Introduction to Features

Check out this video for a comprehensive introduction. Any single action in CAESES is available as a command. For instance, the creation of geometries and the setting of their attributes are done via the graphical user interface, but for each of these actions there is a corresponding command triggered internally. These commands can be encapsulated and put together by the user in order to have complex or repeating command sequences ready to be executed with a single click ("macro").

All of the possible commands are listed in the documentation browser. CAESES distinguishes between *type* and *global commands*: Type commands allow the creation of objects and the setting of object attributes. Global commands are not related to a specific type. Typical global commands are mathematical operations such as sin(), cos() and sqrt() etc.

Features can be utilized for user-defined functions with loops and if-statements, encapsulation of curve definitions for surface modeling (*meta surface*), reading and writing of ASCII files for proprietary data exchange, COM interfacing, and many other things where customization is required.

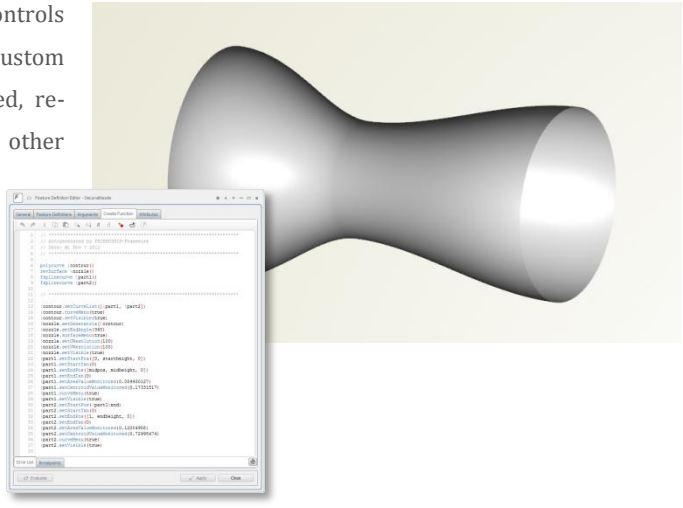
Feature Definitions

Features appear as objects in the object tree. They are based on an individual *feature definition* which holds the following information:

- Arguments: Input data for the feature (values, strings, points, curves, surfaces, …)
- Create Function: User-defined command sequence that processes the input data
- Attributes: Output, i.e. resulting objects, also called *attributes* of the feature

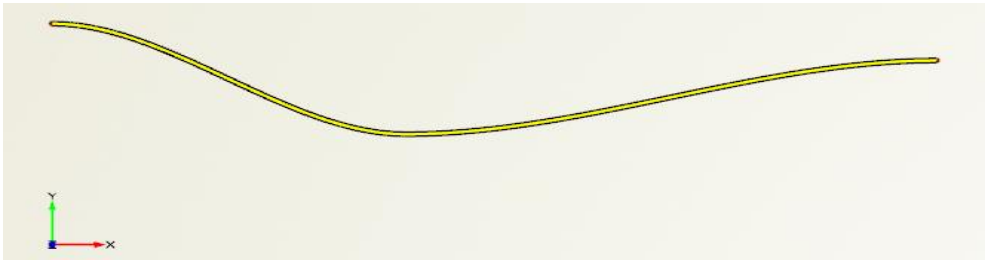
Example: De Laval Nozzle

For a start, this tutorial shows how to create a simple feature definition from scratch: As an example, the *De Laval Nozzle* (also called convergent-divergent nozzle) is modeled first in a simplified manner and then encapsulated in a feature definition. A set of input parameters such as radius, length etc. finally controls the geometry. Such a custom geometry can be documented, re-used and even shared with other users of CAESES.



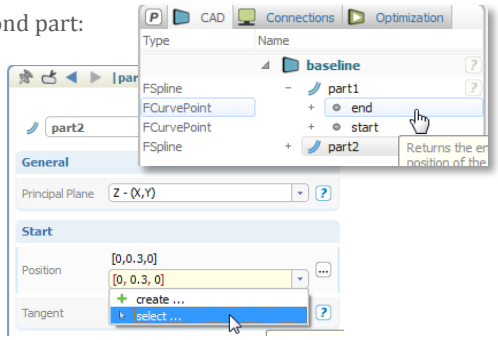
1 Nozzle Contour

The nozzle will be based on a 2D contour that gets rotated so that a surface of revolution is generated. We design the contour curve in a normalized system, e.g. the end position will be located at x=1.

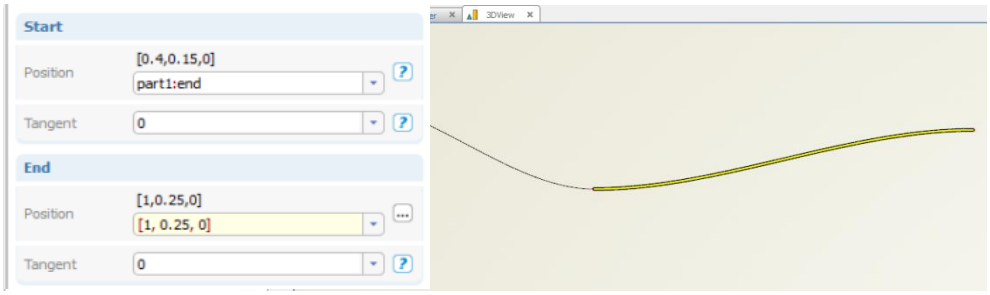


- Click on the z-button at the 3D view in order to switch into the xy-view.
- Create an fspline curve via *CAD > curves > fspline* and call it “part1”.
- Set the start position to [0,0.3,0] and the start tangent to “0” degree.
- Set the end position to [0.4,0.15,0] and the end tangent to “0” degree.

