Contents

Executive Summary **1**

1 Overview **3**

2 Using Parameters and Property Sets **7**

3 Installation and Usage **10**

4 JSON Schema **12**

5 Known Issues and Limitations **16**

6 Next Steps **20**

Document Control **21**

List of figures

Figure 1: Export process overview 3
Figure 2: Form to select the working folder 4
Figure 3: Processing model message box 4
Figure 4: Message box that informs the user of the completion of the operation 4
Figure 5: IFC Parameters on Revit objects 5
Figure 6: Civil 3D property sets converted to Revit shared parameters 5
Figure 7: Civil 3D references and common IFC Shared Parameters in Revit 6
Figure 8: Assigning a name to a filter to recall it in the views 7
Figure 9: Create a rule-based filter in Revit 7
Figure 10: Tags can be used to display the shared parameter values 8
Figure 11: In Revit it is possible to calculate totals based on Civil 3D parameters 9
Figure 12: The external tools loading should be authorized once 10
Figure 13: Access the Revit command 11
Figure 14: A null structure in Revit 17
Figure 15: Curved pipes 3D geometries comparison in elevation 18
Figure 16: Curved pipes 2D geometries comparison in plan 18

List of tables

Table 1: JSON Schema 12

1 Overview

This document covers the process to use the custom tools developed by Autodesk Consulting to export Civil 3D pipe and pressure pipe networks to Revit. The intention is to create a Revit model that contains the same information as the Civil 3D model for pipe networks (geometry and metadata). The tools are based on a common library (PipeNetworks.dll) that contains the functionalities called by the Civil 3D and Revit custom commands.

The common library can be used to eventually develop a custom Dynamo package that accesses the same functionalities.

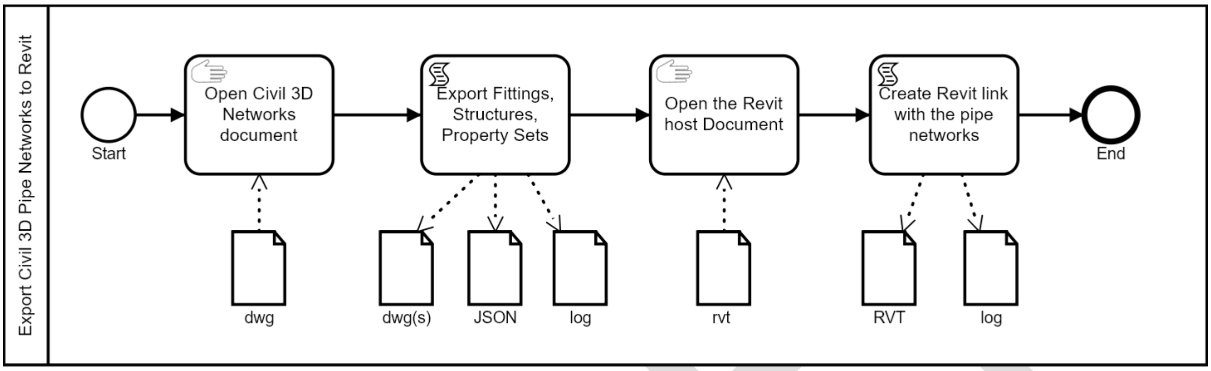

Export Civil 3D Pipe Networks to Revit
Start
Open Civil 3D Networks document
Export Fittings, Structures, Property Sets
Open the Revit host Document
Create Revit link with the pipe networks
End
dwg
dwg(s)
JSON
log
rvt
RVT
log

Figure 1: Export process overview

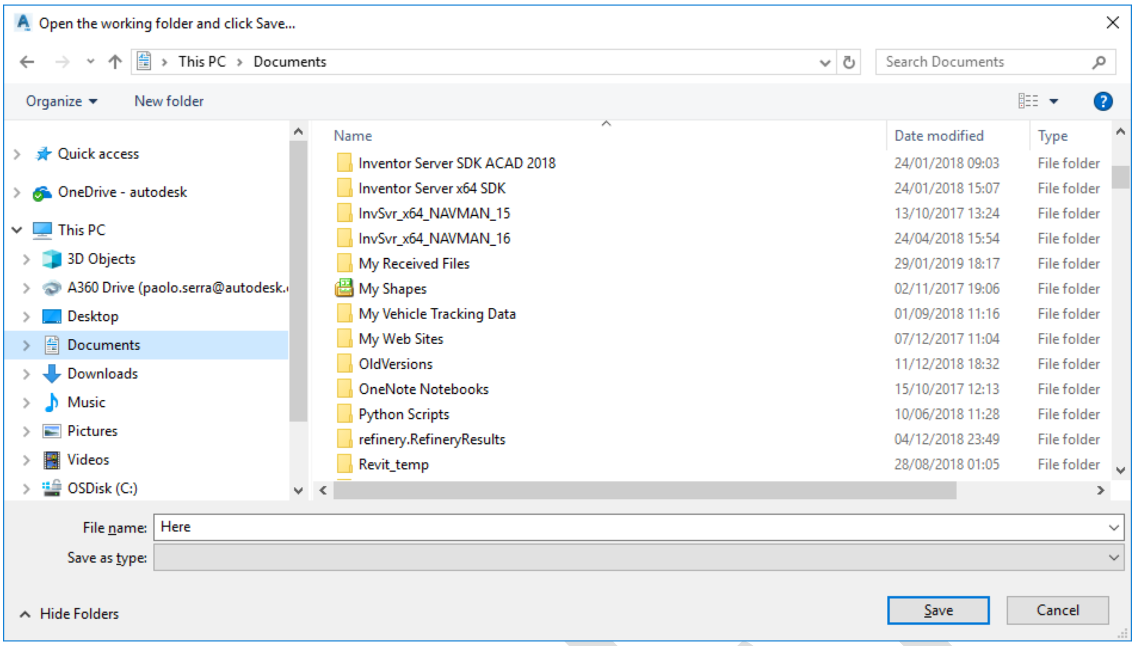
Here is the high-level workflow to follow:

1. Open the Civil 3D model that contains the network(s); this works also for short cut models.
2. Orient the model to an Isometric view making sure the pipe networks elements are all visible.
3. Run the Civil 3D command *EPN* to export the JSON file and extract Fittings, the Ancillaries and Structures from the model and wait for completion.
4. Eventually check the *<Civil3DFileName>_DumpPipeAndPressurePipeNetworks.log* and *<Civil3DFileName>_StructuresFittingsExport.log* in the temp folder to see if there were any errors.
5. Open the Architectural Revit model (or the host Revit model with the Shared Coordinates setup).
6. Run the Revit external command *Import Networks* to create import the pipe networks and wait for the completion.
7. Eventually check the *<RevitFileName>_Civil3DtoRevitPipeNetworks.log* in the temp folder to see if there were any errors.

