources are specific to a user on this machine. "System" data sources can be used by all users on this machine, or by a system-wide service.

Choose the type of DSN to create…



Create New Data Source

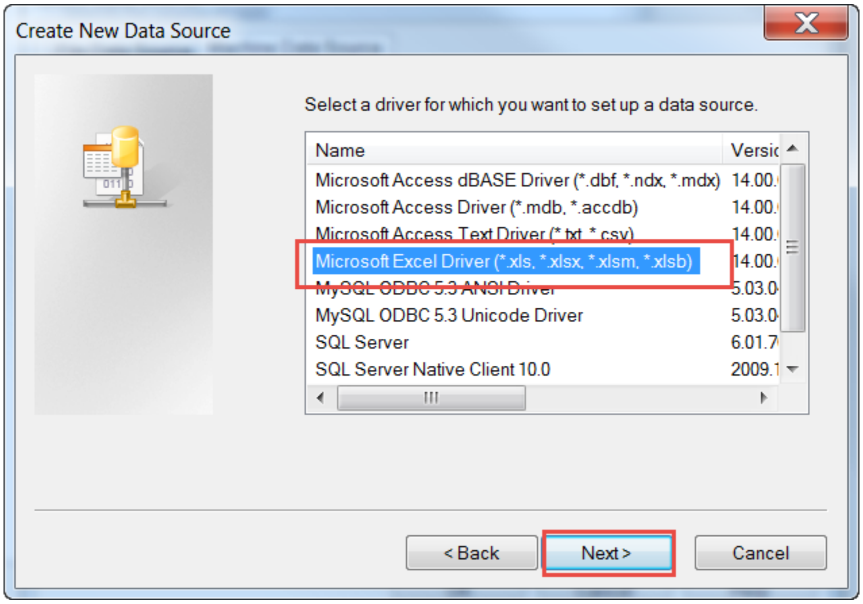
Select a type of data source:

● User Data Source (Applies to this machine only)

○ System Data Source (Applies to this machine only)

Selecting User Data Source creates a data source which is specific to this machine, and visible only to you.

Since our file is in Excel, we will use the Excel driver…

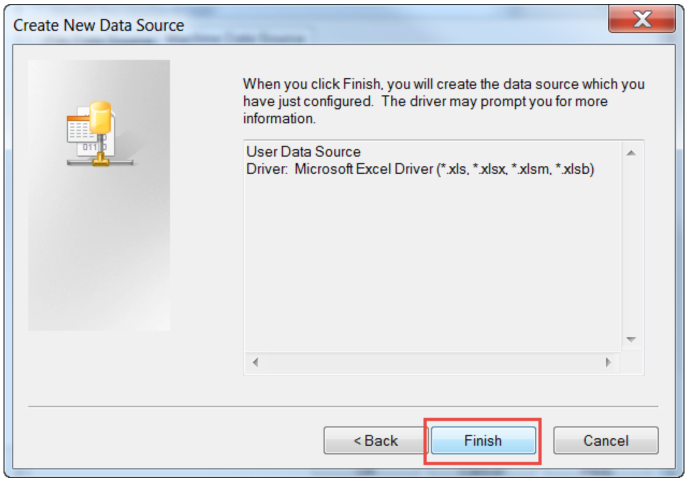


Create New Data Source

Select a driver for which you want to set up a data source.

Name	Versic
Microsoft Access dBASE Driver (*.dbf, *.ndx, *.mdx)	14.00.
Microsoft Access Driver (*.mdb, *.accdb)	14.00.
Microsoft Access Text Driver (*.txt, *.csv)	14.00.
Microsoft Excel Driver (*.xls, *.xlsx, *.xlsm, *.xlsb)	14.00.
MySQL ODBC 5.3 ANSI Driver	5.03.0
MySQL ODBC 5.3 Unicode Driver	5.03.0
SQL Server	6.01.7
SQL Server Native Client 10.0	2009.1

Click Finish.

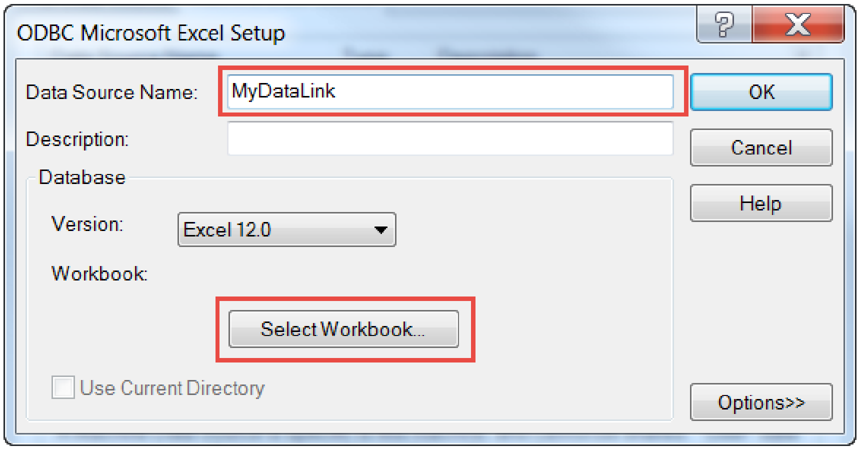


Create New Data Source

When you click Finish, you will create the data source which you have just configured. The driver may prompt you for more information.

