



CATIA V5 Training
Foins

Product Knowledge Template

Version 5 Release 19
January 2009
EDU_CAT_EN_PKT_FI_V5R19

Copyright DASSAULT SYSTEMES

Instructor Notes:

About this course

Objectives of the course

Upon completion of this course you will be able to:

- Create and reuse Power Copies and User Defined Features.
- Create and reuse advanced instantiation features like Knowledge Pattern.
- Create Part and Assembly Templates and reuse them in a new context.

Targeted audience

CATIA V5 users

Prerequisites

Students attending this course should have knowlegde of CATIA V5 Fundamentals and Knowledgeware Basics.



Copyright DASSAULT SYSTEMES

Instructor Notes:

Table of Contents (1/3)

Product Knowledge Template Workbench Presentation	6
What are Templates?	7
Examples of Templates	9
Accessing the Workbench	10
User Interface	11
User Settings	12
Creating and Using PowerCopies	16
PowerCopy Presentation	17
Creating a PowerCopy	22
Saving a PowerCopy	29
Instantiating a PowerCopy	32
KeyWay Recap Exercise	37
To Sum Up	38
Creating and Using User Defined Features	39
User Defined Features: Presentation	40
Creating a User Defined Feature	47
Saving a User Defined Feature	58
Instantiating a User Defined Feature	60

Copyright DASSAULT SYSTEMES

Instructor Notes:

Table of Contents (2/3)

UDF Meta Inputs	70
User Defined Features Recap Exercises	76
Creating and Using Part and Assembly Templates	79
Presentation of Document Templates	80
Creating a Document Template	85
Saving a Document Template	91
Instantiating a Document Template	94
Document Templates Recap Exercises	99
Managing Standard Components	103
Introduction	104
Methodology Overview	105
Knowledge Environment Settings	106
About ARM Catalogs	107
Creating an ARM Catalog	108
Choosing the right standard component	111
Creating and Using Knowledge Pattern	116
Where do we use 'Knowledge Pattern' ?	117
Example of Knowledge Pattern	118

Copyright DASSAULT SYSTEMES

Instructor Notes:

Table of Contents (3/3)

The Mechanism of Knowledge Pattern	119
General Process - Knowledge Pattern – UDF Instantiation	121
General Process - Knowledge Pattern – Datum Creation	122
Knowledge Pattern – Script Explanations	131
Additional Information – Knowledge Pattern	136
Summary	139

Copyright DASSAULT SYSTEMES

Instructor Notes:

PKT Workbench Presentation

You will learn the concept of templates and about the user interface and specific settings of the Product Knowledge Template Workbench.

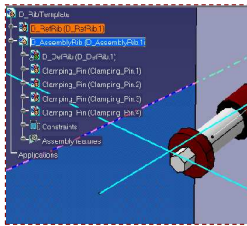


Copyright DASSAULT SYSTEMES

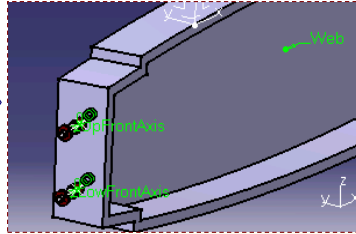
Instructor Notes:

What are Templates? (1/2)

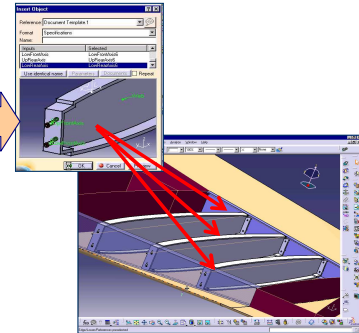
- ❏ A template is a user-defined reusable component which automates engineering tasks.
- ❏ A template is built 100% interactively by generalization of an existing design :
 - ❖ The generalization is performed by selecting the elements required in the template: documents, geometric elements, parameters, rules, etc.
 - ❖ CATIA V5 will automatically determine which inputs will be necessary to re-create these elements while instantiating the template (template inputs).



Interactive design of the model



Generalization



Multiple Instantiation

Copyright DASSAULT SYSTEMES

Instructor Notes:

What are Templates? (2/2)

- ▣ Users can create three types of templates:
 - ◆ **User Feature/PowerCopy:** A collection of CATIA features, including knowledge features, that can be reused in a part's design.
 - Allows customers to manipulate their own semantic objects in place of V5 standard objects.
 - Once instantiated, users get a black box (in case of UDF) behaving like any other feature with published parameters that can be edited.

 - ◆ **Part Template:** A part and its associated documents (drawing, analysis, process) can be reused inside products.
 - Once instantiated, the part is duplicated and you get an independent component which is adapted to the new context.

 - ◆ **Assembly Template:** A whole assembly and its associated documents can be reused inside products.
 - Once instantiated, the assembly is duplicated and the embedded parts can be independent or as a reference to the original one.

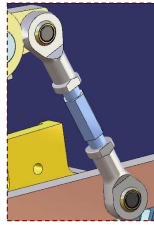
Copyright DASSAULT SYSTEMES

Instructor Notes:

Example of Templates

ASSEMBLY TEMPLATE

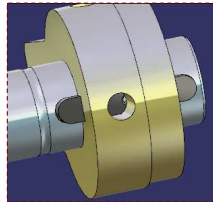
- ▣ Whole assembly duplication mechanism with associated documents
- ▣ Parts in Instance (copy) or Reference mode



Connecting Rod

PART TEMPLATE

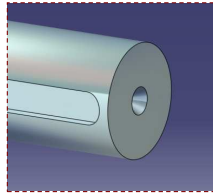
- ▣ Part duplication mechanism
 - ◆ Part number generation (New from)
- ▣ Associated documents can be part of the template definition (drawing, analysis)



Parallel Key

POWERCOPY / UDF

- ▣ Set of features including knowledge features
- ▣ Input selection
- ▣ Published parameters valuation
- ▣ Icon, Grab screen



Center Hole

Copyright DASSAULT SYSTEMES

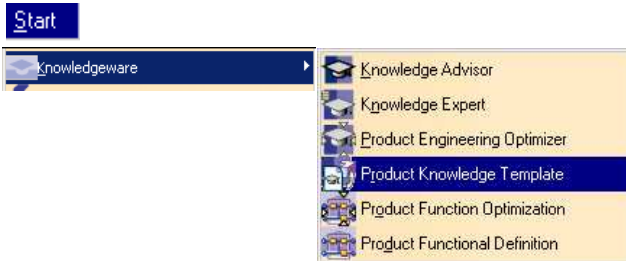
Instructor Notes:

Accessing the Workbench

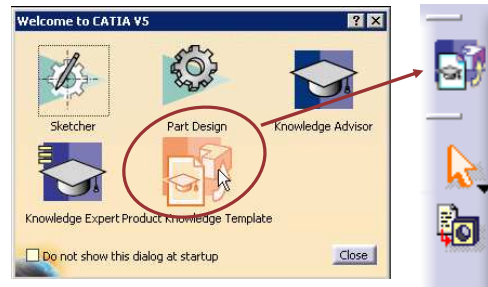
Access from:

- 1- The Start menu.
- 2- The Workbench Icon.

1-



2-



Use Tools/Customize+Start Menu to include Product Knowledge Template in your favourite workbenches.

Copyright DASSAULT SYSTEMES

Instructor Notes:

This is a series of Job aids to present the WorkBench during the introduction lesson.

This series is made of five Jobaids:

Accessing WorkBench

Exploring The User Interface

Checking User Settings

Seeing Terminology

Understanding the general process.

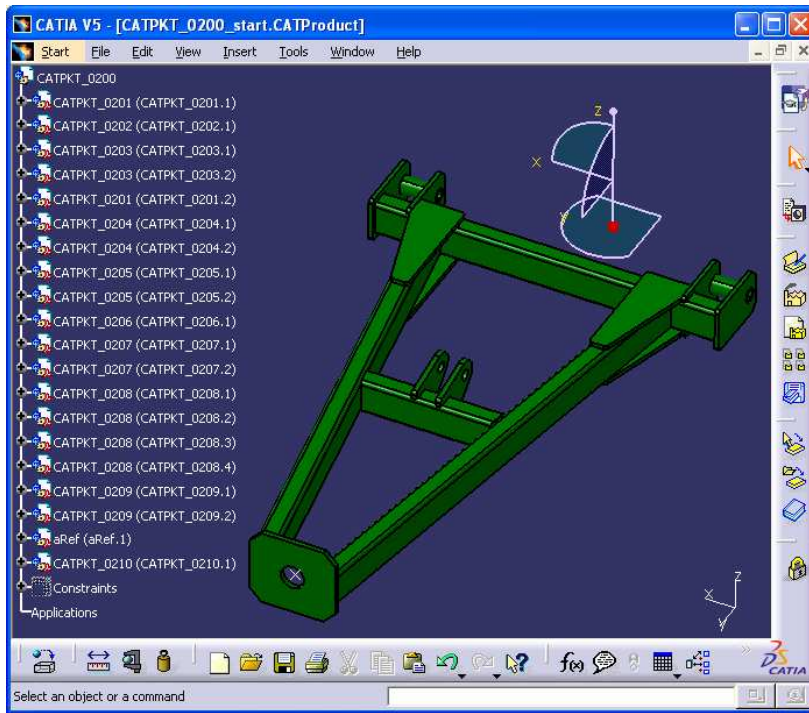
If you think that some are not necessary for the product you present, just remove them.

1/ Modify the following example

2/ Keep the 'Job Aids' icon on the top and the default title for each job aids

2/ No other recommendations besides the general ones

User Interface



- Create a Generative Script *
- Create a PowerCopy
- Create a User Feature
- Create a Document Template
- Create a Knowledge Pattern *
- Save in Catalog
- Instantiate from Selection
- Instantiate from Document
- Open Catalog
- Debug an UDF

Copyright DASSAULT SYSTEMES

* : Topics not covered in this course. Refer to CATIA documentation for information.

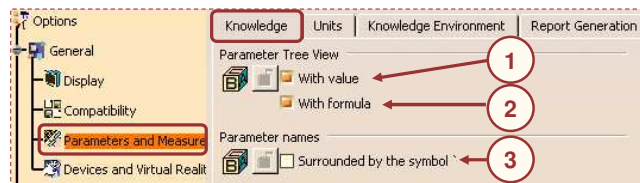
Instructor Notes:

User Settings (1/4)

Display General Settings:

In Tools> Options> General> Parameters and Measure, in the 'Knowledge' Tab, check the corresponding option if you need:

- (1) The value of the parameter to appear in the tree.
- (2) The formula driving the parameter to appear in the tree beside the parameter.
- (3) To work with non-latin characters. Otherwise, parameter names have to be renamed in latin characters when used.



Copyright DASSAULT SYSTEMES

Instructor Notes:

This is a series of Job aids to present the WorkBench during the introduction lesson.

This series is made of five Jobaids:

Accessing WorkBench

Exploring The User Interface

Checking User Settings

Seeing Terminology

Understanding the general process.

If you think that some are not necessary for the product you present, just remove them.

1/ Modify the following example

2/ Keep the 'Job Aids' icon on the top and the default title for each job aids

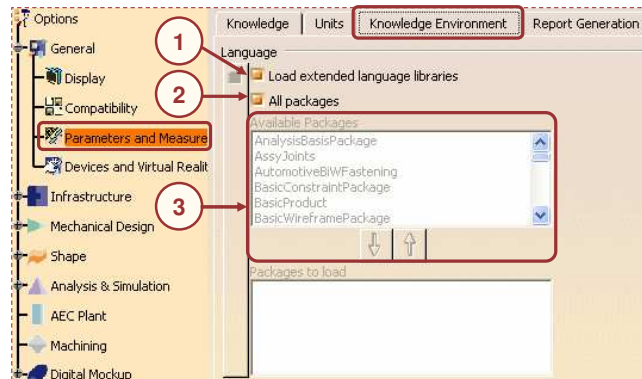
3/ No other recommendations besides the general ones

User Settings (2/4)

Language Settings:

In Tools> Options> General> Parameters and Measure in the 'Knowledge Environment' tab:

- (1) Check this option to have access to more language libraries. Which means more functions will be available for the Edition of Relations.
- (2) Check this button to load ALL the available libraries.
- (3) Otherwise, select libraries packages in the list and use the arrows to add or retrieve them to the list of libraries to be loaded.



Copyright DASSAULT SYSTEMES

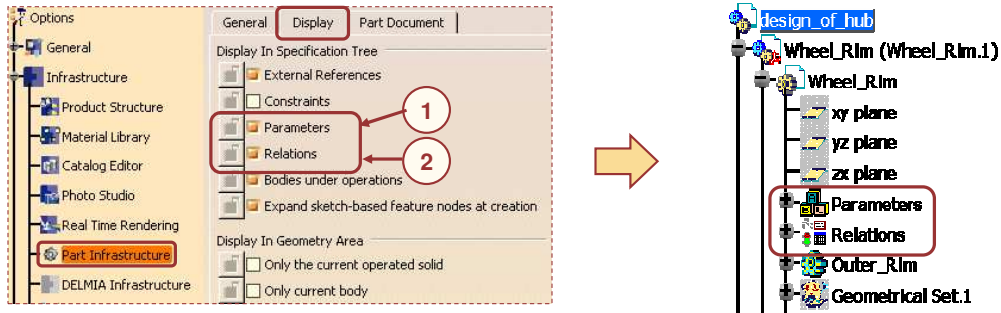
Instructor Notes:

User Settings (3/4)

Part Infrastructure Settings:

In Tools > Options > Infrastructure > Part Infrastructure, check the corresponding option if you need:

- (1) The parameters of the part to be displayed in the specifications tree.
- (2) The relations of the part to be displayed in the specifications tree.



Copyright DASSAULT SYSTEMES

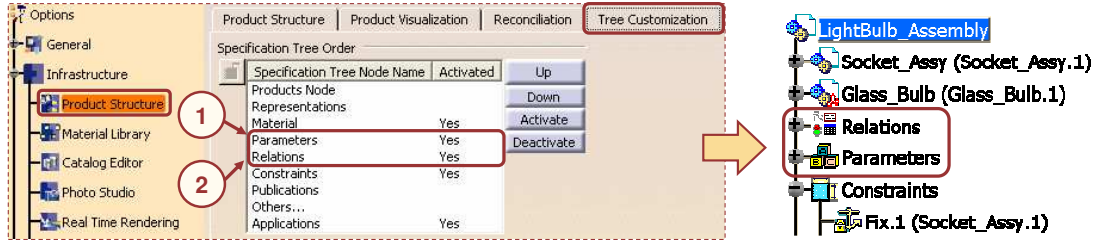
Instructor Notes:

User Settings (4/4)

Product Structure Settings:

In Tools> Options> Infrastructure> Product Structure, activate the following options if you need:







- (1) The parameters of the product to appear in the specifications tree.
- (2) The relations of the product to appear in the specifications tree.



Copyright DASSAULT SYSTEMES

Instructor Notes:

Creating and Using PowerCopies

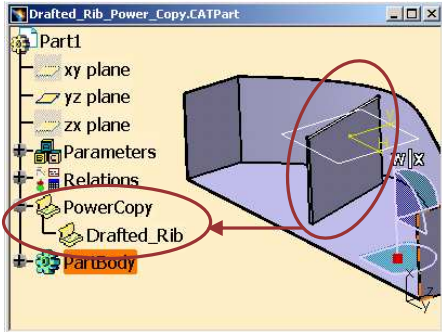
-  **PowerCopy Presentation**
-  **Creating a PowerCopy**
-  **Saving a PowerCopy**
-  **Instantiating a PowerCopy**
-  **KeyWay Recap Exercise**
-  **To Sum Up**

Copyright DASSAULT SYSTEMES

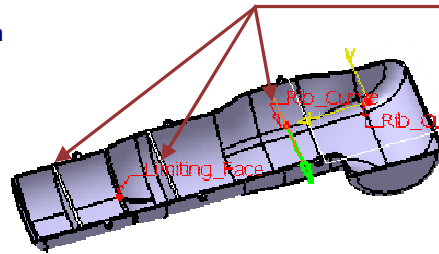
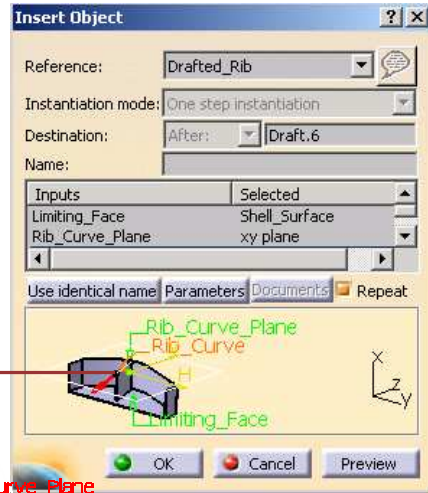
Instructor Notes:

PowerCopy Presentation

In this lesson, you will have an overview of 'PowerCopy' and the way in which it can be used.



PowerCopy definition



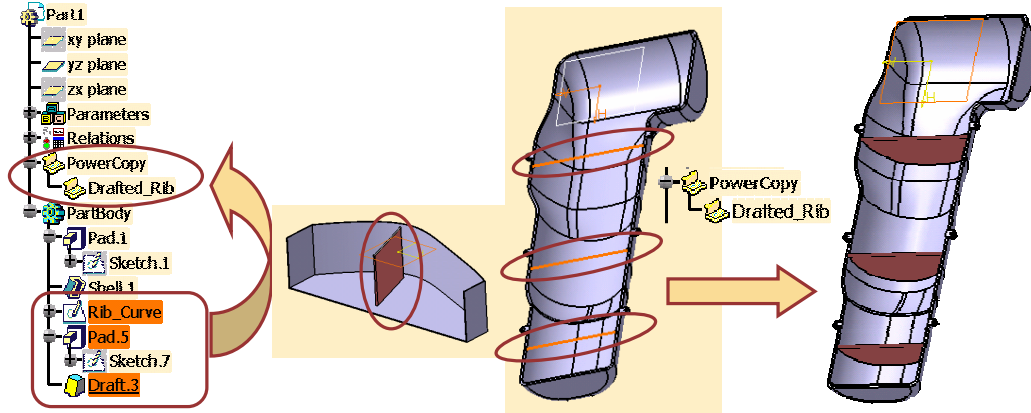
PowerCopy instantiation




Copyright DASSAULT SYSTEMES

Instructor Notes:

What is a PowerCopy?

- PowerCopy is a set of design features grouped together in order to be reproduced. It is a kind of advanced copying tool.
 - ◆ While defining it, you can specify the inputs that the user must provide.
 - ◆ During instantiation, you can customize it and insert it in the design of any part.

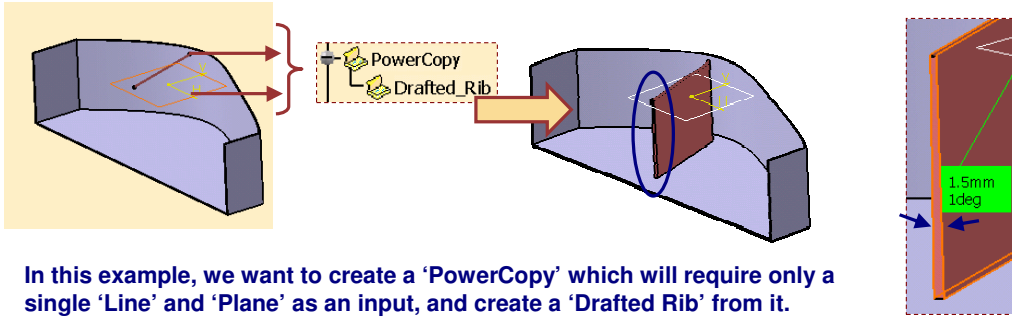


- PowerCopy tools are available in the Insert menu > Knowledge Templates of the following workbenches:
 - ◆ Part design 
 - ◆ GSD 
 - ◆ SheetMetal Design 

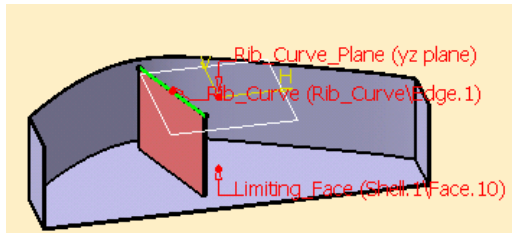
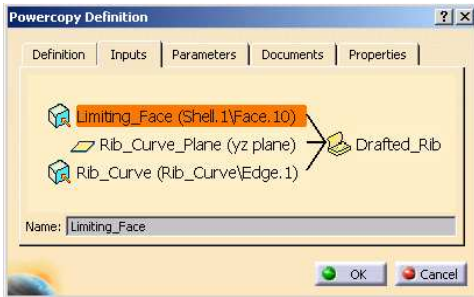
Copyright DASSAULT SYSTEMES

Instructor Notes:

Example of PowerCopy (1/3)



In this example, we want to create a 'PowerCopy' which will require only a single 'Line' and 'Plane' as an input, and create a 'Drafted Rib' from it.

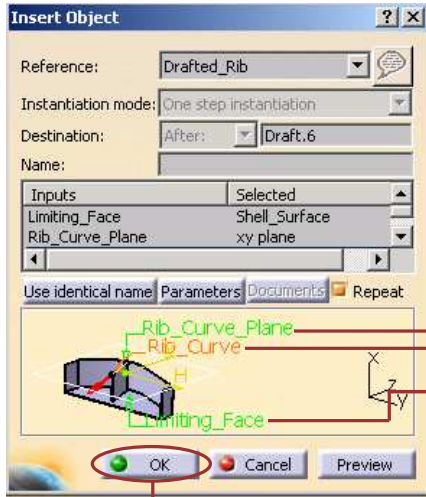


These are the inputs that the user will specify during the instantiation of the 'PowerCopy'.

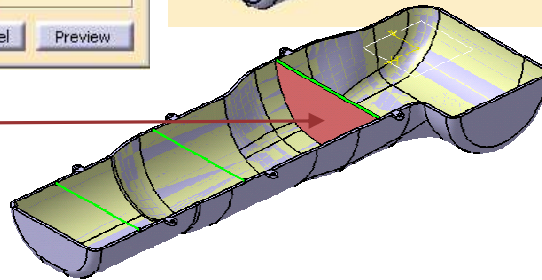
Copyright DASSAULT SYSTEMES

Instructor Notes:

Example of PowerCopy (2/3)



During the instantiation of the 'PowerCopy', the user has to select the inputs with respect to the destination part.

