

GJax Workspace

From Intranet

Table of contents

- 1 gJax Workspace
 - 1.1 Main Menu
 - 1.2 Folder Panel
 - 1.3 Welcome Screen
 - 1.4 Toolbar
 - 1.5 Context Menus
 - 1.6 Folder Panel and Message Panel
 - 1.7 JavaScript

gJax Workspace

When you run gJax for the first time it launches the standard Workspace. Initially the workspace starts in it's default configuration. You can override many of the features of the standard workspace to give you application it's own identity.

The workspace comprise the following main elements:

- Menu Bar - a standard menu bar running across the top of the workspace (mnuMain)
- Toolbar - a toolbar running horizontally immediately below the menu bar (toolbar.xsl)
- Main Panel - a panel below the toolbar where the main application runs (divMain)
- Folder Panel - a resizable panel glued to the left hand side of the workspace (mnuFolder)
- Message Panel - a resizable panel glued to the bottom of the workspace (divMessages)

Main Menu

The main menu can be overridden with an application specific main menu. First create the menu that you want to use. We recommend that you call it xxx.mnuMain (where xxx is a short prefix that identifies your application). Go to the Setup/Menu folder, open then menu called mnuMain and at the prompt 'Override this menu with the following menu:' enter the name of your application specific menu. Normally you should override it 'System-wide'. But if you are in the process of developing the menu or just experimenting, you may want to override it just for 'This user only'. In this case the new menu will only apply to sessions that are logged in with your user account.

Overriding menus for 'This user only' can also be a way of providing personalised menus for different users.

When you override the main menu you will need to disconnect and reconnect for the change to take effect.

Folder Panel

The folders displayed in the left hand folder panel can be override with an application specific set of folders. Folders are configured in exactly the same way as menus. First create the folder that you want to use. We recommend that you call it xxx.mnuFolder (where xxx is a short prefix that identifies your application). Go to the Setup/Menu folder, open the menu called mnuFolder and at the prompt 'Override this menu with the following menu:' enter the name of your application specific folder. Normally you should override it 'System-wide'. But if you are in the process of developing the menu or just experimenting, you may want to override it just for 'This user only'. In this case the new menu will only apply to sessions that are logged in with your user account.

When you override the top level folder you will need to disconnect and reconnect for the change to take effect.

Welcome Screen

The default welcome screen is an html page that is displayed in the main panel when the workspace is first started. By default it displays gJax product information and logo. This can be replaced with an application specific welcome screen.

- First take a copy of the file 'welcome.htm' from the gjax directory and place it in your application directory.
- Next customise it to suit the needs of your application. Note that there are a few lines of html that need to be retained in order for gJax to be able to manipulate it.
- To override the default welcome screen set the following global node in your application namespace:

```
>set ^%vcvc("welcome")=" ../welcome.htm"
```

Note that the ../ is required because gjax runs in a sub-directory of your application directory and so the ../ is needed to refer to gjax's parent directory.

When you override this setting you will need to restart the workspace for the change to take effect.

Toolbar

The default toolbar contains some standard buttons (Home, Open and New) and some buttons that are specific to the gJax development environment (Enable, Disable debugging). You will want to replace this toolbar with one that contains buttons and controls that are relevant to your own application.

- First take a copy of the file 'toolbar.xml' from the gjax directory and place it in your application directory.
- Next customise it suit the need of your application.
- To override the default toolbar set the following global node in your application namespace:

```
>s ^%vcvc("toolbar")=" ../toolbar.xml"
```

Note that the ../ is required because gjax runs in a sub-directory of your application directory and so the ../ is needed to refer to gjax's parent directory.

When you override this setting you will need to restart the workspace for the change to take effect.

Context Menus

All right-click context menus used by your application need to be pre-loaded at the time the workspace for your application is started. The menus that get loaded are controlled by a launcher menu containing a list of menus to be loaded. To define which menus should get loaded for your application:

- Create a menu called xxx.mnuPopup
- Add to this menu all of the right-click context menus that you will need in your application
- Override the default popup menu 'mnuPopup' with your own 'xxx.mnuPopup' menu.

When you override the right-click context menu you will need to restart the workspace for the change to take effect.

Folder Panel and Message Panel

The initial state of the left hand folder panel and the message panel at the bottom of the workspace can be set to open or closed. However, once a user has logged in and manually adjusted the state of these folders then those changes will persist from one session to the next.

TODO: Need instructions on how to do this.

JavaScript

Custom JavaScript functions can be loaded into the workspace at startup time by adding JavaScript files to the list of files that should be pre-loaded. JavaScript functions may be required to support application functionality that is implemented in the main panel.

Any JavaScript file (.js) listed in the JavaScript master file will be pre-loaded at the time the workspace is started. We recommend that each file contain methods relating to just one class or other logical grouping.

Because gJax runs in the gjax sub-directory of your application directory, JavaScript files need to be added with a relative path prefix of ../ For example:

```
../Customer.js
```

Retrieved from "http://10.101.68.18/intranet/index.php/GJax_Workspace"

- This page was last modified 09:34, 28 Jan 2006.