

Tips and Techniques

Links in CATIA, Part 3: Context Link

Julie Cyrenne, Dassault Systemes

Introduction

In the previous article, we saw how to do design in context, where the child part (also known as contextual part) utilizes elements from the parent part to complete its geometrical definition. As a result, the parent part has no external link while the child part has 2 external link types: import and context. This article will explain when and how the context link is created. All the concepts will be illustrated using the example of a spline.

The Context Link

In CATIA V5, a context link is created when an import link is created in a CATPart. However, there may only be one context link per part, regardless of the number of import links. The context link points from a CATPart (the child) to a CATProduct (the parent).

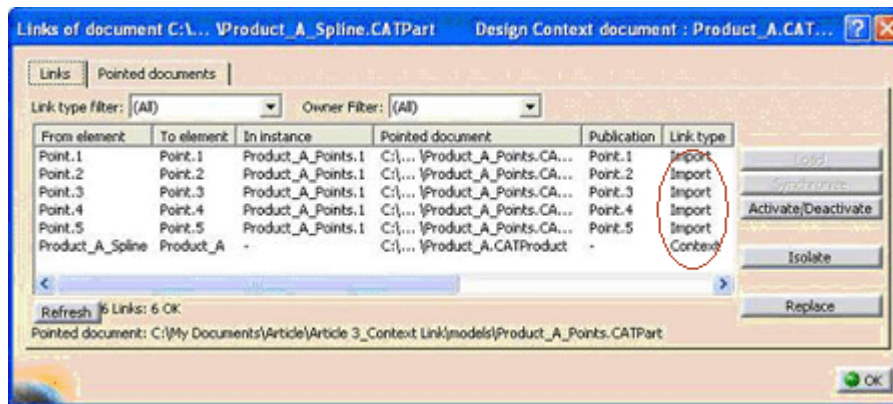


Figure 1: Edit/ Links panel showing 5 import links but only one context link

The context is the product in which a part designed in context is associatively designed. In other words, if this product is opened, all the parent parts of the contextual part (part with import links) will be present and all the links will be resolved. You can also think of the context as the entity which needs to be opened for synchronizing links and updating a contextual part.

Choosing a context

In Tools/ Options/ Infrastructure/ Part Infrastructure/ General, there is one option associated with the context link: 'Use root context in assembly'.

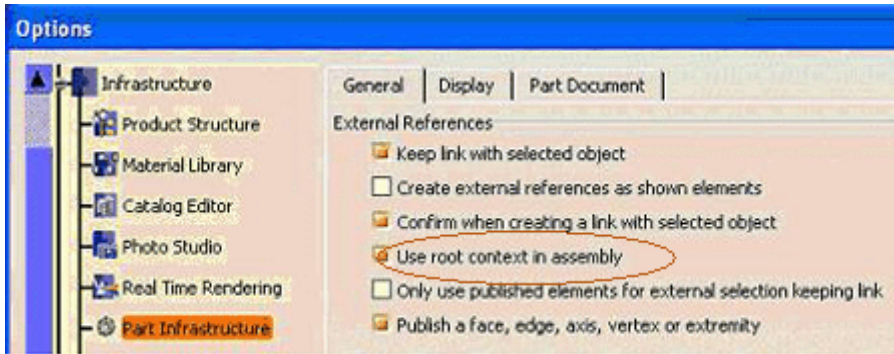


Figure 2: 'Use root context in assembly' option

When the option 'Use root context in assembly' is not selected, the minimum context is used. In other words, when creating a part's first import link, the first product that is a common parent of the two parts involved in the link becomes the context of the contextual part. The two left cases in Figure 3 below illustrate the result of using the minimum context.

When the option 'Use root context in assembly' is selected, the root product opened in session becomes the context of the child part. The right case in Figure 3 below illustrates the result of using the root context.