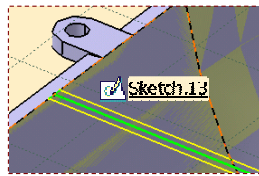
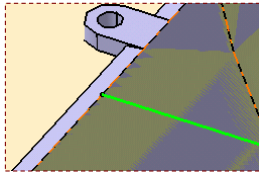


Copyright DASSAULT SYSTEMES

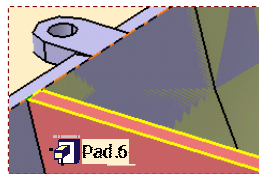
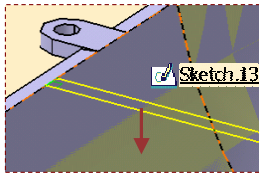
Instructor Notes:

Example of PowerCopy (3/3)

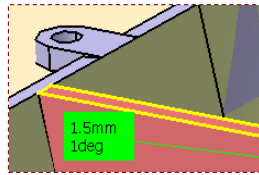
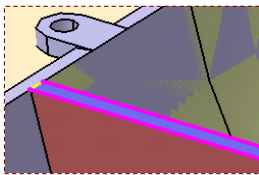
In this case, these are the geometries that the 'PowerCopy' feature creates automatically.



Creation of rectangular sketch from the selected rib line.



Extrusion of this sketch up to the selected 'Limiting Surface'.



Application of 'Draft' to the extruded faces.

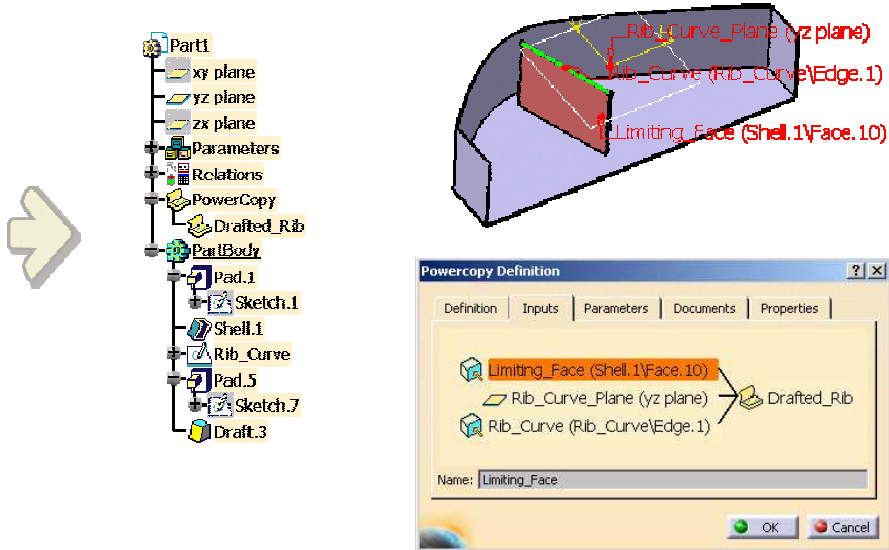
Copyright DASSAULT SYSTEMES

Thus, in this example you have seen how a PowerCopy feature can create a 'Drafted Rib' from a single 'Line' as input.

Instructor Notes:

Creating a PowerCopy

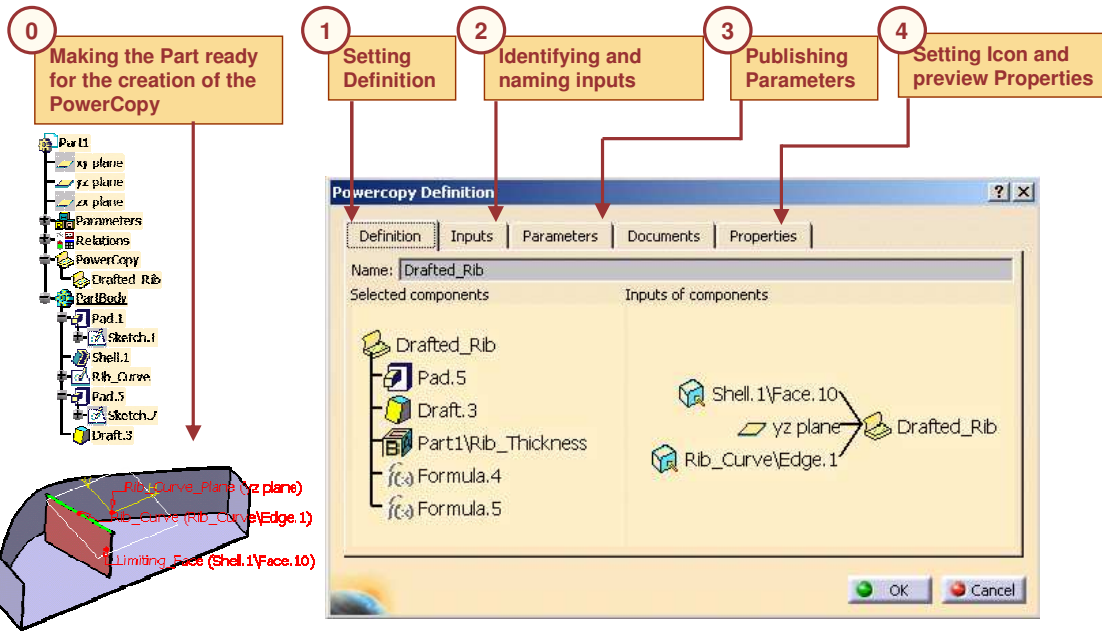
You will learn how to create a PowerCopy.



Instructor Notes:

Process for PowerCopy Creation

Creation of PowerCopy consists of the following steps:



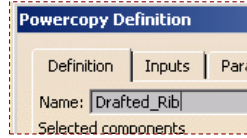
Instructor Notes:

How to Create a PowerCopy (1/5)

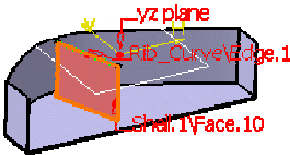
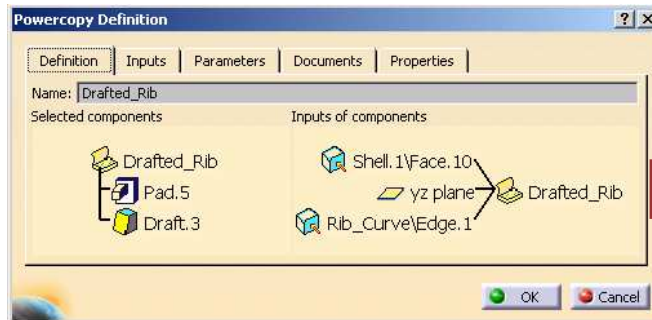
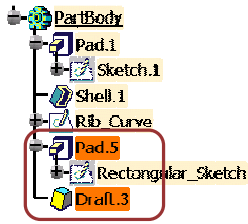
Once you have the right geometry in your CATPart, you can create the PowerCopy.

1a Select PowerCopy from the menu. (Insert > Knowledge Templates > PowerCopy)

1b Type the name of the PowerCopy in the 'Definition Tab' of the 'PowerCopy Definition' dialog box.



1c From the specification tree, select the features that will make your 'PowerCopy'.



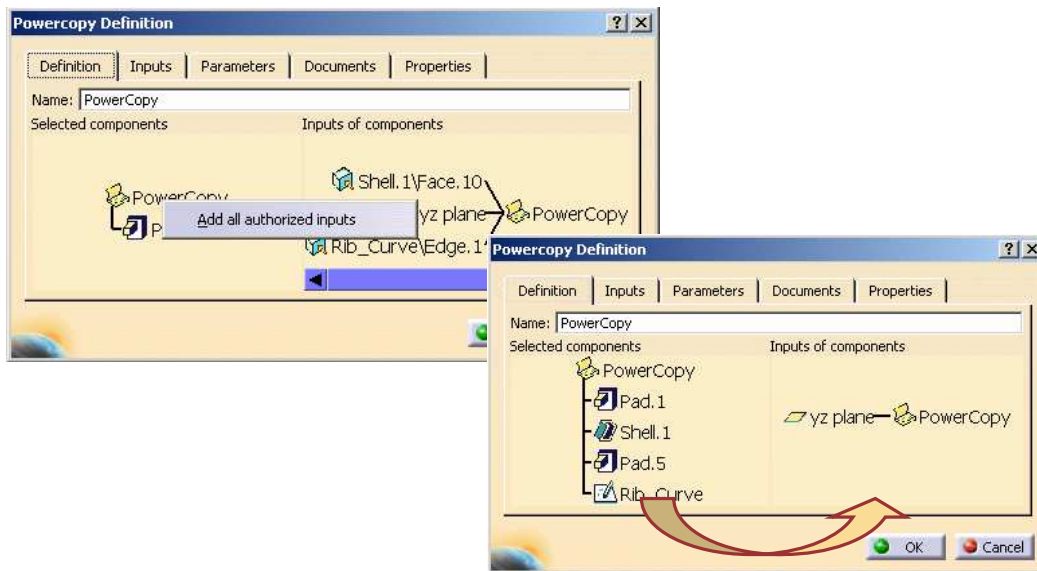
On selecting the features, the 'Inputs of components' are identified. These depend upon the features that you select to make your PowerCopy.

Copyright DASSAULT SYSTEMES

Instructor Notes:

How to Create a PowerCopy (2/5)

TIP: The contextual menu 'Add all authorized inputs' allows you to select all the possible components that can be created using minimum number of inputs.



Copyright DASSAULT SYSTEMES



Note that this tip is not used in this scenario. However, it can be used to select all the features and to later move the features from the 'Selected Components' field to 'Inputs of Components' field.

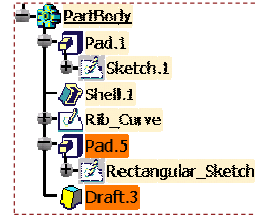
Instructor Notes:

How to Create a PowerCopy (3/5)

After selecting the features that make the PowerCopy, you can give names to the geometric inputs. During instantiation, the user will be prompted to select the geometries based on these new names.

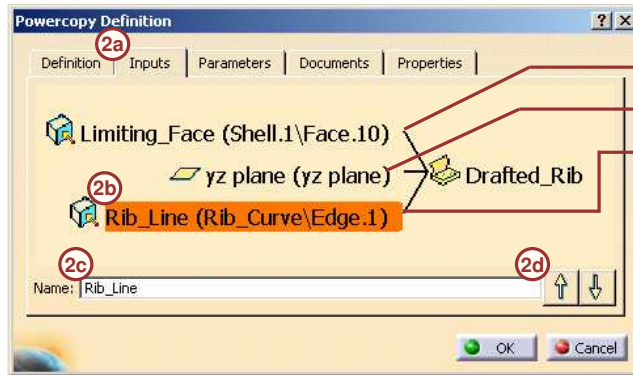
In our case there are three inputs:

- A. The edge (Edge.1) from 'Rib_Sketch' -> Using this sketch, the PowerCopy creates the 'Rectangular Sketch'.
- B. The YZ plane on which the 'Rib_Curve' has been created.
- C. The shell face (Face.10) up to which the 'Pad.5' was extruded.



Let us give new names to these inputs from instantiation point of view.

- 2a Select the Inputs tab
- 2b Select the input to be renamed
- 2c Type a new name for the input
- 2d Using the arrow keys reorder the inputs, if required



New Name:
 Limiting_Face
 Rib_Curve_Plane
 Rib_Curve

Reordering the inputs is sometimes required for displaying the inputs in a specific order in the PowerCopy instantiation dialog box.

Instructor Notes:

How to Create a PowerCopy (4/5)

After renaming the geometric inputs you can publish the parameters. During instantiation, the user can specify values for these published parameters.

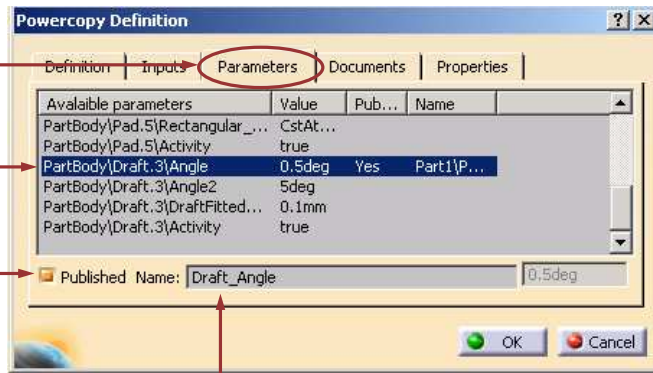
To publish the parameters,

3a Select Parameters tab

3b Select the parameter

3c Check the 'Published' option

3d If necessary, rename the parameter



Copyright DASSAULT SYSTEMES



Note that it will be easier for you to recognize the parameters if you have already renamed them with the knowledgware tools. [f(x)]

Instructor Notes:

How to Create a PowerCopy (5/5)

Once the parameters are published, you can select the icon for your PowerCopy and make a screen grab to create a preview of your PowerCopy for catalogs.

4a Select 'Properties' tab

4b Select any icon from the available list

4c Prepare the CATPart window for the screen grab

4d Click 'Grab screen' to make a screen grab, and click OK to validate

The image shows two overlapping windows. The 'Powercopy Definition' window has tabs for 'Definition', 'Inputs', 'Parameters', 'Documents', and 'Properties'. The 'Properties' tab is active, showing an 'Icon' field with a dropdown menu and a 'Preview' section containing 'Grab screen' and 'Remove preview' buttons. The 'Icons Browser' window is open, displaying a grid of 141 icons. Red arrows and callouts indicate the steps: 4a points to the 'Properties' tab, 4b points to the icon selection area, 4c points to the 'Grab screen' button, and 4d points to the 'Grab screen' button.

Copyright DASSAULT SYSTEMES



To prepare the screen grab, you can remove the tree and compass from the window and get the correct zoom and orientation.

Instructor Notes:

Saving a PowerCopy

You will learn how to save the PowerCopy in a catalog.



Copyright DASSAULT SYSTEMES

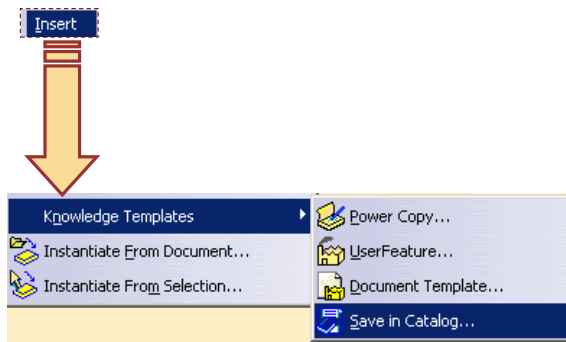
Instructor Notes:

Saving a PowerCopy

If you do not save the CATPart containing your PowerCopy, you will not be able to instantiate the PowerCopy.

You can save the PowerCopy in a new catalog and also in an existing catalog.

You can also update a catalog which makes reference to the PowerCopies of your CATPart.

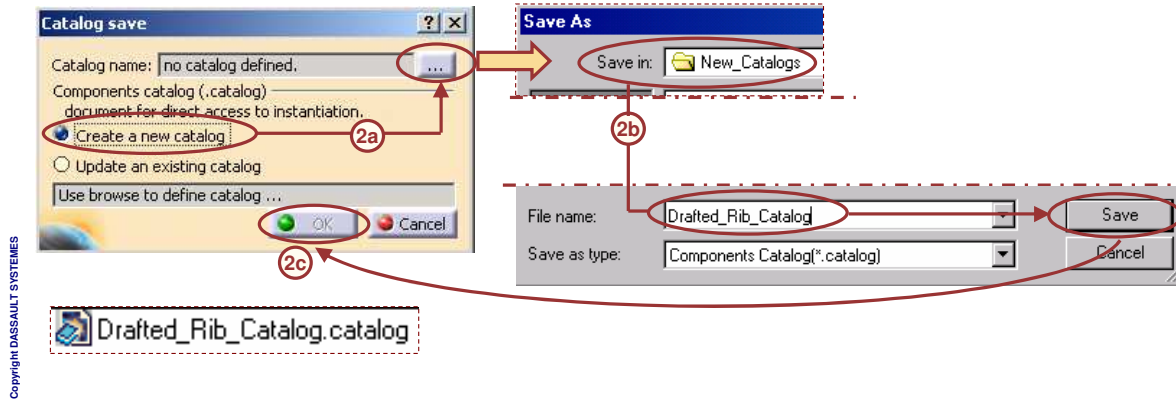


Copyright DASSAULT SYSTEMES

Instructor Notes:

How to Save a PowerCopy in a Catalog

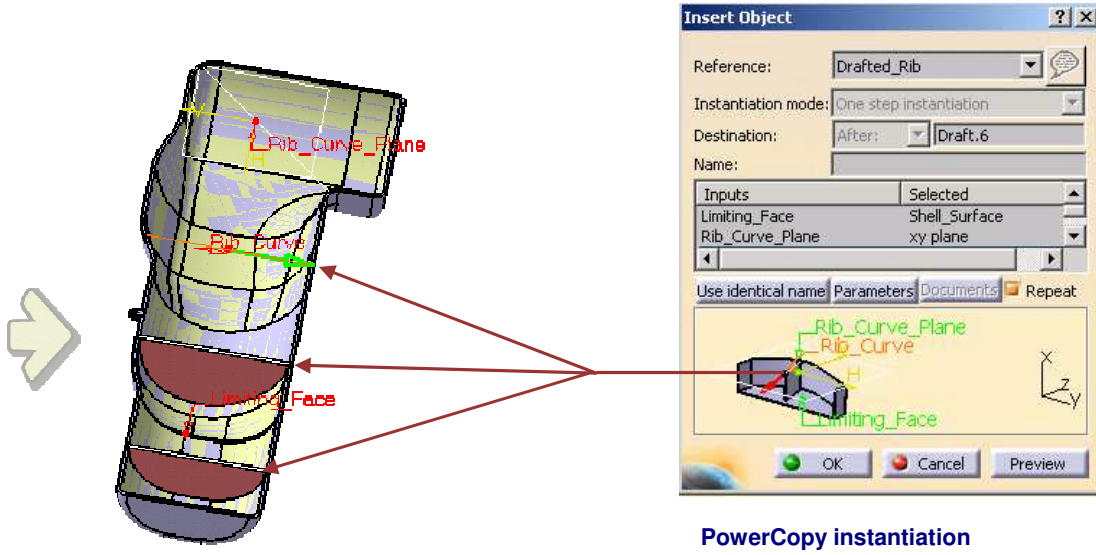
- 0 Save the CATPart containing your PowerCopy.
- 1 From the menu, select – Insert > Knowledge Templates > Save in Catalog.
- 2a Select the 'Create a new catalog' option and click the browse button (. . .) to define the path for new catalog.
- 2b Select the correct path, type the new name of the catalog and click Save. (The OK button of the 'Catalog save' dialog box will now be active)
- 2c Now click OK to the 'Catalog save' dialog box.



Instructor Notes:

Instantiating a PowerCopy

You will learn how to instantiate a PowerCopy differently at different places by varying the geometric inputs and the parameters while instantiating.



Copyright DASSAULT SYSTEMES

Instructor Notes:

How to Instantiate a PowerCopy (1/4)

The first step of PowerCopy instantiation is accessing the PowerCopy.

You can access it:

- a) From the CATPart file containing it.
- b) From a catalog having its reference.



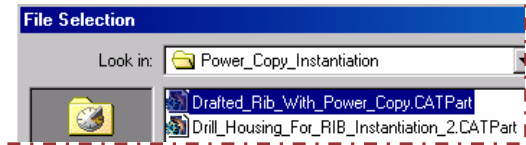
You can also use a VB macro to instantiate the PowerCopy. Refer CATIA online documentation for more information.

Before proceeding, please save all the CATIA documents that are attached to this screen to a local folder.

0 Open the CATPart in which you want to instantiate the PowerCopy.

1 From the menu, select:
Insert > Instantiate From Document OR

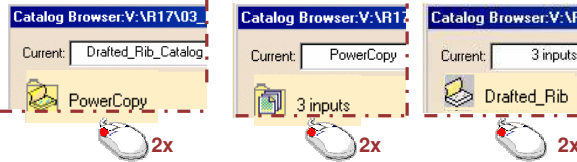
2 Select the CATPart file which contains your PowerCopy. OR



1 Click on Catalog Browser and browse for the catalog.



2 After opening the catalog, double-click on 'PowerCopy', then on '3 inputs' and finally on 'Drafted_Rib' to open the instantiation dialog.



Copyright DASSAULT SYSTEMES

Instructor Notes:

How to Instantiate a PowerCopy (2/4)

The second step of instantiation is selecting the geometric inputs of the PowerCopy.

- 3a Select the geometric inputs of the PowerCopy as shown. For this example, select the 'Limiting Surface' and 'Rib_Curve_Plane' as shown.

The 'Insert Object' dialog box is shown with the following settings:

- Reference: Drafted_Rib
- Instantiation mode: One step instantiation
- Destination: Alter Sketch.1
- Name: (empty)

Inputs	Selected
Limiting_Face	Shell_Surface
Rib_Curve_Plane	xy plane
Rib_Curve	Edge

Use identical name Parameters Documents Repeat

OK Cancel Preview

Now the first two inputs remain the same for all the three green 'Rib_Curves'. So in this case, you can use the 'Repeat' option.

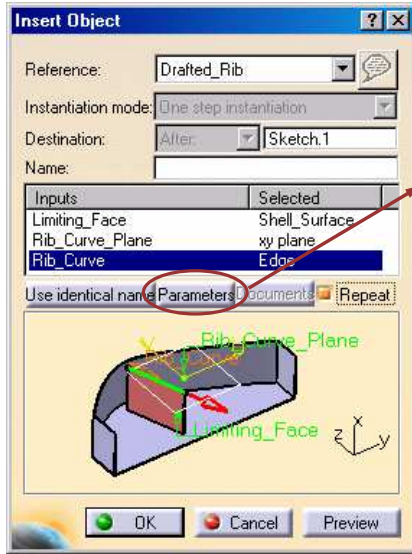
- 3b Select the 'Repeat' option, select any one of the three green lines and click OK. Repeat the same process for any one of the remaining two green rib lines.

Instructor Notes:

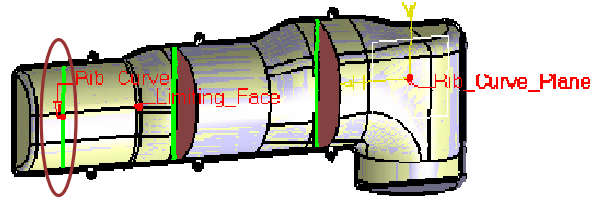
How to Instantiate a PowerCopy (3/4)

You can also change the values of the parameters that you have published during the PowerCopy creation.

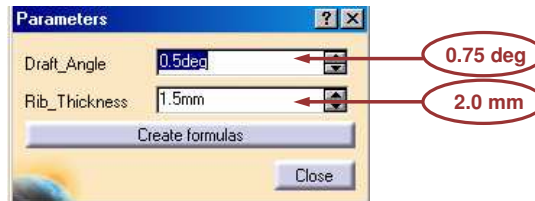
In this example, we will enter different values for the last rib line.