User Data Source
Driver: Microsoft Excel Driver (*.xls, *.xlsx, *.xlsm, *.xlsb)

Give the DSN a name, then select the Excel workbook to use.



ODBC Microsoft Excel Setup

Data Source Name: MyDataLink

Description:

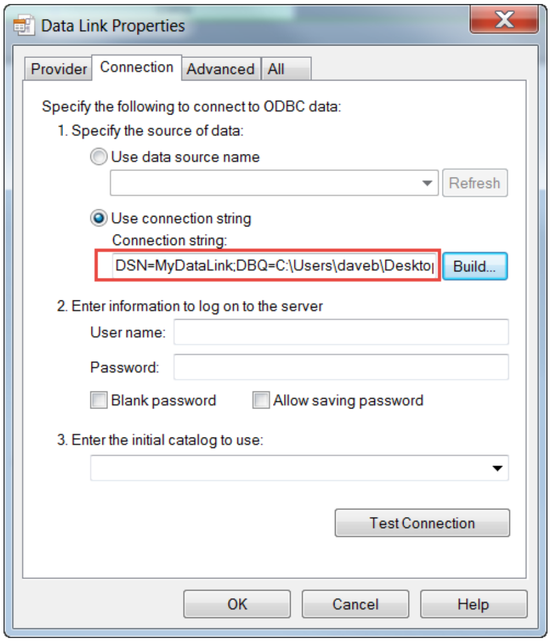
Database

Version: Excel 12.0

Workbook:

□ Use Current Directory

Click OK until you return to the Data Link Properties dialog.



Data Link Properties

Specify the following to connect to ODBC data:

1. Specify the source of data:

○ Use data source name

● Use connection string

Connection string: DSN=MyDataLink;DBQ=C:\Users\daveb\Deskto|

2. Enter information to log on to the server

User name:

Password:

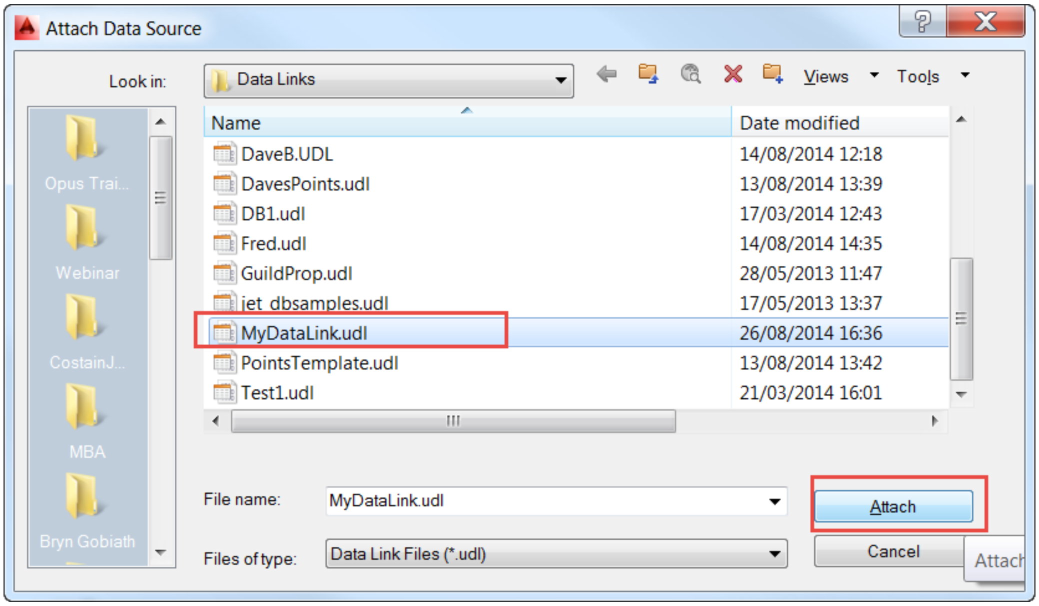
□ Blank password □ Allow saving password

3. Enter the initial catalog to use:

Check the connection string has been added, then click OK.

Attaching the Data Source

Go back to Map Explorer, right click on Data Sources and choose “Attach”. Select the new UDL file and choose “Attach”.



