



Asbru Ltd.  
www.asbrusoft.com  
info@asbrusoft.com

Asbru Ltd.



# Asbru Web Content Management System

## Programming API Guide

*Easily & Inexpensively  
Create, Publish & Manage Your Websites*



## Copyright and Proprietary Information

Copyright Asbru Ltd 1999–2014. This user guide constitutes proprietary information of Asbru Ltd. No part of this user guide may be reproduced, transmitted, transcribed, stored in a retrieval system or translated into any language in any form, by any means, without the written permission of Asbru Ltd.

## Notice

Asbru Ltd. reserves the right to make changes in this user guide at any time and without notice. Asbru Ltd. makes no warranties, express or implied, in this user guide. In no event shall Asbru Ltd. be liable for any indirect, special, incidental or consequential damages arising out of purchase or use of this user guide or the information contained herein.

## Licenses and Trademarks

Asbru Web Content Management and the Asbru logo are trademarks or registered trademarks of Asbru Ltd. in the United Kingdom and other countries. All other company, product, or trade names are trademarks or registered trademarks of their respective holders.

Asbru Web Content Management includes and uses the `wz_dragdrop.js` library, Copyright (c) 2002–2003 Walter Zorn ([www.walterzorn.com](http://www.walterzorn.com)), licensed under the terms of the GNU Lesser General Public License (LGPL) (<http://www.gnu.org/copyleft/lesser.html>).

Asbru Web Content Management includes and uses the `wz_jsgraphics.js` library, Copyright (c) 2002–2004 Walter Zorn (<http://www.walterzorn.com>), licensed under the terms of the GNU Lesser General Public License (LGPL) (<http://www.gnu.org/copyleft/lesser.html>).

Asbru Web Content Management includes and uses the Dynarch DHTML Calendar library, Copyright (c) 2002–2005 Mihai Bazon (<http://www.bazon.net/mishoo> - <http://www.dynarch.com/projects/calendar>), licensed under the terms of the GNU Lesser General Public License (LGPL) (<http://www.gnu.org/licenses/lgpl.html>).

Asbru Web Content Management includes and uses the Kryogenix sortable library, Copyright (c) 1997–2005 Stuart Langridge (<http://www.kryogenix.org/code/browser/sortable/>), licensed under the terms of the MIT License (<http://www.kryogenix.org/code/browser/license.html>).

Asbru Web Content Management includes and uses the SWFupload component and library, Copyright (c) 2006–2007 Lars Huring, Olov Nilzén and Mammon Media, and Copyright (c) 2007–2008 Jake Roberts (<http://www.swfupload.org/>), licensed under the terms of the MIT License (<http://www.opensource.org/licenses/mit-license.php>).

Asbru Web Content Management includes and uses the Prototype library, Copyright (c) 2005 Sam Stephenson (<http://prototype.conio.net/>), licensed under the terms of an MIT-style License (<http://prototype.conio.net/>).

Asbru Web Content Management includes and uses the Scriptaculous library, Copyright (c) 2005 Thomas Fuchs (<http://script.aculo.us/>), licensed as free software.

Asbru Web Content Management includes and uses parts of the Rico library, Copyright (c) 2005 Sabre Airline Solutions (<http://openrico.org/>), licensed under the terms of the Apache License, Version 2.0.

Asbru Web Content Management includes and uses the Rico Livegrid Plus library, Copyright (c) 2006 Matt Brown (<http://dowdybrown.com/>), licensed under the terms of the Apache License, Version 2.0.

Asbru Web Content Management includes and uses the Lightbox library, Copyright (c) 2006 Lokesh Dhakar (<http://www.huddletogether.com/>), licensed under the Creative Commons Attribution 2.5 License (<http://creativecommons.org/licenses/by/2.5/>).

Asbru Web Content Management includes and uses the TableKit library, Copyright (c) 2007 Andrew Tetlaw & Millstream Web Software (<http://www.millstream.com.au/view/code/tablekit/>), licensed as free software.

Asbru Web Content Management includes and uses the jQuery library, Copyright (c) 2011 John Resig (<http://jquery.org/>), licensed under the terms of the MIT License (<http://jquery.org/license/>).

Asbru Web Content Management includes and uses the jsTree library, Copyright (c) 2010 Ivan Bozhanov (<http://jstree.com/>), licensed under the terms of the MIT License (<http://www.opensource.org/licenses/mit-license.php>).

