



Creo Parametric 1.0

Creating Ribbon Tabs, Groups, and Menu Items in JLink

This document shows how to add a simple JLink application to a ribbon within the Creo 1.0 interface. The JLink applications are supported by using the Customize Ribbon tab in the Creo Parametric Options interface. The Customize Ribbon interface saves a layout definition in a ribbon definition file named `tooltribbonui.rbn`. The `.rbn` file is saved by default in the text folder specified in the JLink application registry file (`protk.dat`) from the Customize Ribbon interface.

The JLink API has deprecated the `UIAddButton` method and is replaced using the `UICommand.Designate` method.

Note:

Creo 1.0 requires Java 1.6.0_13 or higher to use for JLink projects or the application fails to load

Add three lines of code in your JLink application.

Step 1. Create Command

In the simple JLink application define the command definition as shown below.

```
UICommand cmd = curSession.UICreateCommand("CreoApp", new MenuButtonListener());
```

Button Listener

The name specified in the `protk.dat` file

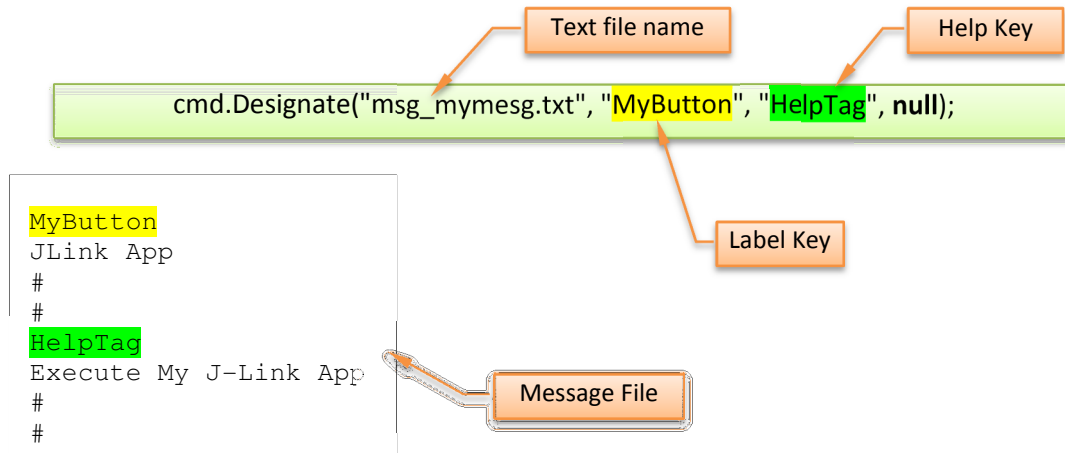
```
name      CreoApp
startup   java
java_app_class  com.ribbon.CreoSynchApp
java_app_start  start
java_app_stop   stop
java_app_classpath  C:\EclipseProjects\WorkspaceCreo\CreoRibbon\jar\creo.jar
allow_stop      true
delay_start     false
text_dir        C:\EclipseProjects\WorkspaceCreo\CreoRibbon\text
end
```

protk.dat File



Step 2. Designate Command

Designate the command by specifying the message file, label, help, and the description as shown below.

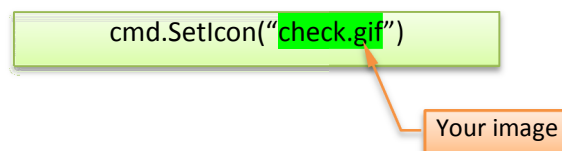


Step 3. Set Icon (Optional)

You can set an image for your button. The image can be either a Creo Parametric .bif file or, a custom .gif file. In order to fit with standard Creo Parametric button sizes, the size of the icon should be at most 20 x 20 pixels. The default search paths for finding the icons are:

- Creo Parametric loadpoint/text/resource
- **Application text dir/resource** ← This option is recommended
- Application text dir/(language)/resource

The location of the application text directory is specified in the registry file.





