NX Ribbon – Customization and Transition

Contents

Overview

The Ribbon interface introduced in NX 9 is fully customizable and supports both file based customizations and interactive customizations.

This purpose of this document is to:

1. Describe when to use each type of customization
2. Describe how to load and deploy each type of customization
3. Detail the new file types and keywords introduced as part of the Ribbon
4. Present options to help with the transition of toolbar content to Ribbon content

File Based Customization

File based customization is preferred for site administrators, 3rd party applications, and any others interested in producing or deploying a consistent UI baseline. In the classic toolbars interface, this is done by creating/modifying MEN and TBR files in order to customize the menus and toolbars. In Ribbon mode, this process is exactly the same except the TBR format has been extended to support the styles and containers introduced in the new interface.

Interactive Customization

Interactive customization is ideal for site administrators who wish to deploy role files and for end user customization. In the classic toolbars interface, all interactive UI changes are stored in user.mtx and can also be saved as a role. These interactive changes only apply to the current application. In Ribbon mode, this process is exactly the same.

The MTX format stores delta changes made on top of the UI baseline established by file based customization. This means that if “Reset” is chosen on any UI element, the MTX content will be removed and the baseline will be restored. The Ribbon has introduced new interactive customization scenarios through item MB3 menus (ex. “Add to Right Border Bar”) as well as enhancing the Tools ▭ Customize dialog to support more in-depth Ribbon customizations (ex. Creating a new tab). All of these new forms of UI customization will be saved in user.mtx when the session exits.

Loading a File

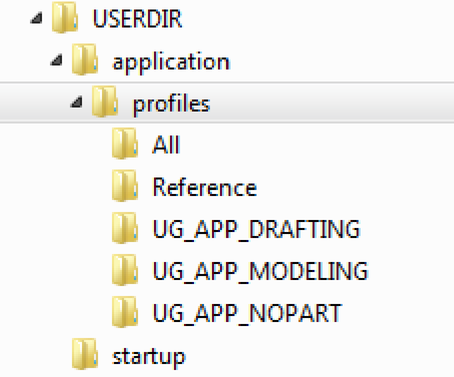
The process of loading file based customizations into NX remains largely unchanged from previous versions. The one notable exception is the new profiles structure. The primary benefit of the profiles structure is to allow TBR formatted files (toolbars and Ribbon files) to be managed on an application level. In previous versions, only toolbars created interactively supported the concept of application specific customization. MEN files have always supported this by modifying an

APPLICATION_BUTTON definition and using the MEN_FILES keyword to specify new MEN files to load for that application.

Here are the ways to load file based customizations:

1. Place file in NX Open “startup” directory
    - Each file in the startup directory will be loaded automatically at the beginning of the session and will be available in all applications.
2. Place file in NX Open “application” directory
    - The files in the application directory will not be loaded automatically but will be available to be loaded through NX Open API.
    - Ex. UF_UI_create_toolbar, UF_UI_create_ribbon
3. Place file in specific directory under NX Open “application\profiles”. There are three types of directories under application\profiles:
    - Application name. Any file in one of the application specific directories will be automatically loaded when that application is entered and automatically unloaded when exiting the application.
        - Ex. application\profiles\UG_APP_MODELING\my_modeling_tab.rtb will be available in Modeling only.
    - “All”. This is similar to placing a file in the startup directory. It will be automatically loaded at the beginning of the session and available in all applications.
    - “Reference”. This is used when a subset of applications want to load the same file. In order for an application to pick up a file in the Reference directory, it must be specified in that application’s DTX file.
        - Ex. application\profiles\Reference\my_shared_tab.rtb can be picked up in Modeling and Drafting by modifying application\profiles\UG_APP_MODELING\UG_APP_MODELING.dtx and application\profiles\UG_APP_DRAFTING\UG_APP_DRAFTING.dtx to include:

```
<Reference>
  <ActionList name="my_shared_tab.rtb"/>
</Reference>
```



Loading an MTX File

The MTX file loading process is the same as in previous versions. The default role is applied first followed by any user customizations from user.mtx. All other role files are loaded and accessible via the Roles palette in the Resource Bar.

1. Default Role is defined by:
   - UGII_DEFAULT_ROLE=my_default_role.mtx
2. User customizations are picked up from:
   - [NX User Data]\user.mtx
3. Roles available from the Roles palette are picked up from:
   - startup\Roles
   - [NX User Data]\Roles

File Types

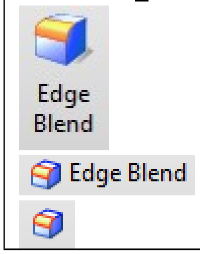
Several new file types have been introduced in NX 9 and all are based on the same syntax as existing TBR files. The new extensions allow the containers to be easily identified.