Asbru Web Content Management includes and uses the JavaBeans Activation Framework library, Copyright (c) Sun Microsystems (<http://www.sun.com/>), licensed under the terms of the Sun Microsystems, Inc. Binary Code License Agreement.



Asbru Web Content Management includes and uses the JavaMail library, Copyright (c) 2009 Sun Microsystems (<http://www.sun.com/>), licensed under the terms of the Sun Microsystems, Inc. Binary Code License Agreement.

Asbru Web Content Management includes and uses the Apache Jakarta JCS library, Copyright (c) 2001-2007 The Apache Software Foundation (<http://www.apache.org/>), licensed under the terms of the Apache License, Version 2.0.

Asbru Web Content Management includes and uses the Apache Commons Logging library, Copyright (c) 2003-2007 The Apache Software Foundation (<http://www.apache.org/>), licensed under the terms of the Apache License, Version 2.0.

Asbru Web Content Management includes and uses the Apache log4j library, Copyright (c) 2010 The Apache Software Foundation (<http://www.apache.org/>), licensed under the terms of the Apache License, Version 2.0.

Asbru Web Content Management includes and uses the concurrent library, Copyright (c) Doug Lea (<http://gee.cs.oswego.edu/dl/classes/EDU/oswego/cs/dl/util/concurrent/intro.html>), licensed as free software.

Asbru Web Content Management includes and uses the slidetabs library, Copyright (c) WebStack (<http://www.slidetabs.com/>).

### **Asbru Ltd**

Asbru Ltd. provides Internet/Web services, consultancy and solutions for businesses and individuals. Registered in England - Company Registration No. 3865324 - [www.asbrusoft.com](http://www.asbrusoft.com)



# Asbru Web Content Management System

*Easily & Inexpensively  
Create, Publish & Manage Your Websites*

## Introduction

This document is the Programming API guide for the Asbru Web Content Management System. The user guide describes how you, the website programmer, install, configure and use your own program scripts to be used in and with the Asbru Web Content Management System to create, publish and manage your websites.

This user guide is divided into five main parts:

Part 1 describes how to develop, install and configure your own add-on modules with additional website administration sections, menu items and functionality.

Part 2 describes how to develop, install and configure your own extension program scripts with additional programmed website content for your website pages.

Part 3 describes how to develop, install and configure your own automated product availability and product delivery program scripts for the E-Commerce Add-On Module.

Part 4 describes how to develop, install and configure your own workflow action program scripts for your website content administration workflow processes.

Part 5 describes how to develop, install and configure your own web content editor program scripts for your website content administration editing.

Part 6 describes how to develop, install and configure your own program scripts to customize/extend some of the website administration functionality such as publishing and unpublishing content items; handling of uploaded images and files; validating content details; and validating user account details.

Part 7 describes how to use content from the web content management system in your own custom website content delivery scripts/templates; and with third-party web applications.



## Table of Contents

<b>INTRODUCTION .....</b>	<b>4</b>
<b>TABLE OF CONTENTS .....</b>	<b>5</b>
<b>1 CUSTOM / THIRD-PARTY ADD-ON MODULES .....</b>	<b>7</b>
1.1 Installation and Configuration .....	7
1.2 Development.....	7
1.3 Payment Processing.....	13
<b>2 CUSTOM / THIRD-PARTY EXTENSIONS .....</b>	<b>14</b>
2.1 Installation and Configuration .....	14
2.2 Development.....	14
2.3 Usage.....	15
<b>3 PRODUCT AVAILABILITY AND DELIVERY CUSTOM /THIRD-PARTY EXTENSIONS ....</b>	<b>16</b>
3.1 Product Availability Custom/Third-Party Extensions .....	16
3.1.1 Installation and Configuration .....	16
3.1.2 Development.....	16
3.2 Product Delivery Custom/Third-Party Extensions.....	17
3.2.1 Installation and Configuration .....	17
3.2.2 Development.....	17
3.3 Usage.....	18
<b>4 WORKFLOW ACTION CUSTOM /THIRD-PARTY EXTENSIONS .....</b>	<b>19</b>
4.1 Installation and Configuration .....	19
4.2 Development.....	19
4.3 Usage.....	19
<b>5 WEB CONTENT EDITOR CUSTOM /THIRD-PARTY EXTENSIONS .....</b>	<b>20</b>



