



WFF Dynamic Menu System User Guide

**Dynamic menu and navigation based on access
control settings.**

Version 1.0



Table of Contents

1. Introduction.....	5
1.1 What is a menu system?.....	5
2. Setting up a dynamic menu and navigation system for an a new application.....	8
2.1 Change from static to dynamic menu manager.....	9
2.2 Starting the RBAC Manager	11
2.3 Accessing the RBAC Manager.....	12
2.4 Add localized text keys for all tasks	13
2.5 Adding navigation buttons for parent tasks.....	17
2.6 Adding menu buttons for the Cookbook menu.....	22
3. Testing the dynamic menu	24
3.1 Check that Menu Manager is not overridden	24
3.2 Starting the Cookbook application.....	25
3.3 Accessing the Cookbook application.....	26

Table of Figures

Figure 1-1: Hierarchical menu system	5
Figure 1-1: Dynamic Menu construction process	7
Figure 2-1: The RBAC Manager Web Application	12
Figure 2-2: Cookbook Task List	14
Figure 2-3: Update Category Add Task	15
Figure 2-4: Updated Category Add Task	16
Figure 2-4: The parent tasks selected	17
Figure 2-6: Maintain Navigation Buttons screen	18
Figure 2-7: Task Choose screen	18
Figure 2-8: Task Choose screen with navigation tasks selected	19
Figure 2-9: Navigation Buttons for the Categories List Form	20
Figure 2-10: Navigation Buttons for the Recipes List Form	21
Figure 2-11: Selecting a menu task for maintaining menu buttons	22
Figure 2-12: Maintain Menu Buttons screen	22
Figure 2-13: Menu Buttons for the Cookbook Menu	23
Figure 3-1: The Cookbook Web Application	26
Figure 3-2: List Categories screen	26
Figure 3-3: List Recipes screen for the CB_VIEWER role	27

Revision History

Date	Version	Description	Author
2006-06-25	1.0	Initial version	Martin Bergljung
2006-07-17	1.1	Version for Beta 3 package	Martin Bergljung
2006-08-28	1.2	Updated server port to 9090	Martin Bergljung

1. Introduction

1.1 What is a menu system?

Within an application that may contain dozens or even hundreds of application functions a method is required whereby the user can quickly activate the function that will allow him or her to perform the desired task. This is accomplished by providing a menu system in the form of a list presented on the computer screen with a mechanism so that simply selecting a menu item would cause the associated application function to be activated.

With large systems it can be cumbersome to have a list presented in a single unit, so it is usually split into smaller units or pages. These pages are arranged in a hierarchical structure, and links on one page can activate other pages at a lower level. This is called a *hierarchical menu system*. See the following picture:

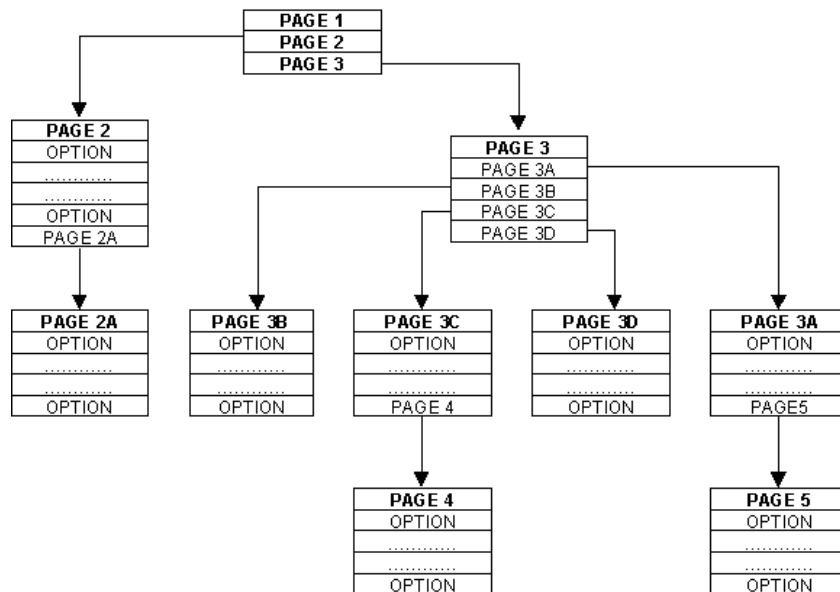


Figure 1-1: Hierarchical menu system

In some systems the menu pages are always hard-coded in a fixed and rigid structure. This method has the following disadvantages:

- The menu pages have to be designed and constructed before development can start.
- The menu pages can not be altered without changing and recompiling a piece of program code.
- Every user always starts at the same menu page after logging on, even if the only options which they are permitted to access existed on a sub-menu.
- Options which are not accessible to a user are still visible on the menu screens.

WFF Dynamic Menu System User Guide

- Context can not be passed from one screen to another. This mean that if the user finds a record on one screen and wanted to go to another screen to see more details the record identity has to be entered manually into the second screen. This is slow and frustrating work, especially if a mistake was made when keying in the record identity.

The menu and navigation system described in this document does not include these disadvantages. It is integrated with the role-based access control security database and provides the following advantages:

- Tables to hold Menu and Navigation data are added to the security database so that menu screens can be constructed dynamically from the contents of the database. As the database contains a list of tasks for security purposes the same data can be used in the definition and construction of menu screens.
- This means that menu pages can be constructed and modified without the need for programmer intervention. There is no need to define the entire menu structure at the start of the project as it can be modified *on the fly*.
- Where a user only has permission to access application functions (tasks) that exist on a sub-menu it can be allowed to appear immediately after the login screen, thus bypassing all unnecessary menu screens.
- As the menu screens are constructed from the database, and as the database also contains the user permissions, it is a simple matter to filter out all those options to which the user does not have permission to access. Instead of having to build a separate set of menu screens for each user a single set can be build which can be shared by all users, and each user will only see those options which he or she are permitted to access.
- A mechanism exists whereby context can be passed from one screen to another, thus removing the need to enter record identities manually.

As can be seen a role-based access control system and a menu system share a lot of common data, so it makes sense to combine them into a single system.

If you have not read the **WFF Role-based Access Control User Guide** now is a good time to do so as this user guide assumes you have setup the access control for the Cookbook application we are going to use for demonstration.

