

Smile EZ Publish Eclipse Plugin: An Installation Walkthrough

DRAFT Version 0.12

Paul Wilson, 14th March 2007

Document Purpose

The purpose of this document to provide an overview of the Smile EZ Publish Eclipse Plugin and its installation process.

This is Draft material developed to help the EZ community, and because this extension looks like it can bring some important capabilities and opportunities to EZ and the EZ community.

Many thanks to the team at Smile (www.smile.fr).

This document probably contains more detail than most experienced EZ developers/users will need, but it is presented in this way to help people who are new to EZ to understand and make use of this extension. This is important because helping more people to get more value out of EZ Publish more quickly is a way to grow and strengthen the EZ community ... which helps us all.

I hope this helps you.

Regards

Paul Wilson
March 2007

Contents

1	Eclipse – Overview Information	3
2	Smile EZ Publish Eclipse Plugin	4
3	Installation – Eclipse & EZ Publish.....	7
3.1	EZ Publish - Installing the SmileClasses Extension	7
3.2	Download Eclipse	8
3.3	Adding the Smile Plugin to Eclipse (Automatic Update Method).....	10
3.4	Adding the Smile Plugin to Eclipse (Manual Method).....	14
3.5	Loading EZ Publish Content Classes into Eclipse.....	15
4	Smile EZ Publish Eclipse Plugin - What it Does / a Brief How-to	18
4.1.1	Template Syntax Highlighting	18
4.1.2	Content Class Browser.....	18
4.1.3	Code generation for operator extensions	20

Important Disclaimer

This documentation is provided on a no liability basis. It is based on personal notes developed from using EZ Publish. Please contact EZ Systems (www.ez.no) or an EZ accredited partner / professional developer if you need to have reliable, thorough, and up-to-date information or services.

About the Author

Paul Wilson

B.Com, G.Dip Sc (IT), B.Comp (Hons)

Some of Paul's work experience has included:

- Designing strategies, projects and education programs to accelerate the adoption of e-commerce by businesses and industries in regional Australia – as part of the Australian Government's first major program of this type (1998-2002);
- Co-authoring university (MBA) and national industry training courses on e-commerce and Internet-related business strategies;
- Co-authoring research and strategy reports on national ICT training needs across regions and industries for Australian State and National government organisations.

Paul has been using EZ Publish to further his (now discontinued) PhD studies and professional work, with a view to creating shared knowledge systems for businesses, industries and community organisations.

Paul lives in Tasmania, Australia. He enjoys getting away from a computer screen for mountain-biking and (too infrequently now) sailing.

1 Eclipse – Overview Information

Some introductory information on Eclipse follows.

“Eclipse is an open source community whose projects are focused on building an open development platform comprised of extensible frameworks, tools and runtimes for building, deploying and managing software across the lifecycle. A large and vibrant ecosystem of major technology vendors, innovative start-ups, universities, research institutions and individuals extend, complement and support the Eclipse platform.”

(source: www.eclipse.org, accessed 13th March 2007)

What is Eclipse?

Eclipse is an open source community whose projects are focused on building an extensible development platform, runtimes and application frameworks for building, deploying and managing software across the entire software lifecycle. Many people know us, and hopefully love us, as a Java IDE but Eclipse is much more than a Java IDE.

The Eclipse open source community has over 60 open source projects. These projects can be conceptually organized into seven different "pillars" or categories:

1. *Enterprise Development*
2. *Embedded and Device Development*
3. *Rich Client Platform*
4. *Rich Internet Applications*
5. *Application Frameworks*
6. *Application Lifecycle Management (ALM)*
7. *Service Oriented Architecture (SOA)*

The Eclipse community is also supported by a large and vibrant ecosystem of major IT solution providers, innovative start-ups, universities and research institutions and individuals that extend, support and complement the Eclipse Platform.

(source: www.eclipse.org/home/newcomers.php, accessed 13th March 2007)

References - Eclipse

- Eclipse Site
www.eclipse.org
- Eclipse Download Page
<http://www.eclipse.org/downloads/> (version 3.2.2 is around 125Mb)
- Eclipse - EZpedia Entry
<http://ezpedia.org/wiki/en/ez/eclipse>

2 Smile EZ Publish Eclipse Plugin

Information from EZ Publish Contributions area (see link below):

Author	Version	Compatible with
Nicolas Richeton	0.0.14	eZ Publish 3.x, Eclipse 3.2, Java 1.5

eZ Publish plugin for the Eclipse IDE. Syntax highlighting, code generation, content classes browser. This plugin is developed by Smile (<http://www.smile.fr>) and released under GPL.

Provides :

- * TPL Syntax highlighting (see below)
- * Code generation for operator extensions (see below)
- * Browser for all content classes defined in eZ Publish. (see below)

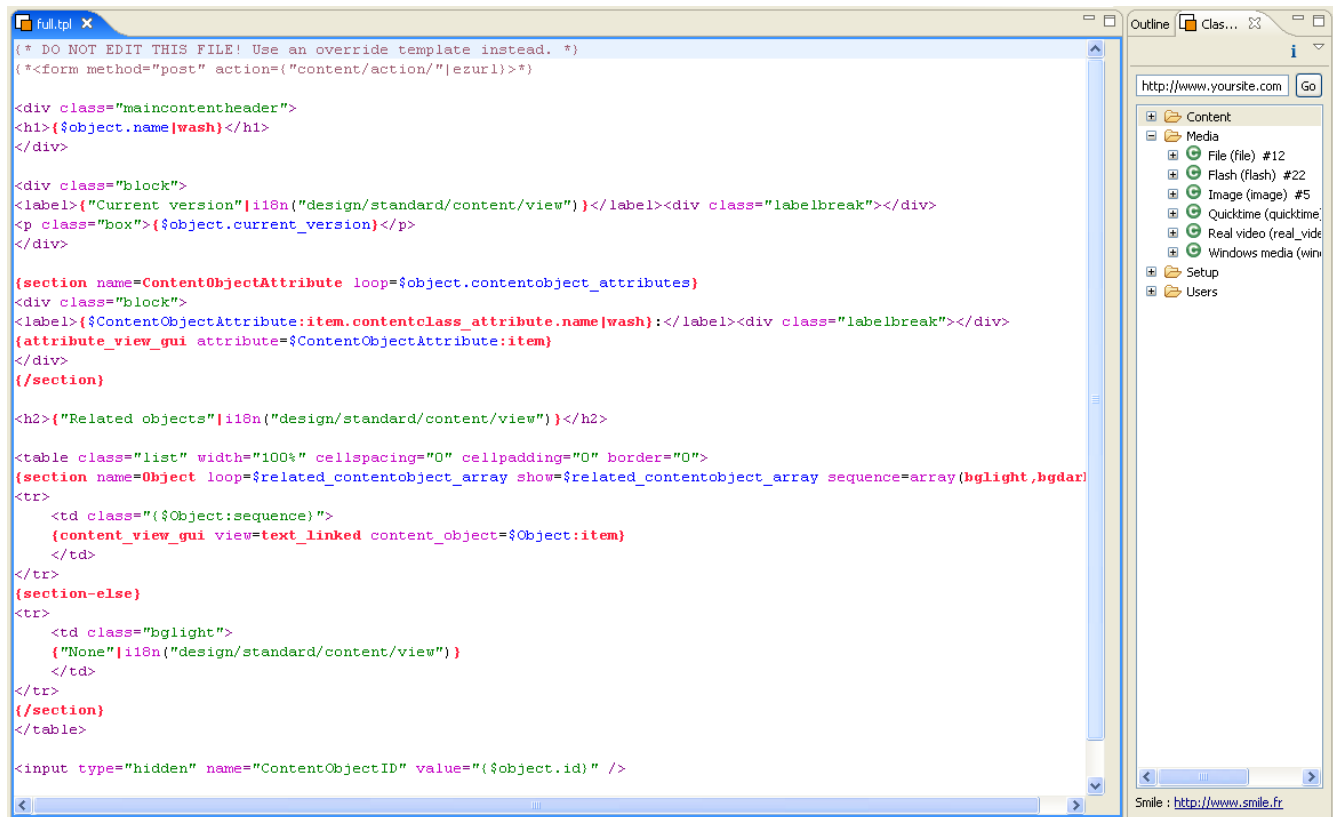


image: EZ tpl syntax highlighting and EZ Publish class browser (at right)

