

Using the Field Mapper

Contents

Overview	4
Field Mapper	5
Mapping Sets	6
Editing “Fields in file”	7
Updating fields from file	7
Updating a field using the Editor	8
Field Editor	8
Mapping a field to a property part or cut	9
Mapping Actions	9
Action Types	9
Copy field value	9
Use the default parser	9
Use the custom parser	9
Custom action	10
Available property action types	10
Setting a Mapping Action	11
Mapping Action Examples	12
Copy field value	12
Use the default parser	13
Use the custom parser	14
Use the default parser, then custom parser	15
Olet specific hole sizes	16
Mapping a field to multiple properties	17
About Regular Expressions	18
Cuts Templates	18
Configuring a cuts template custom action	20
Cuts Template Editor	21
Editor button groups	22

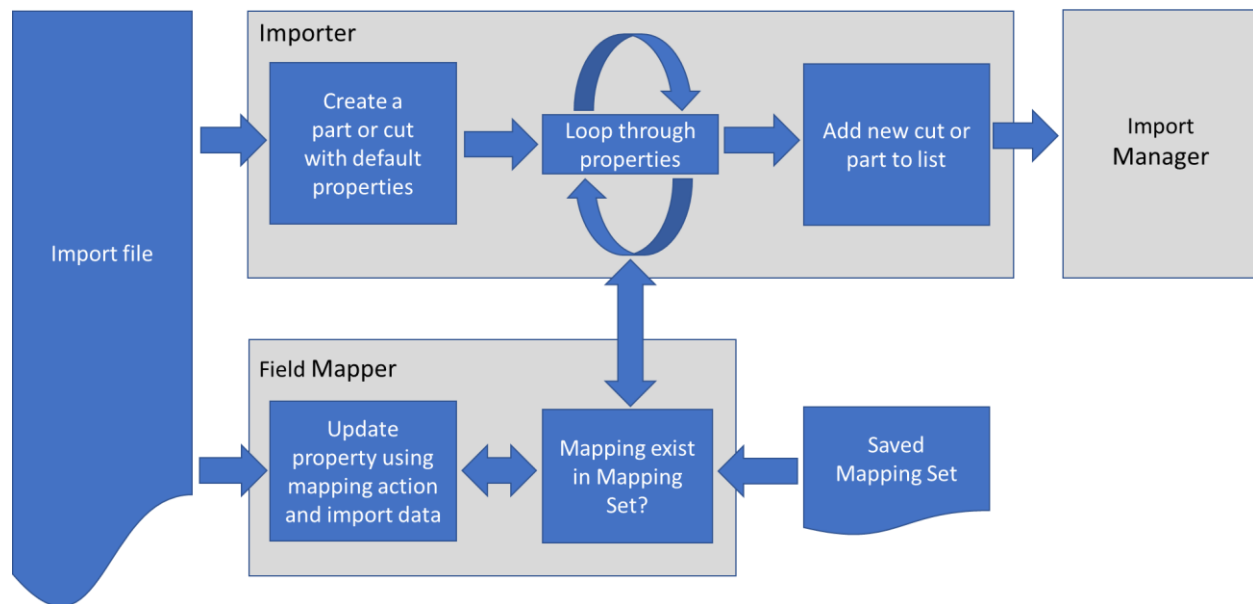
Editing cuts.....	22
Order of values applied during an import.....	22
Schema specific cut definitions.....	22
CSV schema	23
IFC schema	23
PCF schema	25
Appendix A: Importers supporting Field Mapping.....	29
CSV Importer	29
IFC Importer	29
PCF Importer	31
PypeServer Importer.....	33
Stratus Importer.....	33
SysQue Importer	34
Tekla Importer	34
Victaulic Importer	36
Appendix B: Field Mappings regular expressions	37
All supported importers.....	37
Outer Diameter	37
Schedule.....	37
Verify	38
Wall Thickness.....	38
CSV Importer	38
Default Cuts Template	38
IFC Importer	39
Alternate Cuts Template	39
Default Cuts Template	39
Hole Diameter	40
Miter End Cut.....	40
PCF Importer	40
Bevel Angle (deprecated).....	40
Hole Cut Template	41
Hole Diameter	41

Mate Pipe End Cut Template 42

Overview

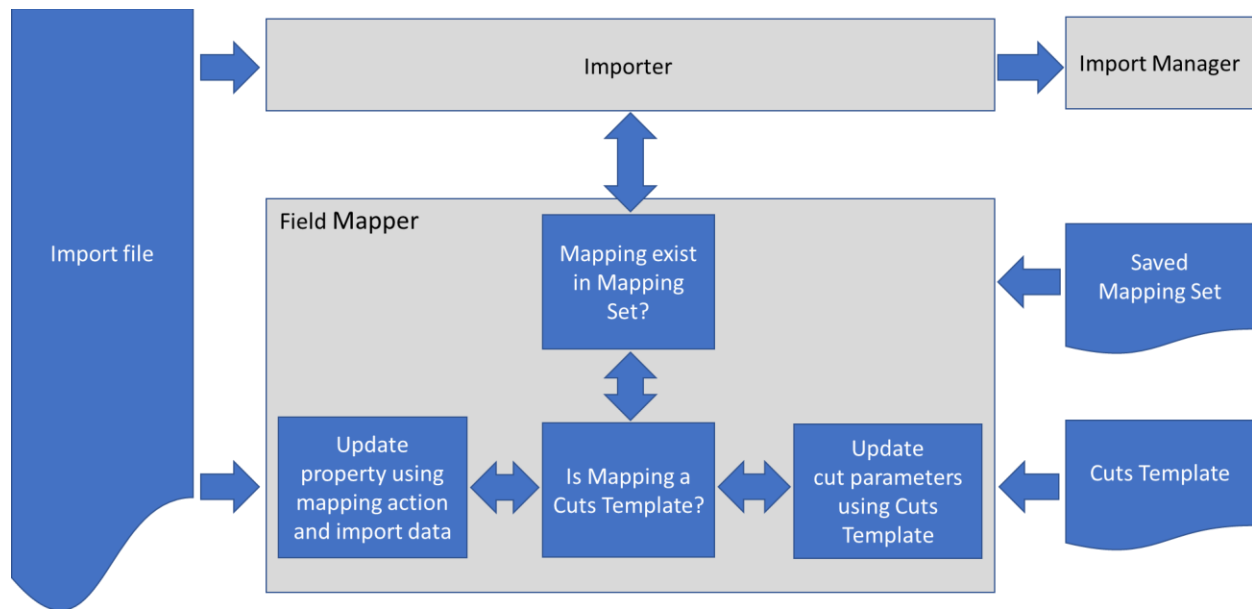
A Field Mapper enables a user to specify which values from the import file are to be copied to a part or cut during importing. It does this by keeping a “map” of available fields from the import file, and which of these fields will have their values copied to the part or cut during the import.

During import time, whenever a part or cut is created the Field Mapper will update the properties of that part or cut using the active field mappings that were set earlier. The update action can either copy the incoming data to the property, parse the incoming data using one or more regular expressions, or invoke a custom action.



Importer and Field Mapper

Custom actions perform a complex transformation of incoming data. A “Cuts Template” is an example of a custom action. A cuts template action applies a set of values to an imported cut using a special part which contains a set of template cuts. When a that part’s cut name and type match the mapped data then the values from that cut in the template are applied onto the imported cut.



Applying a cuts template or mapping

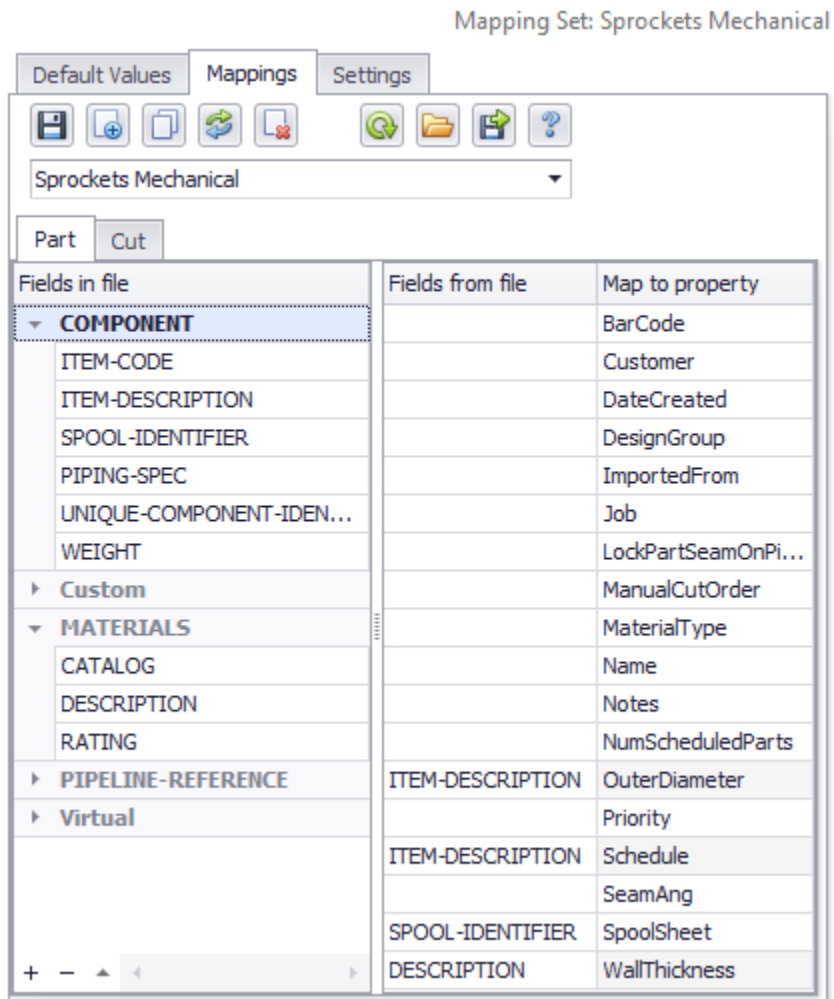
A list of “Fields in file” is the starting point for configuring Field Mapper and this list can be loaded from an import file or created manually. After import fields are added to the list they can be dragged over to the “Mapped to property” column where they are mapped to a property for a part or cut. A field name can be unmapped from the “Fields from file” column by dragging it back to the “Fields in file” column.

A configured set of mapped fields make up a “Mapping Set” which can be named and saved to build a list of mapping sets used for importing files from different sources and formatting specifications.

Only some Importers currently support the Field Mapper feature. When an Importer supports a Field Mapper it will appear as an additional “Mappings” tab in the lower left area of the Import Manager. See “Appendix A: Importer” for a list of Importers and technical details.