The following picture shows what components are involved in the dynamic menu system:

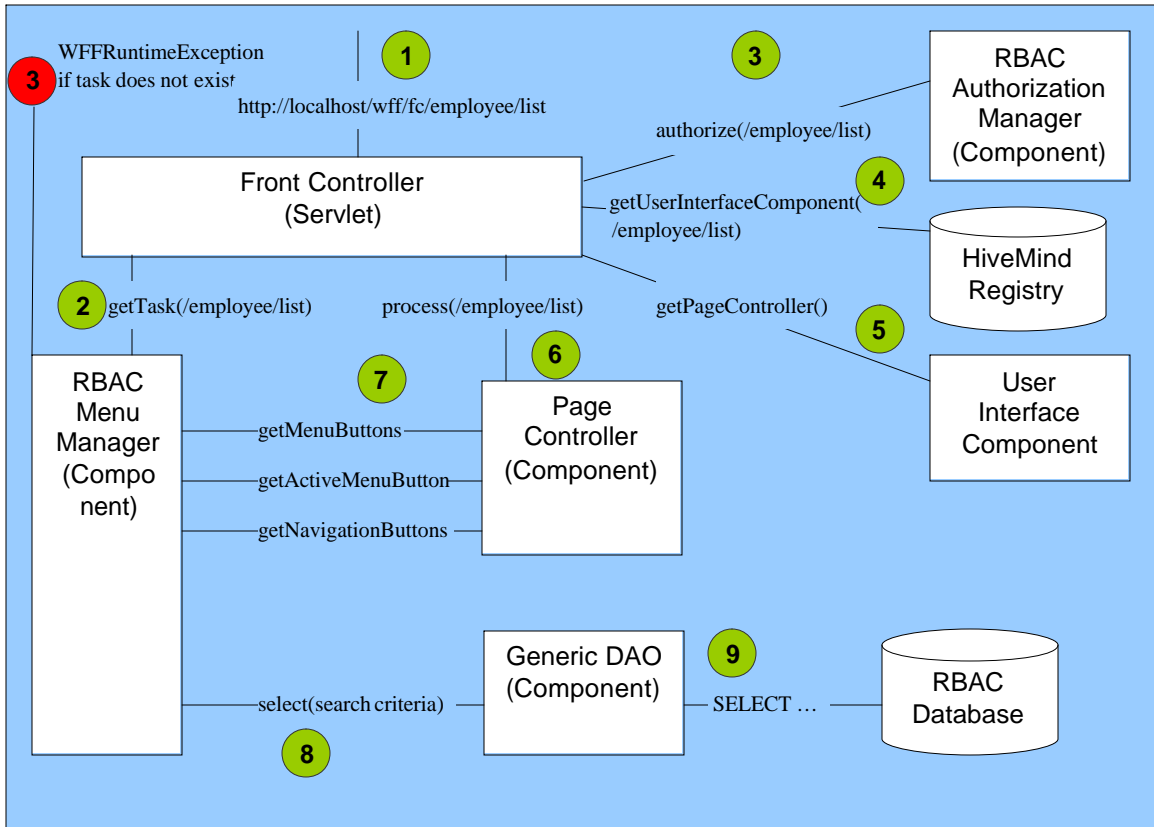


Figure 1-2: Dynamic Menu construction process

The above picture shows that a request comes in via the FrontController Servlet and the RBAC menu manager is called to get the task for the URL path info. If a task is not found in the RBAC database for the URL Path info, the system halts with a WFF Runtime exception as this is considered a programming error that is not supposed to happen under normal operation.

The task is then authorized via the RBAC Authorization Manager (see more about this in the WFF Role-based access control user guide). If authorization is successful the component that matches the URL Path info '/employee/list' is fetched from the HiveMind registry.

The FrontController will then ask the user interface component for the Page Controller to use. The page controller in turn will ask the menu manager, which is installed at the moment, for the menu buttons to display, including which one should be displayed as active, and what Navigation Buttons should be available in the form.

2. Setting up a dynamic menu and navigation system for an a new application

The following chapter goes through how to setup a dynamic menu and navigation system for a new application developed with Web Fast Forms (WFF). We are going to use the Cookbook application that comes with WFF framework when explaining how to setup dynamic menu and navigation. If you have not done so yet walk thru the Cookbook sample user guide and create the application with the WFF Code Generator.

The following steps are the minimum number of steps needed to setup a dynamic menu and navigation system for a new application:

- Setup role-based access control for the new application according to instructions in the **WFF Role-based Access Control User Guide**.
- Change from static menu manager to a dynamic menu manager.
- Add localized text keys for all tasks.
- Add navigation buttons for parent tasks.
- Add menu buttons for menu tasks.

The dynamic menu system is closely related to the role-based access control system, as mentioned earlier, so the RBAC Manager is used to administer the dynamic menu settings as well as the role-based access control settings.

2.1 Change from static to dynamic menu manager

When you design and generate code for an application with Web Fast Forms code generator it creates a so called Form Family file for a group of components. A standard form family consists of a parent form (usually a LIST form) and one or more child forms such as DELETE, UPDATE, and SEARCH etc.

The form family files contain all information about the application tasks such as localized text keys (these keys points to localized texts for the form title text, breadcrumb text, navigation button text, menu button text etc), navigation buttons for a parent task, menu button for a parent task and so on. When using a dynamic menu and navigation system you need to input this information into the RBAC database instead of having it statically hard coded in the Form Family files.

The form Family Files of an application are instantiated in a *Static Menu Manager*. The static menu manager also creates the menu hierarchy statically and keeps a Map of all available tasks in the application.

When the Form Family file information is in the RBAC database a *Dynamic Menu Manager* can search in it and extract just the right information (i.e. what menu buttons to display, what navigation buttons to display etc) depending on the user's permissions.

So we need to install a dynamic menu manager that operates against the RBAC database. The WFF framework comes with such a menu manager called the RBAC Menu Manager. To use it instead of the static menu manager you need to open the HiveMind descriptor file for the application and comment out the static menu manager definition so the system uses the default menu manager, which is in fact the RBAC Menu Manager.