5.1	Installation and Configuration .....	20
5.2	Development.....	20
5.3	Usage.....	20
6	WEBSITE ADMINISTRATION FUNCTIONALITY EXTENSIONS .....	21
6.1	External Website Publishing/Archiving Programming API.....	21
6.2	File Upload Programming API.....	21
6.3	Validate Content Data Programming API .....	22
6.4	Validate User Data Programming API.....	23
7	WEB APPLICATION INTEGRATION AND TEMPLATE PROGRAMMING API.....	24
8	CLOUD DEPLOYMENT AND MEDIA STORAGE .....	27
8.1	Media Cloud Storage API.....	27
8.2	Cloud Deployment API .....	28



## 1 Custom / Third-Party Add-On Modules

The Asbru Web Content Management system enables you to create your own custom add-on modules and to use third-party developed add-on modules.

Custom / third-party add-on modules can be integrated with the Asbru Web Content Management system administration pages and can have their own administration section and/or add menu-items to the existing administration sections.

Custom /third-party add-on modules can also be payment service provider modules for use with the E-Commerce Add-On module

### 1.1 Installation and Configuration

To install an additional add-on module the module files must be copied to the web server (as default as a new folder under the "/webadmin/module/" folder).

The Asbru Web Content Management system must be configured to load the additional add-on module by editing the "/webadmin/module/config.xxx" file (where "xxx" is "aspx", "jsp" or "php" depending on which version of the Asbru Web Content Management system you are using) and adding an additional line to the bottom of the file. For example to activate the "example" module add the following lines:

- .NET:  
`<!-- #include file="example/config.aspx" -->`
- JSP:  
`<%@ include file=" example /config.jsp" %>`
- PHP:  
`<?php include " example /config.php"; ?>`

If multiple add-on modules are configured the order of them determines the order their administration sections and menu items are displayed on the web content management system administration pages.

### 1.2 Development

A custom add-on module can be almost anything:

- Additional functionality tightly integrated with the Asbru Web Content Management system.
- A separate application with its own functionality and administration lightly integrated with the Asbru Web Content Management system for easy access through a single administration system.
- An external web service with its own functionality and administration lightly integrated with the Asbru Web Content Management system for easy access through a single administration system.



The Asbru Web Content Management system defines a simple interface for adding additional sections and/or menu items and/or content administration tabs to the web content management system administration pages.

A custom add-on module must include a module configuration file, which defines if, where and how additional sections and menu items and content administration tabs should be integrated with the web content management system administration pages.

Please see the `"/webadmin/module/example/"` example module for details on a module configuration file. The module configuration file definitions are:

- `module`  
The module name and version number.
- `Toolbar` section
  - `moduleToolbarTitle`  
The module title to be displayed in the web content management system administration pages toolbar.
  - `moduleToolbarImage`  
The module image to be displayed in the web content management system administration pages toolbar.
  - `moduleToolbarLink`  
The web address which the module's web content management system administration pages toolbar title and image should link to.
- `Home`
  - `moduleHomeIntroTitle`  
The module introduction title to be displayed on the web content management system Home page.
  - `moduleHomeIntroText`  
The module introduction text to be displayed on the web content management system Home page.
  - `moduleHomeIntroImage`  
The module introduction image to be displayed on the web content management system Home page.
  - `moduleHomeIntroLink`  
The web address which the module's Home introduction title, text and image should link to.
  - `moduleHomeMenuTitle`  
The module menu item title to be displayed on the web content management system Home page.





- `moduleHomeMenuLink`  
The web address which the module's Home menu item should link to.
- Content administration - Website Content
  - `moduleContentIntroTitle`  
The module introduction title to be displayed on the web content management system content administration page.
  - `moduleContentIntroText`  
The module introduction text to be displayed on the web content management system content administration page.
  - `moduleContentMenuTitle`  
The module menu item title to be displayed on the web content management system content administration page.
  - `moduleContentMenuLink`  
The web address which the module's content administration menu item should link to.
- Content administration - Website Content – Administration Tab
  - `moduleContentTab`  
The module HTML/Javascript id to be used on the web content management system content administration page.
  - `moduleContentTabTitle`  
The module title to be displayed on the web content management system content administration tab.
  - `moduleContentTabContent`  
The module HTML/text content to be displayed on the web content management system content administration page tab, initially.
  - `moduleContentTabURL`  
The web address of the HTML/text content to be loaded and displayed on the web content management system content administration page tab when the tab is selected.
  - `moduleContentTabScript`  
The Javascript code to be executed on the web content management system content administration page tab when the tab is selected.
- Library administration - Media Library
  - `moduleLibraryIntroTitle`  
The module introduction title to be displayed on the web content management system library administration page.