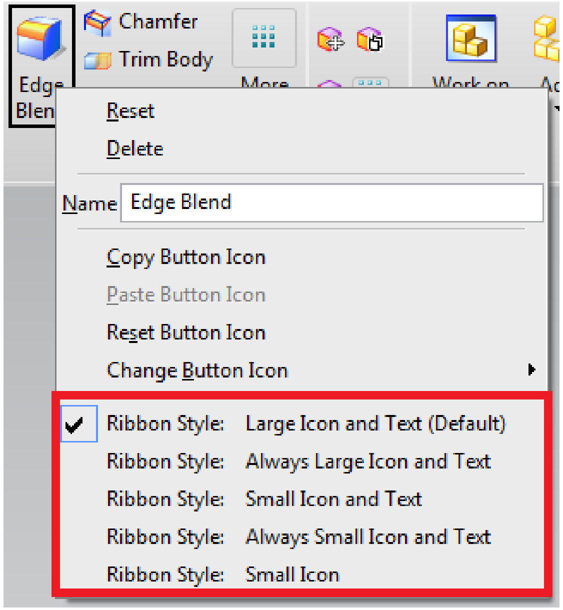
Extension	**UI object**
.tbr	Toolbar
.grb	Group
.gly	Gallery
.ddb	Dropdown
.rtb	ribbon tab
.csb	Cascade
.abr	attachment

Keywords

In addition to the new file types, many new keywords are now used to support the new styles available in the Ribbon containers.

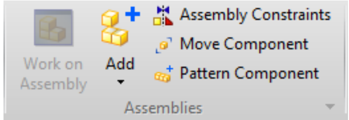
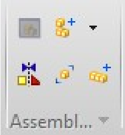
Ribbon Styles

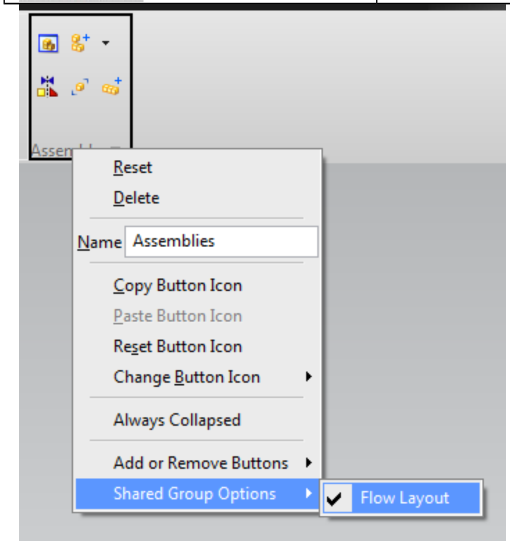
Keyword	Value	Description
RIBBON_STYLE  	ALWAYS_LARGE_IMAGE LARGE_IMAGE ALWAYS_SMALL_IMAGE_AND_TEXT SMALL_IMAGE_AND_TEXT SMALL_IMAGE	Specifies preferred ribbon style. Available for buttons, dropdowns, and cascades. If no style is specified, then the item will start as large image and will shrink down according to the resize rules of the container.



Edge Blen
Chamfer
Trim Body
More
Work on
Ac
Reset
Delete
Name
Edge Blend
Copy Button Icon
Paste Button Icon
Reset Button Icon
Change Button Icon
✓ Ribbon Style: Large Icon and Text (Default)
Ribbon Style: Always Large Icon and Text
Ribbon Style: Small Icon and Text
Ribbon Style: Always Small Icon and Text
Ribbon Style: Small Icon

Groups

Keyword	Value	Description
GROUP	Referenced Item ID	Specifies item that references **group** defined in a separate file. To start a new group in the current file, use BEGIN_GROUP.
BEGIN_GROUP END_GROUP	New Item ID	Specifies group item and its content inside the current file.
GROUP_STYLE  Work on Assembly Add Assembly Constraints Move Component Pattern Component Assemblies 	DEFAULT FLOWLAYOUT	Specifies preferred group style. Default will honor all Ribbon styles determined by each button. Flow Layout will display two rows of small buttons and will honor item style that NX uses for toolbars in Classic mode.