For the Cookbook application the HiveMind descriptor file is called `cookbook-webapp-managers-submodule.xml` and is located in the `cookbook-webapp/src/main/resources/META-INF` directory. Open this file and you should see something like the following:

```
<?xml version="1.0"?>
<module id="cookbook.webapp.managers" version="1.0.0">
  <!--
  *****
  Contributions
  *****
  -->
  <contribution configuration-id="wff.managers.PropertyFileConfig">
    <propertyfile-config resource="CookbookUserInterfaceTexts"
      type="text" path="/WEB-INF/resources/i18n/" />
  </contribution>
```

WFF Dynamic Menu System User Guide

```
<!--
*****
Services that can be overridden (i.e. the implementation is defined inside
the service-point and is considered the default implementation)
*****
-->

<implementation service-id="wff.managers.MenuManager">
  <invoke-factory>
    <construct class="com.marversolutions.wff.samples.cookbook.menu.manager.impl.
                                          CookbookStaticMenuManagerImpl"/>
  </invoke-factory>
</implementation>

<!--
*****
Service Points
*****
-->

<!--
*****
Implementations (cannot be overridden)
*****
-->

</module>
```

As you can see the default Menu Manager for the framework has been overridden with the Cookbook static menu manager. The default menu manager is the RBAC Menu Manager and the one we want so just comment out the menu manager definition section like this:

```
...
<!--
*****
Services that can be overridden (i.e. the implementation is defined inside
the service-point and is considered the default implementation)
*****
<implementation service-id="wff.managers.MenuManager">
  <invoke-factory>
    <construct class="com.marversolutions.wff.samples.cookbook.menu.manager.impl.
                                          CookbookStaticMenuManagerImpl"/>
  </invoke-factory>
</implementation>
-->
...
```

Save the file. Now the system will use the RBAC Menu Manager and expect all task information to be available in the RBAC database.

2.2 Starting the RBAC Manager

Before we can continue you need to start the RBAC Manager application which is used for administering the dynamic menu settings for an application. Issue the following command from the directory as shown below:

```
c:\wff-1.0-beta3\rbac-webapp>mvn jetty6:run-war
[INFO] Scanning for projects...
[INFO] Searching repository for plugin with prefix: 'jetty6'.
[INFO] -----
[INFO] Building WFF Role-based Access Control (RBAC) Web Application
[INFO]   task-segment: [jetty6:run-war]
[INFO] -----
[INFO] Preparing jetty6:run-war
[INFO] [resources:resources]
[INFO] Using default encoding to copy filtered resources.
[INFO] [compiler:compile]
[INFO] Nothing to compile - all classes are up to date
[INFO] [resources:testResources]
[INFO] Using default encoding to copy filtered resources.
[INFO] [antrun:run {execution: prepare-db}]
[INFO] Executing tasks

drop.sql.objects:
  [sql] Executing file: D:\work-wff\trunk\rbac-webapp\src\main\sql\mysql\drop-objects.sql
  [sql] 13 of 13 SQL statements executed successfully
  [sql] Executing file: D:\work-wff\trunk\core-webapp\src\main\sql\mysql\drop-objects.sql
  [sql] 5 of 5 SQL statements executed successfully

create.sql.objects:
  [sql] Executing file: D:\work-wff\trunk\core-webapp\src\main\sql\mysql\audit.sql
  [sql] 5 of 5 SQL statements executed successfully
  [sql] Executing file: D:\work-wff\trunk\rbac-webapp\src\main\sql\mysql\rbac-schema.sql
  [sql] 13 of 13 SQL statements executed successfully

create.sql.data:
  [sql] Executing file: D:\work-wff\trunk\rbac-webapp\src\main\sql\mysql\rbac-data.sql
  [sql] 377 of 377 SQL statements executed successfully

prepare.db:
[INFO] Executed tasks
[INFO] [compiler:testCompile]
[INFO] No sources to compile
[INFO] [surefire:test]
[INFO] No tests to run.
[INFO] [war:war]
[INFO] Exploding webapp...
[INFO] Copy webapp resources to D:\work-wff\trunk\rbac-webapp\target\rbac-webapp
[INFO] Assembling webapp rbac-webapp in D:\work-wff\trunk\rbac-webapp\target\rbac-webapp
[INFO] Overlaying 1 war(s).
[INFO] Generating war D:\work-wff\trunk\rbac-webapp\target\rbac-webapp.war
[INFO] Building war: D:\work-wff\trunk\rbac-webapp\target\rbac-webapp.war
[INFO] [jetty6:run-war]
```

```
[INFO] Configuring Jetty for project: WFF Role-based Access Control (RBAC) Web Application
[INFO] Using JSP2.1 for jdk1.5 runtime
[INFO] Property org.apache.commons.logging.simplelog.showShortLogname=false was set
:INFO: Logging to STDERR via org.mortbay.log.StdErrLog
[INFO] Context path = /wff
[INFO] Tmp directory = D:\work-wff\trunk\rbac-webapp\target\work
[INFO] Web defaults = jetty default
[INFO] Starting jetty 6.0.0beta15 ...
:INFO: jetty 6.0.0beta15
:INFO: Extract jar:file:/D:/work-wff/trunk/rbac-webapp/target/rbac-webapp.war!/ to D:\work-
wff\trunk\rbac-webapp\target\work\webapp
:INFO: Started SelectChannelConnector @ 0.0.0.0:8080
[INFO] Started Jetty Server
```

This will build the RBAC Manager application and start it in the Jetty6 Web Server.

The RBAC Manager is build with the WFF development framework so you can also look at the code for it to figure out how different screens are implemented etc.

2.3 Accessing the RBAC Manager

To access the RBAC Manager from a web browser type in the following URL:

<http://localhost:9090/wff/fc/rbac/menu>

Then login with user ID = RBACADMIN and password = test. This user has access to all the tasks/screens in the RBAC Manager application. You should now see the following menu:



Figure 2-1: The RBAC Manager Web Application

2.4 Add localized text keys for all tasks

The form title text and the breadcrumb text for each task are fetched from property files. These files are generated by the code generator automatically. When using a static menu manger the definition of the keys for fetching these texts are done in so called Form Family files which are instantiated by the static menu manager.

When using the dynamic menu manager it reads all task information from the RBAC database and creates task objects based on it for each incoming application request. The dynamic menu manager that comes with WFF does not cache any data. So the data for each task in the RBAC database need to be appended with the text keys for title and breadcrumb texts.