- moduleLibraryIntroText  
The module introduction text to be displayed on the web content management system library administration page.
  - moduleLibraryMenuTitle  
The module menu item title to be displayed on the web content management system library administration page.
  - moduleLibraryMenuLink  
The web address which the module's library administration menu item should link to.
- E-Commerce administration - Products & Orders (E-Commerce Add-On only)
  - moduleEcommerceIntroTitle  
The module introduction title to be displayed on the web content management system e-commerce administration page.
  - moduleEcommerceIntroText  
The module introduction text to be displayed on the web content management system e-commerce administration page.
  - moduleEcommerceMenuTitle  
The module menu item title to be displayed on the web content management system e-commerce administration page.
  - moduleEcommerceMenuLink  
The web address which the module's e-commerce administration menu item should link to.
- Community administration (Community Add-On only) (for future use)
  - moduleCommunityIntroTitle  
The module introduction title to be displayed on the web content management system community administration page.
  - moduleCommunityIntroText  
The module introduction text to be displayed on the web content management system community administration page.
  - moduleCommunityMenuTitle  
The module menu item title to be displayed on the web content management system community administration page.
  - moduleCommunityMenuLink  
The web address which the module's community administration menu item should link to.
- Databases administration (Databases Add-On only) (for future use)



- moduleDatabasesIntroTitle  
The module introduction title to be displayed on the web content management system databases administration page.
  - moduleDatabasesIntroText  
The module introduction text to be displayed on the web content management system databases administration page.
  - moduleDatabasesMenuTitle  
The module menu item title to be displayed on the web content management system databases administration page.
  - moduleDatabasesMenuLink  
The web address which the module's databases administration menu item should link to.
- Statistics administration (Statistics Add-On only) (for future use)
  - moduleStatisticsIntroTitle  
The module introduction title to be displayed on the web content management system statistics administration page.
  - moduleStatisticsIntroText  
The module introduction text to be displayed on the web content management system statistics administration page.
  - moduleStatisticsMenuTitle  
The module menu item title to be displayed on the web content management system statistics administration page.
  - moduleStatisticsMenuLink  
The web address which the module's statistics administration menu item should link to.
- Users administration - User Database
  - moduleUsersIntroTitle  
The module introduction title to be displayed on the web content management system users administration page.
  - moduleUsersIntroText  
The module introduction text to be displayed on the web content management system users administration page.
  - moduleUsersMenuTitle  
The module menu item title to be displayed on the web content management system users administration page.
  - moduleUsersMenuLink  
The web address which the module's users administration menu item should link to.



- Configuration
  - moduleConfigIntroTitle  
The module introduction title to be displayed on the web content management system configuration page.
  - moduleConfigIntroText  
The module introduction text to be displayed on the web content management system configuration page.
  - moduleConfigMenuTitle  
The module menu item title to be displayed on the web content management system configuration page.
  - moduleConfigMenuLink  
The web address which the module's configuration menu item should link to.
- Hosting administration - Hosting Clients (Hosting Edition only)
  - moduleHostingIntroTitle  
The module introduction title to be displayed on the web content management system hosting administration page.
  - moduleHostingIntroText  
The module introduction text to be displayed on the web content management system hosting administration page.
  - moduleHostingMenuTitle  
The module menu item title to be displayed on the web content management system hosting administration page.
  - moduleHostingMenuLink  
The web address which the module's hosting administration menu item should link to.
- Payment Processing configuration (E-Commerce Add-On module only)
  - modulePaymentTitle  
The module introduction title to be displayed on the E-Commerce Add-On module Payment Processing configuration page.
  - modulePaymentOptions  
The payment service provider introduction text and options to be displayed on the E-Commerce Add-On module Payment Processing configuration page. Any number and type of payment processing options can be configured. Please see the included PayPal module for an example.

Please note that all module configuration file definitions must be specified even if they are not used. Set unused module configuration file definitions to "".



### 1.3 Payment Processing

For payment processing custom / third-party add-on modules, an additional program file must also be provided. For a payment processing custom / third-party add-on module titled “PayPal”, a “/webadmin/module/PayPal/payment.aspx”, “/webadmin/module/PayPal/payment.jsp” or “/webadmin/module/PayPal/payment.php” (depending on your programming language version of the web content management system) program file must exist and handle the actual payment processing.