Field Mapper

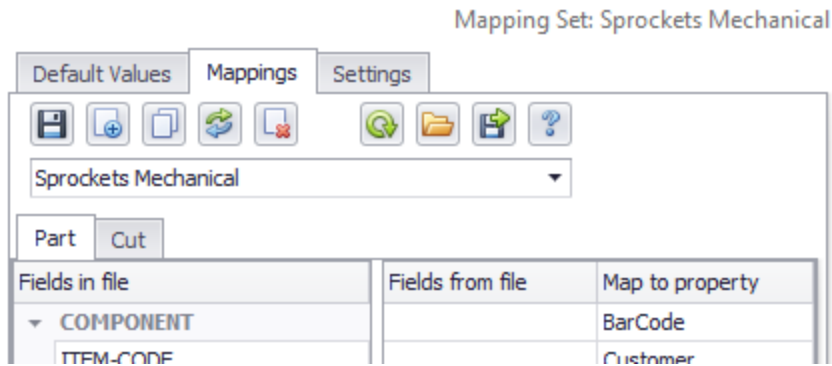
The “Mapping Sets” section at the top of the Field Mapper manages saving/restoring mapping sets. Below that is the “Mappings” section with tabs for import types “Part” and “Cut”, each containing a set of mappings with a list of “Fields in the file” on the left, and on the right a list of “Map to property” items where field mappings for each imported property can be configured.



Field Mapper area

Mapping Sets

A group of mapped fields comprise a “Mapping Set” which can be saved to build a list of mapping sets that can be reused with different sources and formatting specifications. “Mapping Sets” are managed using a dropdown list of mapping sets and a set of buttons for managing the currently selected set.



Mapping Set area

Mapping Set buttons - from left to right

- Save - Save the current mapping set
- Create – Create a new mapping set
- Copy – Copy the current mapping set to a new set
- Rename - Rename the current mapping set
- Delete - Delete the current mapping set
- Refresh - Refreshes the current set using data from the currently open import file.
 - New fields are added
 - Old fields are not deleted
 - Existing customized descriptions and sample data are not updated
- Import - Import a mapping set file
- Export - Export a mapping set file
- Help – Displays the Importer Field Mapper reference

Mapping Set dropdown list

- Selects a mapping set from a list of saved mapping sets for this importer

Editing “Fields in file”

A list of “Fields in file” is used for mapping the values from import data to part and cut properties during import. New fields can be added to a mapping set by either updating a mapping set from an open import file, or by creating items manually using the editor.

Edits are not “auto-saved” to allow easier testing, remember to save updates after editing and import testing.

Updating fields from file

To update a “Fields in file” list from an open import file;

- 1) Select a mapping set from the mapping set dropdown or create a new one.
- 2) Load some parts from an import file into the Import Manager.
- 3) Click on the “Refresh this Mapping Set” button.

- 4) Edit any new or existing fields if needed.
- 5) Click on the “Save” button to save your changes.

Updating a field using the Editor

At the bottom of the “Fields in file” column are buttons to manually create, delete and edit fields for the “Fields in file” list.



Editor Buttons

Editor buttons - from left to right

- 1) Add a new item to the “Field in file” column
- 2) Delete the currently selected field
- 3) Edit the currently selected field

Field Editor

The Field Editor is displayed by selecting a row from the “Fields in file” column and then double-click on the row or click the “Edit” button in the “Fields Editor” button set.

Field Editor

Field Editor controls

- Name – The name of the field as it appears in the import file
 - This is disabled for fields that are imported from files
- Description – User defined description of this field and usage
- Sample Data – Sample data that appears in the tooltip and Mapping Action Editor
 - This is auto-populated with data when the field is imported from a file and can be edited.

Mapping a field to a property part or cut

The Field Mapper has a tab for each of the import types; Part and Cut.

Part		Cut	
Fields in file		Fields from file	Map to property
▼ COMPONENT			BarCode
ITEM-CODE			Customer
ITEM-DESCRIPTION			DateCreated
SPOOL-IDENTIFIER			DesignGroup

Import types – Part/Cut

Associating a “Field in file” to a Part or Cut value is called “mapping” and is done by dragging Field Names between the “Fields in file” and “Fields from file” columns.

Map a field to a part or cut property

Drag a field from the “Fields in file” column to a row in the “Map to property” column where it will be mapped to that property for the part or cut.

Remove a mapping from a part or cut property

Drag the field from the “Fields from file” column back to the “Fields in file” column. The mapping will be removed from the mapping set.

Mapping Actions

A Mapping Action determines how a field from the import file will be loaded into the import property.

Action Types

Action types include copying the field value into the property, parsing a value using a default or custom regular expression, and applying a custom process such as a cuts template. The resulting value after processing is then copied into the property.

Copy field value

- Copies the input value to the mapped value
- The default mapping action
- Available for all properties

Use the default parser

- Parse the import value using a regular expression and copy the match to the mapped value
- Regular expression matches are single line and case insensitive
- Default regular expressions are sometimes available for supporting common industry keywords. See the section “Appendix B: Field Mappings regular expressions” for additional information.

Use the custom parser

- Parses the import value using a regular expression and copies the match to the mapped value
- Regular expression matches are single line and case insensitive

- Available for all properties

Custom action

- Executes an Importer custom action using the mapped value returned from any of the previous options.
- An importer specific extension to support advanced operations such as applying a set of calculated properties to a cut based on a parsed value.
- Is usually selectable via a custom property. If the property supports a custom action, then the Mapping Action Editor will include new settings to configure the action's behavior.

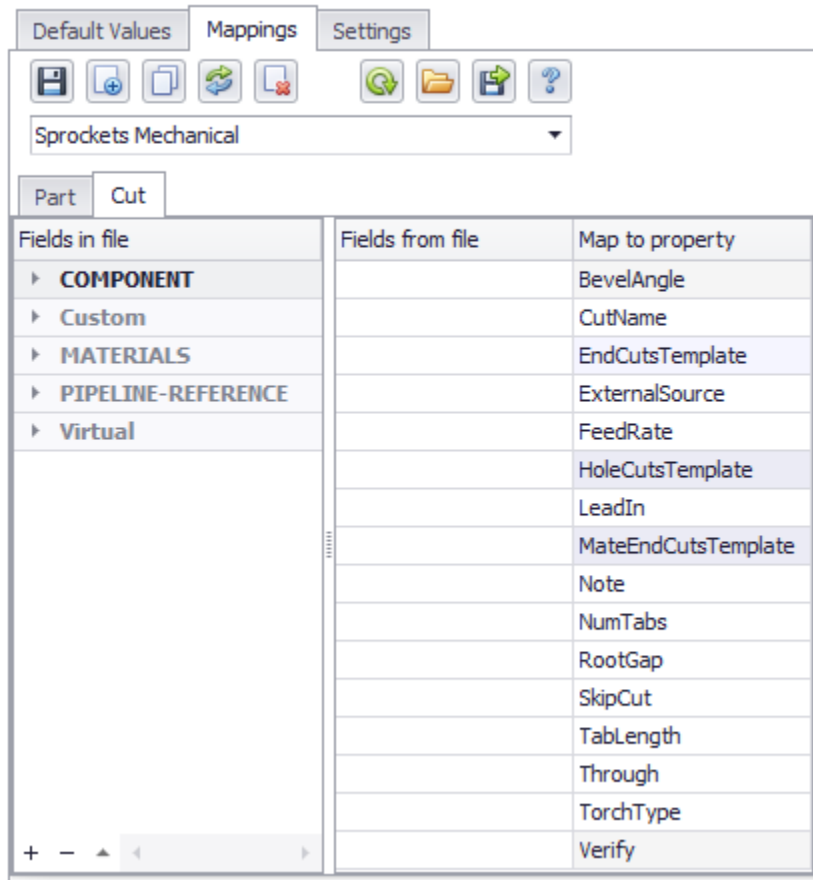
Note: When both "Use the default parser" and "Use the custom parser" are selected, the importer will first parse the input value using the default parser regular expression, and if a match is not found it will parse the input value using the custom parser regular expression and return the result from that match.

Available property action types

Each property may have their own unique set of available action types based on the data type and constraints of each property. Identification of which action types are available for a property can be determined by the color of the Field Mapper's "Map to property" column. See the figure below for examples.

Available action types identification

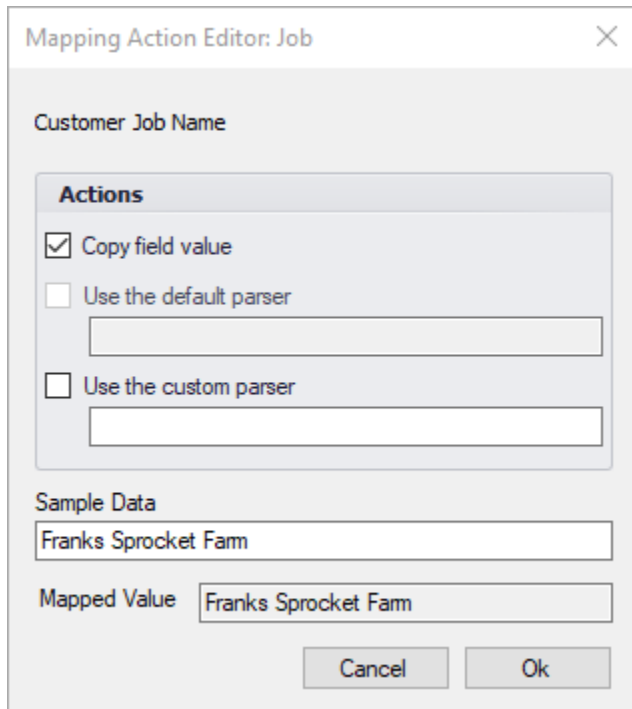
- White (example CutName)
 - Copy field value
 - Use the custom parser
- Shaded (examples – BevelAngle, Verify)
 - Copy field value
 - Use the default parser
 - Use the custom parser
- Blue (example – EndCutsTemplate)
 - Copy field value
 - Use the custom parser
 - Custom Action (Cuts Template)
- Shaded blue (examples – HoleCutsTemplate, MateEndCutsTemplate)
 - Copy field value
 - Use the default parser
 - Use the custom parser



Action Types color coding in “Map to property” column

Setting a Mapping Action

To open the Mapping Action Editor dialog, click on the “Map to property” cell to edit and then double-click the cell or press the Enter key.



The image shows a dialog box titled "Mapping Action Editor: Job". It has a close button (X) in the top right corner. The dialog is divided into several sections. At the top, there is a label "Customer Job Name". Below this is a section titled "Actions" which contains three options: "Copy field value" (checked), "Use the default parser" (unchecked), and "Use the custom parser" (unchecked). Each of the two unchecked options has an empty text input field next to it. Below the "Actions" section is a section labeled "Sample Data" with a text input field containing the text "Franks Sprocket Farm". Below that is a section labeled "Mapped Value" with a text input field also containing "Franks Sprocket Farm". At the bottom right of the dialog are two buttons: "Cancel" and "Ok".