The following tasks were added as part of setting up role-based access control for the Cookbook application. The text keys have been specified here for each task:

TASK ID	Form Tile Text Key	Breadcrumb Text Key
cb_cookbook_menu	cookbook.menu.form.title	cookbook.menu.breadcrumb
cb_categories_add1	categories.add1.component.form.title	categories.add1.component.breadcrumb
cb_categories_delete1	categories.delete1.component.form.title	categories.delete1.component.breadcrumb
cb_categories_enquire1	categories.enquire1.component.form.title	categories.enquire1.component.breadcrumb
cb_categories_update1	categories.update1.component.form.title	categories.update1.component.breadcrumb
cb_categories_list1	categories.list1.component.form.title	categories.list1.component.breadcrumb
cb_recipes_add1	recipes.add1.component.form.title	recipes.add1.component.breadcrumb
cb_recipes_delete1	recipes.delete1.component.form.title	recipes.delete1.component.breadcrumb
cb_recipes_enquire1	recipes.enquire1.component.form.title	recipes.enquire1.component.breadcrumb
cb_recipes_update1	recipes.update1.component.form.title	recipes.update1.component.breadcrumb
cb_recipes_search1	recipes.search1.component.form.title	recipes.search1.component.breadcrumb
cb_recipes_list1	recipes.list1.component.form.title	recipes.list1.component.breadcrumb

If you want to update yourself with info about how these text keys where specified for the static menu management take a look in the `RecipesFormFamilyConfigImpl.java`, `CategoriesFormFamilyConfigImpl.java`, and `CookbookStaticMenuManagerImpl.java` files.

To add these text keys for the Cookbook application tasks press the ‘Tasks’ menu button. This will display a list of current tasks in the system. Click the ‘Search’ button and select the Cookbook subsystem from the dropdown. Then click ‘Search’ which should display the following list of tasks for the Cookbook application. Also switch to 25 rows per page:

You are logged in as: RBAC Administrator [logout](#) | [logout all](#) | [new session](#) | [print view](#) | [help](#)

Users Roles **Tasks** Help Topics Subsystems Web Page Templates

RBAC»Tasks

List Tasks

New Search

Selections: [select all](#) | [unselect all](#) [show 10](#) | [show 25](#) | show 50 | show 100

View Update Delete Maintain Navigation Buttons Maintain Menu Buttons
Set Task Access Rights Maintain Task Fields Set Field Access Rights Audit Logs

Select	Task ID [▲]	Description	Type
<input type="checkbox"/>	cb_categories_add1	Add a cookbook category	Child Form
<input type="checkbox"/>	cb_categories_delete1	Delete a cookbook category	Child Form
<input type="checkbox"/>	cb_categories_enquire1	View a cookbook category	Child Form
<input type="checkbox"/>	cb_categories_list1	List cookbook categories	Parent Form
<input type="checkbox"/>	cb_categories_update1	Update a cookbook category	Child Form
<input type="checkbox"/>	cb_cookbook_menu	Cookbook Application Menu	Menu
<input type="checkbox"/>	cb_recipes_add1	Add a cookbook recipe	Child Form
<input type="checkbox"/>	cb_recipes_delete1	Delete a cookbook recipe	Child Form
<input type="checkbox"/>	cb_recipes_enquire1	View a cookbook recipe	Child Form
<input type="checkbox"/>	cb_recipes_list1	List cookbook recipes	Parent Form
<input type="checkbox"/>	cb_recipes_search1	Search for cookbook recipes	Child Form
<input type="checkbox"/>	cb_recipes_update1	Update a cookbook recipe	Child Form

«FIRST <PREV (Page 1 of 1) NEXT> LAST»

Reset

page created 0.062 seconds (XSLT= 0.609 seconds)

Figure 2-2: Cookbook Task List

To access the update screen for these tasks click on the ‘select_all’ link and then on the ‘Update’ navigation button.

This brings up the first Cookbook task in edit mode:

You are logged in as: RBAC Administrator [logout](#) | [logout_all](#) | [new_session](#) | [print_view](#) | [help](#)

Users Roles Tasks Help Topics Subsystems Web Page Templates

RBAC»Tasks»Update

Update Task	
Task ID	cb_categories_add1
Description	Add a cookbook category
Type	Child Form
Action (URL path info)	/categories/add1
Web Page Template	ADD1 - Insert a single record
Subsystem	Cookbook Sample Application
Form Title Text Key	
Breadcrumb Text Key	
Help Topic	>>
Initial Passthru Task Action	>>
WHERE SQL clause (fixed)	
WHERE SQL clause (temporary)	
Extra parameters (XSL/App)	
ORDER BY SQL clause	
Task Disabled?	<input type="radio"/> Yes <input checked="" type="radio"/> No
Created Date	7/17/06 10:54:00 AM
Created By	RBACADMIN
Revised Date	
Revised By	
«FIRST <PREV (Record 1 of 12) NEXT> LAST»	
Save	Save and Go to Next
Copy Close	
page created 0.079 seconds (XSLT= 0.922 seconds)	

Figure 2-3: Update Category Add Task

Here you may notice a few extra fields at the bottom of the screen that holds information about who created the data row and who was last to change it.

For detail information about what field changed when and how you would go back to the task list and select all data rows but press the ‘Audit Logs’ navigation button instead.

What we want to do now is fill in the Form Title Text Key field and the Breadcrumb Text Key field like follows and then press ‘Save’:

You are logged in as: RBAC Administrator [logout](#) | [logout_all](#) | [new_session](#) | [print_view](#) | [help](#)

Users Roles Tasks Help Topics Subsystems Web Page Templates

RBAC»Tasks»Update

Update Task	
Task ID	cb_categories_add1
Description	Add a cookbook category
Type	Child Form
Action (URL path info)	/categories/add1
Web Page Template	ADD1 - Insert a single record
Subsystem	Cookbook Sample Application
Form Title Text Key	categories.add1.component.form.title
Breadcrumb Text Key	categories.add1.component.breadcrumb
Help Topic	>>
Initial Passthru Task Action	>>
WHERE SQL clause (fixed)	
WHERE SQL clause (temporary)	
Extra parameters (XSL/App)	
ORDER BY SQL clause	
Task Disabled?	<input type="radio"/> Yes <input checked="" type="radio"/> No
Created Date	7/17/06 10:54:00 AM
Created By	RBACADMIN
Revised Date	7/18/06 5:11:11 PM
Revised By	RBACADMIN

Record updated OK

«FIRST <PREV (Record 1 of 12) **NEXT>** LAST>

Save Save and Go to Next Copy Close

page created 0.687 seconds (XSLT= 0.593 seconds)

Figure 2-4: Updated Category Add Task

Alright, ready to move on to next Cookbook task so click the ‘NEXT>’ link. In that screen update the text in the same way as we did for the Categories ADD1 task. However, after typing in the text keys click on the ‘Save and Go to Next’ button to save and move directly to next data row. Repeat this for the rest of the tasks.

