Dynamic Feature Scoring

Bentley Map has the ability to open and display the contents of a MicroStation design file. However, to use the export tools in the *Map Interoperability* dialog, the *Map Manager*, the *Data Browser*, or publish the elements for *Bentley Map Mobile*, the elements need to be seen as XFM features.

To do that, you have two choices:

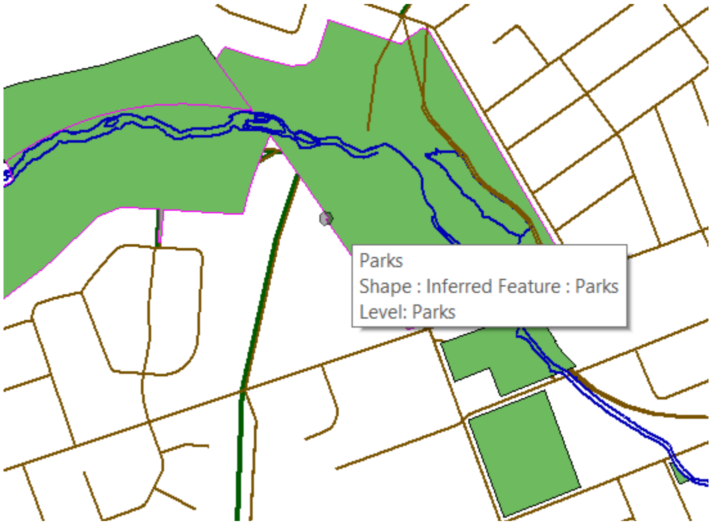
1) Use the *Promote* tool and convert the elements to native XFM features.
2) Use *Dynamic Feature Scoring (DFS)* which simply identifies or flags elements as XFM features without actually making changes to the elements.

Tip: DFS is a display-only function and does not make changes to the actual file. For this reason, it also works on read-only files.

Tip: When DFS is applied to elements in a design file, you retain the ability to use the Map Manager and its functionality. This includes symbology overrides and thematic resymbolization.

Enabling or Disabling DFS

DFS is enabled by default is applied to all new elements and when references are attached. It results in the word *Inferred Feature* when you hover your mouse over a non-feature element. This allows the plain non-feature elements to be available for use with Bentley Map tools as described above.

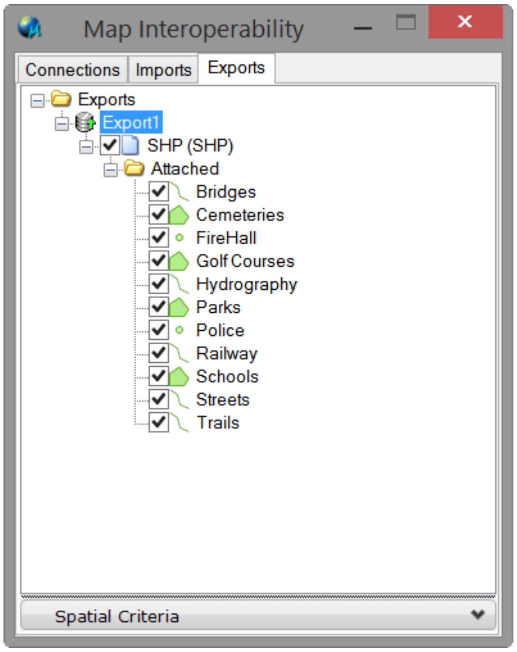


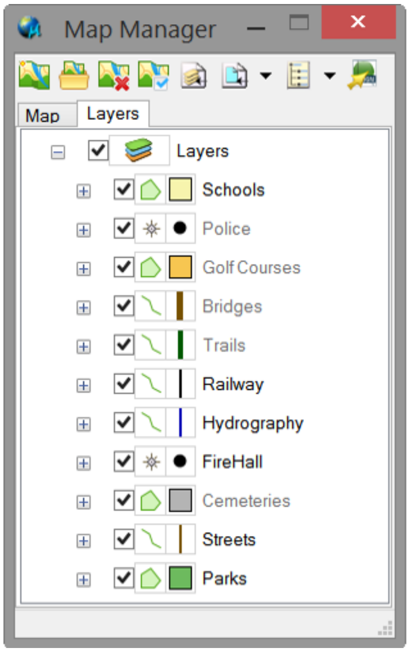
Parks
Shape : Inferred Feature : Parks
Level: Parks

In addition to the *Inferred Feature* note, you can also perform the following test to verify the presence of DFS.

> From the *Map Interoperability* dialog, navigate to the *Exports* tab. Create a new export and add a Shapefile directory.
>
> Expand the *Attached* node and notice one feature class per inferred feature in the design file.

Likewise, the inferred features are available for use with the Map Manager.



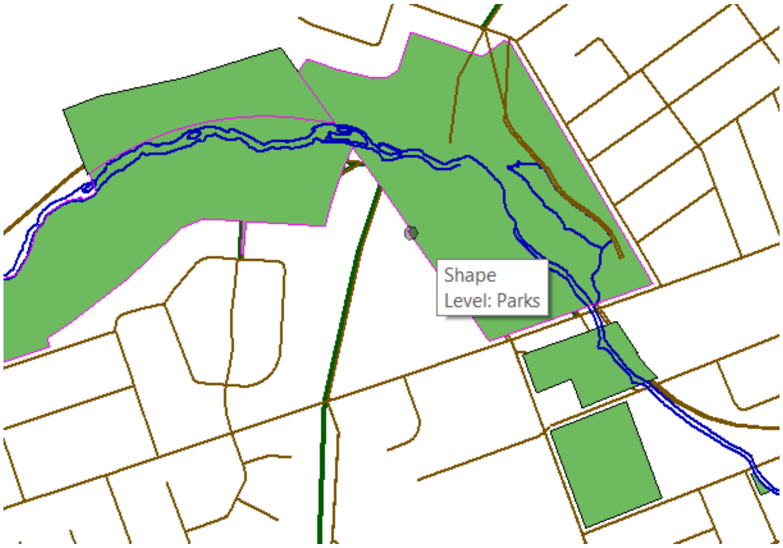


Defining MS_GEOXFM_NO_DYNAMICSCOREDFEATURES will turn off the DFS engine which results in the elements being identified the same as in MicroStation.