Mapping Action Editor

Within the Mapping Action Editor mapping actions can be selected and the values for sample data and custom parser can be edited. Any text for “Mapped Value” will be updated as changes are made to allow testing of parsing actions and reviewing the values the importer will find during import.

Note that the “Copy field value” and the parsing actions are mutually exclusive. For example, if the “Copy field value” action is selected, both parsers will be deselected. Either or both parsers may be selected but doing so will automatically deselect the “Copy field value” action.

Mapping Action Examples

Some basic mapping scenarios and solutions are provided here as examples of how to build custom field mapping sets to solve common importing issues.

Copy field value

Use scenario: The DESCRIPTION field from the MATERIALS section contains information that should be copied to the MaterialType property of the part during import.

Mappings Table

Fields from file	Map to property
	BarCode
	Customer
	DateCreated
	DesignGroup
	ImportedFrom
	Job
	LockPartSeamOnPi...
	ManualCutOrder
ITEM-DESCRIPTION	MaterialType
	Name
	Notes
	NumScheduledParts
	OuterDiameter
	Priority
	Schedule
	SeamAng
	SpoolSheet
	WallThickness

The field DESCRIPTION from the MATERIALS section is mapped to the property MaterialType

Mapping Action Editor

The Mapping Action is set to "Copy field value"

The Mapped Value shows data will be copied during import.

Use the default parser

Use scenario: The ITEM-DESCRIPTION field from the PIPE component has the NPS Schedule embedded in it using a common industry pattern of "SCH xx". The schedule value needs to be parsed from the field during import and the resulting value stored in the virtual property Schedule for later calculation of pipe wall thickness.

Mappings Table

Fields from file	Map to property
	BarCode
	Customer
	DateCreated
	DesignGroup
	ImportedFrom
	Job
	LockPartSeamOnPi...
	ManualCutOrder
	MaterialType
	Name
	Notes
	NumScheduledParts
	OuterDiameter
	Priority
ITEM-DESCRIPTION	Schedule
	SeamAng
	SpoolSheet
	WallThickness

The field ITEM-DESCRIPTION from the COMPONENT (pipe) section is mapped to the virtual property Schedule

Mapping Action Editor

Mapping Action Editor: Schedule

NPS or DN Pipe Schedule

Actions

☐ Copy field value

☒ Use the default parser

(?<=SCHEDULE)(?:\.[s+l,H/](w+))?(?<=SCHED)(?:\.[

☐ Use the custom parser

Sample Data

A53B GR B ERW WALL 0.125 SCH 40

Mapped Value

40

Cancel
Ok

The Mapping Action is set to "Use the default parser"

The Mapped Value shows data will be parsed during import and the matching value (40 in this sample data) will be copied to the virtual property Schedule for later processing.

Use the custom parser

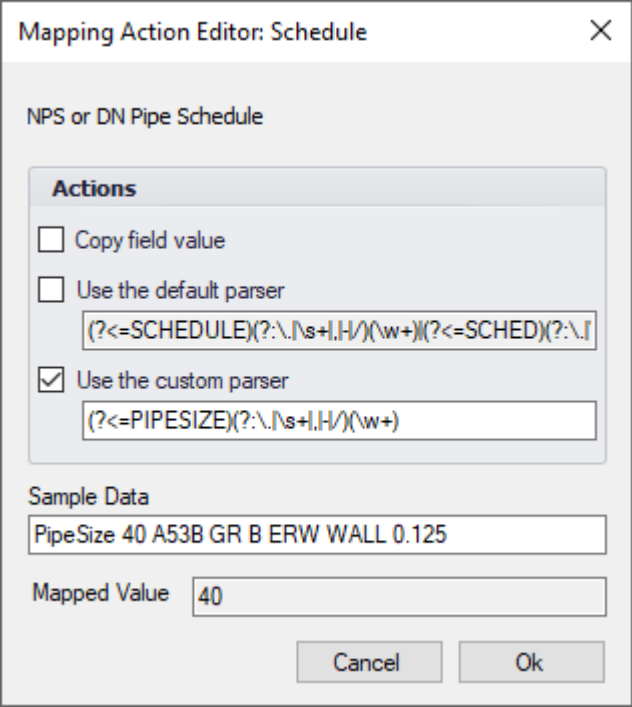
Use scenario: The ITEM-DESCRIPTION field from the PIPE component has the NPS Schedule embedded in it using a non-standard pattern of "PipeSize xx" . The schedule value needs to be parsed from the field during import and the resulting value stored in the virtual property Schedule for later calculation of pipe wall thickness.

Mappings Table

Fields from file	Map to property
	BarCode
	Customer
	DateCreated
	DesignGroup
	ImportedFrom
	Job
	LockPartSeamOnPi...
	ManualCutOrder
	MaterialType
	Name
	Notes
	NumScheduledParts
	OuterDiameter
	Priority
ITEM-DESCRIPTION	Schedule
	SeamAng
	SpoolSheet
	WallThickness

The field ITEM-DESCRIPTION from the COMPONENT (pipe) section is mapped to the virtual property Schedule

Mapping Action Editor



The dialog box is titled "Mapping Action Editor: Schedule". It contains a section "NPS or DN Pipe Schedule" with an "Actions" group box. In the "Actions" group, the "Use the custom parser" checkbox is checked. Below it, a text box contains the regular expression: `(?<=PIPE SIZE)(?:\.\s+|\./)(\w+)`. Below the "Actions" group is a "Sample Data" section with a text box containing "PipeSize 40 A53B GR B ERW WALL 0.125". At the bottom, a "Mapped Value" text box contains the number "40". There are "Cancel" and "Ok" buttons at the bottom right.

The Mapping Action is set to "Use the custom parser"
A custom regular expression has been configured

The Mapped Value shows data will be parsed during import and the matching value (40 in this sample data) will be copied to the virtual property Schedule for later processing.

Use the default parser, then custom parser

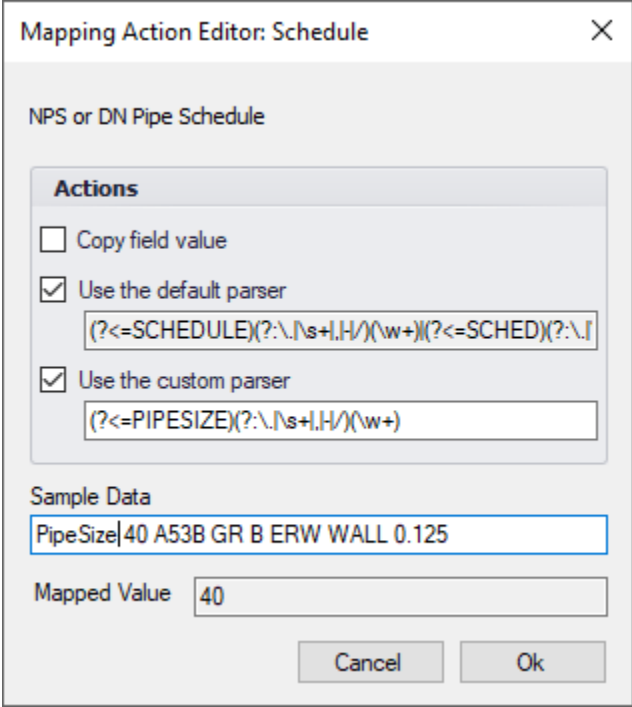
Use scenario: The ITEM-DESCRIPTION field from the PIPE component has the NPS Schedule embedded in it. Sometimes the embedded schedule value uses a common industry pattern like "SCH xx" or "SCHEDULE xx" and sometimes it is using a non-standard pattern of "PipeSize xx". The schedule value needs to be parsed from the field during import and the resulting value stored in the virtual property Schedule for later calculation of pipe wall thickness.

Mappings Table

Fields from file	Map to property
	BarCode
	Customer
	DateCreated
	DesignGroup
	ImportedFrom
	Job
	LockPartSeamOnPi...
	ManualCutOrder
	MaterialType
	Name
	Notes
	NumScheduledParts
	OuterDiameter
	Priority
ITEM-DESCRIPTION	Schedule
	SeamAng
	SpoolSheet
	WallThickness

The field ITEM-DESCRIPTION from the COMPONENT (Pipe) section is mapped to the virtual property Schedule

Mapping Action Editor



The dialog box is titled "Mapping Action Editor: Schedule". It contains a section "NPS or DN Pipe Schedule". Under the "Actions" section, there are two checkboxes: "Copy field value" (unchecked) and "Use the default parser" (checked). Below "Use the default parser" is a text box containing the regular expression: `(?<=SCHEDULE)(?:\.\s+|,|H/)(\w+)`. Below that, the "Use the custom parser" checkbox is also checked, with a text box containing the regular expression: `(?<=PIPESIZE)(?:\.\s+|,|H/)(\w+)`. The "Sample Data" section shows a text box with the value: "PipeSize|40 A53B GR B ERW WALL 0.125". The "Mapped Value" section shows a text box with the value: "40". At the bottom are "Cancel" and "Ok" buttons.

Both "Use the default parser" and "Use the custom parser" actions have been selected

A custom regular expression has been configured

During import, if the default parser cannot find a value, then the custom parser attempts to find and return a value.

The Mapped Value shows data will be parsed during import and the matching value (40 in this sample data) will be copied to the virtual property Schedule for later processing.

Olet specific hole sizes

Use scenario: An olet requires a specific hole diameter which is different than the nominal pipe size specified for the hole in the file. This value needs to be parsed from the Olet description during import and copied to the hole-cut's HoleDiameter property for later calculating the diameter of the olet or tee hole.

Mappings Table

Fields from file	Map to property
	BevelAngle
	CutName
	EndCutsTemplate
	ExternalSource
	FeedRate
	HoleCutsTemplate
ITEM-DESCRIPTION	HoleDiameter
	LeadIn
	MateEndCutsTemp...
	Note
	NumTabs
	RootGap
	SkipCut
	TabLength
	Through
	TorchType
	Verify

The field ITEM-DESCRIPTION from the COMPONENT (olet) section is mapped to the virtual property HoleDiameter

Mapping Action Editor

Mapping Action Editor: HoleDiameter

Diameter of the hole cut, overrides all other discovered diameters for cut.

Actions

☐ Copy field value

☒ Use the default parser

(?<=HOLEDIAMETER)(?:\.[s+|.|-])([d.]*)

☐ Use the custom parser

Sample Data

Aesterion R245 HoleDiameter 2.22

Mapped Value 2.22

Cancel Ok

The Mapping Action is set to "Use the default parser"

The Mapped Value shows data will be parsed during import and the matching value (2.22 in this sample data) will be copied to the virtual property HoleDiameter for later processing..

Mapping a field to multiple properties

Often a field in a file can carry information for more than one property. For example, consider a PCF file that has a PIPE component with an ITEM-DESCRIPTION which contains both the NPS schedule and material information.

ITEM-DESCRIPTION A53B GR B ERW SCH40

The entire description should be mapped to the MaterialType property, while the NPS Schedule should be parsed from ITEM-DESCRIPTION and mapped to the virtual property Schedule for later use when calculating pipe wall dimensions.