2.5 Adding navigation buttons for parent tasks

Each parent task, such as a form based on one of the LIST or MULTI web page templates, can have one or more navigation buttons that take the user to a child form where data can be for example updated or deleted.

In the Cookbook application we have two parent tasks with the following navigation buttons:

- CB_CATEGORIES_LIST – with the following navigation buttons:
 - Add
 - View
 - Update
 - Delete
- CB_RECIPES_LIST – with the following navigation buttons:
 - Add
 - Search
 - View
 - Update
 - Delete

To add these navigation buttons mark first the two parent tasks like this:

View		Update		Delete		Maintain Navigation Buttons		Maintain Menu Buttons	
Set Task Access Rights			Maintain Task Fields			Set Field Access Rights			Audit Logs
Select	Task ID ^	Description	Type						
<input type="checkbox"/>	cb_categories_add1	Add a cookbook category	Child Form						
<input type="checkbox"/>	cb_categories_delete1	Delete a cookbook category	Child Form						
<input type="checkbox"/>	cb_categories_enquire1	View a cookbook category	Child Form						
<input checked="" type="checkbox"/>	cb_categories_list1	List cookbook categories	Parent Form						
<input type="checkbox"/>	cb_categories_update1	Update a cookbook category	Child Form						
<input type="checkbox"/>	cb_cookbook_menu	Cookbook Application Menu	Menu						
<input type="checkbox"/>	cb_recipes_add1	Add a cookbook recipe	Child Form						
<input type="checkbox"/>	cb_recipes_delete1	Delete a cookbook recipe	Child Form						
<input type="checkbox"/>	cb_recipes_enquire1	View a cookbook recipe	Child Form						
<input checked="" type="checkbox"/>	cb_recipes_list1	List cookbook recipes	Parent Form						
<input type="checkbox"/>	cb_recipes_search1	Search for cookbook recipes	Child Form						
<input type="checkbox"/>	cb_recipes_update1	Update a cookbook recipe	Child Form						

Figure 2-5: The parent tasks selected

Then click on the ‘Maintain Navigation Buttons’ button.

This displays the following form:

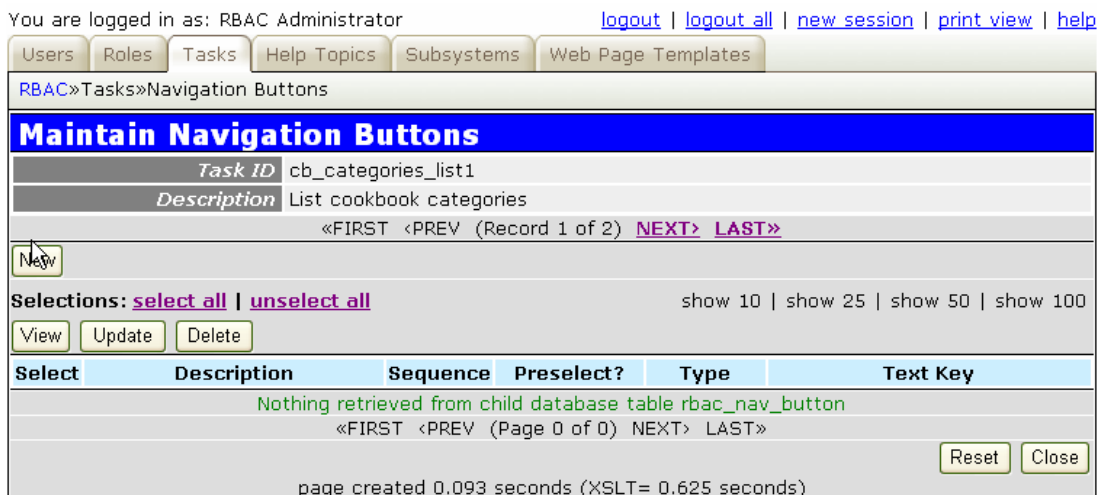


Figure 2-6: Maintain Navigation Buttons screen

In this form click the ‘New’ navigation button to add tasks that should act as navigation buttons for the List cookbook categories parent form. A task picker screen is then displayed as follows:

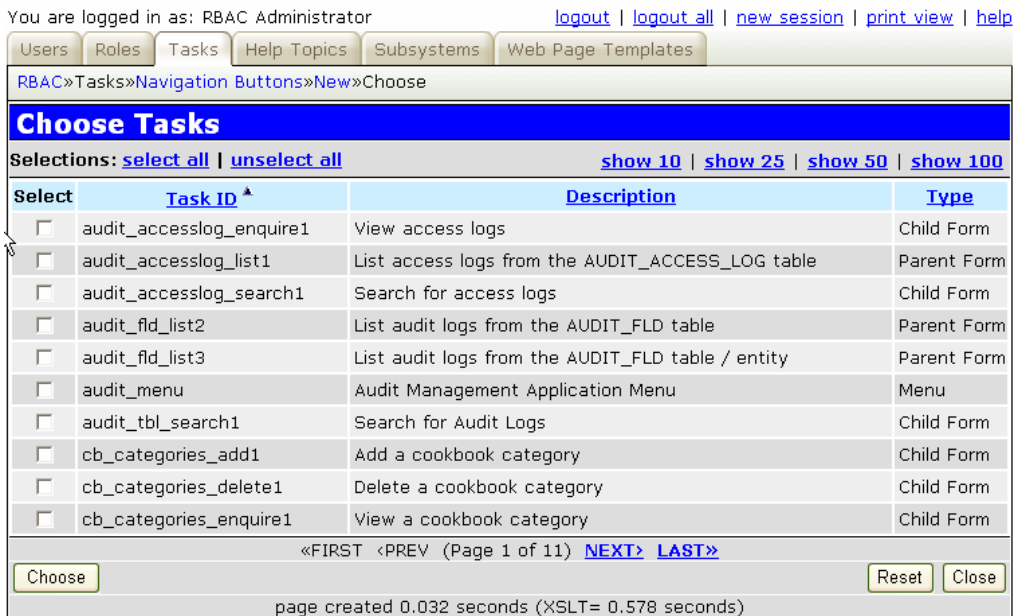


Figure 2-7: Task Choose screen

Here you will see all tasks as most tasks can be selected as navigation buttons for a parent task. This is where you would start to realize that this system provides for quite a lot of reuse if used correctly. For example, in the RBAC Manager itself we use the ‘List Audit logs for an Entity’ component in most forms. The only thing you need to do to use this component is to add a navigation button for it and the page context is transferred between linked pages automatically.