4a Select the remaining Rib_Curve and click the 'Parameters' button.



4b Enter the values for the parameters as shown and close the 'Parameters' dialog box.



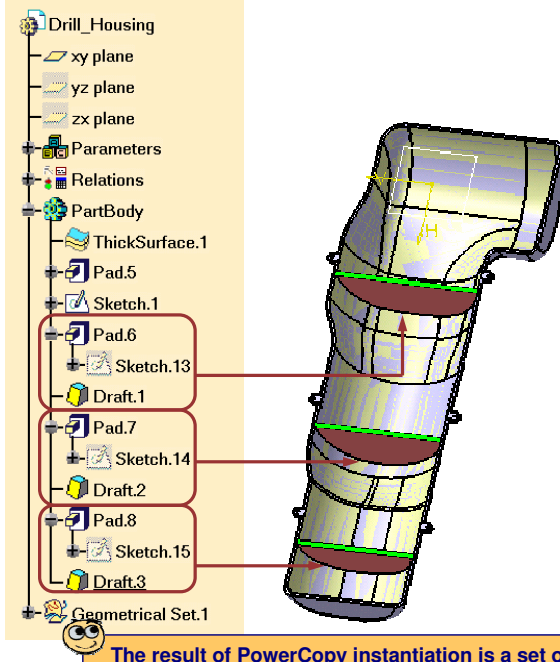
4b Click OK on the 'Insert Object' dialog box to instantiate the last rib, and then click 'Cancel' to dismiss it.

Copyright DASSAULT SYSTEMES

Instructor Notes:

How to Instantiate a PowerCopy (4/4)

The result of the PowerCopy instantiation is inserted after the “in work object”.



The result of PowerCopy instantiation is a set of editable features. They are not linked to the original features of the PowerCopy CATPart.

Instructor Notes:

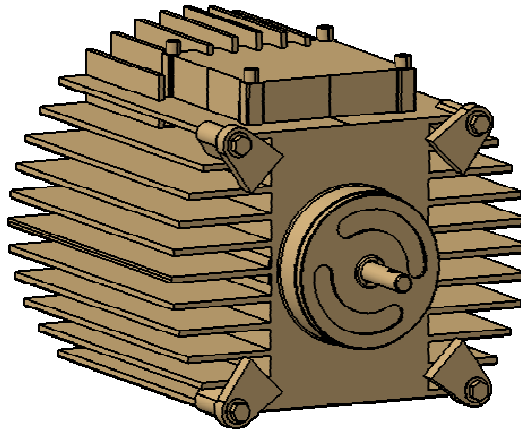
Key-Way

PowerCopy Recap Exercise



The objective of this exercise is to create a PowerCopy of a key-way which will always be compliant with a specific standard.

Prerequisites:
Knowledge Advisor



Copyright DASSAULT SYSTEMES

Instructor Notes:

To Sum Up ...

You have learned:

What is a PowerCopy

- ◆ A PowerCopy is a set of design features grouped together to be reproduced. It is an advanced copy tool. PowerCopy tools are available in the Insert menu in Part design, Wireframe and surface, and Sheet metal design workbenches.



How to create a PowerCopy

- ◆ During creation, you have to set the definition, identify and name the inputs, publish the parameters, choose an icon and preview.

How to save a PowerCopy

- ◆ Saving a PowerCopy is necessary. If not saved, a PowerCopy can never be instantiated. This can be done through Insert menu > Advanced replication tools > Save in catalog.







How to instantiate a PowerCopy

- ◆ For instantiation, you have to first select a previously created PowerCopy. This can be done in two ways. The first way is through a catalog, and the second way is from Insert menu > Instantiate from document.

Copyright DASSAULT SYSTEMES

Instructor Notes:

Creating and Using User Defined Features

-  **User Defined Features: Presentation**
-  **Creating a User Defined Feature**
-  **Saving a User Defined Feature**
-  **Instantiating a User Defined Feature**
-  **UDF Meta Inputs**
-  **User Defined Features Recap Exercises**

Copyright DASSAULT SYSTEMES

Instructor Notes:

User Defined Features: Presentation

You will learn what are the benefits of advanced replication tools called User Defined Features.

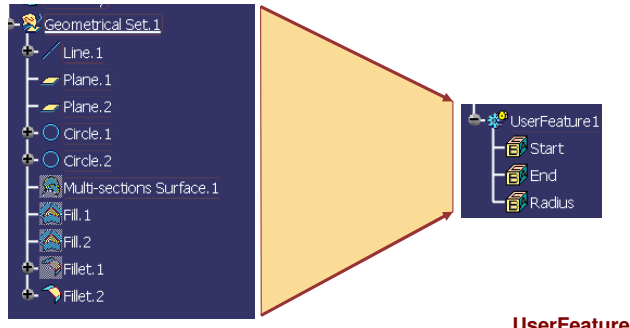


Copyright DASSAULT SYSTEMES

Instructor Notes:

User Defined Features vs PowerCopies

- User Defined Features (also called UDF) are similar to PowerCopies (at definition stage).
- But when instantiated, you get only one feature like any other V5 feature.



PowerCopies are mainly used to accelerate the dramatically design generation

User Defined Features allow the customer to manipulate their own semantic objects in the place of V5 standard objects

It introduces a true component-based approach for building designs

Copyright DASSAULT SYSTEMES

Instructor Notes:

What is a User Defined Feature? (1/2)

- A User Defined Feature is a template that works at the part level. From a collection of features (geometries, literals, formulas, constraints, etc.), the user can create his/her own feature. The result is a Part Design feature or a Wireframe and Surface feature that can be reused in the design of another part. The created feature can be saved in a catalog.

- A User Defined Feature:
 - ◆ Allows you to create applicative features
 - ◆ Allows you to hide design specifications and preserve confidentiality (for instance, to sub-contractors)

- The User Defined features (like a line for Drafting or a check for Knowledge Advisor) are open and shareable objects. This capability significantly increases the potential application of the user defined features, since it enables you to:
 - ◆ Find the user defined features by attributes
 - ◆ Generate the user defined features with the scripting language to simplify the process of creating scripts
 - ◆ Define the expert rules that work on user defined features with Knowledge Expert
 - ◆ Use the user defined features in Knowledge Advisor reactions
 - ◆ Develop the CAA functions based on the user defined variables

Copyright DASSAULT SYSTEMES

Instructor Notes:

What is a User Defined Feature? (2/2)

- ▣ A UserFeature is a design feature made up of a group of other design features.
 - ◆ You can edit it (set contained features, entries, previews ...)
 - ◆ You can instantiate and customize it in the design of any part
 - ◆ Instance of a UserFeature is a black box (users do not have any access to its contents)

- ▣ The UserFeature tools are available in the Insert menu (Knowledge Templates) of the following workbenches:
 - ◆ Part Design 
 - ◆ Generative Sheetmetal Design 
 - ◆ Generative Shape Design 

Copyright DASSAULT SYSTEMES

Instructor Notes:

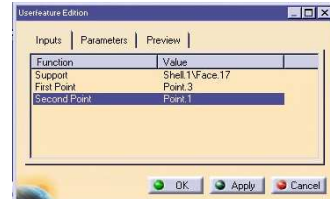
User Defined Features Benefits

- Simplification of designs and better evolutivity**
 - ◆ The complexity in terms of the number of features used in a model is reduced
 - ◆ The component generated as a UserFeature is easier to understand and modify (edit...)
 - ◆ Designers do not lose time in dealing much with the geometry

- Insurance of best practices usage**
 - ◆ Rules embedded in the UserFeatures cannot be violated within a company

- Intellectual Property Protection**
 - ◆ While exchanging a model using the UserFeatures, it is impossible to understand what is inside for the receiver (even if he can update it)

- Object-oriented designs**
 - ◆ UserFeatures inherit from features standard behaviors
 - ◆ UserFeatures are recognized by Knowledgeware as any other V5 object
 - ◆ Behaviors can be added to a userfeature (thanks to reactions)

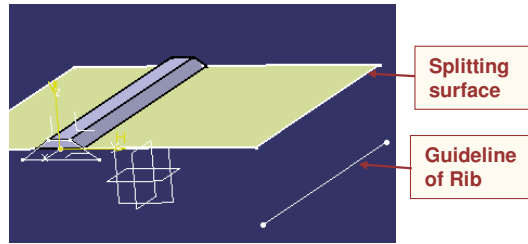


Instructor Notes:

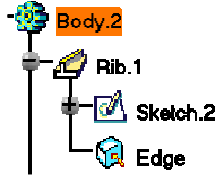
Example of User Defined Feature (1/2)

The UserFeature that we are going to create and instantiate is made up of...

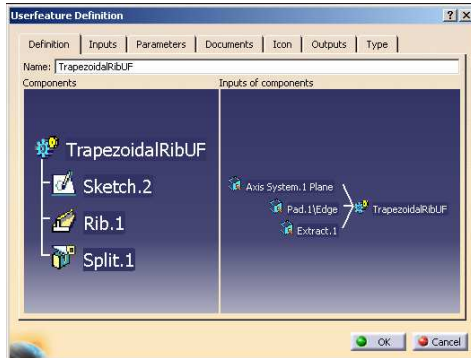
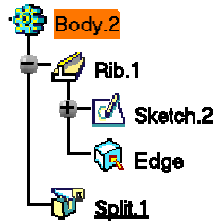
...one sketch (lying on a face of an axis system)...



... one rib, based on this sketch and guided by a line inside an external body ...



... and one split of the body by a surface



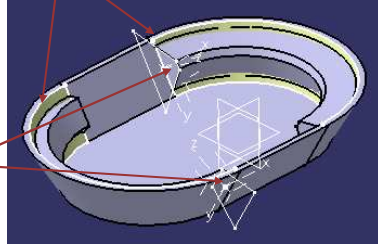
Copyright DASSAULT SYSTEMES

Instructor Notes:

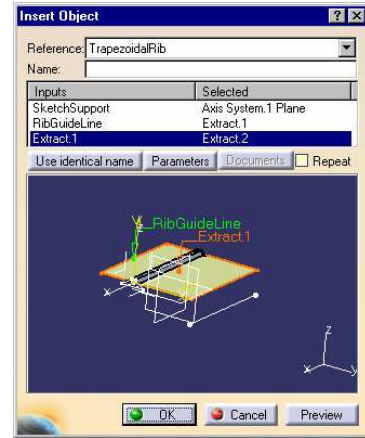
Example of User Defined Feature (2/2)

- While instantiating the UserFeature, you will be able to customize:

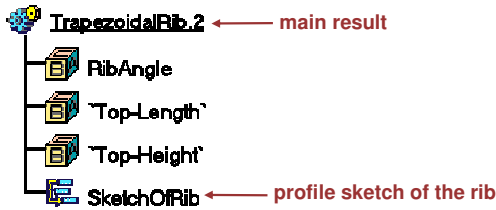
...The Inputs of the geometric data...



... The values of Top-Length, Top-Height and Rib-Angle parameters...



- Instance of the userfeature will give out two geometric outputs: the main result and the profile sketch of the rib



Copyright DASSAULT SYSTEMES

Instructor Notes:

Creating a User Defined Feature

You will learn how to group the existing features in a black box in order to reuse them in another context.



Copyright DASSAULT SYSTEMES

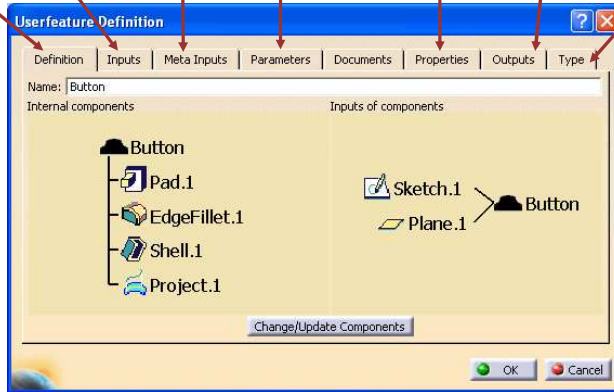
Instructor Notes:



Process for UDF Creation

User Defined Feature creation process includes several steps:

- 1 Selecting the existing features
- 2 Naming the input geometry
- 3 Optional: Selecting the meta inputs
- 4 Optional: Selecting and naming the public parameters
- 5 Optional: Selecting the icon and creating the preview
- 6 Optional: Selecting the outputs
- 7 Optional: Creating a new Type to define the UDF



Copyright DASSAULT SYSTEMES

Instructor Notes:

Creating a User Defined Feature (1/9)

Open the User Defined Feature Definition panel by clicking on the Create User Feature icon in the Product Knowledge Templates workbench.



The following panel appears.

The Definition tab allows you to key in the name of the UserFeature and see the features selected for its definition.

Select the features to be included in the UserFeature (the ones that will be reproduced during the UserFeature instantiation).

Type here the name of the UserFeature

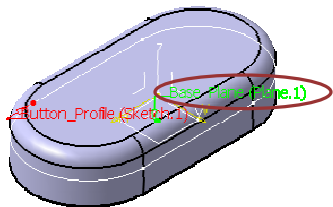
Inputs are displayed in the 3D

Copyright DASSAULT SYSTEMES

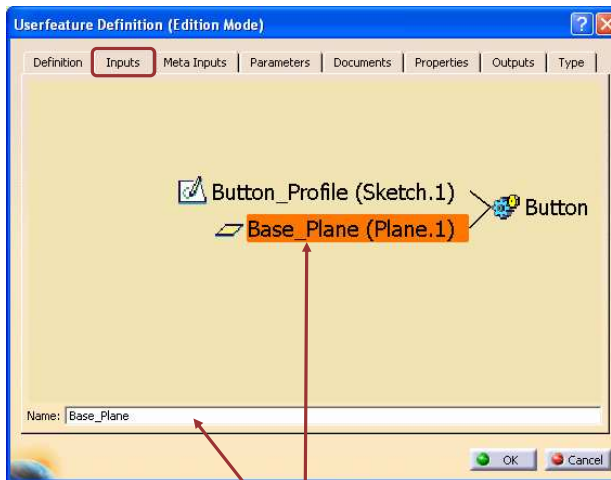
Instructor Notes:

Creating a User Defined Feature (2/9)

The Inputs tab allows you to see and rename the required geometric inputs for the instantiation of the userfeature.



While editing the inputs tab, required geometric inputs are shown in the geometry.



Name field:
To rename an input, select it, select the name field and key in the new name.

Copyright DASSAULT SYSTEMES

Instructor Notes:

Creating a User Defined Feature (3/9)

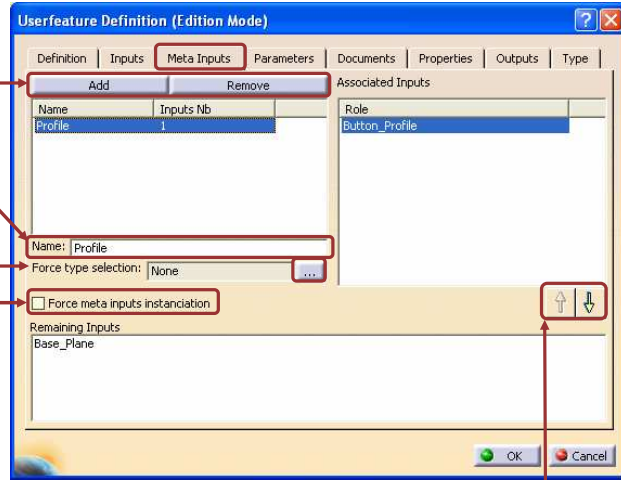
The Meta Inputs tab allows you to define the meta inputs and their association with real inputs. It also allows you to optionally associate a type with each meta input.

The Add/Remove buttons enable you to add or remove the meta inputs.

The Name field enables you to enter the name assigned to the meta input.

The ... button enables you to associate a type to the meta input.

The Force meta inputs instantiation option enables you to decide if you want the user to select the instantiation mode. If checked, only the Meta inputs instantiation mode will be available in the Insert Object dialog box. If unchecked, the user will be able to choose the instantiation mode he wants to use i.e. the Meta inputs instantiation mode or the Meta inputs normal instantiation mode.



The arrow buttons enable you to remove or add the inputs from the list of inputs that makes the meta input.

Copyright DASSAULT SYSTEMES

Instructor Notes:

Creating a User Defined Feature (4/9)

The Parameters tab allows you to see all the parameters participating in the definition of the Userfeature and allows you to make them Published for instantiation.

Published status column

Published name column

Available parameters	Value	Published	Name
Pad.1\FirstLimit\Height	4mm	Yes	Height
Pad.1\SecondLimit\Length	0mm		
Pad.1\ThickThin1	1mm		
Pad.1\ThickThin2	0mm		
Pad.1\Activity	true		
EdgeFillet.1\CstEdgeRibbon.1.Radius	1mm	Yes	Radius
EdgeFillet.1\Activity	true		
Shell.1\Thickness	0.5mm	Yes	Thickness
Shell.1\OutsideThickness	0mm		
Shell.1\Activity	true		
Project.1\Activity	true		

Check this button to make the selected parameter public.

Type here an explicit name for the published parameter.

Published Name: Radius

Copyright DASSAULT SYSTEMES



Parameters have the same names as in the formulas editor, if you want to recognize them easily, rename them with knowledgware common tools.

Instructor Notes:

Creating a User Defined Feature (5/9)

The Documents tab shows the complete path and role of design tables referenced by an element included in the UserFeature.

This tab does not exhibit any document because only the design tables belonging to the selected object are displayed. While instantiating or editing the UserFeature, you will be able to change the document pointed by the internal design table.



Copyright DASSAULT SYSTEMES

Instructor Notes:

Creating a User Defined Feature (6/9)

The Properties tab allows you to modify the icon identifying the UserFeature in the specification tree.

A subset of icons is available from the Icon choice.

Click on ... to open the Icon Browser. You will have the choice between all icons that are loaded in your CATIA session.

Click on Grab screen button to capture an image of the UserFeature that will be stored with its definition.

This preview will be useful while referring to a UserFeature in a catalog...

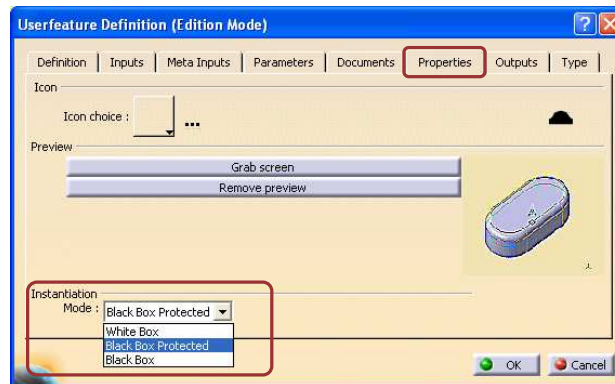
The image shows the 'Userfeature Definition' dialog box with the 'Properties' tab selected. The 'Icon choice' field has a dropdown arrow. A callout points to this arrow, stating that clicking it opens the 'Icon Browser'. The 'Icon Browser' window is shown to the right, displaying a grid of icons. Another callout points to the 'Grab screen' button in the 'Userfeature Definition' dialog, explaining its function. A third callout points to the 'Preview' area, which shows a 3D model of a part and a 'Grab screen' button. A yellow callout box with a smiley face icon explains that this preview is useful for referring to a UserFeature in a catalog. The 'Instantiation Mode' is set to 'Black Box'. The 'OK' and 'Cancel' buttons are at the bottom.

Copyright DASSAULT SYSTEMES

Instructor Notes:

Creating a User Defined Feature (7/9)

The Instantiation Mode combo box list enables you to choose the view that will be created at instantiation.



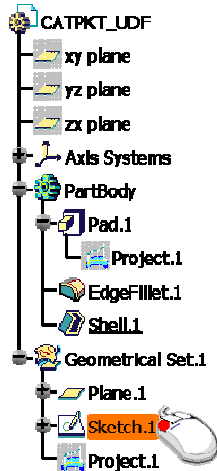
- Select the **White Box** mode if you want the end-user to display the UserFeature internals.
- Select the **Black Box** mode if you want the end-user to be able to lock and unlock the UserFeature instance.
- Select the **Black Box Protected** mode if you do not want the end-user to access the internals. This mode is the standard User Defined Feature view.

Copyright DASSAULT SYSTEMES

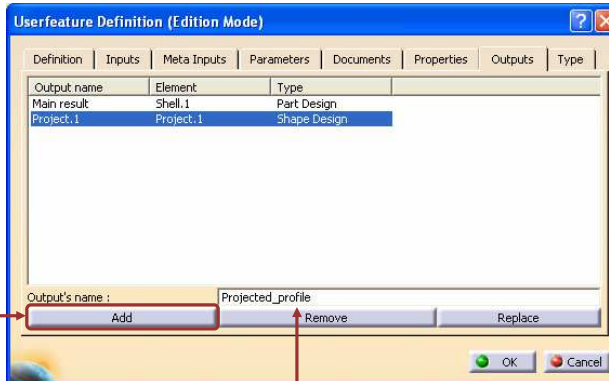
Instructor Notes:

Creating a User Defined Feature (8/9)

The Outputs tab allows you to select geometric outputs other than the Main result for instantiation.



Copyright DASSAULT SYSTEMES



Click on Add button...

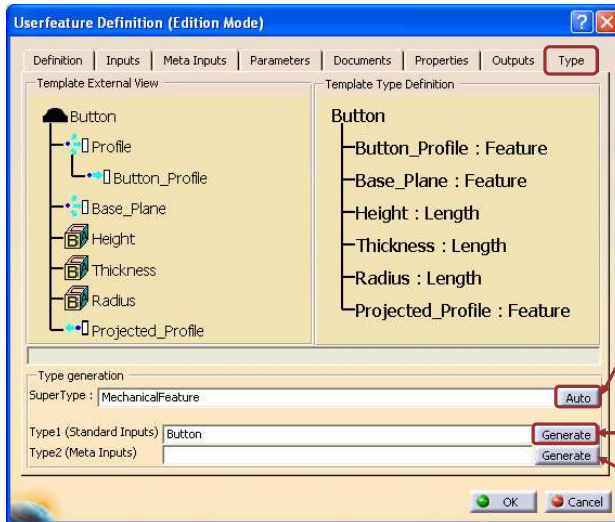
...and select in the tree the element of the UserFeature you want to recover with instantiation

Key in a new output name after output selection.

Instructor Notes:

Creating a User Defined Feature (9/9)

The Type tab provides you a way to associate a Type with a UserFeature. This can be used in search operations, expert checks, and to instantiate UDFs using Knowledge Pattern Functions.



First click the Auto button to have the super type automatically displayed by the application

Enter the name of the type that you want to assign to the UserFeature (button in this example) and click the Generate button.