This is the first part of the contour. Let’s continue with the second part:



- Copy and paste the first curve via CTRL-C and CTRL-V.
- Set the start position to be the end position of “part1”: Choose *select* from the position pull-down menu and click onto the end position in the tree (alternatively, drag & drop the end position into the *position* editor).
- Set the end position to [1,0.25,0].
- Check if both tangents are set to “0” degree.



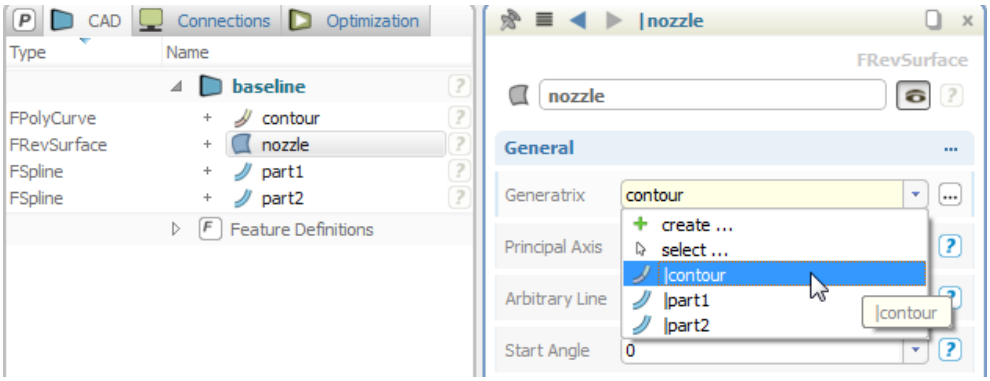
Now, we put the two parts together into a *polycurve* in order to have a single curve defined.

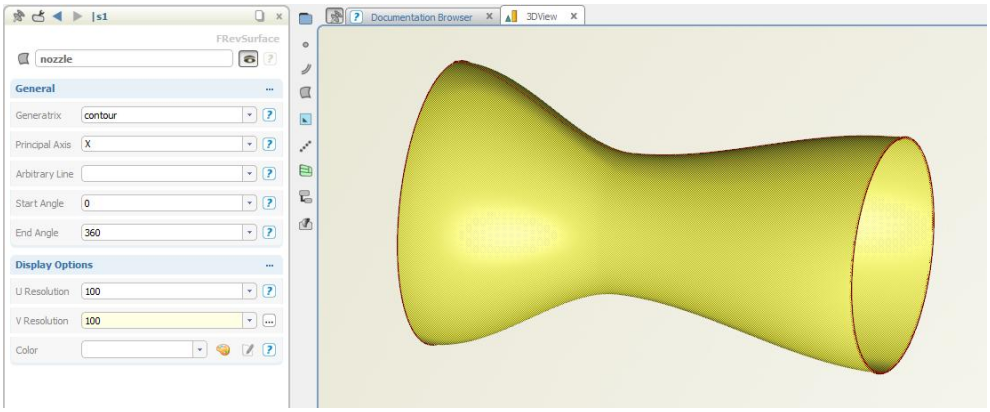
- Select the two curves i.e. first, select “part1” and then “part2”.
- Choose *CAD > curves > polycurve* and call the new curve “contour”.

2 Nozzle Surface

Based on the single curve “contour” we can now create a surface of revolution:

- Choose *CAD > surfaces > surface of revolution*.
- Set the name to “nozzle”.
- Set “contour” as input for *generatrix*.
- Set the start angle to “0” and the end angle to “360” degrees.
- In the category *display options* of the surface, set the *u- and v-resolution* from 20x20 to e.g. 100x100 for a smoother visualization.