The Civil 3D command reads all the pipe and pressure pipe networks in the Civil 3D model and writes a JSON file that contains the relevant data (geometry and property sets) to be transferred to Revit.

Note the Civil 3D pipe network objects behave in many aspects as Multiview block references: they have different 2D and 3D definitions. For this process it is necessary to orient the view to an isometric view so that only the 3D definition will be taken into consideration. If the view orientation is not suitable, the user will receive a message.

Note the user is asked to open a working folder in which to collect the structures and fittings solid representations. This is to enable the workflow even with files that are controlled via a Data Management system (e.g. Vault, ProjectWise, etc.). The user can choose a local folder instead of dealing with the deep folder structures imposed by the Data Management system.



Open the working folder and click Save...

Name	Date modified	Type
Inventor Server SDK ACAD 2018	24/01/2018 09:03	File folder
Inventor Server x64 SDK	24/01/2018 15:07	File folder
InvSvr_x64_NAVMAN_15	13/10/2017 13:24	File folder
InvSvr_x64_NAVMAN_16	24/04/2018 15:54	File folder
My Received Files	29/01/2019 18:17	File folder
My Shapes	02/11/2017 19:06	File folder
My Vehicle Tracking Data	01/09/2018 11:16	File folder
My Web Sites	07/12/2017 11:04	File folder
OldVersions	11/12/2018 18:32	File folder
OneNote Notebooks	15/10/2017 12:13	File folder
Python Scripts	10/06/2018 11:28	File folder
refinery.RefineryResults	04/12/2018 23:49	File folder
Revit_temp	28/08/2018 01:05	File folder

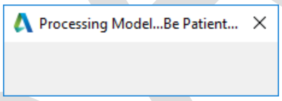
File name: Here

Save as type:

Figure 2: Form to select the working folder

The geometry of fittings, ancillaries and structures is stored in the working folder which can be a different location from the Civil 3D model. For each network it creates a folder that contains either a Structures folder (if it is a pipe network) or a Fittings folder (if it is a pressure pipe network).

The extraction operation can take quite some time due to the size of the model, the user is prompt a small window for the duration of the extraction. At the end of this process, the user will be informed via a message box.



Processing Model...Be Patient...

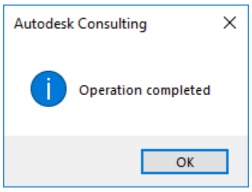
Figure 3: Processing model message box

The Revit command creates a new document named after the Civil 3D model with a ".RVT" extensions (all capitals) to distinguish it from a user created Revit file. The new document will have the same shared coordinates, orientation and system units as the host file used to launch the command.

Before running the command, the user needs to make sure that there are no Revit link instances with the same name already loaded. If that is the case, the user needs to remove the link.

The user is asked to select the DWG file of the Civil 3D model that contains the networks. This will tell Revit where to look for the JSON file that contains the data and for the working folder that contains the DWGs extracted by the previous command in Civil 3D.

The creation operation can take quite some time due to the size of the model, the user will see a small window for the duration of the operation. At the end of this process, the user will be informed via a message box.



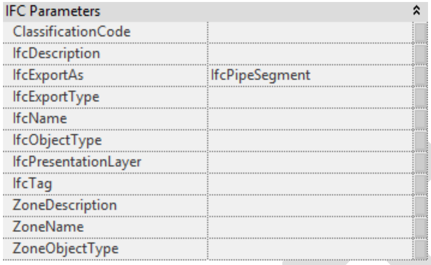
Autodesk Consulting

Operation completed

Figure 4: Message box that informs the user of the completion of the operation

The information in this document is to be considered Confidential to Autodesk, Inc., and cannot be reproduced or redistributed without the prior written consent except internally for purposes of review and consideration of this Proposal / Report.

The command creates shared parameters for the Revit IFC Export plugin (e.g. IfcExportAs, IfcName, etc.) under the **IFC Parameters** group; these parameters are bound to multiple categories. The command creates also a PipeNetwork shared parameter under the **Data** group to store the name of the pipe network; this is bound to the Generic Models category only. The GUIDs of the IFC parameters and the PipeNetwork parameter are hardcoded in the command and they do not change regardless the project source or destination. This enables the safe creation of schedules and tags based on these parameters.



IFC Parameters	
ClassificationCode	
IfcDescription	
IfcExportAs	IfcPipeSegment
IfcExportType	
IfcName	
IfcObjectType	
IfcPresentationLayer	
IfcTag	
ZoneDescription	
ZoneName	
ZoneObjectType	

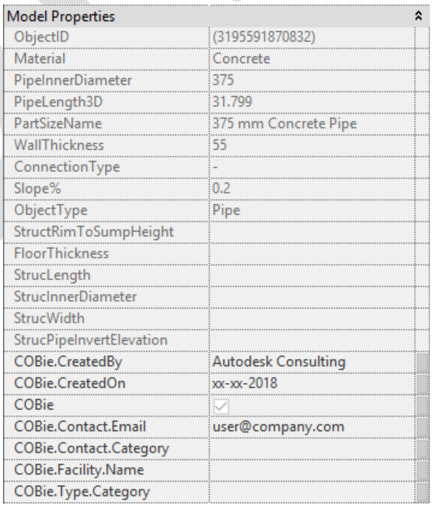
Figure 5: IFC Parameters on Revit objects

The command also creates a shared parameters file dedicated to the Civil 3D property sets definitions contained in the JSON file. These shared parameters are bound to the Generic Models category only and grouped under **Model Properties**. The name of the parameters will contain the name of the property set between brackets to avoid duplicates and to show the origin of the shared parameter once in Revit. The shared parameter file is saved with the Revit document and named after it:

<RevitDocumentName>_SharedParameters.txt

The command checks first if the parameter exist in the shared parameter file for the Revit document, otherwise it creates one. During the first run, the command creates parameter groups in the shared parameters file named as the Civil 3D Property sets. If the property set name is new it creates a new definition group otherwise it uses an existing one. The command checks if among the parameter definitions under the definition group there is already a parameter with the same name and type and if it is visibile and modifiable to the end user. In case of a perfect match the command uses the parameter definition in the shared parameter file, otherwise it creates a new parameter definition with the appropriate characteristics.

Do not delete the shared parameter file to preserve the parameter GUIDs to enable consistent scheduling and tagging through out the life cycle of a project.