Type2 allows you to define a type for the meta input.

Copyright DASSAULT SYSTEMES



If you want to reuse the generated type in another CATIA session, save the CATGScript file in the Directory indicated in the Reference Directory for Types field (see Tools>Options>Parameters and Measure>Knowledge Environment tab).

Instructor Notes:

Saving a User Defined Feature

You will learn how to store a User Defined Feature in a catalog document in order to make it available for other users.



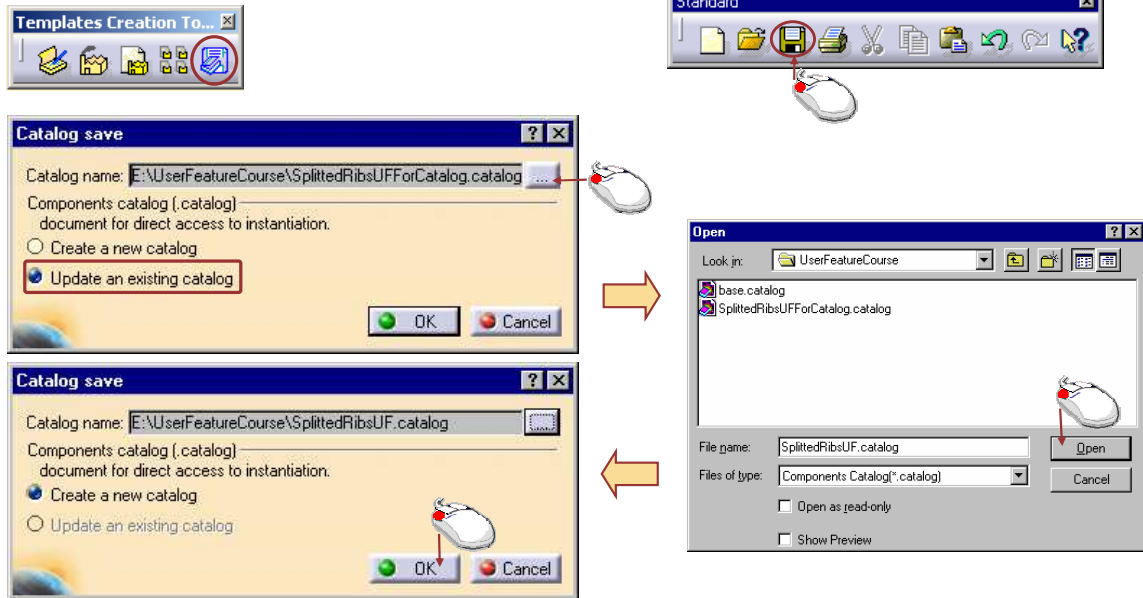
Copyright DASSAULT SYSTEMES

Instructor Notes:



Saving a User Defined Feature

You are to save the CATPart file containing the UserFeature, but you can also reference all the userfeatures of the edited CATPart in a catalog by using one of the UserFeature Tools.



Copyright DASSAULT SYSTEMES

Instructor Notes:

Instantiating a User Defined Feature

You will learn how to import an existing User Defined Feature from a catalog in your document, and how to make it fit to the specifications of your design.



Copyright DASSAULT SYSTEMES

Instructor Notes:



Instantiating a User Defined Feature (1/3)

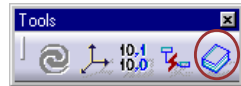
There are different ways to launch the instantiation of a UserFeature.



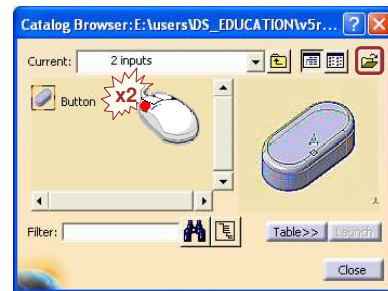
Select the **Instantiate From Document** option in CATIA Insert menu.
A document browser appears to allow you to select the document containing the UDF to instantiate.




If the document containing the UDF is already opened in a session:
Select **Instantiate From Selection** option in CATIA Insert menu.
Switch to the window of the document containing the UDF to instantiate and select the UDF in the tree.



If the UDF is stored in a catalog:
Open the CATIA catalog browser. Open the catalog containing the UDF to instantiate. Browse it in order to access the UDF component in the catalog.
Double-click on the UDF component to launch its instantiation.



Copyright DASSAULT SYSTEMES

 You can also use a VB macro to launch the instantiation. Refer the CATIA online documentation for more information.

Instructor Notes:

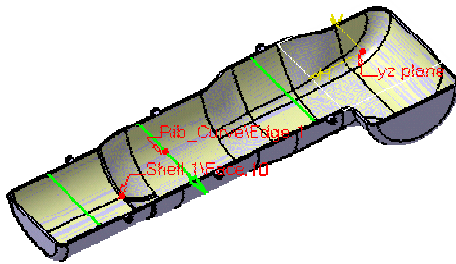
Instantiating a User Defined Feature (2/3)

- UserFeature instantiation is made of several steps:
 - ◆ Selection of the geometric inputs
 - ◆ Setting of the published parameters values

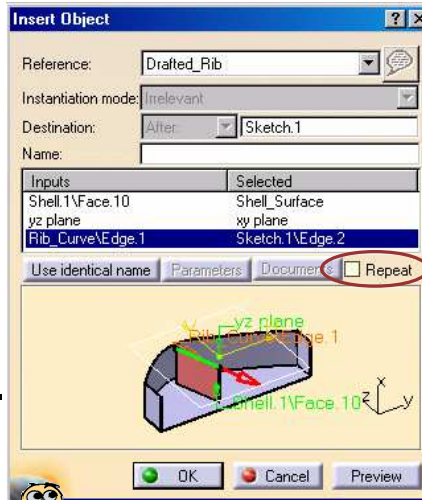
Select the inputs in the receiving part. You can select them either from the graphic area or in the tree.

Click the Use identical name button to have the inputs automatically filled in. Use this option only if the inputs of the receiving document have the same name than the inputs of the UDF.

If needed, click on the arrow in the 3D to invert it.



Copyright DASSAULT SYSTEMES



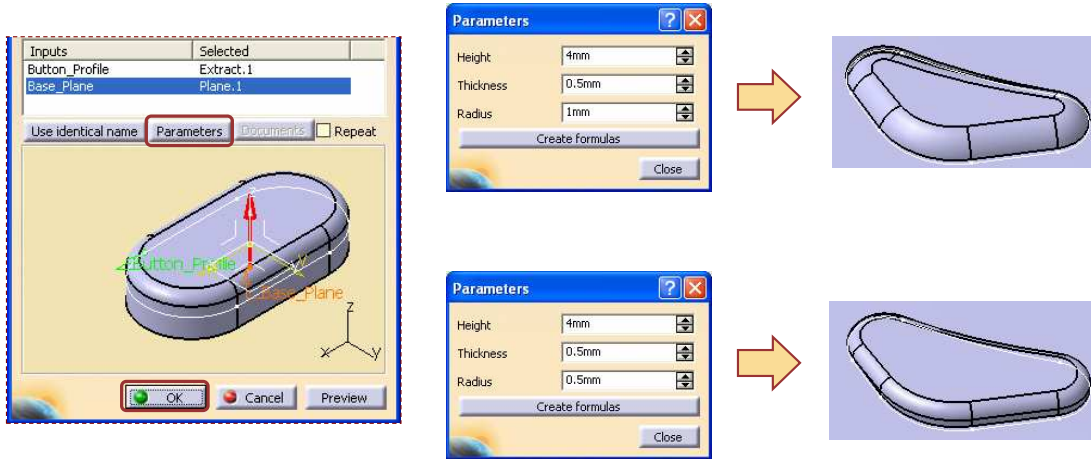
Check the Repeat button if you want to instantiate the UDF several times.

Be very careful to the orientation of the arrows while selecting the geometric inputs. It may change drastically the result of the instantiation.

Instructor Notes:

Instantiating a User Defined Feature (3/3)

Once all the geometric inputs are selected, the Parameters button allows you to change the value of the published parameters.



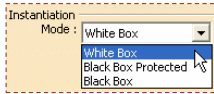
Click on Preview to check the result and then on OK to validate the creation of the UDF.

Copyright DASSAULT SYSTEMES

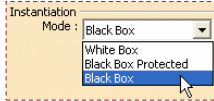
Instructor Notes:

Debugging a UDF

The UDF Debug icon allows you to visualize what is inside the User Feature instance. It allows you to switch between the expanded and the simplified view mode of the UDF.

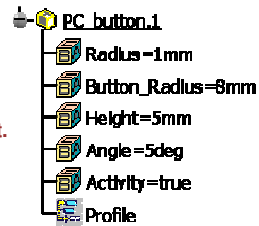


If the instantiation mode of the UDF is White Box, you will visualize what is inside the UserFeature instance at instantiation.

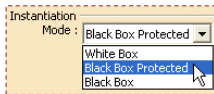
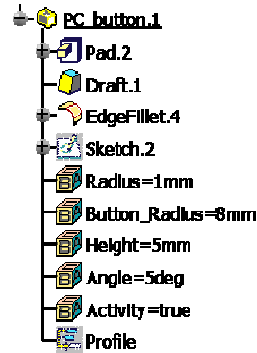


If the instantiation mode of the UDF is Black Box, the UserFeature instance view will be simplified at instantiation.

Simplified view.
Black Box default.



Expanded view.
White Box default.



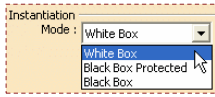
The Black Box Protected mode ensures a locked view of the UserFeatures, thus ensuring secure exchanges. In this mode, the UDF cannot be extended using the UDF Debug button.

Copyright DASSAULT SYSTEMES

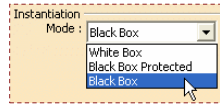
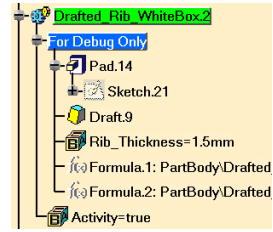
Instructor Notes:

How to debug a UDF (1/5)

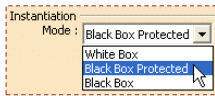
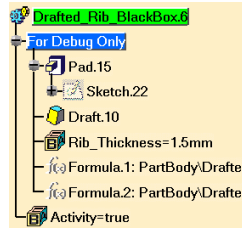
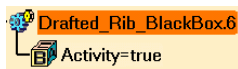
The UDF Debug icon allows you to visualize what is inside the User Feature instance. It allows you to switch between the expanded and the simplified view mode of the UDF.



If the instantiation mode of the UDF is White Box, you can visualize the contents of the UserFeature instance for debug purposes.



If the instantiation mode of the UDF is Black Box, using the 'Udf Debug' tool, you can expand and visualize the contents of the UDF instance.



The Black Box Protected mode ensures a locked view of the UserFeatures, thus ensuring secure exchanges. In this mode, the UDF cannot be extended using the UDF Debug tool.

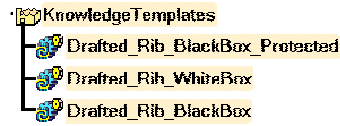
Copyright DASSAULT SYSTEMES

Instructor Notes:

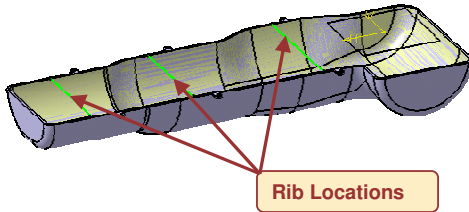
How to debug a UDF (2/5)

-  CATIA data used: Drill_Housing_For_RIB_UDF.CATPart
-  CATIA data used: Drafted_Rib_UDF_Source.CATPart

- 1 Open the attached CATIA documents and save it to some location.
- 2 Open Drafted_Rib_UDF_Source.CATPart and note that there are three UDFs present in it.



- 3 Close this part and open 'Drill_Housing_For_RIB_UDF.CATPart'.



This part contains three locations where you will instantiate UDFs in the form of 'Ribs' using the three different UDF templates from the 'Drafted_Rib_UDF_Source.CATPart' part.

Copyright DASSAULT SYSTEMES

Instructor Notes:

How to debug a UDF (3/5)

- 4 Instantiate the ribs using the three different UDFs from 'Drafted_Rib_UDF_Source.CATPart'. Select the inputs as shown below.

Three UDFs

Inputs	Selected
Limiting_Face	Shell_Surface
Rib_Curve_Plane	xy plane
Rib_Curve	Sketch.1\Edge.2

OK Cancel Preview

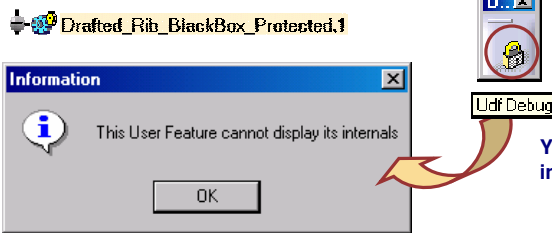
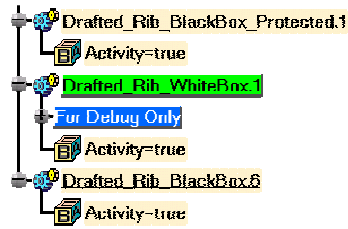
Copyright DASSAULT SYSTEMES

The inputs 'Limiting_Face' and 'Rib_Curve_Plane' are the same for all the three cases. The only variable input in the three UDF instantiations is the 'Rib_Curve', for which you have to select the three green lines in each individual case.

Instructor Notes:

How to debug a UDF (4/5)

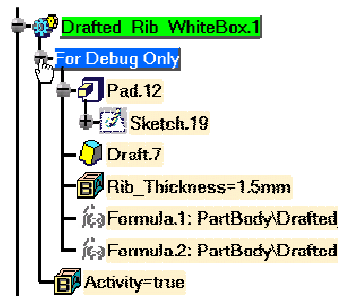
- 5 Note the three different UDFs in the specification tree.
- 6 Click on 'Drafted_Rib_BlackBox_Protected.1' in the specification tree and click the 'Udf debug' tool.



You will receive this message because the internals of the UDF are protected.

- 7 Expand the 'Drafted_Rib_WhiteBox.1' and note the internals of the UDF.

These internals are used for debugging purposes in case of any errors after instantiation.

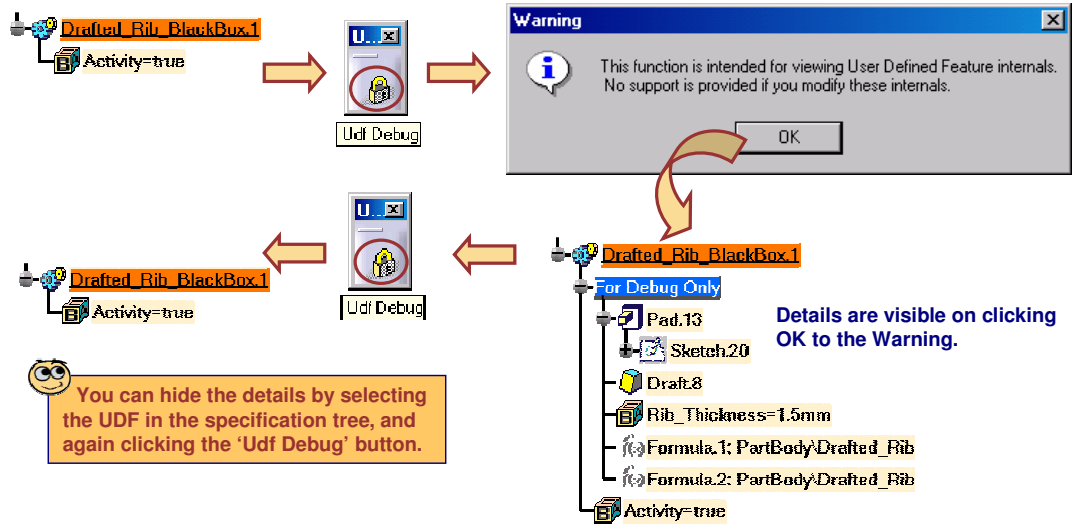


Copyright DASSAULT SYSTEMES

Instructor Notes:

How to debug a UDF (5/5)

- 8 To view the details of the 'Drafted_Rib_BlackBox.1' UDF, select it from the specification tree and click the 'Udf Debug' tool.



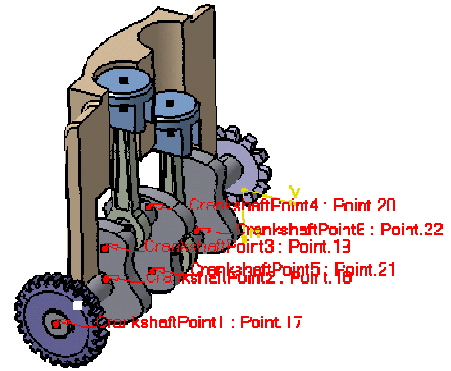
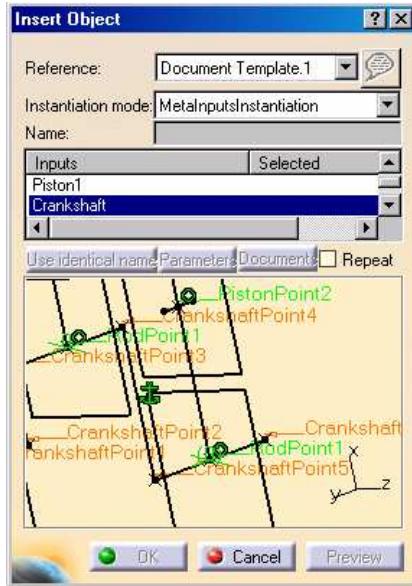
You can hide the details by selecting the UDF in the specification tree, and again clicking the 'Udf Debug' button.

Copyright DASSAULT SYSTEMES

Instructor Notes:

UDF – Meta Inputs

In this lesson, you will learn how to define and use the 'Meta Inputs' feature of UDF.



Copyright DASSAULT SYSTEMES

Instructor Notes:



What are 'Meta Inputs'

The Meta Inputs tab provides a facility to directly select a group of inputs simply by selecting a component in the specification tree during UDF instantiation.

The image shows a workflow for selecting meta inputs. On the left, the 'Document Template Definition' dialog has the 'Meta Inputs' tab active. It contains a table with the following data:

Name	Inputs Nb	Role
Rod1	1	CrankshaftPoint1
Piston1	2	CrankshaftPoint2
Crankshaft	6	CrankshaftPoint3
Piston2	2	CrankshaftPoint4
Rod2	1	CrankshaftPoint5
		CrankshaftPoint6

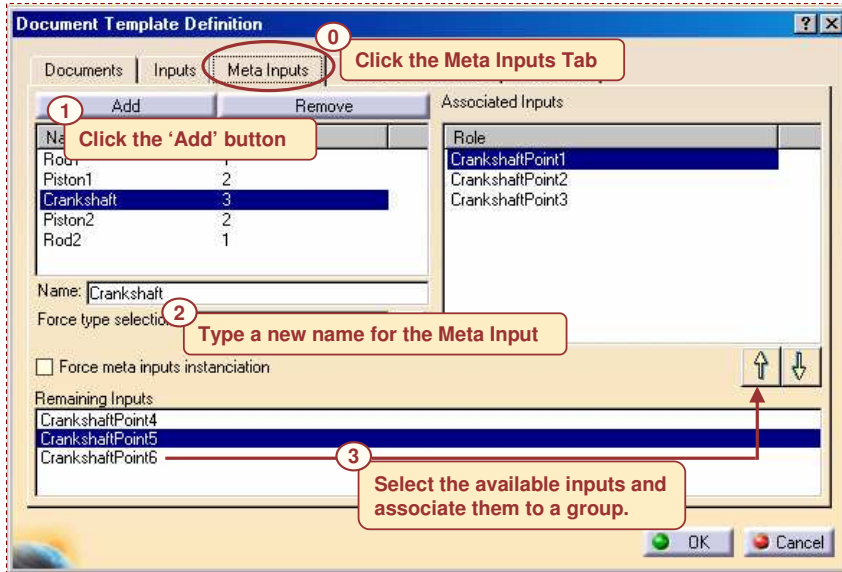
Red circles highlight 'Crankshaft' in the table and 'Crankshaft' in the 'Insert Object' dialog's 'Inputs' list. Arrows show the selection path from the table to the specification tree, and from the tree to the dialog. The 'Insert Object' dialog also shows a 3D model of a crankshaft mechanism with various points labeled.

Instructor Notes:



How to define 'Meta Inputs' for UDF

Defining the 'Meta Inputs' involves creating a 'Group of Inputs' and associating a number of individual inputs to this group.



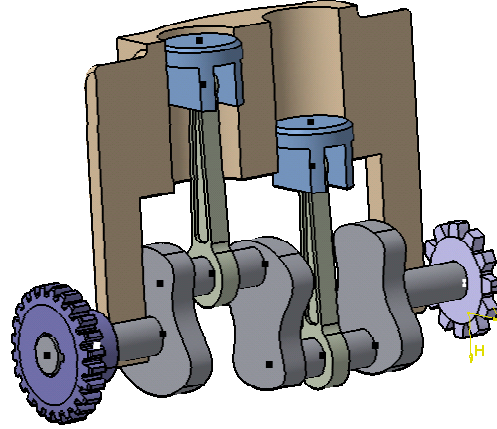
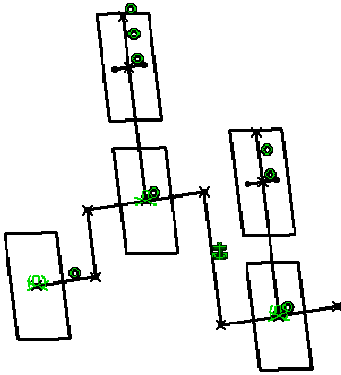
Copyright DASSAULT SYSTEMES

Instructor Notes:

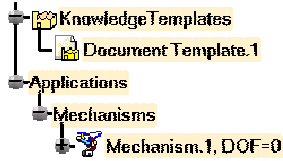


Example of Meta Inputs

For example, the geometry shown below (image on the left) consists of a wireframe mechanism of a two-cylinder engine. A document template of this product can be created and instantiated in an assembly of a two-cylinder engine (image on the right).



Copyright DASSAULT SYSTEMES



Instructor Notes:



How to select 'Meta Inputs' during UDF Instantiation

To be able to select the Meta Inputs during UDF instantiation, you have to select the 'Instantiation Mode' as 'MetaInputsInstantiation' and select the relevant components in the specification tree.