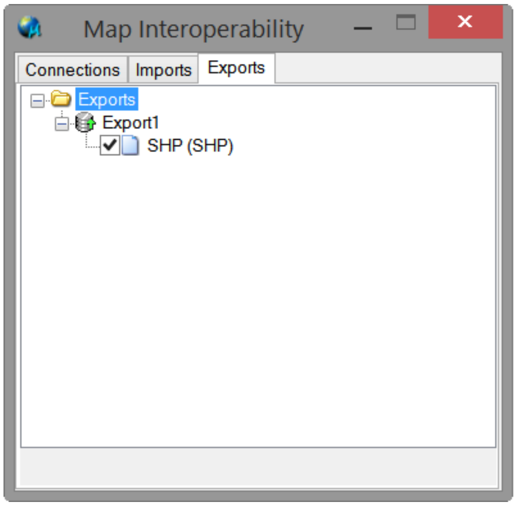
Since this variable only needs to be defined, you can set it either of the following:

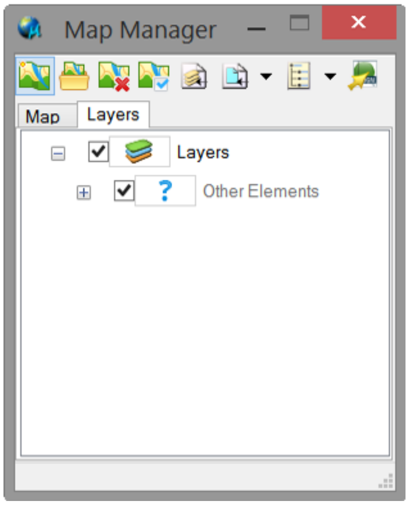
MS_GEOXFM_NO_DYNAMICSCOREDFEATURES = 0
MS_GEOXFM_NO_DYNAMICSCOREDFEATURES = 1

Hint: For testing and learning purposes, set this as a temporary variable with this key in: *EXPAND SET MS_GEOXFM_NO_DYNAMICSCOREDFEATURES = 1 .* It will require you to re-open the design file, but is disabled when a new session of Bentley Map is started.



Since the elements are not being read as features, they are not available to Bentley Map tools such as Exporting to shapefiles or the Map Manager.



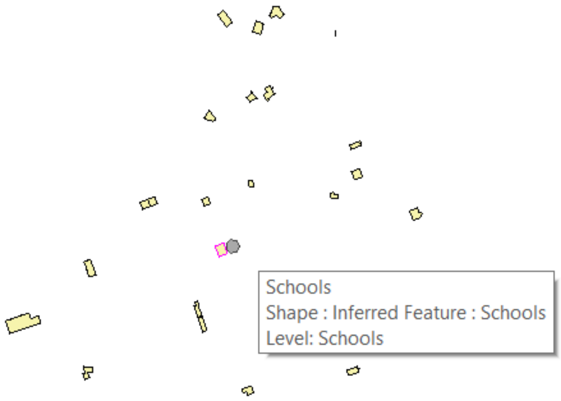


The First Element Found

When you start Bentley Map, and if there are no native features found, *it will look at the first element on each level and set the class type to match*.   In other words, if there is a line and a polygon on the Schools level, only the line is found and the feature class is set to line.

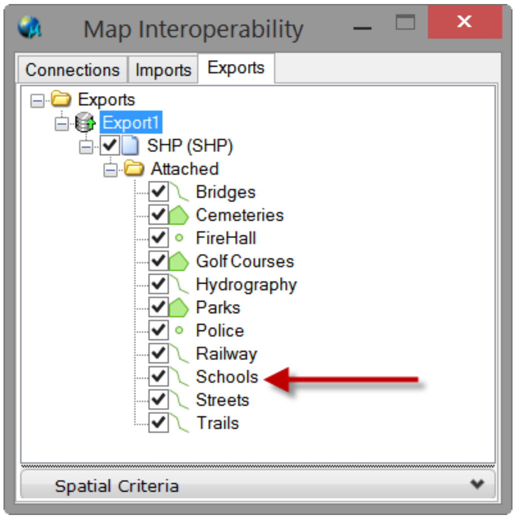
Let's take a closer look.

At first glance, the *Schools* level appears to contain polygons to represent individual school grounds.



Schools
Shape : Inferred Feature : Schools
Level: Schools

However, creating a Shapefile directory shows that the School level is being read as a linear feature class when it should be a polygon.

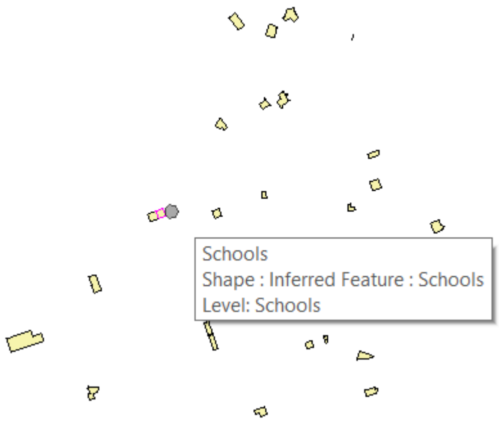


Map Interoperability

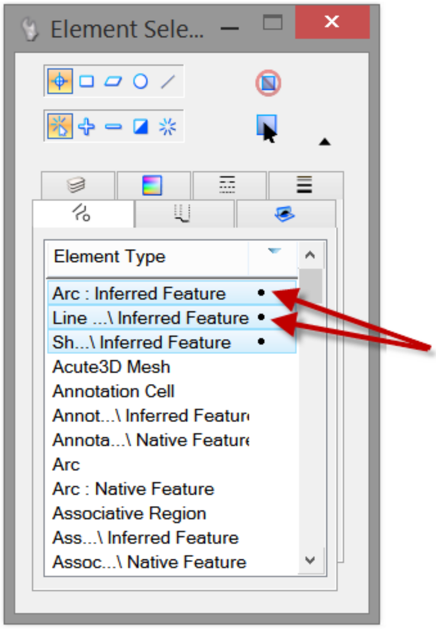
- Exports
  - Export1
    - ☑ SHP (SHP)
      - Attached
        - ☑ Bridges
        - ☑ Cemeteries
        - ☑ FireHall
        - ☑ Golf Courses
        - ☑ Hydrography
        - ☑ Parks
        - ☑ Police
        - ☑ Railway
        - ☑ Schools
        - ☑ Streets
        - ☑ Trails

When this is encountered, it's up to you to resolve the issue which may require a detailed analysis of each level. This is especially true when CAD standards have not been followed and levels contain a blend of element types.