Creating a mapping set with the following mappings will enable these values to automatically be parsed and added to the imported parts.

Fields from file	Map to property
	BarCode
	Customer
	DateCreated
	DesignGroup
	ImportedFrom
	Job
	LockPartSeamOnPi...
	ManualCutOrder
ITEM-DESCRIPTION	MaterialType
	Name
	Notes
	NumScheduledParts
	OuterDiameter
	Priority
ITEM-DESCRIPTION	Schedule
	SeamAng
	SpoolSheet
	WallThickness

Example mapping set

Original Value:

ITEM-DESCRIPTION **A53B GR B ERW SCH40**

Map to property: MaterialType

Field from file: ITEM-DESCRIPTION

Action Type: Copy field value

Mapped Value: **A53B GR B ERW SCH40**

Map to property: Schedule

Field from file: ITEM-DESCRIPTION

Action Type: Use the default parser

Mapped Value: **40**

About Regular Expressions

Regular expressions provide a powerful, flexible, and efficient method for processing text. The extensive pattern-matching notation of regular expressions enables one to quickly parse large amounts of text to find specific character patterns. A regular expression (or regex for short) is a special text string for describing a search pattern. When importing a file, the Importer can use a pattern to search for and return values that might be embedded in the middle of a large description or other text. When it finds that value it will copy the parsed value to the property instead of the entire value.

Regular expression syntax is not intuitive and providing direction on how to use Regular Expressions is beyond the scope of this document. The internet has several Regular Expression tutorials, reference guides, and tools which can be used to design your custom patterns.

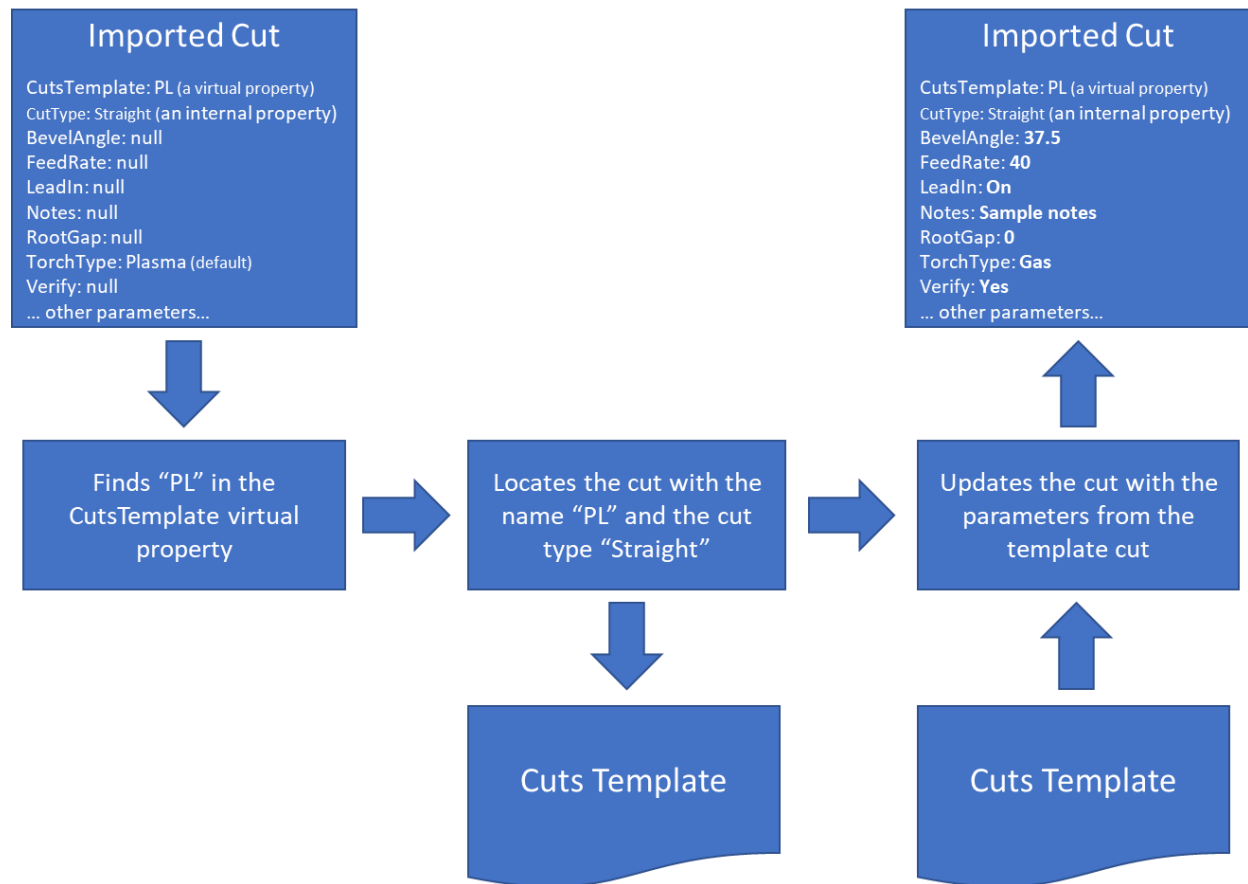
See “Appendix B: Field Mappings regular expressions” for more information on expressions used by PytypeServer.

Cuts Templates

Often designs can have several sets of cuts with similar parameters. Exporting those parameters so they're available during import would save effort and reduce error, but unfortunately exporting those cut parameters are often not natively supported by either the export schema or CAD system.

Cuts Templates allow designers to define and name common sets of cut parameters and embed these tags into any descriptive text attached to cuts during export. During import the Field Mapper can be configured to look for those tags and apply the named set of cut parameters to the imported cut.

The Cuts Template action is a custom importer action that may be attached to one or more virtual “Map to property” rows. This feature allows a CAD system to tag various components in an import file with sets of pre-defined cut parameters.



Applying a Cuts Template during import

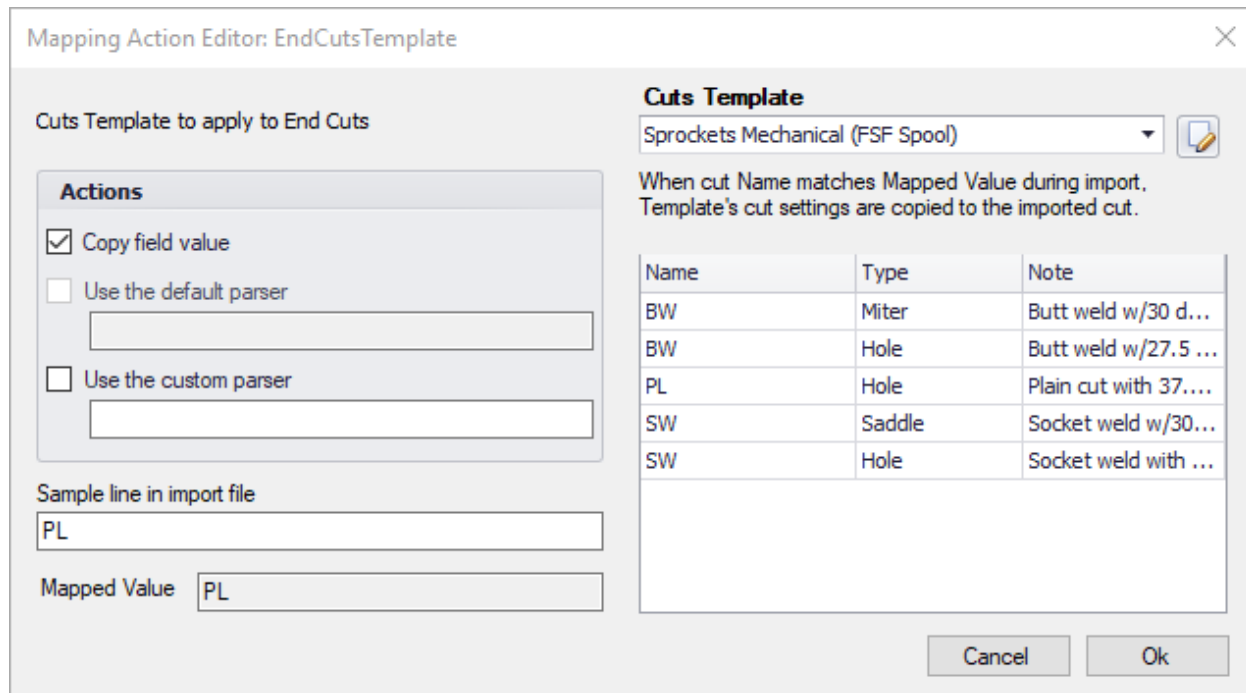
At import time an imported value is used to look up a cut from a previously selected cuts template and copy the properties from the template cut to the cut being imported. After the template has been applied to the cut, the importer will then apply any field mappings and calculated values, potentially overwriting values set by the cut template. See “Appendix A: Importer” for a list of importers and which cut properties they calculate during import, these may potentially overwrite a value set by a cuts template.

Cuts templates are stored and edited as a list of PS_JSON_Cut objects and not Importer specific, they can be reused among different Importers. Note though that Importers can have different rules governing what and how they import values, so the values in a cuts template that may set in one

importer many not be set in some other importer. See “Appendix A: Importer” and the document “Importable Fields” in the PyServer documentation for additional information.

Configuring a cuts template custom action

A virtual property containing a cuts template custom mapping action will display a Mapping Action Editor with an additional section for selecting a cuts template. This section has a dropdown list of available cuts templates and a detail section showing which cuts are available in the currently selected cuts template.



