



# **QuickClip Portlet, BEA WebLogic Edition Developer Guide**

Version 6.2  
February 2007



## Copyright

Copyright © 1995-2005 BEA Systems, Inc. All Rights Reserved.

## Restricted Rights Legend

This software is protected by copyright, and may be protected by patent laws. No copying or other use of this software is permitted unless you have entered into a license agreement with BEA authorizing such use. This document is protected by copyright and may not be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form, in whole or in part, without prior consent, in writing, from BEA Systems, Inc.

Information in this document is subject to change without notice and does not represent a commitment on the part of BEA Systems. THE DOCUMENTATION IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FURTHER, BEA SYSTEMS DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE DOCUMENT IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE.

## Trademarks and Service Marks

Copyright © 1995-2005 BEA Systems, Inc. All Rights Reserved. BEA, BEA JRockit, BEA WebLogic Portal, BEA WebLogic Server, BEA WebLogic Workshop, Built on BEA, Jolt, JoltBeans, SteelThread, Top End, Tuxedo, and WebLogic are registered trademarks of BEA Systems, Inc. BEA AquaLogic, BEA AquaLogic Data Services Platform, BEA AquaLogic Enterprise Security, BEA AquaLogic Service Bus, BEA AquaLogic Service Registry, BEA Builder, BEA Campaign Manager for WebLogic, BEA eLink, BEA Liquid Data for WebLogic, BEA Manager, BEA MessageQ, BEA WebLogic Commerce Server, BEA WebLogic Communications Platform, BEA WebLogic Enterprise, BEA WebLogic Enterprise Platform, BEA WebLogic Enterprise Security, BEA WebLogic Express, BEA WebLogic Integration, BEA WebLogic Java Adapter for Mainframe, BEA WebLogic JDriver, BEA WebLogic Log Central, BEA WebLogic Network Gatekeeper, BEA WebLogic Personalization Server, BEA WebLogic Personal Messaging API, BEA WebLogic Platform, BEA WebLogic Portlets for Groupware Integration, BEA WebLogic Server Process Edition, BEA WebLogic SIP Server, BEA WebLogic WorkGroup Edition, Dev2Dev, Liquid Computing, and Think Liquid are trademarks of BEA Systems, Inc. BEA Mission Critical Support, BEA Mission Critical Support Continuum, and BEA SOA Self Assessment are service marks of BEA Systems, Inc.

All other names and marks are property of their respective owners.

# Contents

<b>QuickClip Portlet, BEA WebLogic Edition Developer Guide.....</b>	<b>1</b>
<b>Contents.....</b>	<b>iii</b>
<b>Developer Guide .....</b>	<b>1</b>
Introduction .....	1
Who Should Read This Guide .....	1
Requirements .....	1
<b>Getting Started.....</b>	<b>2</b>
Setting up a portal project for the QuickClip Portlet .....	2
Creating a Datasync Project .....	3
Creating a Portal Web Project.....	4
Installing the QuickClip Portlet into a Portal Web Project .....	4
Creating a Portal.....	5
Adding a QuickClip Portlet to Your Portal .....	6
Running the portal .....	6
<b>Product Overview .....</b>	<b>8</b>
Preferences.....	9
Creating a new QuickClip Portlet .....	13
<b>The QuickClip Sample Portal .....</b>	<b>15</b>
Running the Sample Portal .....	15
The Sample Portlets .....	17
<b>Configuration.....</b>	<b>22</b>
Starting a RoboServer.....	22
Changing the RoboServer Port Number .....	22
Changing the Source Folder .....	23
Localization .....	24
<b>Limitations .....</b>	<b>25</b>
QuickClip .....	25
Web Clipping in General .....	26
<b>Index.....</b>	<b>27</b>



# Developer Guide

This guide describes QuickClip Portlet, BEA WebLogic Edition for BEA WebLogic Portal 9.2.

## ***Introduction***

The QuickClip Portlet is a lightweight web clipping portlet for BEA WebLogic Portal created by Kapow Technologies. Web Clipping is the reuse the UI (or part of) of a web based application in another web based application, for example, a portal.

Let us say that you want to surface a standard intranet web application in a portal, but you do not want to re-implement the application as a portlet. You want everything from the application to appear and function as in the application itself. If you use QuickClip Portlet, BEA WebLogic edition, you can surface the web application in a portlet. As you navigate through the application, it remains in the portlet with the same look and feel as the original application. Even if the web application requires authentication, you can log into the portal, and QuickClip Portlet, BEA WebLogic Edition can log you into the application automatically.

## ***Who Should Read This Guide***

You should read this guide, if you are a developer that wants to create QuickClip Portlets for BEA WebLogic Portal in BEA Workshop for WebLogic.

## ***Requirements***

### **Technical Requirements**

The QuickClip Portlet requires WebLogic Workshop 9.2. Besides this all you need to do is to install the BEA WebLogic QuickClip Portlet in your BEA home. Installation is described in the BEA WebLogic QuickClip Portlet Installation Guide. The installation will take care of setting up the necessary run-time environment for the QuickClip Portlet to run. How to add the QuickClip Portlet feature to a given portlet project is described in this guide.

### **Reader Requirements**

As a reader you must be familiar with basic use of BEA WebLogic Workshop. For this we refer you to the on-line documentation of Workshop. Especially you should have a basic knowledge of how to create Java Portlet and portals in Workshop.

## Getting Started

This chapter gives a short tutorial that shows how you create your first QuickClip portlet.

Once you have installed the QuickClip Portlet feature in your BEA WebLogic installation you are ready to create a portal with instances of portlets. But before you get to the point where you can test your first QuickClip Portlet there are some basic steps that you need to go through, most of which is not particularly linked to the QuickClip Portlet. We will briefly describe these steps for the completeness of the example. What we will do is create a new portal application and in this a new portal project. If you already have a portal project that you want to use for trying out the QuickClip Portlet, then you may use this and therefore skip the steps below that create the portal application and the portal project.

### ***Setting up a portal project for the QuickClip Portlet***

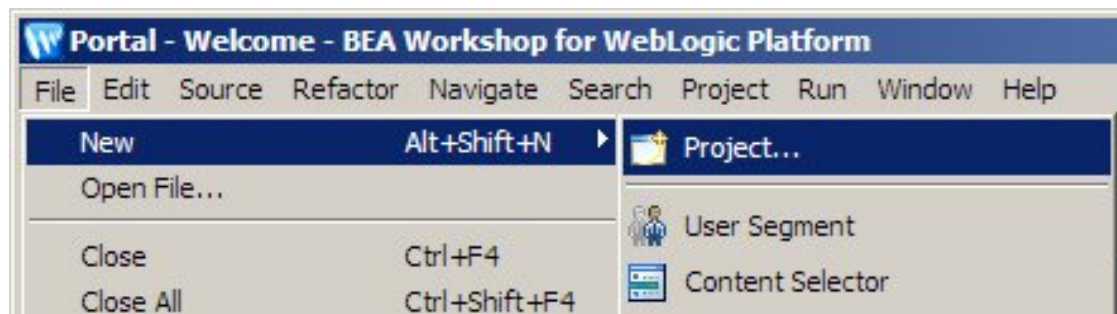
When you installed the BEA WebLogic QuickClip Portlet you also installed the runtime system that is responsible for accessing the remote web site (the original web application that we clip from). The QuickClip Portlet runtime system is just a Kapow RoboServer. In order to make the QuickClip Portlet work you need to start the RoboServer that were installed with the QuickClip Portlet. There are several ways to ensure this and this will be described in detail later in the guide.