In this example, the School level appears to have shapes, lines and arcs. Before proceeding, the lines and arcs should be evaluated and cleaned-up as necessary.



Schools
Shape : Inferred Feature : Schools
Level: Schools

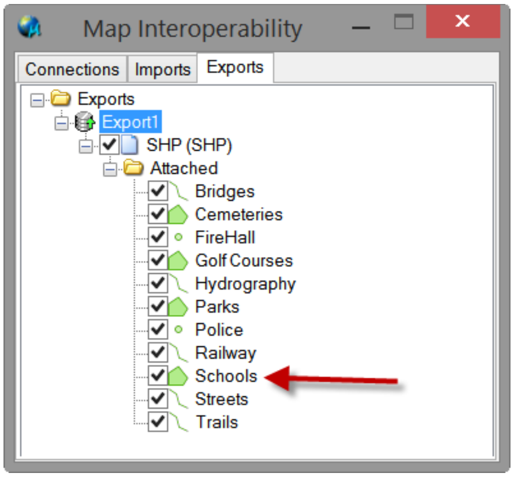


Element Sele...

Element Type

- Arc : Inferred Feature •
- Line ...\ Inferred Feature •
- Sh...\ Inferred Feature •
- Acute3D Mesh
- Annotation Cell
- Annot...\ Inferred Featur
- Annota...\ Native Feature
- Arc
- Arc : Native Feature
- Associative Region
- Ass...\ Inferred Feature
- Assoc...\ Native Feature

Once cleaned, a new export shows the School level being treated as a polygon feature class which is correct for this particular design file.



Map Interoperability

Connections | Imports | Exports

```
Exports
  Export1
    ☑ SHP (SHP)
      Attached
        ☑ Bridges
        ☑ Cemeteries
        ☑ FireHall
        ☑ Golf Courses
        ☑ Hydrography
        ☑ Parks
        ☑ Police
        ☑ Railway
        ☑ Schools
        ☑ Streets
        ☑ Trails
```

Behind the scenes

DFS uses *scoring rules* to dynamically evaluate the elements at run time. For those familiar with MicroStation GeoGraphics, DFS is similar to the *Attach Best Feature* command in which various element criteria was evaluated in order to meet or exceed a minimum score before feature tagging selected elements.

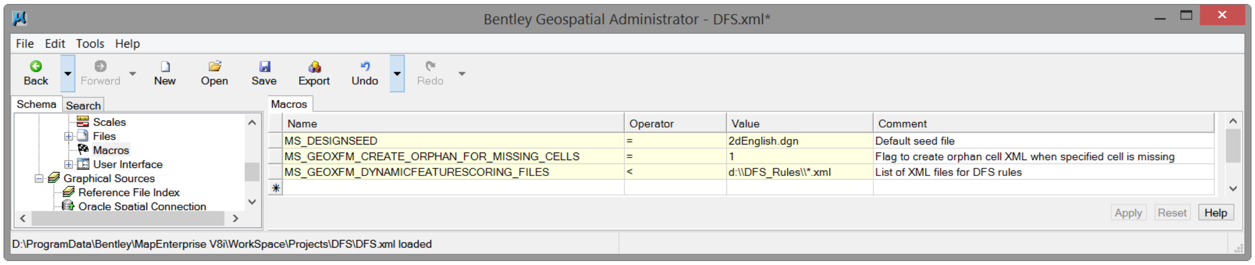
A similar element evaluation and scoring system used with DFS with the most notable differences being that all evaluations are made at runtime as opposed to a batch method, and no persisted feature tags are required.

DFS Rules Configuration

During Bentley Map file open and load events, all non-XFM feature elements are evaluated against the set of DFS rules. DFS rules can be manually defined by you, by other processes such as Bentley Geospatial Administrator and DGN2SDO, or be generated on-the-fly in order to maximize the amount of data which can be consumed without the need for bulk data conversion.

The DFS engine maintains a feature rules list read from XML files specified in the configuration variable *MS_GEOXFM_DYNAMICFEATURESCORING_FILES*

Edit this variable to point to a list of XML files containing your DFS rules. This variable can be added to the *Macros* node of your project schema, or to the User Configuration File (.ucf) from the Workspace Configuration settings.



Name	Operator	Value	Comment
MS_DESIGNSEED	=	2dEnglish.dgn	Default seed file
MS_GEOXFM_CREATE_ORPHAN_FOR_MISSING_CELLS	=	1	Flag to create orphan cell XML when specified cell is missing
MS_GEOXFM_DYNAMICFEATURESCORING_FILES	<	d:\\DFS_Rules*.xml	List of XML files for DFS rules
*			

D:\ProgramData\Bentley\MapEnterprise V8i\WorkSpace\Projects\DFS\DFS.xml loaded

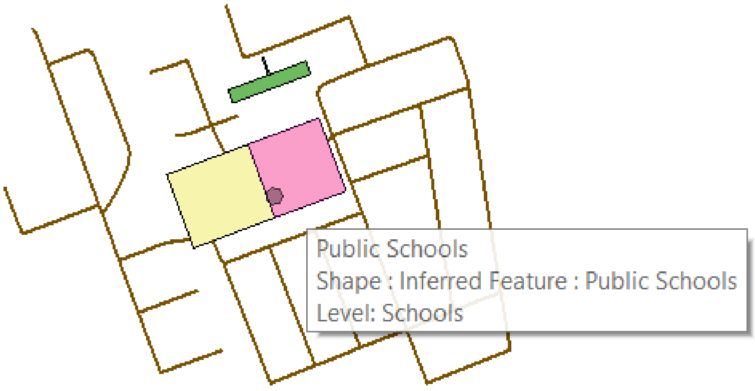
Feature Rule Scoring

A feature rule is considered a match to an element when the combined score for the feature's rule(s) is greater than or equal to the *minScore* defined in the DFS rules file.