The dialog box is titled "Mapping Action Editor: EndCutsTemplate". It contains several sections:

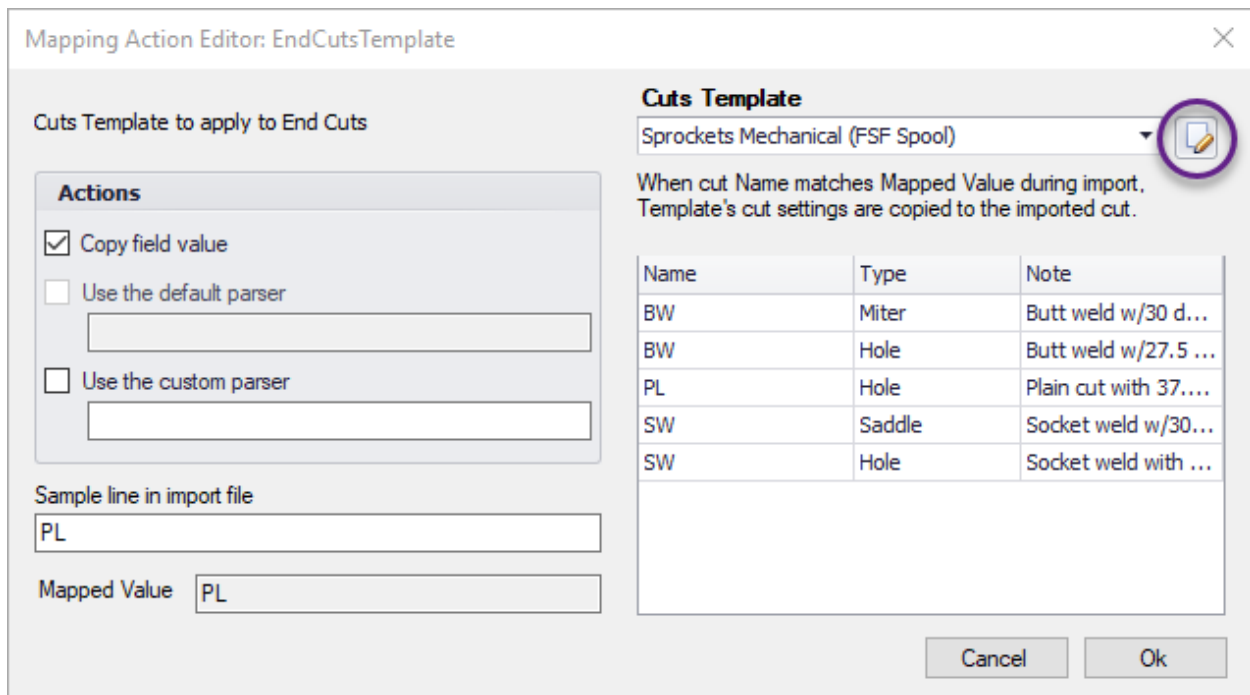
- Cuts Template to apply to End Cuts:** A dropdown menu showing "Sprockets Mechanical (FSF Spool)" with a document icon to its right.
- When cut Name matches Mapped Value during import, Template's cut settings are copied to the imported cut.** A text label.
- Actions:** A section with two checkboxes: "Copy field value" (checked) and "Use the default parser" (unchecked). Below "Use the default parser" is a text input field. Another checkbox "Use the custom parser" (unchecked) is below it, with another text input field.
- Sample line in import file:** A text input field containing "PL".
- Mapped Value:** A text input field containing "PL".
- Cuts Template:** A table showing the details of the selected template.

Name	Type	Note
BW	Miter	Butt weld w/30 d...
BW	Hole	Butt weld w/27.5 ...
PL	Hole	Plain cut with 37....
SW	Saddle	Socket weld w/30...
SW	Hole	Socket weld with ...

At the bottom right are "Cancel" and "Ok" buttons.

Mapping Action Editor for EndCutsTemplate property

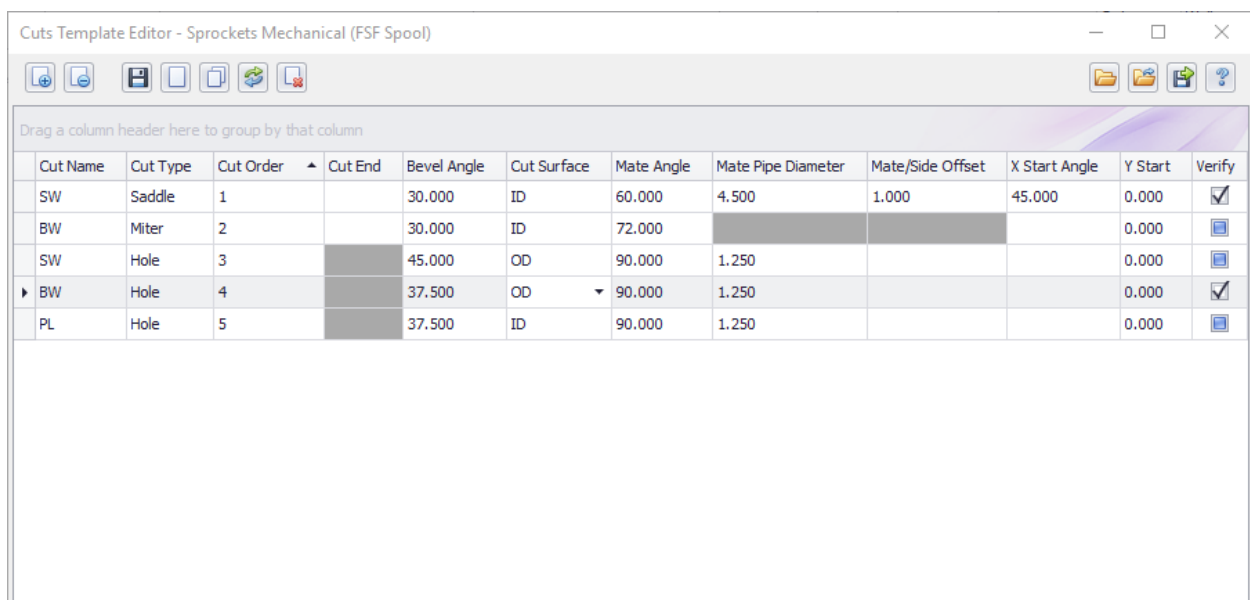
Cuts templates are managed via the “Cuts Templates” custom action control, from within the Mapping Actions Editor.



Mapping Action Editor Cuts Templates button

Cuts Template Editor

The Cuts Template Editor dialog contains 3 button groups, and a table with columns representing PS_JSON_Cut properties and rows representing cuts for the current cuts template. Property columns can be added and removed from the table as needed during editing. Note that some properties are cut type specific and when a property doesn't apply to the cut type then the column cell for that cut will be unavailable, as shown in the below example.



Cuts Template Editor dialog

Editor button groups

- The left button group adds and removes rows of cuts
- The center button group creates, saves copies, renames, or removes the current template
- The right button group allows the importing and exporting of cut templates formatted as PS_JSON_Cut objects and accessing additional Field Mapping help documentation. There is also a CSV file import feature, where the CSV column header row should have names that match PS_JSON_Cut property names.

Editing cuts

Cuts are added and deleted using the left button group, and cuts are edited simply by filling out values for a row.

Note that any values that are set in a cut template will be copied to a cut during import and Importers usually will not override a cut value with a calculated value if a value was already set by a template or mapping. It is strongly recommended that only values required for the job be set so mapped, calculated, and default values can be set later. See “Appendix A: Importer” for more information.

Order of values applied during an import

There is a specific order in which values are applied to a cut during a mapping operation.

- 1) Values from any assigned Cuts Templates are applied
 - If a cuts template is active and relevant for this cut, then any non-null values from that template are applied to the cut.
- 2) Mapped field values are parsed and applied
 - For any imported file fields that are mapped to a part or cut property, each field will be parsed, and that value applied to the part or cut.
 - Note, for cut properties a value from a mapped field will overwrite a value applied by a cuts template.
- 3) Dimensional values such as Diameter, WallThickness or YStart are calculated and applied.
 - Most importers calculate dimensional values based on geometry information encoded in the file. In most cases, any values set by a cuts template or field mapping will be overwritten by these Importer calculated values. See “Appendix A: Importer” for importer specific details.
- 4) Default values defined in Import Manager are applied during PyServer import
 - After the Importer processes the file, the Importer Manager may apply additional default values for parts and cuts that have null values. These values are set by the user in the Default Values tab in the Importer Manager.

Schema specific cut definitions

While a cuts template name could be defined from any import field, import file specifications often have a component or property which either implies or specifically states a connection type or cut between components. Understanding how an import file schema defines connections between components is critical for configuring a cuts template and mapping set which generates error free imports.

When this information is available, importers will often expose internal schema values and relationships as virtual fields in the “Fields in file” column and provide matching virtual properties in the “Map to property” column which support cuts templates. These virtual fields and properties are used for mapping these schema values and relationships to cuts during import.

CSV schema

CSV files are free-form and have no schema. Any imported values can be mapped to any end-cut property and no dimensional values are calculated during import. Hole cuts are not supported because there are no geometric or other connection information available.

End Cuts

A cuts template can be applied to end cuts by mapping a field containing an embedded keyword and cut name to the virtual property “CutsTemplate” and configuring the mapping action as needed to retrieve the name of the cut to apply from the cuts template. The importer provides a default regular expression which can be used for a case-insensitive match of the keyword CUTSDEFAULTNAME and retrieve the first word after that keyword.

Example

Mappings Table

Fields from file	Map to property
	Bevel Angle
	Cut Name
	Cut Order
CutsTemplate	Cuts Template
	Feed Rate
	Hole Diameter
	Lead In
	Miter
	Note
	Num Tabs
	Root Gap
	Skip Cut
	Through
	Torch Type
	Verify

The column CutsTemplate from the CSV file is mapped to the virtual property CutsTemplate

Mapping Action Editor

Mapping Action Editor: CutsTemplate

Cuts Template
Sprockets Mechanical (FSF Spool)

When cut Name matches Mapped Value during import, Template's cut settings are copied to the imported cut.

Name	Type	Note
BW	Miter	Butt weld w/30 d...
BW	Hole	Butt weld w/27.5 ...
PL	Hole	Plain cut with 37...
SW	Saddle	Socket weld w/30...
SW	Hole	Socket weld with ...

Actions

☒ Copy field value

☐ Use the default parser

☐ Use the custom parser

Sample line in import file: PL

Mapped Value: PL

Cancel Ok

Cuts Template “Sprocket Mechanical” selected
Sample Data/Mapped Value is “PL”

At import time, any row in the CSV file where the column CutsTemplate has a value of PL and a cut type of Miter will have the PL cut properties copied to it. .

IFC schema

IFC files have a connection element type named IfcRelConnectsPortToElement which defines connections for IfcFlowSegments (pipes) and IfcFlowDistributionElements (fixtures). This connection element can be used to map a cuts template for all standard end and hole cuts being created during an import. The IFC Importer displays the IfcRelConnectsPortToElement element as the ConnectionType section in the “Fields in file” column which provides access to the element’s Name and Description properties.

At import time, the importer determines which IfcRelConnectsPortToElement is associated with the cut being imported and can parse the contents of the Name or Description properties of that element to determine whether to apply a cuts template.

End and Hole Cuts

A cuts template can be applied to end and hole cuts by mapping a field containing an embedded keyword and cut name to the virtual property “CutsTemplate” and configuring the mapping action as needed to retrieve the name of the cut to apply from the cuts template. The importer provides a default regular expression which can be used for a case-insensitive match of the keyword CUTSDEFAULTNAME and retrieve the first word after that keyword.

Example

Mappings Table

Fields from file	Map to property
	AltCutsTemplate
	BevelAngle
	CutName
Description	CutsTemplate
	ExternalSource
	FeedRate
	LeadIn
	Miter
	Note
	NumTabs
	RootGap
	SkipCut
	TabLength
	Through
	TorchType
	Verify

The field Description from the ConnectionType section is mapped to the virtual property CutsTemplate

Mapping Action Editor

Mapping Action Editor: CutsTemplate

The cuts template to apply to cuts

Actions

☐ Copy field value

☒ Use the default parser

(?<CUTSDEFAULTNAME\s+)(\w+)(?=\s*)(?:T)?(\w)

☐ Use the custom parser

Sample Data

CutsDefaultName: PL: Ø00LB RF SLIP ON FLG

Mapped Value: PL

Cuts Template

Sprockets Mechanical (FSF Spool)

When cut Name matches Mapped Value during import, Template's cut settings are copied to the imported cut.

