ENTERPRISE ARCHITECT

User Guide Series

HTML Reports

Author: Sparx Systems

Date: 2021-09-02

Version: 15.2

CREATED WITH ENTERPRISE ARCHITECT

Table of Contents

HTML Reports 3
- Create an HTML Report 4
- Create HTML Style Templates 7
  - HTML Template Fragments 9
- Model Documents and Report Packages 19
- Publish to Joomla! 20
  - Install Joomla! Locally 21
  - Set up a Joomla! ODBC Link 24
  - Publish as Joomla! Article 26
  - Joomla! Content Templates 29
  - Example - Package Published as Joomla! Article 31
  - Enable Search in Joomla! Website 33

HTML Reports

As you develop your model you can publish it on the web, where the outline structure closely mirrors the model hierarchy and makes it very simple to explore models on-line. With an elegant web design and the ability to explore very large models efficiently on-line, the web-publishing capability is a significant enhancement. You can export either an entire model or a single branch of the model to the web pages. The report provides an easy-to-use, highly detailed, JavaScript-based model tree. In addition, hyperlinked elements make it very simple to browse to related information.

Publishing to Joomla! is also supported, with a custom article generation capability.

Note that, while the generation of static pages through the HTML report generation capabilities is still supported, Sparx Systems recommends the use of PCS with WebEA and/or Prolaborate as the most efficient and effective means of publishing your model to a broader audience, using a local LAN or WAN or the Internet and a Browser based delivery.

Notes

- The HTML output is compatible with any standard web server, on either Unix or Windows platforms - simply bundle up the entire output directory and place it within the context of your web server; all path names should be relative and case sensitive
- You can create HTML Style Templates to customize your web output
- You can use Report Packages and Model Documents to select specific areas of your model for reporting
- In the Corporate, Unified and Ultimate Editions of Enterprise Architect, if security is enabled you must have 'Generate Documents' permission to generate HTML documents

Create an HTML Report

If you want to create a report on a Package to display as a web page, you can select the Package in the **Browser window** and perform a simple procedure using the 'Publish as HTML' dialog. All child Packages of the selected Package are automatically included in the report, but you can also select an option to remove specific child Packages if you prefer (see the *Exclude Package from Document and HTML Reports* Help topic).

Note that this method publishes a report via one of several types of Browser, and Browsers are marketable products that can change over time, such as the Browser no longer allowing access to files within the same directory. See the *Browser Behavior* section of this topic. As an alternative, you could consider making the live model information available via a web server, in WebEA or Prolaborate. See the *WebEA* and *Prolaborate* Help topics.

Access

Ribbon	Publish > Model Reports > HTML > Standard HTML Report or Specification-Specify > Element > Documentation > Publish as HTML
Browser Window Toolbar	| Documentation | Publish as HTML
Keyboard Shortcuts	**Shift+F8**

Create a Web Page Report

Field/Button	Action
Package	Confirm the name of the selected Package, for which you are creating the web documentation. If you want to change the Package, close the dialog, select the new Package and select the 'HTML Report' option again.
Title	Defaults to the Package name. If you want the report to have a different title, overtype the field contents with your preferred text.
Output to	Type in or browse for the directory path your report is to be saved to.
Style	(Optional) Click on the **drop-down arrow** and select a custom web style template to apply to your documentation, in place of the default template. Use the <default> value for the standard, system-provided template.
File extension	Type the file extension for your web documentation files; the default is .htm.
Theme	Click on the **drop-down arrow** and select a Diagram Theme to apply the color, line thickness and font style to diagrams of elements in the report, if those elements are using default values. If you set the field to '<none>' the report will apply the system default theme defined in the 'Preferences' dialog, 'Diagram > Themes' page.

Header Image	If you want the report to have an identifying image at the top, type in or browse for the image file directory path and filename. If you do not specify a path, the image defaults to the Enterprise Architect logo.
Preserve White space in Notes	Select the checkbox to reproduce the spacing of any element notes. Deselect the checkbox to remove white space.
No Page for Note and Text Items	Select the text box to exclude Note elements and Text elements from the report.
Hide Stereotype in Browser Window	Select this checkbox to turn off the inclusion of element stereotypes in element names in the Table of Contents (the **Browser window**) of the generated HTML report. Deselect the checkbox to show the stereotypes.
External Hyperlink target	Provides options for how hyperlinked external documents are opened and displayed to the user. Click on the **drop-down arrow** and select the appropriate option: • '_top' - open the Linked Document in the full body of the current window (the default) • '_blank' - open the Linked Document in a new window or tab • '_self' - open the Linked Document within the same frame as the clicked link
Image Format	Select the appropriate file format in which to store images - PNG or GIF.
Include	Select the appropriate checkboxes to include, in your web document: • Maintenance Items • Resource allocations • Hyperlinked files • Non-printable elements • Test Cases • Glossary • Model Tasks • Model Issues
Default Diagram	Select the appropriate radio button to begin the web report with: • The default Model Diagram • The currently-selected diagram • Another diagram, for which you can browse by clicking on the button • No diagrams (the 'None' radio button defaults to selected)
Generate	Click on this button to generate the report.
Progress	Shows the accumulating percentage completion of the report generation process.
View	Once the report is complete, click on this button to launch your default web browser and view the web pages.

View page for specific diagram or element

When you view the web report in your web browser, you can switch directly to a page for a specific diagram or element by specifying the appropriate GUID after the report web address. That is:

http://path/path/path/Index.htm?guid=xxxxxxxxxxxx

Type the word guid in lower case, and do not include braces { } in the value; for example:

http:// .../path/Index.htm?guid=DC62B0DA-0D60-4447-85E6-B9BBAE7FC90F

To obtain the GUID, right-click on the Package or element in the **Browser window** and select the 'Copy/Paste' option, or right-click on the diagram and select the 'Copy Reference' option.

Browser Behavior

The display of your web report can be affected by the browser that you use. Specifically:

- Firefox automatically converts the diagram or element page locator path to a file:///C:/path protocol and actions it
- Opera automatically adds file://localhost/ to the start of the page locator path, in the same way as Firefox adds the file:///C:/path protocol
- Opera by default disables XMLHttpRequest when loading a file, and the web report detects this and prompts the user to change settings; there are no problems when the page is hosted on a web server
- Microsoft Internet Explorer does not operate on the page locator path directly, but the protocol in Firefox can also be applied in Internet Explorer; therefore, to use the absolute references without a web server, you must access the path using the file:/// protocol
- If you are using Internet Explorer and you do not have it open, its security profile might block the report display; click on the explanation banner at the top of the screen and select the 'Allow Blocked Content' option
- To display web reports directly from the file system, Chrome and Chromium-based Edge require the command line argument *chrome.exe --allow-file-access-from-files*; there are no problems with direct access when the page is hosted on a webserver

Notes

- To generate a demonstration web report right now, open the EAExample project, select the 'System' Model Package and work through the previous steps
- You can also select Packages in your model to exclude them (and their subordinate Packages) from the web reports you generate

Create HTML Style Templates

If you want to customize the look and feel of an HTML report for your company or client, you can use the **HTML and CSS Style Editor** to change the HTML associated with the various sections of the report. This editor is derived from, and provides the facilities of, the common Code Editor. The HTML and CSS Style Editor contains a set of HTML fragments for modification and customization, listed in the left hand 'Templates' panel of the dialog.

Each fragment typically contains HTML plus one or more special tag names that are replaced with information during generation. The special tag names are delimited by two # characters; for example, #NOTES#, as you can see if you click on a template fragment name to display its contents in the right hand 'Current Modified Template' panel. The HTML report generator produces output based on the fragments, either in their default state or as you have customized them. Currently you cannot alter the content within the tag names, but you can omit a complete tag by removing it, or alter its basic display properties in the surrounding HTML.

Access

Ribbon	Configure > Reference Data > Resources > HTML Publishing

Create or edit HTML style templates

Task	Action
Create a new template	Right-click on the *HTML Publishing* folder and select the 'Create HTML Template' option. Type the name of the new template when prompted to do so. The '**HTML and CSS Style Editor**' displays.
Edit an existing template	Expand the *HTML Publishing* folder and either: • Double-click on the template name or • Right-click on the template name and select the 'Modify HTML Style Template' option The **HTML and CSS Style Editor** displays.
Delete a customized template	Expand the *HTML Publishing* folder, right-click on the template name and select the 'Delete HTML Template' option.

Within the HTML and CSS Style Editor

Task	Action
Edit a template fragment	Click on the required fragment in the 'Templates' list; the fields in that fragment display in the 'Current Modified Template' panel. Change the field values, or delete fields, as necessary.

Save your style changes	Click on the **Save button** to save your version of this template fragment only. Save each fragment before you move on to another fragment.
Delete a modified template fragment	Click on the template fragment in the left-hand list and click on the **Delete button**. This removes the modified version of the fragment, which causes the report generator to use the default (system-provided) fragment during report generation.
Retrieve the original, default template fragment	Click on the template fragment in the left hand list and click on the **Get Default button**.

Notes

- Each time you generate the HTML report it overwrites the templated files, so you must back up your modified versions and copy them back in after every update

HTML Template Fragments

When you create an HTML report template in the **HTML and CSS Style Editor**, it automatically contains a wide range of system-provided HTML Template fragments. You can edit the fields of each separate fragment, or use the unchanged default.

Access

Ribbon	Configure > Reference Data > Resources > HTML Publishing > right-click > Create HTML Template, or Configure > Reference Data > Resources > HTML Publishing > <template name> > right-click > Modify HTML Style Template

Fragment Fields

Fragment	Fields
Body - Diagram	• #AUTHOR# - Diagram author • #CREATEDATE# - Diagram created date • #CSS# - Style Sheet to use • #GUID# - Diagram GUID • #IMAGE# - Image of the diagram • #IMAGES# - Image path • #LOCKED# - Is the diagram locked (true or false) • #MODDATE# - Diagram modified date • #NAME# - Diagram name • #NOTE# - Diagram notes • #STEREOTYPE# - Diagram stereotype • #TITLE# - Diagram title • #TYPE# - Diagram type • #VERSION# - Diagram version
Body – Object	• #ABSTRACT# - Element abstract (if true, = abstract) • #ACTIVE# - Element isActive (true or false) • #ALIAS# - Element alias ("#ALIAS#") • #AUTHOR# - Element author • #BEHAVIOR# - Returns the object behavior. • #CLASSIFIER# - Element classifier • #CLASSIFIERREF# - Returns the HREF of the classifier • #COMPLEXITY# - Element complexity • #CREATEDATE# - Element created date • #CSS# - Style Sheet to use

	• #DIAGRAMS# - List of diagrams the element is on • #DIFF# - Element difficulty • #GUID# - Element GUID • #IMAGES# - Image path • #KEYWORDS# - Element keywords • #LANGUAGE# - Element language • #LEAF# - Element isLeaf (true or false) • #LEVELNUMBER# - Element level number • #MODDATE# - Element modified date • #MULTIPLICITY# - Element multiplicity ("Multiplicity: #MULTIPLICITY#") • #NAME# - Element name • #NOTE# - Element notes • #PHASE# - Element phase • #PRIORITY# - Element priority • #ROOT# - Element isRoot (true or false) • #SCOPE# - Element scope • #STATUS# - Element status • #STEREOTYPE# - Element stereotype(s) • #TYPE# - Element type; for example, Class, Object • #VERSION# - Element version
Body – Project Glossary	• #CONTENT# - Loops Body – **Project Glossary** Item for each Project Glossary item
Body - Project Glossary Item	• #MEANING# - Glossary Meaning • #TERM# - Glossary Term • #TYPE# - Glossary Type
Body – Project Issue	• #CONTENT# - Loops Body – Project Issue Item for each Project Issue item
Body - Project Issue Item	• #DATERESOLVED# - Project Issue resolved date (blank if no date entered) • #ISSUE# - Project Issue name • #ISSUEDATE# - Project Issue issue date • #NOTES# - Project Issue notes • #OWNER# - Project Issue owner • #RESOLUTION# - Project Issue resolution • #RESOLVER# - Project Issue resolver • #STATUS# - Project Issue status
Body – Project Task	• #CONTENT# - Loops Body – Project Task Item for each Project Task item
Body - Project Task Item	• #ENDDATE# - Project Task end date • #NAME# - Project Task name • #NOTES# - Project Task notes • #OWNER# - Project Task owner • #PHASE# - Project Task phase

	• #PRIORITY# - Project Task priority • #STARTDATE# - Project Task start date • #STATUS# - Project Task status • #TYPE# - Project Task type
Content – Attributes	• #CONTENT# - Loops Content – Attributes Item for each attribute on an element
Content – Attributes Item	• #ALIAS# - Attribute alias ("Alias: #ALIAS# ") • #ALLOWDUPLICATES# - For Non-Table elements, returns **True** or **False** • #COLLECTION# - For Non-Table elements, returns True or False • #CONST# - Attribute is constant value ("const " <-- Note Space) • #CONSTRAINT# - Attribute constraint • #DEFAULT# - Attribute default value ("Initial: #DEFAULT ") • #DERIVED# - Attribute is Derived • #FK# - Foreign Key, for Table elements returns True or False • #ISID# - Attribute is ID • #ISORDERED# - For Non-Table elements, returns True or False • #NAME# - Attribute name • #NOTE# - Attribute notes • #NOTNULL# - For Table elements, returns True or False • #ORDERED# - Attribute Is ordered value ("Ordered ") • #PK# - Primary Key, for Table elements returns True or False • #RANGE# - If lower != 1 ("Range:<lower> to <upper>") • #SCOPE# - Attribute scope ("# SCOPE# " <-- Note space) • #STATIC# - Attribute is static value (“static ” <-- Note Space) • #STEREOTYPE# - Attribute stereotype(s) • #TAGS# - Attribute tags ("Property Name=Property Value ") • #TRANSIENT# - Attribute is Transient • #TYPE# - Attribute type (if Column, “#TYPE(Column Precision, Scale)") • #UNIQUE# - For Table elements, returns True or False
Content - Checklist	#CONTENT# - List of Content - Checklist Item
Content - Checklist Item	• #DESCRIPTION# - CheckList Description • #CHECKED# - **True**/**False** value if the item is checked • #CHECKEDXD# - x/- value if the item is checked
Content – Constraints	• #CONTENT# - Loops Content – Constraints Item for each constraint on an element
Content – Constraints Item	• #CONSTRAINT# - Constraint name • #NOTES# - Constraint notes • #STATUS# - Constraint status • #TYPE# - Constraint type
Content – Custom	• #CONTENT# - List of all Content – Custom Properties - Item

Properties	
Content – Custom Properties - Item	• #PROPERTY# - Custom property name • #VALUE# - Custom property value (if type equals Boolean true or false else #VALUE#)
Content - Element Effort	• #CONTENT# - List of Content - Element Effort Item
Content - Element Effort Item	• #EFFORT# - Effort name • #TYPE# - Effort Type • #TIME# - Effort Time • #NOTES# - Effort Notes
Content - Element Risk	• #CONTENT# - List of Content - Element Risk Item
Content - Element Risk Item	• #RISK# - Risk name • #TYPE# - Risk Type • #WEIGHT# - Risk Weight • #NOTES# - Risk Notes
Content - Element Metric	• #CONTENT# - List of Content - Element Metric Item
Content - Element Metric Item	• #METRIC# - Metric name • #TYPE# - Metric type • #WEIGHT# - Metric weight • #NOTES# - Metric notes
Content – Files	• #CONTENT# - List of Content - Files Item
Content – Files Item	• #DATE# - File date • #FILENAME# - The name of the file. • #LINKPATH# - Identifies the file path; if the 'Hyperlinked Files' checkbox is selected, this path is shown as the file path relative to the generated HTML directory; if the checkbox is not selected, the path is shown as the Absolute file path to the source directory for the file • #NAME# - Shows the Absolute source directory path and file name of the linked file; if the 'Hyperlinked Files' checkbox is selected, the hyperlink is actually to the file path relative to the generated HTML directory; if the checkbox is not selected, the link is actually to the Absolute file path; you cannot change this tag • #NOTES# - File notes • #SIZE# - File size • #TYPE# - File type The #FILENAME# and #LINKPATH# tags can be used together to create custom hyperlink target types. For example: #FILENAME#

Content - Inherited Attributes	• #CONTENT# - Loops Content – Inherited Attributes Item for each attribute on an element
Content - Inherited Attributes Item	• #ALIAS# - Attribute alias ("Alias: #ALIAS# ") • #CONST# - Attribute is constant value ("const " <-- Note Space) • #CONSTRAINT# - Attribute constraint • #DEFAULT# - Attribute default value ("Initial: #DEFAULT ") • #NAME# - Attribute name • #NOTE# - Attribute notes • #ORDERED# - Attribute Is ordered value ("Ordered ") • #RANGE# - If lower != 1 ("Range:<lower> to <upper>") • #SCOPE# - Attribute scope ("#SCOPE# " <-- Note space) • #STATIC# - Attribute is static value (“static ” <-- Note Space) • #STEREOTYPE# - Attribute stereotype(s) • #TAGS# - Attribute tags ("Property Name=Property Value ") • #TYPE# - Attribute type (if Column, “#TYPE(Column Precision, Scale)")
Content - Inherited Operations	• #CONTENT# - List of Content - Inherited Operations Item
Content - Inherited Operations Item	• #ABSTRACT# - Operation abstract (abstract) • #ALIAS# - Operation alias (“Alias: #ALIAS# ”) • #CONCURRENCY# - Operation concurrency (blank if not set) • #CONST# - Operation constant (const) • #CONSTRAINTS# - List of Method Constraint • #ISQUERY# - Operation IsQuery (isQuery) • #NAME# - Operation name • #NOTE# - Operation notes • #PARAMS# - List of Content – Operation Item Parameters • #SCOPE# - Operation Scope • #STATIC# - Operation IsStatic (static) • #STEREOTYPE# - Operation stereotype • #TAGLABEL# - Static text (Tags) • #TAGS# - Attribute tags ("Property Name=Property Value ") • #TYPE# - Operation type
Content – Notes	• #VALUE#> - Notes text
Content – Operations	• #CONTENT# - List of Content - Operations Item
Content – Operations Item	• #ABSTRACT# - Operation abstract (abstract) • #ALIAS# - Operation alias (“Alias: #ALIAS# ”) • #CONCURRENCY# - Operation concurrency (blank if not set) • #CONST# - Operation constant (const) • #CONSTRAINTS# - List of Method Constraint • #ISQUERY# - Operation IsQuery (isQuery)

	• #NAME# - Operation name • #NOTE# - Operation notes • #PARAMS# - List of Content – Operation Item Parameters • #SCOPE# - Operation Scope • #STATIC# - Operation IsStatic (static) • #STEREOTYPE# - Operation stereotype • #TAGLABEL# - Static text (Tags) • #TAGS# - Attribute tags ("Property Name=Property Value ") • #TYPE# - Operation type
Content – Operations Item Parameters	• #DEFAULT# - Op Parameter default • #GUID# - Op Parameter GUID • #KIND# - Op Parameter kind • #NAME# - Op Parameter name • #NOTES# - Op Parameter notes • #STEREOTYPE# - Op Parameter stereotype • #TYPE# - Op Parameter type
Content – Resource Allocation	• #CONTENT# - List of Content – Resource Allocation Item
Content – Resource Allocation Item	• #ACTUAL# - Resource actual time • #ENDDATE# - Resource end date • #EXPECTED# - Resource expected date • #NOTES# - Resource notes • #PERCENT# - Resource percent complete • #RESOURCE# - Resource name • #ROLE# - Resource role • #STARTDATE# - Resource start date • #TIME# - Resource time
CSS – Main	• None
CSS – ToC	• None
Feature Notes	• #FIELD# • #VALUE#
JavaScript	• None
Link (Association)	• #CONTENT#
Link (Association) Item	• #CONNECTION# - Connector type • #DIRECTION# - Connector direction • #ELEMNAME# - Name of the element at the source/destination of the connector • #IMAGE# - The file path of the images

	• #LINK# - ("Connection Name") • #LINKREF# - Page name of the element at the source/destination of the connector • #NAME# - Connector Name • #NOTES# - The connector notes • #NUMBER# - A unique number used to identify div elements • #SOURCEROLE# - Source role • #SOURCEROLENOTES# - Source role notes • #STEREOTYPE# - Connector stereotype • #TARGETROLE# - Target role • #TARGETROLENOTES# Target role notes • #TYPE# - Connector type
Link (Flow)	• #CONTENT#
Link (Flow) Item	• #DIRECTION# - Connector direction • #ELEMNAME# - Name of the element at the source/destination of the connector • #IMAGE# - The file path of the images • #LINK# - ("Connection Name") • #LINKREF# - Page name of the element at the source/destination of the connector • #NAME# - Connector name • #NOTES# - The connector notes • #NUMBER# - A unique number used to identify div elements • #STEREOTYPE# - Connector stereotype • #TYPE# - Connector type
Link (Other)	• #CONTENT#
Link (Other) Item	• #CONNECTION# - Connector type • #ELEMNAME# - Name of the element at the source/destination of the connector • #IMAGE# - The file path of the images • #LINK# - ("Connection Name") • #LINKREF# - Page name of the element at the source/destination of the connector • #NAME# - Connector name • #NOTES# - The connector notes • #NUMBER# - A unique number used to identify div elements • #SOURCEROLE# - Source role • #SOURCEROLENOTES# - Source role notes • #STEREOTYPE# - Connector stereotype • #TARGETROLE# - Target role • #TARGETROLENOTES# Target role notes • #TYPE# - Connector type

Linked Document	• #LINKDOC# - Linked Document.
Linked Requirement	• #CONTENT# - List of Linked Requirement Item
Linked Requirement Item	• #DIFF# - Linked Requirement difficulty • #NAME# - Linked Requirement name • #PRIORITY# - Linked Requirement priority • #STATUS# - Linked Requirement status
Linked Section	• #ITEMS# • #TITLE#
Maintenance	• #CONTENT# - List of Maintenance Line Item
Maintenance Line Item	• #DATEREPORTED# - Maintenance date reported • #DATERESOLVED# - Maintenance date resolved • #IMAGE# - The file path of the images • #NOTES# - Maintenance notes • #NUMBER# - A unique number used to identify div elements • #PRIORITY# - Maintenance priority • #PROBLEM# - Maintenance name • #REPORTEDBY# - Maintenance reported by • #RESOLVEDBY# - Maintenance resolved by • #RESOLVERNOTES# - Maintenance resolved notes • #STATUS# - Maintenance status • #TYPE# - Maintenance type • #VERSION# - Maintenance version
Message	(Applies only to Sequence messages.) • #CONTENT# - Loops the Message Item for each attribute on an element • #DIRECTION# - Contains the value To or From
Message Item	• #KIND# - The 'Message Kind' field • #MESSAGE# - Connector Message • #NAME# - Name of the Message (<a href="<path>">#NAME#) If Message has a classifier: #NAME#="#NAME# :Classifier" • #NOTES# - The Message notes (Type: #Item Type# #NOTES#) • #SYNCH# - The 'Message Synch' field • #TYPE# - The type of Message
Method Constraint	• #NAME# - Method Constraint name • #NOTES# - Method Constraint notes • #TYPE# - Method Constraint type
Object Requirement	• #CONTENT# - List of Object Requirement Item
Object Requirement Item	• #DIFF# - Requirement difficulty

	• #NAME# - Requirement name • #NOTES# - Requirement notes • #PRIORITY# - Requirement priority • #STABILITY# - Requirement stability • #STATUS# - Requirement status • #TYPE# - Requirement type
Package Content	• #CONTENT# - List of Package Content Row
Package Content Row	• #NAME# - Link to Package (#Package name#) • #TYPE# - Link to Image (<img src="#path to image#">)
Page - Basic template	• #CONTENT# - Contains Body - Diagram through to Body - Object • #TITLE# - Current Package name
Page - Index	• #CSS# - Style Sheet to use • #HOME# - A link to the Start page • #JS# - JavaScript to use • #TITLE# - Current Package name • #TOC# - Table of Contents
Page - ToC	• None
Scenario	• #CONTENT# - List of Scenario Item • #EXCEPTIONS# - List of Structured Scenario exceptions • #STRUCTURED# - List of Structured Scenarios
Scenario Item	• #IMAGE# - The file path of the images • #NOTES# - Scenario notes • #NUMBER# - A unique number used to identify div elements • #SCENARIO# - Scenario name • #TYPE# - Scenario type
Scenario Exception	• #CONTENT# - Loops Scenario Exception Item for each exception
Scenario Exception Item	• #NAME# - Exception name • #STEPNO# - Exception step number • #TYPE# - Exception Type
Scenario Structured	• #CONTENT# - Loops Scenario Structured Items for each Structured Scenario item
Scenario Structured Items	• #ACTION# - Name of the scenario • #STEPNO# - Scenario step number • #RESULT# - Step result value • #USES# - Step uses value • #STATE# - Step state value • #TYPE# - Step type (System or User)

Tagged Value	• #CONTENT# - List of Tagged Value Line Item
Tagged Value Line Item	• #IMAGE# - The file path of the images • #NOTES# - Tagged Value notes • #NUMBER# - A unique number used to identify div elements • #PROPERTY# - Tagged Value name • #VALUE# - Tagged Value if type is Boolean (value is true or false)
Test Cases	• #CONTENT# - List of Test Cases Line Item
Test Cases Line Item	• #ACCEPTANCE# - Test Case acceptance notes • #CHECKEDBY# - Test Case checked by • #CLASS# - Test Case Class (Unit, Integration, System, Acceptance, Scenario) • #IMAGE# - The file path of the images • #INPUT# - Test Case input notes • #NOTES# - Test Case notes • #NUMBER# - A unique number used to identify div elements • #RESULTS# - Test Case result notes • #RUNBY# - Test Case run by • #RUNDATE# - Test Case last run • #STATUS# - Test Case status • #TEST# - Test Case name • #TYPE# - Test Case type

Model Documents and Report Packages

You can develop virtual documentation for output in both printable document and web page formats, using Report Package and Model Document elements. The process for generating web virtual documentation differs from that for printable documentation in two ways:

- You use a standard or customized HTML template, which you select in the 'Style' field on the 'Publish as HTML' dialog; you do not specify or use any templates defined in the Model Document element 'RTFTemplate' Tagged Value
- You create a Package list to select the elements for the report, and not a **Model Search** (which would be defined in the 'SearchName' and 'SearchValue' Tagged Values in the Model Document elements)

Access

Ribbon	Publish > Model Reports > HTML > Standard HTML Report or Specification-Specify > Element > Documentation > Publish as HTML
Keyboard Shortcuts	**Shift+F8**
Context Menu	On a diagram, right-click on the Model Document element | Documentation | HTML Report In the **Browser window**: | Documentation | Publish as HTML

Generate web documentation from a Report Package and/or Model Documents

Step	Action
1	On the 'Publish as HTML' dialog, set the options for your web document as required. Select your preferred template - such as a customized Model Document template - in the 'Style' field.
2	Click on the **Generate button** to generate the documentation. The HTML Report Generator works through the defined content of the Report Package element and/or the Model Document elements, and pulls in the information from the listed Packages, formatted according to the template identified in the 'Style' field.
3	Click on the **View button** to view the documentation.

Publish to Joomla!

Enterprise Architect provides a facility for publishing model content as Articles (HTML pages) in a local Joomla! installation. As you develop your model you can publish it in the local Joomla! Repository, where the outline structure closely mirrors the model hierarchy and makes it very simple to explore models on-line. You can publish either an entire model or a single branch of the model into the local Joomla! Repository. This local Repository can then be easily transferred to a live Joomla! Server.

Publish model content into Joomla!

To publish the content of a model as Articles in Joomla! you work through these stages:

Stage	Action
1	Install Joomla! locally and create a Joomla! repository.
2	Set up an ODBC link to the created Joomla! repository.
3	Connect to the Joomla! repository from Enterprise Architect and publish Articles.
4	(Optional) Transfer the local Joomla! site to a live Joomla! server. (See *Learn more*.)

Notes

- Publish to Joomla! is supported in all Editions of Enterprise Architect
- In the Corporate, Unified and Ultimate Editions of Enterprise Architect, if security is enabled you must have 'Generate Documents' permission to Publish as Joomla! Article

Install Joomla! Locally

Before you can publish model contents as Joomla! Articles, you must install Joomla! locally on your workstation.

There are three stages in the process of installing Joomla! locally:

1. Download and install software that Joomla! requires to run locally
2. Download and install Joomla! itself
3. Configure Joomla! and create a MySQL repository into which to publish the Articles

Stage 1: Download and install software that Joomla! requires to run locally

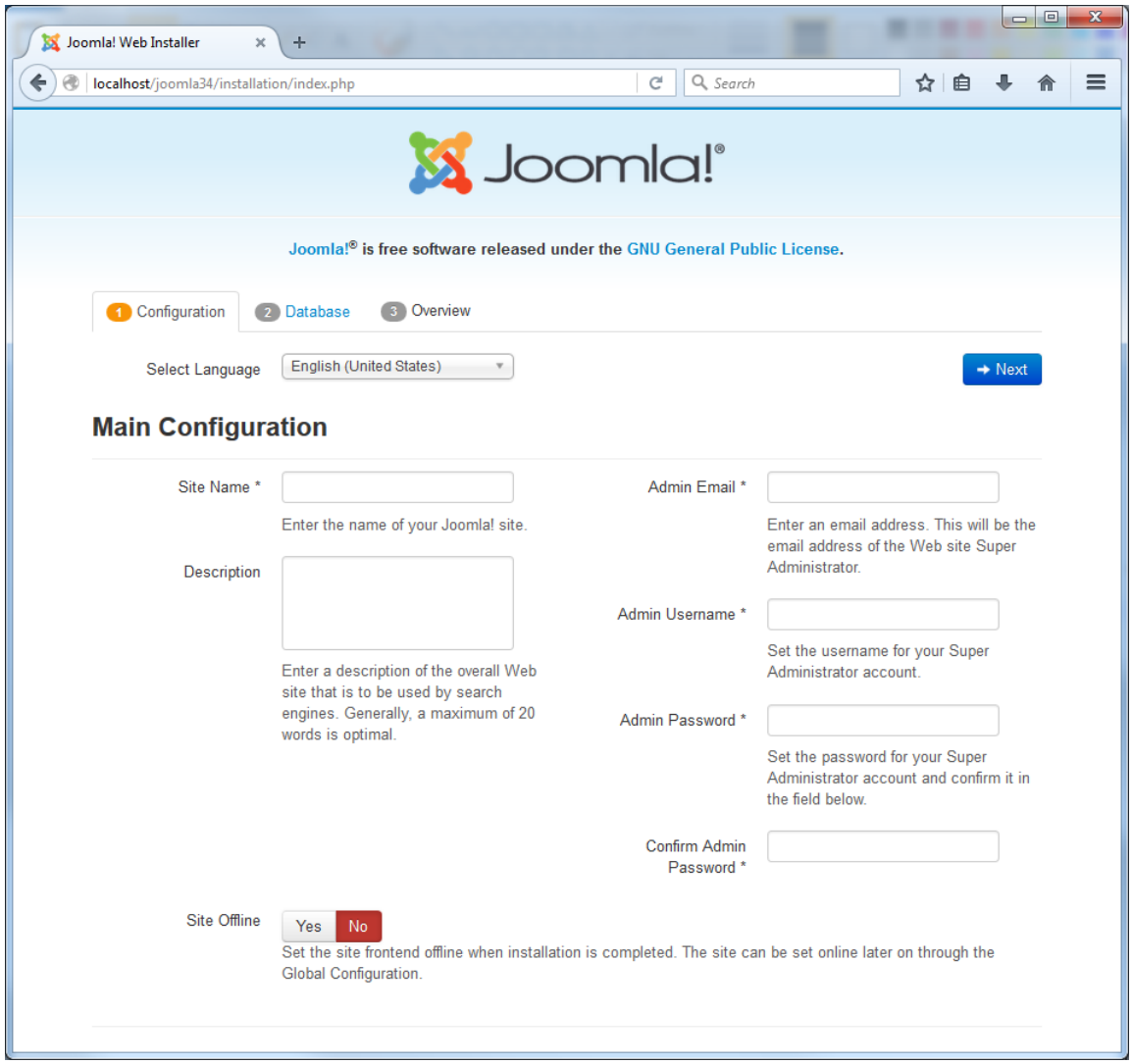
Step	Action
1	Joomla! depends on these software applications to run locally: • Apache web server • MySQL DBMS • PHP scripting language If you do not already have these applications, you must install them on your workstation. To do this, you can use a solution package such as XAMPP, which you can download from this site. The instructions for installing XAMPP are available here.

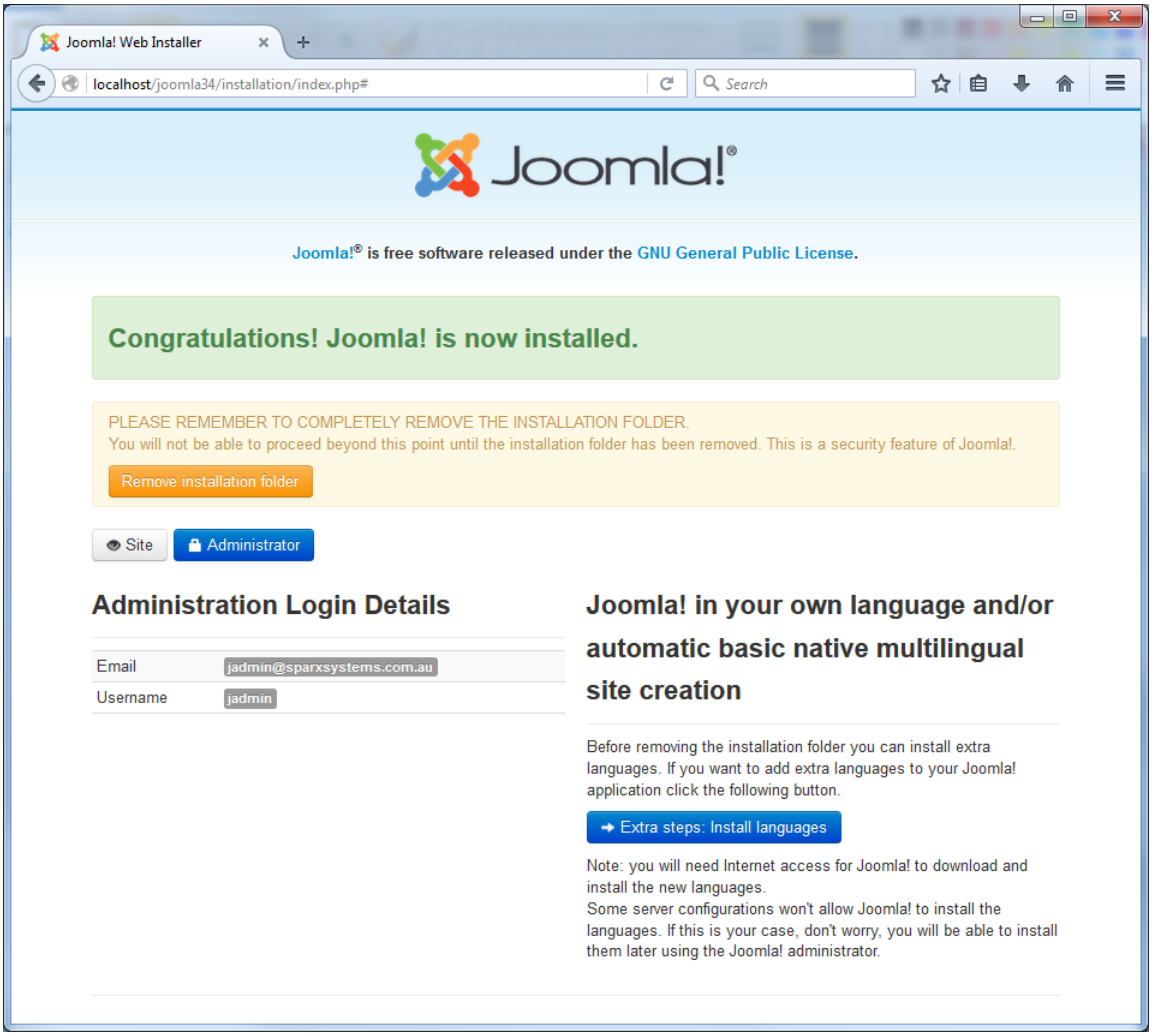
Stage 2: Download and install Joomla!

Step	Action
1	In the xampp\htdocs folder, create a folder for the Joomla! installation (so, if you have installed XAMPP in the default location then create the folder under c:\xampp\htdocs). Give this folder a suitable name. **Note - this folder name will be used as the Joomla! Repository name in the MySQL database in Stage 3.**
2	Download the Joomla! Full package from here.
3	Unzip the contents of the downloaded .zip file into the Joomla! folder you created in Step 1. For detailed instructions on installing Joomla! locally, refer to the section *Download and install Joomla!*.

Stage 3: Configure Joomla! and create a MySQL repository in which to publish Articles

Step	Action

1	Check that the Apache and MySQL modules are running in the 'XAMPP Control Panel'.
2	Open a web browser and type in the URL: http://localhost/<joomla_folder_name> where <joomla_folder_name> is the name of the Joomla! folder that you created in Stage 2. The Joomla! Web Installer page displays. 
3	Complete all the required fields on the 'Main Configuration' tab, and click on the **Next button**. The 'Database Configuration' tab displays.
4	Complete the fields on the 'Database Configuration' tab as shown: • Database Type = **MySQL** • Host Name = **localhost** • Username = database user (the default MySQL user is **root**) • Password = database user password (the default MySQL user **root** has no password) • Database Name = **this must be same as the name of the Joomla! folder you created earlier in Step 1 of Stage 2** Click on the **Next button**; the 'Finalization' tab displays.

5	On the 'Finalization' tab, in the 'Install Sample Data' section, select the 'None' option. Click on the **Install button** to initiates the Joomla! repository creation process.
6	When the Joomla! repository installation is complete, the 'Congratulations' and 'Administration Login Details' panels display. 
7	Click on the **Remove Installation folder button**, and then on the **Site button**. This opens the Joomla! site, which is now ready to receive the Articles that you create.

Notes

- Enterprise Architect has been tested with Joomla! versions 3.4/3.5/3.6 and XAMPP versions 5.6.12/5.6.23 (XAMPP Control Panel version 3.2.1/3.2.2 respectively)

Set up a Joomla! ODBC Link

After you have created a Joomla! repository in your MySQL database, you must set up the MySQL ODBC driver so that you can connect to this repository from Enterprise Architect and publish model contents as Articles.

Prerequisites

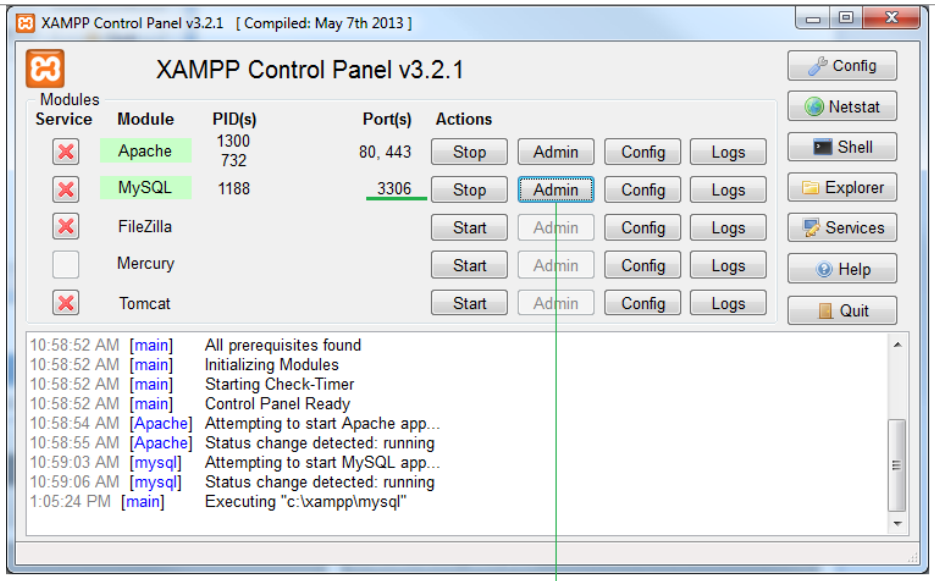
Install:

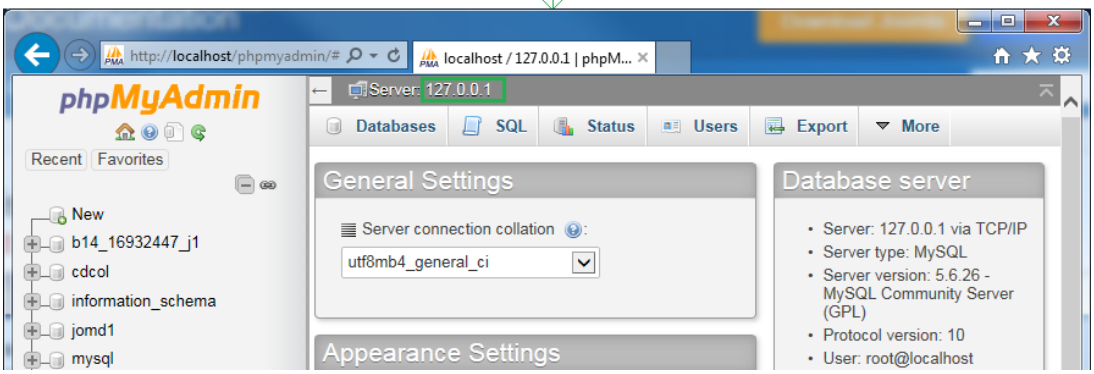
- MySQL DBMS (this will be part of the XAMPP installation)
- MySQL ODBC driver software version 5.3.4 or higher

Set up the ODBC Driver

Your MySQL ODBC driver is now available to connect to the Joomla! Repository from Enterprise Architect.

Step	Action
1	Under both 32-bit and 64-bit operating systems, Enterprise Architect requires 32-bit ODBC drivers to connect to a repository through ODBC. You can quickly load the correct 32-bit ODBC Data Source Administrator by selecting the 'Configure > User Tools > ODBC Data Sources' ribbon option. The **ODBC Data Source Administrator window** displays.
2	Click on the **Add button** in 'User DSN' tab of the dialog. The 'Create New Data Source' dialog displays, through which you add a new DSN.
3	Click on the appropriate MySQL ODBC driver in the list, and click on the **Finish button**. The 'MySQL Connector/ODBC' dialog displays.
4	Enter these configuration details: • A data source name for the connection • (Optional) a description of the connection • The host address of the DBMS server - – if you have installed MySQL DBMS using XAMPP, then the host address and Port can be determined as boxed and underlined, respectively, in green on these control panels:





- User name and password (the default user name for a MySQL repository is **root** with no password)
- The database name on the selected server - this name will be the name of the Joomla! folder you created when you installed Joomla! locally

5	Click on the Details>> button to set the advanced options.
6	Select these checkboxes (where provided): • 'Return matched rows instead of affected rows' ('Conversion' or 'Cursors/Results' tab) • 'Allow big result sets' ('Connection' tab)
7	Click on the **Test Connection button** to confirm that the configuration is correct.
8	If the test succeeds, click on the **OK button** to complete the configuration. If the test fails, review your settings.

Publish as Joomla! Article

Enterprise Architect Models can be published as Articles (HTML page) in a local Joomla! installation, using the 'Publish as Joomla! Article' dialog. All child Packages of the selected Package will be automatically included for publishing.

Prerequisites

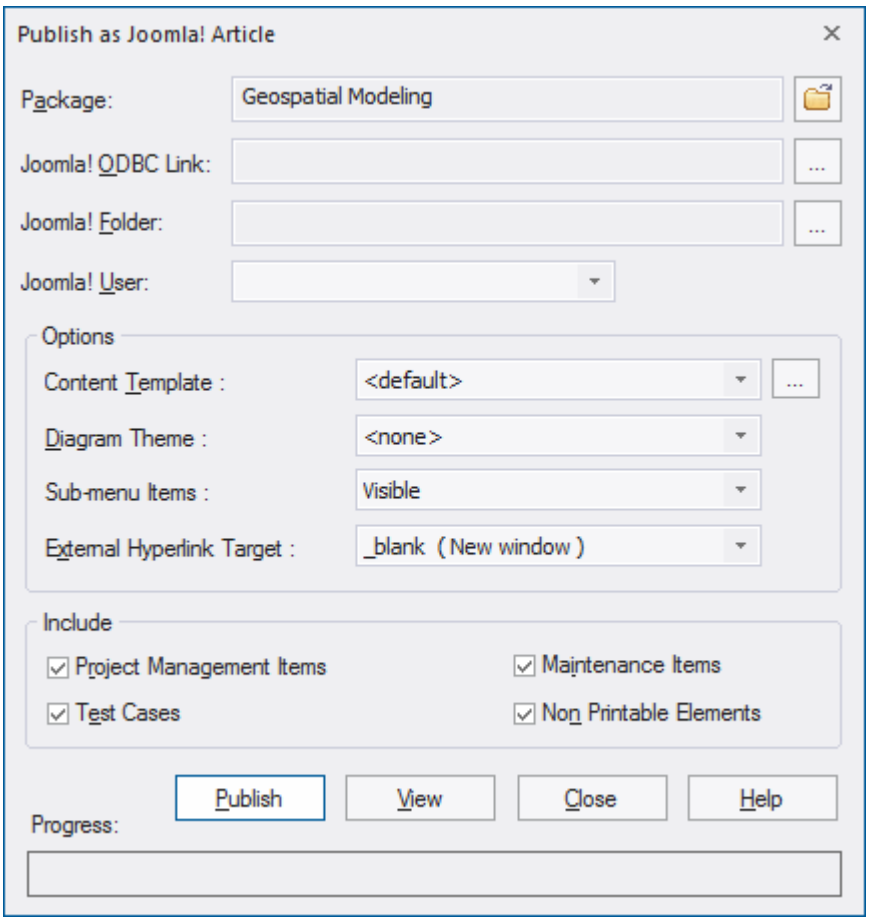
- Install Joomla! locally and create a Joomla! repository
- Set up an ODBC link to the Joomla! repository

Access

In the **Browser window**, click on the Package to be published.

Ribbon	Publish > Model Reports > HTML > Publish to Joomla! or Specification-Specify > Element > Documentation > Publish as Joomla! Article

The 'Publish as Joomla! Article' dialog



Create Joomla! Articles

Use the 'Publish as Joomla! Article' dialog to capture the Package contents as a Joomla! Article.

Field/Button	Action
Package	Displays the name of the currently-selected Package. If this is not the required Package, click on the button and select the correct Package.
Joomla! ODBC Link	Click on the button to browse for and select the Joomla! ODBC Link.
Joomla! Folder	Click on the button to select the Joomla! Folder in your file system. Note that this folder should have the same name as the database in the selected Joomla! ODBC Link, and will usually be located at *C:\xampp\htdocs* (if you have installed XAMPP).
Joomla! User	Click on the **drop-down arrow** and select the user in the specified Joomla! Repository, to act as author for the created Article(s).
Content Template	(Optional) Click on the **drop-down arrow** and select a custom content template to apply, in place of the default content template. Use the '<default>' value for the standard, system-provided content template. Click on the button to open the 'Joomla Content Template' dialog for creating, modifying, and deleting content templates.
Diagram Theme	Click on the **drop-down arrow** and select a Diagram Theme to apply the color, line thickness and font style to the diagrams of elements in the article, if those elements are using default values. If you set the field to '<none>' the system default theme defined in the 'Preferences' dialog, 'Diagram > Themes' page will be applied.
Sub-menu Items	Click on the **drop-down arrow** and select whether to show/hide sub-menu items. • Hidden - sub-menu items are visible only when a parent menu item is clicked • Visible - sub-menu items are always visible
External Hyperlink Target	Click on the **drop-down arrow** and select whether to show external hyperlinks in the same window or tab, or in a new window or tab. • _blank (New window) - opens the hyperlink in a new window or tab • _self (Same frame) - opens the hyperlink in the same window or tab
Include	Select the appropriate checkboxes to include, in the published Article: • Project Management Items – element Resource Allocation, Efforts, Risks and Metrics • Test Cases – element Test Cases • Maintenance Items – element Changes, Issues, Defects, Tasks, Decisions and Events

	• Non Printable Elements – elements that are set as non-printable on the diagram
Publish	Click on this button to publish the selected Package as a Joomla! Article. Messages concerning the progress of publishing are displayed in the **System Output** window.
View	Once the publishing process has completed, you can click on this button to launch your default web browser and view the published Article. The Article will be found under the URL: http://localhost/<joomla_folder_name>/ where <joomla_folder_name> is the name of the selected Joomla! folder.
Close	Click on this button to close the 'Publish as Joomla! Article' dialog.
Help	Click on this button to display this Help topic.

Notes

- Publish to Joomla! is supported in all Editions of Enterprise Architect
- In the Corporate, Unified and Ultimate Editions of Enterprise Architect, if security is enabled you must have 'Generate Documents' permission to Publish as Joomla! Article
- Model items published as Joomla! Articles will be displayed in the **System Output** window – double-click on a Package, element or diagram in the System Output window to highlight it in the **Browser window**
- When publishing large models, it is recommended that you hide the sub-menu items; the option 'Sub-menu Items' is similar to Lazy Load in Enterprise Architect - when sub-menu items are hidden, the 'EA Browser window' menu will initially show only the top-level menu items; clicking on a top-level menu item will then show the child menu items, if any
- If the parent Package of the Package being published in the Enterprise Architect model already exists in Joomla!, then the selected Package will be published as a child of that parent Package in Joomla!
- You can search published Articles by enabling Joomla!'s built-in Search functionality in the website
- It is recommended that you do not access or refresh the Joomla! website when articles are being published from Enterprise Architect
- If you have accidentally accessed or refreshed the Joomla! Website, or you see a PHP script error on top of the web page, log into the Joomla! 'Administrator Control Panel' and click on the Rebuild button in the Joomla! 'Menu Manager'
- The Joomla! 'Administrator Control Panel' can be accessed through this URL:
  http://localhost/<joomla_folder_name>/administrator
  where <joomla_folder_name> is the name of the Joomla! folder

Joomla! Content Templates

You can use the Joomla! Content Template Editor to customize the content of published Joomla! Articles.

Access

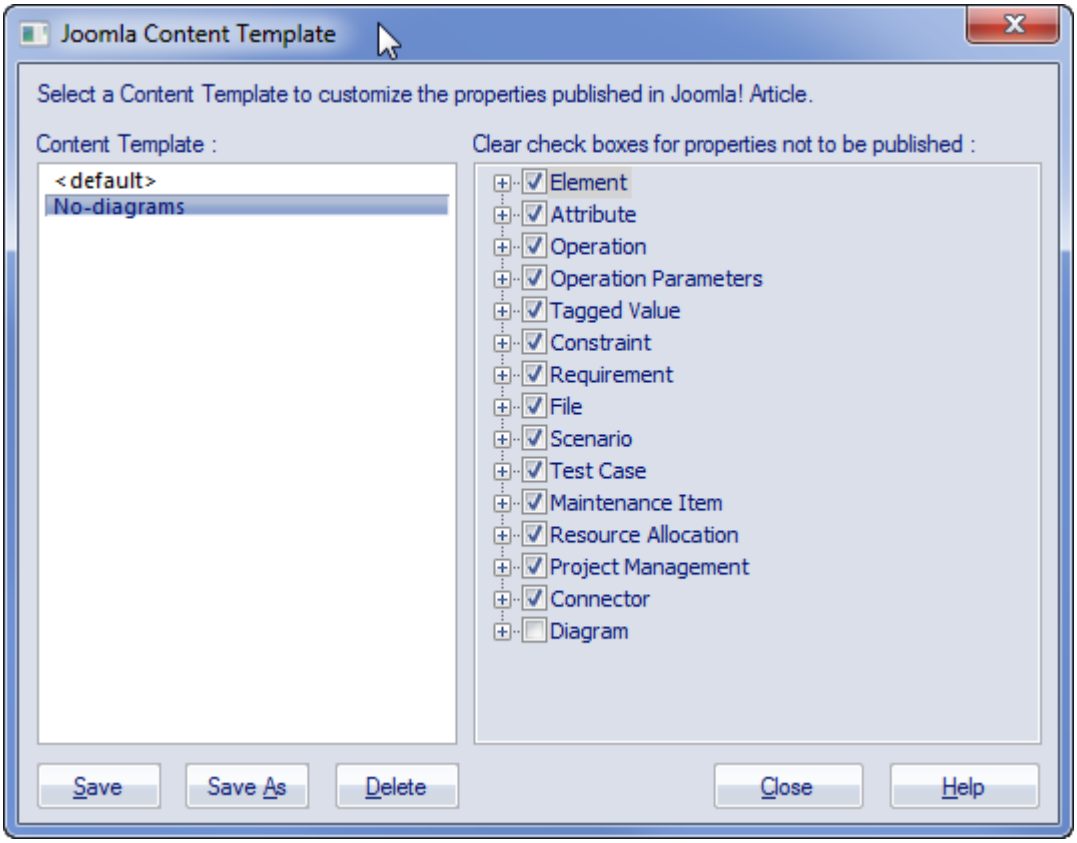
On the 'Publish as Joomla! Article' dialog, click on the 'Content Template' button

Ribbon	Publish > Model Reports > HTML > Publish to Joomla! or Specification-Specify > Element > Documentation > Publish as Joomla! Article

Manage Joomla! Content Template

Field/Button	Action
Content Template	Lists the available Content Templates. The <default> template is a predefined system template that is protected from editing and deletion. You can copy it to create new Content Templates.
Clear checkboxes for properties not to be published	This panel displays a full list of the object types that are supported as content. You can click on the expansion box against an object type to display the properties of that type. Clear the checkbox against each object or property that you do not want to include in the Joomla! Articles.
Save	Click on this button to save any changes you have made to the selected content template.
Save As	Click on this button to save the changes you have made under a different template name (that is, copy the selected template).
Delete	Click on this button to delete the selected template.
Close	Click on this button to close the 'Joomla! Content Template' dialog.
Help	Click on this button to display this Help topic.