The feature's rules are evaluated and scored if evaluated to true until either the total score reaches *minScore* or no more feature rules exist.

The following is an example of a very simple rule defined in an XML text file. It specifies that if a shape element (Type *6*) is found on a level named *Schools*, it will be given a feature class name *Public Schools.*

```
<FeatureScoringRules minScore="100">
        <Feature useCriteria="" name="Public Schools">
                <Rule type="Element" score="100" levelName="Schools" elementType="6" />
        </Feature>
</FeatureScoringRules>
```



Public Schools
Shape : Inferred Feature : Public Schools
Level: Schools

If no rule matches the element, the default behavior is to identify the element as a named feature based upon the level or cell name. This behavior can be overwritten by the following variables:

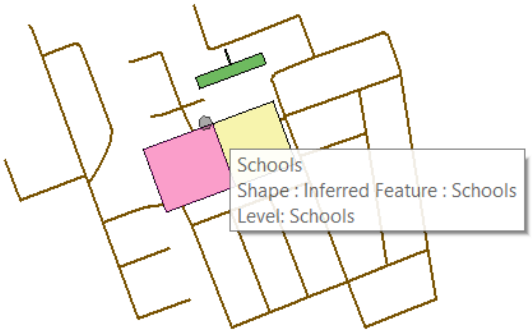
MS_GEOXFM_NO_LEVELFEATURES = 1
MS_GEOXFM_NO_CELLNAMEFEATURES = 1

Inferred Level Features (MS_GEOXFM_NO_LEVELFEATURES)

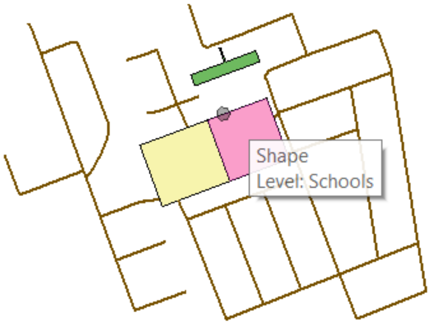
By default, non-cell inferred features are defined using the level name as the feature class name. In addition:

- If the level name contains invalid characters, they are converted to underscores.
- If the level name begins with a number, the feature name is prefixed with *Level_*
- If the level name retrieval by the level ID fails, then the feature name will be *Level_ + LevelID*

To override this behavior, set MS_GEOXFM_NO_LEVELFEATURES = 1



DFS Enabled:
MS_GEOXFM_NO_LEVELFEATURES undefined



DFS Enabled:
MS_GEOXFM_NO_LEVELFEATURES = 1

Inferred Cell Features (MS_GEOXFM_NO_CELLNAMEFEATURES)

For elements inferred as features using cell names, the feature's class name is set to the cell name. In addition:

- If the level name contains invalid characters, they are converted to underscores.
- If the cell name begins with a number then it is prefixed with *Cell_* , for example *Cell_4Pole.*
- Orphan cells with cell elements on multiple levels have their feature class name prefixed with *Cell_* followed by the Level ID.
- Orphan cells with all elements on the same level have feature class name of that level name. If the level starts with a number, it will be prefixed with *Level_* . For example, *Level_4Pole*

To override this behavior, set MS_GEOXFM_NO_CELLNAMEFEATURES = 1

Example – DFS Rules for Element Symbology

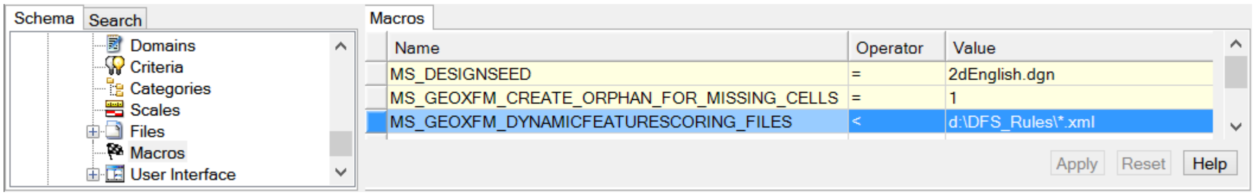
The following can be used as a simple exercise to illustrate using DFS rules to evaluate an elements colour, weight and style. If they meet the required minimum score of 100, the element will be inferred as a Cable.

Note that the elements level is not considered, and the values for the colour, weight, and style are set to 5, 3, and 2 respectively.

To create this DFS rule file, simply create a new text file in any folder and add the following text:

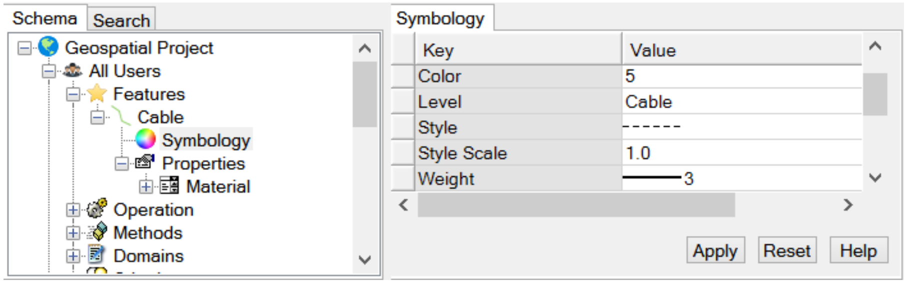
```
<FeatureScoringRules minScore="100">
        <Feature useCriteria="" name="Cable">
                <!-- color, weight, style -->
        <Rule score="100" type="cexpr">XFMCompareElementSymbology(5, 3, 2)</Rule>
        </Feature>
</FeatureScoringRules>
```

Using the Geospatial Administrator, create a new project and define a workspace. In the Macro node of the workspace definition, set the variable MS_GEOXFM_DYNAMICFEATURESCORING_FILES=1 to the location of the DFS rule file created above.