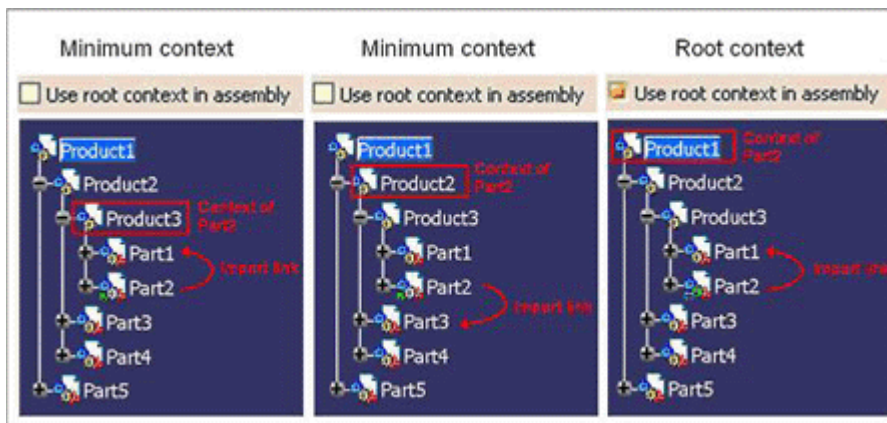


Figure 3: Impact of the 'Use root context in assembly' option

Once the context link is created, it does not change if the contextual part is opened in another configuration (unless explicitly changed by the designer as described in the 'Managing the context' section)

Considerations when choosing a context

- 1) It is impossible to create links with parts outside the context

In the example below, the Profile part contains a sketch of an I-beam profile. The Beam part is a pad using the Profile's sketch, hence creating an import link in the Beam part. The context link of the Beam points on Product2, as the 'use root context in assembly' was not selected. Later, if the designer wants to split the Beam with the Surface, he will get an error message (Figure 5 and Figure 6). Since the Surface part is outside Product2 (the context), it is impossible to create an import link from the Beam to the Surface.

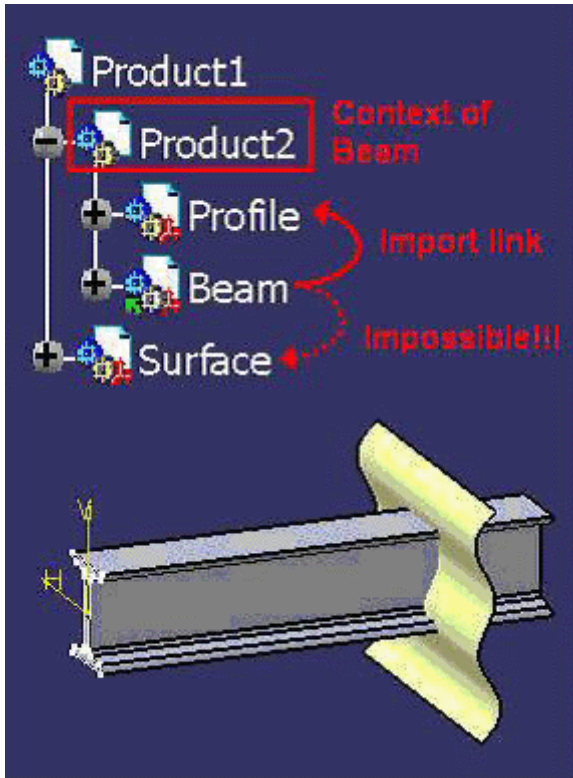


Figure 4: Example for 'impossible to create links outside the context'

Face/Extrude.1/Geometrical Set.1
 Impossible to create an external reference : selection in Part3.1 is forbidden since Part2.1 was designed in context Product2

Figure 5: Pop-up message when trying to select an object outside the context for contextual design