Same goes for us with the Cookbook application pages, just link them together with navigation buttons and the page context is handled automatically. First select the 25 rows per page so we can see all the Cookbook tasks as follows. And mark also the once that should be navigation buttons for the Categories list form:

You are logged in as: RBAC Administrator [logout](#) | [logout all](#) | [new session](#) | [print view](#) | [help](#)

Users Roles Tasks Help Topics Subsystems Web Page Templates

RBAC»Tasks»Navigation Buttons»New»Choose

Choose Tasks

Selections: [select all](#) | [unselect all](#) [show 10](#) | [show 25](#) | [show 50](#) | [show 100](#)

Select	Task ID ^	Description	Type
<input type="checkbox"/>	audit_accesslog_enquire1	View access logs	Child Form
<input type="checkbox"/>	audit_accesslog_list1	List access logs from the AUDIT_ACCESS_LOG table	Parent Form
<input type="checkbox"/>	audit_accesslog_search1	Search for access logs	Child Form
<input type="checkbox"/>	audit_fld_list2	List audit logs from the AUDIT_FLD table	Parent Form
<input type="checkbox"/>	audit_fld_list3	List audit logs from the AUDIT_FLD table / entity	Parent Form
<input type="checkbox"/>	audit_menu	Audit Management Application Menu	Menu
<input type="checkbox"/>	audit_tbl_search1	Search for Audit Logs	Child Form
<input checked="" type="checkbox"/>	cb_categories_add1	Add a cookbook category	Child Form
<input checked="" type="checkbox"/>	cb_categories_delete1	Delete a cookbook category	Child Form
<input checked="" type="checkbox"/>	cb_categories_enquire1	View a cookbook category	Child Form
<input type="checkbox"/>	cb_categories_list1	List cookbook categories	Parent Form
<input checked="" type="checkbox"/>	cb_categories_update1	Update a cookbook category	Child Form
<input type="checkbox"/>	cb_cookbook_menu	Cookbook Application Menu	Menu
<input type="checkbox"/>	cb_recipes_add1	Add a cookbook recipe	Child Form
<input type="checkbox"/>	cb_recipes_delete1	Delete a cookbook recipe	Child Form
<input type="checkbox"/>	cb_recipes_enquire1	View a cookbook recipe	Child Form
<input type="checkbox"/>	cb_recipes_list1	List cookbook recipes	Parent Form
<input type="checkbox"/>	cb_recipes_search1	Search for cookbook recipes	Child Form
<input type="checkbox"/>	cb_recipes_update1	Update a cookbook recipe	Child Form
<input type="checkbox"/>	emp_department_add1	Add a department	Child Form
<input type="checkbox"/>	emp_department_delete1	Delete a department	Child Form
<input type="checkbox"/>	emp_department_enquire1	View a department	Child Form
<input type="checkbox"/>	emp_department_list1	List departments	Parent Form
<input type="checkbox"/>	emp_department_search1	Search for departments	Child Form
<input type="checkbox"/>	emp_department_update1	Update a department	Child Form

«FIRST <PREV (Page 1 of 5) [NEXT](#)> LAST»

Choose Reset Close

page created 0.063 seconds (XSLT= 1.171 seconds)

Figure 2-8: Task Choose screen with navigation tasks selected

Now click the ‘Choose’ button to select these Cookbook tasks as navigation buttons for the Categories list form.

The following screen is displayed:

You are logged in as: RBAC Administrator [logout](#) | [logout_all](#) | [new_session](#) | [print_view](#) | [help](#)

Users Roles Tasks Help Topics Subsystems Web Page Templates

RBAC»Tasks»Navigation Buttons

Maintain Navigation Buttons

Task ID cb_categories_list1
Description List cookbook categories
 «FIRST <PREV (Record 1 of 2) **NEXT**> LAST»

Selections: [select all](#) | [unselect all](#) show 10 | show 25 | show 50 | show 100

Select	Description	Sequence	Preselect?	Type	Text Key
<input type="checkbox"/>	View a cookbook category	001	<input checked="" type="radio"/> Yes <input type="radio"/> No	View	standard.navigation.button.read
<input type="checkbox"/>	Add a cookbook category	002	<input type="radio"/> Yes <input checked="" type="radio"/> No	New	standard.navigation.button.new
<input type="checkbox"/>	Update a cookbook category	003	<input checked="" type="radio"/> Yes <input type="radio"/> No	Update	standard.navigation.button.update
<input type="checkbox"/>	Delete a cookbook category	004	<input checked="" type="radio"/> Yes <input type="radio"/> No	Delete	standard.navigation.button.delete

4 record(s) were inserted into rbac_nav_button
 «FIRST <PREV (Page 1 of 1) NEXT> LAST»

page created 0.062 seconds (XSLT= 0.547 seconds)

Figure 2-9: Navigation Buttons for the Categories List Form

As you can see all the data for the navigation buttons are filled in for us as these buttons are standard navigation buttons. The localized text for these standard buttons comes in system property files with text keys as indicated above. The ‘Preselect?’ parameter indicates if the user needs to select one or more data rows before the button can be pressed. It also decides where in the Navigation Bar the button will be displayed. In the top area for buttons that does not require pre-selection and in the bottom area for buttons that does require selection of data rows.

Now click on the ‘NEXT>’ link at the top to scroll to the next parent task.