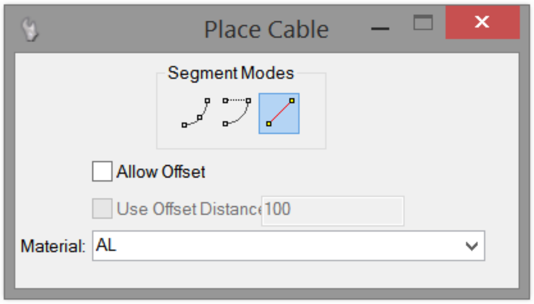
Name	Operator	Value
MS_DESIGNSEED	=	2dEnglish.dgn
MS_GEOXFM_CREATE_ORPHAN_FOR_MISSING_CELLS	=	1
MS_GEOXFM_DYNAMICFEATURESCORING_FILES	<	d:\DFS_Rules\*.xml

Create a simple linear element for a Cable and add a property or two. Set the symbology as noted in the graphic below. Update the Command Manager then save and export the workspace and related metadata.



Key	Value
Color	5
Level	Cable
Style	------
Style Scale	1.0
Weight	3

To test the DFS rule, run the workspace, create a new design file and draw a Cable feature. As expected, the result is a *Native Feature:Cable.*

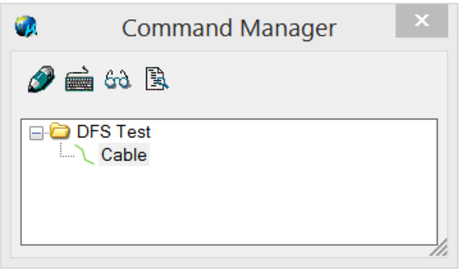

Place Cable

Segment Modes

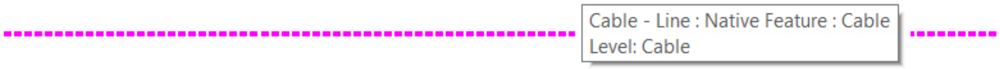
☐ Allow Offset

☐ Use Offset Distanc 100

Material: AL

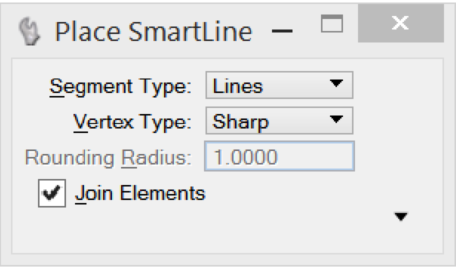

Command Manager

DFS Test
Cable


Cable - Line : Native Feature : Cable
Level: Cable

As well, create a new level called Highway and draw a line with any colour, weight and style.

Since this new element does not meet the minimum score as defined in the DFS rule, it's identified as an *Inferred Feature: Highway*, and is added as a Session feature in the Command Manager.

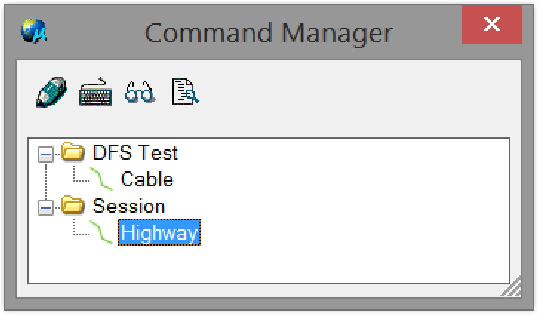

Place SmartLine

Segment Type: Lines

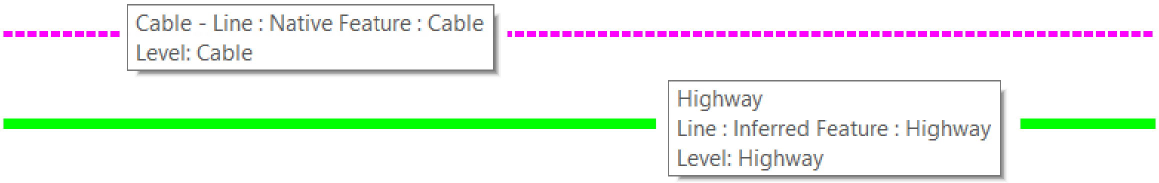
Vertex Type: Sharp

Rounding Radius: 1.0000

☑ Join Elements

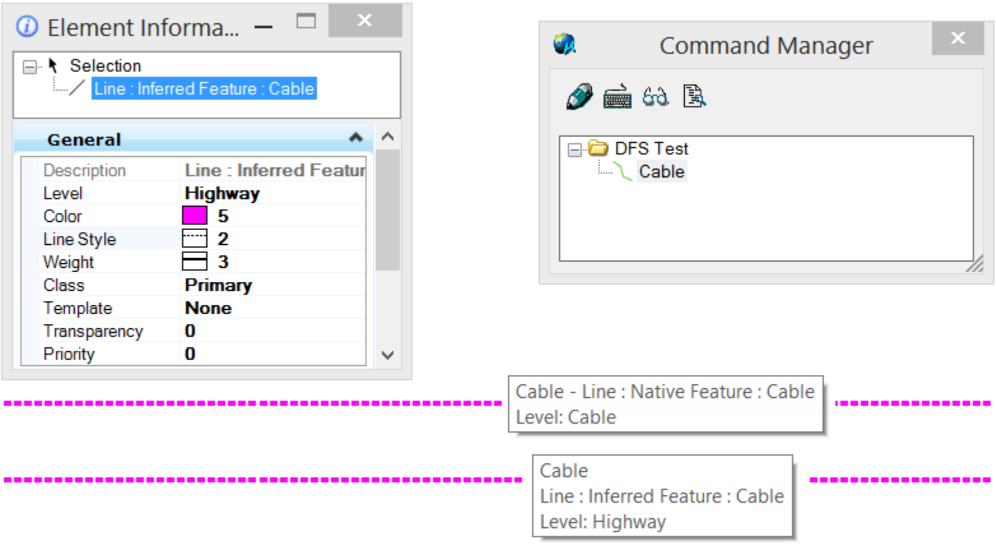

Command Manager

DFS Test
Cable
Session
Highway


Cable - Line : Native Feature : Cable
Level: Cable

Highway
Line : Inferred Feature : Highway
Level: Highway

The next step is to verify that the DFS rule is working correctly. To do that, change the symbology of the Highway to match the Cable (CO=5, LC=2 and WT=3). Bentley Map now recognizes the element to be a *Inferred Feature:Cable* which indicates that it met the minimum score of 100 in the DFS rule. As well, the Highway session feature was removed from the Command Manager.