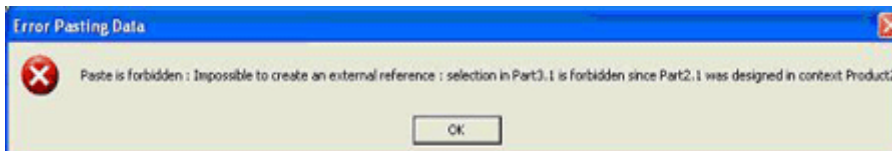


Figure 6: Error message when trying to paste with link an element from outside the context

2) It is impossible to update links if the context is not loaded in session

In the example below, the Cylinder is a pad defined up to the Surface part. The import link points from the Cylinder to the Surface. Since the option 'use root context in assembly' was active when the import link was created, the context link points from the Cylinder to the Product1. If a designer wants to work in Product2, any modification to the Surface will not be represented in the Cylinder because the context product is not in session (even though all the parts involved in the import link are loaded in session).

This example shows how, in a context of concurrent engineering, the root context should be used with caution!

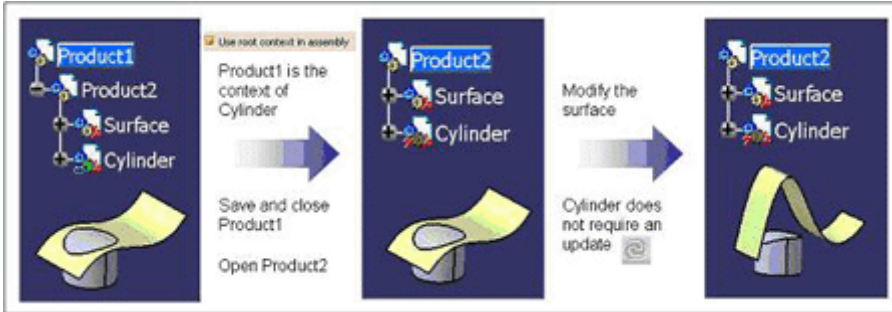


Figure 7: Modification and update of a contextual part outside its context

3) In specific use cases, it may be mandatory to work with a specific context

I don't want to enter into details as this is outside the scope of this article, but an example is when building a 'structure exposed' assembly in ENOVIA, the root context must be used,

Managing the context

It is possible to change the context of a part to a higher level or a lower level product.

Open Right-click on the part whose context should be changed, go to 'Components' and 'Define Contextual Links'. In the panel that appears, simply click on OK. The new context of the part will be the active product in session (regardless of the option 'use root context in assembly')

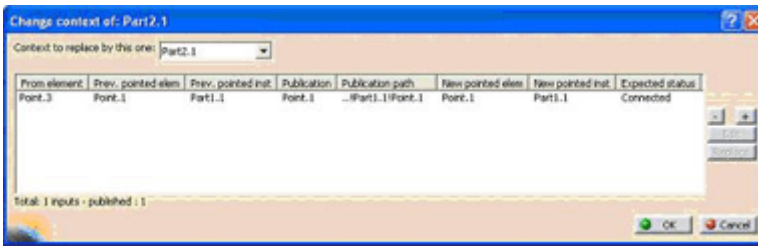


Figure 8: 'Define contextual links' panel

Icons

There are four CATPart icons to inform on the part's context. You may understand their purpose from the images in the examples. The next article will explain the meaning of each icon,

	Non-contextual part
	Definition contextual part
	Instance of the definition contextual part
	Instance of a contextual part

The spline example

Two designers are working in their separate assemblies, Product_A and Product_B.

In Product_A, the first CATPart, Points_A, contains 5 points that are published. The second CATPart, Spline_A, has import links to the 5 points as it uses them for the definition of the spline. Whether the option 'use root context in assembly is enabled or not is irrelevant: because Product_A only has one level, the context link points to Product_A in either case.

Product_B is architected the same way.