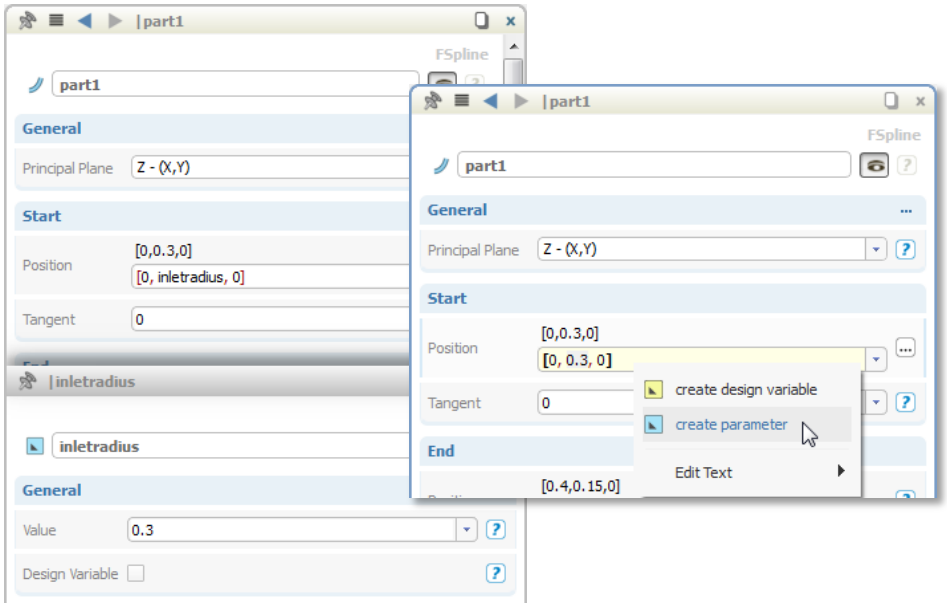




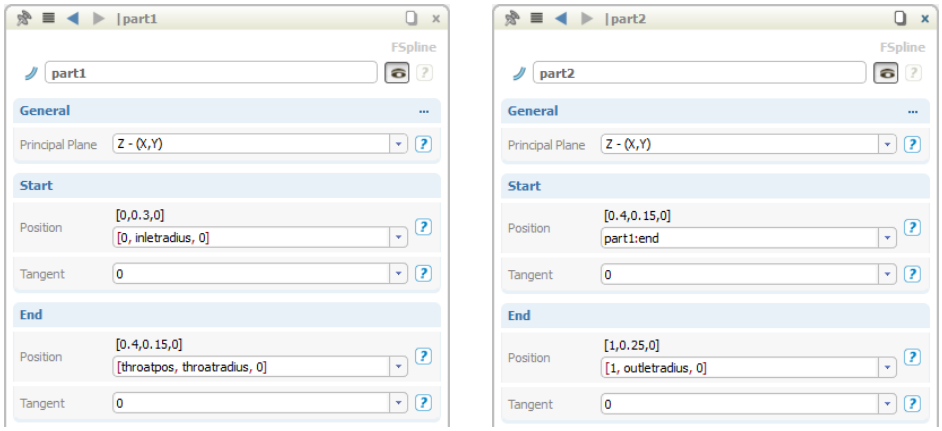
3 Parameters

Before we create a feature definition from this basic model, let's introduce some parameters which will be the input for the feature:

- Select "part1" and create a parameter for the y-value ("0.3") of the start position: Mark the value and choose from the context menu via right mouse click.
- Set the name of the new parameter to "inletradius".



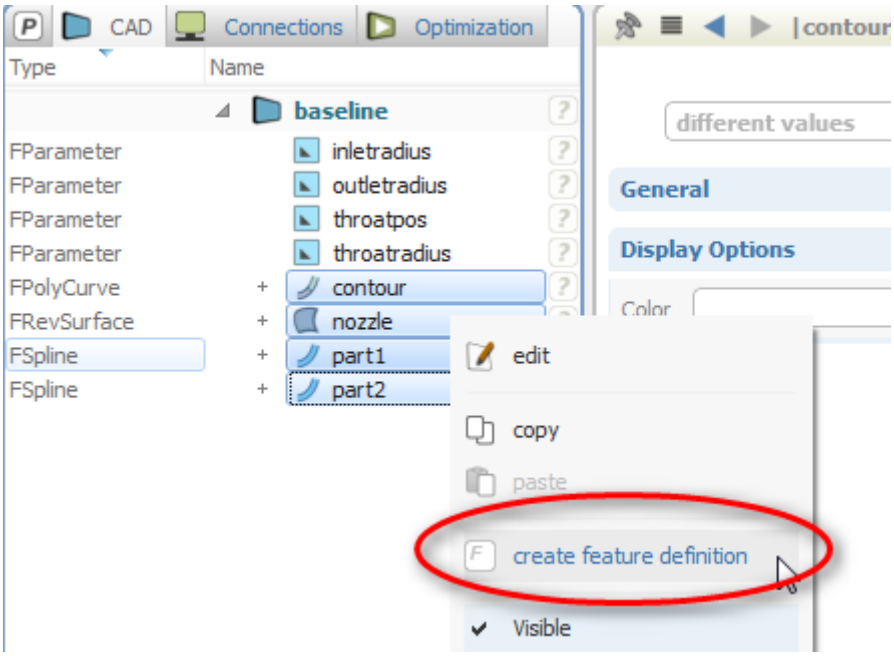
- For the end position of "part1", create two more parameters for the x- and y-value in the same manner. Call the x-parameter "throatpos" and the y-parameter "throatradius".
- For the end position of "part2", create a final parameter for the y-value and call it "outletradius".