Model Properties	
ObjectID	(3195591870832)
Material	Concrete
PipeInnerDiameter	375
PipeLength3D	31.799
PartSizeName	375 mm Concrete Pipe
WallThickness	55
ConnectionType	-
Slope%	0.2
ObjectType	Pipe
StructRimToSumpHeight	
FloorThickness	
StrucLength	
StrucInnerDiameter	
StrucWidth	
StrucPipeInvertElevation	
COBie.CreatedBy	Autodesk Consulting
COBie.CreatedOn	xx-xx-2018
COBie	☑
COBie.Contact.Email	user@company.com
COBie.Contact.Category	
COBie.Facility.Name	
COBie.Type.Category	

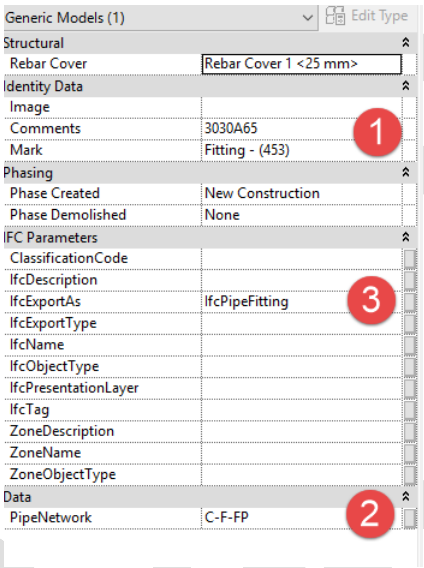
Figure 6: Civil 3D property sets converted to Revit shared parameters

The command scans the JSON file and creates for each network the pipe geometries first and then the fittings and ancillaries (for pressure pipe networks) and the structures (for pipe networks).

Once completed it assigns the values to the shared parameters from the JSON file. It also assigns some default values to the IfcExportAs parameter:

- *IfcPipeSegment* for pipes
- *IfcPipeFitting* for fittings
- *IfcFlowController* for ancillaries
- *IfcDistributionChamberElement* for structures

The handle and the obejct name are stored in the Comments and the Mark instance parameters (1), the Pipe Network parameter contains the network name (2). It is possible to change the target IFC class for an export (3).



Generic Models (1)

Structural	
Rebar Cover	Rebar Cover 1 <25 mm>
Identity Data	
Image	
Comments	3030A65
Mark	Fitting - (453)
Phasing	
Phase Created	New Construction
Phase Demolished	None
IFC Parameters	
ClassificationCode	
IfcDescription	
IfcExportAs	IfcPipeFitting
IfcExportType	
IfcName	
IfcObjectType	
IfcPresentationLayer	
IfcTag	
ZoneDescription	
ZoneName	
ZoneObjectType	
Data	
PipeNetwork	C-F-FP

Figure 7: Civil 3D references and common IFC Shared Parameters in Revit

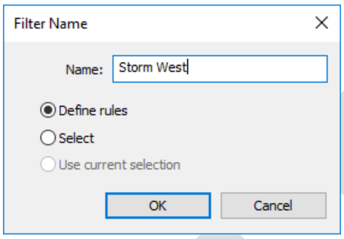
Once completed the Revit document containing the networks is loaded as a link in the Revit host document.

When the model needs to be updated, repeat the procedure starting form Civil 3D and before running the Import Network command in Revit, remember to remove the previous Revit link. The Revit link will be overwritten so it is good practice to save a copy for submission and archive.

2 Using Parameters and Property Sets

In Revit it is possible to create selection filters based on element properties or directly selecting the elements in the model. The *PipeNetwork* parameter can be used to group the elements that belong to the same network and, for example, apply color overrides to projection and cut surfaces or to hide elements in views.

To create a named, rule-based selection, **Manage > Selection > Edit > New…,** assign a name (e.g. the name of a network) and in the following form select Generic Models for the category.



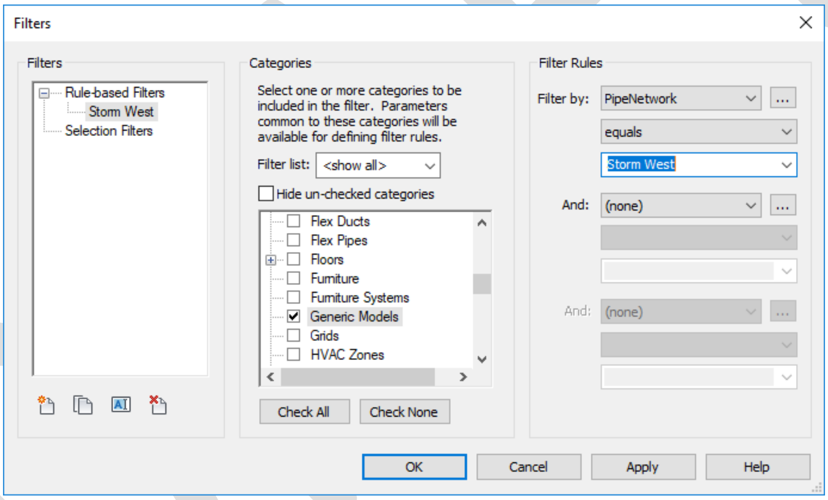
Filter Name

Name: Storm West

● Define rules
○ Select
○ Use current selection

Figure 8: Assigning a name to a filter to recall it in the views

Under *Filter Rules*, set the *Filter by* to a parameter of the selected category (e.g. PipeNetwork), specify *equals* in the comparison drop-down and add the value of the parameter to be used to filter the selection (e.g. the network name).



Filters

Filters
- Rule-based Filters
  - Storm West
- Selection Filters

Categories

Select one or more categories to be included in the filter. Parameters common to these categories will be available for defining filter rules.

Filter list: <show all>

□ Hide un-checked categories

□ Flex Ducts
□ Flex Pipes
□ Floors
□ Furniture
□ Furniture Systems
☑ Generic Models
□ Grids
□ HVAC Zones

Filter Rules

Filter by: PipeNetwork
equals
Storm West

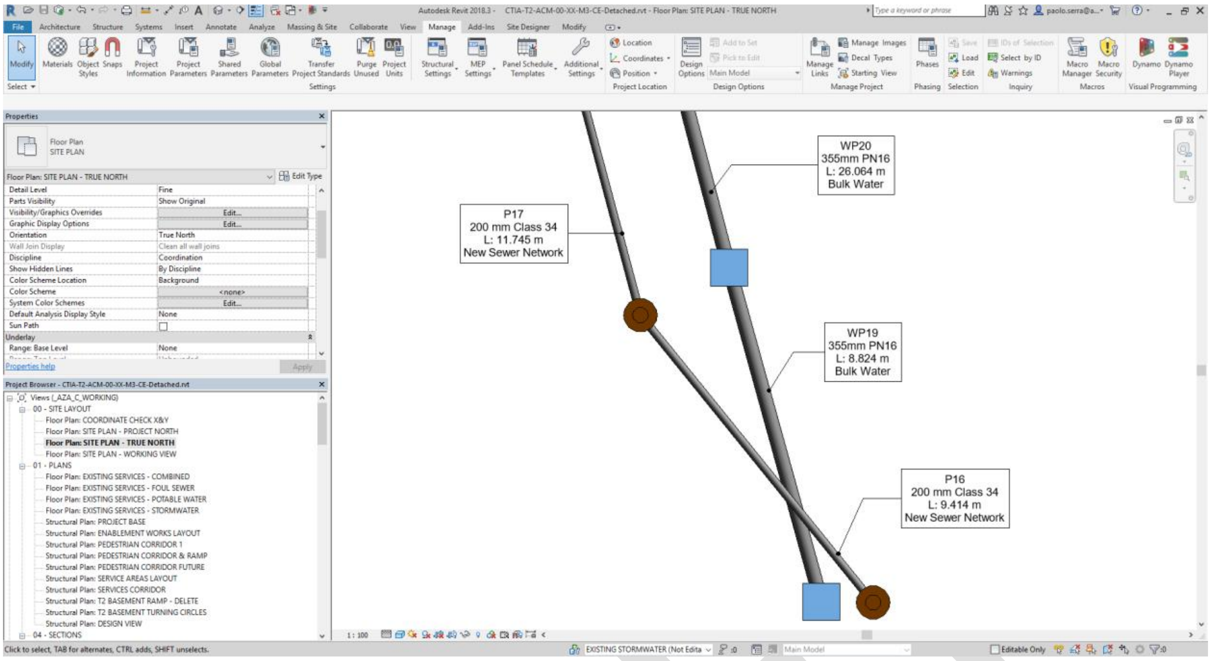
And: (none)