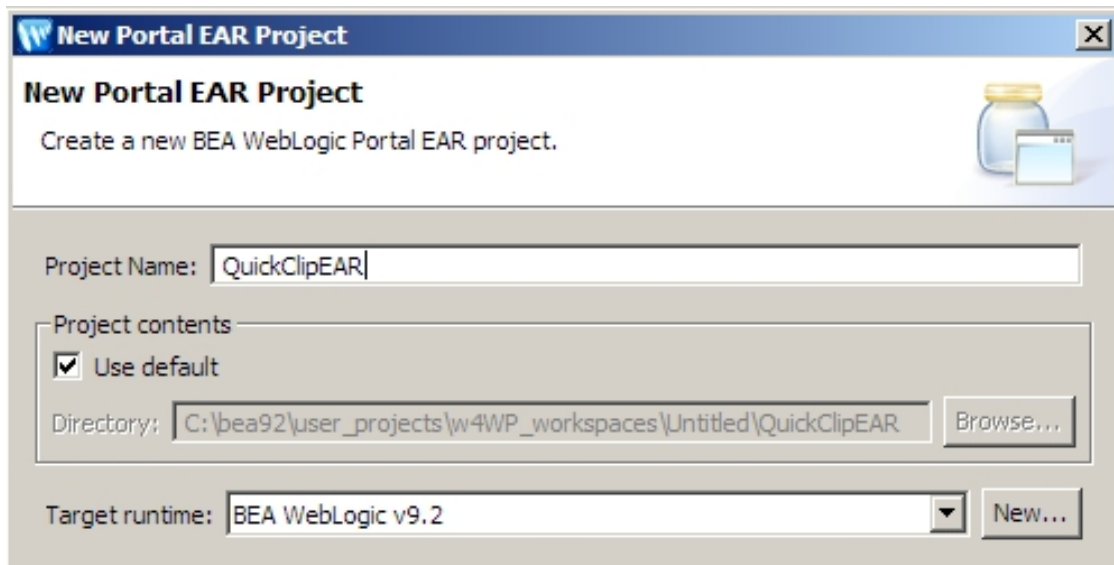
The easiest way to startup the RoboServer is to use the provided shortcuts installed in the Workshop toolbar. Clicking the 'Start RoboServer' icon will open a window running the RoboServer terminal with default settings. The 'Stop RoboServer' icon can be used to shutdown the RoboServer again.

A WebLogic Workshop project is the top level unit that you manipulate in Workshop. You will always have to create a project to which you can add your portlets, portals etc. When building portals one way to deploy them is to embed them in an EAR (Enterprise Archive) project. To create a new EAR project in Workshop you open the File menu and select **New > Project...** In the dialog you should select a Portal EAR Project in the WebLogic Portal folder.

This will start the New Portal Ear Project dialog. Name the project *QuickClipEAR* and click finish to use the default values.

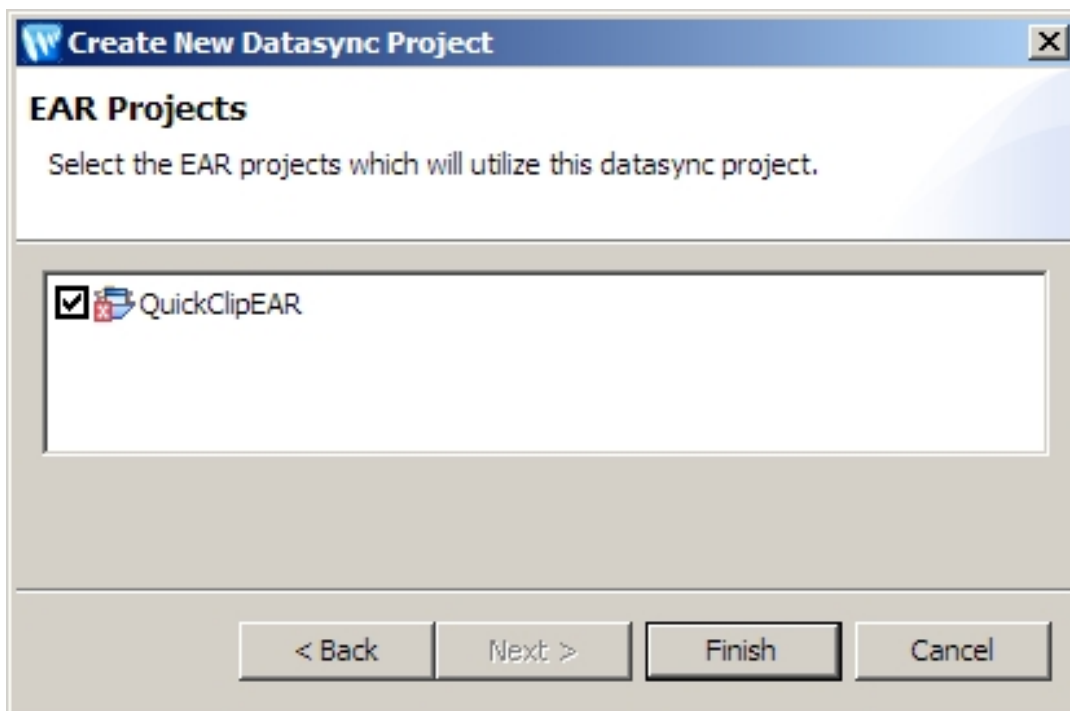
**Figure 1: Selecting a new Project in Workshop**



**Figure 2: Naming a Portal EAR Project**

### ***Creating a Datasync Project***

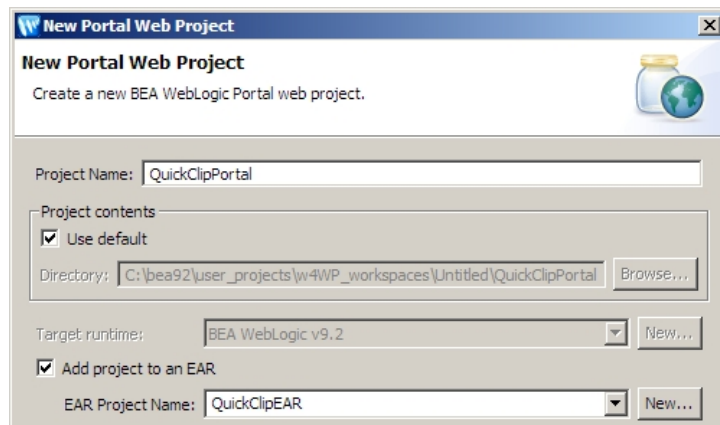
To allow your project to keep track of users, passwords and other data, we will add a Datasync Project to your Portal Ear Project. Open the file menu and select **New > Project...** In the dialog you should select a Datasync Project in the WebLogic Portal folder. Name the new project *QuickClipData* and click next. In the next step you should select QuickClipEAR to allow this project to utilize the datasync project.

**Figure 3: Adding the datasync project to the ear project**

## Creating a Portal Web Project

Now we are ready to add an actual Portal Web Project to our ear. Again we select **New > Project...** in the File menu. In the dialog we select a Portal Web Project under the WebLogic Portal folder. Name the project *QuickClipPortal* and select the Add project to EAR option, and make sure the QuickClipEAR is selected.