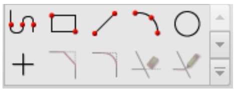
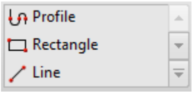


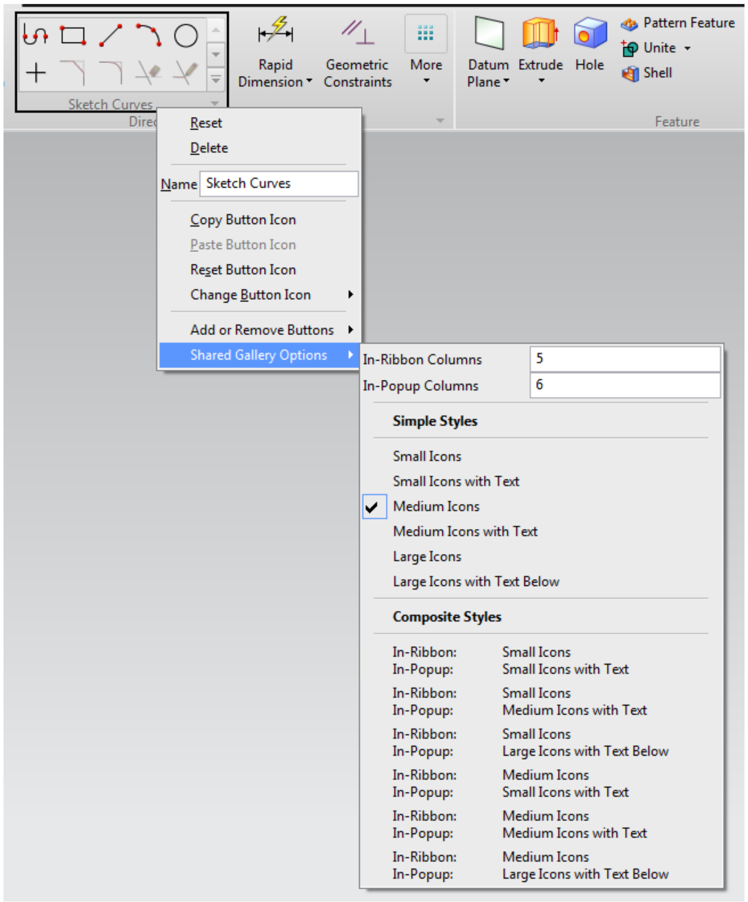
Name Assemblies

☑ Flow Layout

Galleries

Keyword	Value	Description
GALLERY	Referenced Item ID	Specifies item

		that references **gallery** defined in a separate file. To start a new gallery in the current file, use BEGIN_GALLERY.
BEGIN_GALLERY END_GALLERY	New Item ID	Specifies gallery item and its content inside the current file.
GALLERY_STYLE   	**Simple Styles:** SMALL_IMAGE SMALL_IMAGE_AND_TEXT MEDIUM_IMAGE MEDIUM_IMAGE_AND_TEXT LARGE_IMAGE LARGE_IMAGE_AND_TEXT_BELOW **Composite Styles:** SMALL_IMAGE_INPLACE_SMALL_IMAGE_AND_TEXT_INPOPUP SMALL_IMAGE_INPLACE_MEDIUM_IMAGE_AND_TEXT_INPOPUP SMALL_IMAGE_INPLACE_LARGE_IMAGE_AND_TEXT_BELOW_INPOPUP MEDIUM_IMAGE_INPLACE_SMALL_IMAGE_AND_TEXT_INPOPUP MEDIUM_IMAGE_INPLACE_MEDIUM_IMAGE_AND_TEXT_INPOPUP MEDIUM_IMAGE_INPLACE_LARGE_IMAGE_AND_TEXT_BELOW_INPOPUP	Specifies the preferred gallery style.
COLUMN_IN_RIBBON	Number of columns in Ribbon	Defines the largest size that the in-place gallery can expand to when space is available.
COLUMN_IN_POPUP	Number of columns when expanded	Defines the minimum size of the popup gallery.