And: (none)

Figure 9: Create a rule-based filter in Revit

The named selection will be available for view filters (Rule-based Filters). The end user has the flexibility to toggle the visibility or change the representation of the objects via a view filter, defined by the data attached to the object.

It is possible to transfer the project standards from the pipe networks model to the architectural model to incorporate the shared parameters. It is also possible extract the shared parameters from the Revit link and use them to create tags to display the parameter values.



Autodesk Revit 2018.3 - CTIA-T2-ACM-00-XX-M3-CE-Detached.rvt - Floor Plan: SITE PLAN - TRUE NORTH

Properties

Floor Plan
SITE PLAN

Floor Plan: SITE PLAN - TRUE NORTH

Detail Level	Fine
Parts Visibility	Show Original
Visibility/Graphics Overrides	Edit...
Graphic Display Options	Edit...
Orientation	True North
Wall Join Display	Clean all wall joins
Discipline	Coordination
Show Hidden Lines	By Discipline
Color Scheme Location	Background
Color Scheme	<none>
System Color Schemes	Edit...
Default Analysis Display Style	None
Sun Path	□
Underlay	
Range: Base Level	None

Project Browser - CTIA-T2-ACM-00-XX-M3-CE-Detached.rvt

- Views (_AZA_C_WORKING)
  - 00 - SITE LAYOUT
    - Floor Plan: COORDINATE CHECK X&Y
    - Floor Plan: SITE PLAN - PROJECT NORTH
    - **Floor Plan: SITE PLAN - TRUE NORTH**
    - Floor Plan: SITE PLAN - WORKING VIEW
  - 01 - PLANS
    - Floor Plan: EXISTING SERVICES - COMBINED
    - Floor Plan: EXISTING SERVICES - FOUL SEWER
    - Floor Plan: EXISTING SERVICES - POTABLE WATER
    - Floor Plan: EXISTING SERVICES - STORMWATER
    - Structural Plan: PROJECT BASE
    - Structural Plan: ENABLEMENT WORKS LAYOUT
    - Structural Plan: PEDESTRIAN CORRIDOR 1
    - Structural Plan: PEDESTRIAN CORRIDOR & RAMP
    - Structural Plan: PEDESTRIAN CORRIDOR FUTURE
    - Structural Plan: SERVICE AREAS LAYOUT
    - Structural Plan: SERVICES CORRIDOR
    - Structural Plan: T2 BASEMENT RAMP - DELETE
    - Structural Plan: T2 BASEMENT TURNING CIRCLES
    - Structural Plan: DESIGN VIEW
  - 04 - SECTIONS

Click to select, TAB for alternates, CTRL adds, SHIFT unselects.

Figure 10: Tags can be used to display the shared parameter values

Civil 3D property sets can be customized using formulas, this gives the user the possibility to capture calculated values form Civil 3D to other platforms such as Revit and Navisworks. The JSON file also contains the information that is available on the Civil 3D objects that can be consumed in Revit via automation (e.g. Dynamo).

The following formula returns the Part Size Name.

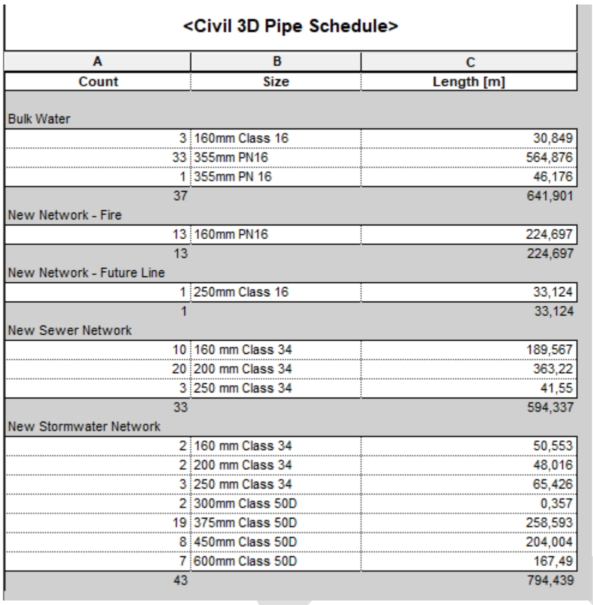
```
RESULT = "--"
On ERROR RESUME NEXT
Set app = GetObject(, "AutoCAD.Application")
Set civApp = app.GetInterfaceObject("AeccXUILand.AeccApplication.12.0")
Set obj = civApp.ActiveDocument.ObjectIdToObject("[ObjectID]")
RESULT = obj.PartDataRecord.FindByContext(204).Tag
```

The following formula returns the Length 3D of a pipe.

```
RESULT = 0
On ERROR RESUME NEXT
Set app = GetObject(, "AutoCAD.Application")
Set civApp = app.GetInterfaceObject("AeccXUILand.AeccApplication.12.0")
Set obj = civApp.ActiveDocument.ObjectIdToObject("[ObjectID]")
RESULT = Round(obj.Length3D, 3)
```

However, the result of a formula is a text therefore the JSON dump will capture this attribute as Text.

The user can manually change the DataType attribute of a property definition from *Text* to *Real* or *Integer* so that the Import Network command will convert the value respectively to *Number* or to *Integer*. This will enable the sum and calculation of totals in Revit.