**Figure 4: Creating a New Portal Web Project**

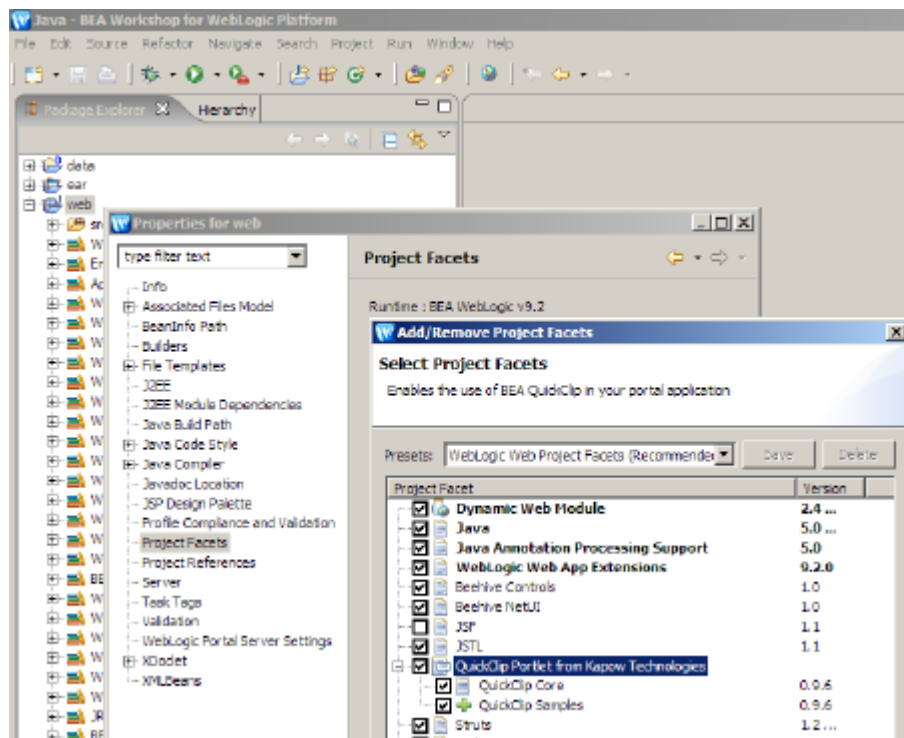


## Installing the QuickClip Portlet into a Portal Web Project

In order to use the QuickClip Portlet in an existing Portal Project you need to add the QuickClip facet to your project. In order to use the QuickClip Portlet in a new project, you should just create a new portal project and then follow the procedure described below.

To install the QuickClip Portlet files in a Portal Project right click on the project in the Package Explorer and select **Properties > Project facets > Add/Remove Project Facets**. On the list of available facets you should select the QuickClip Portlet.

You can choose to install only the QuickClip Core facet, which will install all the necessary files for you to be able to add a QuickClip portlet to your portals in the given project. Or you can choose to also install the QuickClip Sample facet, which will install a sample portal with a number of different QuickClip portlets.

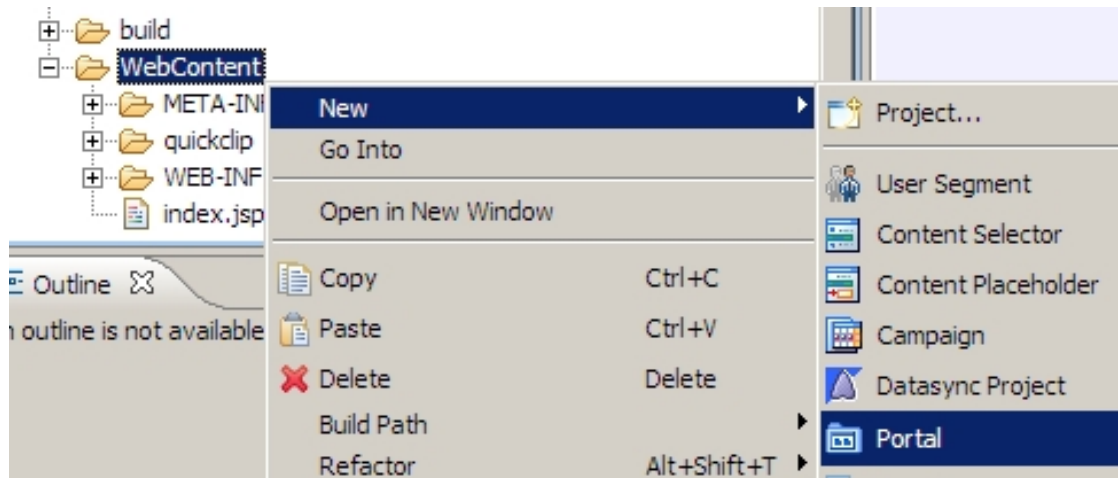
**Figure 5: Adding the QuickClip Portlet facet to a Portal Project**

The installation will create a folder called `quickclip` in the root folder of your portal project. This folder will contain most of the files necessary to run the portlet. But since the QuickClip portlet is a Java Portlet, its implementation also contains some Java code (classes) which will be installed in the `lib` folder of the portals web project, that is in `WEB-INF/lib`.

If you have chosen to install the QuickClip Samples you will also have a folder inside the `quickclip` folder containing a sample portal and some portlets.

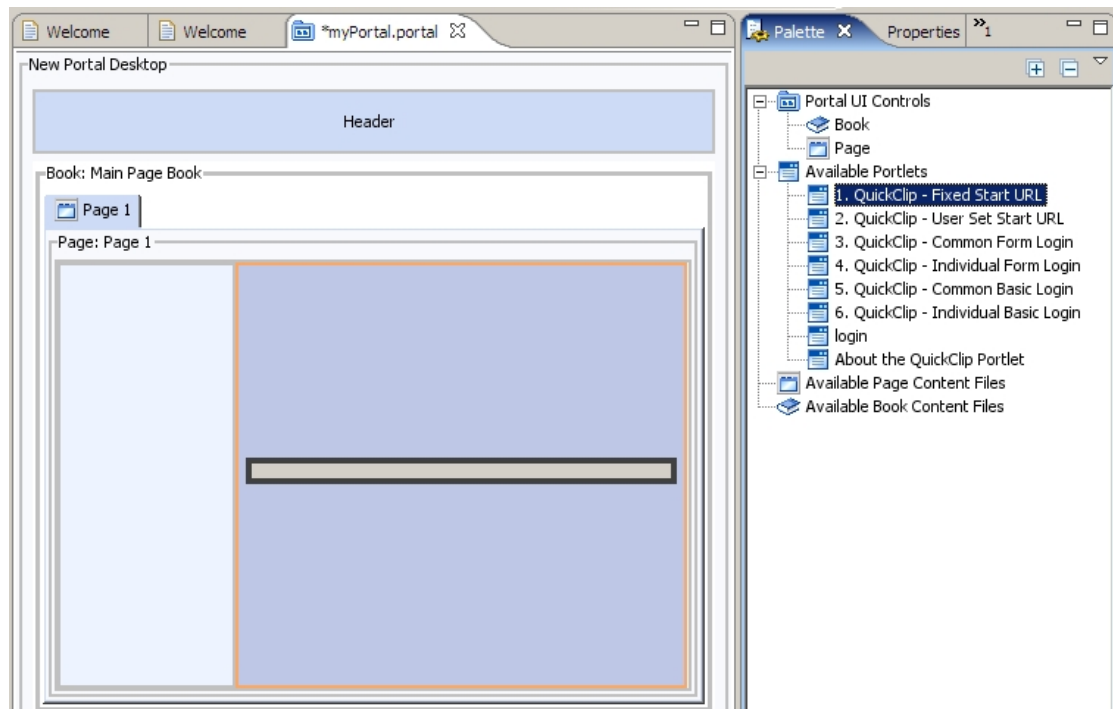
## Creating a Portal

The next step is to create a portal on which to place QuickClip portlets. Expand the QuickClipPortal project and right-click on the WebContent folder in the Package Explorer. Select **New > Portal**, and name your portal *myPortal*. This will create a file called `myPortal.portal` and open a graphical editor in the main window of Workshop.