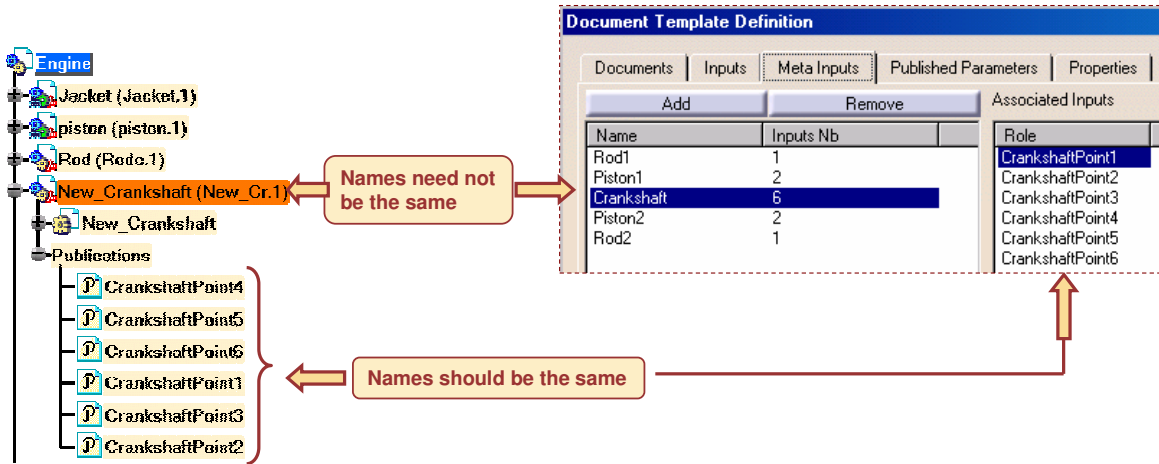
Copyright DASSAULT SYSTEMES

After selecting the relevant components, the published elements with the same names get selected.

Instructor Notes:



How to select 'Meta Inputs' during UDF Instantiation



It is not necessary that the names of the 'Meta Inputs' should match with the names of the components in the specification tree. However, it is necessary that the names of the published elements in the components should match with the names of the 'Associated Inputs' of the 'Meta Inputs' in the definition dialog box.

Copyright DASSAULT SYSTEMES

Instructor Notes:



User Defined Features Recap Exercises

You will practice on User Defined Features through two exercises:

-  Reactive Hole Recap Exercise
-  Center Hole Recap Exercise

Copyright DASSAULT SYSTEMES

Instructor Notes:

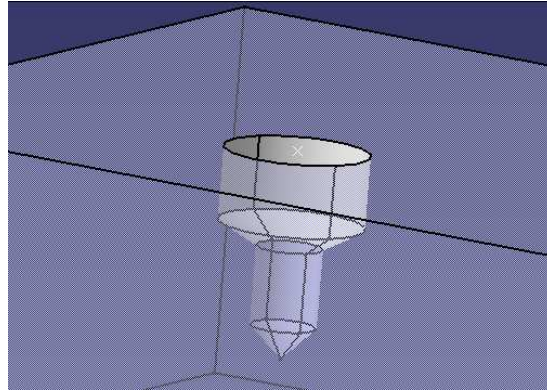
Reactive Hole

User Defined Features Recap Exercise



The objective of this exercise is to create a reactive Hole feature.
During the instantiation step, the hole feature will display a message indicating if a minimum distance between itself and the support face boundary is respected.

Prerequisites:
Knowledge Advisor



Copyright DASSAULT SYSTEMES

Instructor Notes:

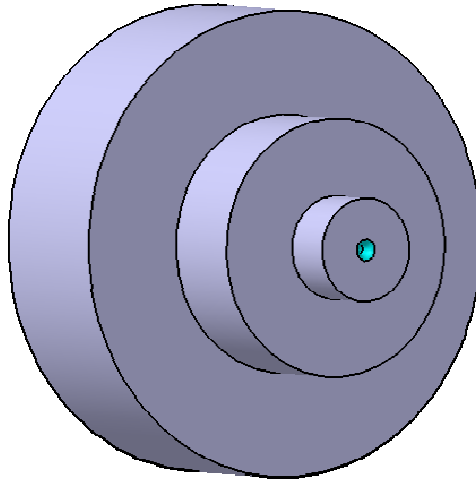
Center Hole

User Defined Features Recap Exercise



The objective of this exercise is to create a DIN standard center hole feature.

Prerequisites:
Knowledge Advisor



Copyright DASSAULT SYSTEMES

Instructor Notes:

Creating and Using Part and Assembly Templates

-  **Presentation of Document Templates**
-  **Creating a Document Template**
-  **Saving a Document Template**
-  **Instantiating a Document Template**
-  **Document Templates Recap Exercises**

Copyright DASSAULT SYSTEMES

Instructor Notes:

Presentation of Document Templates

You will learn about the benefits of Document Templates and their differences with respect to PowerCopies and User Defined Features.



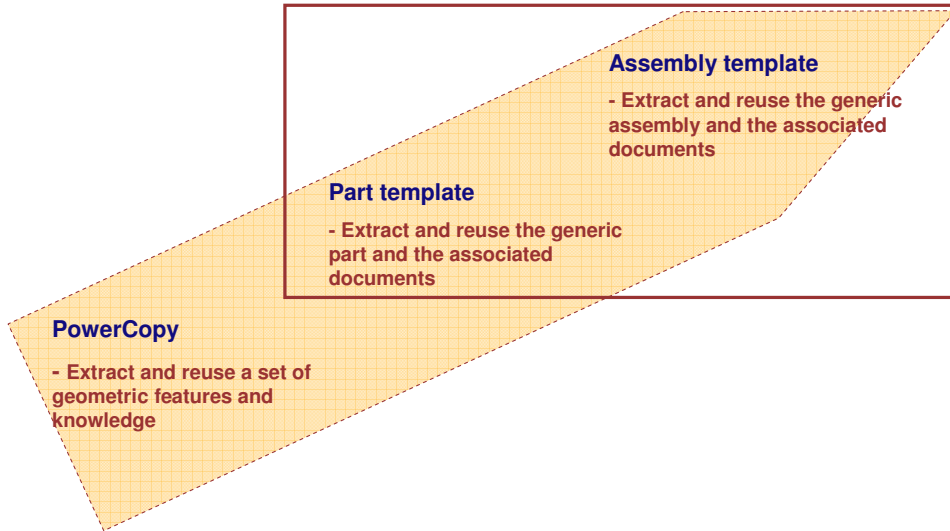
Copyright DASSAULT SYSTEMES

Instructor Notes:



Document Templates vs PowerCopies (1/2)

- Part and Assembly Templates are an extension of the PowerCopy capability at the level of the assembly.

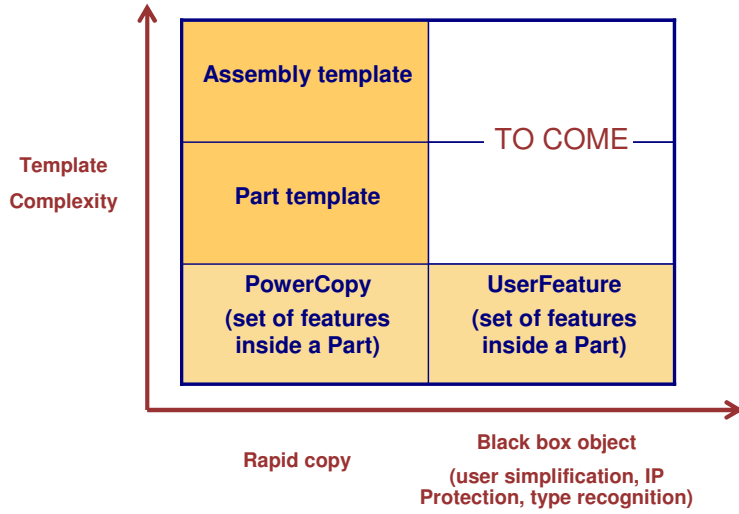


Copyright DASSAULT SYSTEMES

Instructor Notes:

Document Templates vs PowerCopies (2/2)

- Those templates are similar to PowerCopies and not to UserFeatures: they do not produce a single object when instantiated.



Copyright DASSAULT SYSTEMES

Instructor Notes:

What is a Part Template?

- A Part created in CATIA may contain user parameters and geometry data. It is not a contextual part. The user can create a part template that references that part. This template is a feature that is created in the CATPart document itself (very similar to the PowerCopy definition) and stored in a catalog. Several part templates may be defined in the same CATPart document.

- To create a Part Template, the user:
 - ◆ Selects parameters and geometry data that will be considered as the template inputs (he can assign a role and a comment to each input).
 - ◆ Publishes some internal parameters (name and comment). The part number is automatically published.
 - ◆ Gives a name, comment, URL, and icon to this template.

- Once the template is created, the user stores it in a catalog and uses it in another context. In product structure context, the part is inserted as a component of the current product.

Copyright DASSAULT SYSTEMES

Instructor Notes:

What is an Assembly Template?

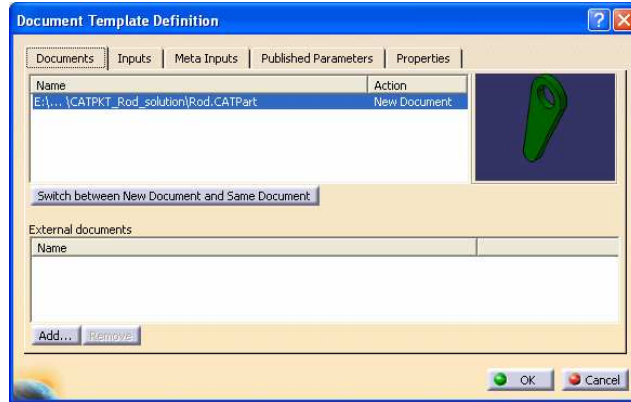
- A user creates an Assembly interactively. Then, he wants to create an Assembly Template that references the root product of this assembly.
- To create an assembly template, the user:
 - ◆ Selects parameters and geometry data that will be considered as the template inputs (he can assign a name to each input).
 - ◆ Publishes some internal parameters (name and comment).
 - ◆ Chooses if:
 - The part numbers of replicated components are automatically published
 - For each part or each sub-assembly this sub-component will be replicated at instantiation, or if only a reference to this sub-component will be created (a standard component)
 - He wants to select external documents (Drawings / Analysis) that references elements of the product structure. Those elements will be replicated at instantiation
 - ◆ Assigns a name, comment, URL and icon to this template.
- Once this template is created, the user stores it in a catalog and uses it in another context.
- The template definition is a feature located in the CATProduct document itself. Several assembly templates may be defined in the same CATProduct document.

Copyright DASSAULT SYSTEMES

Instructor Notes:

Creating a Document Template

You will learn how to store documents in a Template Feature in order to reuse them later in another context.



Copyright DASSAULT SYSTEMES

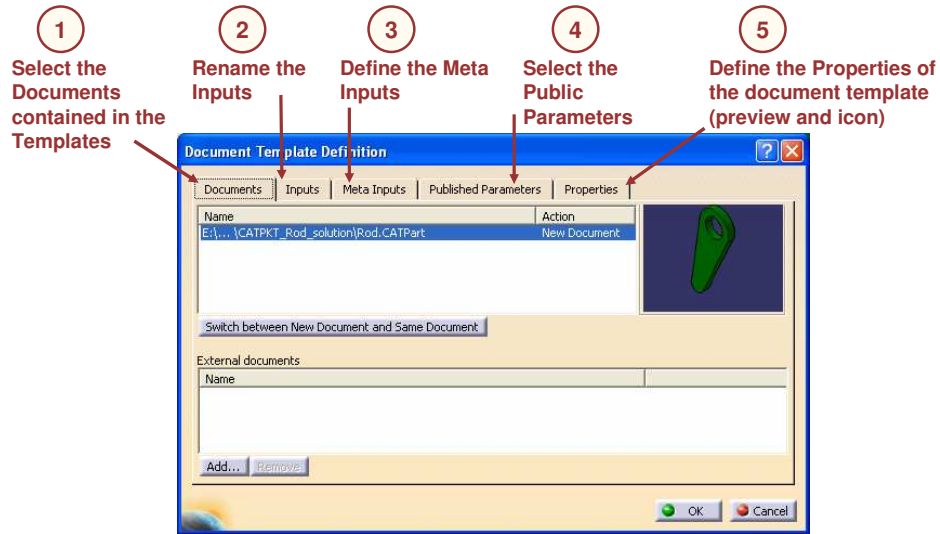
Instructor Notes:



Document Templates Creation Process

- The Document Template definition can be accessed in the Insert menu (Document Template Creation) of these workbenches:

- Part Design 
- Generative Shape Design 
- Assembly Design 
- Product Structure 



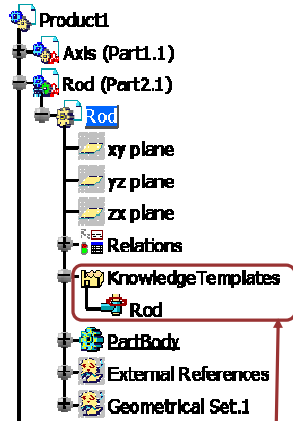
Copyright DASSAULT SYSTEMES

Instructor Notes:

Creating Document Templates (1/4)

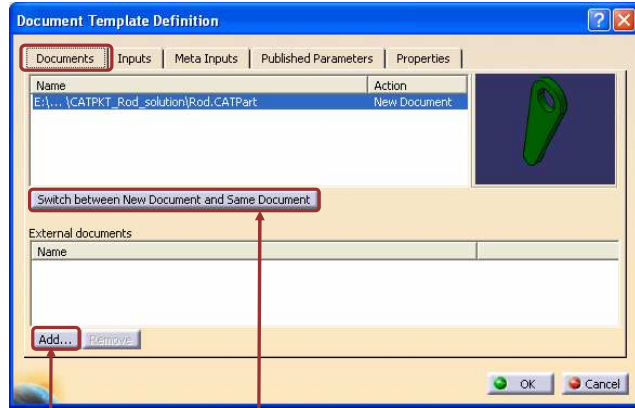
The Documents tab shows the complete path and action of the files referenced in the template.

Be active at the level of the document you want to create when launching the Document Template creation.



Created Document Templates are stored under the Knowledge Templates node

Insert External documents to the Template (CATDrawing for instance)



Switch between the Instanciated mode (the document has no links with the original component) and the Referenced mode (links maintained with the original file)

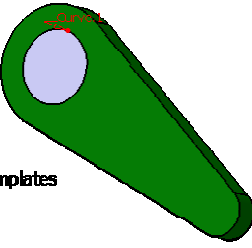
Copyright DASSAULT SYSTEMES

Instructor Notes:

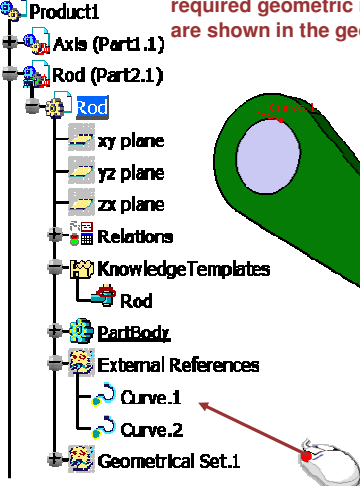
Creating Document Templates (2/4)

The Inputs tab enables you to define the reference elements (making up the template) by selecting them in the geometry or in the specification tree.

While editing the Inputs tab, required geometric inputs are shown in the geometry

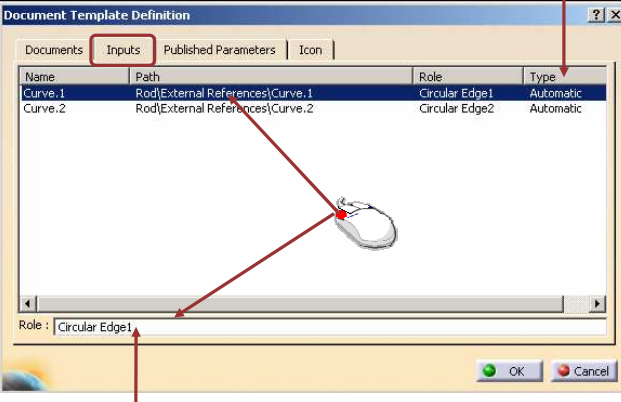


For manual inputs, selection is done by clicking on the features in the tree



Copyright DASSAULT SYSTEMES

If you have selected a document designed in context, the Inputs will be automatically selected



Name	Path	Role	Type
Curve.1	Rod External References Curve.1	Circular Edge1	Automatic
Curve.2	Rod External References Curve.2	Circular Edge2	Automatic

Role : Circular Edge1

OK Cancel

Instructor Notes:

Creating Document Templates (3/4)

The Published Parameters tab enables you to define which of the parameter values used in the template you will be able to modify when instantiating it.

The parameters have the same name than in formula editor, if you want to recognize them easily, rename them with the knowledgware tools.

The Edit List... button enables you to access the list of parameters, and to select those that you want to publish.

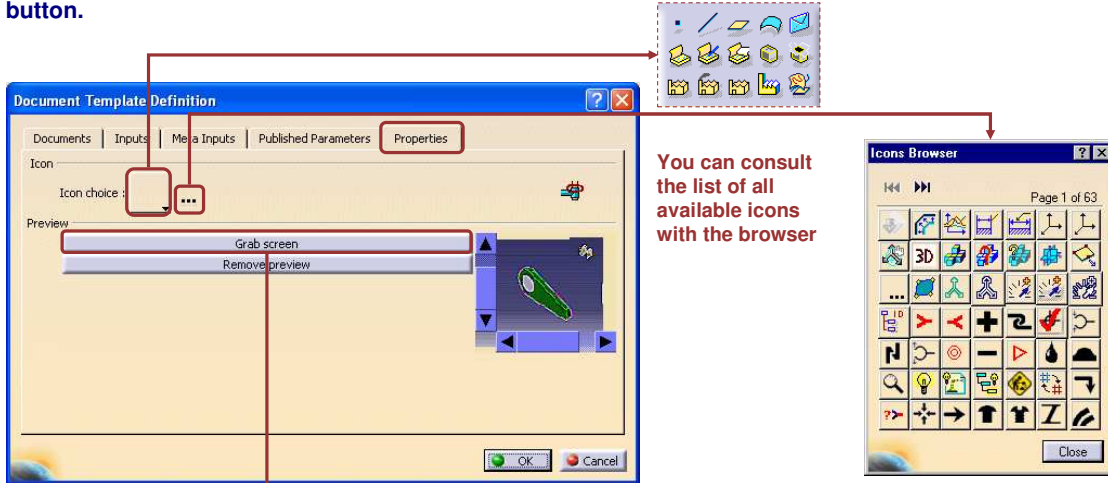
The Auto modify part numbers with suffix check box, if checked, automatically modifies the part numbers at instantiation if the part numbers already exist.

Copyright DASSAULT SYSTEMES

Instructor Notes:

Creating Document Templates (4/4)

The Properties tab enables you to modify the icon identifying the template in the specifications tree. A subset of icons is available while clicking the Icon choice button.




You can consult the list of all available icons with the browser

The Grab screen button enables you to capture an image of the template to be stored along with its definition

The Grab screen makes a grab of CATIA Window to put it as the preview of the Document Template: you can prepare the CATIA window for the grab (remove dialog box, compass and tree, and make the correct zoom)

Copyright DASSAULT SYSTEMES

 Preview will be useful while referencing a UserFeature in a catalog.

Instructor Notes:

Saving a Document Template

You will learn how to store a Document Template in a catalog in order to share it with other users or to reuse it later in another context.



Copyright DASSAULT SYSTEMES

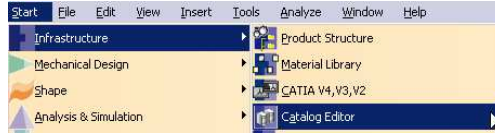
Instructor Notes:



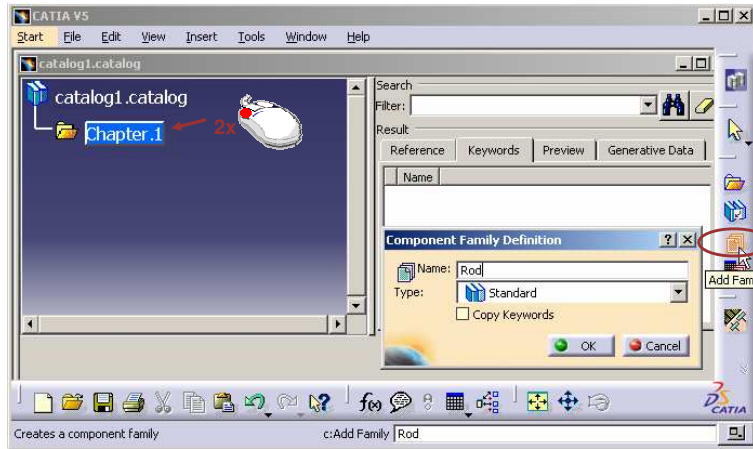
Saving Document Templates in a Catalog (1/2)

Save the file containing the Document Template.

From the Start menu, select the Infrastructure->Catalog Editor command. The Catalog Editor opens.



Double-click Chapter.1 and click the Add Family icon to create a family. Indicate the name of the family in the Name field, Rod in this scenario, and click OK. The Rod family is added below Chapter.1 in the tree.

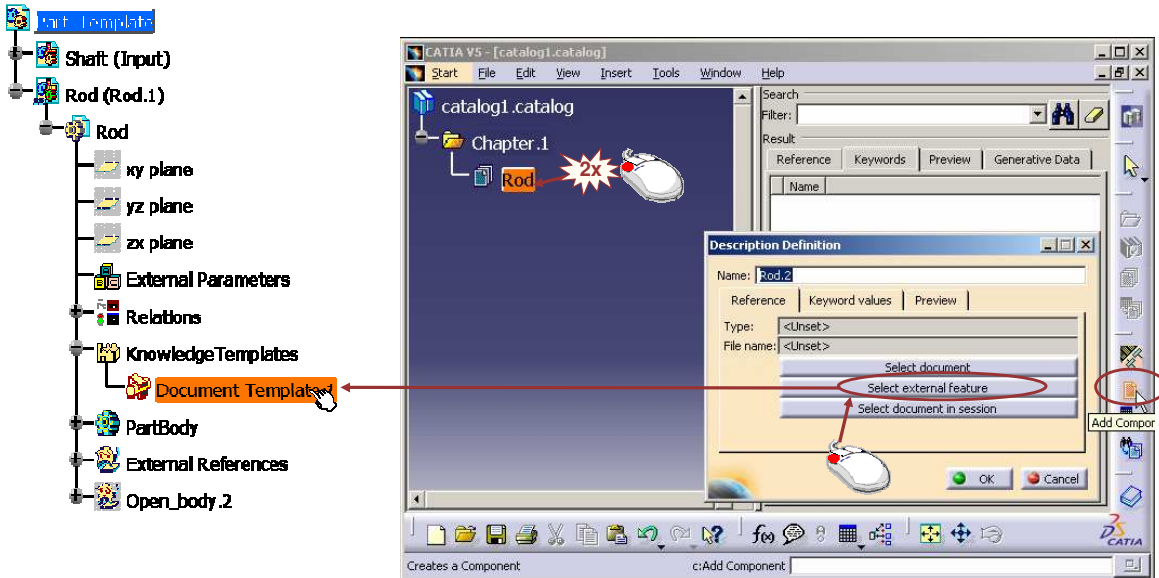


Copyright DASSAULT SYSTEMES

Instructor Notes:

Saving Document Templates in a Catalog (2/2)

Double-click Rod in the tree and click the Add Component icon. The Description Definition dialog box appears. Click the Select external feature button and click the Document Template in the file to select it. The template is added to the Description Definition window. Click OK.