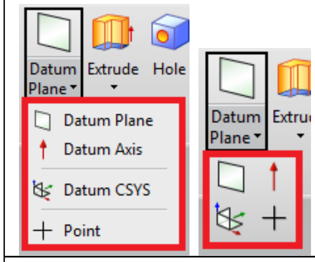
Name: Sketch Curves

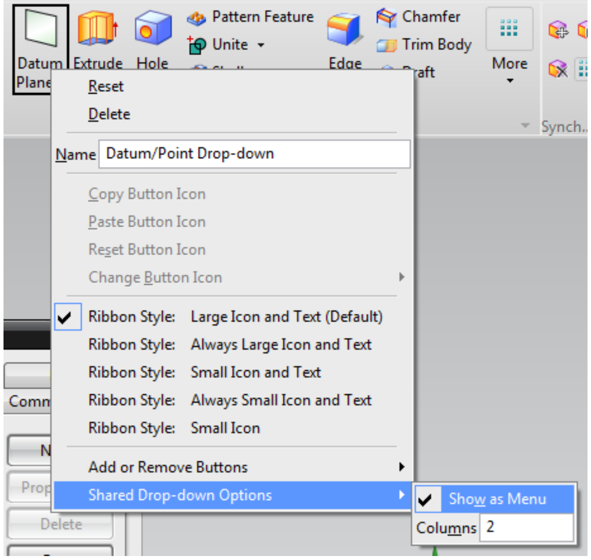
In-Ribbon Columns: 5

In-Popup Columns: 6

☑ Medium Icons

Dropdowns

Keyword	**Value**	**Description**
DROPDOWN	Referenced Item ID	Specifies item that references **dropdown button** defined in a separate file.
BEGIN_DROPDOWN END_DROPDOWN	New Item ID	Specifies cascade item and its content inside the current file.
DROPDOWN_STYLE  Datum Plane Extrude Hole Datum Plane Datum Axis Datum CSYS Point Datum Plane Extru	AS_POPUP_MENU PALETTE	Specifies preferred dropdown style.
COLUMN_DROPDOWN	Number of columns in drop down	



Name Datum/Point Drop-down

☑ Ribbon Style: Large Icon and Text (Default)

☑ Show as Menu

Columns 2

Other

Keyword	Value	Description
CASCADE	Referenced Item ID	Specifies item that references **cascade** defined in a separate file.
BEGIN_CASCADE END_CASCADE	New Item ID	Specifies cascade item and its content inside the current file.
ATTACHMENT_TARGET	TopBackStageGroup LeftBackStageGroup RightBackStageGroup BottomBackStageGroup QuickAccessBar RibbonSystemBar SelectionBar TopBorderBar BottomBorderBar LeftBorderBar RightBorderBar	Used to attach new content to a built-in container or bar. Only to be used in .abr files and is declared at the beginning of the file.
CONTEXT_TITLE	Alternate title	Used for differentiating between groups in Customize dialog
STYLE	DEFAULT TEXTONLY_ALWAYS IMAGE_AND_TEXT TEXTONLY_MENU	Specifies item style for regular toolbars and menus

Transition

NX development has created options to help with the transition of Classic Toolbar content to Ribbon content for both interactive customizations and file based customizations.

Interactive Transition

In NX 9 Ribbon mode, each existing toolbar (TBR) file that is loaded into NX will be converted on the fly to be available as a Ribbon group. Even if no action is taken prior to launching NX, these groups can be found in the Customize dialog on the Commands tab under the "Classic Toolbars (As Groups)" node. This also applies to custom toolbars that were created interactively using the Customize dialog.

Toolbars as groups can then be used in Ribbon mode by dragging them onto Ribbon tabs or border bars. Like all interactive customizations, any changes can be saved in a role file to be used later or distributed.

File Based Transition

The new containers and keywords available in NX 9 have already been described earlier in this document. This section will reference those new containers and keywords and present a few alternate choices when converting a toolbar to the Ribbon. The example below will start with one MEN file containing 5 customer-created buttons and one TBR file that references them. By the end of the example, one Ribbon tab will be created with these same 5 buttons appearing in three separate Ribbon groups. Each of these groups will be created in a different way. The differences will depend on how much Ribbon specific layout each group requires.

Option 1

The first group in the example below represents a Ribbon group that references an existing toolbar with no Ribbon style changes.

This is the quickest way to add existing toolbar content into Ribbon mode but provides no Ribbon specific layout.

Option 2

The second group in the example below represents a Ribbon group that references an existing toolbar with Ribbon specific style changes added to it. There is no difference in how this toolbar is referenced in the .rtb file compared to the first toolbar as a group. The only differences from the first group exist within the .tbr file itself.

This approach has advantages for customers that will deploy both Ribbon and Classic Mode customizations since all the content is defined in the .tbr file and it will stay synchronized in both modes. Only one set of customizations will need to be maintained while still making use of RIBBON_STYLE to control the presentation of the buttons on the Ribbon.