Name	Type	Note
PL	Straight	Plan cut w/10 degree be...
BW	Miter	Butt weld w/30 degree b...
SW	Saddle	Socket weld w/30 degree...
PL	Hole	Plain cut w/37.5 degree ...
BW	Hole	Butt weld w/37.5 degree...
SW	Hole	Socket weld with/45 degr...

Cancel Ok

Cuts Template “Sprocket Mechanical” selected
Sample Data/Mapped Value is “PL”

At import time, any explicitly defined end or hole cut with a PL connection type and the same cut type will have the PL cut properties copied to it.

Note: There are no Miter or Saddle cuts defined in the cuts template that have a PL name, and any Miter or Saddle cuts with a PL connection type will not have any template cuts properties applied.

Alternate End Cuts

While the EndCutsTemplate mapping will support most cuts template scenarios, a scenario exists where a pipe end cut may not have an explicit connection defined via a IfcRelConnectsPortToElement element, either because it is not included in the import file or the pipe was an “extended tee” which was derived from a pipe fitting and would have no explicit end connection on the tee end of the pipe. This scenario is not explicitly supported by the IFC specification but is exported by several industry CAD systems.

A cuts template can be applied to this type of end cut by mapping a field containing an embedded keyword and cut name to the virtual property “AltCutsTemplate” and configuring the mapping action as needed to retrieve the name of the cut to apply from the cuts template. The importer provides a default regular expression which can be used for a case-insensitive match of the keyword ALT CUTSDEFAULTNAME and retrieve the first word after that keyword.

During import time, an alternate cuts template will be applied to an end cut when all the following conditions are met.

- There is a mapping for the virtual property AltCutsTemplate
- A cuts template has not already been applied to this cut.
- The mapping parses and returns a cut name that matches a template cut
- The cut being processed, and the template cut are of the same type

Note that if neither end cut of a pipe has an associated IfcRelConnectsPortToElement element and a cuts template has not already been applied, the alternate cut template could be applied to both end cuts of the pipe.

Example

Mappings Table

Fields from file	Map to property
Description	AltCutsTemplate
	BevelAngle
	CutName
	CutsTemplate
	ExternalSource
	FeedRate
	LeadIn
	Miter
	Note
	NumTabs
	RootGap
	SkipCut
	TabLength
	Through
	TorchType
	Verify

The field Description from the ConnectionType section is mapped to the virtual property AltCutsTemplate

Mapping Action Editor

Cuts Template “Sprocket Mechanical” selected
Sample Data/Mapped Value is “PL”

At import time, any end cut without an explicit connection type that exists on a pipe with a mapped value which matches a cut name and cut type in the cuts template will have the PL cut properties copied to the imported cut.

PCF schema

PCF files have an explicitly defined “end connection type” as the last value of the component attributes END-POINT and BRANCH-POINT. This “end connection type” value can be used to map a cuts template for all standard end cuts created during import. The PCF Importer supports this by providing a virtual field named ConnectionType which exposes the PCF “end connection type” in the “Fields in file” column. The PCF Importer also provides the virtual import properties EndCutsTemplate, HoleCutsTemplate, and MateEndCutsTemplate, to support cuts templates mapping.

End Cuts

A cuts template is applied to end cuts by mapping the virtual field ConnectionType to the virtual property “EndCutsTemplate” and then configuring the Action to “Copy field value” and selecting a pre-configured cuts template which has cuts defined with names that match the imported values from the ConnectionType.

Example

Mappings Table

Fields from file	Map to property
	BevelAngle
	CutName
ConnectionType	EndCutsTemplate
	ExternalSource
	FeedRate
	HoleCutsTemplate
	LeadIn
	MateEndCutsTemp...
	Note
	NumTabs
	RootGap
	SkipCut
	TabLength
	Through
	TorchType
	Verify

The virtual field ConnectionType is mapped to the virtual property EndCutsTemplate

Mapping Action Editor

Mapping Action Editor: EndCutsTemplate

Cuts Template to apply to End Cuts: Sprockets Mechanical (FSF Spool)

Actions

- ☒ Copy field value
- ☐ Use the default parser
- ☐ Use the custom parser

Sample Data: PL

Mapped Value: PL

Cuts Template

When cut Name matches Mapped Value during import, Template's cut settings are copied to the imported cut.

Name	Type	Note
PL	Straight	Plan cut w/10 degree be...
BW	Miter	Butt weld w/30 degree b...
SW	Saddle	Socket weld w/30 degree...
PL	Hole	Plain cut w/37.5 degree ...
BW	Hole	Butt weld w/37.5 degree...
SW	Hole	Socket weld with/45 degr...

Cancel Ok

Cuts Template "Sprocket Mechanical" selected
Sample Data/Mapped Value is "PL"

At import time, a straight cut with a PL connection type will have the PL straight cut properties copied to it.

Note: There are no Miter or Saddle cuts defined in the cuts template that have a PL name, and any Miter or Saddle cuts with a PL connection type will not have any template cuts properties applied.

Hole Cuts

A cuts template can be applied to hole cuts by mapping a field containing an embedded keyword and cut name to the virtual property "HoleCutsTemplate" and configuring the mapping action to retrieve the name of the cut to apply from the cuts template. The importer provides a default regular expression which can be used for a case-insensitive match of the keyword HOLECUTDEFAULTNAME and retrieve the first word after that keyword.

Example

Mappings Table

Fields from file	Map to property
	BevelAngle
	CutName
	EndCutsTemplate
	ExternalSource
	FeedRate
DESCRIPTION	HoleCutsTemplate
	LeadIn
	MateEndCutsTemp...
	Note
	NumTabs
	RootGap
	SkipCut
	TabLength
	Through
	TorchType
	Verify

The field DESCRIPTION is mapped to the virtual property HoleCutsTemplate

Mapping Action Editor

Cuts Template “Sprocket Mechanical” selected
Sample Data/Mapped Value is “SW”

At import time, a hole cut defined by a TEE-SET-ON component with a Description that contains the phrase “HoleCutDefaultName SW” will have the SW hole cut properties copied to it.

Mate End Cuts for Extended TEE

The PCF Importer supports an “extended tee” scenario where the center and branch points of a TEE-SET-ON component are used to imply a PIPE component via the “Create pipe from Extended TEE” setting on the Settings tab. This scenario is not explicitly supported by the PCF specification but is exported by several industry CAD systems and introduces a situation where an end connection type is not specified for the tee end of the pipe.

To resolve this edge scenario, the importer provides a virtual property “MateEndCutsTemplate” and a default regular expression to match and return the first word after the keyword MATEPIPECUTDEFAULTNAME”.

During import time, a cuts template will be applied to an end cut when all the following conditions are met.

- The importer setting “Create pipe from Extended TEE” is enabled
- The end cut being processed doesn’t have an explicit end connection type defined
- The pipe was generated from an extended TEE-SET-ON or TEE-STUB-IN
- There is a mapping for the virtual property MateEndCutsTemplate
- The mapping parses and returns a cut name that matches a template cut
- The cut being processed, and the template cut are of the same type

Example

Mappings Table

Fields from file	Map to property
	BevelAngle
	CutName
	EndCutsTemplate
	ExternalSource
	FeedRate
	HoleCutsTemplate
	LeadIn
DESCRIPTION	MateEndCutsTemp...
	Note
	NumTabs
	RootGap
	SkipCut
	TabLength
	Through
	TorchType
	Verify

The field DESCRIPTION is mapped to the virtual property MateEndCutsTemplate

Mapping Action Editor

Mapping Action Editor: MateEndCutsTemplate

cut template specified by the MatePipeCutDefaultName keyword.
Mate pipe end cuts are implied when creating pipes from

Actions

☐ Copy field value

☒ Use the default parser

☐ Use the custom parser

(?<=MATEPIPECUTDEFAULTNAME\s+)(\w+)(?=\s"

Sample Data

MatePipeCutDefaultName BW DESCRIPTION 3/4" X 24" F

Mapped Value BW

Cuts Template

Sprockets Mechanical (FSF Spool)

When cut Name matches Mapped Value during import, Template's cut settings are copied to the imported cut.

Name	Type	Note
PL	Straight	Plan cut w/10 degree be...
BW	Miter	Butt weld w/30 degree b...
SW	Saddle	Socket weld w/30 degree...
PL	Hole	Plain cut w/37.5 degree ...
BW	Hole	Butt weld w/37.5 degree...
SW	Hole	Socket weld with/45 degr...

Cancel

Ok

Cuts Template "Sprocket Mechanical" selected
Sample Data/Mapped Value is "BW"

At import time, an end cut defined by the CENTRE-POINT of a TEE-SET-ON component with a Description that contains the phrase "MatePipeCutDefaultName BW" will have the BW end cut properties copied to it.

Appendix A: Importers supporting Field Mapping

Importers currently supporting the Field Mapper feature with technical details and mapping related behaviors.

CSV Importer

Description

a comma-separated values (CSV) file is a delimited text file that uses a comma to separate values. A CSV file stores tabular data (numbers and text) in plain text where each line of the file is a data record. Each record consists of one or more fields, separated by commas.

Supported import file types

Any comma delimited file where the first line contains a list of field names for the remaining records.

Settings

- Import as Metric
 - When enabled, import all measurements as metric
 - Default setting is off.
- Cuts Template can set cut type
 - When enabled, a cuts template matching only by name can be used to set the cut type. For example, a cut template with a cut type of Saddle can be applied to the end cut of a part even though it was originally defined as a Miter cut.
 - Default setting is off.

Properties populated during import

The following properties are calculated and populated during import and will overwrite values that may have been previously set by a cuts template.

End cut properties

Importer Property	Note
CutEnd	
CutName	“End1” or “End2”
CutType	If setting “Cuts Template can set cut type” is not enabled
YStart	

Hole cut properties

The CSV Importer does not support hole cuts.

IFC Importer

Description

The Industry Foundation Classes IFC specification is a neutral data format to describe, exchange and share information typically used within the building and facility management industry sector. IFC is the international standard for openBIM and with IFC4 an ISO standard (ISO 16739).

The IFC specification is developed and maintained by buildingSMART International, bSI, (formally known as International Alliance for Interoperability, IAI).

Supported import file types

Industry Foundation Classes (.ifc, .ifcxml, ifczip, step21, .xbim) version 2x3 and 4.0

See the IFC Wiki for more information.

Settings

- **Create pipe from pipe fittings**
 - Creates pipes from pipe fittings when fittings have pipe like characteristics. Some files will use a separate butt weld pipe fitting with the same pipe wall geometry as the pipe itself to represent the join between two pipe sections, even though one pipe is intended to be welded directly to the other.
 - Default setting is on.
- **Auto-Merge Pipe Fittings with Pipes**
 - Merges same sized pipe fittings on pipe ends into the pipe when those pipe fittings have the same pipe wall geometry. Some files represent a pipe to pipe connection using a pipe fitting with the same pipe wall dimensions as the pipe. The pipe and pipe fitting could effectively merge together as one unit. Setting this allows the Importer to merge like pipes ends with their connected fittings together into one pipe.
 - Default setting is on.
- **Normalize diameters to schedule sizes**
 - Rounds non-standard pipe diameters to the nearest diameter in the schedule. Some pipes will have a pipe wall geometry that while close doesn't exactly match those in the standard pipe schedules.
 - Default setting is on
- **Coordinates Tolerance**
 - Sets the distance tolerance used when detecting whether components are connected to each other. Some exports have larger tolerances than the default and some files might have an issue where the coordinates of two separate components that are connected do not have exactly the same coordinates.
 - Default setting is 0.1

Properties populated during import

The following properties are calculated and populated during import and will overwrite values that may have been previously set by a cuts template.

End cut properties

Importer Property	Note
CutEnd	
CutName	"End1" or "End2"
CutType	
Mate1Angle	If not previously set by a mapping
Mate1Diameter	If not previously set by a mapping
Mate1Offset	If not previously set by a mapping
SizeOnSize	
XAngleStart	If not previously set by a mapping
YStart	

Hole cut properties

Importer Property	Note
BevelAngle	Set to 0 only if currently set machine has only 4 axis capability
CutName	Mate pipe's name, if one exists
CutType	
Mate1Angle	
Mate1Diameter	If not previously set by a mapping
Mate1Offset	
SizeOnSize	
YStart	

PCF Importer

Description

A Piping Component File (PCF) is a text-based English language style file format for defining piping isometrics and transferring information between piping design systems such as Isogen.

Supported import file types

Piping Component File (.pcf), version 2014 R1 (12.1)

See the Piping Component File (PCF) Reference Guide published by Alias Piping Solutions (Alias Limited).

Settings

- **Search descriptions for pipe values**
 - The Importer will search the ITEM-DESCRIPTION and DESCRIPTION attributes for pipe values when it cannot locate them using the standard PCF attributes. Some PCF files do not use the standard PCF attributes to encode pipe values such as outer diameter, wall thickness, or pipe schedule. Instead they encode them in the component's ITEM-DESCRIPTION attribute or the DESCRIPTION attribute located in the MATERIALS section.
 - Default setting is on.
- **Automatically fix reversed TEE coordinates**
 - The Importer will automatically reverse TEE-SET-ON or TEE-STUB coordinates when the BRANCH1-POINT and CENTRE-POINT coordinates are detected as backwards in relation to the TEE's mate pipe centerline. A TEE-SET-ON or TEE-STUB should have a CENTRE-POINT coordinate that is on the centerline of the mate pipe for that TEE component.
 - Some PCF files have TEE-SET-ON or TEE-STUB components where the TEE's BRANCH1-POINT coordinates are on the mate pipe's centerline while the CENTRE-POINT coordinates are where the BRANCH1-POINT should be.
 - Default setting is on.
- **Automatically fix hole cut diameters**
 - The Importer will use the mate pipe's diameter for hole cuts when the TEE's BRANCH1-POINT diameter does not match its mate pipe's END-POINT diameter. A TEE-SET-ON or TEE-STUB should have a BRANCH1-POINT diameter that matches the END-POINT diameter for the pipe that is attached to that BRANCH1-POINT. Some exported PCF files might have an issue where the TEE's BRANCH1-POINT diameter is set to the diameter of mate pipe attached to the TEE's CENTRE-POINT instead of the mate pipe attached to the TEE's BRANCH1-POINT.
 - Default setting is on
- **Create pipe from Extended TEE**
 - Creates a pipe when a TEE-SET-ON or TEE-STUB has a BRANCH1-POINT coordinate that extends beyond the mate pipe's outer diameter. A common design/export scenario exists where all

pipes are not explicitly specified in the PCF file. Instead they are implied using the PCF components TEE-SET-ON or TEE-STUB and setting a BRANCH1-POINT coordinate that extends beyond the TEE mate pipe's outer surface. The new pipe will start from the mate pipe's outer surface and end at the TEE's BRANCH1-POINT. The TEE BRANCH1-POINT will be updated to a location on the pipe's outer surface.

- Default setting is on
- **Pipe to Pipe Connections**
 - Creates matching saddle and hole cuts when a pipe end intersects with the center-line of another pipe. A common design/export scenario exists where TEE-SET-ON, TEE_STUB and CROSS-SET-ON coupling components are not explicitly specified. Instead they are implied when a pipe with an END-POINT coordinate intersects with the center-line of another pipe.
 - Default setting is on
- **Coordinates Tolerance**
 - Sets the distance tolerance used when detecting whether components are connected to each other. Some exports have larger tolerances than the default and some files might have an issue where the coordinates of two separate components that are connected do not have the same coordinates.
 - Default setting is 0.1

Properties populated during import

The following properties are calculated and populated during import and will overwrite values that may have been previously set by a cuts template.

End cut properties

Importer Property	Note
BevelAngle	If not previously set by a mapping
CutEnd	Either End1 or End2
CutName	
CutSurface	
CutType	
Mate1Angle	If not previously set by a mapping
Mate1Diameter	If not previously set by a mapping
Mate1Offset	If not previously set by a mapping
SizeOnSize	
XAngleStart	
YBranchAngle	
YStart	

Hole cut properties

Importer Property	Note
BevelAngle	Set to 0 only if currently set machine has only 4 axis capability
CutName	PCF component's ITEM-CODE or MATERIAL-IDENTIFIER
CutType	
Mate1Angle	
Mate1Diameter	If not previously set by a mapping
Mate1Offset	
SizeOnSize	

PypeServer Importer

Description

A text file containing PypeServer parts and cuts defined using a JSON schema defined by PypeServer.

Supported import file types

Files ending with psvr or json that are formatted using the PypeServer JSON schema. Please contact PypeServer for schema details.

Settings

- **Normalize diameters to schedule sizes**
 - Rounds non-standard pipe diameters to the nearest diameter in the schedule. Some pipes will have a pipe wall geometry that while close doesn't exactly match those in the standard pipe schedules. Setting this enables the Importer to adjust non-standard pipe wall geometry to the closest matching schedule geometry. Example: A pipe is imported with an outer diameter of 8.6 inches and the closest NPS nominal diameter is 8 inches. The pipe outer diameter will be adjusted to be 8.625 inches per the NPS schedule.
 - Default setting is off.

Special notes

PypeServer will by default import all data contained in the file.

Stratus Importer

Description

STRATUS is a web-based software application which integrates with Autodesk BIM 360, Fabrication, AutoCAD, and Revit.

Supported import file types

Stratus web service using a REST API. See <https://api.gtpstratus.com/>

Settings

- **Transactional Importing**
 - Stratus sends cut lists to a tool queue sequenced and sorted by pipe type and the next sequence of parts is not sent to the tool download queue until the tool queue has been cleared of the previous set. During manual importing, this sequencing of sets can require the operator to select and import parts several times until the Stratus tool download queue is empty. When this setting is disabled, parts are immediately deleted from the tool queue and PypeServer will continue checking the tool queue for new parts until it has confirmed the queue is empty. Note: When this setting is disabled, any parts not imported are still removed from the tool queue. The cut lists for those parts must be resent from Stratus to download these parts again.
 - Default setting is on.
- **Normalize diameters to schedule sizes**
 - Stratus sends cut lists to a tool queue sequenced and sorted by pipe type and the next sequence of parts is not sent to the tool download queue until the tool queue has been cleared of the previous set. During manual importing, this sequencing of sets can require the operator to

select and import parts several times until the Stratus tool download queue is empty. When this setting is disabled, parts are immediately deleted from the tool queue and PypeServer will continue checking the tool queue for new parts until it has confirmed the queue is empty. Note: When this setting is disabled, any parts not imported are still removed from the tool queue. The cut lists for those parts must be resent from Stratus to download these parts again.

- Default setting is off.

Properties populated during import

Stratus queues part cut lists to tool download queue using the PypeServer JSON schema. PypeServer will by default import all data contained in the JSON objects downloaded from Stratus.

Special notes

When creating a new “Mapping Set”, PypeServer will only show the core Stratus part fields. To see any custom/extended part properties one must first download (but don’t import yet) a set of Stratus cut-lists/parts from Stratus that contain those properties and then click the “Refresh” button to copy the extra Stratus part properties from those parts into the new Mapping Set.

SysQue Importer

Description

An Excel file or comma-delimited file containing parts defined and exported from SysQue.

Supported import file types

Excel xls, xlsx file and comma-delimited files.

Settings

- **Normalize diameters to schedule sizes**
 - Stratus sends cut lists to a tool queue sequenced and sorted by pipe type and the next sequence of parts is not sent to the tool download queue until the tool queue has been cleared of the previous set. During manual importing, this sequencing of sets can require the operator to select and import parts several times until the Stratus tool download queue is empty. When this setting is disabled, parts are immediately deleted from the tool queue and PypeServer will continue checking the tool queue for new parts until it has confirmed the queue is empty. Note: When this setting is disabled, any parts not imported are still removed from the tool queue. The cut lists for those parts must be resent from Stratus to download these parts again.
 - Default setting is off.
- **Treat 'Pipe Fitting' parts as 'Pipe' parts**
 - SysQue parts in the import file that are labeled with a 'Pipe Fitting' PartType will be converted to a PartType of 'Pipe' and imported as a PypeServer part.
 - Default setting is off.

Special notes

PypeServer will use the provided end-point coordinates to and add saddle and hole cuts where parts intersect.

Tekla Importer

Description

An XML formatted text file developed by the Tekla Corporation.

Supported import file types

XML files formatted to conform to the Tekla schema referenced at <http://www.hgg.nl/schemas/procam>.

Settings

- **Normalize diameters to schedule sizes**
 - Stratus sends cut lists to a tool queue sequenced and sorted by pipe type and the next sequence of parts is not sent to the tool download queue until the tool queue has been cleared of the previous set. During manual importing, this sequencing of sets can require the operator to select and import parts several times until the Stratus tool download queue is empty. When this setting is disabled, parts are immediately deleted from the tool queue and PypeServer will continue checking the tool queue for new parts until it has confirmed the queue is empty. Note: When this setting is disabled, any parts not imported are still removed from the tool queue. The cut lists for those parts must be resent from Stratus to download these parts again.
 - Default setting is off.

Special notes

PypeServer will use the provided end-point coordinates to and add saddle and hole cuts where parts intersect.