Save the catalog and close it

Instructor Notes:

Instantiating a Document Template

You will learn how to import a Document Template in a new context and how to adapt it to this context.



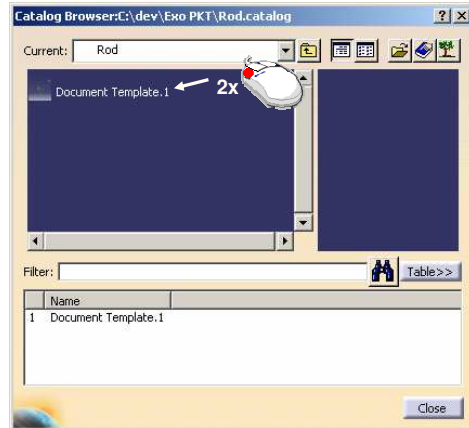
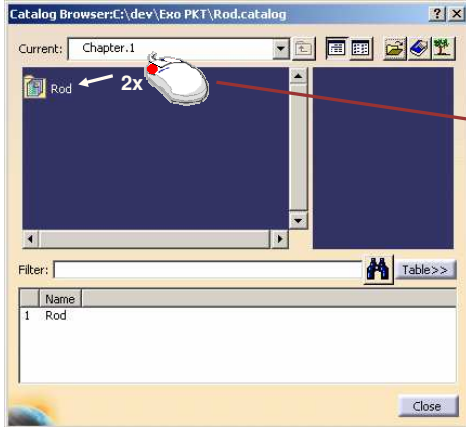
Copyright DASSAULT SYSTEMES

Instructor Notes:



Instantiating Document Templates from a Catalog (1/4)

Click the Catalog icon and select the catalog you have created.



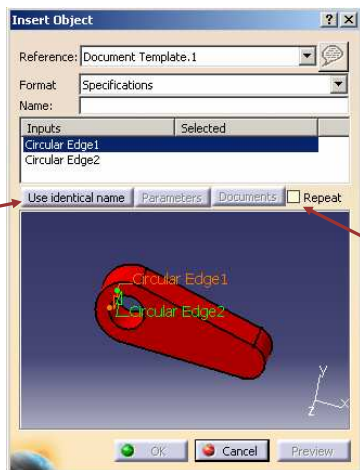
Double-click the Keypad family and the Document Template.1 template

Copyright DASSAULT SYSTEMES

Instructor Notes:

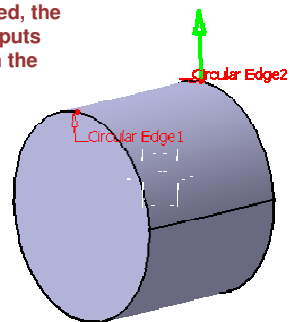
Instantiating Document Templates from a Catalog (2/4)

Select in the geometry the inputs of the Document Template.



Using identical names allows the automatic selection of the geometric inputs that have the same name as those used for the creation of the PowerCopy

When selected, the geometric inputs are shown in the geometry



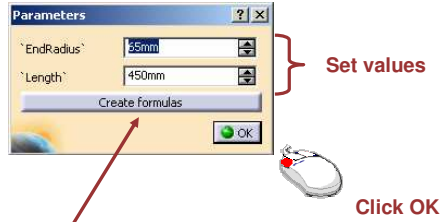
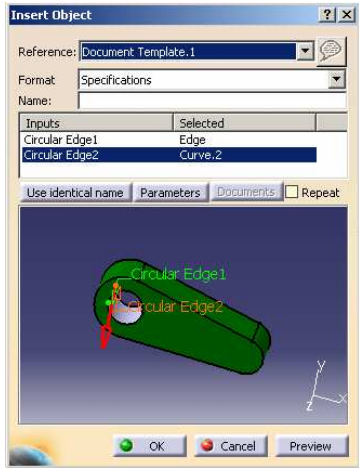
If you want to insert the document template several times, check the repeat option

Copyright DASSAULT SYSTEMES

Instructor Notes:

Instantiating Document Templates from a Catalog (3/4)

Click on Parameters to set values of parameters that have been published at the creation of the Document Template.



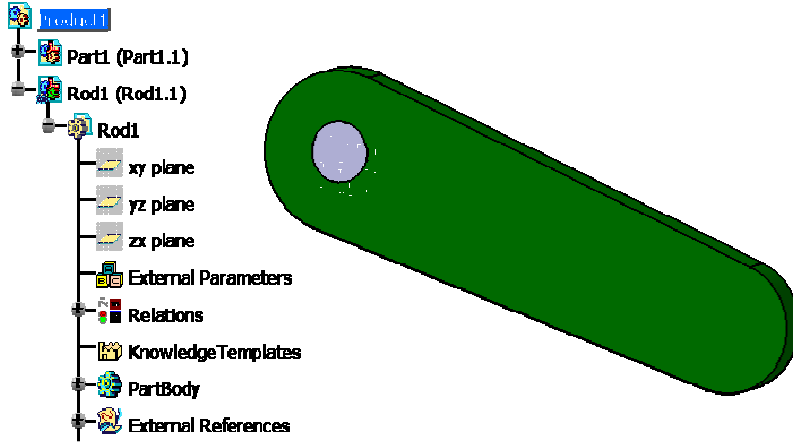
If you want, you can create on the fly formulas with parameters having the same names

Copyright DASSAULT SYSTEMES

Instructor Notes:

Instantiating Document Templates from a Catalog (4/4)

Click OK to the 'Insert Object' dialog box to instantiate the template.



Copyright DASSAULT SYSTEMES

Instructor Notes:

Document Templates Recap Exercises

You will practice Document Templates through three exercises:

-  **Rod Part Template Recap Exercise**
-  **Support Part Template Recap Exercise**
-  **Tow Hook Assembly Template Recap Exercise**

Copyright DASSAULT SYSTEMES

Instructor Notes:

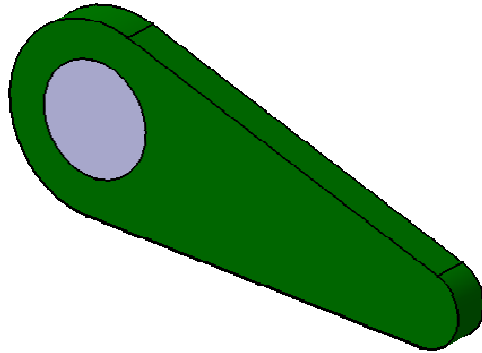
Rod Exercise

Part Templates Recap Exercise



10 min

In this exercise, you will create a Part Template based on a connecting rod design. Store it in a catalog and reuse it in an assembly.



Copyright DASSAULT SYSTEMES

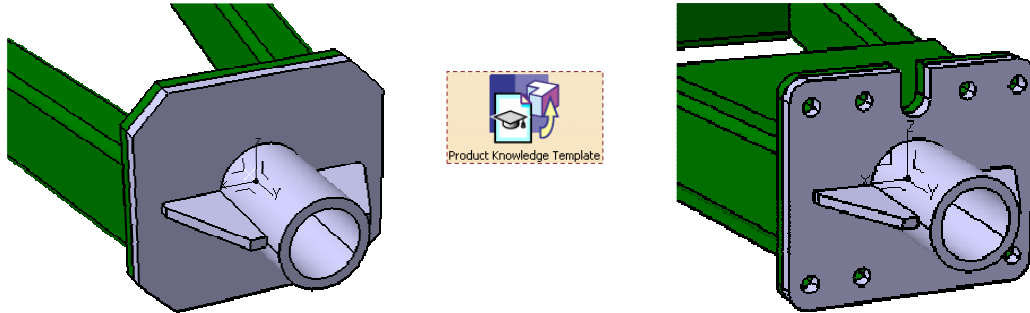
Instructor Notes:

Support

Part Templates Recap Exercise Presentation



In this exercise you will create a Part Template, store it in a catalog, and reuse it in an assembly.



Copyright DASSAULT SYSTEMES

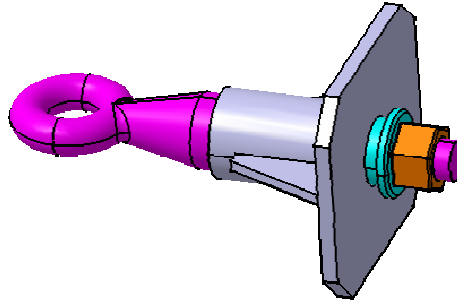
Instructor Notes:

Tow Hook

Assembly Template Recap Exercise



In this exercise you will create an assembly template, store it in a catalog, and reuse it in an assembly.



Copyright DASSAULT SYSTEMES

Instructor Notes:

Managing Standard Components

You will learn how to deal with standard components in Document Templates.



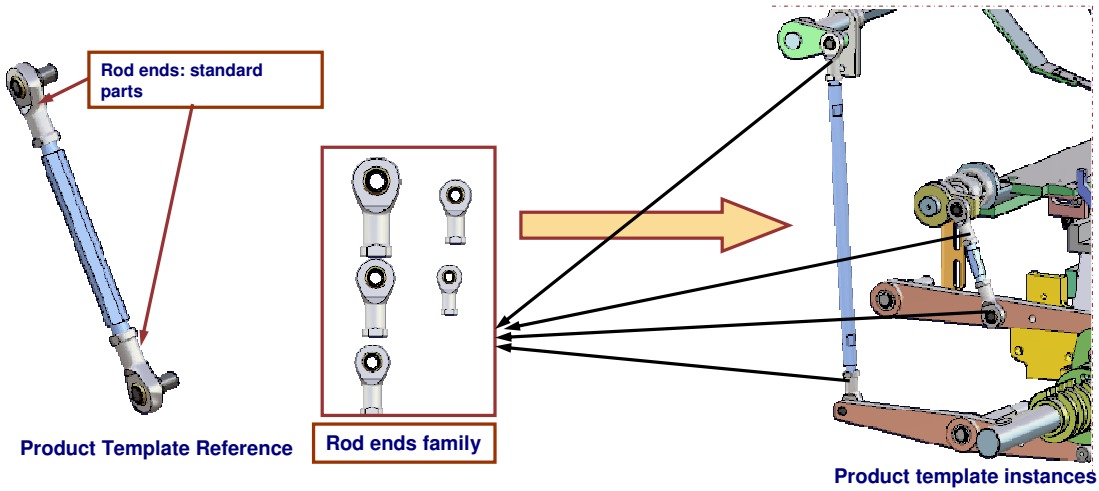
Copyright DASSAULT SYSTEMES

Instructor Notes:



Introduction

- An assembly template is usually made of a mix of specific components and standard components. In the example below, the female rod ends are standard parts, whereas the connecting bar and the pins are specific for each connecting rod:



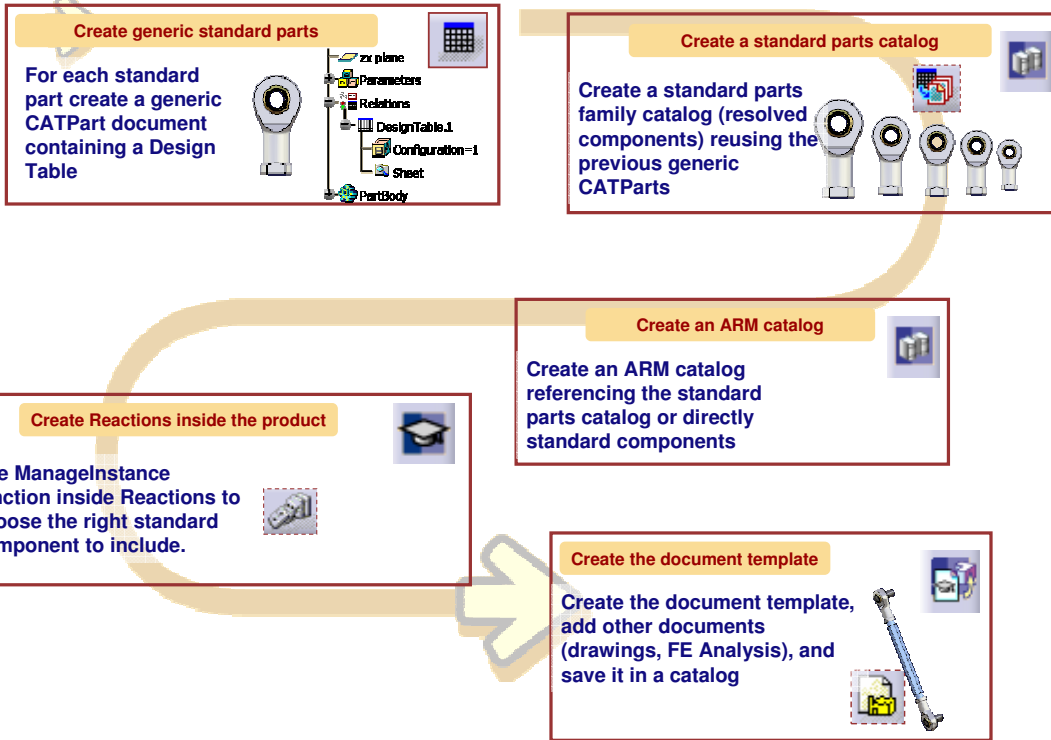
Copyright DASSAULT SYSTEMES

- Expected behavior after template instantiation: no new documents are generated for the rod ends. The Rod ends family documents are used inside the template instances.
- CATIA V5 R18 knowledge language enhancements allow to reach this behavior.

Instructor Notes:



Methodology Overview



Copyright DASSAULT SYSTEMES

Instructor Notes:



Knowledge Environment Settings

- ▶ Fill the “Architect Resources Creation Path”, it corresponds to the folder that contains other subfolders, among them: “knowledgeResourcesCatalogs” which contains the ARM catalogs (see next screen)

The screenshot shows the 'Options' dialog box with the 'Knowledge Environment' tab selected. The 'Architect Resources Creation Path' field is highlighted with a red circle and contains the path 'E:\users\sit\VS\Knowledge\Resources'. An inset window shows the file explorer view of this path, with a red arrow pointing from the field to the folder structure.

Instructor Notes:



About ARM Catalogs

- ▣ Application Resource Management (ARM) .catalog files establish a link between the logical name of a resource and the physical resource referenced through the catalog. The objective of the catalogs is to answer this simple question: "give me the object named XXX".
- ▣ ARM uses the "logical" referencing mechanism: the resource is referenced from the application by using the "logical name" instead of using its full path.
- ▣ The logical name is then used as the keyword in the ARM catalog.
- ▣ The catalog is the standard CATIA catalog created in the Catalog Editor. It must be based on a fixed structure containing the following keywords:
 - ▣ Name: corresponds to the name of the resource i.e the one created by default by the catalog application
 - ▣ Logical Name: corresponds to the logical name of the resource. It represents the resource identifier. The value must be unique since it will be used by ARM to find the corresponding resource in the catalog
 - ▣ Type: corresponds to the type of resource that you want to reach
 - ▣ Usage: corresponds to a comment indicating what this resource is used for

Copyright DASSAULT SYSTEMES



Type and Usage keyword values can be kept unset

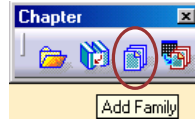
Instructor Notes:



Creating an ARM Catalog (1/3)

Create the Keywords

- 1 Create a new CatalogDocument (File>New).
- 2 Click the Add Family icon to add a family to your catalog. Click OK in the Component Family Definition dialog box. Double-click the family in the tree.
- 3 Click the Add Keyword icon or select Insert > Add Keyword... from the main menu to display the Keyword Definition dialog box.
- 4 Specify a name for the new keyword, Logical Name
- 5 Use the drop-down list to select the keyword Type, String. This list provides all knowledge types, i.e. Integer, String, Boolean, Angle, and so on.



Copyright DASSAULT SYSTEMES

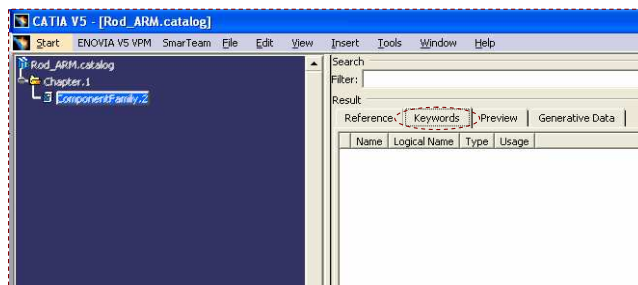
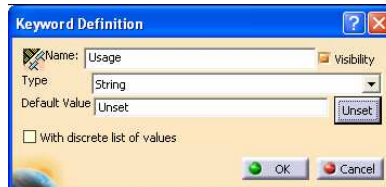
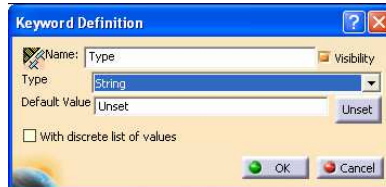
Instructor Notes:



Creating an ARM Catalog (2/3)

Create the Keywords

- 6 Repeat steps 3 & 4 to create the new keyword Type
- 7 Use the drop-down list to select the keyword Type, String
- 8 Repeat steps 3 & 4 to create the new keyword, Usage
- 9 Use the drop-down list to select the keyword Type, String



Copyright DASSAULT SYSTEMES

Instructor Notes:



Creating an ARM Catalog (3/3)

Add a standard parts family catalog

1 In the Catalog Editor, click the Add Component icon (). The Description Definition dialog box opens.

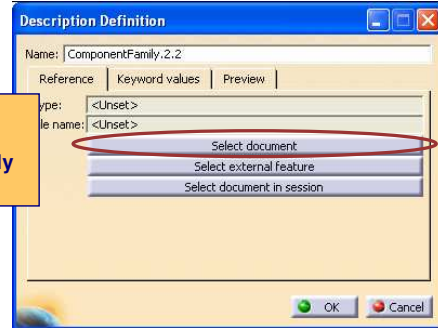


2 Click the Select document button.

3 Change the Files of type field to All Files (*.*) and select the .catalog file containing the standard parts family. Ignore the error message which is displayed.



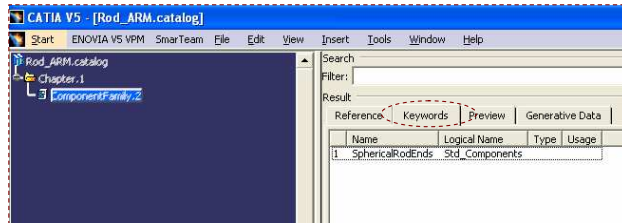
Refer to the Knowledge Advisor lesson to learn how to create a part family catalog.



4 Click the Keyword values tab. Click the Logical Name line and enter the desired logical name, click OK when done.

5 You can keep the default value Unset for Type and Usage keywords.

6 Save the catalog in the \\xxx\knowledgeResourcesCatalogs directory to be taken into account by the Application Resources Management system.



Copyright DASSAULT SYSTEMES

Instructor Notes:



Choosing the Right Standard Component (1/5)

- The Knowledgeware language provides the ManagelInstance and the RemoveInstance functions to address three major cases:
 - ◆ Case 1: need to switch between different standard components
 - ◆ Case 2: need to switch on/off a standard component
 - ◆ Case 3: need to search for the right element inside a standard parts family
- Syntax
 - ◆ Product->ManagelInstance(*arm* : String, *chapterName* : String, *query* : String, *instanceName* : String): Product
 - ◆ ARM: Application Resource Management string. It is composed of two parts separated by "|": <catalogName>|<value of the keyword "Logical Name" of the catalog description>. The catalog description has to reference either a CATPart document or a CATProduct document (case 1 & 2) or a catalog document (case 3).
 - ◆ Chaptername: This argument is only used in case 2. In this case the ARM resource is a catalog document, and if the chapter name is specified, the system looks for the catalog chapter of this name.
 - ◆ Query: This argument is only used in case 2. In this case, the query is used to retrieve the part family elements that fit this query, either in a specific chapter i.e. the *chapterName* argument is filled, or in the whole catalog.
 - ◆ InstanceName: The ManagelInstance method either creates or replaces a product instance. This argument is used to retrieve the existing instance, if any. It is also used to rename the created instance.
 - ◆ Product->RemoveInstance(*instanceName* : String

Copyright DASSAULT SYSTEMES

Instructor Notes:



Choosing the Right Standard Component (2/5)

Case 1 example: Switching standard motors inside a conveyor

```

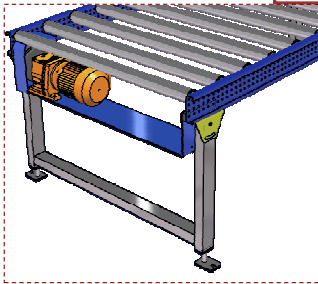
Let Prod (Product)
if (Motor=="0.5HP")
  Prod = Conveyor->ManageInstance("ARM|Motor0.5HP", "", "", "Motor")
if (Motor=="1HP")
  Prod = Conveyor->ManageInstance("ARM|Motor1HP", "", "", "Motor")
    
```

In this case, the second and the third arguments are useless

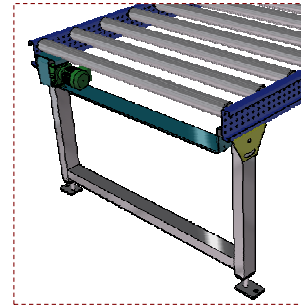
Assembly that contains the standard component

Logical name for component corresponding to a logical resource in an ARM (Application Resources Management) file referencing the standard components

Name of the instance is used to determine whether we are in the insertion or the replace mode



Automatic motor switch



Use Publications inside standard components to keep the assembly constraints, and the external references connected after the replace operation.