The order details for completed orders are available to the payment processing module. The payment processing module should use the order details to process or prepare the payment for the order, and return payment instructions or payment confirmation to the E-Commerce Add-On module for display on the Order Completed shopping cart / checkout page. Please see the included PayPal module for an example.



## 2 Custom / Third-Party Extensions

The Asbru Web Content Management system enables you to create your own custom extensions and to use third-party developed extensions.

Custom / third-party extensions can be integrated with the Asbru Web Content Management system content to include content from other applications and sources than the web content management system.

### 2.1 Installation and Configuration

To install an additional extension the extension file must be copied to the web server (as default as a new file under the "/webadmin/extension/" folder). No configuration is required.

### 2.2 Development

A custom extension can be almost anything:

- A simple text or HTML file.
- An ASP, .NET, JSP or PHP program script, which generates content programmatically.
- A .NET, JSP or PHP program script, which reads content from a file or a database.
- A .NET, JSP or PHP program script, which reads content from an external web service.

The custom extension is simply included/executed when its special code is used in the web content management system's content, and the output from the custom extension is included in the web content management system's content.

Please see the "/webadmin/extension/" example extension "hello" for details on a simple extension file. The extension file is simply a .NET, JSP or PHP file depending on which programming language you are using and it must generate some output. Parameters (if any) are passed from the web content management system to the extension as a simple session string variable named "extension":

- .NET - hello.aspx:  
`<%@ Page Language="C#" Debug="true" ValidateRequest="false" %>  
<%= "Hello " + Session["extension"] %>`
- JSP - hello.jsp:  
`<%= "Hello " + session.getValue("extension") %>`
- PHP - hello.php:  
`<?php echo "Hello " . $_SESSION["extension"] ?>`

In addition to the "extension" session parameter, the extension can use the standard web server request variables etc. such as the .NET "Request", the JSP "request" and the PHP "\$\_GET" and "\$\_POST".



Typically, custom extensions simply process some input parameters and outputs some content to be displayed as part of the viewed web page. Optionally, a custom extension can also redirect the website user to another web page. To do this the custom extension must output/return a specially formatted redirection command as its only output: "WCM:REDIRECT:URL" (replace "URL" with a web address such as "http://www.asbrusoft.com/" or "/index.html").

## 2.3 Usage

Custom extensions are included in the web content management system content using simple special codes similar to the other special codes used in templates etc:

<code>@@@extension:hello@@@</code>	Defines where the output generated by the extension named "hello" is to be inserted. No parameter is passed to the "hello" extension.
<code>@@@extension:hello(World)@@@</code>	Defines where the output generated by the extension named "hello" is to be inserted. "World" is passed as a parameter to the "hello" extension.



## 3 Product Availability and Delivery Custom /Third-Party Extensions

The Asbru Web Content Management system E-Commerce Add-On module enables you to create your own product availability and delivery custom extensions and to use third-party developed extensions.

Product availability and delivery custom / third-party extensions can be integrated with the Asbru Web Content Management system E-Commerce Add-On module to include content from other applications and sources than the web content management system.

### 3.1 Product Availability Custom/Third-Party Extensions

Product availability custom/third-party extensions can be used on product, shopping cart and checkout pages to check if a product is available. For example, to check your own external stock inventory system to see if a product is in stock; or to check a supplier's external stock inventory system to see if a product is in stock and can be ordered; or to check if non-physical products such as usernames, email addresses and Internet domain names are available or have already be registered.

#### 3.1.1 Installation and Configuration

To install an additional product availability extension the extension file must be copied to the web server (as default as a new file under the "/webadmin/productavailability/" folder). No configuration is required.

#### 3.1.2 Development

A product availability custom extension can be almost anything:

- A simple text or HTML file.
- A .NET, JSP or PHP program script, which reads content from a file or a database.
- A .NET, JSP or PHP program script, which reads content from an external web service.

The product availability custom extension is simply included/executed when a product page or a shopping cart/checkout page with products for which the product delivery custom extension is selected, and the output from the product availability custom extension can be included on the page displayed to the customer.