Option 3

The third group in the example below represents a Ribbon group that has been created explicitly as a Ribbon group and will not exist in Classic Toolbar mode. This is the preferred option since the user will be able to take advantage of all the layout styles and containers offered by the Ribbon.

However, for customers that will deploy both Ribbon and Classic Mode customizations, this approach will result in two sets of customizations with no synchronization between them.

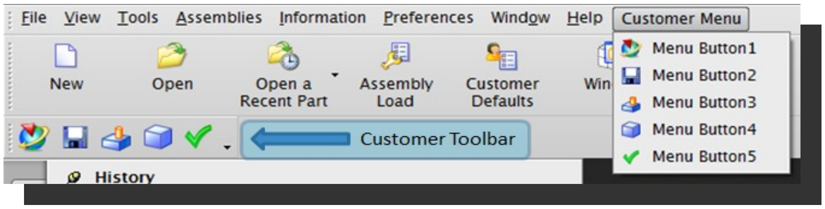
Example

Classic Toolbar mode

Files used:

- customer_men.men
- customer_toolbar.tbr

Screenshot:

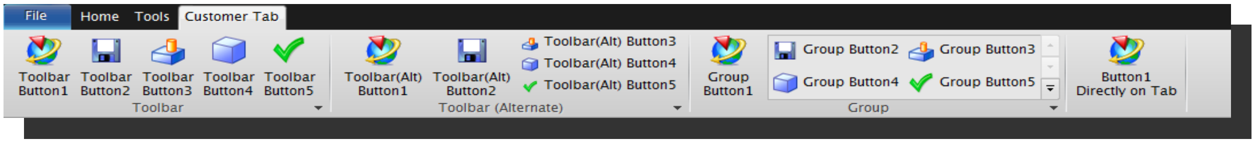


Ribbon Mode

Files used:

- customer_men.men
- customer_tab.rtb
- customer_toolbar.tbr
- customer_toolbar_alternate.tbr
- customer_group.grb

Screenshot:



customer_men.men

```
!  Customer menu file
!
!  Contains button definitions to be used in
!  toolbars and Ribbon files
!

VERSION 120

EDIT UG_GATEWAY_MAIN_MENUBAR

TOP_MENU
    CASCADE_BUTTON CUSTOMER_MENU
    LABEL Customer Menu

END_OF_TOP_MENU

MENU CUSTOMER_MENU

    BUTTON CUSTOMER_BUTTON_1
    LABEL Menu Button1
    BITMAP nx_app

    BUTTON CUSTOMER_BUTTON_2
    LABEL Menu Button2
    BITMAP filesave

    BUTTON CUSTOMER_BUTTON_3
    LABEL Menu Button3
    BITMAP current_feature

    BUTTON CUSTOMER_BUTTON_4
    LABEL Menu Button4
    BITMAP block

    BUTTON CUSTOMER_BUTTON_5
    LABEL Menu Button5
    BITMAP checkgreen

END_OF_MENU
```

customer_ribbon_tab.rtb

```
!
!  Customer Ribbon Tab
!
!  This file will create a new customer tab containing three groups and
!  one top level button.
!

TITLE  Customer Tab
VERSION 170

!  The first group represents a Ribbon group that references an existing
!  toolbar with no Ribbon style changes.
!
!  This is the quickest way to add existing toolbar content into Ribbon mode.
!
GROUP  customer_toolbar.tbr

!  The second group represents a Ribbon group that references an existing
!  toolbar with Ribbon specific style changes added to it.  There is no difference
!  in how this toolbar is referenced in the .rtb file compared to the first
!  toolbar as a group.  The only differences from the first group exist within
!  the .tbr file itself.
!
!  This approach has advantages for customers that will deploy both Ribbon and
!  Classic Mode customizations since all the content is defined in the .tbr
!  file and it will stay synchronized in both modes.  Only one set of customizations
!  will need to be maintained while still making use of RIBBON_STYLE to
!  control the presentation of the buttons on the Ribbon.
!
GROUP  customer_toolbar_alternate.tbr


!  The third group represents a Ribbon group that has been created explicitly
!  as a Ribbon group and will not exist in Classic Toolbar mode.  This is the
!  preferred option since the user will be able to take advantage of all the
!  layout styles and containers offered by the Ribbon.
!
!  For customers that will deploy both Ribbon and Classic Mode customizations,
!  this approach will result in no synchronization and two sets of customizations
!  will need to be maintained.
!
!  A group can alternatively be defined directly inside the .rtb file by using the
!  keywords BEGIN_GROUP and END_GROUP.
!
GROUP  customer_group.grb

!  Example of a button that is placed directly on the Ribbon without being
!  contained in a group.  This is a simple way to add a button to a tab
!  but is not recommended for many buttons since groups have advantages.
!
!  A few advantages of groups:
!     1. Easier to move many buttons around together in quick customization.
!        For example, MB3 on a group and choose "Add to Left Border Bar".
!        This will move the whole group to a new location.  If buttons are
!        not in a group, they need to be moved one at a time.
!
!     2. Provide a label to describe its contents.
!
BUTTON CUSTOMER_BUTTON_1
LABEL Button1 Directly on Tab
```