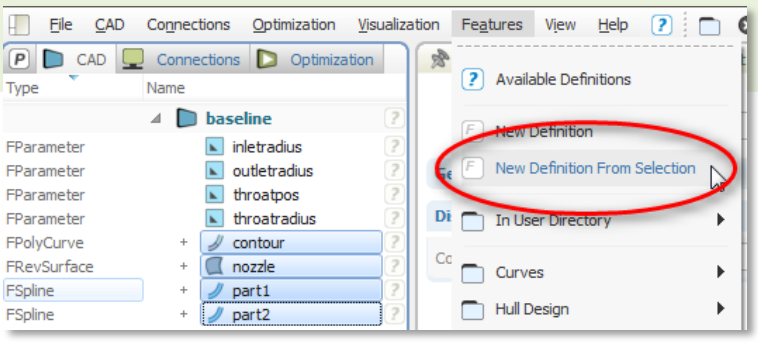
4 Feature Definition from Selection

In this step we will to put the nozzle in a modular feature definition. Anything that will be encapsulated needs to be selected:

- Select all 4 geometry objects but not (!) the 4 parameters – these will not be part of the upcoming feature definition, but will be used as arguments for input.
- From the context menu (right mouse button), choose *create feature definition*.



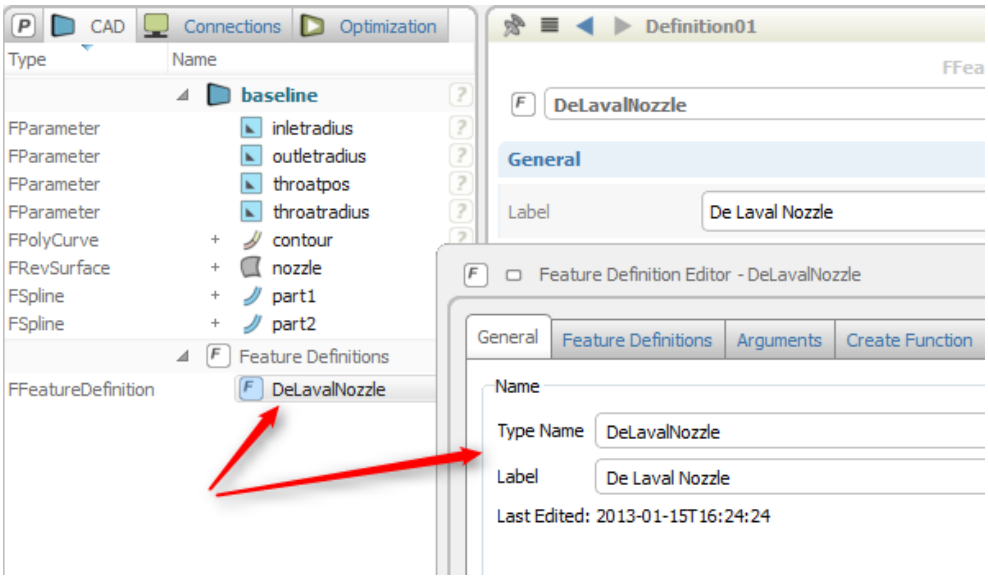
✓ The context menu is a short cut – there is an entry in the menu *Features* as well.

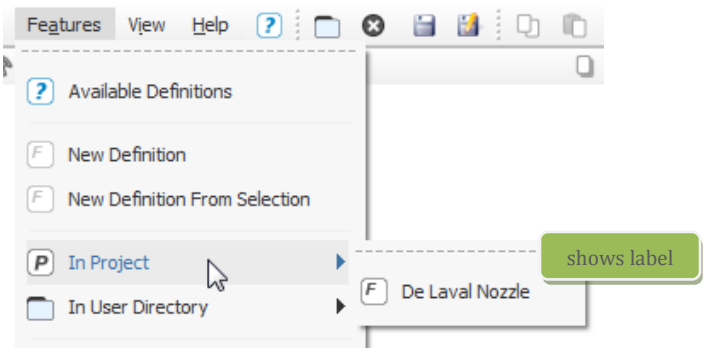


5 Feature Definition Name

The *feature definition editor* of our new definition now is open. Basically, the definition is ready for use but will modify it slightly:

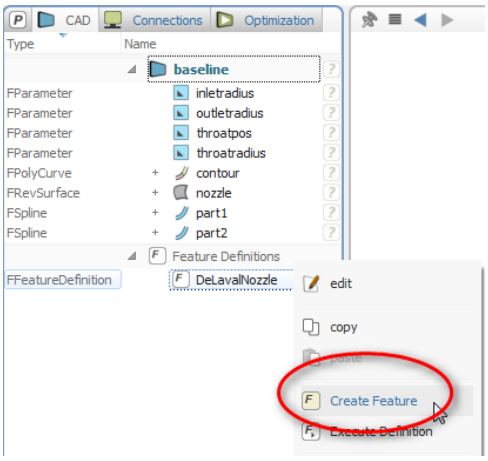
- First of all, click on the *general* tab of the dialog and set the *type name* to "DeLavalNozzle".
- For the attribute *label*, we enter a more readable name using spaces (the label will be shown in the menu *features,* see the second screenshot below).
- Press the *apply* button (at the lower right corner) so that these first changes are applied to the definition.
- Close the dialog. We will have a look at it again in the next steps.





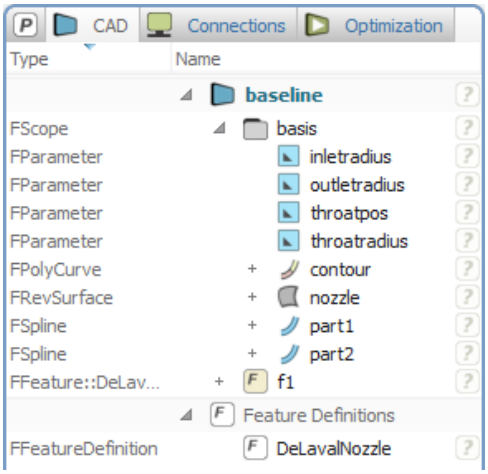
6 Create a Feature from the New Definition

The *feature definition* itself is only an abstract description, similar to a template or macro definition. It is stored in the project and can be accessed in the *CAD* tree. It can also be exported for use in other projects. We will now create an object (instance) that is based on our feature definition:



- In the tree, select the feature definition "DeLavalNozzle" and choose *create feature* from the context menu.

A new feature "f1" gets created in the tree. Since the new feature and the initial geometry are coincident, we want to hide the initial geometry for the time being.

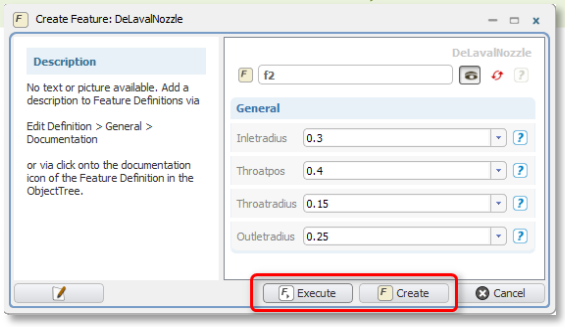


- Select all the geometry and parameter objects apart from the new feature "f1" and choose *CAD > scope.*
- Set the name of the new scope to "basis" and set it invisible by clicking on the scope icon left of the name "basis" in the tree.

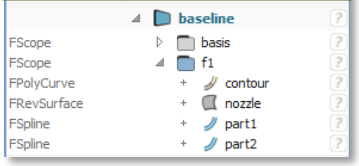
Now, only the feature "f1" is visible in the 3D view.

- Change the input parameters slightly in order to test your feature output.

✓ Alternatively, project feature definitions can also be accessed via the menu *features > in project*. Try it out and choose *create* from the upcoming dialog. Try out the dialog's button *execute* as well: The definition is only executed once and all created objects are given in the tree (we refer to it as "transient" execution, no feature object is then available in the tree).



execute feature definition (result)

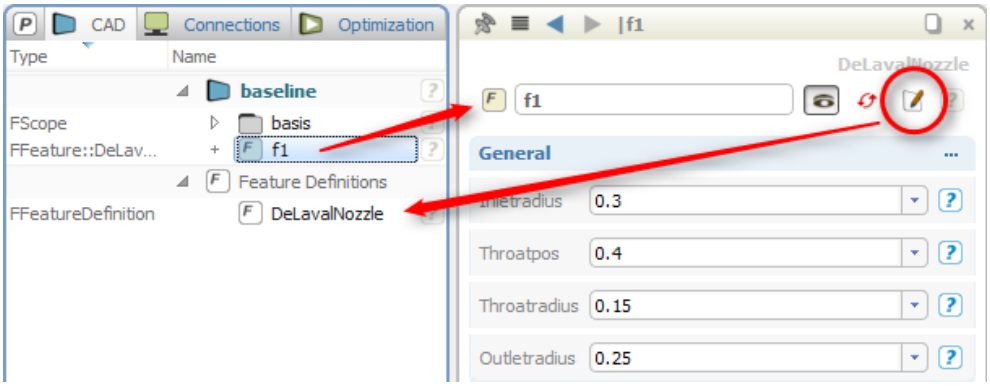


7 Input Arguments

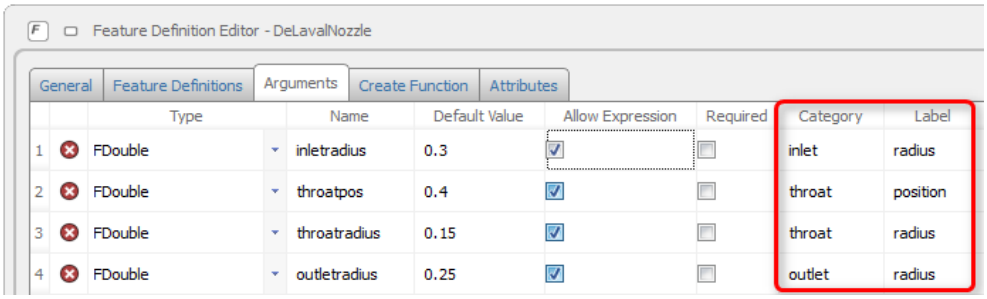
Next we will modify the *arguments* of the feature definition to customize the feature input interface and to have more readable input names.

- Select the feature "f1" and click in the upper right corner of the editor as shown below.

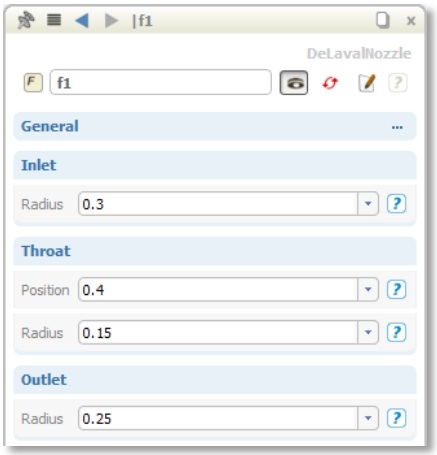
This is a short cut and it always opens the associated feature definition (which in our case is "DeLavalNozzle"). Alternatively, double-click on the definition "DeLavalNozzle" in the tree.



- See the screenshot below, tab *arguments*: Set category names for inlet, throat and outlet.
- Set labels to the input arguments.
- Press the *apply* button.



- Refresh the feature interface of "f1" by e.g. deselecting and selecting it again (see the screenshot for the result: more readable names and a better interface structure, compare it with the first screenshot on this page).

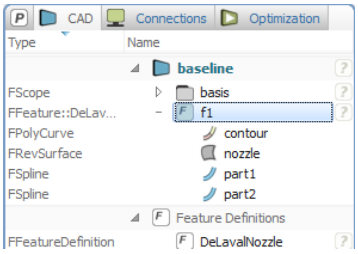


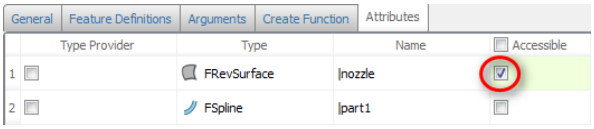
8

Attributes

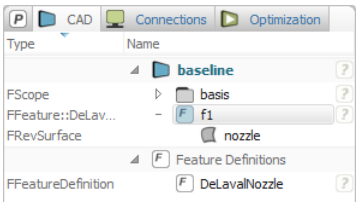
In our case, the attributes of this definition are the contour curves and the final nozzle surface of revolution. Here are some explanations:

- Expand the feature "f1" in the tree to see all attributes.
- In the feature definition dialog of "DeLavalNozzle", activate the *attributes* tab and make only the surface "|nozzle" accessible.

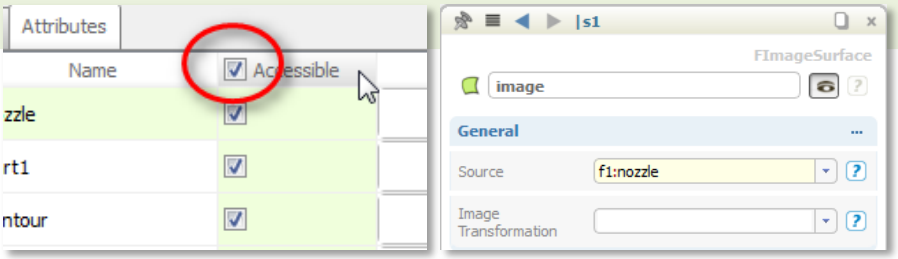




- Press the *apply* button.
- Take a look at "f1" again: only the nozzle surface is now accessible in the tree.



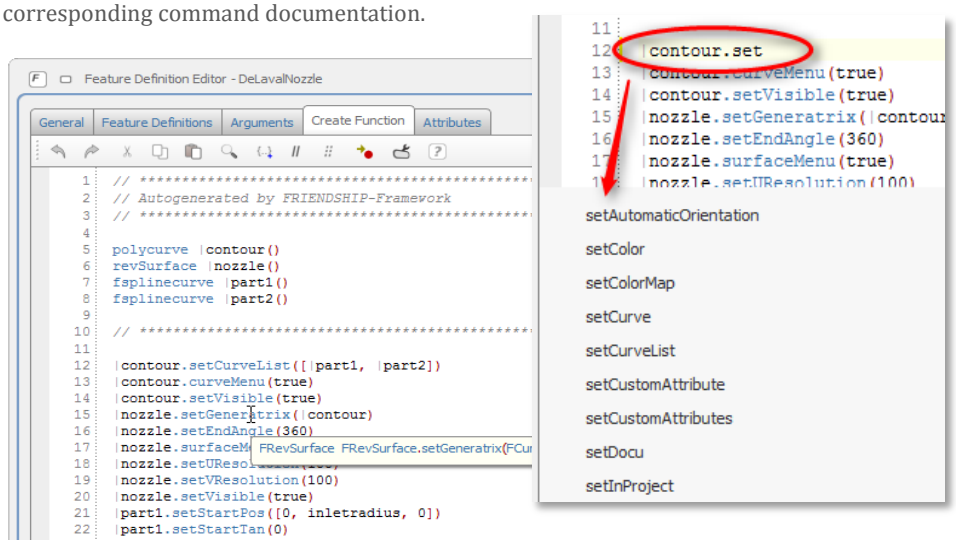
✓ Accessibility is helpful if many auxiliary attributes are created and given in a definition. These attributes can be "switched off" so that only relevant results are visible and accessible. In the header of tab *accessible*, you can check/uncheck all attributes at once. Accessible attributes can be further utilized in subsequent modeling processes such as using drag and drop to connect the nozzle surface to other objects outside of the feature. For instance, if you want to set the nozzle as input for an *image surface (CAD > surface > image surface)*, the source contains "f1:nozzle".



9 A First Look at the Command Sequence

The command sequence, given in the tab *create function* of the feature definition, describes the process based on the input arguments.

- Click on the tab *create function* of the feature definition “DeLavalNozzle”.
- Move the cursor over a command or an object (e.g. “|nozzle”) in order to see the corresponding command documentation.



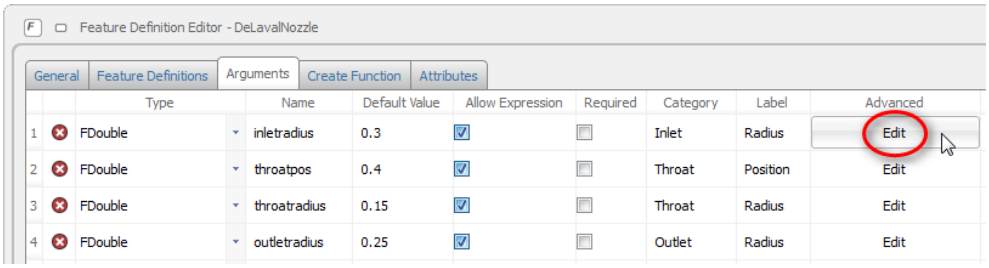
This command sequence can also be written manually and modified by simply changing it. Typically, auto-completion is used for syntax support:

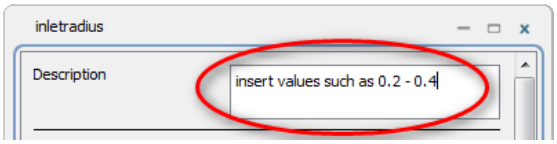
For instance, “contour.set” + CTRL + SPACE provides the list of commands starting with “set”, see the screenshot above.

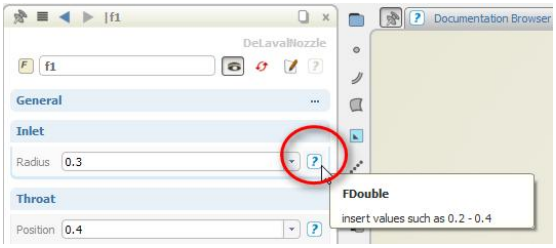
✓ The first block calls default creator commands such as the one of type *FPolycurve*. The command *polycurve* is followed by the name of the new object and parentheses, for instance “|contour()”. The pipe “|” is not relevant and can be skipped for your own written feature definitions (it means root scope). Creators can be called anywhere in the definition, not necessarily at the beginning.

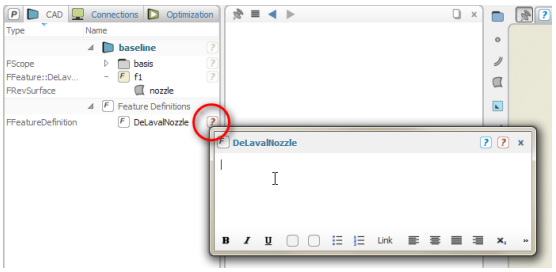
10 Documentation

Let's add some documentation to the definition which is helpful if things are getting more complex or, for instance, if you want to recall the exact meanings of a larger set of input arguments after a while. Here is an example:





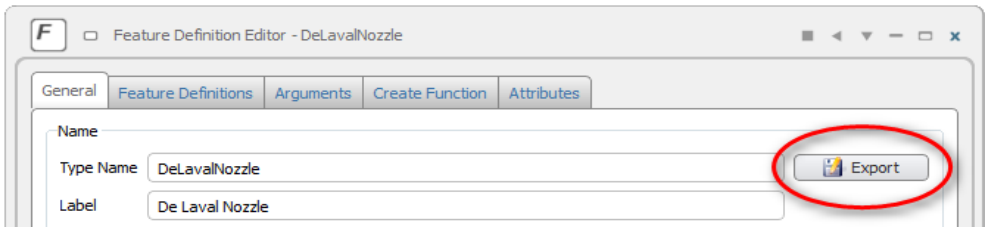




- In the argument list, click the *edit* button of the *advanced* column of e.g. "inletradius".
- Add a short description for this input argument in the *description* field.
- Close the dialog.
- Optionally, do this for the other arguments as well.
- Press the *apply* button and close the feature definition dialog.
- Click on the help icons of the feature argument to see your argument documentation.
- Click on the help icon next to the feature definition "DeLavalNozzle" in the tree in order to add a short description. If you want to add a screenshot or more detailed information, click on the red "?" icon of the editor dialog (Windows: upper right corner / Linux: bottom left corner). Subsequent clicks on the "?" icon in the tree then show your documentation. Mouse over "DeLavalNozzle" shows a tool tip which is also shown in the menu *features > in project*.