Look in: Data Links

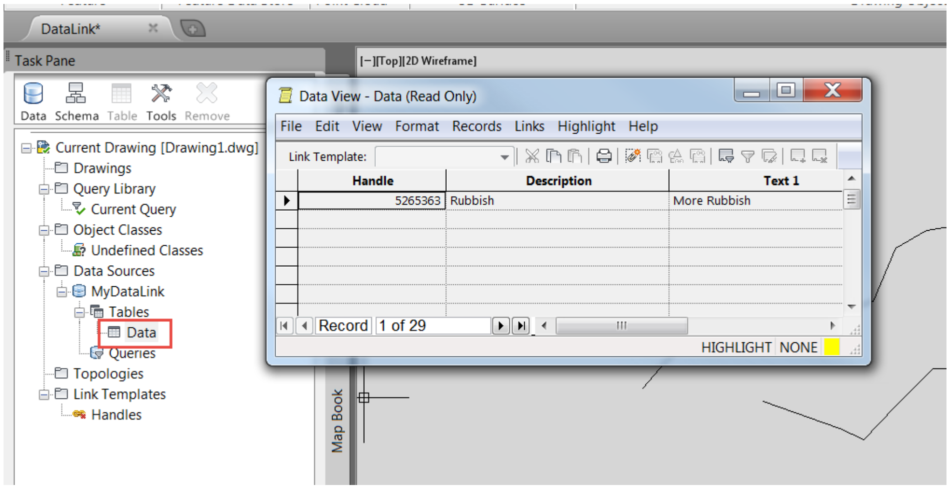
Name	Date modified
DaveB.UDL	14/08/2014 12:18
DavesPoints.udl	13/08/2014 13:39
DB1.udl	17/03/2014 12:43
Fred.udl	14/08/2014 14:35
GuildProp.udl	28/05/2013 11:47
jet_dbsamples.udl	17/05/2013 13:37
MyDataLink.udl	26/08/2014 16:36
PointsTemplate.udl	13/08/2014 13:42
Test1.udl	21/03/2014 16:01

File name: MyDataLink.udl

Files of type: Data Link Files (*.udl)

The data source will now be attached in Map Explorer.

Double click on the table name in Map Explorer (this is the named range in the Excel Spreadsheet) to see the table view in Map 3D.

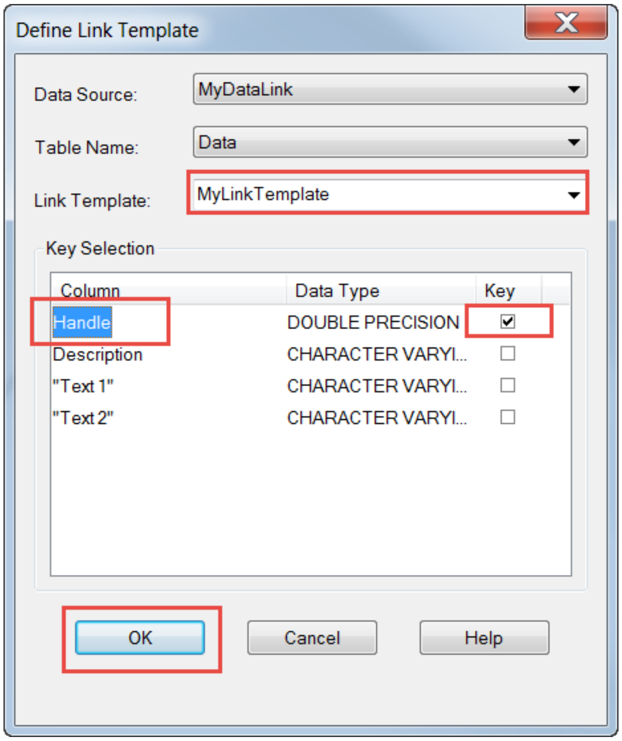


Handle	Description	Text 1
5265363	Rubbish	More Rubbish

Record 1 of 29

Creating a Link Template

In Map Explorer, right click on Link Templates and choose Define Link Template.



Define Link Template

Data Source: MyDataLink

Table Name: Data

Link Template: MyLinkTemplate

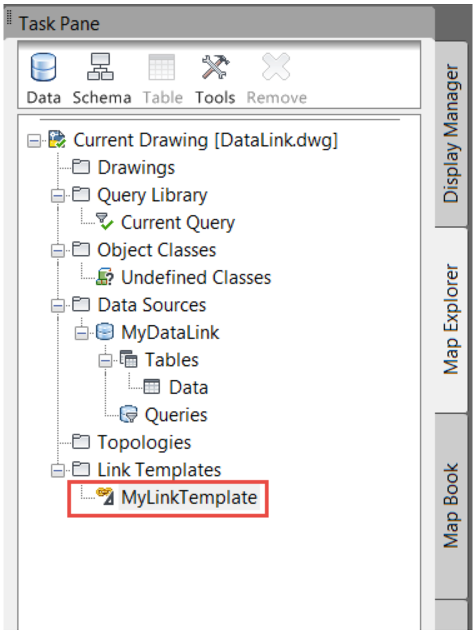
Key Selection

Column	Data Type	Key
Handle	DOUBLE PRECISION	☑
Description	CHARACTER VARYI...	□
"Text 1"	CHARACTER VARYI...	□
"Text 2"	CHARACTER VARYI...	□

Ensure the correct Data Source and Table Name are selected, and enter a name for the Link Template.

Select the Key field for the data link.