Please see the "/webadmin/productavailability/" example extension "hello" for details on a simple product availability extension file. The extension file is simply a .NET, JSP or PHP file depending on which programming language you are using and it may generate some output:

- .NET - hello.aspx:  

```
<% Session("productavailability") = "+1 123 In Stock" %>  
<% Session("productavailability") = "0 Unavailable" %>  
<% Session("productavailability") = "-1 345 Available From Supplier" %>
```





- JSP - hello.jsp:  

```
<% session.setValue("productavailability ", "+1 123 In Stock "); %>
<% session.setValue("productavailability ", "0 Unavailable "); %>
<% session.setValue("productavailability ", "-1 345 Available From Supplier "); %>
```
- PHP - hello.php:  

```
<?php $_SESSION["productavailability "] = "+1 123 In Stock "; ?>
<?php $_SESSION["productavailability "] = "0 Unavailable "; ?>
<?php $_SESSION["productavailability "] = "-1 345 Available From Supplier "; ?>
```

In addition to the "extension" session parameter, the extension can use the standard web server request variables etc. such as the .NET "Request.QueryString", the JSP "request" and the PHP "\$\_GET" and "\$\_POST".

The output from the product availability custom extension must be in a specific format – a number followed by a blank followed by some text. The number is not displayed by the web content management system but indicates if a product is available or not:

- A positive number indicates that the product is in stock. The product can be ordered and the customer will be charged for it immediately when the order is placed.
- A negative number indicates that the product is not in stock but that it can still be (back/pre)ordered. The product can be ordered but the customer will not be charged for it immediately when the order is placed (i.e. you will charge the customer when you process the order, manually or through an external system, at a later time).
- The number 0 indicates that the product is unavailable. The product cannot be ordered and it will be removed from the order if an order is placed and the customer will not be charged for it.

The remaining text output will be displayed if/where a @@@availability@@@ special code is used on the product, shopping cart and checkout pages. If the number of products in stock etc. should be displayed then the number should be repeated/included in the text – the first number is not displayed, but only used to indicate the availability.

## 3.2 Product Delivery Custom/Third-Party Extensions

Product delivery custom/third-party extensions can be used to automatically generate digital products and to update external systems when a product has been ordered. For example, to update your own external stock inventory system; or place an order with a supplier; or to generate/deliver non-physical products such as usernames, email addresses and Internet domain names.

### 3.2.1 Installation and Configuration

To install an additional product delivery extension the extension file must be copied to the web server (as default as a new file under the "/webadmin/productdelivery/" folder). No configuration is required.

### 3.2.2 Development

A product delivery custom extension can be almost anything:

- A simple text or HTML file.



- A .NET, JSP or PHP program script, which generates content programmatically.
- A .NET, JSP or PHP program script, which reads content from a file or a database.
- A .NET, JSP or PHP program script, which reads content from an external web service.

The product delivery custom extension is simply included/executed upon successful order and payment completion of a product for which the product delivery custom extension is selected, and the output from the product delivery custom extension can be included in the product delivery page displayed to the customer and/or in the email sent to the customer.

Please see the "/webadmin/productdelivery/" example extension "hello" for details on a simple product delivery extension file. The extension file is simply a .NET, JSP or PHP file depending on which programming language you are using and it may generate some output:

- .NET - hello.aspx:  
`<% Session("productdelivery") = "Hello " %>`
- JSP - hello.jsp:  
`<% session.setValue("productdelivery", "Hello"); %>`
- PHP - hello.php:  
`<?php $_SESSION["productdelivery"] = "Hello"; ?>`

In addition to the "extension" session parameter, the extension can use the standard web server request variables etc. such as the .NET "Request.QueryString", the JSP "request" and the PHP "\$\_GET" and "\$\_POST".

### 3.3 Usage

The product availability/delivery custom extensions are selected in the web content management system through the Custom Extension Program input field on the Product Delivery tab when adding and updating products in the Products & Orders administration section of the web content management system.



## 4 Workflow Action Custom /Third-Party Extensions

The Asbru Web Content Management system enables you to create your own workflow action custom extensions and to use third-party developed extensions.

Workflow action custom / third-party extensions can be integrated with the Asbru Web Content Management system to make and log content administration actions through other applications and sources than the web content management system.

### 4.1 Installation and Configuration

To install an additional workflow action extension the extension file must be copied to the web server (as default as a new file under the "/webadmin/workflowaction/" folder). No configuration is required.

### 4.2 Development

A workflow action custom extension can be almost anything:

- A .NET, JSP or PHP program script, which reads/writes content from/to a file or a database.
- A .NET, JSP or PHP program script, which reads/writes content from/to an external web service.

The workflow action custom extension is simply executed when a workflow action for which the workflow action custom extension is configured is applied to a content item or an e-commerce module order in the web content management system.

Please see the "/webadmin/workflowaction/" example extensions "Workflow Action Module Template" and "Test Workflow Action" for details on a simple workflow action extension file. The "Workflow Action Module Template" simply shows how to read the relevant data from the web content management system. The "Test Workflow Action" logs the workflow action taken to the content item's Revision History data.

The extension file is simply a .NET, JSP or PHP file depending on which programming language you are using and it should read and eventually update the relevant workflow and content or order data from the web content management system as shown in the example extensions.

### 4.3 Usage

The workflow action custom extensions are selected in the web content management system through the Content Changes - Custom Extension Program input field when adding and updating workflow actions in the Configuration administration section of the web content management system. The workflow action custom extension is then automatically executed when the workflow action is applied to a content item or an e-commerce module order in the web content management system.



## 5 Web Content Editor Custom /Third-Party Extensions

The Asbru Web Content Management system enables you to create your own web content editor custom extensions and to use third-party developed extensions.

Web content editor custom / third-party extensions can be integrated with the Asbru Web Content Management system to use another web content editor than the Asbru Web Content Editor included with the web content management system.

### 5.1 Installation and Configuration

To install an additional web content editor extension the extension file must be copied to the web server (as default as a new file under the "/webadmin/webeditors/EXTENSION NAME/" folder). No configuration is required.

### 5.2 Development

A web content editor custom extension is a .NET, JSP or PHP program script, which defines the Javascript functions required and used for the content administration by the web content management system.

The web content editor custom extension is simply included when a content item or an e-commerce module product is added or updated through the administration pages in the web content management system.

Please see the "/webadmin/webeditors/" example "WebEditorAPI" extension (as well as the default Asbru Web Content Editor "HardCore", "HardCore1", "HardCore2" and "textarea" extensions) for details on web content editor extension files.

The extension file is simply a .NET, JSP or PHP file depending on which programming language you are using and it should define the Javascript functions required and used for the content administration by the web content management system as well as load other program scripts required by the custom / third-party web content editor.

### 5.3 Usage

The web content editor custom extension is selected in the web content management system through the Configuration / Features / Content Editing / Content Editor administration page in the web content management system. The web content editor custom extension is then automatically included when a content item or an e-commerce module order is added or updated in the web content management system.



## 6 Website Administration Functionality Extensions

The Asbru Web Content Management system supports customization/extension of some of the website administration functionality through your own custom “/webadmin/api/” program scripts when certain functionality on your website and on your web content management administration pages is used.

### 6.1 External Website Publishing/Archiving Programming API

As default the Asbru Web Content Management system runs on the actual website and delivers the website content dynamically. However, for special requirements you may want to program your own program scripts to be executed when a content item is published or unpublished in the web content management system - for example to copy the file to another web server or an archive/backup server.

When a new or updated content item with a “static filename” is published the web content management system will check if a “/webadmin/api/published”, “/webadmin/api/published.bat” or “/webadmin/api/published.sh” file exists and execute it with the published content item’s static filename as parameter.

When a content item with a “static filename” is unpublished the web content management system will check if a “/webadmin/api/unpublished”, “/webadmin/api/unpublished.bat” or “/webadmin/api/unpublished.sh” file exists and execute it with the published content item’s static filename as parameter.

### 6.2 File Upload Programming API

As default the Asbru Web Content Management system simply adds uploaded images and other files as content items in the web content management system. However, for special requirements you may want to program your own program scripts to be executed when an “image” or a “file” is uploaded to the web content management system – for example to check files for virus infections or to convert the files to other formats or sizes.

When an “image” or a “file” is uploaded the web content management system will check if a “/webadmin/api/image”, “/webadmin/api/image.bat”, “/webadmin/api/image.sh”, “/webadmin/api/file”, “/webadmin/api/file.bat” or “/webadmin/api/file.sh” file exists and execute it with the uploaded file’s filename as parameter. Depending on what the program script does and what the web content management system should do the program script must return/output:

- The same filename as passed to the program script as a parameter  
If the program script has not renamed, moved or deleted the uploaded file.
- The uploaded file’s new filename  
If the program script has renamed or moved the uploaded file. The web content management system will then update the content item with the new filename.
- Nothing  
If the program script has deleted the uploaded file. The web content management system will then also delete the content item.