**Figure 6: Adding a New Portal**

### ***Adding a QuickClip Portlet to Your Portal***

Initially we will just add one of the sample portlets to the portal. Look in the Palette area of Workshop and find the QuickClip portlets listed under Available Portlets. To add a simple portlet, just drag the first example (1. QuickClip – Fixed Start URL) on to your portal in the editor window.

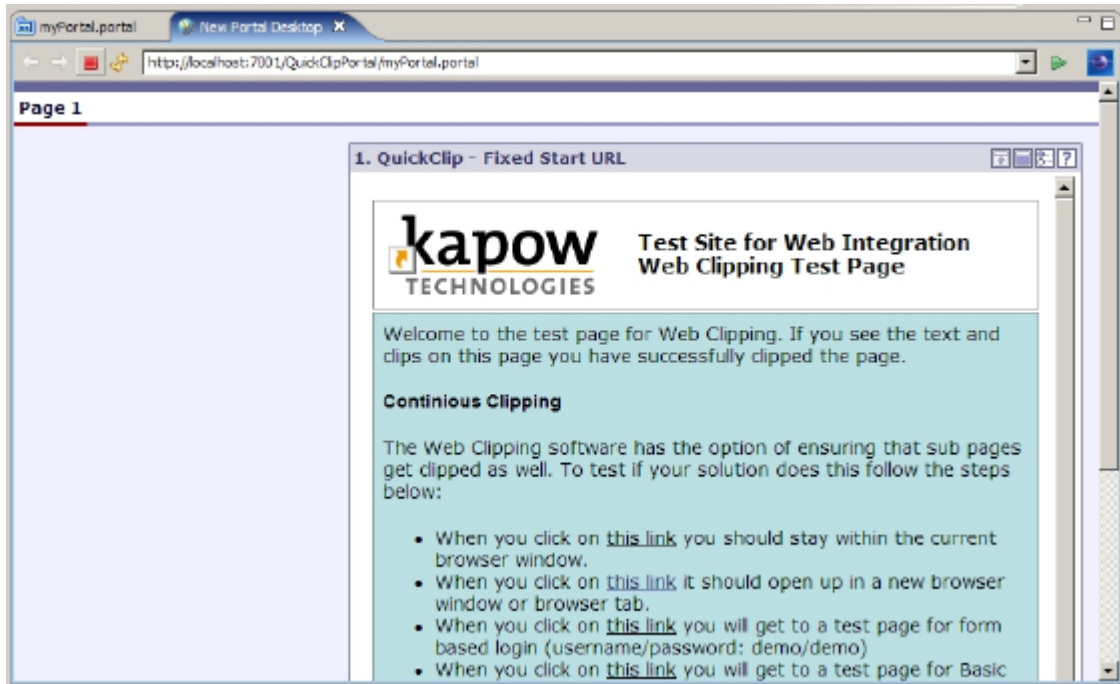
**Figure 7: Adding a QuickClip Portlet to a Portal**

### ***Running the portal***

To run your portal right click on the myPortal.portal file in the Package Explorer. Select **Run as... > Run on Server**. Workshop may ask you to select sources to save before starting. If so,

make sure your portal project is included in the list. Your portal should now show up in a browser window inside Workshop. If you have not yet started RoboServer you will get an error message telling you that the portlet is not available. Just go ahead and start the server from the 'Start RoboServer' icon from the toolbar and reload your portal.

**Figure 8: The QuickClip Portlet**



## Product Overview

In this chapter we will give an overview of the components that the QuickClip Portlet consists of.

The QuickClip Portlet is a Java Portlet which means that it is a portlet based on The Java Portlet Specification (JSR 168). A Java portlet is basically implemented through one or more Java classes, but the implementation may also consist of JSP files for the presentation layer as well as other resources like images. Java Portlets are configured through a file called `portlet.xml` which is placed in the `WEB-INF` folder of the portals web application.

The QuickClip Portlet consists of two large scale components: a client side component and a server side component. The client side component is what is installed into a project when you choose to install the QuickClip Portlet in a project and it consists of the following subcomponents:

- A portlet class: `com.kapowtech.quickclip.QuickClip`. This is a Java Portlet class. The class is located in a Jar-file called `kapow-quickclip.jar` which also contains some additional classes used by the `QuickClip` class. The Jar-file is located in the `lib` folder under `WEB-INF` in the portal projects web application.
- A servlet class called `WebLogic81ClipServlet`. This servlet is responsible for handling the clip request from the portlet (not the request from the edit and help mode).
- Some JSP pages with additional resources, for example, images. These pages implement the UI of the edit and help modes. There are also several pages responsible for handling different types of error cases.
- A clip configuration file called `quickclip.clip`. This file is an XML-file and configures the web clipping functionality, for example, how to contact the server side component.
- Two extra Jar-files: `robosuite-api.jar` and `robosuite-clip.jar`. These Jar-files will be located in the `lib` folder under `WEB-INF` in the portal projects web application.

All these files are copied to their appropriate location when you install the QuickClip into you project. Besides this a few other things happens. Since the portlet depends on a servlet this servlet must be properly registered in the web applications `web.xml` file. This means adding a servlet element and a servlet-mapping element to the `web.xml` file. This will all happen automatically when you install the QuickClip into you project. Besides this the `portlet.xml` file is placed in the `WEB-INF` folder of your web application. This contains declarations of the sample portlets.

The server side component was installed when you installed the QuickClip Portal in you BEA WebLogic installation (See the BEA WebLogic QuickClip Portlet Installation Guide on how to do this) and is the same for all application. The server is a RoboServer from Kapow Technologies. This will run a web clipping robot that is responsible for creating the web clip presented in the portlet. In this guide we will not dive much deeper into the subject of web clipping robots. We will only give a very short description of the QuickClip Portlet cooperate with the RoboServer. When a QuickClip Portlet presents a clip in the portlet (in view mode) the portlet will send a request to the RoboServer to get the web clip. This request is sent over a socket to the RoboServer. This means that the portlet and the server will have to agree on a port number to communicate over. This port number can be set in the `quickclip.clip` file and how to do this will be described in the Configuration Section at the end of the guide

When the QuickClip portlet initially contacts the RoboServer a new session will be established and if login is required the RoboServer will perform a login to the remote site and maintain a session to the remote site. The subsequent requests to the RoboServer from the portlet will not

need to do a login unless the session hold by the server to the remote site has timed out. If this happens then the RoboServer will automatically login again.

## Preferences

All the information that a given QuickClip portlet instance need to pass to the RoboServer such that this may access the remote site and send back a web clip, for example, the start URL of the clip, the credential, etc. is stored in portlet preferences. The initial values for these portlet preferences are specified in portlet elements in the `portlet.xml` file. Figure 9 shows part of such an element. The QuickClip Portlet has so many preference values that it would take up to much space here to show them all in one example. The ... represents stuff that are left out.

**Figure 9: A Portlet Element**

```
<portlet>
  <description>A QuickClip Portlet that has a fixed start URL set by an administrator.</description>
  <portlet-name>QuickClipFixedStartURL</portlet-name>
  <portlet-class>com.kapowtech.quickclip.QuickClip</portlet-class>
  <init-param>
    <description></description>
    <name>com.kapowtech.robosuite.ClipDescriptor.Location</name>
    <value>/quickclip/quickclip.clip</value>
  </init-param>
  <init-param>
    <description></description>
    <name>com.kapowtech.robosuite.jspPath</name>
    <value>/quickclip/</value>
  </init-param>
  <supports>
    <mime-type>text/html</mime-type>
    <portlet-mode>edit</portlet-mode>
    <portlet-mode>help</portlet-mode>
  </supports>
  <portlet-info>
    <title>1. QuickClip - Fixed Start URL</title>
  </portlet-info>
  <portlet-preferences>
    <preference>
      <name>startUrl</name>
      <value>http://www.google.dk</value>
      <read-only>>false</read-only>
    </preference>
    ...
  </portlet-preferences>
</portlet>
```

All the preferences will be described below and Table 2 will summarize these and their possible values.