Add navigation buttons as follows:

You are logged in as: RBAC Administrator [logout](#) | [logout all](#) | [new session](#) | [print view](#) | [help](#)

Users Roles Tasks Help Topics Subsystems Web Page Templates

RBAC»Tasks»Navigation Buttons

Maintain Navigation Buttons

Task ID cb_recipes_list1
Description List cookbook recipes
 «FIRST <PREV (Record 2 of 2) NEXT> LAST»

Selections: [select all](#) | [unselect all](#) show 10 | show 25 | show 50 | show 100

Select	Description	Sequence	Preselect?	Type	Text Key
<input type="checkbox"/>	Delete a cookbook recipe	001	<input checked="" type="radio"/> Yes <input type="radio"/> No	Delete	standard.navigation.button.delete
<input type="checkbox"/>	Search for cookbook recipes	002	<input type="radio"/> Yes <input checked="" type="radio"/> No	Search	standard.navigation.button.search
<input type="checkbox"/>	Add a cookbook recipe	003	<input type="radio"/> Yes <input checked="" type="radio"/> No	New	standard.navigation.button.new
<input type="checkbox"/>	View a cookbook recipe	004	<input checked="" type="radio"/> Yes <input type="radio"/> No	View	standard.navigation.button.read
<input type="checkbox"/>	Update a cookbook recipe	005	<input checked="" type="radio"/> Yes <input type="radio"/> No	Update	standard.navigation.button.update

5 record(s) were inserted into rbac_nav_button
 «FIRST <PREV (Page 1 of 1) NEXT> LAST»

page created 0.062 seconds (XSLT= 0.547 seconds)

Figure 2-10: Navigation Buttons for the Recipes List Form

2.6 Adding menu buttons for the Cookbook menu

The last thing needed for the Cookbook application to work with the RBAC Menu Manager is to feed the RBAC database with the menu hierarchy. When using a static menu manager the menu is statically defined and cannot be dynamically changed depending on user, role, and permissions.

In the case of the Cookbook menu it is extremely simple with just one level and two menu buttons. So let's select the menu task from the task list and then click on the 'Maintain Menu Buttons' navigation button.

Select	Task ID [▲]	Description	Type
<input type="checkbox"/>	cb_categories_list1	List cookbook categories	Parent Form
<input type="checkbox"/>	cb_categories_update1	Update a cookbook category	Child Form
<input checked="" type="checkbox"/>	cb_cookbook_menu	Cookbook Application Menu	Menu
<input type="checkbox"/>	cb_recipes_add1	Add a cookbook recipe	Child Form

Figure 2-11: Selecting a menu task for maintaining menu buttons

That would bring up the following form:

You are logged in as: RBAC Administrator [logout](#) | [logout all](#) | [new session](#) | [print view](#) | [help](#)

[Users](#) [Roles](#) [Tasks](#) [Help Topics](#) [Subsystems](#) [Web Page Templates](#)

RBAC»Tasks»Menu Buttons

Maintain Menu Buttons

Menu Id cb_cookbook_menu
Description Cookbook Application Menu

[New](#)

Selections: [select all](#) | [unselect all](#) show 10 | show 25 | show 50 | show 100

[View](#) [Update](#) [Delete](#)

Select	Description	Sequence	Text Key
Nothing retrieved from child database table rbac_menu_button			
«FIRST <PREV (Page 0 of 0) NEXT> LAST»			

[Reset](#) [Close](#)

page created 0.062 seconds (XSLT= 0.547 seconds)

Figure 2-12: Maintain Menu Buttons screen

To add parent tasks as menu buttons for the Cookbook application menu click the 'New' button. This brings up the task picker screen again in the same way as when we were adding navigation buttons.

Select the categories list form and the recipes list form as menu buttons so you end up with the following screen on the way back from the task picker:

You are logged in as: RBAC Administrator [logout](#) | [logout_all](#) | [new_session](#) | [print_view](#) | [help](#)

Users Roles Tasks Help Topics Subsystems Web Page Templates

RBAC»Tasks»Menu Buttons

Maintain Menu Buttons

Menu Id cb_cookbook_menu
Description Cookbook Application Menu

New

Selections: [select all](#) | [unselect all](#) show 10 | show 25 | show 50 | show 100

View Update Delete

Select	Description	Sequence	Text Key
<input type="checkbox"/>	List cookbook categories	001	categories.list1.component.menubut
<input type="checkbox"/>	List cookbook recipes	002	recipes.list1.component.menubutton

2 record(s) were updated in rbac_menu_button
Update was Successful

«FIRST <PREV (Page 1 of 1) NEXT> LAST»

Save Save and Go to Next Reset Close

page created 0.265 seconds (XSLT= 0.938 seconds)

Figure 2-13: Menu Buttons for the Cookbook Menu

The text keys have been updated to `categories.list1.component.menubutton.name` and `recipes.list1.component.menubutton.name` to reflect the auto generated key names used by the properties files.

3. Testing the dynamic menu

Let's test the dynamic menu and navigation system and see if it displays only those menu options and navigation options that the user has access right to. This assumes of course that you have followed the setup of role-based access control for the Cookbook application described in the WFF Role-based Access Control Guide.

3.1 Check that Menu Manager is not overridden

First, check that the default RBAC Menu Manager is not overridden with a static Menu Manager which would mean that there will be no dynamic menus and the menu would not adjust to role etc.

Open the `cookbook-webapp-managers-submodule.xml` descriptor and make sure there is no active section like the following one:

```
<implementation service-id="wff.managers.MenuManager">
  <invoke-factory>
    <construct class=
"com.marversolutions.wff.samples.cookbook.menu.manager.impl.CookbookStaticMenuManagerImpl
"/>
  </invoke-factory>
</implementation>
```