Now you should have a very simple project with one class, a package, and a JAR file like the one shown below.

```
package com.ribbon;

import com.ptc.cipjava.jxthrowable;
import com.ptc.pfc.pfcCommand.DefaultUICommandActionListener;
import com.ptc.pfc.pfcCommand.UICommand;
import com.ptc.pfc.pfcGlobal.pfcGlobal;
import com.ptc.pfc.pfcSession.Session;

public class CreoSynchApp {

    public static void start() {

        try { // Begin of try block
            Session curSession = pfcGlobal.GetProESession();

            UICommand cmd = curSession.UICreateCommand("CreoApp", new MenuButtonListener());
            cmd.Designate("msg_mymesg.txt", "MyButton", "HelpTag", null);
            cmd.SetIcon("check.gif");

            // End of try block
        } catch (jxthrowable x) {
            System.out.println("Something Wrong");
        } // End of catch
    } // End of Start

    public static void stop() {
    }
} // End of class

class MenuButtonListener extends DefaultUICommandActionListener {

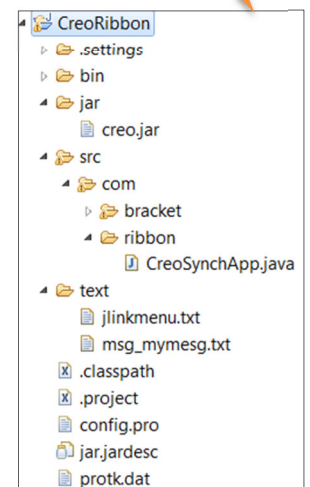
    public void OnCommand()
    {
        try {
            // Get current Session
            Session currSession = pfcGlobal.GetProESession();

            currSession.UIShowMessageDialog("I added a button!", null);
        }
        catch (jxthrowable x) {
        }
    }
} // End of class
```

The 3 steps explained

```
UICommand cmd = curSession.UICreateCommand("CreoApp", new MenuButtonListener());
cmd.Designate("msg_mymesg.txt", "MyButton", "HelpTag", null);
cmd.SetIcon("check.gif");
```

Project Navigator



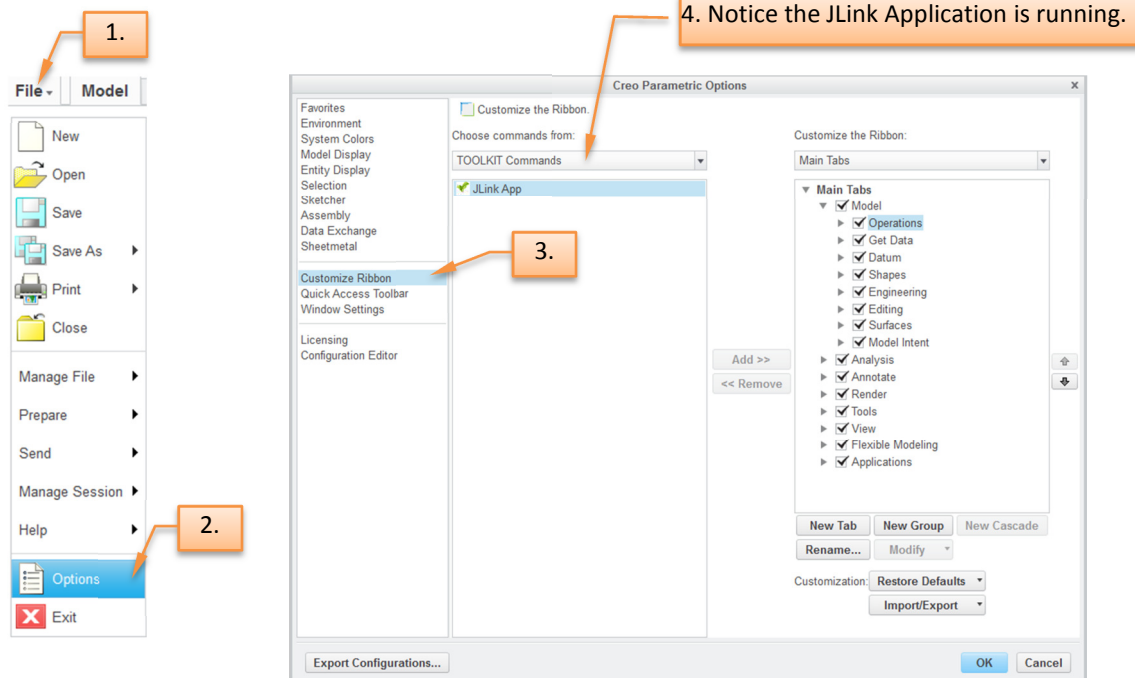


Create a Custom Ribbon & Add Application

We need to create a custom ribbon in Creo 1.0 by using the Customize Ribbon interface. **JLink does not create the ribbon or places the command within a ribbon.** This is done manually by the developer. The next steps will show you how to create a ribbon, group and add your JLink running application to a group. You need to decide what mode the application is used or viewed, in the part, assembly, or drawing. The ribbon definition file is defined in each mode. Therefore, if the application needs to be shown in the part mode, then a ribbon definition file is created in the part mode.

