Benefits of using User-defined Features to Generate Preliminary Geometry

Liam King 24 October 2017
Introduction to UDFs
User-Defined Features

A User-defined Feature (UDF) is a functionality in Siemens NX that allows a developer to create a feature which combines several NX objects to provide a single object to the end user, with expressions and modelling methods embedded into the features that provide embedded functionality to the user.
User-Defined Features

Expert NX users put together the UDFs to make them as stable as possible over a wide range of parameters. This enables a relatively inexperienced user to assemble the UDFs to represent the geometry of interest to them.

The UDFs are parameterised in a Design Intent fashion, which makes them suitable for geometry optimisation by automated means.

Default values are calculated to ensure first-time instantiation but can be overridden.
UDF Example
Here is a simple hole in a block.

Requirements:
The hole is always in the centre of the face and it’s default diameter is 20% of the height
The width of the block is always 75% of the height – but the user can override this if required
The depth is not to change
It’s origin is a predefined point which must be followed

These rules can be embedded into the UDF
A UDF is created to meet these criteria
UDF Example - Edit

This document is subject to the following Export Control information:

<table>
<thead>
<tr>
<th>Export Classification</th>
<th>UKDL_PL9009.C</th>
<th>18th Oct 2017</th>
</tr>
</thead>
</table>

© 2017 Rolls-Royce plc
Example UDF – Disc, Square Bore

This is an example of a UDF for a disc – note many default values ensuring first-time instantiation.
Example UDF – Disc, Square Bore

The 2 discs shown right were instantiated one after the other – the only thing the end user did was to change the external reference for the annulus line
Example UDF – Disc, Square Bore

This is an example of a UDF for a disc.

The UDF contains several NX Objects, as shown to the right, although the user does not see this.

This UDF is one of the simpler ones: some contain over 100 objects!
User-Defined Features

In the example shown here, the user has assembled 6 UDFs to represent a fan assembly.
User-Defined Features

This actually represents over 150 normal features in NX. Using UDFs, this takes less than 15 mins to assemble. Using traditional methods, it would take over a day.
UDF Updates in a Model

There are 3 ways of modifying a UDF within a model:

1. Interactively by clicking on the UDF to redisplay the UI
2. Import an expression file and regenerate (this can also be used by an iSight optimisation loop)
3. Full NXOpen
Project Deliverables
Project Deliverables

• Library of UDFs to produce preliminary design geometry
• Library mastered in Teamcenter for control and distribution
• Comprehensive Help file for each UDF
• Sample assemblies of UDFs to show how complete components can be produced
• Best practice document for creating UDFs
• Training guide for UDF users
Multiple Disciplines Supported

The product of the UDF assembly can be used in various downstream analysis process:

3D Solid Model      2D Section      3D Sheets
Tagging and Sectioning Tools

The UDFs carry some basic tagging information

A separate part of the project is developing a tagging utility to provide comprehensive tagging capability and a composite section tool to develop non-axi-symmetric features onto the section plane.
End User Guide

A user guide has been produced and published on the Intranet

UDF User Help Guides

**Prelim Design UDF Family User Guide**

This document provides a basic introduction to UDFs and provides guidance for a new user to get the best out of the UDF's. The guide provides initial setup references, and gives hints and tips to help the end user get the best out of the UDF family.

**Mapping the UDF Folder in Teamcenter**

This document provides guidance to the end user as to how to map the UDF family library to their Teamcenter Home collection, and how to add this as a reference to the Reuse Library in NX.

---

This document is subject to the following Export Control information:

<table>
<thead>
<tr>
<th>Export Classification</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>UKL_PL9009.C</td>
<td>18th Oct 2017</td>
</tr>
</tbody>
</table>
Help Files

Every UDF has a comprehensive Help file which is linked to the UDF so that the end user can access the help with a single click. They are also available via the website.

Each Help file describes the feature as a whole, describes the function of each end user parameter and provides a diagram showing what the parameter controls. The Help files also note any known restriction on the feature.
UDF Help Website

Prelim Design UDF Index and Guide

- UDF Index
- What's New
- UDF Assembly Catalogue
- UDF User Guides
- UDF Widgets
- UDF Videos

UDF Listing

Casings
Compressors
Externals
Fans
Power Transmissions
Turbines
Combustion

Casings

UDF Name
Casing, Burner Ring
Casing, Skin, Annulus Ref
Support Struts

Version
1.2
2.1
2.0

Link to Help
UDF.CASING_RING_1.2
UDF.CASING SKIN_
UDF.SUPPORT STRUT

This document is subject to the following Export Control information:

<table>
<thead>
<tr>
<th>Export Classification</th>
<th>18th Oct 2017</th>
</tr>
</thead>
<tbody>
<tr>
<td>UK</td>
<td>UKDL_PL9009.C</td>
</tr>
</tbody>
</table>

© 2017 Rolls-Royce plc
Assembly Guides

Due to the way some of the UDFs work, it may be important to get the sequence of an assembly of UDFs correct in order to get the best final component. Therefore, a series of guides are being developed to take a user through how UDFs can be used to create a particular type of feature.

The example given here is for a Combustion Rear Inner Case in the style of a Trent engine.
Assembly Guides

Subsystem Objective:
Develop geometry feature templates using NX UDF for CRIK that will
- Allow visualization of geometry
- Contain base parameters that allow preliminary (Stage 0 to Stage 2) structural analysis

Notes:
- Number of Casing Skin UDFs and Fixing UDFs depends on the desired CRIK architecture.
- Edge blends, Face Blends and Chambers may be required to tidy up geometry.
- Alternatively, using “Casing Annulus Ref” UDFs may result in fewer UDFs being required and will allow for curved casings. However, annular lines for each section would be required and user would not have parameters to change direction unless built into their annular line sketch.

Example of Trent 1000 CRIK Assembly: TCG016001404453

Trident WORK Camera Trismatic

Export Classification

<table>
<thead>
<tr>
<th>Export Classification</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>UKDL_PL9009.C</td>
<td>18th Oct 2017</td>
</tr>
</tbody>
</table>
UDF Creation Best Practice Guide

A guide has been created during the project to capture the lessons learned with regard to UDF creation, and also lay down some best practices (such as numbering of the UDFs to ensure traceability). This is available in Teamcenter.
UDF Version Control

- Every UDF is version-controlled in Teamcenter
- Each version is given a unique version number
- The version number is visible in the inheriting part and so is traceable
Limitations

- UDFs allow Lists Of Values but the resulting expressions can’t be changed by expression files
- UI can be lengthy and awkward to navigate for complex UDFs
- Help function doesn’t work after a UDF is inserted with Reuse Library (PR7868255)
Case Studies

1. VESTA engine
2. Casing Model
Case Study 1 - VESTA

There is a collaborative project involving companies and organisations outside of Rolls-Royce. In sharing models for testing, given the use of real engine data, there was concern about IP protection.

The decision was therefore taken to create an artificial engine, loosely based on Trent architecture, so that data could be more freely shared. This became known as VESTA (Virtual Engine for Structural and Thermal Analysis). This engine was to be constructed entirely from UDFs.
LP Rotor System – VESTA

This took 1 day to design and model
Comprises of 18 Components
45 UDF’s used
Case Study 2 – Casing Model

An investigation was carried out of the thermal response of a casing produced in one of 3 ways:

• Plain, unfeatured casing
• UDF-based casing
• Fully-featured casing

The results are still being analysed and are shown here purely for illustrative purposes, but they show that the results obtained from the UDF-driven casing are close to the fully-featured casing, but derived in a fraction of the time.
Geometry

- Three versions of the HPC Rear casing used were as follows:

<table>
<thead>
<tr>
<th>Casing Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plain Casing</td>
<td>UDF Based Casing Component, no offtake bosses</td>
</tr>
<tr>
<td>Featured Casing</td>
<td>UDF Based Casing Component, featuring UDF generated offtake and instro bosses to mimic the fully detailed EBOM geometry</td>
</tr>
<tr>
<td>Fully Detailed Casing</td>
<td>Manufacturing standard geometry from EBOM, including offtake and instro bosses, associated bolt holes and fillets (flange bolt holes removed to aid meshing)</td>
</tr>
</tbody>
</table>
Results – Thermal Asymmetry Comparison

Location 1 - Centre of HPC rear casing inner flange

Maximum peak-to-peak asymmetry for whole circumference plotted vs time

Max. Delta = 0.006mm
### Timescales

- Creation of the UDF Featured part is a simple process. The HPC rear casing was created by a novice NX user in approx. 4 hrs;
- Face tagging of the component is, at present a manual process – took approx. 1 hr to tag and check;
- UDF parts have simplified geometry by definition, so require fewer elements to achieve a mesh which adequately describes the geometry, with no idealisation required, and very little mesh control necessary.

- It is estimated that the Fully Detailed manufacturing part would take a minimum of 2 weeks to model;
- Face tagging would be slightly more involved – more of them!
- Certain parts will require defeaturing and idealisation prior to meshing, which can be time consuming in the extreme;
- Some areas require careful mesh control. Identification of these areas is an iterative process, again potentially time consuming;
- More elements required to adequately capture the detail on these components.

<table>
<thead>
<tr>
<th></th>
<th>Modelling</th>
<th>Tagging</th>
<th>Meshing</th>
<th>Update via expressions file</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>UDF Featured Casing</td>
<td>4 hours</td>
<td>1 hour</td>
<td>2 minutes, no errors</td>
<td>Approx. 5 minutes</td>
<td>&lt;6 hrs</td>
</tr>
<tr>
<td>Fully Detailed Casing</td>
<td>2 weeks</td>
<td>2 hours</td>
<td>2 minutes, high potential for mesh failure, thus idealisation required (several additional hours!)</td>
<td>Not possible</td>
<td>&gt;60 hrs</td>
</tr>
</tbody>
</table>

This document is subject to the following Export Control information:

<table>
<thead>
<tr>
<th>Export Classification</th>
<th>UKDL_PL9009.C</th>
<th>18th Oct 2017</th>
</tr>
</thead>
</table>

© 2017 Rolls-Royce plc
Live Demos
Questions?
Demo 1 – HP Drum Build and Edit
Initially build the Compressor Drum, Shaft and Turbine Disc as shown right.
Note OD of drum as 328mm
The UDFs used are shown in the bottom panel.
Demo 1 – HP Drum Build and Edit

The Compressor Drum is scaled by programmatically moving the annulus line upon which it is based to a 50% reduced diameter.

Note OD of drum as 149mm