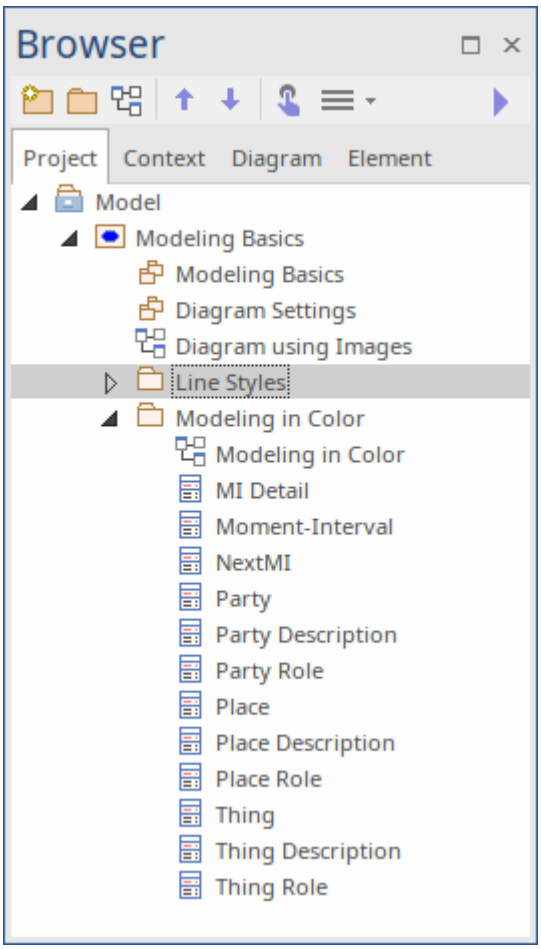
The 'Joomla! Content Template' dialog


Joomla Content Template
Select a Content Template to customize the properties published in Joomla! Article.
Content Template :
<default>
No-diagrams
Clear check boxes for properties not to be published :
Element
Attribute
Operation
Operation Parameters
Tagged Value
Constraint
Requirement
File
Scenario
Test Case
Maintenance Item
Resource Allocation
Project Management
Connector
Diagram
Save
Save As
Delete
Close
Help

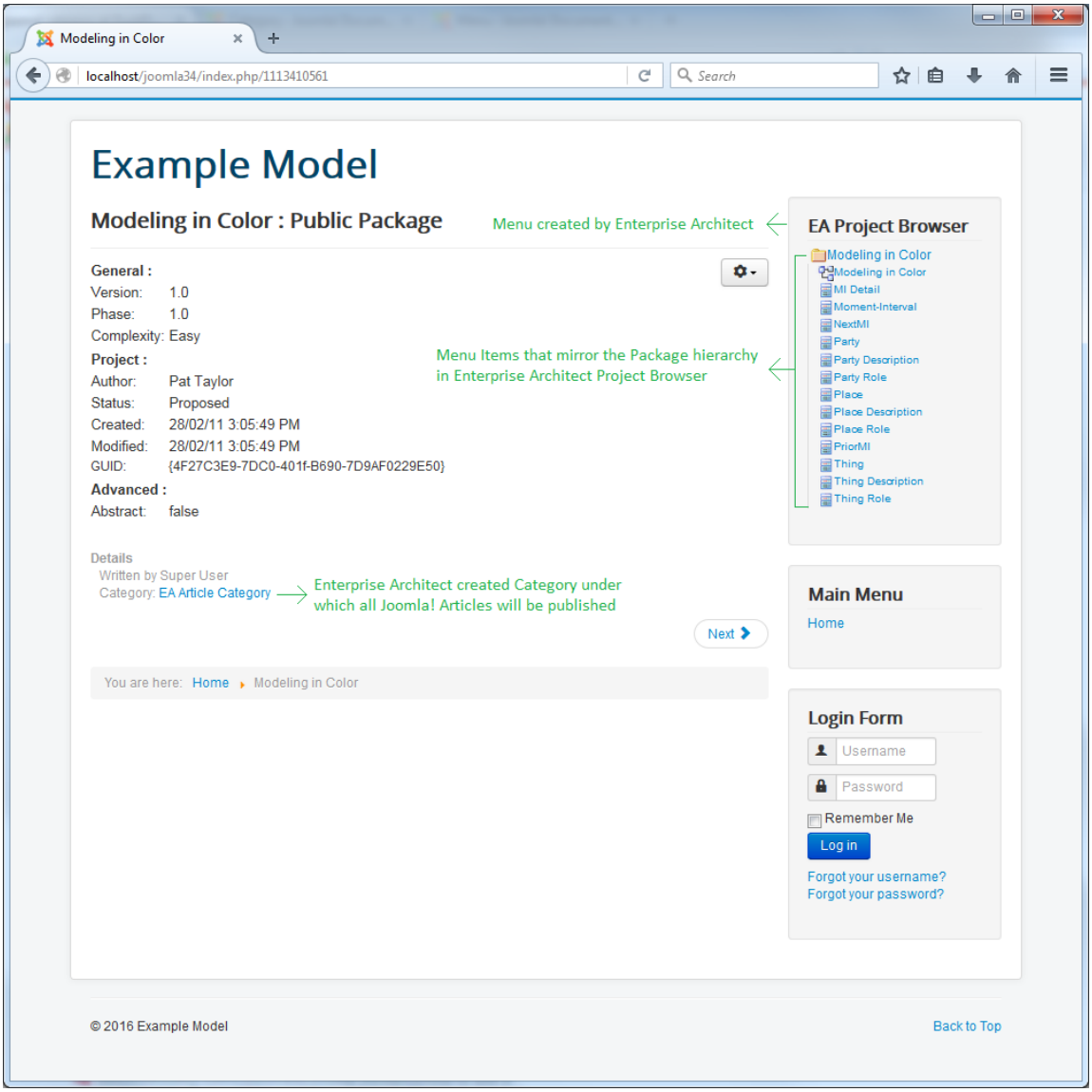
Example - Package Published as Joomla! Article

This example shows the Package 'Modeling in Color' published as a Joomla! Article.

This is the expanded Package as it appears in the Enterprise Architect **Browser window**:



This is the Package as it appears as a Joomla! Article on the Joomla! site:



Notes

- When a Package is published as a Joomla! Article, Enterprise Architect will:
  - Create a Joomla! *Category* called 'EA Article Category', and publish all the Articles under this category
  - Create a Joomla! *Menu* called 'EA Project Browser' and add menu items for each Article published
- If the 'EA Project Browser' menu already exists when you publish a Package, then Enterprise Architect will reuse this menu and will update it with new menu items

Enable Search in Joomla! Website

Once Enterprise Architect models have been published as Articles (HTML pages) using the 'Publish as Joomla! Article' dialog in the local Joomla! installation, you can search for items within these Articles using Joomla!'s built-in Search facility.

Prerequisites

- Install Joomla! locally and create a Joomla! repository
- Set up an ODBC Link to the created Joomla! repository
- Publish model content as a Joomla! Article

Enable the Joomla! Search Facility

Step	Action
1	Open the 'Joomla! Administrator Control Panel' using this URL: http://localhost/<joomla_folder_name>/administrator where <joomla_folder_name> is the name of your Joomla! folder.
2	Select the 'Extensions | Modules' menu option.
3	Click on the **New button** and select the 'Search' option.
4	Type in a title (such as 'Search'), select a position in the 'Position' drop-down field (this determines where the 'Search' text box is displayed in the website), and click on the Save & **Close button**.
5	Access the Joomla! website by typing this URL in the 'Web Address' field: http://localhost/<joomla_folder_name> where <joomla_folder_name> is the name of your Joomla! folder. You will now see a 'Search' text box displayed.