<Civil 3D Pipe Schedule>

A	B	C
Count	Size	Length [m]
Bulk Water		
3	160mm Class 16	30,849
33	355mm PN16	564,876
1	355mm PN 16	46,176
37		641,901
New Network - Fire		
13	160mm PN16	224,697
13		224,697
New Network - Future Line		
1	250mm Class 16	33,124
1		33,124
New Sewer Network		
10	160 mm Class 34	189,567
20	200 mm Class 34	363,22
3	250 mm Class 34	41,55
33		594,337
New Stormwater Network		
2	160 mm Class 34	50,553
2	200 mm Class 34	48,016
3	250 mm Class 34	65,426
2	300mm Class 50D	0,357
19	375mm Class 50D	258,593
8	450mm Class 50D	204,004
7	600mm Class 50D	167,49
43		794,439

Figure 11: In Revit it is possible to calculate totals based on Civil 3D parameters

3 Installation and Usage

This section of the document covers how to install the tools for Civil 3D 2018 and Revit 2018.

For the tools to work the user should have administrator rights on the machine or at least it should be possible to write in the following folders:

- *C:\ProgramData\Autodesk\ApplicationPlugins\*
- *C:\ProgramData\Autodesk\Revit\Addins\2018\*

The user can manually create the following folder structure copying the provided files as follows.

Civil 3D

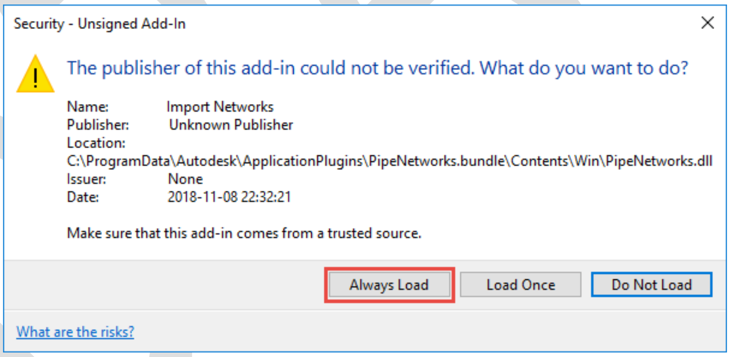
- *C:\ProgramData\Autodesk\ApplicationPlugins\PipeNetworks.bundle\PackageContents.xml*
- *C:\ProgramData\Autodesk\ApplicationPlugins\PipeNetworks.bundle\Contents\Win\ PipeNetworks.dll*
- *C:\ProgramData\Autodesk\ApplicationPlugins\PipeNetworks.bundle\Contents\Win\ C3dCmdExportPipeNetworks.dll*

Revit

- *C:\ProgramData\Autodesk\ApplicationPlugins\PipeNetworks.bundle\Contents\Win\ RvtCmdImportPipeNetworks.dll*
- *C:\ProgramData\Autodesk\Revit\Addins\2018\RvtCmdImportPipeNetworks.addin*

In alternative, assuming the user has the necessary writing rights on the machine, launch the *InstallPipeNetworks.exe* to automatically create the folder structure and copy the provided files.

When launching Civil 3D and Revit the first time after the installation answer “Always Load” to the Security form that appears.



Security - Unsigned Add-In

The publisher of this add-in could not be verified. What do you want to do?

Name: Import Networks
Publisher: Unknown Publisher
Location:
C:\ProgramData\Autodesk\ApplicationPlugins\PipeNetworks.bundle\Contents\Win\PipeNetworks.dll
Issuer: None
Date: 2018-11-08 22:32:21

Make sure that this add-in comes from a trusted source.

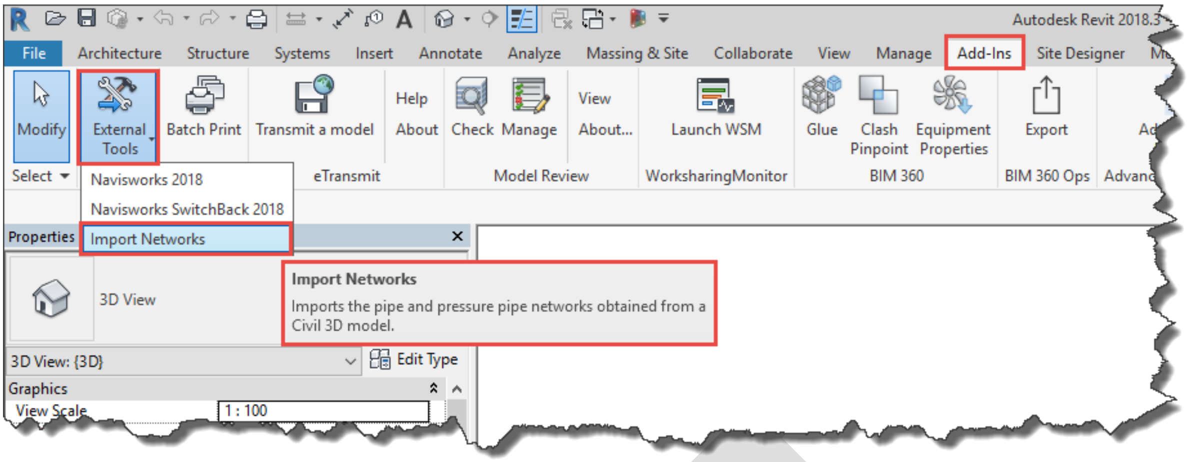
Figure 12: The external tools loading should be authorized once

Civil 3D Command

At the command line type **EPN**.

Revit Command

Navigate to the Add-Ins tab, click on the External Tools drop down and select the **Import Networks** command.



Import Networks

Imports the pipe and pressure pipe networks obtained from a Civil 3D model.

Figure 13: Access the Revit command

4 JSON Schema

This section of the document covers the JSON file used to transfer data from Civil 3D to Revit.

This file is used to collect the relevant objects used during the conversion. An object in this schema has a *Type* attribute that specifies the nature of the object and other attributes that describe the status of the object.

For pipes and structures, some of the attributes are immediately available as properties of the objects and they have bene included in the schema to enrich the export. For pressure pipes and ancillaries have been implemented methods that are not immediately available (e.g. *MidPoint*).

Table 1: JSON Schema

Types	Attributes	Value Types	Notes
JDocumentPropertySets	User	String	The user name creating the dump
	Date	String	The date and time of the dump
	WorkingFolder	String	The folder that contains the result of the Structure and Fittings dump
	Document	String	The path of the Civil 3D model
	Units	String	The units of the Civil 3D model
	PropertySetDefinitions	Collection	
	CivilObjects	Collection	
	PressurePipeNetworks	Collection	
	PipeNetworks	Collection	
JPropertySetDefinition	Name	String	Name of the property set definition
	Description	String	Additional description
	AppliesToAll	Boolean	
	AppliesToFilter	Collection	List of Civil 3D object type names
	Definitions	Collection	
JPropertyDefinition	Name	String	Property Definition Name
	Description	String	Property Definition Description
	DataType	String	Supported types: Text, Double, Boolean
	Automatic	Boolean	If true, the shared parameter in Revit will be read-only
JCivilObject	Name	String	An object with at least a property set associated
	Handle	String	The handle of the object
	PropertySets	Collection	
JPropertySet	Name	String	The name of the instance of the Property Set
	Handle	String	The handle of the property set instance
	AppliesTo	Collection / String	List of the Civil 3D object type names or “All”
	PropertySetDefinition	String	The name of the property set definition associated to the property set instance