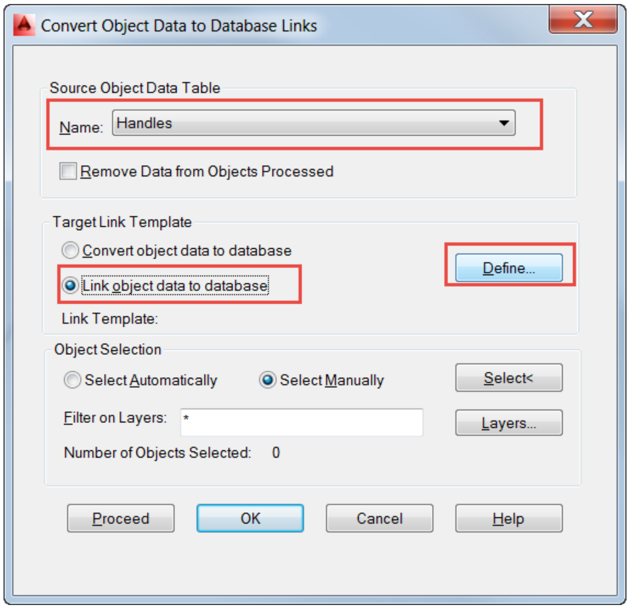
The Link Template will then be visible in Map Explorer.



Create Links Automatically

Although it is possible to manually select a row in the data table and link the row to an object in the drawing, a more efficient process is to create the links automatically.

To create links automatically, run the command MAPOD2ASE. This command is not available on the ribbon.



Convert Object Data to Database Links

Source Object Data Table

Name: Handles

□ Remove Data from Objects Processed

Target Link Template

○ Convert object data to database

● Link object data to database

Link Template:

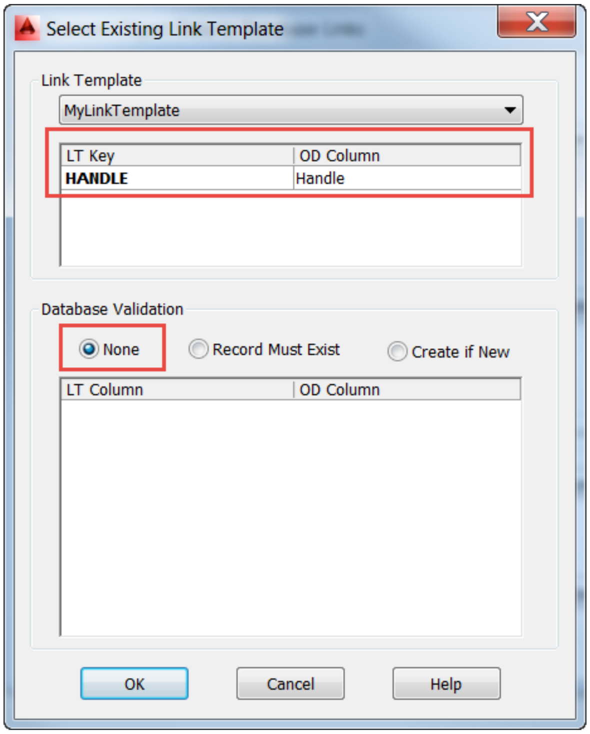
Object Selection

○ Select Automatically ● Select Manually

Filter on Layers: *

Number of Objects Selected: 0

Select the Object Data in the drawing to use, choose Link Object Data to Database, then click on the Define button.



Select Existing Link Template

Link Template

MyLinkTemplate

LT Key	OD Column
HANDLE	Handle

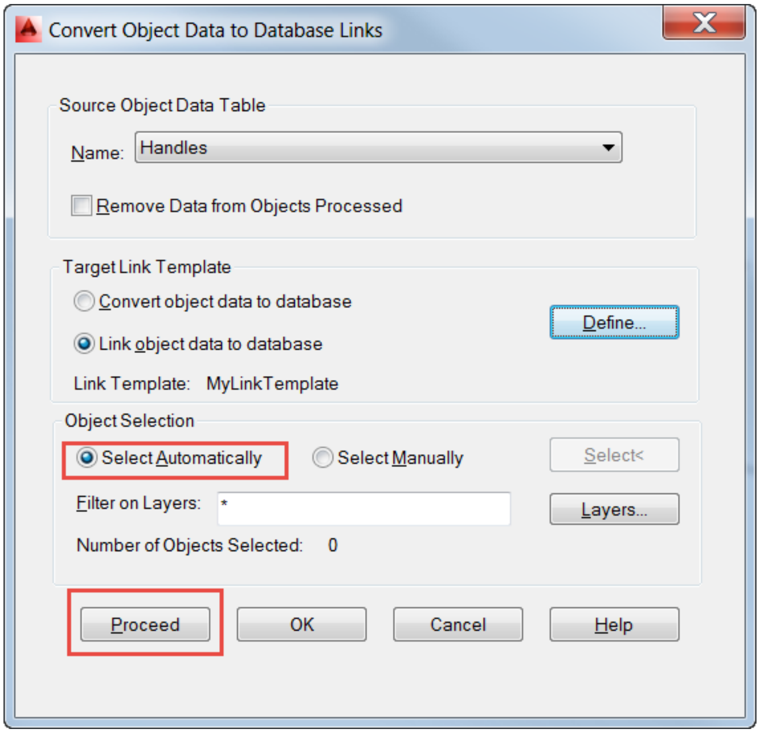
Database Validation

● None ○ Record Must Exist ○ Create if New

LT Column	OD Column

Ensure that the key fields in the data source and the Object Data are correct, and choose None for the Database Validation. (NOTE: If a database type datastore was being used, it would be possible to create additional rows in the table as necessary. However, with an Excel data source, the table is effectively read-only, which means that we can create links but not update the table).