## startUrl

This preference value determines both the first page shown in the clip and the part of the remote sites domain that will be shown in the portlet. If the startURL is <http://www.kapowtech.com> then the QuickClip Portlet will start loading this URL into its view. If a user navigates in the portlet,

for example, click on a link then if the URL of this link is in a domain that matches the pattern \*.kapowtech.com then the new page will be shown in the portlet's view; otherwise it would be shown in a new browser window or it would not be shown at all (this depend on the setting of the browseOutsideDomain preference described shortly). If the startUrl does not start with www then only URLs with domain equal to that of the startUrl will be shown in the portlets view. Depending on the portlet's type (the value of the preference portletType), the startUrl may be set by an administrator or by an end-user. If the type is Common (portletType=**COMMON**), then the startUrl is set by an administrator. If the type is Individual (portletType=**INDIVIDUAL**) and does not require authentication (authMethod=**NONE**), then the startUrl may be change by an end-user. In this case, an administrator will set the initial value of the startUrl, but the end-user can then change this from the portlet's edit page.

### browseOutsideDomain

This preference value is closely coupled to the startUrl. It determines what will happen when an end-user clicks on a link in the portlet to a page out side the domain specified by the startUrl. If browseOutsideDomain is set to **TRUE** then such an action will result in the target pages opening in a new browser window. If it is set to **FALSE** then the link will be deactivated and the target page may not be accessed.

### authMethod

This preference value specify whether the portlet will have to authenticate the user on the remote site and if so what method to use. The possible values are:

- **NONE** meaning that authentication is not required.
- **BASIC** meaning that basic HTTP based authentication is required by the remote site.
- **FORM** meaning that form based authentication is required by the remote site.

### portletType

This preference value determines the type of the portlet. By type we mean whether the portlet may be configured individually by an end-user or not. What the end-user may configure depends on values of other preferences, but the portlet type should always be set to **INDIVIDUAL** in such cases. If the portlet type is set to **COMMON** then all preferences have to be set by an administrator. The rule determining which preferences the end-user may configure when the portlet type is **INDIVIDUAL** is as follows:

**Figure 10: The AuthMethod Preference**

AuthMethod=**NONE**: end-user may configure startUrl and browseOutsideDomain  
AuthMethod?**NONE**: end-user may configure individualUsername and individualPassword.

### individualUsername

This is the individual user name supplied by end-users. It should never be set by administrators, but the preference should always be present in the portlet element for the portlet if this requires authentication.

## individualPassword

This is the individual password supplied by end-users. It should never be set by administrators, but the preference should always be present in the portlet element for the portlet if this requires authentication. The password is encrypted before it is saved in the portal's backing store. If an administrator sets the password by accident, then the portlet will ignore this since this would be an error and it would most likely not be encrypted correctly.

## loginFormAction, loginFormMethod, usernameParameterName and passwordParameterName

The loginFormAction preference is the URL value from the action attribute of the login form of the remote site. The loginFormMethod specifies whether the login form uses POST or GET method to submit the form. The usernameParameterName and the passwordParameterName are the parameter names used by the login form to submit the username and password values. As an example take look at the login form in Figure 11. In this case the preferences should be set to the values shown in Table 1. If the action is not an absolute URL you should make it absolute.

**Figure 11: A Sample Login Form**

```
<form method="POST" action="http://www.kapowtech.com/formauth/login/j_security_check">
  <table>
    <tr><td width="100">Username:</td><td><input type="text" name="j_username"></td></tr>
    <tr><td>Password:</td><td><input type="password" name="j_password"></td></tr>
    <tr><td></td><td><input type="submit" value="Login"></td></tr>
  </table>
</form>
```

**Table 1: Preferences Values for the Sample Login Form**

loginFormAction	http://www.kapowtech.com/formauth/login/j_security_check
loginFormMethod	POST
usernameParameterName	j_username
passwordParameterName	j_password

## administratorRole

This preference specifies which roles on a portal should be considered as administrators with respect to the QuickClip Portlet. The only effect this has is that user belonging to any of these user groups will see more detail error messages when error occurs in a QuickClip Portlet. The purpose of this is to make it easier for creators of QuickClip Portlets to verify that their portlets are working correctly and in case of errors quickly see what the cause is without having to browse through thousands of lines of log files.

## Preference Summary

Table 2 summarizes the preferences used by the QuickClip Portlet.

**Table 2: The QuickClip Preferences**

Preference name	Meaning	Examples
startUrl	The URL first loaded into the clip. The start URL also specifies the	http://www.kapowtech.com

	clipped domain.	
browseOutsideDomain	The value of this preference specifies what is going to happen with links that lead outside the clipped domain. If the value is set to FALSE the links are inactive and if it is set to TRUE, the links will open in a new browser window.	TRUE, FALSE
authMethod	This is the authentication method. Possible values are: NONE, if the site requires no authentication, FORM if the site uses form based login, BASIC if the site uses basic HTTP based authentication	NONE, FORM, BASIC
portletType	This specifies whether the portlet is for common or individual use.	COMMON, INDIVIDUAL
individualUsername	The username used for individual login.	joe
individualPassword	The username used for individual login. This can only be set by a portal user from the edit page of the portlet.	xY4ZQ5
commonUsername	The username used for single login, i.e. when the credentials are set by the administrator. Can never be set from the edit page.	joe
commonPassword	The password used for single login, i.e. when the credentials are set by the administrator. Can never be set from the edit page.	xY4ZQ5
loginFormAction	This is the action from the form that the remote site uses for login.	<a href="http://www.kapowtech.com/formauth/login/j_security_check">http://www.kapowtech.com/formauth/login/j_security_check</a>
loginFormMethod	This specifies whether the login form uses method GET or POST. Possible values are POST and GET.	POST, GET
usernameParameterName	This is the name of the username field in the login form.	j_username
passwordParameterName	This is the name of the password field in the login form.	j_password
administratorRole	This specifies which roles on a portal should be considered as	Admin, PortalSystemAdministrator etc.

	administrators with respect to the QuickClip Portlet.	
--	---	--

For a given portlet not all preferences may need to be specified. Which preferences that will be needed may depend on values of other preferences. The following preferences are always needed: `startUrl`, `browseOutsideDomain`, `authMethod`, `portletType` and `administratorRole`. The remaining preferences are optional, but may be required if other preferences have given values. Table 3 shows these dependencies.

**Table 3: Preferences Dependencies**

Preferences settings	Required preferences
portletType= <b>COMMON</b> and authMethod= <b>FORM</b>	commonUsername, commonPassword, loginFormAction, loginFormMethod, usernameParameterName, passwordParameterName
portletType= <b>INDIVIDUAL</b> and authMethod= <b>FORM</b>	individualUsername, individualPassword, loginFormAction, loginFormMethod, usernameParameterName, passwordParameterName
portletType= <b>COMMON</b> and authMethod= <b>BASIC</b>	commonUsername, commonPassword, loginFormAction
portletType= <b>INDIVIDUAL</b> and authMethod= <b>BASIC</b>	individualUsername, individualPassword, loginFormAction

## Creating a new QuickClip Portlet

In order to create a new QuickClip Portlet you should add a new portlet element to the `portlet.xml` file. You may do this by copying an existing one, for example, from one of the samples, and you should then configure this to fit your own needs. You could also start from scratch and build up your own portlet element. In the next chapter we will give examples of this which will cover all the possible ways that the preferences of the QuickClip Portlet may be configured. Figure 12 shows a template for such a portlet element. In this you should replace everything in square brackets [] with your own values. The text in the square brackets explains the purpose of the value.