Selection
Line : Inferred Feature : Cable

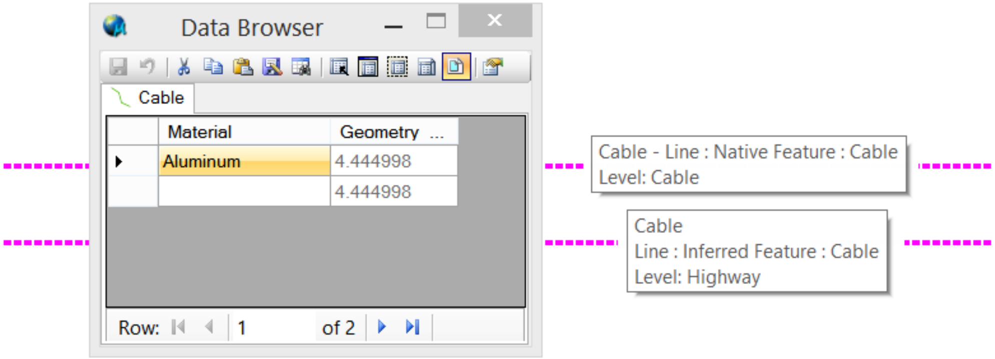
Description	Line : Inferred Featur
Level	Highway
Color	5
Line Style	2
Weight	3
Class	Primary
Template	None
Transparency	0
Priority	0

DFS Test
Cable

Cable - Line : Native Feature : Cable
Level: Cable

Cable
Line : Inferred Feature : Cable
Level: Highway

The Data Browser shows both the native and inferred cable features, however the inferred cable does not have property information and it resides on the incorrect level.

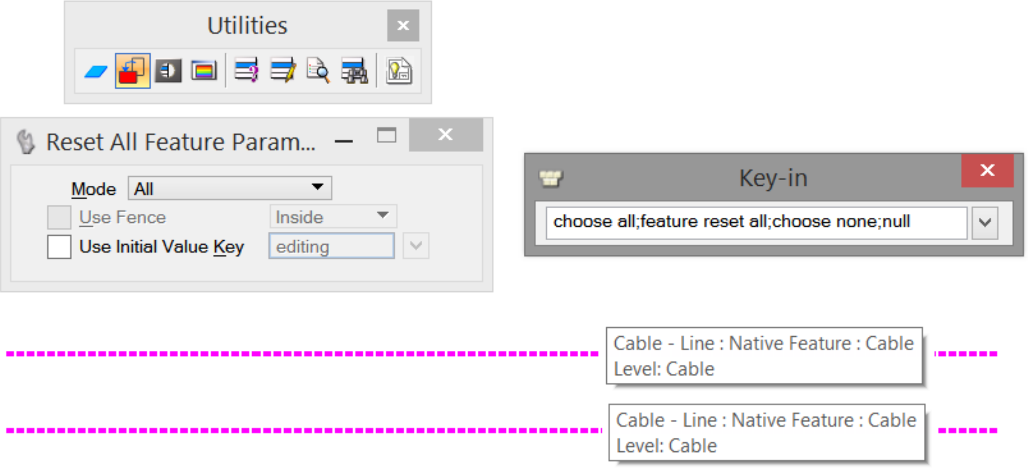


	Material	Geometry ...
▸	Aluminum	4.444998
		4.444998

Cable - Line : Native Feature : Cable
Level: Cable

Cable
Line : Inferred Feature : Cable
Level: Highway

To correct this, use *Reset Feature* from the Geospatial Tools | Utilities toolbox, or issue the following key in: *choose all;feature reset all;choose none;null* to reset all feature properties to the values defined in the geospatial project.



Mode: All

☐ Use Fence: Inside

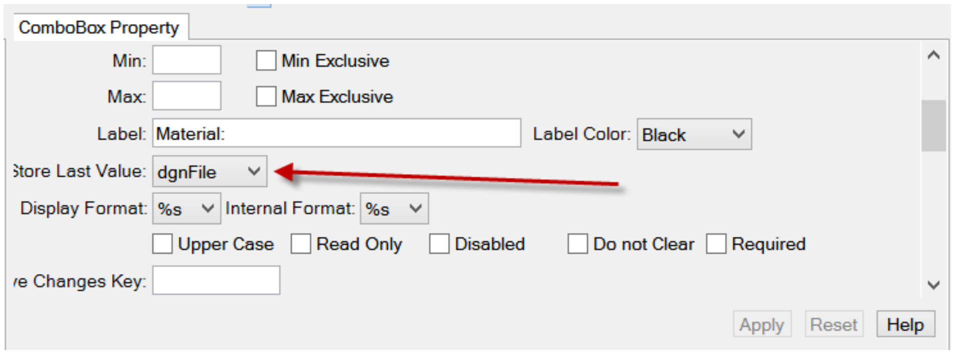
☐ Use Initial Value Key: editing

```
choose all;feature reset all;choose none;null
```

Cable - Line : Native Feature : Cable
Level: Cable

Cable - Line : Native Feature : Cable
Level: Cable

As for the property values, they will inherit the setting as defined in the geospatial project. In this example, the Material property is set to store the last value used in the design file.



Min: ☐ Min Exclusive

Max: ☐ Max Exclusive

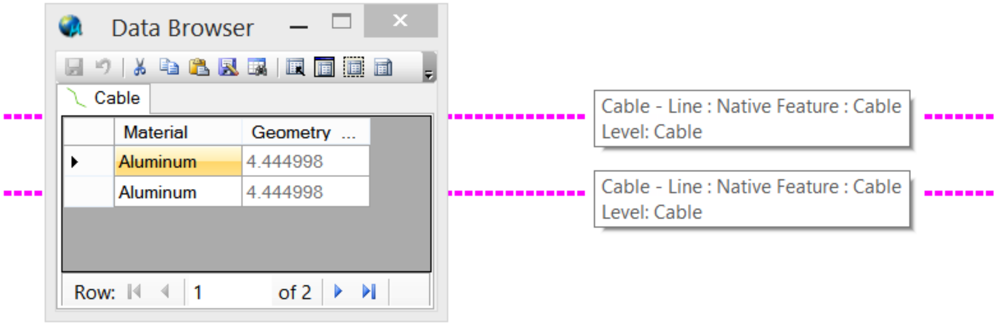
Label: Material: Label Color: Black

Store Last Value: dgnFile

Display Format: %s Internal Format: %s

☐ Upper Case ☐ Read Only ☐ Disabled ☐ Do not Clear ☐ Required

ve Changes Key:



	Material	Geometry ...
▸	Aluminum	4.444998
	Aluminum	4.444998

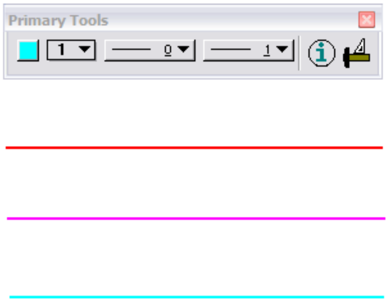
Cable - Line : Native Feature : Cable
Level: Cable

Cable - Line : Native Feature : Cable
Level: Cable

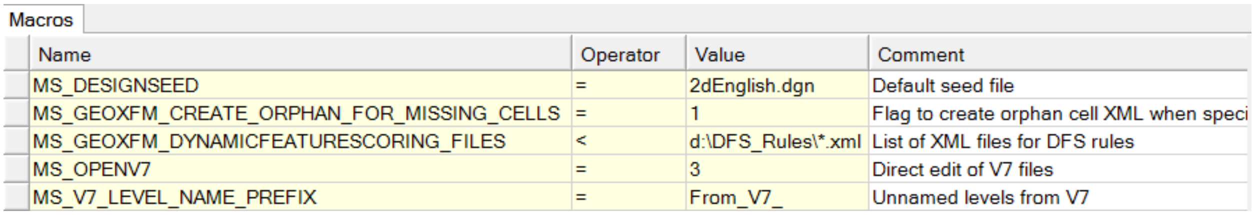
For more options on the *Reset Feature* tool, please refer to Bentley Map help.

Example – DFS Rules to evaluate V7 elements

DFS rules can also be used to evaluate and score elements in a V7 design file. For example, this design file has elements on Level 1 using colours 3, 5 and 7.



In the Macros section of the workspace node, two variables are added:
MS_OPENV7=3 to allow editing of V7 files, and
*MS_V7_LEVEL_NAME_PREFIX=From_V7_* to define the prefix for unnamed V7 levels.



Macros

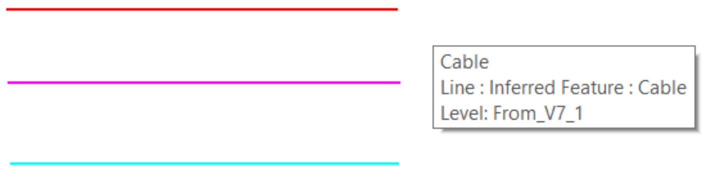
Name	Operator	Value	Comment
MS_DESIGNSEED	=	2dEnglish.dgn	Default seed file
MS_GEOXFM_CREATE_ORPHAN_FOR_MISSING_CELLS	=	1	Flag to create orphan cell XML when speci
MS_GEOXFM_DYNAMICFEATURESCORING_FILES	<	d:\DFS_Rules\*.xml	List of XML files for DFS rules
MS_OPENV7	=	3	Direct edit of V7 files
MS_V7_LEVEL_NAME_PREFIX	=	From_V7_	Unnamed levels from V7

A DFS rule file can be created to evaluate the symbology and level name of an element. In this example, the rule file has three rules for evaluating the element symbology and one for the level name. In order to qualify as a Cable feature, an element must score a minimum of 100 points where 50 points comes from the symbology section and the remaining 50 comes from the level name.

Since the elements are on Level 1 in the originating V7 design file, they will be displayed as *From_V7_1* when opened in Bentley Map. Note that this is new level name is specified in the level rule.

```
<FeatureScoringRules minScore="100">
        <Feature name="Cable" useCriteria="">
                <!-- Colour, Weight, Style -->
                <Rule score="50" type="cexpr">XFMCompareElementSymbology(3, 1, 0)</Rule>
                <Rule score="50" type="cexpr">XFMCompareElementSymbology(5, 1, 0)</Rule>
                <Rule score="50" type="cexpr">XFMCompareElementSymbology(7, 1, 0)</Rule>
                <!-- Level Name -->
                <Rule score="50" type="cexpr">XFMCompareElementLevelName("From_V7_1")</Rule>
        </Feature>
</FeatureScoringRules>
```

Upon opening, the DFS rules are used to evaluate the elements in the V7 design file and reports them as being *Inferred Features:Cable* . As with the previous example, these elements can be converted to Native Features by using Reset Feature or this key in: *choose all;feature reset all;choose none;null*



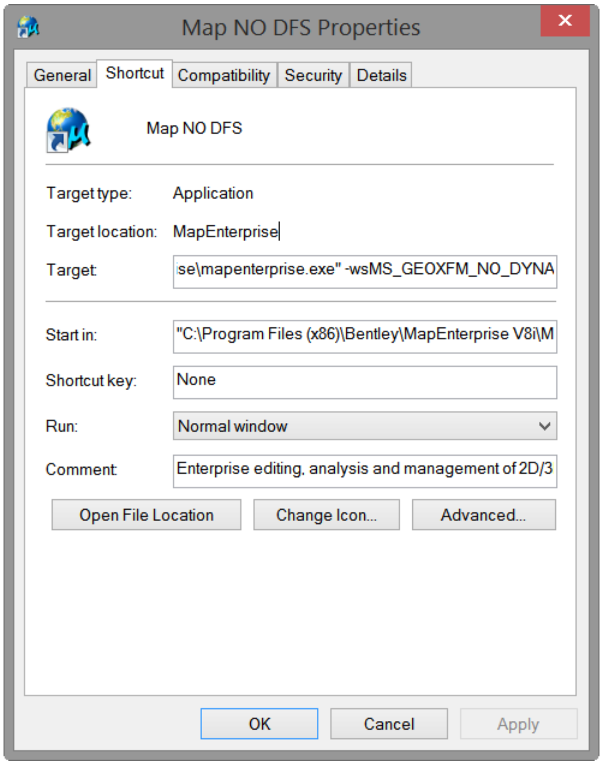
Cable
Line : Inferred Feature : Cable
Level: From_V7_1

DFS Side Effects

When DFS is enabled, it evaluates all elements which includes those of attached references. Depending upon the volume and type of data, this may cause the opening and attaching design files to be slower than expected. This can cause unwanted performance issues that negatively impact your project.