Back in the Convert Object Data to Database dialog, choose Select Automatically and then Proceed.



Convert Object Data to Database Links

Source Object Data Table

Name: Handles

☐ Remove Data from Objects Processed

Target Link Template

○ Convert object data to database

● Link object data to database

Link Template: MyLinkTemplate

Object Selection

● Select Automatically ○ Select Manually

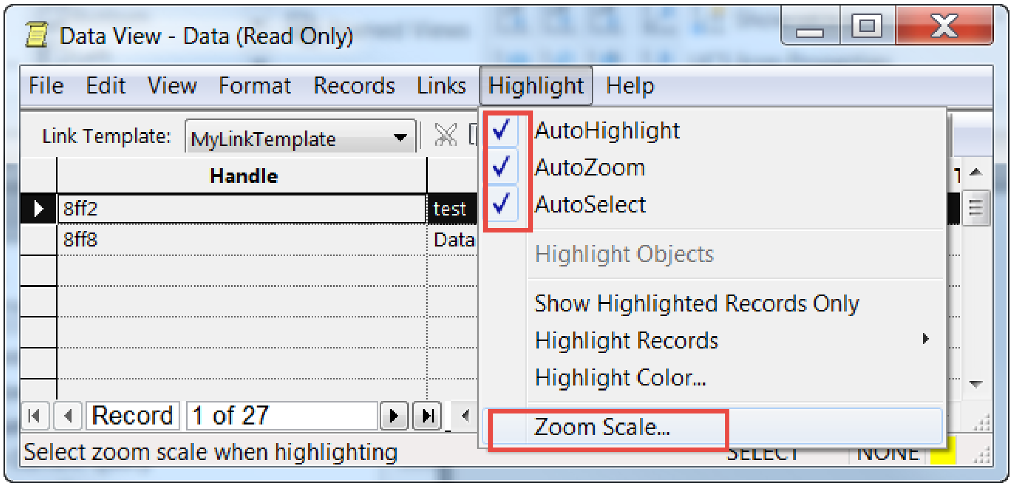
Filter on Layers: *

Number of Objects Selected: 0

The links to the database should now be created.

Validating the Data Links

In the Data View (double click on the table or the link template in Map Explorer), turn on the options in the Highlight menu as shown below. Set the zoom Scale to be 50% of the screen.



Data View - Data (Read Only)