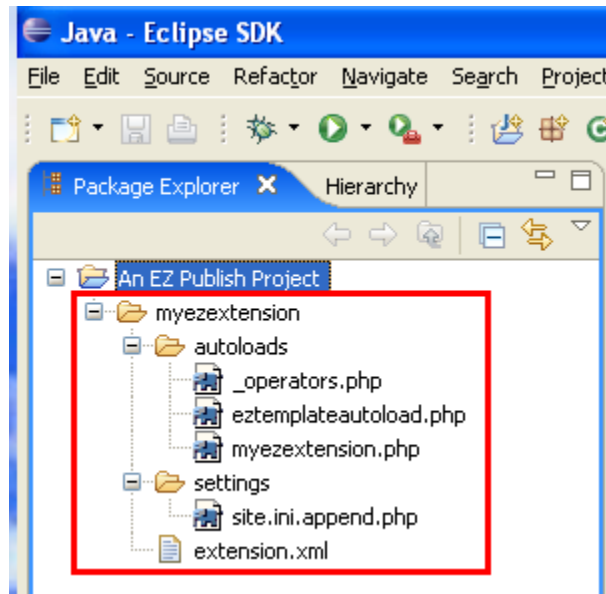


image: Operator Extension Code generation

References - Smile EZ Publish Eclipse Plugin

Note:

“Ideas and bug reports can be submitted on the sf.net site.” (Nicolas Richeton)

Sourceforge Project Homepage

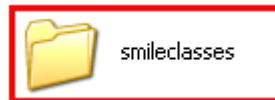
http://sourceforge.net/project/showfiles.php?group_id=190833

- Extension Developer – **Smile**
www.smile.fr
- Contribution information on ez.no
http://ez.no/community/contribs/3rd_party/smile_ez_publish_eclipse_plugin
- Sourceforge Project Homepage
http://sourceforge.net/project/showfiles.php?group_id=190833

3 Installation – Eclipse & EZ Publish

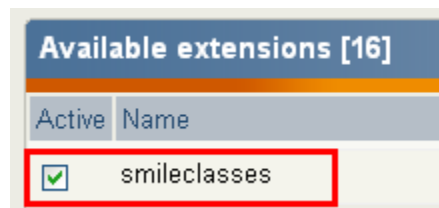
3.1 EZ Publish - Installing the SmileClasses Extension

The Smile plugin includes an EZ Publish plugins folder.



Upload this smileclasses folder to your EZ Publish installation's extension folder.

On your site's Admin/Setup/Extensions page, enable this extension



This is not necessary for setting up the extension, but you can check that this extension is working by going to the page enabled by the extension. For example:

www.yoursite.com/index.php/eng/smileclasses/export

An XML representation of your site's content object will then be generated. For example:

```

Address http://www.yoursite.com/index.php/smileclasses/export
Google G Go [Icons] Bookmarks No popups Check AutoLink AutoFill Send to

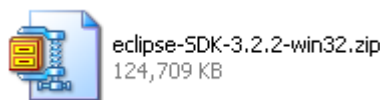
<?xml version="1.0" encoding="UTF-8" ?>
- <contentclasslist>
+ <group id="1" identifier="Content">
+ <group id="2" identifier="Users">
+ <group id="3" identifier="Media">
- <group id="4" identifier="Setup">
  - <class id="14" identifier="common_ini_settings" name="Common ini settings">
    <attribute id="159" identifier="name" name="Name" type="ezstring" is_required="0" />
    <attribute id="160" identifier="indexpage" name="Index Page" type="ezinisetting" is_required="0" />
    <attribute id="161" identifier="defaultpage" name="Default Page" type="ezinisetting" is_required="0" />
    <attribute id="162" identifier="debugoutput" name="Debug Output" type="ezinisetting" is_required="0" />
    <attribute id="163" identifier="debugbyip" name="Debug By IP" type="ezinisetting" is_required="0" />
    <attribute id="164" identifier="debugiplist" name="Debug IP List" type="ezinisetting" is_required="0" />
    <attribute id="165" identifier="debugredirection" name="Debug Redirection" type="ezinisetting" is_required="0" />
    <attribute id="166" identifier="viewcaching" name="View Caching" type="ezinisetting" is_required="0" />
    <attribute id="167" identifier="templatecache" name="Template Cache" type="ezinisetting" is_required="0" />
    <attribute id="168" identifier="templatecompile" name="Template Compile" type="ezinisetting" is_required="0" />
    <attribute id="169" identifier="imagesmall" name="Image Small Size" type="ezinisetting" is_required="0" />
    <attribute id="170" identifier="imagemedium" name="Image Medium Size" type="ezinisetting" is_required="0" />
    <attribute id="171" identifier="imagelarge" name="Image Large Size" type="ezinisetting" is_required="0" />
  </class>
</group>
</contentclasslist>

```

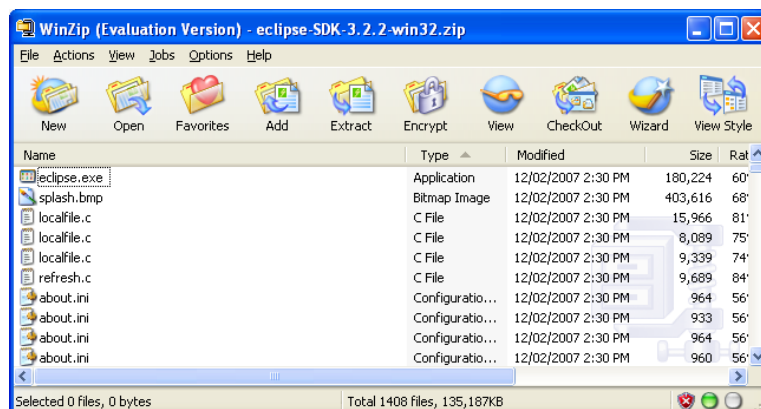
3.2 Download Eclipse

Eclipse is required for this extension to function.

If you do not already have Eclipse (version 3.2), it is available for download from the eclipse.org site's download page (<http://www.eclipse.org/downloads/>). Eclipse version 3.2.2 is around a 125Mb download.



Once downloaded, unzip Eclipse – perhaps to a folder in your programs directory.



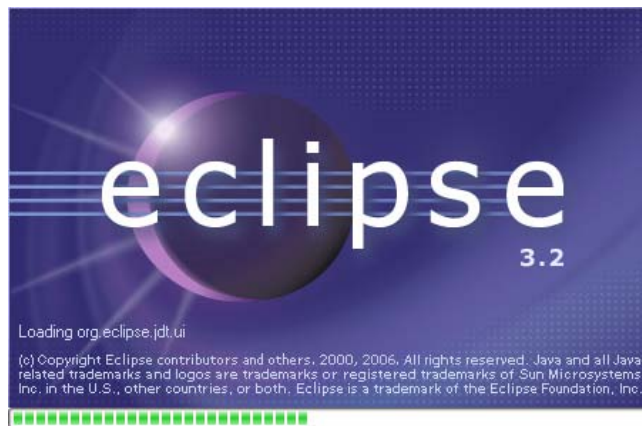
For more information and instructions, please check the Eclipse site.

Once Eclipse has been extracted, it can be run using “eclipse.exe” in the root directory of the extracted eclipse files.



(For usability purposes, if you are using Microsoft Windows, you may want to create a short-cut to eclipse.exe on your desktop)

The eclipse loading page will then be displayed and then environment loaded.