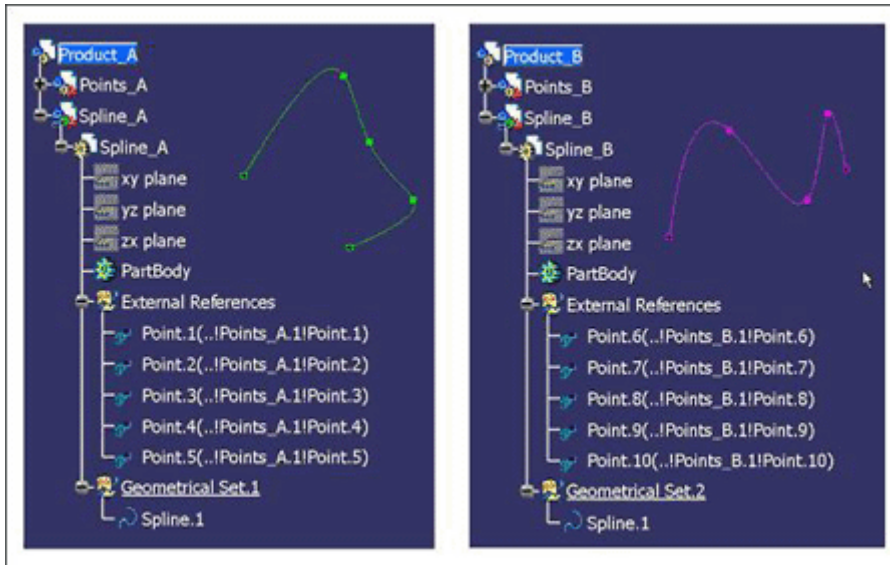


Figure 9: The spline example: designer's initial state

Once both designer's work is done, the two products are inserted in a higher level product, Integration_A-B. The two splines must be connected by using the last point of Point_A as a point for Spline_B. The figure below shows how the software will not allow this operation.

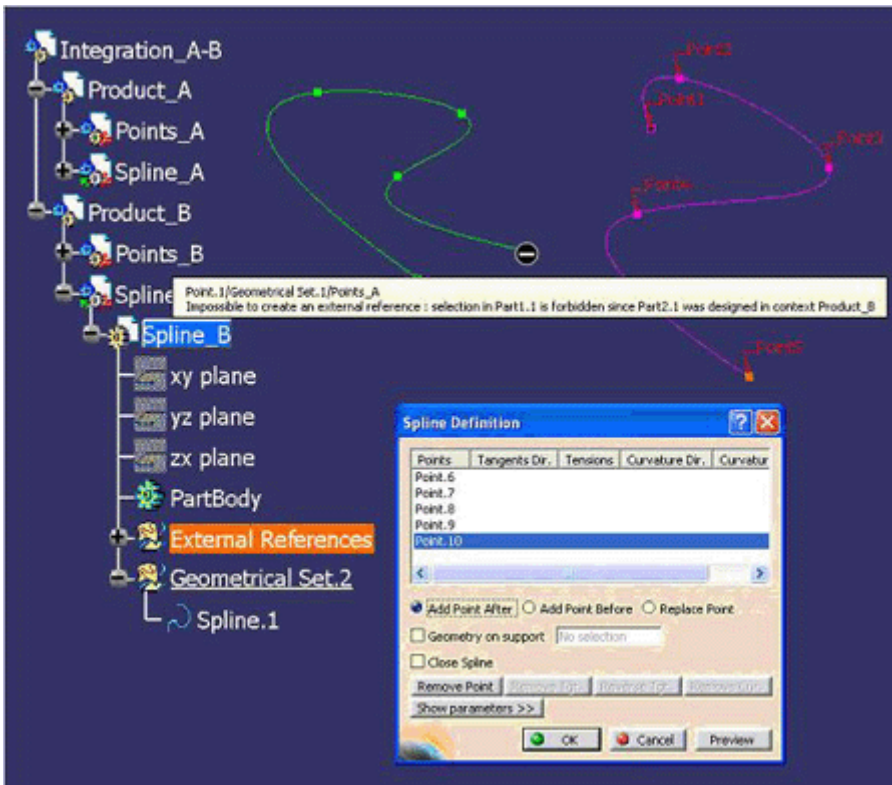


Figure 10: The spline example: Spline_B's context does not allow to import with link elements from Points_A

In order to enable this operation, the context of Spline_B should be changed from Product_B to Integration_A-B using 'Define Contextual Links'. Once the context is changed, Figure 12 shows how the operation will be allowed. The context of Spline_A does not need to be changed: the part does not have any external links outside Product_A to create.