Link Template: MyLinkTemplate

Handle	
8ff2	test
8ff8	Data

Record 1 of 27

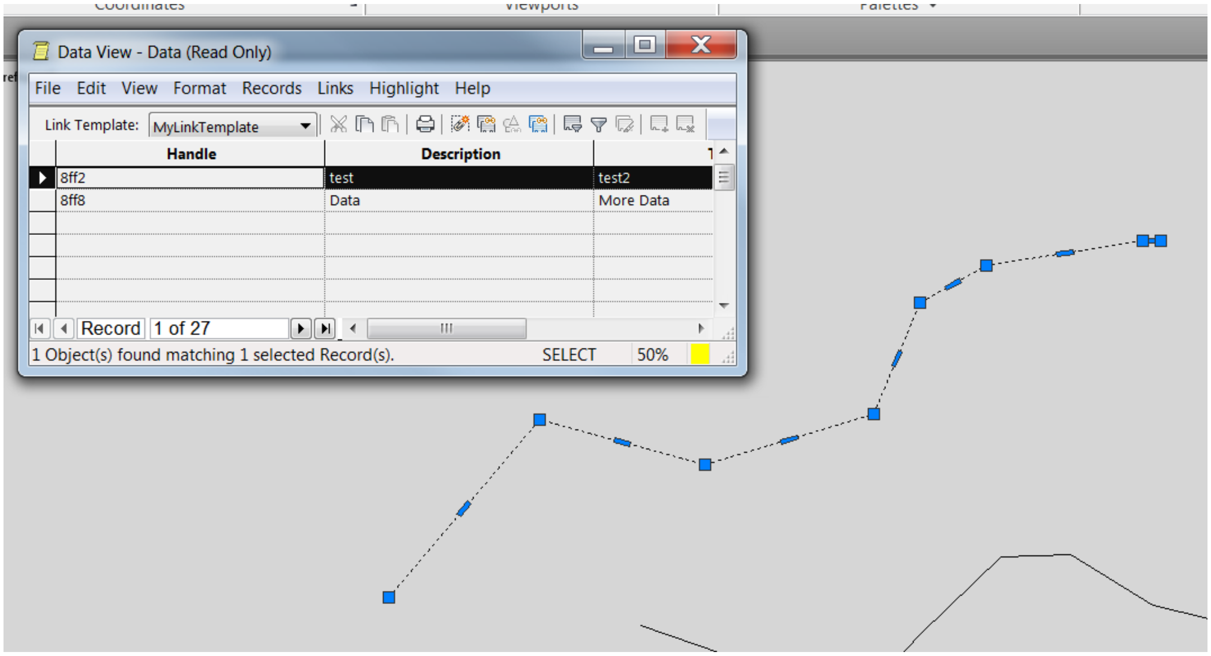
☑ AutoHighlight

☑ AutoZoom

☑ AutoSelect

Select zoom scale when highlighting

Select a row in the table, and the object should be selected in the drawing automatically.


Data View - Data (Read Only)
File Edit View Format Records Links Highlight Help
Link Template: MyLinkTemplate

Handle	Description	
8ff2	test	test2
8ff8	Data	More Data

Record 1 of 27

1 Object(s) found matching 1 selected Record(s). SELECT 50%

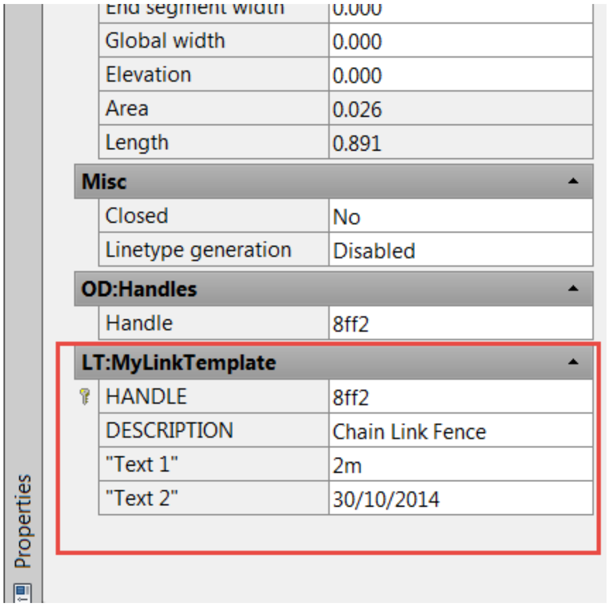
Editing the Data

Although with Excel, data cannot be edited in the Data View, it is possible to change the data by editing the linked object properties.

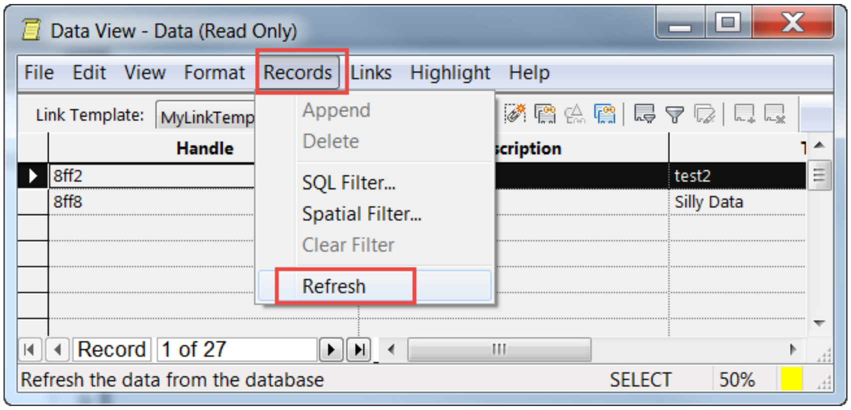
With a linked item selected in the table, and the corresponding object selected in the drawing, right click in the model space and call up the AutoCAD properties dialog.

The Link Template data is shown in the Properties panel.

The data can be edited in the Properties panel and the Data View will update after refreshing the records.