Types	Attributes	Value Types	Notes
	Properties	Collection	
JProperty	Name	String	The property instance name, it
	Value	String, Boolean, Double	
JPressurePipeNetwork	Name	String	The name of the pressure pipe network
	Handle	String	The handle of the pressure pipe network
	Pipes	Collection	
	Fittings	Collection	
	Ancillaries	Collection	
JPressurePipe	StartPoint	JPoint	
	EndPoint	JPoint	
	MidPoint	JPoint	Null if the pipe is linear
	EndFitting	String	The handle of the end fitting or null
	StartFitting	String	The handle of the start fitting or null
	Shape	String	Only “Circular”
	InnerDiameter	Double	
	OuterDiameter	Double	
	EndElevation	Double	
	StartElevation	Double	
	WallThickness	Double	
	IsLinear	Boolean	False if the pressure pipe is curved, otherwise true
	Name	String	The name of the pressure pipe
	Network	String	The name of the network the pressure pipe belongs
	Description	String	The description of the pressure pipe
	Handle	String	The handle of the pressure pipe
JPressureFitting	StartPoint	JPoint	
	EndPoint	JPoint	
	Name	String	The name of the fitting
	Network	String	The name of the network the fitting belongs
	Description	String	The description of the fitting
	Handle	String	The handle of the fitting
JPressureAncillary	StartPoint	JPoint	
	EndPoint	JPoint	
	Name	String	The name of the ancillary
	Network	String	The name of the network the ancillary belongs
	Description	String	The description of the ancillary
	Handle	String	The handle of the ancillary

Types	Attributes	Value Types	Notes
JPipeNetwork	Name	String	The name of the pipe network
	Handle	String	The handle of the pipe network
	Pipes	Collection	
	Structures	Collection	
JPipe	StartPoint	JPoint	
	EndPoint	JPoint	
	MidPoint	JPoint	Null if the pipe is linear
	StartAngle	Double	0 if the pipe is linear
	EndAngle	Double	0 if the pipe is linear
	Center	JPoint	Null if the pipe is linear
	IsLinear	Boolean	False if the pipe is curved, otherwise true
	Shape	String	“Circular”, “Rectangular”, “EggShaped”, “Elliptical”, “HorizontalElliptical”
	CoverOfEndpoint	Double	
	CoverOfStartPoint	Double	
	EndOffset	Double	
	EndStation	Double	0 if the pipe is not related to an alignment
	EndStructure	String	The handle of the end structure or null
	InnerDiameterOrWidth	Double	
	InnerHeight	Double	
	Material	String	
	MaximumCover	Double	
	MinimumCover	Double	
	OuterDiameterOrWidth	Double	
	OuterHeight	Double	
	PartDescription	String	
	PartSizeName	String	
	PartSubType	String	
	PartType	String	
	Radius	Double	0 if the pipe is linear
	Slope	Double	
	StartOffset	Double	
	StartStation	Double	
	StartStructure	String	
	WallThickness	Double	
	Name	String	The name of the pipe
	Network	String	The name of the network the pipe belongs
	Description	String	The description of the pipe
	Handle	String	The handle of the pipe
JStructure	StartPoint	JPoint	

Types	Attributes	Value Types	Notes
	EndPoint	JPoint	
	BarrelPipeClearance	Double	
	Shape	String	
	ConeHeight	Double	
	Cover	Double	
	OuterDiameterOrWidth	Double	
	FloorThickness	Double	
	Frame	String	
	FrameDiameter	Double	
	FrameHeight	Double	
	Grate	String	
	Height	Double	
	InnerDiameterOrWidth	Double	
	InnerLength	Double	
	Length	Double	
	Material	String	
	Offset	Double	
	PipeLowestBottomDepth	Double	
	PipeUpperTopDepth	Double	
	PartDescription	String	
	PartSizeName	String	
	PartSubType	String	
	PartType	String	
	RimElevation	Double	
	RimToSumpHeight	Double	
	Rotation	Double	
	SumpElevation	Double	
	Station	Double	0 if the structure is not related to an alignment
	SumpDepth	Double	
	WallThickness	Double	
	Name	String	The name of the structure
	Network	String	The name of the network the structure belongs
	Description	String	The description of the structure
	Handle	String	The handle of the structure
JPoint	X	Double	
	Y	Double	
	Z	Double	

5 Known Issues and Limitations

This section of the document covers the known issues and limitations emerged during the workflow analysis and development of the tools.

Civil 3D Property Sets and Revit Shared Parameters

The property sets in Civil 3D and the shared parameters in Revit are very different.

- In Civil 3D objects of the same classification can have a different set of property sets associated or even no property sets at all; in Revit, all the objects of a category will have the same property sets associated.
- The property sets in Civil 3D can have user defined names and they are displayed as groups of parameters; in Revit the parameter groups are defined internally, and they cannot be changed.
- Civil 3D property sets support enumerated lists, Revit does not yet offer this feature.

Revit Categories

In this implementation all the pipe network objects from Civil 3D are converted to Generic Models in Revit. The decision to use only Generic Models was made because other categories in Revit do not offer the same flexibility in terms of representation (projection and cut). This means that if there are two different property sets defined, one for pipes and one for structures, their properties will appear mixed all together on Revit Generic Models. During the export towards IFC only the compiled parameters will end up in the output.

Civil 3D Pipes and Revit Pipes

Revit has a native pipe category and pipe systems to create networks, but they offer a different kind of objects and features from Civil 3D pipes and networks.

- Civil 3D pipes can be curved, in Revit they are only straight (or flexible but not exactly curved).
- Civil 3D pipes can have different cross sections: circular, rectangular, egg shaped, elliptical and horizontal elliptical; in Revit pipes are only circular.
- The structure objects in a pipe network in Civil 3D can connect more than four pipe objects at different heights; Revit fittings can connect to a maximum of four pipes, but the centerline of the pipes should meet at the center of the fitting and this limits their usability in gravity networks context.
- Using other objects with a custom number of connectors makes it very difficult to make any adjustments to the networks as any attempt to modify the network manually can break the connections and raise a Revit warning after which it will be unpractical to continue editing the network.

Civil 3D Catalogs and Revit Families

Civil 3D handles the parametric behavior of the pipe network items using catalogs and block references. A catalog is a SQLite database that collects the possible dimensions for a given object type (e.g. fittings or structures).