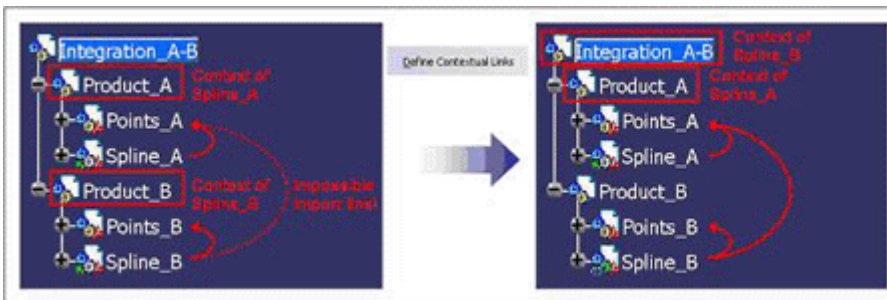


Figure 11: The spline example: changing the context of Spline_B

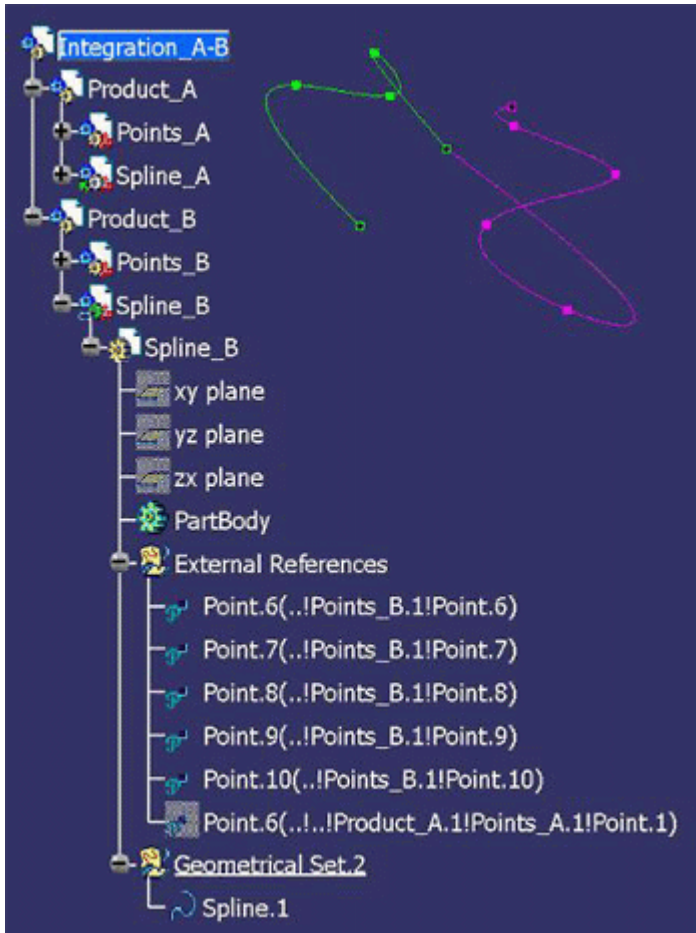


Figure 12: The spline example: Spline_B successfully imported with link an element form Points_A

Conclusion This article explained the context link and the means to manage it. There is no good or bad way: it depends on concurrent engineering, data architectures, etc.

One useful source of information to interpret the context is the CATPart icon. This will be explained in the next article.

Julie Cyrenne
Dassault Systemes