Copyright DASSAULT SYSTEMES

Instructor Notes:



Choosing the Right Standard Component (3/5)

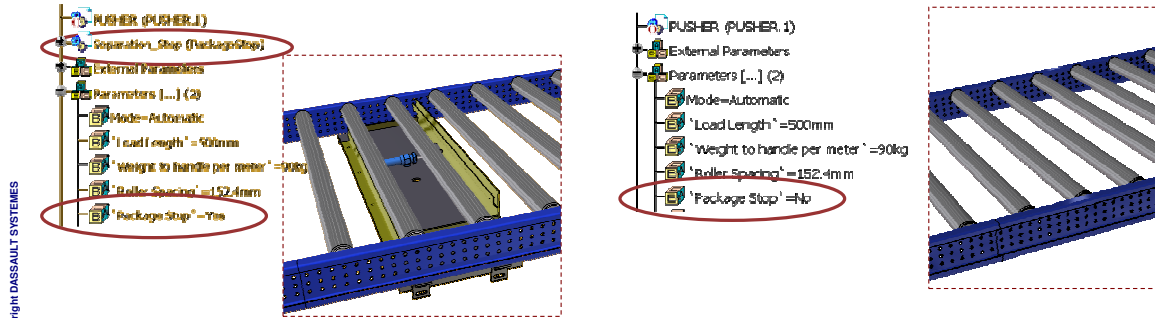
Case 2 example: Switching on/off a package stop in a conveyor

```

let optionproduct(Product)
if `Package Stop` == "Yes"
  {set optionproduct= Master\Conveyor.1 ->ManageInstance("ConveyorR18|Stop", "", "", "PackageStop")
  Constraints\Coincidence.77\Coincidence.77\Activity =true
  Constraints\Coincidence.78\Coincidence.78\Activity =true
  Constraints\Coincidence.79\Coincidence.79\Activity =true}
if `Package Stop` == "No"
  {Constraints\Coincidence.77\Coincidence.77\Activity =false
  Constraints\Coincidence.78\Coincidence.78\Activity =false
  Constraints\Coincidence.79\Coincidence.79\Activity =false
  Master\Conveyor.1 ->RemoveInstance("PackageStop")}
    
```



Do not forget to activate/inactivate the assembly constraints linked to the standard component.



Package stop option: Yes/No

Instructor Notes:



Choosing the Right Standard Component (4/5)

Case 3 example: Choosing the right standard rod ends inside a connecting rod assembly

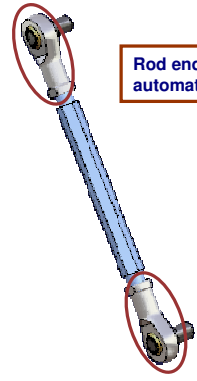
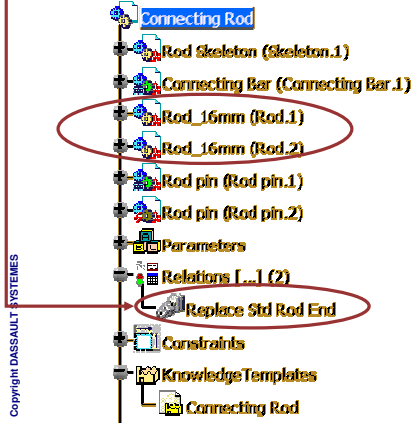
```

let newinstance (Product)
let instancename1 (String)
let instancename2 (String)
let query(String)
instancename1="Rod.1"
instancename2="Rod.2"
query="x.D=" + ToString(' Rod Skeleton\End Rod Diameter' )
newinstance= `Connecting Rod` ->ManageInstance("Rod_ARM|Std_Components","SphericalEndRods",query,instancename1)
newinstance= `Connecting Rod` ->ManageInstance("Rod_ARM|Std_Components","SphericalEndRods",query,instancename2)
    
```

Query to perform in the catalog: search for rod ends with a diameter (D) equal to End Rod Diameter parameter value

Logical name for the catalog corresponding to the logical resource in an ARM (Application Resources Management) file, referencing a catalog

Name of the chapter in the catalog



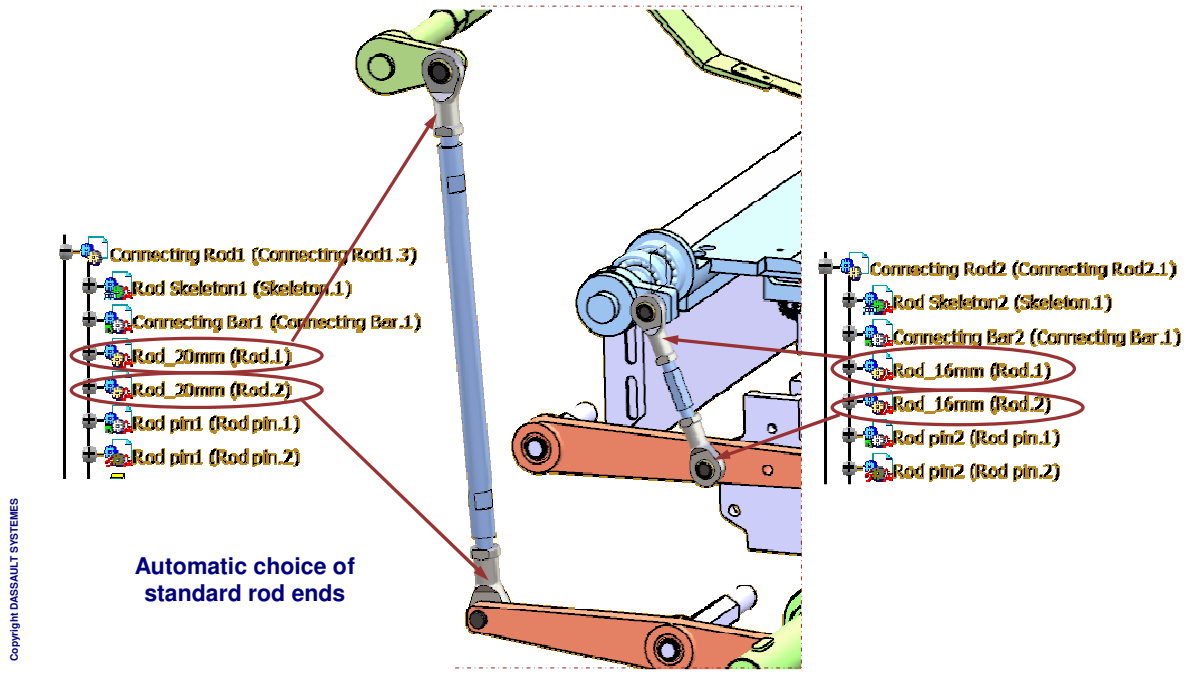
Add the knowledge instructions inside a Reaction, which reacts to a skeleton's parameter value change, End Rod Diameter parameter in this case. This parameter is computed from the selected geometry at the template instantiation step.

Instructor Notes:



Choosing the Right Standard Component (5/5)

Case 3 example: Product structure result after document template instantiation

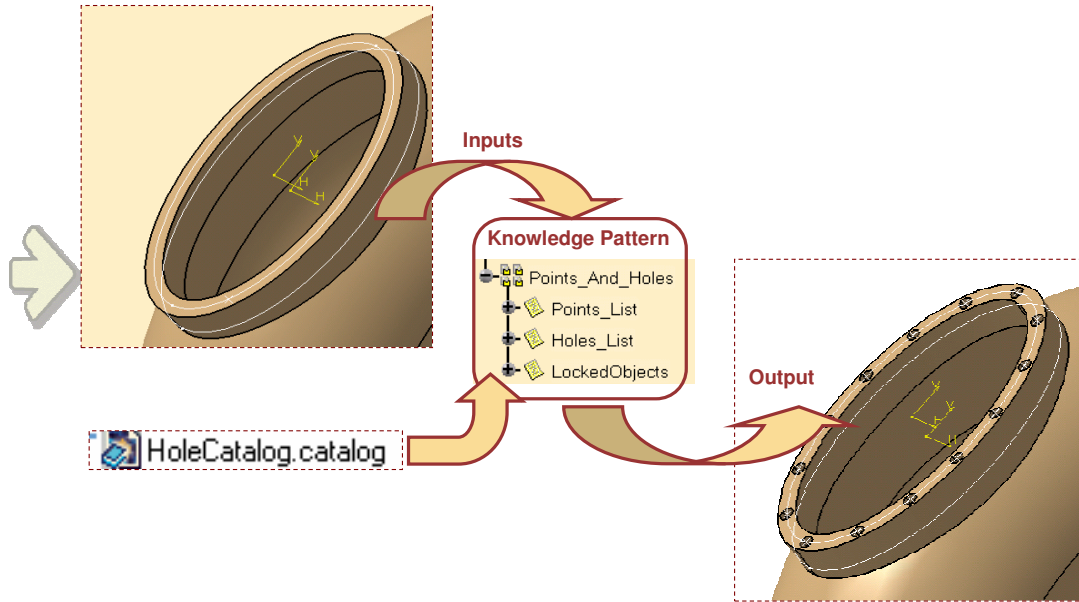


Instructor Notes:



Knowledge Pattern

In this skillet, you will learn the mechanism of 'Knowledge Pattern' and its applications. You will also learn how to create and instantiate 'Knowledge Pattern'.



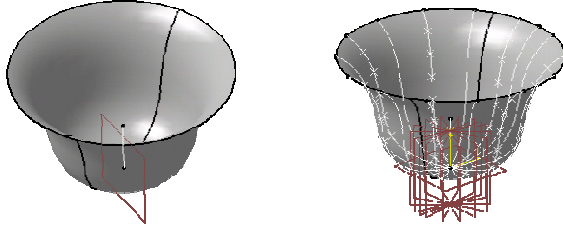
Instructor Notes:



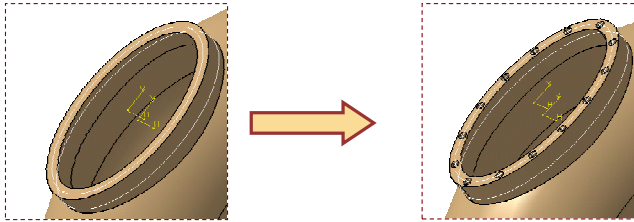
Where do we use 'Knowledge Pattern'?

Knowledge Pattern is primarily used for the following purposes:

- To create a sequence of geometric elements resulting from certain user defined instructions.



- To instantiate a series of user defined features.



- Knowledge Pattern can also be used in place of the KWA 'Loop' functionality, which is not obsolete.

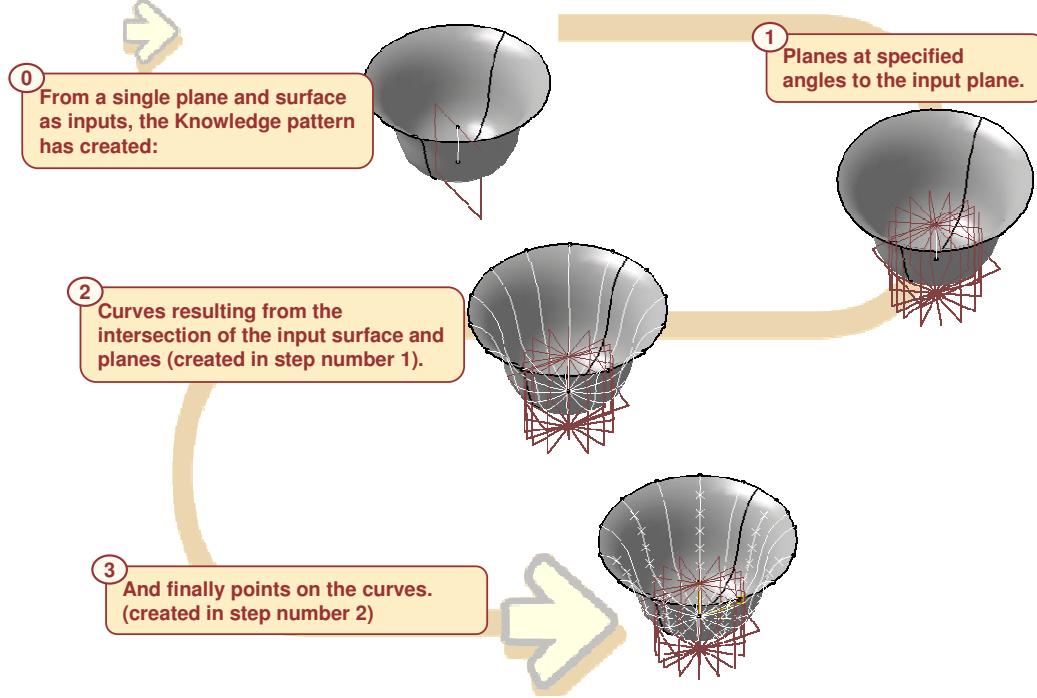
Copyright DASSAULT SYSTEMES

Instructor Notes:



Example of Knowledge Pattern

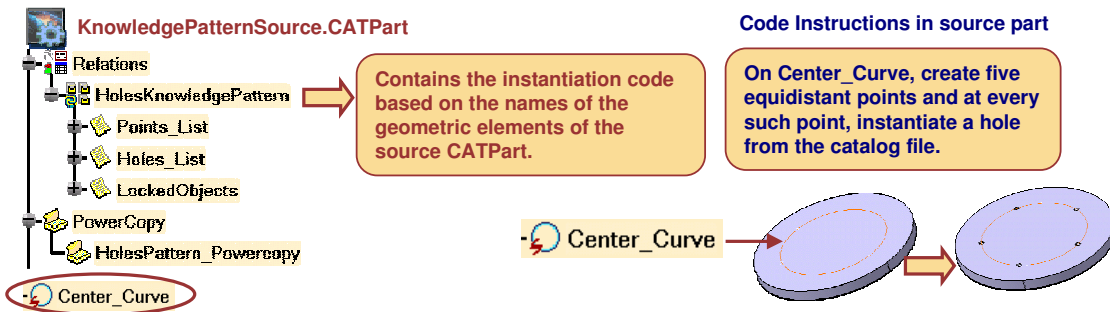
In this example, you can notice how the 'Knowledge Pattern' is used to create sequence of geometric elements one after the other.



Instructor Notes:

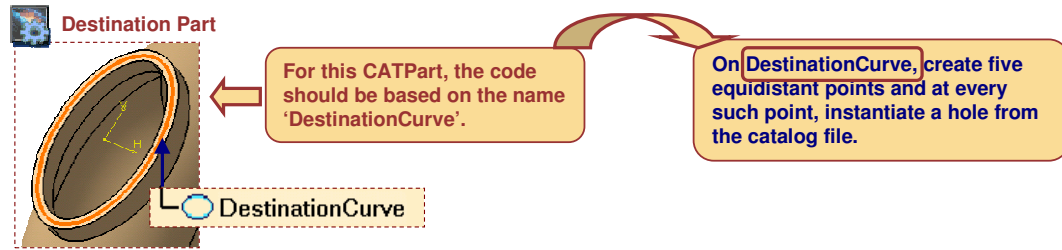


The Mechanism of Knowledge Pattern (1/2)



The mechanism of Knowledge Pattern involves geometry creation / instantiation by writing code of instructions, using the 'Knowledge Pattern Feature'.

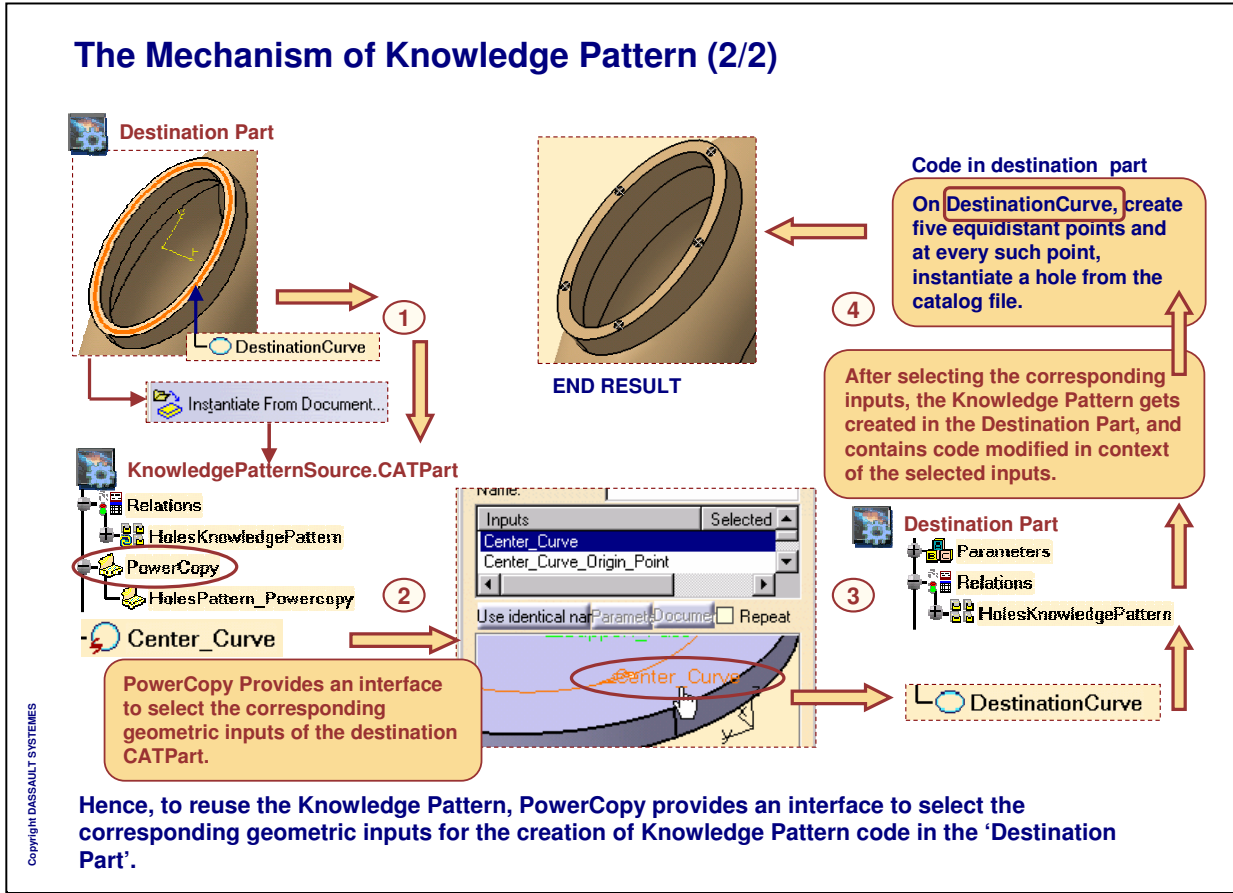
However, if this Knowledge Pattern is to be reused, as the names of the geometrical elements in the CATPart could be different for different CATParts, the same code of instructions may not work for other CATParts.



Instructor Notes:



The Mechanism of Knowledge Pattern (2/2)

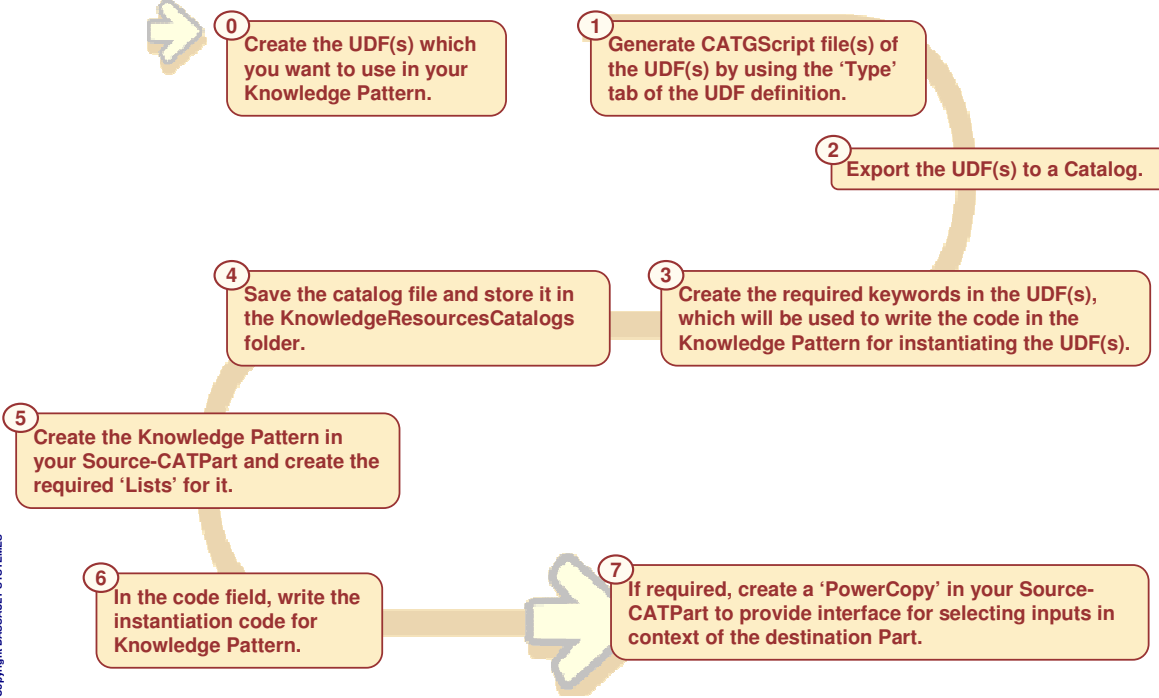


Instructor Notes:



General Process - Knowledge Pattern – UDF Instantiation

Following is the General Process followed for creating Knowledge Pattern that involves “Instantiation of a UDF”.

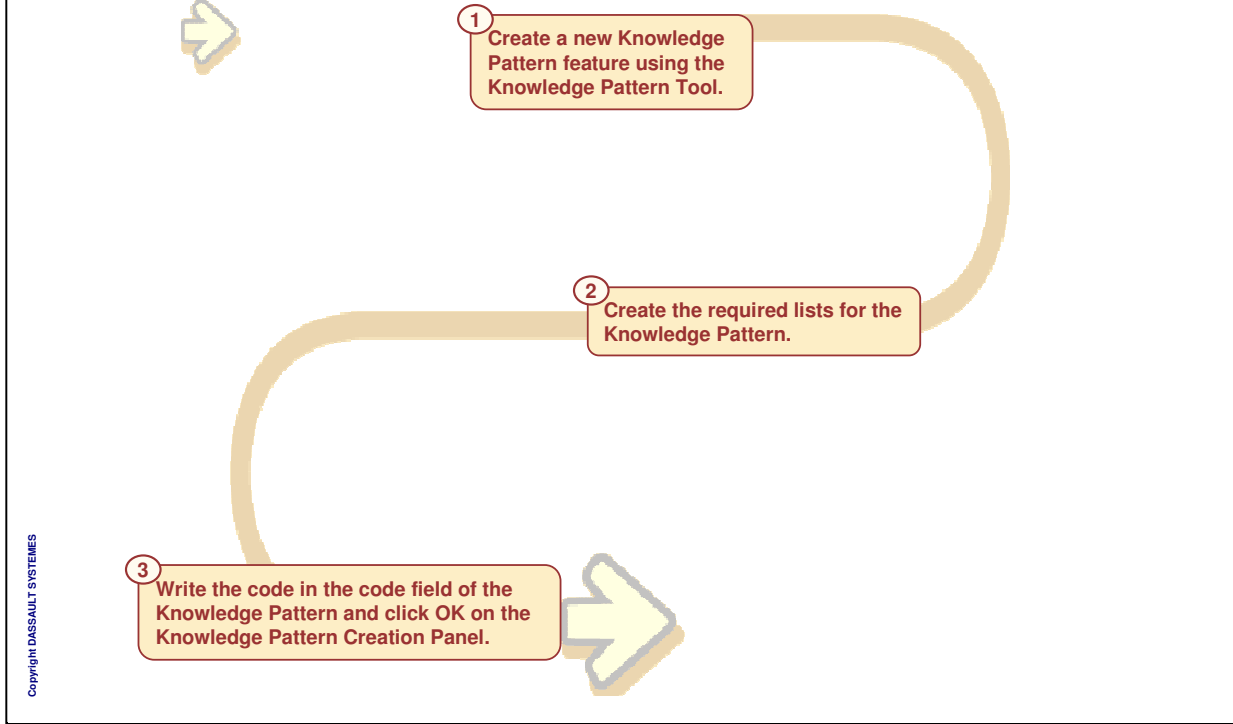


Instructor Notes:



General Process - Knowledge Pattern – Datum Creation

Following is the general process followed for creating the Knowledge Pattern that involves creation of 'Datums'.