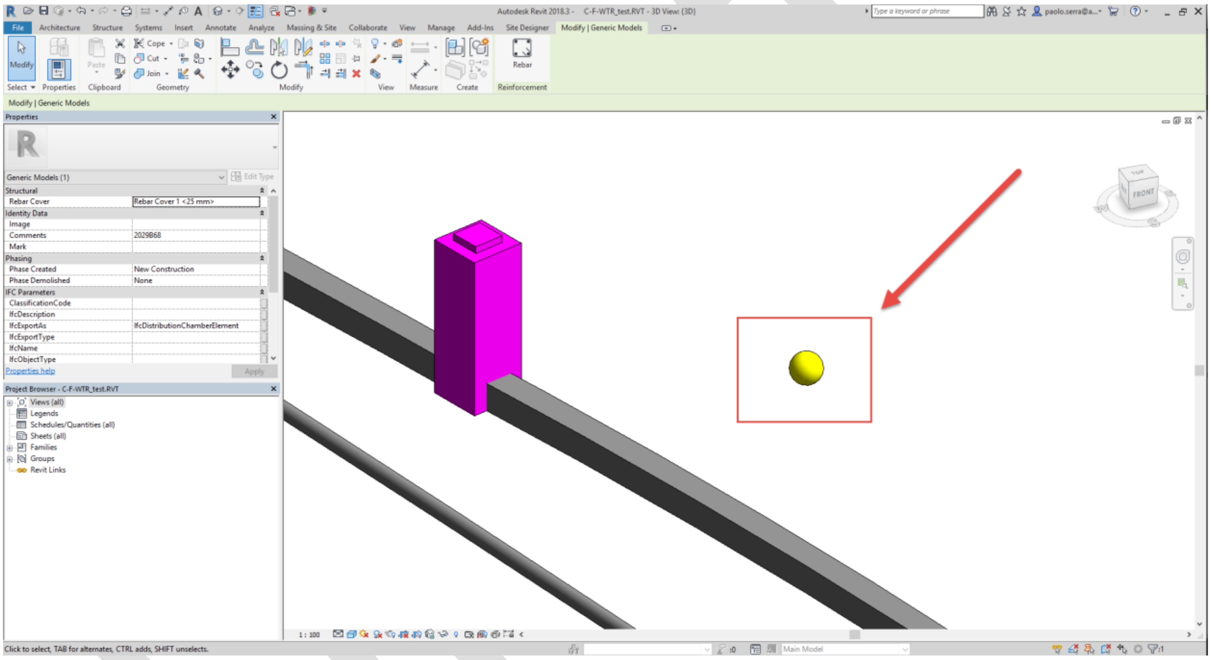
The items are loaded in a Civil 3D style and then the user can insert them in the model. These items are referring to underlying CAD block references that are adjusted and rotated by the application (e.g. the fittings) or that are modified at insertion to consider references to other elements such as a ground surface (e.g. the structures).

Revit handles the parametric objects via families that contain the geometric information and the parameters that are necessary to enable the modification to the base geometry. This approach allows to have a few types of objects loaded in the model and then control each instance to suit the project needs. Creating this parametric content requires a good level of expertise with Revit parametric components to achieve the right representation for different views and with different level of detail.

Ideally, the Civil 3D catalogs and Revit families should be aligned and reflect the same content. This option was not considered for the implementation because it would have been time consuming and, in principle, it could change on a project basis.

Instead, each instance of fittings, ancillaries or structures solid is exported from Civil 3D as a separate DWG named after the object handle. The separate DWG contain only the solid of the original item, positioned, rotated and scaled accordingly to the original model, capturing the state of the model at the time of the export. The DWGs are then processed in Revit as DirectShape element types and assigned to the Generic Model category. This approach simplifies the overall complexity of having to align the Civil 3D catalogs and the Revit families on a project basis but has the downside of increasing the file size.

Currently, for null structures, the representation in Revit results as a spherical object: this can be changed in future development of the libraries for example excluding the null structures all at once.



Autodesk Revit 2018.3 - C-F-WTR_test.RVT - 3D View: {3D}

Modify | Generic Models

Properties

Generic Models (1)

Structural	
Rebar Cover	Rebar Cover 1 <25 mm>
Identity Data	
Image	
Comments	2029B68
Mark	
Phasing	
Phase Created	New Construction
Phase Demolished	None
IFC Parameters	
ClassificationCode	
IfcDescription	
IfcExportAs	IfcDistributionChamberElement
IfcExportType	
IfcName	
IfcObjectType	

Click to select, TAB for alternates, CTRL adds, SHIFT unselects.

Figure 14: A null structure in Revit

Curved Pipes

Civil 3D uses a cylindrical helix to generate the centre line of the curved pipes when the start and end points have a different elevation (or equally when the slope of the pipe is not zero). The cross-section profile is placed orthogonally to the centre line and is swept along the helix.

Revit uses planar curves to create swept solids (e.g. lines and arcs) and it is possible to create a cylindrical helix only for stairs or via API, but they cannot be used as a path for a swept solid. To create these pipes in Revit it was used a *Swept Blend* instead. The horizontal, planar path (a circular arc) is derived from the pipe in Civil 3D, the cross-section profiles needed are placed at the start and at the end of the arc. The profiles must be orthogonal to the path; therefore, they are contained in vertical planes. The profile at the end is translated on the global Z Axis by the difference between the end and start point elevations. The result is close to the Civil 3D pipe but not exactly the same in 3D. Conversely, Civil 3D projection in 2D of the curved pipe is the same as in Revit.

The information in this document is to be considered Confidential to Autodesk, Inc., and cannot be reproduced or redistributed without the prior written consent except internally for purposes of review and consideration of this Proposal / Report.