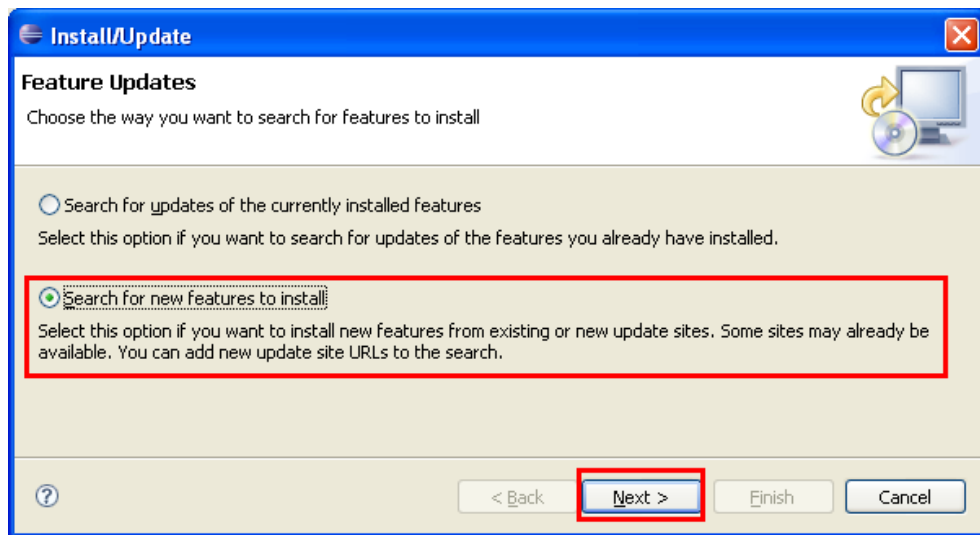


3.3 Adding the Smile Plugin to Eclipse (Automatic Update Method)

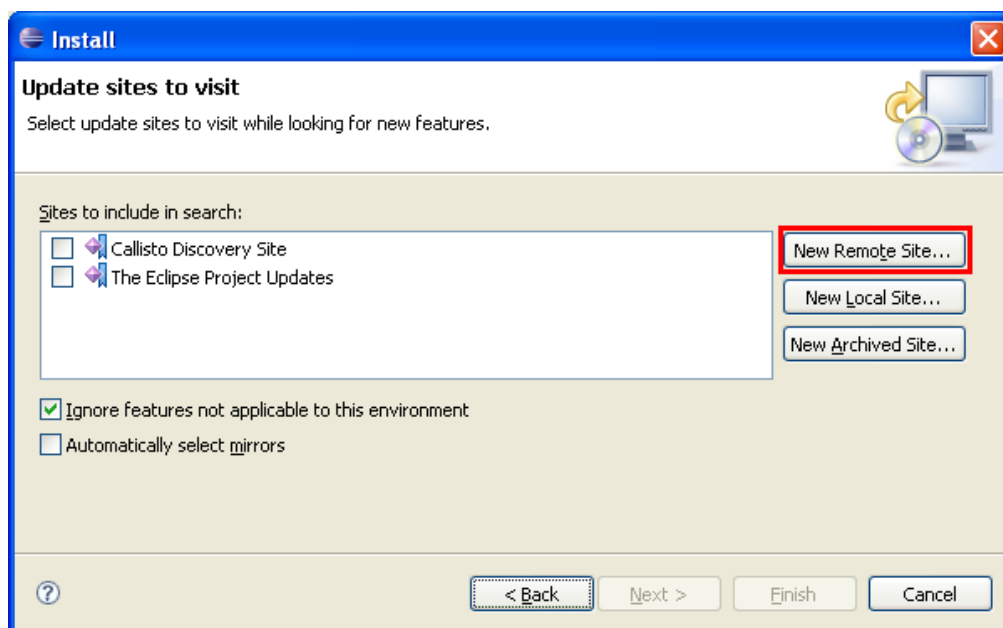
There are two methods for adding the Smile Plugin to Eclipse – an automatic method described here, and a manual one, described in section 2.6, below. Only one of these is required. This first Automatic Update method is more straightforward.

In Eclipse, from the main menu, select Help > Software Updates > Find and Install

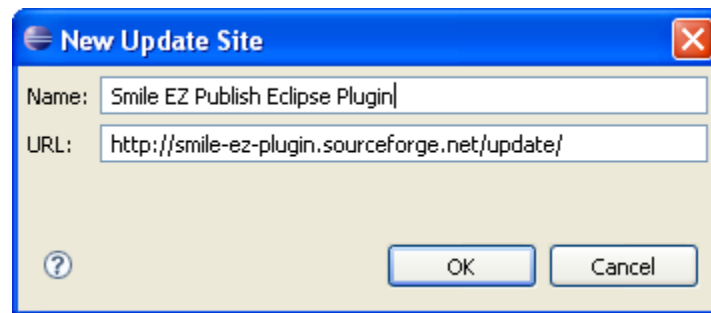
This will bring up the following window.



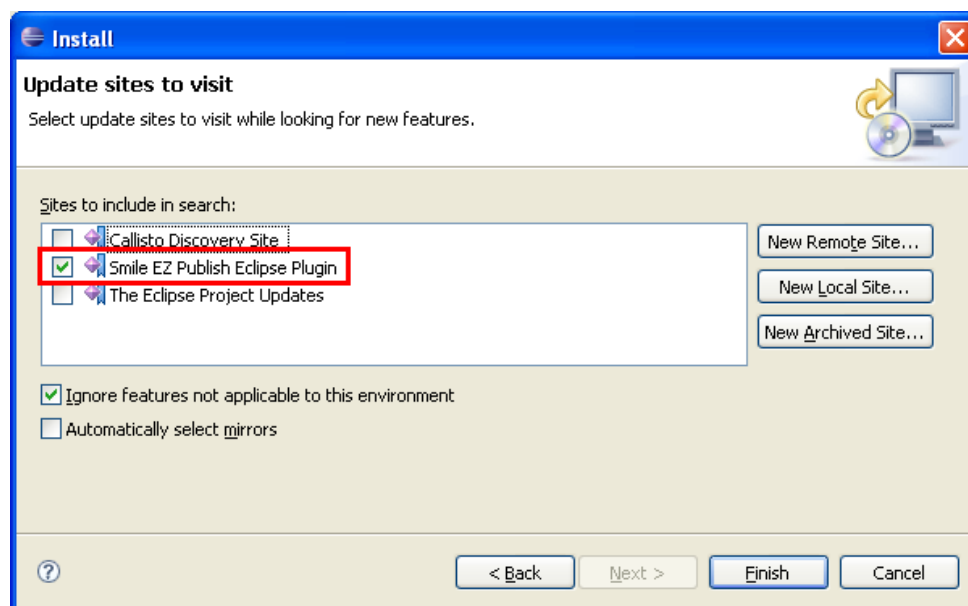
Select “search for new features to install” and then click the “next” button. The following page will be loaded.



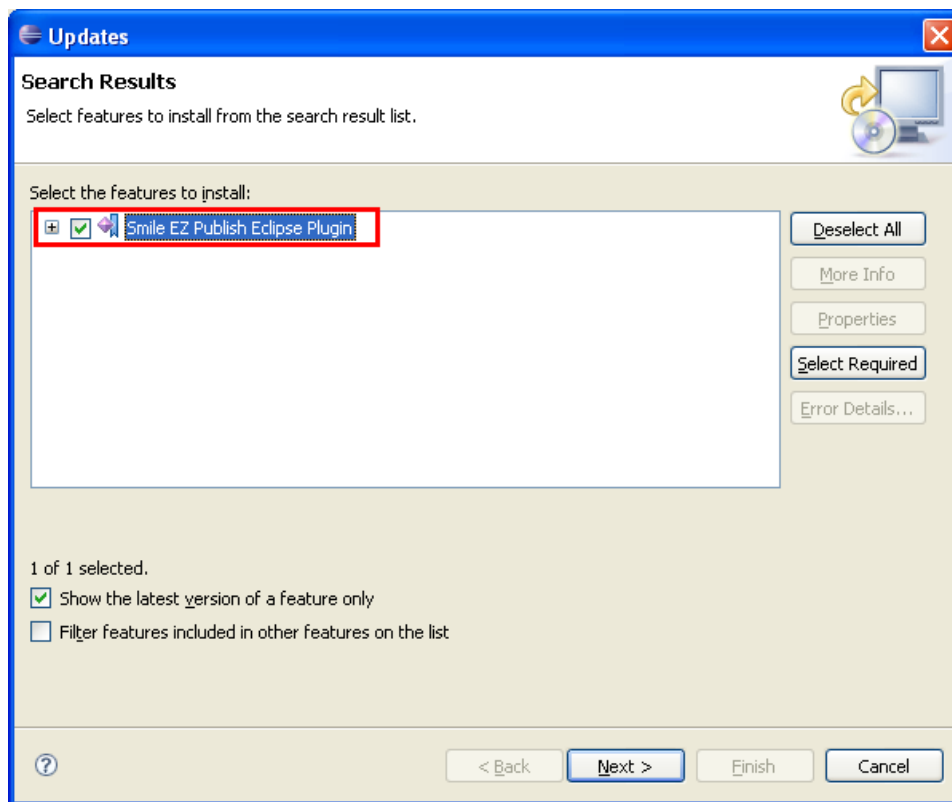
The following window will be displayed ... add the information shown below and click “ok”. (The actual name used does not matter, but is required by Eclipse).



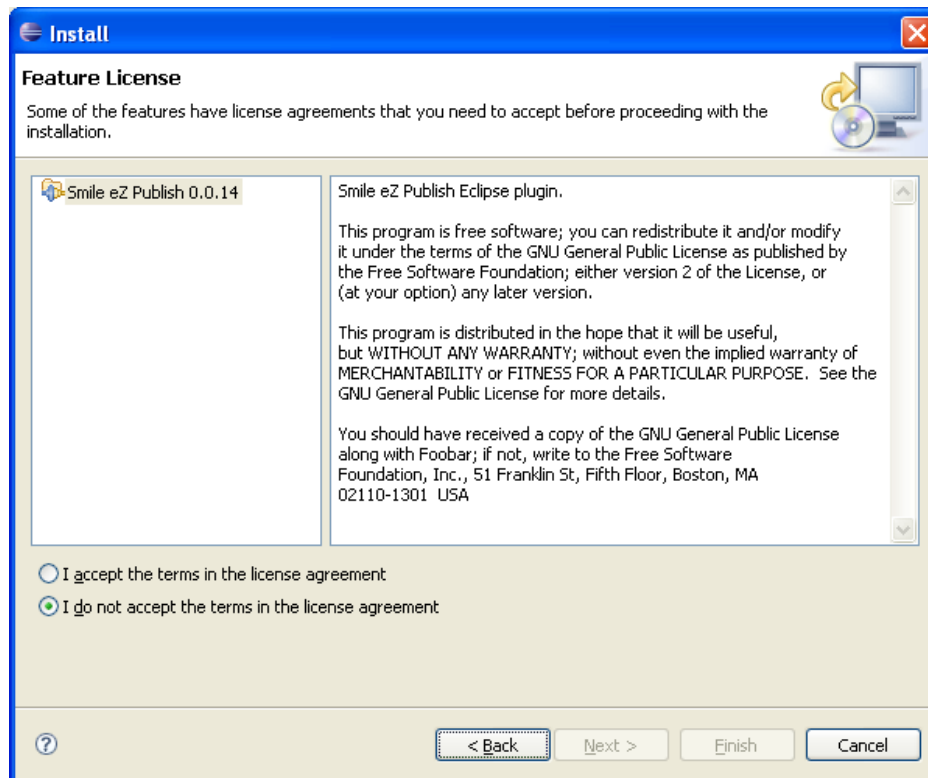
The Smile EZ Publish Eclipse Plugin will now be listed in the update sites to visit, as shown below.



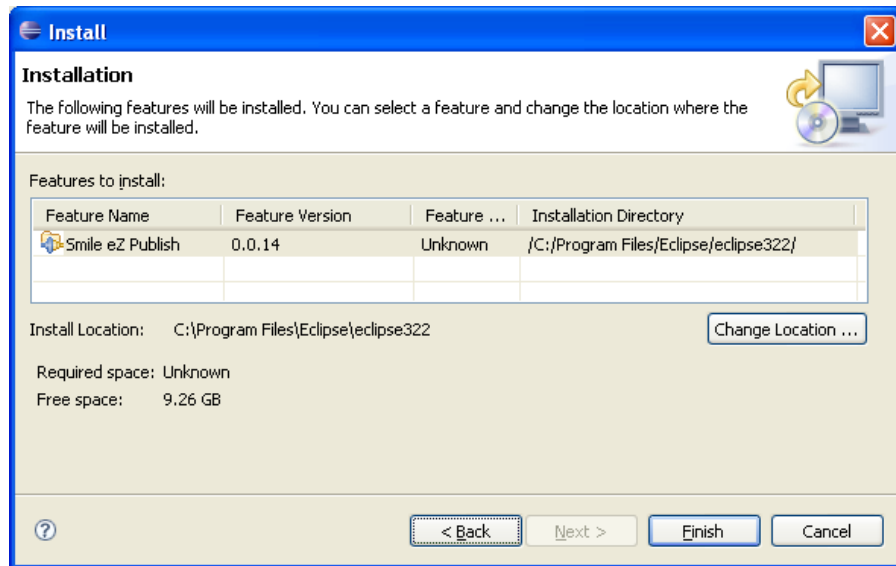
Click on Finish. Eclipse will then immediately search for updates, and will then open the following window.



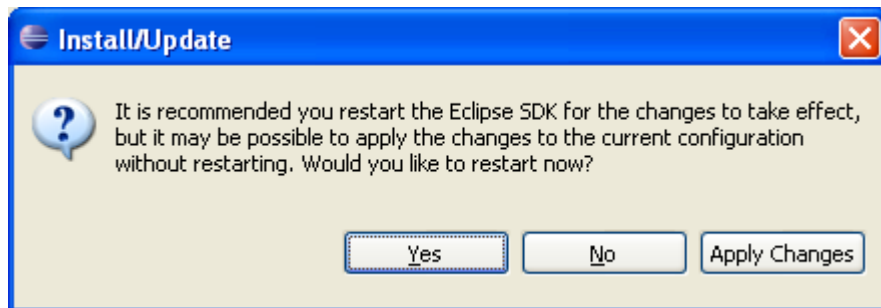
Click on the check-box adjacent to the Smile EZ Publish Eclipse Plugin and then click the “Next” button. The following page will be displayed.



If you have read the licence agreement and choose to accept it, click the option for this, and then click the “Next” button. The following page will be displayed.



If this information is correct, then click “Finish” to complete the installation. (At this stage Eclipse may show further information about the plugin before proceeding with the installation). Eclipse will then install the plugin and seek to restart.



3.4 Adding the Smile Plugin to Eclipse (Manual Method)

[This step is not required if you have already successfully completed the steps explained in section 2.5, above]

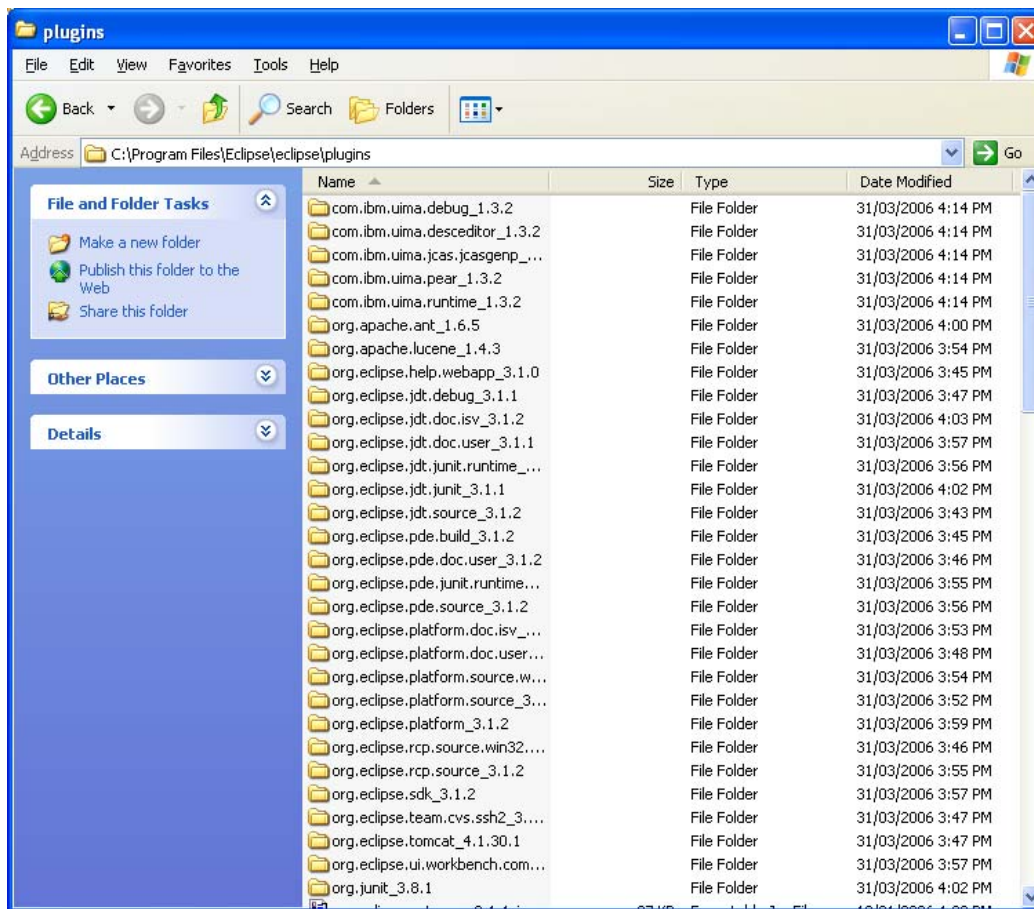
The Smile plugin includes an EZ Publish plugins folder.



The files in this folder are:

Name	Size	Type	Date Modified
org.apache.xerces_2.8.0.v200606131651		File Folder	13/03/2007 9:28 AM
fr.smile.ezpublish.content_1.0.4.jar	30 KB	Executable Jar File	8/03/2007 6:29 PM
fr.smile.ezpublish.core_0.0.8.jar	26 KB	Executable Jar File	8/03/2007 6:29 PM
fr.smile.ezpublish.extension_1.0.4.jar	44 KB	Executable Jar File	8/03/2007 6:29 PM

Copy these to the “plugins” folder in Eclipse



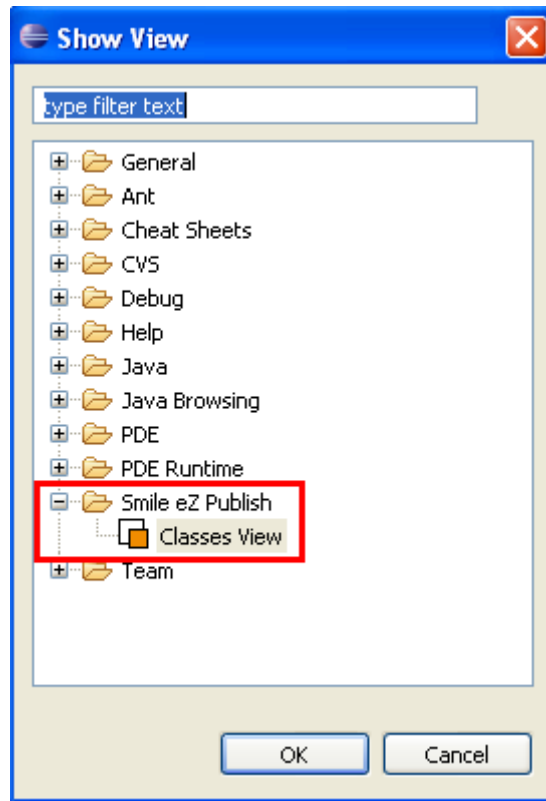
You should close and restart Eclipse at this point.

3.5 Loading EZ Publish Content Classes into Eclipse

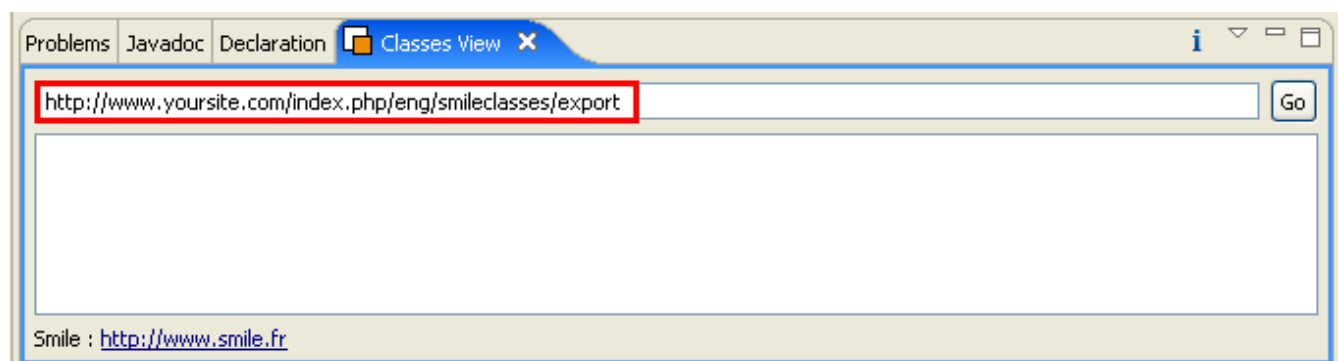
In Eclipse, from the top menu, select:

Window > Show View > Other

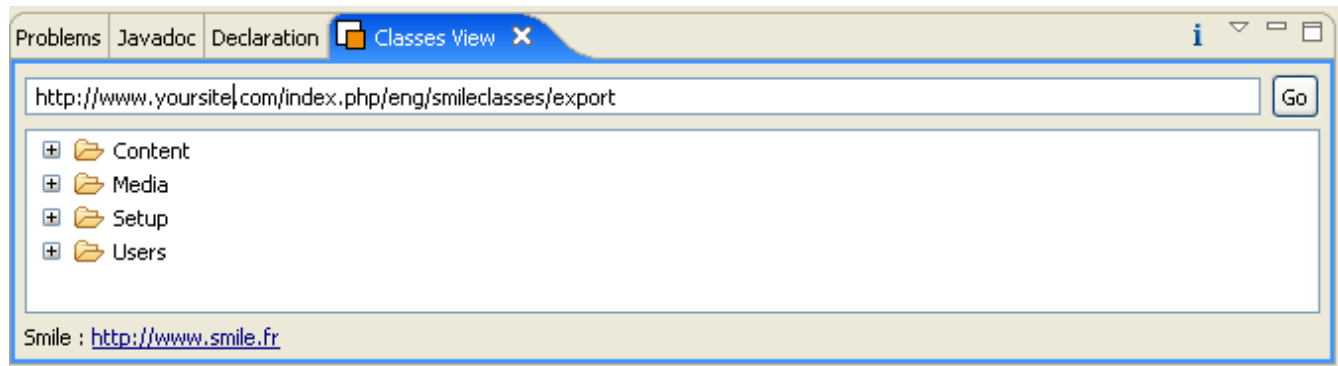
This will bring up the following window, which includes Smile eZ Publish and a sub-item “Classes View”.



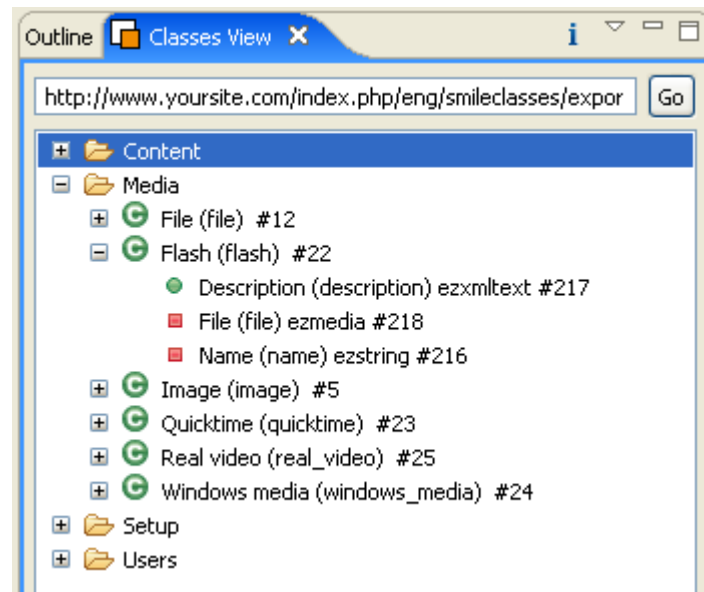
Select “Classes” View and then click “Ok”. This will open a “Classes View” window adjacent to the main working area of Eclipse. Enter your site’s URL and path to the smileclasses/export page in the top line of this view window.



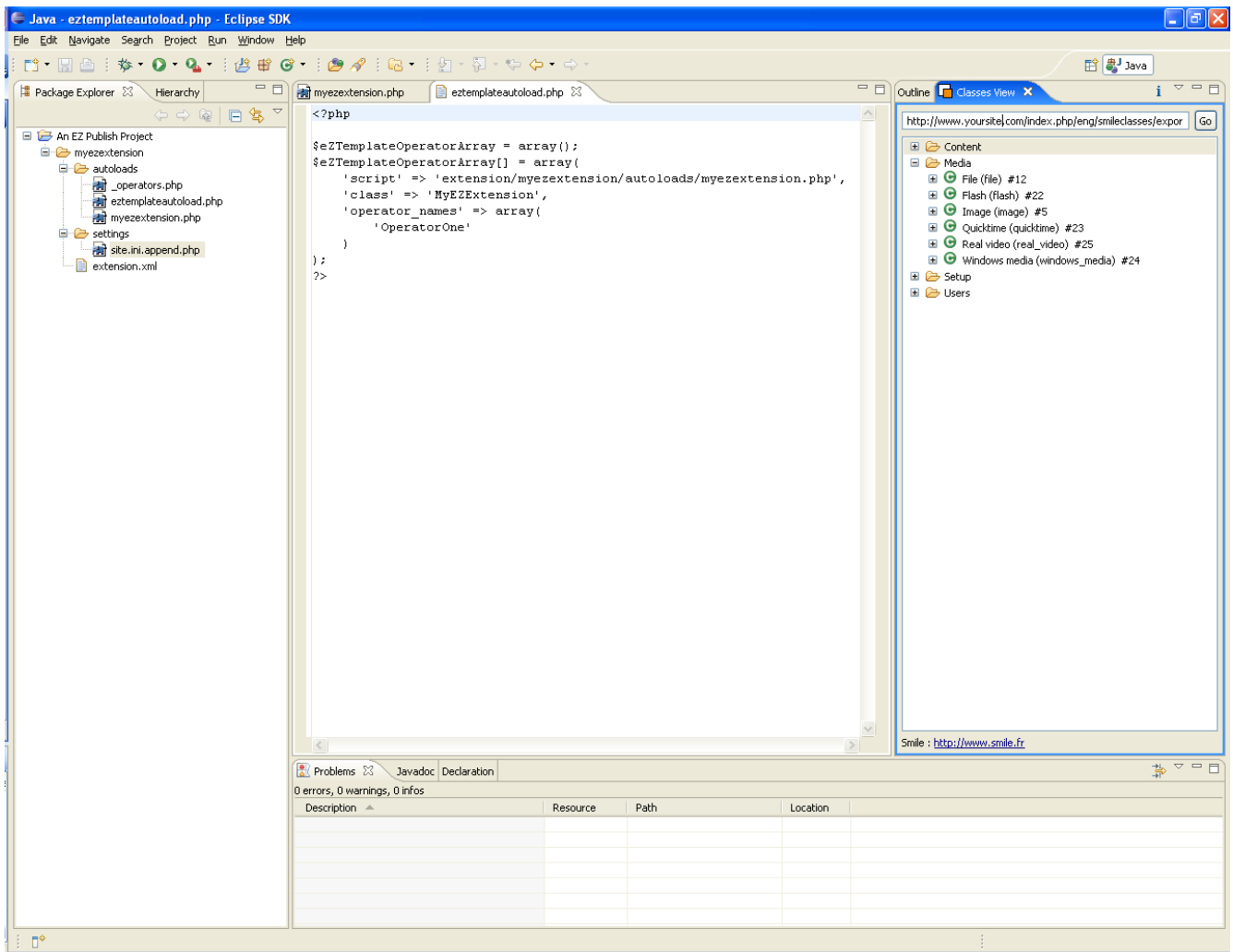
Click on the “Go” button and the following will be displayed. It has been loaded from your site.



Different content sections and classes can now be browsed from within Eclipse.



This window can then be used for reference purposes while working with EZ Templates and code.



4 Smile EZ Publish Eclipse Plugin - What it Does / a Brief How-to

The Smile Plugin does three main things:

- Template Syntax Highlighting
- Code Generation for Operator Extensions
- Browser for your site's EZ Publish Content Classes

These are explained further below.

4.1.1 Template Syntax Highlighting

EZ Publish template files loaded into Eclipse show syntax highlighting. For example:



```
{* DO NOT EDIT THIS FILE! Use an override template instead. *}
{*<form method="post" action="{ "content/action/"|ezurl}>*)

<div class="maincontentheader">
<h1>{$object.name|wash}</h1>
</div>

<div class="block">
<label>{"Current version"|i18n("design/standard/content/view")}</label><div class="labelbreak"></div>
<p class="box">{$object.current_version}</p>
</div>

{section name=ContentObjectAttribute loop=$object.contentobject_attributes}
<div class="block">
<label>{$ContentObjectAttribute:item.contentclass_attribute.name|wash}</label><div class="labelbreak"></div>
{attribute_view_gui attribute=$ContentObjectAttribute:item}
</div>
{/section}
```

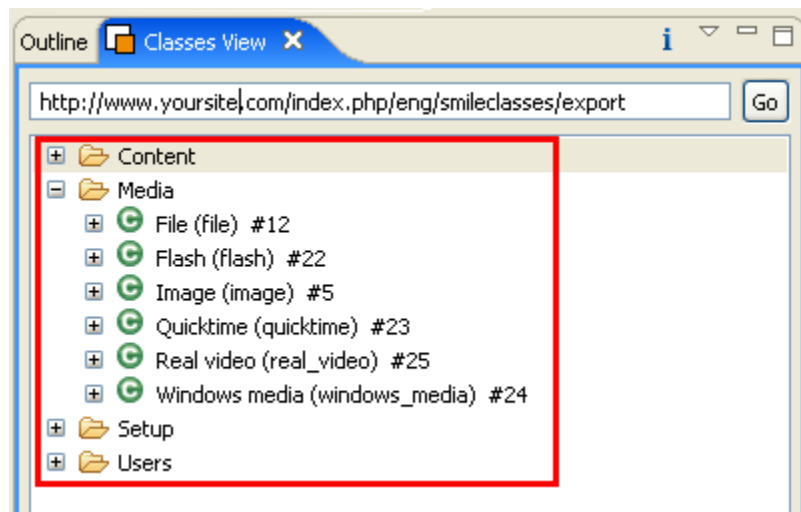
4.1.2 Content Class Browser

The plugin includes an EZ Publish extension called smileclasses. This generates an XML page that includes EZ Publish class information. For example:

```
Address http://www.yoursite.com/index.php/smileclasses/export
Google G Go No popups Check AutoLink AutoFill Send to

<?xml version="1.0" encoding="UTF-8" ?>
- <contentclasslist>
+ <group id="1" identifier="Content">
+ <group id="2" identifier="Users">
+ <group id="3" identifier="Media">
- <group id="4" identifier="Setup">
  - <class id="14" identifier="common_ini_settings" name="Common ini settings">
    <attribute id="159" identifier="name" name="Name" type="ezstring" is_required="0" />
    <attribute id="160" identifier="indexpage" name="Index Page" type="ezinisetting" is_required="0" />
    <attribute id="161" identifier="defaultpage" name="Default Page" type="ezinisetting" is_required="0" />
    <attribute id="162" identifier="debugoutput" name="Debug Output" type="ezinisetting" is_required="0" />
    <attribute id="163" identifier="debugbyip" name="Debug By IP" type="ezinisetting" is_required="0" />
    <attribute id="164" identifier="debugiplist" name="Debug IP List" type="ezinisetting" is_required="0" />
    <attribute id="165" identifier="debugredirection" name="Debug Redirection" type="ezinisetting" is_required="0" />
    <attribute id="166" identifier="viewcaching" name="View Caching" type="ezinisetting" is_required="0" />
    <attribute id="167" identifier="templatecache" name="Template Cache" type="ezinisetting" is_required="0" />
    <attribute id="168" identifier="templatecompile" name="Template Compile" type="ezinisetting" is_required="0" />
    <attribute id="169" identifier="imagesmall" name="Image Small Size" type="ezinisetting" is_required="0" />
    <attribute id="170" identifier="imagemedium" name="Image Medium Size" type="ezinisetting" is_required="0" />
    <attribute id="171" identifier="imagelarge" name="Image Large Size" type="ezinisetting" is_required="0" />
  </class>
</group>
</contentclasslist>
```

This XML page is used within Eclipse to provide a browser for your site's EZ Publish content classes. For example:



4.1.3 Code generation for operator extensions

The following material adapted from a quick tutorial on the operator extension **provided by Nicolas Richeton** (plugin author/contributor). His original notes and comments can be found here: http://ez.no/community/contribs/documentation/installation_documentation_smile_ez_publish_eclipse_plugin)

Important Note

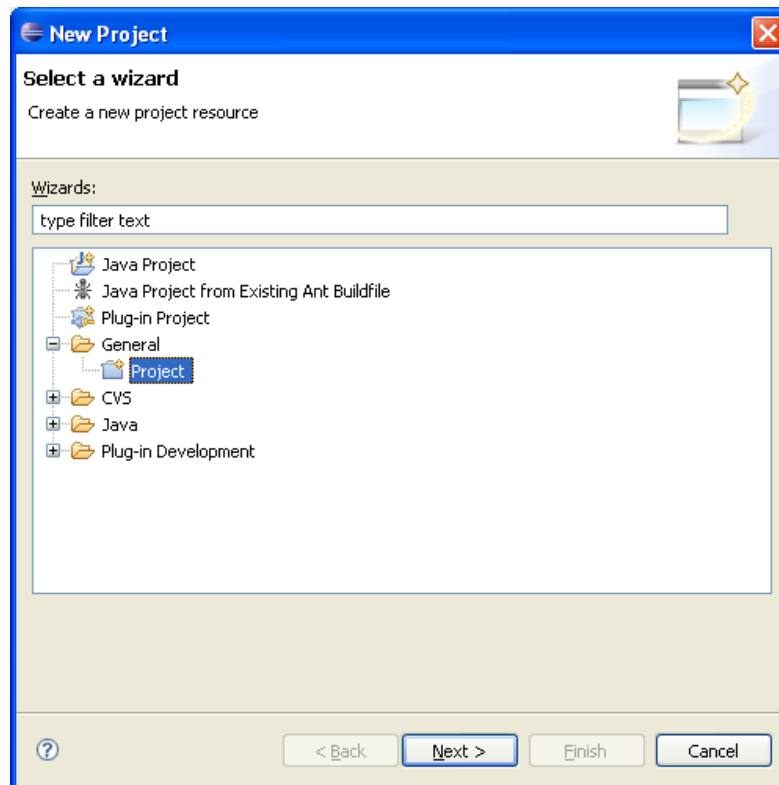
Be careful with this part, which can generate a new operator extension in a few steps. Some files will be automatically created and others **OVERWRITTEN** (see below). However, we included checks to prevent bad use and accidental loss of data.

Nicolas Richeton (source:

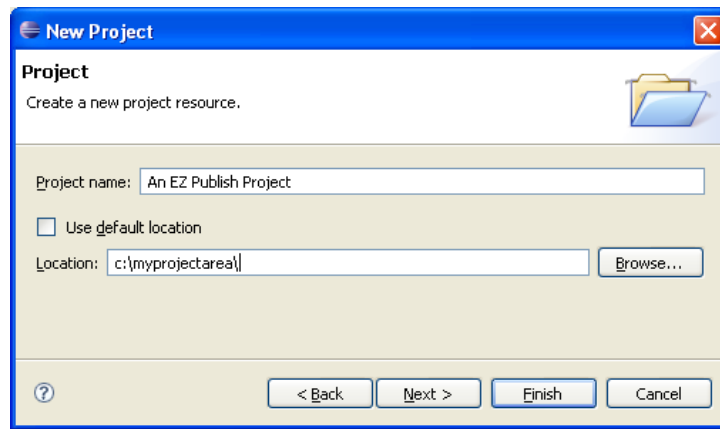
http://ez.no/community/contribs/documentation/installation_documentation_smile_ez_publish_eclipse_plugin)

1. *Creating a Project*

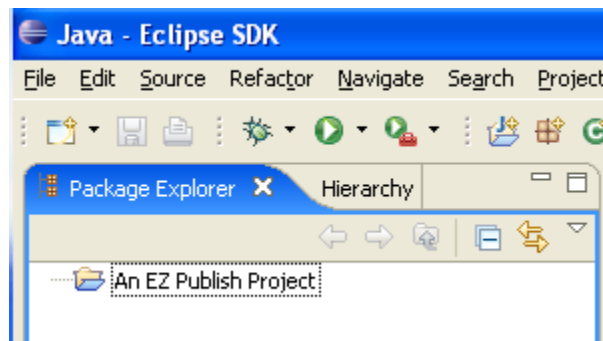
In Eclipse, from the top menu, select File > New > Project. The following window will be displayed.



There may be other ways to do this, but selecting General > Project seems appropriate. Click the Next button, and the following screen will appear.

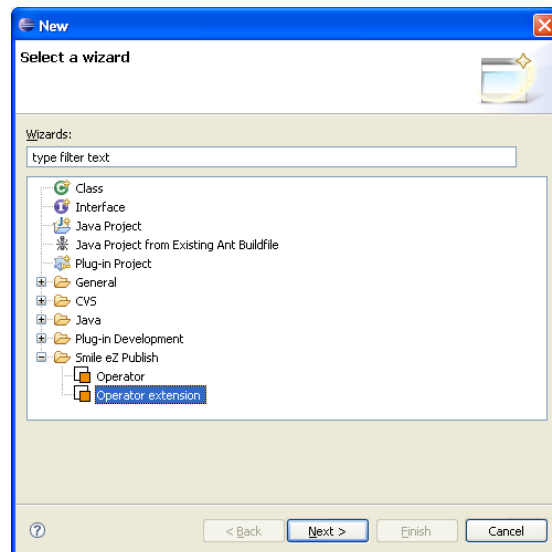


Fill in a name for the project and click Finish. The project will now be listed in the package explorer.

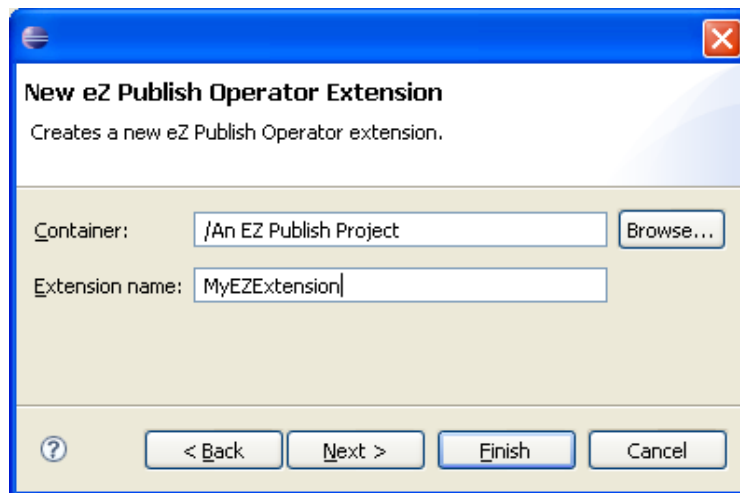


2. Creating an Extension

From main menu of Eclipse, select File > New > Other. The following window will then be displayed, showing wizards for creating EZ Publish code.

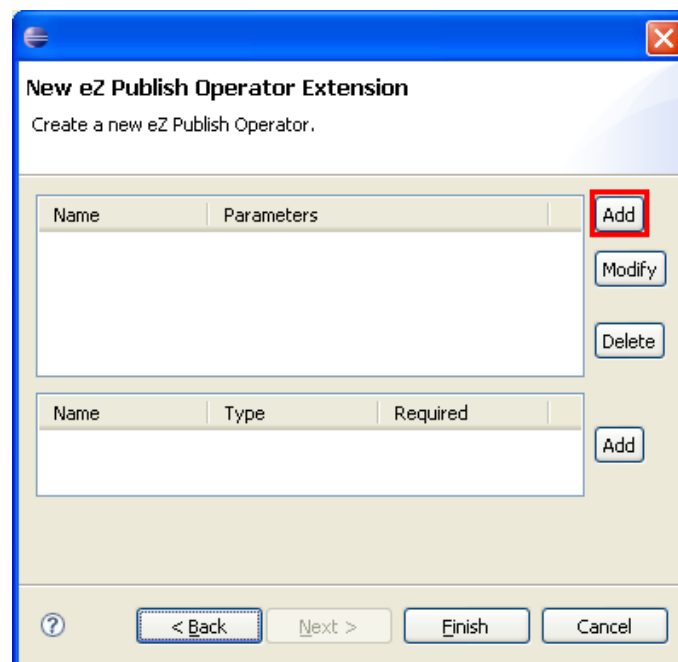


Select the (say) Operator Extension and click Next. The following window will be displayed,



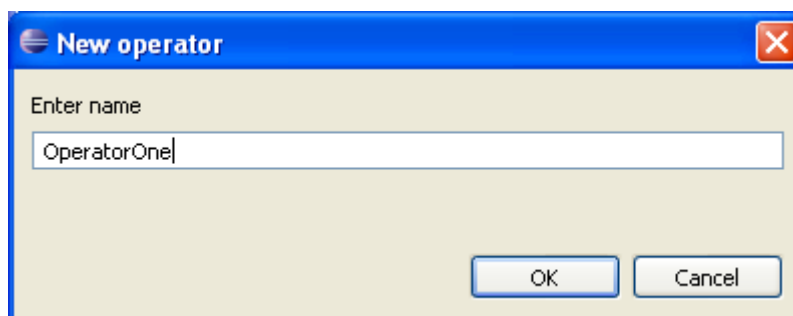
The dialog box is titled "New eZ Publish Operator Extension" and contains the text "Creates a new eZ Publish Operator extension." It has two input fields: "Container:" with the value "/An EZ Publish Project" and a "Browse..." button, and "Extension name:" with the value "MyEZExtension". At the bottom are buttons for "< Back", "Next >", "Finish", and "Cancel".

Choose a name for your extension and click on Next. The following window will be displayed.



The dialog box is titled "New eZ Publish Operator Extension" and contains the text "Create a new eZ Publish Operator." It features two tables. The first table has columns "Name" and "Parameters" and an "Add" button to its right. The second table has columns "Name", "Type", and "Required" and an "Add" button to its right. At the bottom are buttons for "?", "< Back", "Next >", "Finish", and "Cancel".

Click on the Add button to create any number of operators.



The dialog box is titled "New operator" and contains the text "Enter name". It has a single input field with the value "OperatorOne". At the bottom are buttons for "OK" and "Cancel".

These will appear in the Operator List

New eZ Publish Operator Extension
Create a new eZ Publish Operator.

Name	Parameters
OperatorOne	

Add
Modify
Delete

Name	Type	Required
------	------	----------

Add

? < Back Next > Finish Cancel

You can then add parameters for each operator by selecting an operator in the top area, and then clicking the “Add” button in the lower area of this window.

New eZ Publish Operator Extension
Create a new eZ Publish Operator.

Name	Parameters
OperatorOne	

Add
Modify
Delete

Name	Type	Required
------	------	----------

Add

? < Back Next > Finish Cancel

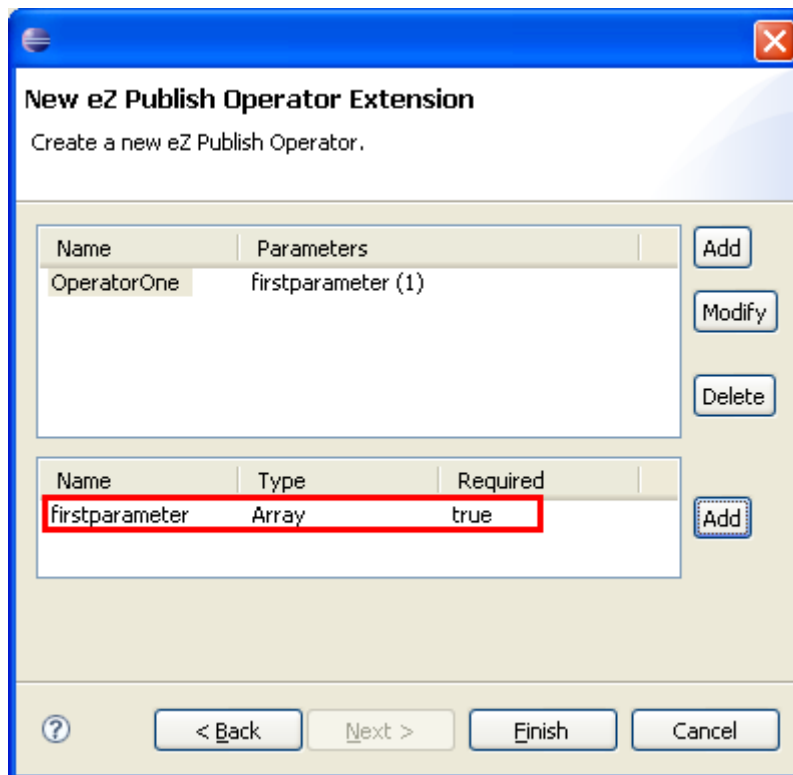
The following window will be displayed.



The 'Edit parameter' dialog box has a title bar with a close button. It contains three fields: 'Name' with the text 'parameterName', 'Type' with a dropdown menu showing 'String', and 'Required' with an unchecked checkbox. At the bottom are 'OK' and 'Cancel' buttons.

[The available types are String and Array]

Give the parameter a name, select the parameter type, and whether or not it is required. When OK is clicked, the parameter will now be added to the parameter list, as shown below.



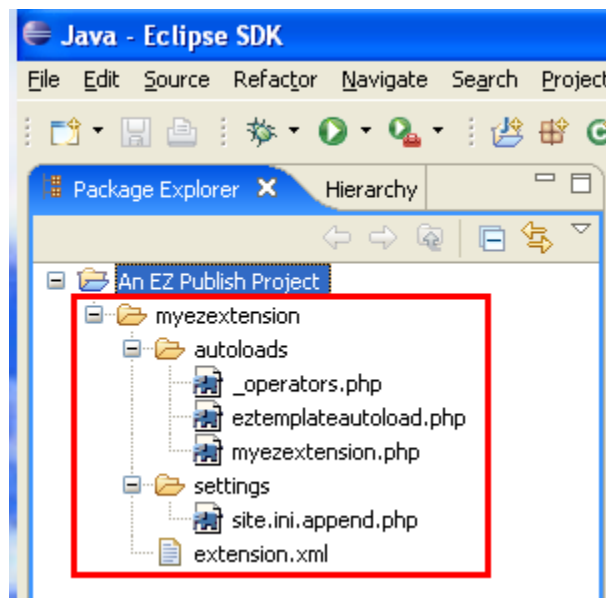
The 'New eZ Publish Operator Extension' dialog box has a title bar with a close button. Below the title is the instruction 'Create a new eZ Publish Operator.' It features two tables. The first table, 'OperatorOne', has columns 'Name' and 'Parameters' and contains one row with 'firstparameter (1)'. To its right are 'Add', 'Modify', and 'Delete' buttons. The second table has columns 'Name', 'Type', and 'Required' and contains one row with 'firstparameter', 'Array', and 'true'. This row is highlighted with a red border. To its right is an 'Add' button. At the bottom are buttons for '?', '< Back', 'Next >', 'Finish', and 'Cancel'.

Name	Parameters
OperatorOne	firstparameter (1)

Name	Type	Required
firstparameter	Array	true

Important Note: all files, except <yourextension>.php will be generated each time you click on finish. Don't modify these files manually. <yourextension>.php is generated only the first time, you have to update this file manually when you add/remove operators. (Nicolas Richeton)

When you click on Finish, extension directories and files will be automatically created, as shown below.



I have not got this next step worked out, but **Nicolas Richeton** seems to say that extension parameters may be updated/edited by a similar process; he says:

To add a new operator, select the extension and use New -> other -> new operator. The previous wizard will open directly on the operators page, and you will be able to add / update or delete operators.

(ref:

http://ez.no/community/contribs/documentation/installation_documentation_smile_ez_publish_eclipse_plugin)

If you are doing this, pay attention to Nicolas' note from the same source:

Important Notes

*Be careful with this part, which can generate a new operator extension in a few steps. Some files will be automatically created and others **OVERWRITTEN**. However, we included checks to prevent bad use and accidental loss of data.*

All files, except <yourextension>.php will be generated each time you click on finish. Don't modify these files manually. <yourextension>.php is generated only the first time, you have to update this file manually when you add/remove operators.

(Nicolas Richeton, see source referenced above)

[End of Document]