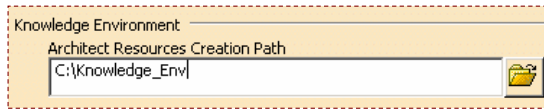
Instructor Notes:



How to Create Knowledge Pattern (1/9)

Before creating a Knowledge Pattern, you can set the folder for “Architect Resources Creation Path” folder.

This setting can be accessed in Tools > Options > General > Parameters and Measures > Knowledge Environment tab.



After doing this setting, the files which are created by Knowledge Pattern functionalities fall in this folder.

Otherwise, the files are created in the installation folder of CATIA V5.
(...\intel_aresources\Knowledge)



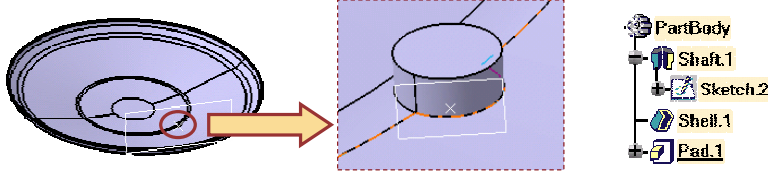
Note that you will have to restart CATIA for this setting to take effect.

Instructor Notes:

How to Create Knowledge Pattern (2/9)

 CATIA Data Used: Hole_UDF.CATPart

- 1 Open Hole_UDF.CATPart and save it to some location.



Notice Pad.1. In this example, you will instantiate Pad.1 into a destination body and use the 'Remove' Boolean operation to create a hole.

- 2 Double-click the UDF – 'Hole_UDF' and notice the inputs of the UDF in the 'Inputs' tab.



Copyright DASSAULT SYSTEMES

Instructor Notes:

How to Create Knowledge Pattern (3/9)

3 Generate the CATGScript file of the UDF using the 'Type' tab of the UDF definition. Click the 'Auto' button to have the SuperType = MechanicalFeature. Click the 'Generate' button of the 'Type1' (Standard Inputs) field.

Copyright DASSAULT SYSTEMES

A CATGScript file will be created in the folder which is specified in the Knowledge Environment > Architect Resource Creation Path.

Instructor Notes:

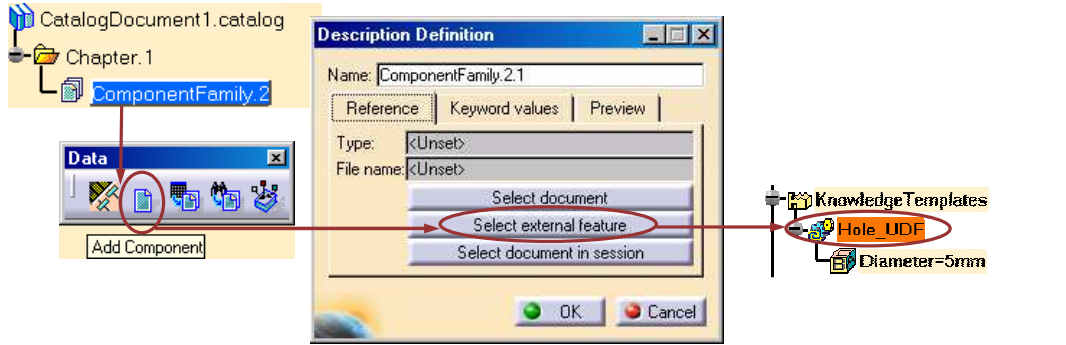
How to Create Knowledge Pattern (4/9)

4 Export the Hole_UDF to a catalog file and create the required keywords in the catalog file.

4a Create a new Catalog Document and add a new family to the default Chapter.1.



4b Activate the component family and add the Hole_UDF feature as a component in this catalog file.



Click 'Add Component'

Click 'Select external feature'

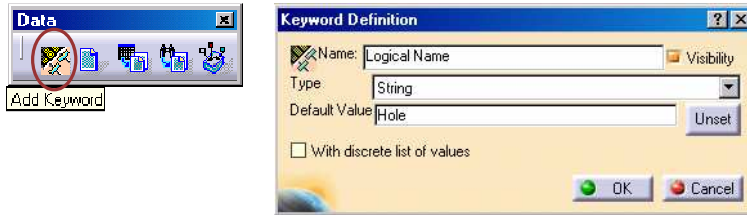
Select the UDF Hole_UDF from Hole_UDF.CATPart and click OK

Copyright DASSAULT SYSTEMES

Instructor Notes:

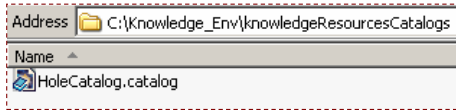
How to Create Knowledge Pattern (5/9)

- 4a) Click the 'Add Keyword' tool, specify the name, type, and default value as shown, and click OK.



The keyword name and its value is used as an identifier to write the instantiation code in the Knowledge Pattern Feature.

- 5) Save this Catalog Document by the name 'HoleCatalog.Catalog' and copy it to the 'KnowledgeResourcesCatalog' folder of the "Architect Resource Creation Path" – Folder.



Copyright DASSAULT SYSTEMES

Instructor Notes:

How to Create Knowledge Pattern (6/9)

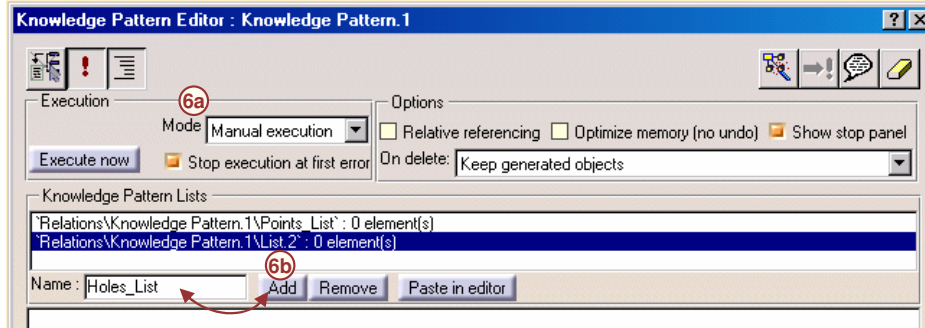


CATIA Data Used: KnowledgePatternSource.CATPart

- 6 Open the KnowledgePatternSource.CATPart, and create a 'Knowledge Pattern Feature' by clicking the 'Knowledge Pattern Tool' of the Product Knowledge Template workbench.



Creates a Knowledge Pattern feature



- 6a In the dialog box, select the 'Manual Execution' mode.
- 6b Add the lists named 'Points_List' and 'Holes_List' for the Knowledge Pattern by clicking the 'Add' button.



To add the lists, click the 'Add' button, type the name of the list, and again click the 'Add' button.

- 7 After creating the required lists, click OK on the 'Knowledge Pattern Editor' dialog box, and rename the 'Knowledge Pattern Feature' to "HolesKnowledgePattern".

Copyright DASSAULT SYSTEMES

Instructor Notes:

How to Create Knowledge Pattern (7/9)

Data Used: HolesKnowledgePattern_Code.txt

- 8 Double-click the 'Knowledge Pattern' feature in the specification tree, copy the code from the attached text file and paste it in the 'Code Field' of the 'Knowledge Pattern Feature' dialog box. Click OK on the dialog box.

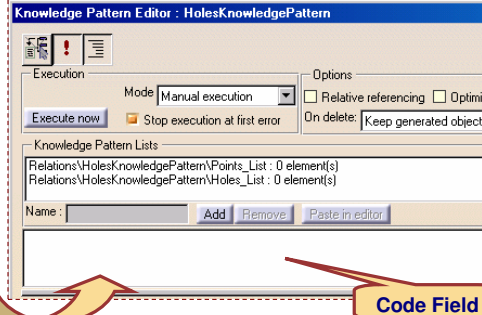
```

HolesKnowledgePattern_Code.txt - Notepad
File Edit Format Help
let p (point)
let Spacing (length)
let StartPoint (point)
let udf (Hole_UDF)
let n (integer)

i=1

l=length('Geometrical Set.1\Circle.1')
Spacing=l/(Holes_Number)

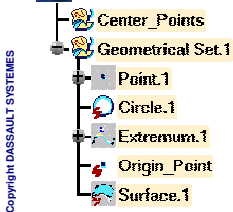
Relations\HolesKnowledgePattern\Points_List.AddItem('Geometrical
Set.1\Origin_Point',1)
    
```



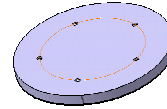
Copy-Paste

Code Field

KnowledgePatternSource.CATPart



This code is written in context of the geometric inputs of the source part. In the above code, instructions are written to create equidistant points on the curve 'Circle.1', and to create a hole at every such point. If you run this feature, you will get the result as shown in the adjoining image.



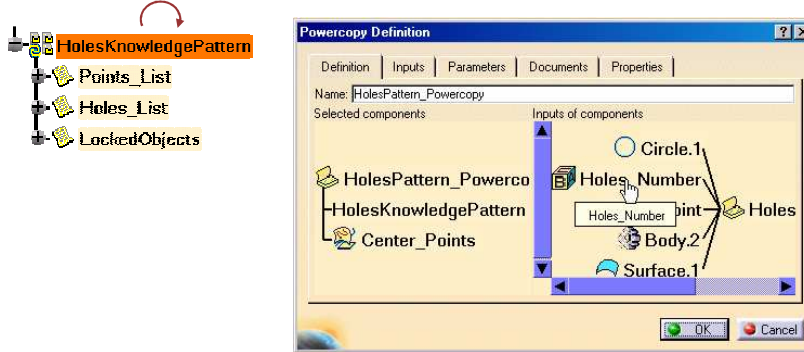
Instructor Notes:

How to Create Knowledge Pattern (8/9)

9 Create a PowerCopy feature using this 'Knowledge Pattern' feature as shown.

9a Go to the Part Design workbench, from the menu, select 'Insert > Knowledge Templates > PowerCopy.'

9b Select the Knowledge Pattern feature in the specification tree.



9c From the 'Inputs of components' list, click 'Center_Points' and 'Holes_Numbers' to include them in the 'Selected Components' list.



Creation of PowerCopy is not mandatory. However, if this knowledge pattern is to be reused, PowerCopy provides a convenient interface to select inputs in context of the destination part.

Copyright DASSAULT SYSTEMES

Instructor Notes:

Knowledge Pattern – Script Explanations (1/2)

The following code will explain the method of writing the code for creation of Datums and instantiation of UDF.

```

let l (length)
let i (integer)
let j (integer)
let p (point)
let Spacing (length)
let StartPoint (point)
let udf (Hole_UDF)
let n (integer)

i=1

l=length("Geometrical Set.1\Circle.1")
Spacing=l/[Holes_Number]

Points_List.AddItem("Geometrical Set.1\Origin_Point",1)

i=2
For i while i <=Holes_Number -1
{
    set p = CreateOrModifyDatum("Point",Center_Points ,Points_List ,i)
    p=pointoncurve("Geometrical Set.1\Circle.1", "Geometrical Set.1\Origin_Point",i,Spacing,True)
    p.Name = "Center_Point." + ToString(i)
    p.Show=True
    i=i+1
    i=i+1
}

n=1
For n while n<=Points_List ->Size()
{
    udf=CreateOrModifyTemplate("HoleCatalog\Hole",Body.2 ,Holes_List, n)
    udf.Center_Point = Points_List ->GetItem(n)
    udf.Support_Face="Geometrical Set.1\Surface.1"
    EndModifyTemplate(udf)
    udf.Name="Hole." + ToString(n)
    n=n+1
}
    
```

Variables' Declaration

Use of UDF type defined in the CATGScript file

Loop creations with number of instances to generate

Code for generating Datums

Creation of Datum Point

Code for UDF – Template instantiation using the ARM catalog

Copyright DASSAULT SYSTEMES

Instructor Notes:

Knowledge Pattern – Script Explanations (2/2)

The following code explains the syntax of UDF instantiation and specification of the inputs for the UDF.

```
For n while n<=Points_List ->Size()
{
    udf=CreateOrModifyTemplate("HoleCatalog\Hole" Body.2 ,Holes_List, n)
    udf.Center_Point = Points_List ->GetItem(n)
    udf.Support_Face='Geometrical Set.1\Surface.1'
    EndModifyTemplate(udf)
    udf.Name="Hole."+ToString(n)
    n=n+1
}
```

Instantiation of UDF from the HoleCatalog.catalog file

Specification of the inputs for the UDF

Copyright DASSAULT SYSTEMES

Instructor Notes:

How to Reuse Knowledge Pattern (1/3)



CATIA Data Used: Hole_UDF.CATPart, HoleCatalog.Catalog,
KnowledgePatternSource_Ready.CATPart

Other Files Used: Hole_UDF.CATGScript

Prerequisites for reusing Knowledge Pattern involving UDF instantiation.

For reusing a Knowledge Pattern, you should have created the knowledge pattern, its Power Copy, UDF part and respective catalogs. However, these inputs have been kept ready for you. Following are the prerequisites to reuse the Knowledge Pattern for the example used in this case.

- ① Copy the Hole_UDF.CATPart file to your “c:\temp” directory.
- ② Copy the HoleCatalog.Catalog file to the “\knowledgeResourcesCatalogs” folder of the Knowledge Environment – ‘Architect Resources Creation Path’ (which is specified in the User Settings)
- ③ Open the HoleCatalog.Catalog file and verify that it points to the Hole_UDF feature of Hole_UDF.CATPart that you have stored at “c:\temp”
- ④ Copy the Hole_UDF.CATGScript file to the “\knowledgeTypesCustom” folder of Knowledge Environment – ‘Architect Resources Creation Path’ (which is specified in the User Settings)
- ⑤ In this example, you will be instantiating the knowledge pattern from the source CATPart – KnowledgePatternSource_Ready.CATPart. Save this part to some location and close it.

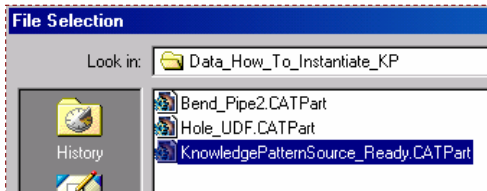
Copyright DASSAULT SYSTEMES

Instructor Notes:

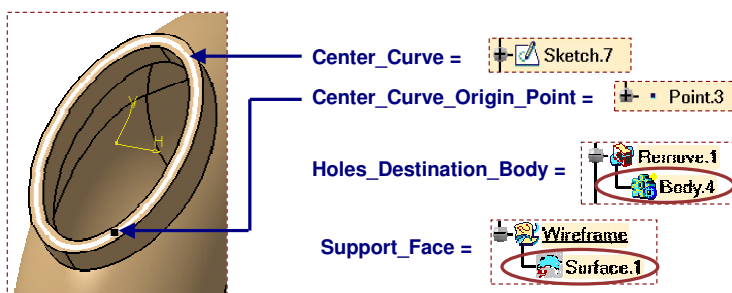
How to Reuse Knowledge Pattern (2/3)

 CATIA Data Used: Bend_Pipe.CATPart

- 1 Open the Bend_Pipe.CATPart and from the menu, select – Insert > Instantiate from Document and select the file “KnowledgePatternSource_Ready.CATPart”.



- 2 Select the inputs in context of the destination part (Bend_Pipe.CATPart) as shown below.

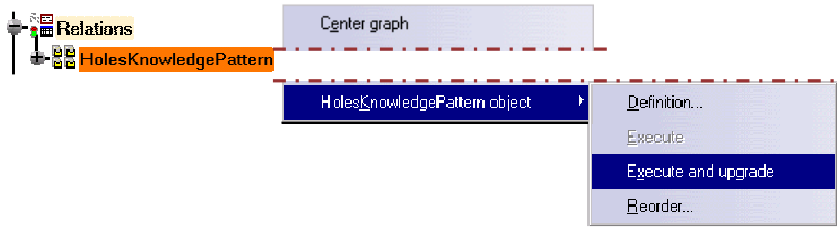


Copyright DASSAULT SYSTEMES

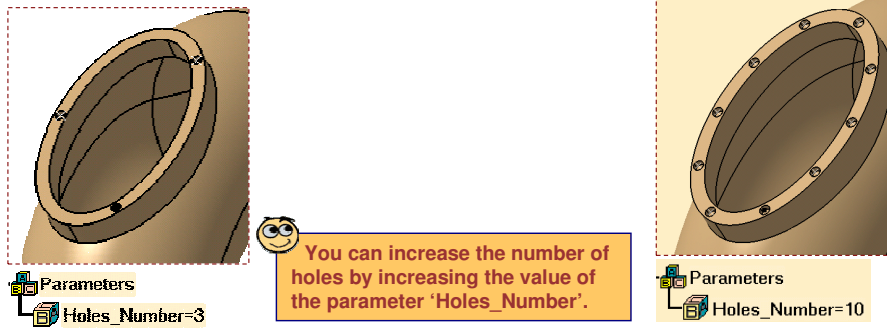
Instructor Notes:

How to Reuse Knowledge Pattern (3/3)

- 3 Knowledge Pattern will be created in the specification tree. Click on the Knowledge Pattern in the specification tree, and from the contextual menu select 'Execute and upgrade'.



- 4 Update the part using the menu Edit > Update and notice the three points and holes that are instantiated.

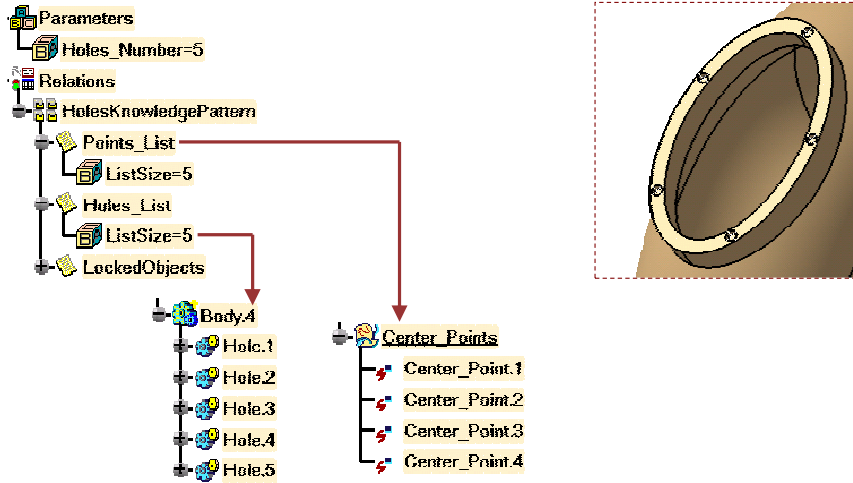


Copyright DASSAULT SYSTEMES

Instructor Notes:

Additional Information – Knowledge Pattern (1/3)

- Knowledge pattern supports the instantiation / creation of the following objects:
 - ◆ User Features
 - ◆ Datums (Planes, Points, Lines, Circles, Curves, Surfaces, and Volumes)
- Lifecycle of the objects instantiated by Knowledge Pattern:
 - ◆ The Lifecycle of the objects instantiated by Knowledge Pattern is controlled through the contents of the 'Lists' that are created in the Knowledge Pattern Feature.

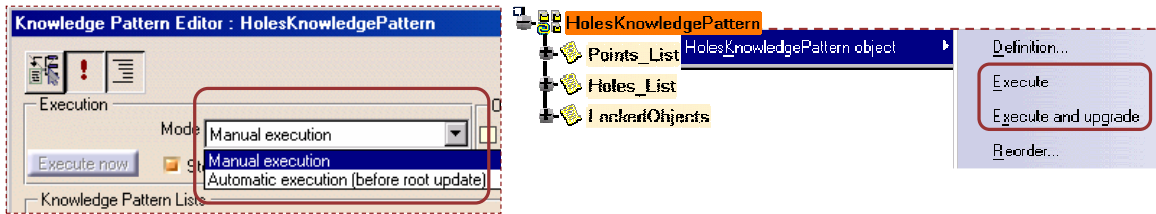


Copyright DASSAULT SYSTEMES

Instructor Notes:

Additional Information – Knowledge Pattern (2/3)

Execution Mode for Knowledge Pattern:



- ◆ In the 'Manual execution' mode, you have to execute the Knowledge Pattern every time you make changes in the code or any parameters related to the Knowledge Pattern.
- ◆ In the 'Automatic execution' mode, the Knowledge Pattern automatically gets executed when you click OK on the 'Knowledge Pattern Editor' dialog box.

Copyright DASSAULT SYSTEMES

Instructor Notes:

Additional Information – Knowledge Pattern (3/3)

Knowledge Pattern assures associativity of the instantiated objects.

- The instantiated objects are associative with respect to:
 - ◆ The inputs defining the Knowledge Pattern
 - ◆ The number of instances
 - ◆ The generated datums
 - ◆ Parameter values of the instances

Copyright DASSAULT SYSTEMES

Instructor Notes:

To Sum Up

In this lesson, you have learned:

- The concept of 'Knowledge Pattern'
- Its applications
- Methods to create and reuse 'Knowledge Pattern'
- Guidelines to write the code of 'Knowledge Pattern'

Copyright DASSAULT SYSTEMES

Instructor Notes:

To Sum Up ...

In this course you have seen:

- How to use PowerCopies
- How to use User Define Features
- How to use Part and Assembly Templates
- How to manage standard components
- How to use advance replication tools like Knowledge Pattern

Copyright DASSAULT SYSTEMES

Instructor Notes: