Knowledge Base

Information

QA Article: QA

Last Update Date: 29.

Rating: Not

Views: 537

Manager dialog template for Abaqus/CAE plug-in development

Portfolio / Domain: SIMULIA Abaqus Unified FEA / n/a

Product: n/a

QUESTION

I am developing a plug-in and would like to create a dialog similar to the native Abaqus/CAE managers, such as the Part, Material or Set manager. How can I do this?

ANSWER

(The following applies to Version 6.7 and higher.)

An Abaqus/CAE application for this purpose is attached below. It provides the foundation for any new plug-in that requires an object manager. A template allows you to quickly develop manager dialogs that handle standard functions (Create, Edit, Copy, Rename, and Delete) of custom objects. Also attached is an example plug-in that demonstrates the use of the template.

The manager dialog provides these features:

- *Create, Edit, Copy, Rename, and Delete buttons* - You need to write the create and edit functions, but the others are taken care of automatically.
- *Automatic updating* - The manager keeps its list up-to-date with its assigned repository.
- *Selection handling* - As you select items in the manager the appropriate buttons are enabled/disabled.
- *Name selection* - The manager determines the name of the item that was clicked by the user and passes that to the management functions.
- *Double-click action* - By default, double clicking on an item in the manager invokes the **Edit** procedure.
- *Automatic cancellation of the current function* - To avoid confusion or errors, the manager does not allow you to execute two management functions at once. If a function is currently active when another function is selected, then the first function will be canceled before activating the second function. If the first function prompts the user to confirm exit, and the user chooses to remain in that function, then the new function will not be activated.

Installation

To install the template application and example plug-in, save the attached archive files to one of the following directories:

> *abaqus_dir*\abaqus_plugins where *abaqus_dir* is the Abaqus parent directory
>
> *home_dir*\abaqus_plugins where *home_dir* is your home directory
>
> *current_dir*\abaqus_plugins where *current_dir* is the current directory

Note that if the `abaqus_plugins` directory does not exist in the desired path, it must be created. The *plugin_dir* directory can also be used, where *plugin_dir* is a directory specified in the `abaqus_v6.env` file by the environment variable **plugin_central_dir**. You can store plug-ins in a central location that can be accessed by all users at your site if the directory to which **plugin_central_dir** refers is mounted on a file system that all users can access. For example,

```
plugin_central_dir = r'\\fileServer\sharedDirectory'
```

On Windows platforms, right click on the archive files and select **WinZip** → **Extract to here**. On Linux platforms, type **unzip caeManagerTemplate.zip** and **unzip shape.zip** at the command prompt. Folders named `abq_CaeManagerTemplate` and `shape` will be extracted, respectively. Note that the plug-in will not function properly if this procedure is not followed.

The files under the `shape` folder are meant as an example and basis for future work. They do not always need to be present if you develop your own plug-in at a later time.

Customization

If you need to customize the behavior of the manager, you can derive your manager from the `ManagerDB` base class and make changes. If you need to perform some special action before invoking a management function, you can overwrite the standard function handlers. The following methods get called once the manager has determined that no other functions are currently running:

- `cmdCreate(self)`
- `cmdEdit(self, name)`
- `cmdCopy(self, name)`
- `cmdRename(self, name)`
- `cmdDelete(self)`

Adding additional management buttons

If you want to add additional management buttons to the manager you can use the dialog's `appendActionButton` method. If you need to enable or disable those buttons depending on the selection in the manager, you can use these methods:

- `getSelections()`
- `setButtonStates(self, numSelections)`

Overwrite the `setButtonStates` method. Then, based on the length of the result returned from `getSelections`, you can decide whether to enable or disable your buttons. Then you would call the base class version of `setButtonStates` to have the rest of the buttons managed by the base class.

Displaying multiple labels

If you want to display more than just the name in the manager, pass in a string containing multiple labels separated by `'\t'` in the manager form. The values shown in a row in the manager are given by the results from the following method:

- `getSummary`

MY FAVORITE CONTENT

You would overwrite that method to return a sequence of sequences: each sequence would contain the same number of items as the manager has columns.

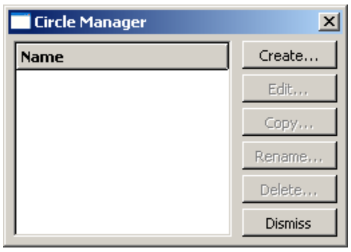
Changing the double-click action

If you want to change the double-click action, you can redefine the following member of the dialog:

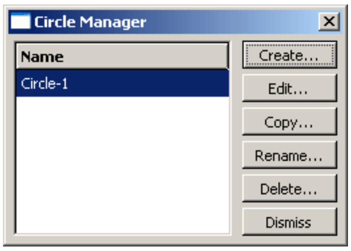
- `self.doubleClickFunction (defaults to self.ID_EDIT)`

Usage

As stated above, the application available in this Answer provides the foundation for a plug-in that requires an object manager. For convenience, an example plug-in is also provided to demonstrate the capabilities of the application. If you have installed the shape plug-in, you can run it by selecting **Plug-ins** → **Tools** → **Circle Manager...** from the Part module. The following dialog is shown:



The manager will be empty if no objects are found. Note that only the **Create** button is enabled at first. In this example, we are creating an object that will have two attributes: name and radius. If **Create** is selected, a new dialog is launched for you to input the name and radius information. A default name is provided for convenience. If **OK** is selected, the object will be constructed and the manager will be automatically updated, as shown in the following dialog:



By selecting **Plug-ins** → **Tools** → **Rectangle Manager...** another manager is shown. If **Create** is selected, a dialog similar to the circle example will be shown, and you can enter name, height and width. A third example is available by selecting **Plug-ins** → **Tools** → **Triangle Manager...**. In this manager, a second column is available to store the hypotenuse information of a triangle. There is also an additional button named **Print** which prints information about the triangle object in the command line interface. The triangle shape example is useful in demonstrating how classes of the manager template can be derived in order to fulfill specific requirements.

The example objects shown above (circle, rectangle and triangle) are stored under `customData`. The manager template is helpful for managing both supported and non-supported objects in Abaqus/CAE. For a detailed example on its use with non-supported objects see Answer VCCT for Abaqus plug-in utility. For a detailed example on its use with supported objects that currently do not have a manager, such as beam orientation, material orientation and Abaqus/Viewer Sets, see Answer Abaqus/CAE plug-in for creating node and element sets in the Visualization module.

Disclaimer

The attachments to this article are subject to certain usage conditions. Please click here for details.

Revision History

10 Sep 08	CAE Manager Template released.
31 Aug 09	Fixed module import order.
19 Aug 11	Fixed expand object name command to work in 6.11.

KEYWORDS **manager, template, gui, customization, dialog, boxes, create, edit, delete, rename, copy, form, repo**

ATTACHMENT

Answer_3825_Fig2.png shape.zip CaeManagerTemplate.zip Answer_3825_Fig1.png
caemanagertemplate.zip answer_3825_fig2.png answer_3825_fig1.png

SUBSCRIBE TO CHANGES ☐

RATING **On a scale of 1-5, how would you rate the technical content of the article?**

Please rate this article...

LET US KNOW IF THIS ARTICLE NEEDS TO BE ENHANCED

UNCLEAR MISSING INFO DUPLICATE OUT OF DATE ERROR DETECTED

MY FAVORITE CONTENT