End cut properties

Importer Property	Note
BevelAngle	If not previously set by a mapping
CutEnd	Either End1 or End2
CutName	
CutSurface	
CutType	
Mate1Angle	If not previously set by a mapping
Mate1Diameter	If not previously set by a mapping
Mate1Offset	If not previously set by a mapping
SizeOnSize	
XAngleStart	
YBranchAngle	
YStart	

Hole cut properties

Importer Property	Note
BevelAngle	Set to 0 only if currently set machine has only 4 axis capability
CutName	PCF component's ITEM-CODE or MATERIAL-IDENTIFIER
CutType	
Mate1Angle	
Mate1Diameter	If not previously set by a mapping
Mate1Offset	
SizeOnSize	
YStart	

Victaulic Importer

Description

A text file containing parts and cuts defined using an extended JSON schema defined by PypeServer.

Supported import file types

Files ending with psvr or json that are formatted using the PypeServer JSON schema. Please contact PypeServer for schema details.

Settings

- **Normalize diameters to schedule sizes**
 - Rounds non-standard pipe diameters to the nearest diameter in the schedule. Some pipes will have a pipe wall geometry that while close doesn't exactly match those in the standard pipe schedules. Setting this enables the Importer to adjust non-standard pipe wall geometry to the closest matching schedule geometry. Example: A pipe is imported with an outer diameter of 8.6 inches and the closest NPS nominal diameter is 8 inches. The pipe outer diameter will be adjusted to be 8.625 inches per the NPS schedule.
 - Default setting is off.

Special notes

PypeServer will use the provided end-point coordinates to and add saddle and hole cuts where parts intersect.

Appendix B: Field Mappings regular expressions

The following default regular expressions are provided for reference only and may change at any time.

Note: All expressions are a single line and are word-wrapped here to fit within the document margins.

All supported importers

The following regular expressions are available on all importers which support a Field Mapper.

Outer Diameter

Default regular expression for the part OuterDiameter property.

Purpose

Sets the outer diameter of a pipe. Useful for explicitly specifying diameter for non-standard sizes or when a schedule identifier is not available.

Supported patterns

- xx.xx O.D., OD xx.xx, O.D. xx.xx, OUTER DIAMETER xx.xx, xx" O.D.
- Space, period, comma or dash as separators

Expression

```
(?<=OD)(?:\.\s+|,|-)([d.]+)|(?<=O.D.)(?:\.\s+|,|-)([d.]+)|(?<=OUTER\sDIAMETER)(?:\.\s+|,|-)([d.]+)|([d.]+)(?:"\s+)(?=O.D.)
```

Examples

O.D. 6.25, o.d.-6.25, OD 6.25, OD-6.25, OUTER DIAMETER 6.25, Outer Diameter-6.25, 6.25" O.D.

Schedule

Default regular expression for the Schedule virtual property.

Purpose

Stores an NPS/DN pipe schedule for later use to calculate the pipe wall dimensions. Useful for calculation pipe dimensions by using a pipe schedule identifier.

Supported patterns

- SCH xxx, SCHED xxx, SCHEDULE xxx, SCHxxx, SCHEDxxx, SCHEDULExxx
- DNxxx, DN xxx
- Space, period, comma or dash as separators
- Trimble specific keywords STD, IPS, XH and XXH

Expression

```
(?<=SCHEDULE)(?:\.\s+|,|-|/)(\w+)|(?<=SCHED)(?:\.\s+|,|-|/)(\w+)|(?<=SCH)(?:\.\s+|,|-|/)(\w+)|(?<=DN)(?:\.\s+|,|-|/)(\w+)|(?<=DN)(\w+)|(?<=\s+|^|-)(STD)(?=\s+|$)|(?<=\s+|^)(IPS)(?=\s+|$)|(?<=\s+|^)(XH)(?=\s+|$)|(?<=\s+|^)(XXH)(?=\s+|$)|(?<=SCHEDULE)(\d+)|(?<=SCHED)(\d+)|(?<=SCH)(\d+)
```

Examples

SCH 40, SCH-xxs, SCHED 40, SCHED-XXS, SCHEDULE 40, Schedule,40, DN 40, DN40, STD, XH, XXH, IPS

Verify

Default regular expression for the Verify property.

Supported patterns

VERIFYHOLE, VERIFYHOLECUT, VERIFYENDCUTS

Purpose

Sets the “Verify” cut property to true. Useful for flagging specific cuts for operator verification.

Expression

(VERIFYHOLECUT)|(VERIFYHOLE)|(VERIFYENDCUTS)

Examples

VERIFYHOLECUT, VERIFYHOLE, VERIFYENDCUTS

Wall Thickness

Default regular expression for the part WallThickness property.

Purpose

Sets the wall thickness of a pipe. Useful for explicitly specifying wall thickness for non-standard sizes or when a schedule identifier is not available.

Supported patterns

- WALL xx.xx, WT xxx, WALL THICKNESS xx.xx, xx.xx" WALL
- Space, period, comma or dash as separators

Expression

(?<=WALL)(?:\.\s+|,|-)([\\d.]+)|(?<=WT)(?:\.\s+|,|-)([\\d.]+)|(?<=WALL\\sTHICKNESS)(?:\.\s+|,|-)([\\d.]+)|([\\d.]+)(?="\\s+WALL)

Examples

WALL 0.028, WALL-0.028, WT .028, WT-0.028, WALL THICKNESS 0.02823478547867, WALL THICKNESS-0.028, WALL THICKNESS,0.028, 0.028" WALL

CSV Importer

The following regular expressions are available within the IFC Importer.

Default Cuts Template

Default regular expression for the CutsTemplate virtual property.

Purpose

Sets the cuts template to use when during import. See the Cuts Template section for more information.

Supported patterns

CUTSDEFAULTNAME <cut name>

Expression

(?<=CUTSDEFAULTNAME\s+)(\w+)(?=\s*)|(?<=CUTSDEFAULTNAME\s+(?:'|"?))([\w|\s]+)(?='?|"?|\s+)

Examples

CUTSDEFAULTNAME TemplateName, CutsDefaultName " Template Name", cutsdefaultname ' Template Name'

IFC Importer

The following regular expressions are available within the IFC Importer.

Alternate Cuts Template

Default regular expression for the AltCutsTemplate virtual property

Purpose

Sets the alternate cuts template to use when during import. See the Cuts Template section for more information.

Supported patterns

ALTCUTSTEMPLATENAME <cut name>

Expression

(?<=ALTCUTSTEMPLATENAME\s+)(\w+)(?=\s*)|(?<=ALTCUTSTEMPLATENAME\s+(?:'|"?))([\w|\s]+)(?='?|"?|\s+)

Examples

ALTCUTSTEMPLATENAME TemplateName, AltCutsTemplateName " Template Name", altcutstemplatename ' Template Name'

Default Cuts Template

Default regular expression for the CutsTemplate virtual property.

Purpose

Sets the cuts template to use when during import. See the Cuts Template section for more information.

Supported patterns

CUTSTEMPLATENAME <cut name>

Expression

(?<=CUTSTEMPLATENAME\s+)(\w+)(?=\s*)|(?<=CUTSTEMPLATENAME\s+(?:'|"?))([\w|\s]+)(?='?|"?|\s+)

Examples

CUTSTEMPLATENAME TemplateName, CutsTemplateName " Template Name", cutstemplatename ' Template Name'

Hole Diameter

Default regular expression for the part HoleDiameter virtual property.

Purpose

Explicitly sets the diameter of a hole cut, overriding any other values that may have been found or calculated. Useful for explicitly specifying exact hole sizes for olets.

Supported patterns

- HOLEDIAMETER xx.xx
- Space, period, comma or dash as separators

Expression

(?<=HOLEDIAMETER)(?:\.\s+|,|-)([\d.]+)

Examples

HOLEDIAMETER 1.234, HoleDiameter 1.234in, holediameter 1.234mm

Miter End Cut

Default regular expression for the IFC Importer Miter virtual property. This value is formatted with two values as xxx/xxx (MateAngle/XAngleStart).

Purpose

Sets the miter end-cut parameters during import. Useful for specifying miter end-cut parameters for a pipe with no explicit connection specified in the import file.

Supported patterns

- ENDCUTMITER 37.5/90

Expression

(?<=ENDCUTMITER)(?:\s+)([\d.]+)/([\d.]+)(?=\s*)

Examples

ENDCUTMITER 37.5/90

PCF Importer

The following regular expressions are available within the PCF Importer.

Bevel Angle (deprecated)

Note: This functionality has been replaced by the cuts templates feature.

Default regular expression for the BevelAngle cut property.

When only one value is specified, it is applied to both end cuts of a pipe. When two values are specified then the first value is applied to the left end and the second value is applied to the right end.

Purpose

Legacy support for specifying bevel angles of end cuts. See the Cuts Template section for more information.

Supported patterns

- BEVELANG xx.xx,
- BEVELANG xx.xx xx.xx

Expression

(?<=BEVELANG)(?:\s+)([\d.]+)(?:\s+)([\d.]+)(?=\s*)|(?<=BEVELANG)(?:\s+)([\d.]+)(?=\s*)

Examples

BEVELANG 37.5, BEVELANG 37.5 45

Hole Cut Template

Default regular expression for the HoleCutsTemplate virtual property.

Purpose

Sets the hole cuts template to use when during import. See the Cuts Template section for more information.

Supported patterns

HOLECUTTEMPLATENAME<cut name>

Expression

(?<=HOLECUTTEMPLATENAME\s+)(\w+)(?=\s*)|(?<=HOLECUTTEMPLATENAME\s+(?:'|"?))([\w|\s]+)(?='?'|"?|\s+)

Examples

HOLECUTTEMPLATENAME TemplateName, HoleCutTemplateName " Template Name ",
holetemplatename ' Template Name'

Hole Diameter

Default regular expression for the part HoleDiameter virtual property.

Purpose

Explicitly sets the diameter of a hole cut, overriding any other values that may have been found or calculated. Useful for explicitly specifying exact hole sizes for olets.

Supported patterns

- HOLEDIAMETER xx.xx
- Space, period, comma or dash as separators

Expression

(?<=HOLEDIAMETER)(?:\.\s+|,|-)([\d.]+)

Examples

HOLEDIAMETER 1.234, HoleDiameter 1.234in, holediameter 1.234mm

Mate Pipe End Cut Template

Default regular expression for the MateEndCutsTemplate virtual property.

Purpose

Sets the mate pipe end cuts template to use when during import. See the Cuts Template section for more information.

Supported patterns

MATECUTTEMPLATENAME <cut name>

Expression

```
(?<=MATECUTTEMPLATENAME\s+)(\w+)(?=\s*)|(?<= MATECUTTEMPLATENAME\s+(?:'|"?)?)([\w|\s]+)(?='?|"?"|\s+)
```

Examples

```
MATECUTTEMPLATENAME TemplateName, MateCutTemplateName " Template Name ",  
matecuttemplatename ' Template Name'
```