customer_toolbar.tbr

```
!
!  Customer Toolbar File
!
!  This is similar to an existing toolbar file containing
!  no Ribbon changes.
!
!  In Classic Toolbars mode - This toolbar will still show up in the
!     toolbars list.
!
!  In Ribbon mode - This toolbar will be automatically converted to a
!     Ribbon group and will appear in the "Classic Toolbars as Groups"
!     section in the Customize dialog.
!

TITLE Toolbar
VERSION 170

BUTTON CUSTOMER_BUTTON_1
LABEL Toolbar Button1

BUTTON CUSTOMER_BUTTON_2
LABEL Toolbar Button2

BUTTON CUSTOMER_BUTTON_3
LABEL Toolbar Button3

BUTTON CUSTOMER_BUTTON_4
LABEL Toolbar Button4

BUTTON CUSTOMER_BUTTON_5
LABEL Toolbar Button5
```

customer_toolbar_alternate.tbr

```
!
!  Customer Toolbar File (Alternate)
!
!  This is similar to an existing toolbar file with new
!  Ribbon styles added.
!
!  The RIBBON_STYLE will be honored when the toolbar is loaded
!  in Ribbon mode as a group but will be ignored when the toolbar
!  is displayed in Classic Toolbars mode.
!
!  In Classic Toolbars mode - This toolbar will still show up in the
!     toolbars list.
!
!  In Ribbon mode - This toolbar will be automatically converted to a
!     Ribbon group and will appear in the "Classic Toolbars as Groups"
!     section in the Customize dialog.
!

TITLE Toolbar (Alternate)
VERSION 170

BUTTON CUSTOMER_BUTTON_1
LABEL Toolbar(Alt) Button1
RIBBON_STYLE LARGE_IMAGE

BUTTON CUSTOMER_BUTTON_2
LABEL Toolbar(Alt) Button2
RIBBON_STYLE LARGE_IMAGE

BUTTON CUSTOMER_BUTTON_3
LABEL Toolbar(Alt) Button3
RIBBON_STYLE SMALL_IMAGE_AND_TEXT

BUTTON CUSTOMER_BUTTON_4
LABEL Toolbar(Alt) Button4
RIBBON_STYLE SMALL_IMAGE_AND_TEXT

BUTTON CUSTOMER_BUTTON_5
LABEL Toolbar(Alt) Button5
RIBBON_STYLE SMALL_IMAGE_AND_TEXT
```

customer_group.grb

```
!
!  Customer Ribbon Group File
!
!  In Classic Toolbars mode - This group will be unavailable
!
!  In Ribbon mode - This group will show up in the "All Groups" list
!     in the Customize dialog.
!

TITLE Group
VERSION 170

BUTTON CUSTOMER_BUTTON_1
LABEL Group Button1
RIBBON_STYLE LARGE_IMAGE

!  Example of a gallery defined in a GRB file.
!
!  If this gallery is to be used in other places, then it could be
!  defined in its own gallery (GLY) file and referenced here
!  Example:
!    GALLERY customer_gallery.gly
!
BEGIN_GALLERY  CUSTOMER_GALLERY
COLUMN_IN_RIBBON 2
COLUMN_IN_POPUP 2
LABEL Gallery
GALLERY_STYLE MEDIUM_IMAGE_AND_TEXT

    BUTTON CUSTOMER_BUTTON_2
    LABEL Group Button2

    BUTTON CUSTOMER_BUTTON_3
    LABEL Group Button3

    BUTTON CUSTOMER_BUTTON_4
    LABEL Group Button4

    BUTTON CUSTOMER_BUTTON_5
    LABEL Group Button5

END_GALLERY
```