3.2 Starting the Cookbook application

To start this application issue the following command from the cookbook-webapp directory as shown below:

```
c:\wff-1.0-beta3\cookbook-webapp>mvn jetty6:run-war
[INFO] Scanning for projects...
[INFO] Searching repository for plugin with prefix: 'jetty6'.
[INFO] -----
[INFO] Building WFF Cookbook Sample Web Application
[INFO]   task-segment: [jetty6:run-war]
[INFO] -----
[INFO] Preparing jetty6:run-war
[INFO] [resources:resources]
[INFO] Using default encoding to copy filtered resources.
[INFO] [compiler:compile]
[INFO] Nothing to compile - all classes are up to date
[INFO] [resources:testResources]
[INFO] Using default encoding to copy filtered resources.
[INFO] [compiler:testCompile]
[INFO] No sources to compile
[INFO] [surefire:test]
[INFO] No tests to run.
[INFO] [war:war]
[INFO] Exploding webapp...
[INFO] Copy webapp resources to D:\work-wff\trunk\cookbook-webapp\target\cookbook-webapp
[INFO] Assembling webapp cookbook-webapp in D:\work-wff\trunk\cookbook-
webapp\target\cookbook-webapp
[INFO] Expanding: C:\Documents and
Settings\Martin\.m2\repository\com\marversolutions\wff\core-webapp\1.0-beta3-
SNAPSHOT\core-webapp-1.0-beta3-SNAPSHOT.war into D:\work-wff\trunk\cookbook-
webapp\target\war\work\core-webapp-1.0-beta3-SNAPSHOT
[INFO] Overlaying 1 war(s).
[INFO] Generating war D:\work-wff\trunk\cookbook-webapp\target\cookbook-webapp.war
[INFO] Building war: D:\work-wff\trunk\cookbook-webapp\target\cookbook-webapp.war
[INFO] [jetty6:run-war]
[INFO] Configuring Jetty for project: WFF Cookbook Sample Web Application
[INFO] Using JSP2.1 for jdk1.5 runtime
[INFO] Property org.apache.commons.logging.simplelog.showShortLogname=false was set
:INFO: Logging to STDERR via org.mortbay.log.StdErrLog
[INFO] Context path = /wff
[INFO] Tmp directory = D:\work-wff\trunk\cookbook-webapp\target\work
[INFO] Web defaults = jetty default
[INFO] Starting jetty 6.0.0beta15 ...
:INFO: jetty 6.0.0beta15
:INFO: Extract jar:file:/D:/work-wff/trunk/cookbook-webapp/target/cookbook-webapp.war!/ to
D:\work-wff\trunk\cookbook-webapp\target\work\webapp
:INFO: Started SelectChannelConnector @ 0.0.0.0:8080
[INFO] Started Jetty Server
```

This will build the Cookbook web application and start it in the Jetty6 Web Server.

3.3 Accessing the Cookbook application

To access the Cookbook application from a web browser type in the following URL:

<http://localhost:9090/wff/fc/cookbook/menu>

Then login with one of the users we added such as user ID = CB_ADMIN and password = test. This user has access to all the tasks/screens in the Cookbook application. You should now see the following menu:

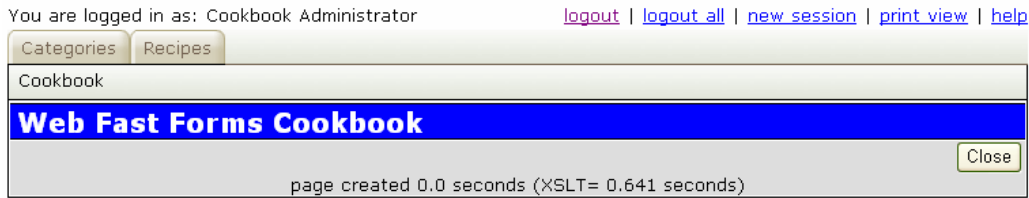


Figure 3-1: The Cookbook Web Application

Click on the ‘Categories’ menu button and you should see a categories list form where you can add, delete, update, and view categories:

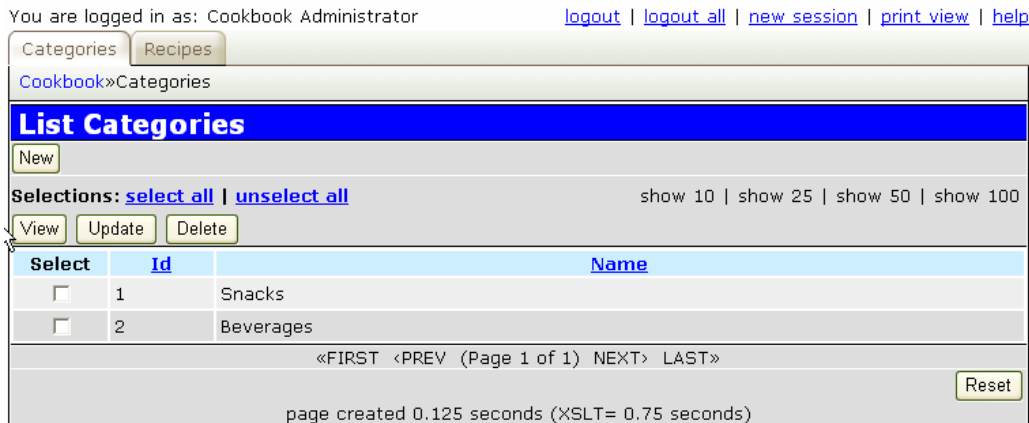


Figure 3-2: List Categories screen

The Cookbook SQL script pre-loads the database with some categories and recipes so you should see the Snacks and Beverages categories as above.

Try and add or update a new category that should work fine as you have full access to all tasks.

Now, logout and then login again with the user ID = CB_VIEWER and password = test. This user has read-only access and should only be able to view categories and recipes. You can also search among recipes. Clicking on the Recipes menu button should display the following form, without navigation buttons that cannot be accessed:

WFF Dynamic Menu System User Guide

You are logged in as: Cookbook Viewer [logout](#) | [logout all](#) | [new session](#) | [print view](#) | [help](#)

Categories Recipes

Cookbook»Recipes

List Recipes

Search

Selections: [select all](#) | [unselect all](#) show 10 | show 25 | show 50 | show 100

View

Select	Recipe	Category	Date
<input type="checkbox"/>	Hot Chips	1	11/11/04
<input type="checkbox"/>	Ice Water	2	11/11/04

«FIRST <PREV (Page 1 of 1) NEXT> LAST»

page created 0.266 seconds (XSLT= 0.718 seconds) [Reset](#)

Figure 3-3: List Recipes screen for the CB_VIEWER role

This is it, the Cookbook application is setup for dynamic menu and navigation operation. The menu and navigation alternatives can now be completely controlled from the RBAC Manager. You can even do changes to the menu and navigation options during runtime as the RBAC Menu Manager does not cache any task data.