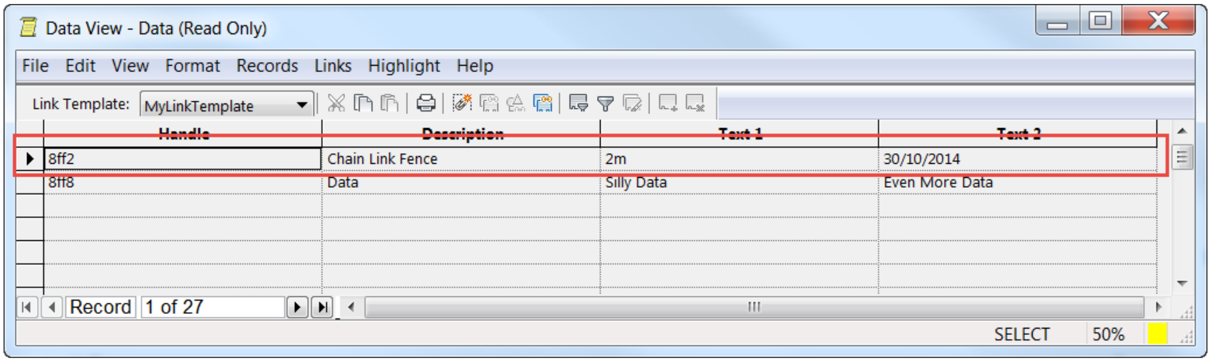


End segment width	0.000
Global width	0.000
Elevation	0.000
Area	0.026
Length	0.891
Misc	
Closed	No
Linetype generation	Disabled
OD:Handles	
Handle	8ff2
LT:MyLinkTemplate	
HANDLE	8ff2
DESCRIPTION	Chain Link Fence
"Text 1"	2m
"Text 2"	30/10/2014

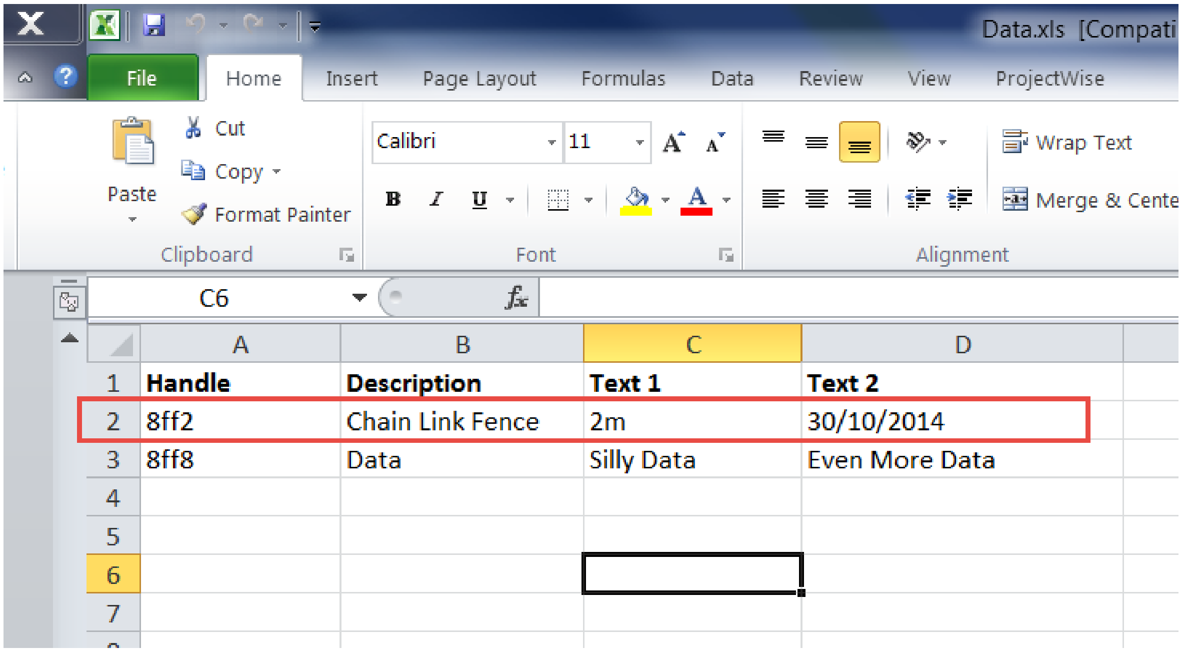


Handle	cription	
8ff2		test2
8ff8		Silly Data

Refresh the data from the database



Handle	Description	Text 1	Text 2
8ff2	Chain Link Fence	2m	30/10/2014
8ff8	Data	Silly Data	Even More Data



	A	B	C	D
1	Handle	Description	Text 1	Text 2
2	8ff2	Chain Link Fence	2m	30/10/2014
3	8ff8	Data	Silly Data	Even More Data
4				
5				
6				
7				