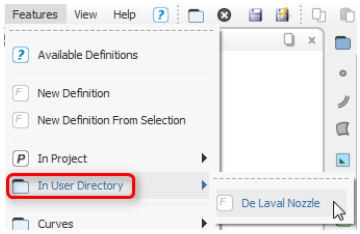
11 Export

Finally, the definition can be exported in order to re-use it in other projects, e.g. for recurrent tasks.

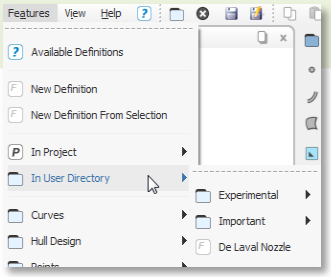


▶ In the feature definition (*general* tab), click on the *export* button in order to save the definition.

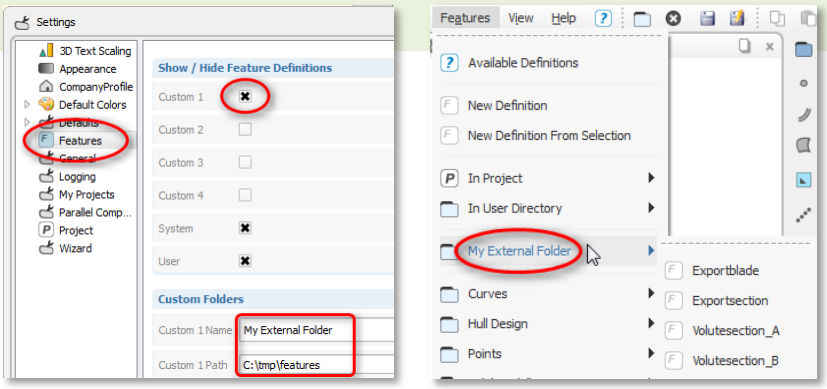
You can now access and import the definition again via the *menu > features > in user directory*.



✓ By default, the CAESES user directory is the target export directory but you can create subfolders in your user directory which will also be shown in the menu.



✓ Involve your own directories: Open the settings via *menu > file > settings* and click on *features*. Activate the toggle of *custom 1* and add a name as well as a path to your feature definitions.



12

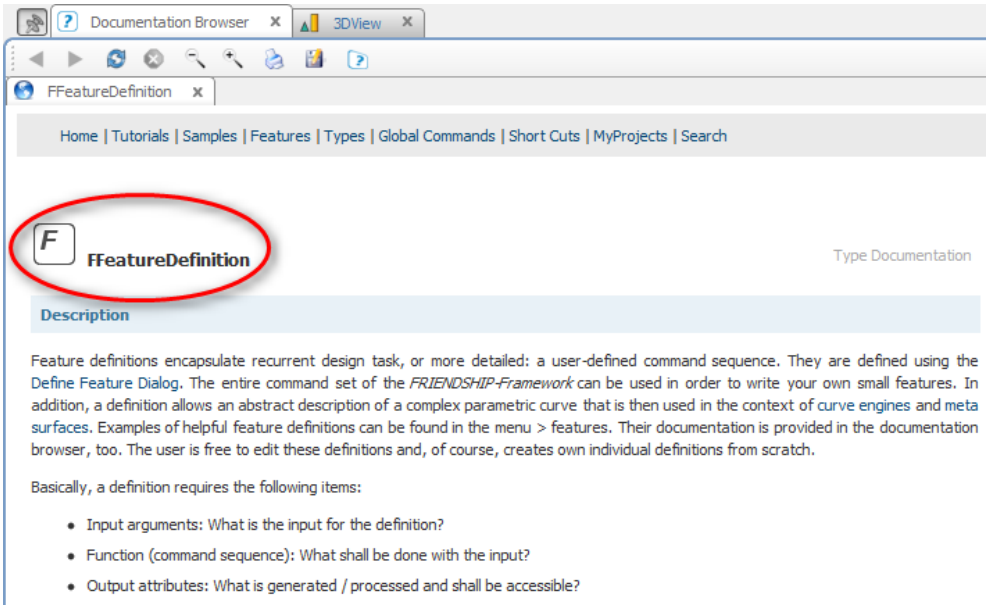
Conclusion

This has been a first introduction to *features*. Basic information is provided about the general structure of feature definitions. A persistent feature gets created based on a definition and the input can be changed manually. Documentation is added to feature arguments as well as to the definition, which is helpful when the feature gets more complex.

Advanced Users

The definition was created from a selection of objects in the tree. For advanced users, the next tutorials show how to write your own custom definitions using control statements like *loops* and *if-else-cases* etc., for which automatic creation from selection is not available. You will also learn more about auto-completion that helps you when writing definitions.

You can find a brief reference of possibilities in the documentation browser, just search for *FFeatureDefinition*:



Features also play a major role in the context of fully-parametric modeling of complex free-form surfaces. See the *meta surface* tutorials section for more information.