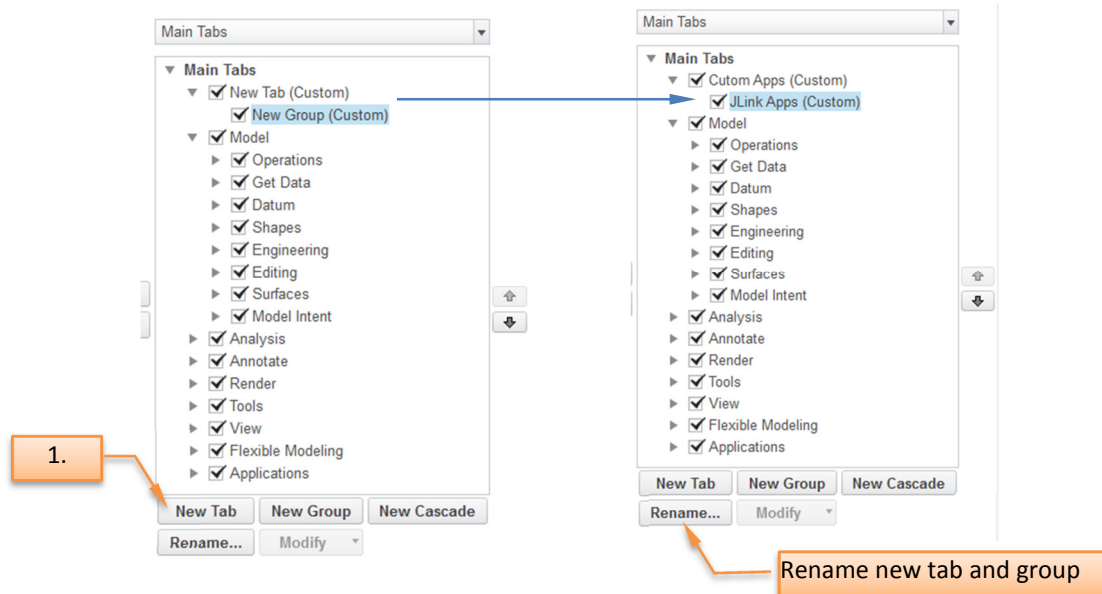
The JLink application needs to be running in order to assign the application to a ribbon.

- A. Open Creo 1.0 & create a new part to get in the part mode.
- B. Go to **File > Options > Customize Ribbon > TOOLKIT Commands**