Automatically Creating Object Data

Example Lisp Routine

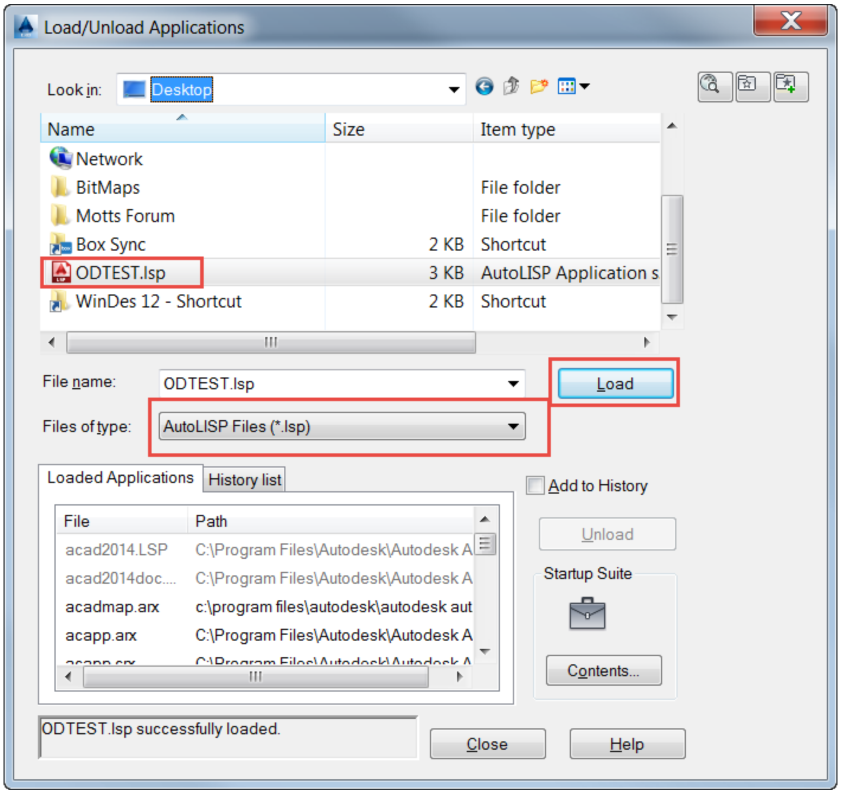
The example Lisp Routine included in the class dataset will prompt the user to select some objects in the drawing. It will then create a new Object Data table called TEST, and populate the Handle field with the data handle of the object and then attach the object data to the item.

This routine could be modified to use some other unique ID, and potentially even use Civil 3D object names instead of entity handles.

Running the Lisp Routine

To run the routine, type the command APPLOAD.

Choose the routine and click Load.



Look in: Desktop

Name	Size	Item type
Network		
BitMaps		File folder
Motts Forum		File folder
Box Sync	2 KB	Shortcut
ODTEST.lsp	3 KB	AutoLISP Application s
WinDes 12 - Shortcut	2 KB	Shortcut

File name: ODTEST.lsp

Files of type: AutoLISP Files (*.lsp)

File	Path
acad2014.LSP	C:\Program Files\Autodesk\Autodesk A
acad2014doc....	C:\Program Files\Autodesk\Autodesk A
acadmap.arx	c:\program files\autodesk\autodesk aut
acapp.arx	C:\Program Files\Autodesk\Autodesk A

ODTEST.lsp successfully loaded.

Type ODTEST on the command line to run the routine.

Lisp Routine

```
(Defun c:ODTEST()                              ;Routine called ODTEST

; Define new table

(setq tabldefn

   '(("tablename" . "TEST")

      ("tabledesc" . "New Sample Table")

      ("columns"

           ; Define a field

           (("colname" . "Handle")

            ("coldesc" . "entity Handle")

            ("coltype" . "character")

            ("defaultval" . "Default Value"))

           ; Define more fields as needed

  )))

; Create the new table

(ade_oddefinetab tabldefn)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;add new fields to table "TEST"

(setq newfields

   '("columns"

        ; Define a character field

        (("colname" . "NEWFIELD1")

         ("coldesc" . "New Field 1 Description")

         ("coltype" . "character")

         ("defaultval" . "0"))
        ; Define an integer field
        (("colname" . "NEWFIELD2")
         ("coldesc" . "New Field 2 Description")
         ("coltype" . "integer")
         ("defaultval" . 1))
        ; Define a point field
        (("colname" . "NEWFIELD3")
         ("coldesc" . "New Field 3 Description")
         ("coltype" . "point")
         ("defaultval" . "4.426217, 7.991379, 1.726213"))
        ; Define a real field
        (("colname" . "NEWFIELD4")
         ("coldesc" . "New Field 4 Description")
         ("coltype" . "real")
         ("defaultval" . 1.2345)) ) )
; Add new fields to existing table
(ade_odaddfield "TEST" newfields);
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;Get some entities ....
(Setq ss1(ssget))   ;get some entities and save them in selection set ss1
(setq loop 0)                                       ;set a counter position 0

                          ;if some entities are selected, then.....
(while ( > (sslength ss1) loop)    ;....provided count is less than or equal to the number of entities in ss1
          (progn                                    ; progression - do the next statements for each entity
            ;(princ loop)
     (setq ent1 (ssname ss1 loop))                                      ; set ent1 as the next entity
(setq edata (entget ent1))                                                   ; get the entity data
```

```
(setq ehandle (cdr(assoc 5 edata)))                    ;get the entity handle


;;;;;;;; creates a new record in table TEST

(setq rec_id(ade_odnewrecord "Test") )

(princ rec_id)

;;;;;;;assigns a value to a field (Handle) in a new record


(ade_odpresetfield rec_id "Handle" ehandle)          ; put handle data ehandle in the data record


;;;;;; attaches a new record to an object


(ade_odattachrecord ent1 rec_id)                       ; attach the new record to the object


;;;;;;; MUST release a new record after creating it

(ade_odfreerec rec_id)

      (setq loop (1+ loop))                                  ; increment the counter loop

)                                                            ;end progression

)                                                            ;end while

)                                                            ;end routine
```