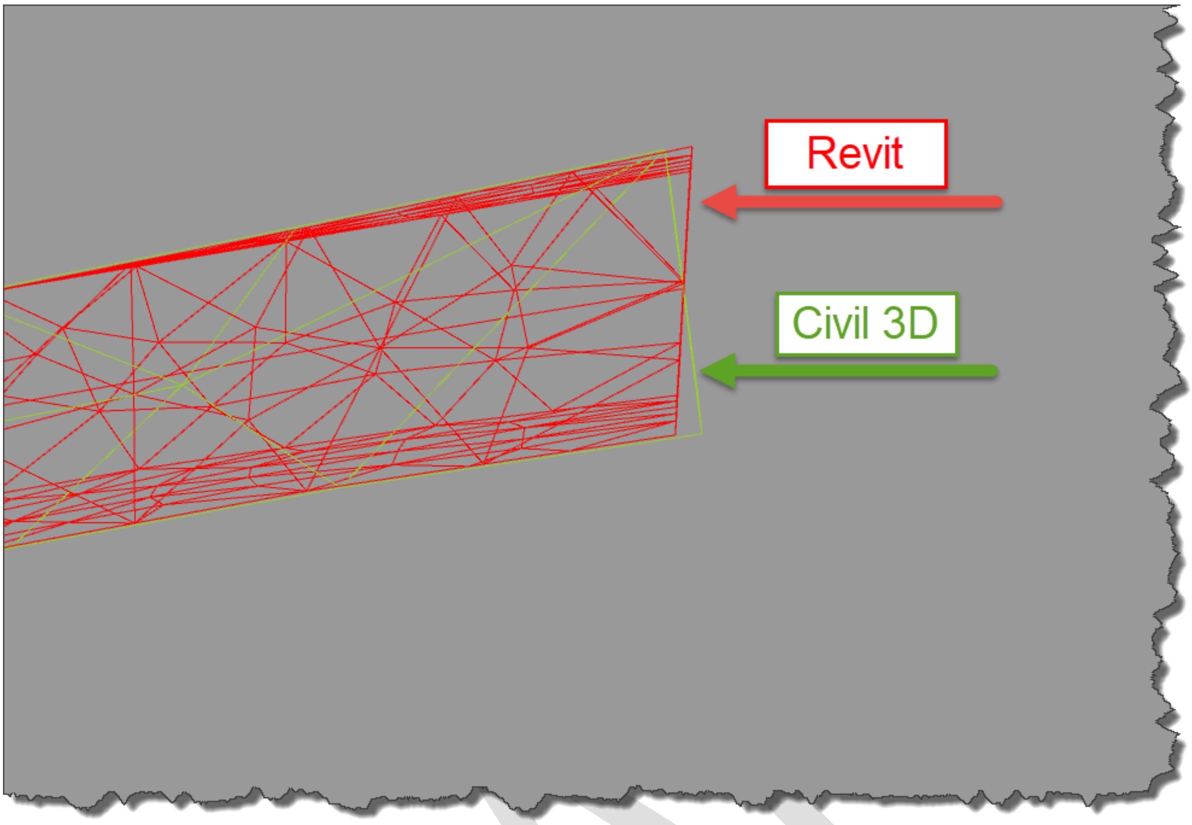
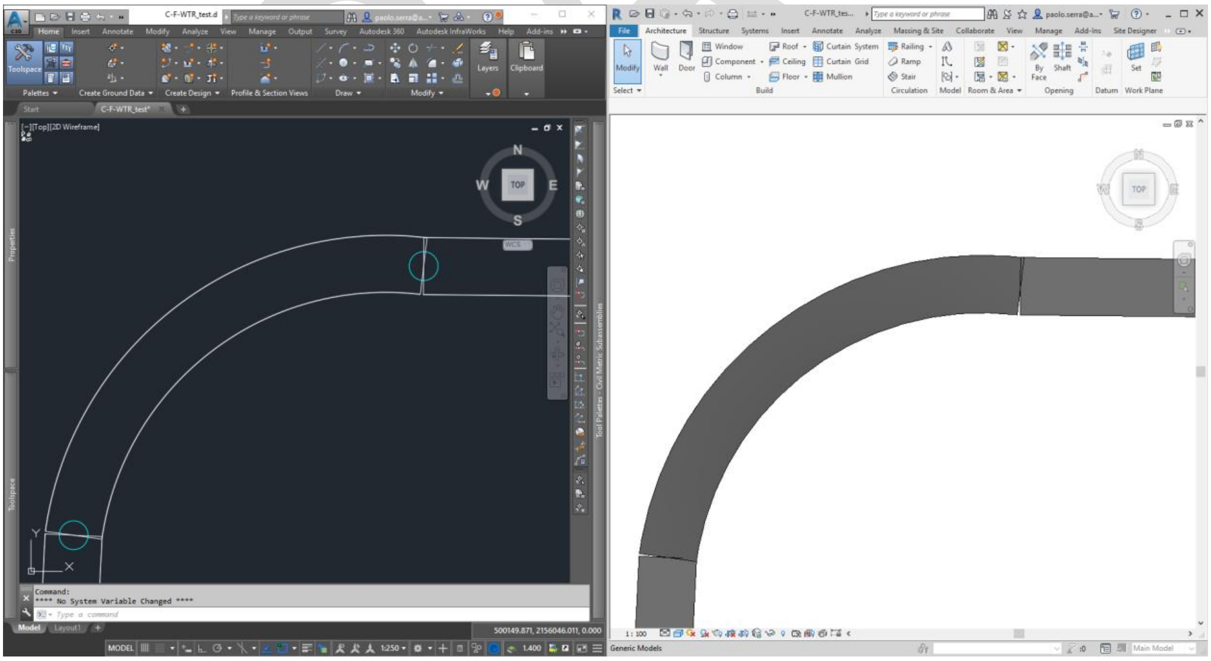


Figure 15: Curved pipes 3D geometries comparison in elevation



```
Command:
**** No System Variable Changed ****
```

Figure 16: Curved pipes 2D geometries comparison in plan

Known Issues

Following some known issues emerged during the development of the tools:

- If the Revit host document shared coordinates are not compatible with the Civil 3D model, the creation command can fail due to geometry limitations.
- During the development it was used a Civil 3D model that lost some of the target references, this caused for some of the pipes to have one of the ends at elevation 0 m. Using this dataset, it emerged that for some of those pipes the rotation of the cross-section profile may differ from the same pipes created in Revit.
- In Civil 3D it is possible that elements of different networks share the same name. The object names are used to populate the *Mark* parameters on the instances. In case of duplicates, Revit will show a warning to the end user, but this can be ignored.
- If the network objects in Civil 3D (particularly structures, fittings and ancillaries) have a style assigned that is hiding the geometry in 3D, these are most likely not going to present in the Revit model. These objects will still be correctly parsed in the JSON file. Looking at the log files can give a good indication if some objects were left out. The implementation is temporarily overriding the style assigned to the objects so that their 3D geometry is always visible.
- The current implementation does not allow to add to an existing model or to update a model but only to export the entire model anew. This would likely change *Elements* and *Guids* of the Revit elements. This could impact other BIM uses that are referring those internal values (e.g. 3D coordination clash tests in Navisworks, tags and dimensions for drawing production, etc.).

6 Next Steps

This section of the document lists some potential improvements to the current implementation.

- Create all the parameters from the property sets as read-only regardless if they are automatic or not to establish the information origin of the network objects to be in Civil 3D only.
- Excluding null structures from the creation process.
- Create and assign Revit materials that reflect the layer colors in Civil 3D to the pipe network objects.
- Select which pipe networks to export via User Interface rather than export all the model at once.
- Update existing models by network rather than overriding the entire model.
- Improving or changing the logging (file location, naming, exceptions classifications, etc.).

Document Control

Change history

Date	Version	Author	Description
10-Feb-19	1.2	Autodesk	Review

Distribution list

Name	Title	Organization

Approval history

Date	Version	Approver	Comments
Click here to enter a date.			

Document references

No.	Title	Version	Author

Abbreviations

Term	Description

Terminology

Term	Description

The information in this document is to be considered Confidential to Autodesk, Inc., and cannot be reproduced or redistributed without the prior written consent except internally for purposes of review and consideration of this Proposal / Report.

Blank End Page