**Figure 12: A Template for a QuickClip Portlet Element**

```
<portlet>
  <description>[A description of your portlet]</description>
  <portlet-name>[The name of your portlet]</portlet-name>
  <portlet-class>com.kapowtech.quickclip.QuickClip</portlet-class>
  <init-param>
    <description></description>
    <name>com.kapowtech.robosuite.ClipDescriptor.Location</name>
    <value>/quickclip/quickclip.clip</value>
  </init-param>
  <init-param>
    <description></description>
    <name>com.kapowtech.robosuite.jspPath</name>
    <value>/quickclip</value>
  </init-param>
</portlet>
```

```
<supports>
  <mime-type>text/html</mime-type>
  <portlet-mode>edit</portlet-mode>
  <portlet-mode>help</portlet-mode>
</supports>
<portlet-info>
  <title>[The title of your portlet]</title>
</portlet-info>
<portlet-preferences>
  [Preferences]
</portlet-preferences>
</portlet>
```

Especially you should replace [Preferences] with your own set of preferences. This means that you for every preference in Table 2 should add an element of the form:

```
<preference>
  <name>[Name] </name>
  <value>[Value]</value>
  <read-only>[true|false]</read-only>
</preference>
```

where Name is the name of the preference and Value its value. If the end-user should be able to modify a preference value then you need to give it the value false for its read-only property. An example would be:

```
<preference>
  <name>startUrl</name>
  <value>http://www.google.dk</value>
  <read-only>false</read-only>
</preference>
```

Once you have made your portlet definition in the `portlet.xml` file you should also create a portlet in your project. To do this right-click on the location in your project where you want to place the portlet file and choose: **New > Portlet**. Give your portlet file a name and press **Create**. Now, select the portlet type Java Portlet and press **Next**. On the next page you choose the Existing Portlet option and select your portlet from the list. Finally, press on finish and you have created a new portlet that you may place on your portal.

## The QuickClip Sample Portal

If you installed the QuickClip Portlet with samples, you may have noticed that a sample portal was also installed. In this chapter we will give a description of this portal and of the six sample portlets on it. When you install the QuickClip Portlet with samples, some extra files are installed in a folder called samples in the quickclip folder. These are:

- `quickclip-sample.portlet`: this is the sample portal.
- `quickClipBasicIndividual.portlet`, `quickClipBasicSingle.portlet`, `quickClipFormIndividual.portlet`, `quickClipFormSingle.portlet`, `quickClipNoAuthIndividual.portlet`, `quickclipNoAuthSingle.portlet`: these are the six sample portlets of the portal
- `about.html`, `about.portlet`, `login.jsp`, `login.portlet`: these are some extra portlets that the sample portal uses.

Besides installing these files one more file will be installed:

- `portlet.xml`: this is a Java Portlet configuration file and it contains the configuration of the six sample portlets.

### Running the Sample Portal

To run the sample portal you should first make sure that you have a RoboServer running.

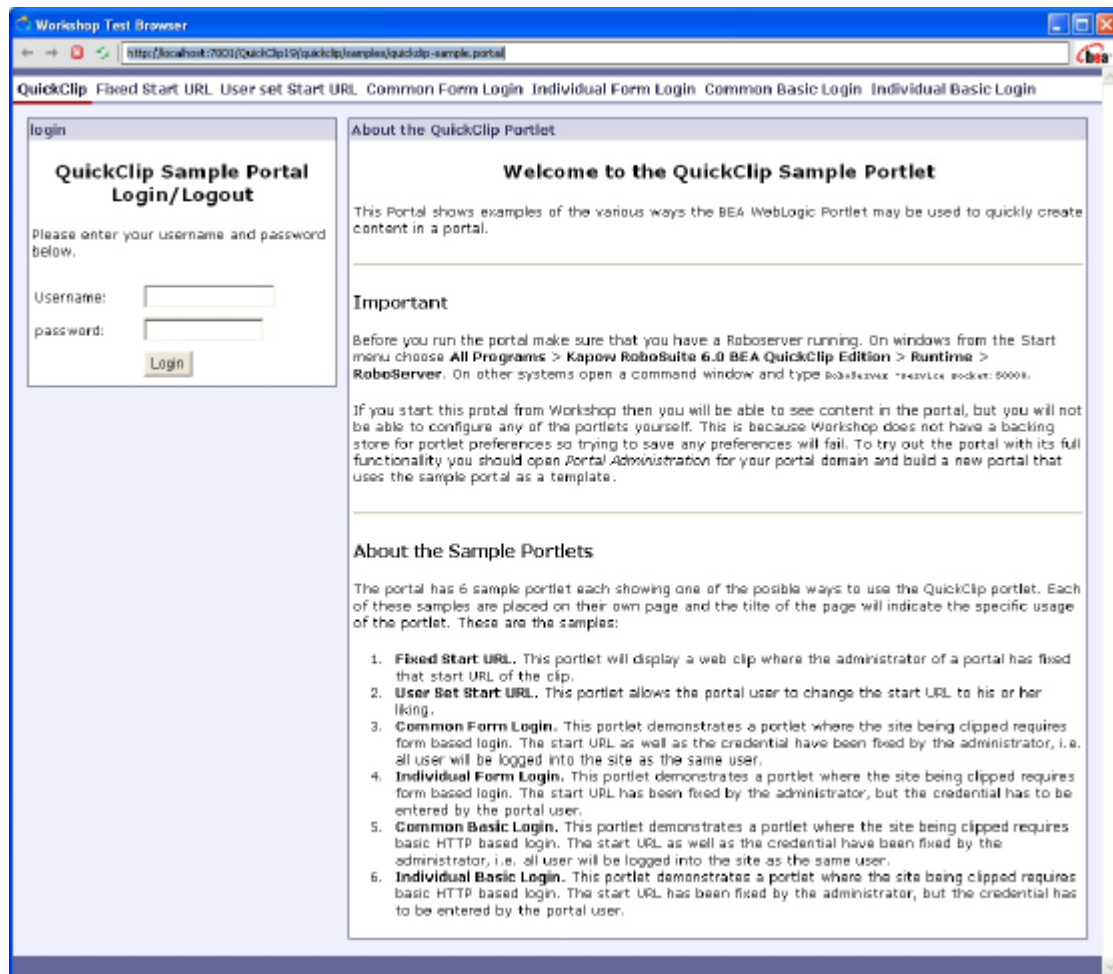
If you are running on a Windows machine you may start a RoboServer by going to the Windows Start Menu and starting it from there. You locate the menu item for this in: **All Programs > BEA QuickClip > Roboserver**. To stop the RoboServer again you locate the menu item for this in: **All Programs > BEA QuickClip > Shutdown Roboserver**.

If you are running on a UNIX systems you should open a command window and type:

```
RoboServer -service socket:50000
```

Once you have started a RoboServer it is time to start the portal. To do this you must first make sure that your portal is shown in the Edit pane of Workshop, then press on the Run button in the Workshop toolbar. Workshop may ask you if it should start the WebLogic server. If you have not done this already and you should accept this. After a while you will see the portal in the Test Browser (See Figure 13). Log in to the portal either as the portal administrator (default credentials: `weblogic/weblogic`) or some other user you know exist on your WebLogic server domain. Once you are logged in you may have a look at the different samples. One thing that you should be aware of is that when a portal is launched from Workshop it will not have a backing store for the portlet preferences. This means that you cannot change any preferences as an end-user. To do this you must create a portal in the Portal Administration console that has a desktop using the sample portlet as a template. How to do this is described in the Administration Guide.

Figure 13: The QuickClip Sample Portal



## The Sample Portlets

We will now describe the six basic use cases that the QuickClip Portlet was designed to handle. These six use cases correspond directly to the six sample portlets that were installed with the QuickClip Portlet samples.

### Fixed Start URL

In this case, the QuickClip Portlet is designed to have a fixed URL that cannot be changed by the end user and where the remote site does not require authentication. This is the same portlet as the one that was used in the tutorial in the previous chapter. For this kind of portlet you only need to configure very few of the preferences. First of all we need to specify a start URL. This is the URL of the page that will first be loaded into the clip. As explained above the start URL also determines the domain that is clipped. Since the remote site does not require authentication, we need to set the authMethod preference to **NONE**. We also need to set the portletType to **COMMON** since this indicates that there is no individual configuration required by the end user. The remaining preference does not need to be set and is ignored if so.

**Table 4: Fixed Start URL Preference Summary**

Preference	Value	Change by End User
startUrl	The start URL of the remote site	No
authMethod	NONE	No
portletType	COMMON	No
browseOutsideDomain	FALSE	No
administratorRole	Admin, PortalSystemAdministrator	No

### Individual Start URL

In this case the QuickClip Portlet is designed to have a URL that can be changed by the end user. It may initially have been set to some URL by the Administrator, but every user can set the URL to his/her own preferred value. The only difference between this setting and the one in the previous examples is that we should change the value of the portletType to **INDIVIDUAL** such that the end user may set the start URL. Besides being able to set the start URL, the end user may now also set the value of the browseOutsideDomain preference.

**Table 5: Individual Start URL Preference Summary**

Preference	Value	Change by End User
startUrl	The start URL of the remote site	Yes
browseOutsideDomain	FALSE	Yes
authMethod	NONE	No
portletType	INDIVIDUAL	No
administratorRole	Admin, PortalSystemAdministrator	No

### Common Form Based Authentication

In this example the remote site will use a form to log the user into the site. In order to clip the site and ensure that the user of the portlet will be always be logged in when she opens the portal with the clip, a credential for the site must be specified as preference values. In this example, we will assume that the administrator will specify a common set of credentials that all users will use. To configure this kind of setup, you need to examine the remote sites login form and find out what the action URL looks like. After the portlet has authenticated the user it will show the content of the start URL (specified by the startUrl preference value).

**Table 6: Common Form Based Authentication Preference Summary**

Preference	Value	Change by End User
startUrl	The start URL of the remote site	No
authMethod	FORM	No
portletType	COMMON	No
browseOutsideDomain	FALSE	No
loginFormAction	The action URL of the remote sites login form	No
loginFormMethod	POST	No
usernameParameterName	The name of the input element of the form that contains the user name, for example, username	No
passwordParameterName	The name of the input element of the form that contains the password, for example, password	No
commonUsername	The actual user name of the common	No

	account	
commonPassword	The actual password of the common account	No

### Individual Form Based Authentication

This is similar to Common Form Based Authentication, except that the end user has to supply the credentials. So setting up the portlet is just as simple as for the common case since the preferences should just be given the same kind of values. However, no user credentials need to be set. Even if one were to set the credential, this would not work. There would be no problem with setting the user name, but it would result in everyone getting the same default user name, which defeats the purpose of having individual credentials. Setting the password would however not work at all, since individual passwords need to be encrypted, and there is no way to get an encrypted password to enter into the password preference. After the portlet has authenticated the user it will show the content of the start URL (specified by the startUrl preference value).

**Table 7: Individual Form Based Authentication Preference Summary**

Preference	Value	Change by End User
startUrl	The start URL of the remote site	No
authMethod	FORM	No
portletType	INDIVIDUAL	No
browseOutsideDomain	FALSE	No
loginFormAction	The action URL of the remote sites login form	No
loginFormMethod	POST	No
usernameParameterName	The name of the input element of the form that contains the user name, for example, username	No
passwordParameterName	The name of the input element of the form that contains the password, for example, password	No
individualUsername	The actual user name of the end users account	Yes
individualPassword	The actual password of the end users account	Yes

## Common Basic HTTP Based Authentication

If the remote site requires Basic HTTP based authentication, all the QuickClip Portlet needs to know is the credential for the site in order to log in to the site. When the RoboServer running the QuickClip robot first access the remote site, it will be challenged to provide the user's credentials to the site and all this will handled automatically by the robot. To specify that the portlet should use Basic HTTP based authentication, the preference authMethod should be set to **BASIC**. After the portlet has authenticated the user it will show the content of the start URL (specified by the startUrl preference value). The rest of the configuration is straightforward and is summarized in Table 6.

**Table 8: Common Basic HTTP Based Authentication Preference Summary**

Preference	Value	Change by End User
startUrl	The start URL of the remote site	No
authMethod	BASIC	No
portletType	COMMON	No
browseOutsideDomain	FALSE	No
commonUsername	The actual user name of the common account	No
commonPassword	The actual password of the common account	No

## Individual Basic HTTP Based Authentication

This is similar to Common Basic HTTP-based authentication, except that the end user supplies the credential. Again the same comments we made about for Individual Form-based authentication about configuring the user name and password apply here. After the portlet has authenticated the user it will show the content of the start URL (specified by the startUrl preference value). Table 9 summarizes the configuration.

**Table 9: Individual Basic HTTP Based Authentication Preference Summary**

Preference	Value	Change by End User
startUrl	The start URL of the remote site	No
authMethod	BASIC	No
portletType	INDIVIDUAL	No
browseOutsideDomain	FALSE	No
individualUsername	The actual user name of the end users account	Yes
individualPassword	The actual password of the end users account	Yes

## Configuration

This chapter describes more advanced configuration of the QuickClip Portlet.

### ***Starting a RoboServer***

There are several different methods that you may use to set up your system to start a RoboServer. The choice of method depends on your system (Windows, Linux, etc.) and on what you are using the QuickClip Portlet for.

#### **On Windows**

On Windows systems a RoboServer will be installed as a service when you install the QuickClip Portlet in your BEA installation. If you want a RoboServer to be running whenever your Windows machine is running, you may just set the RoboServer service to start automatically when Windows starts. You do this by opening Windows Control Panel and from there opening the Administrative Tools. In the Administrative Tools you open Services and locate the service called RoboServer 6.1 QuickClip Edition and set its Startup Type to Automatic.

If you do not want to have a RoboServer running all the time you may just start it whenever you need it by going to the Windows Start Menu and starting it from there. You locate the menu item for this in: **All Programs > BEA QuickClip > Roboserver**. To stop the RoboServer again you locate the menu item for this in: **All Programs > BEA QuickClip > Shutdown Roboserver**.

A last method is to modify the start/stop script for your WebLogic Portal Server domain such that a RoboServer will be started/stopped whenever you start/stop your WebLogic Server. How to do this is described in the Installation Guide under the section Domain Upgrade.

#### **On UNIX**

On UNIX systems you should open a command window and type:

```
RoboServer -service socket:50000
```

To stop the RoboServer again you open a command window and type:

```
ShutdownRoboServer localhost 50000 2
```

If you have started your RoboServer on another port than 50000 you type this instead of 50000.

An alternative method is to modify the start/stop script for your WebLogic Portal Server domain such that a RoboServer will be started/stopped whenever you start/stop your WebLogic Server. How to do this is described in the Installation Guide under the section Domain Upgrade.

### ***Changing the RoboServer Port Number***

Changing the port number used for communication with the RoboServer is done in the following way. You open the clip configuration file (located in the same folder as the JSP files) called quickclip and change the value of the `port` attribute in the `socket-object-protocol`. If you want the new port to be 60000 we change it as follows:

```
<robot-server>
  <socket-object-protocol port="60000"/>
</robot-server>
```

This will change the port that the QuickClip Portlets will use when connecting to the RoboServer, but you also have to make sure that your RoboServer will be listening on the same port.

## In the Windows Start Menu

To change the port for the Windows Start Menu item that starts RoboServer, locate the item in the Start Menu, right-click on this, and select Properties. Find the Target attribute of the shortcut and change the number that follows immediately after the text `socket:` (default value is 50000) to the new port number.

## In Start/Stop Scripts

The start and stop scripts are located in the `weblogic81\portal\thirdparty\quickclip` folder under the BEA home where you installed the QuickClip Portlet. The files are called `startQuickClip` and `stopQuickClip` with `.cmd` extension under Windows and `.sh` under UNIX. Open these files and change the port number in these. In the `startQuickClip` file the port number follows immediately after the text `socket:`. In the `stopQuickClip` file the port number is the follows the text `localhost`.

## In the Windows Service

To do this you should locate the file `robosuite.properties` located in the folder

```
<weblogic_home>/portal/thirdparty/quickclip/Kapow RoboSuite 6.1 BEA QuickClip  
Edition/Configuration
```

where `<weblogic_home>` is the WebLogic home folder, for example, `C:\bea\weblogic81`. In the file `robosuite.properties` locate the line

```
roboServerServiceParameters=-service socket\50000
```

and change the number at the end to your new port number.

## Changing the Source Folder

Although not recommended, it is possible to change the location of the source folder, i.e. where the JSP pages, `quickclip.clip` etc. is located. To change the location you need to make two changes to your portlet definition in the `portlet.xml` file and one in your `web.xml` file. In `portlet.xml` you should change the two init parameters for the portlets such that they include your new location:

- `com.kapowtech.robosuite.ClipDescriptor.Location`. This tells the portlet where the clip configuration file is located and what it is called.
- `com.kapowtech.robosuite.jspPath`. This tells the portlet where the JSP-file is located.

The example below shows how these two parameters have been changed.

```
<init-param>  
  <description>Location of the RoboSuite Clip Descriptor...</description>  
  <name>com.kapowtech.robosuite.ClipDescriptor.Location</name>  
  <value>/resources/quickclip/quickclip.clip</value>  
</init-param>  
<init-param>  
  <description>Location of the RoboSuite Clip Descriptor...</description>  
  <name>com.kapowtech.robosuite.jspPath</name>  
  <value>/jsp/quickclip/</value>  
</init-param>
```

In `web.xml` you need to change the servlet mapping to point to your new location of the `quickclip.clip` file. The example below shows such a change.

```
<servlet-mapping>  
  <servlet-name>QuickClipServlet</servlet-name>
```

## Configuration

```
<url-pattern>/resources/quickclip/quickclip.clip</url-pattern>  
</servlet-mapping>
```

Notice that it is even possible to have different locations for the clip descriptor and the JSP-files. If you change the location of the JSP files for the sample application you also need to make changes to the sample portal and the sample portlets, since these are designed to work with the default settings.

## **Localization**

The QuickClip Portlet has been localized. This is done by having all texts, URLs, etc. be placed in a resource bundle. Initially the only locale supported is US English. The bundle file is called `quickClipBundle.properties` and is placed in the file `kapow-quickclip.jar`. In this JAR-file the bundle is placed in the package `com.kapowtech.quickclip.messages`. In order to create a new bundle for a new locale you should take a copy of this bundle file and rename it with the proper locale identifier appended to `quickClipBundle` for example, `quickClipBundle_fr_FR.properties` for France. You should then translate all the properties in this file to what you find relevant for your locale, for example, you might translate the texts but leave some of the URLs unchanged. Finally you should place the file in the folder `WEB-INF/classes/com/kapowtech/quickclip/messages`.

## Limitations

The QuickClip Portlet has the advantage that it is simple to configure, but it is a light weight version of the Kapow Enterprise Web Clipping platform and as such it only has limited functionality compared with that of an enterprise clipping portlet. In this chapter we will list most of the limitations.

### **QuickClip**

There is no possibility to do any form of restriction on the clips – the QuickClip portlet only does whole site clipping. In the enterprise solution one may remove element from a clip, for example, remove a logout link/button or one may restrict a clip to just part of a remote application. This is not possible with the QuickClip Portlet, since this operates with a fixed robot.

The QuickClip Portlet can only clip from one domain and this always the same as the domain of the start URL in the enterprise solution there is no such restriction.

The QuickClip Portlet can only access a RoboServer running on localhost, i.e. the same machine as the one the portal server is running on. This means that the RoboServer will compete with the portal server for CPU resources. There is no possibility for load balancing RoboServers.

There is no support for resource clipping. Resource clipping means if one is clipping from sites that requires authentication and the resources, for example, images on the web site also require authentication then these will not show up in the clip. The enterprise solution has support for this.

There is no possibility for style (skin) changes to the clip. In the enterprise solution it is possible to add style change rules in robots so that one may change the layout of the clip to make it look like hat of the portal.

Only a fixed set of events on a clip, for example, change, click, double click, etc. This means that more advanced JavaScript events will not work in the clip. In the enterprise solution you may configure which event the clip should be able to handle based on your knowledge of the remote site.

There is no way to apply workarounds to JavaScript loaded sites or other sites that the clipping solution cannot handle.

There is no way to add extra input parameters to further configure a clip. In the enterprise clipping solution one may add extra input objects to a robot and use these to parameterize the way the clip works. In the QuickClip Portlet the robot is fixed, so there is no way to add extra input object to this because the robot would not be able to use these anyway.

Only HTTP Basic and Form based authentication is supported by the QuickClip Portlet and Form based login is limited to form submission that may be expressed as an URL. There is no support for additional post parameters. In the enterprise solution a range of other possibilities for authentication is possible.

Also the QuickClip Portlet has no support for encryption of passwords for common credentials, since these have to be entered via the WebLogic Administration Portal where this is not possible. The enterprise solution has more advanced support for SSO (single sign-on) and has integration into BEA WebLogic Portal's SSO solution.

Furthermore, the enterprise web clipping solution has support for logging, such as for management using RoboManager, and has support for use of proxy server for the access to the remote site.

### ***Web Clipping in General***

The QuickClip Portlet is based on the general Web Clipping solution from Kapow. Although this is a very flexible clipping product it does have some general limitation. For some of these one may find work-around when using full Enterprise Web Clipping while some are for the time being inherent in the product.

There is only very limited support for Flash, ActiveX and other similar plugins. Some Flash application does however work in IE 6.0.

Some JavaScript may not work properly in clips. This mainly has to do with complex DHTML like drop down menus. It is hard to give a complete description of when and when not JavaScript will work and in many case one may find work-around when using full Enterprise Web Clipping.

## Index

- authMethod, 10
- browseOutsideDomain, 10
- individualPassword, 11
- individualUsername, 10
- Java Portlet, 1, 8
- JSR-168, 8
- Kapow, 1
- Kapow Technologies, 1
- Localization, 24
- loginFormAction, 11
- loginFormMethod, 11
- passwordParameterName, 11
- port number, 22
- portlet.xml, 9
- portletType, 10
- QuickClip
  - preferences, 9
- QuickClip Portlet
  - creating a, 13
- RoboServer, 2, 8, 22
  - port number, 22
  - starting, 22
- startUrl, 9
- usernameParameterName, 11
- Web Clipping, 1
- WebLogic, 1
- Workshop, 1, 2
  - application, 2
  - portal, 5