Should DFS be causing unwanted performance issues, you may want to consider disabling the DFS engine by defining MS_GEOXFM_NO_DYNAMICSCOREDFEATURES.

This variable can be set in the project schema, the UCF, PCF, any CFG file, or even on the shortcut with the *–ws* switch:



Map NO DFS

Target type: Application

Target location: MapEnterprise

Target: se\mapenterprise.exe" -wsMS_GEOXFM_NO_DYNA

Start in: "C:\Program Files (x86)\Bentley\MapEnterprise V8i\M

Shortcut key: None

Run: Normal window

Comment: Enterprise editing, analysis and management of 2D/3

Appendix - DFS Configuration Variables

MS_GEOXFM_DYNAMICFEATURESCORING_FILES - Specifies a list of XML files containing DFS rules for processing. For example:
MS_GEOXFM_DYNAMICFEATURESCORING_FILES >
$(MS_GEOXFM_XML_DIR)features/dynamicScoring/.xml*

MS_GEOXFM_NO_DYNAMICSCOREDFEATURES – Defining this variable specifies that the DFS engine is off and will not featurize elements. For example:
MS_GEOXFM_NO_DYNAMICSCOREDFEATURES = 1

MS_GEOXFM_NO_LEVELFEATURES - Specifies that the DFS engine should not identify non-cell elements as features with feature name equal to the element's level name if no DFS rule matches the element. The default behavior is that elements are identified as level features if no DFS rule matches the element. Example: *MS_GEOXFM_NO_LEVELFEATURES = 1*

MS_GEOXFM_NO_CELLNAMEFEATURES - Specifies that the DFS engine should not identify cell elements as features with feature name equal to the element's cell name if no DFS rule matches the element. The default behavior is that cell elements are identified as cell name features if no DFS rule matches the element. Example:
MS_GEOXFM_NO_CELLNAMEFEATURES = 1

Appendix – DFS Rule Types

Cexpr - CExpression rule. Any CExpression containing functions published with mdlCExpression_publishFunction may be used. If no rule type is specified, then this rule type is the default. The following XFM published functions are available for use with CExpression rules evaluation.

Function	Parameters
XFMHasDBLink	int entityNumber, UInt32 mslink
XFMHasDBLinkEntity	int entityNumber, UInt32 mslink
XFMCompareElementSymbology	UInt32 inputColor, UInt32 inputWeight, Int32 inputStyle
XFMCompareElementType	int type
XFMCompareElementLevelId	UInt32 inputLevel
XFMCompareElementLevelName	char* inputLevelName
XFMCompareElementLevelNameToDef	char* featureName
XFMHasXAttribute	UInt16 major, UInt16 minor
XFMHasXAttributeId	UInt16 major, UInt16 minor, UInt32 id
XFMCompareElementCellName	char* inputCellName
XFMCompareElementColor	UInt32 inputColor
XFMCompareElementWeight	UInt32 inputWeight
XFMCompareElementStyle	Int32 inputStyle
XFMCompareElementStyleName	char* pStyleName

DBLink - Database linkage rule. Current element is checked for a MicroStation database linkage with the specified entity and mslink values.

DBEntity - Database entity linkage rule. Current element is checked for a MicroStation database linkage with the specified entity value.

FeatureDef - Feature definition rule. Current element is compared to the XFM feature definition of the given feature class name using the following comparison types:

Type Name	Description
levelName	true or false - If true, element level name must match a level name specified in feature definition. If the Feature definition contains a list of level names then if element level name matches one of the levels in the list then evaluates to true.
cellName	true or false - If true, element cell name must match a cell name specified in feature definition. If the Feature definition contains a list of cell names then if element cell name matches one of the cell names in the list then evaluates to true.

Element - Element rule. Current element is compared to specified element settings using the following comparison types:

<table>
<tr><th>Type Name</th><th>Description</th></tr>
<tr><td>levelName</td><td>name of a level - If specified, element level name must match the specified level name</td></tr>
<tr><td>cellName</td><td>name of a cell - If specified, element cell name must match the specified cell name</td></tr>
<tr><td>elementType</td><td>one or more element types - If specified, element type must match one of the specified element types. A zero length linear is identified as element type 0. This identification may be disabled by setting attribute "checkZeroLengthLineElementType" to "false".
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>3</td><td>element type 3</td></tr>
<tr><td>3-4</td><td>element type 3 or 4</td></tr>
<tr><td>3-4,12</td><td>element type 3, 4 or 12</td></tr>
<tr><td>1-12</td><td>element types 1 through 12</td></tr>
<tr><td>0</td><td>linear element type with two equal points (zero length line)</td></tr>
</table>
</td></tr>
<tr><td>checkZeroLengthLineElementType</td><td>Flag denoting if linear element is checked to determine if element is a zero length line. If a zero length line is identified, then the element type identifier is zero. The default value for this attribute is "true".</td></tr>
</table>

Appendix – Sample DFS Rules

```
<FeatureScoringRules minScore="100">
        <Feature useCriteria="" name="tax_proposed">
                <Rule type="cexpr"
        score="100">XFMHasDBLink(2,150)||XFMCompareElementLevelName("tax proposed")</Rule>
        </Feature>
        <Feature useCriteria="" name="Pole">
                <Rule type="DBLink" score="100" entity="2" mslink="130"/>
        </Feature>
        <Feature useCriteria="" name="Wire">
                <Rule type="DBLink" score="100" entity="2" mslink="130"/>
                <Rule type="FeatureDef" score="100" levelName="true" cellName="false">Wire</Rule>
        </Feature>
        <Feature useCriteria="" name="Cable">
                <Rule type="DBLink" score="100" entity="2" mslink="80"/>
                <Rule type="Element" score="100" levelName="Cable" cellName="Cable"
        elementType="2" checkZeroLengthLineElementType="false"/>
        </Feature>
</FeatureScoringRules>
```