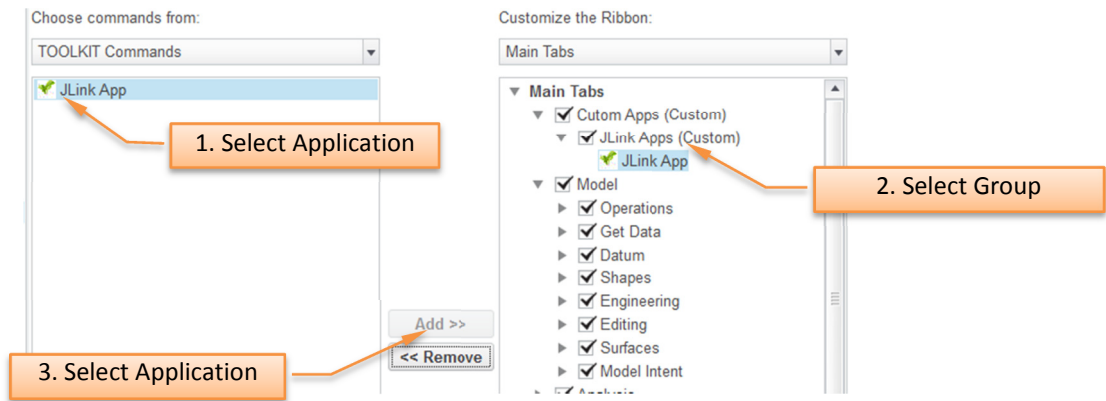




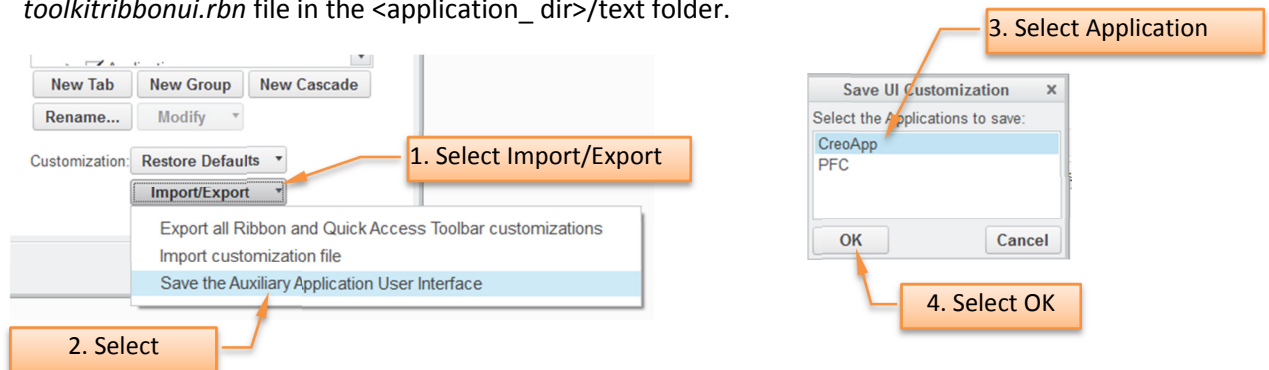
C. Select **New Tab** and rename the tab Custom Apps and group JLink Apps.



D. Select the JLink application > Select the JLink Apps group > Select the **Add>>** button. This will associate the JLink application with the ribbon and group.

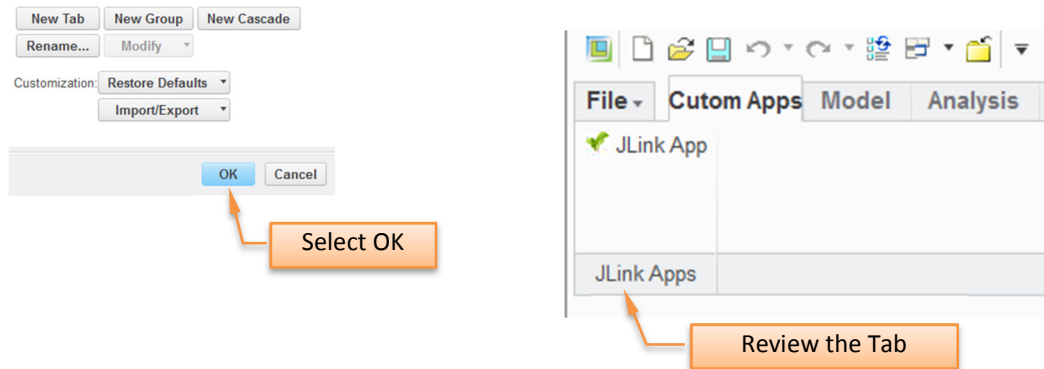


E. Save the ribbon definition file. Select **Import/Export** > Select **Save the Auxiliary Application User Interface** > Select the JLink Application named **CreoAPP**. This will make a *toolkitribbonui.rbn* file in the <application_dir>/text folder.





F. Select **OK** to exit the Custom Ribbon interface and review the tab within Creo!



Close Creo and go into the part mode to review the ribbon.

Recommendation for the JLink application files structure

It is best to move the `tookitribbonui.rbn` file to the `<application_dir>/text/ribbon` folder. This allows the JLink code not to use folder paths to upload the `.rbn` file. All ribbon files should be stored in the application ribbon folder. If more than one application is using the same ribbon and group with various ribbon files, then the ribbon definitions will merge.

The images should be placed in the `<application_dir>/text/resource` folder. This allows the JLink code not to use folder paths to set the image. All images should be stored in the application resource folder.

The recommended folder structure is displayed below.