When an “image” or a “file” is uploaded the web content management system will also check if a “/webadmin/api/image1”, “/webadmin/api/image1.bat”, “/webadmin/api/image1.sh”, “/webadmin/api/image2”, “/webadmin/api/image2.bat”, “/webadmin/api/image2.sh”, “/webadmin/api/image3”, “/webadmin/api/image3.bat”, “/webadmin/api/image3.sh”, “/webadmin/api/file1”, “/webadmin/api/file1.bat”, “/webadmin/api/file1.sh”, “/webadmin/api/file2”, “/webadmin/api/file2.bat”, “/webadmin/api/file2.sh”, “/webadmin/api/file3”, “/webadmin/api/file3.bat” or “/webadmin/api/file3.sh” file exists and execute it with the uploaded file’s filename as parameter. Depending on what the program script does and what the web content management system should do the program script must return/output:

- The filename of new, alternative copy of the uploaded file  
If the program script has created a new, alternative copy of the uploaded file – for example a small resolution version of an image, or a PDF version of a Microsoft Word document, or a compressed version of a program file etc. The web content management system will then create an additional content item for the new file. The original uploaded file’s content item’s corresponding Additional Content / Image 1 / Image 2 / Image 3 / File 1 / File 2 / File 3 attribute will point to the new, alternative file’s content item. The new, alternative file’s content item’s Content Relations / Page Up attribute will point to the original uploaded file’s content item.
- Nothing  
If the program script has not created a new, alternative copy of the uploaded file. No additional content item will be created by the web content management system.

### 6.3 Validate Content Data Programming API

If you have special requirements for the website content you can program your own program scripts to validate content when/before it is saved to the web content management system. If a “/webadmin/api/validatecontent.xxx” (replace “xxx” with your programming language extensions: “aspx”, “jsp” or “php”) program script exists then that will be executed when an added or updated content item is saved. The content item’s data will be posted to the program script as standard HTML POST form data.

The posted content item data can then be validated and the program script should return a structured response to the web content management system:

- “OK”  
The content item data are ok and the content item will be saved.
- “OK:ALERT:MESSAGE”  
The content item data are ok and the content item will be saved and the “MESSAGE” will be displayed to the website administrator (replace “MESSAGE” with your own text).
- “ERROR:CONFIRM:MESSAGE”  
There is a potential problem with the content item data and the website administrator will be prompted with the “MESSAGE” to confirm to save or cancel / re-edit the content item (replace “MESSAGE” with your own text).
- “ERROR:ALERT:MESSAGE”  
There is a problem with the content item data and the content item will not be saved. The “MESSAGE” will be displayed to the website administrator and the website



administrator must re-edit the content before it can be saved (replace “MESSAGE” with your own text).

#### 6.4 Validate User Data Programming API

If you have special requirements for the website user accounts you can program your own program scripts to validate user data when/before they are saved to the web content management system. If a “/webadmin/api/validateuser.xxx” (replace “xxx” with your programming language extensions: “aspx”, “jsp” or “php”) program script exists then that will be executed when an added or updated user account is saved as well as when a user registers on the website. The user account’s data will be posted to the program script as standard HTML POST form data.

The posted user account data can then be validated and the program script should return a structured response to the web content management system:

- “OK”  
The user account data are ok and the user account will be saved.
- “OK:ALERT:MESSAGE”  
The user account data are ok and the user account will be saved and the “MESSAGE” will be displayed to the website administrator (replace “MESSAGE” with your own text). For website user registrations the “MESSAGE” will be displayed to the website user.
- “ERROR:CONFIRM:MESSAGE”  
There is a potential problem with the user account data and the website administrator will be prompted with the “MESSAGE” to confirm to save or cancel / re-edit the user account (replace “MESSAGE” with your own text). For website user registrations the “MESSAGE” will be displayed to the website user.
- “ERROR:ALERT:MESSAGE”  
There is a problem with the user account data and the user account will not be saved. The “MESSAGE” will be displayed to the website administrator and the website administrator must re-edit the user account before it can be saved (replace “MESSAGE” with your own text). For website user registrations the “MESSAGE” will be displayed to the website user.



## 7 Web Application Integration and Template Programming API

As default all content and templates etc. are created in and served by the web content management system with no programming required. However, for special requirements you may want to program your own templates, or you may want to integrate content from the web content management system with your own or third-party applications. An easy to use web content management system programming API can be used to access content from the web content management system.

The web content management system programming API functions are located in the "/webadmin.xxx" files, which your web application should include. The web content management system programming API functions are:

- **ReadContent(id)**  
Reads the content item with the given id.
- **ReadPage(id)**  
Reads the page with the given id including its template and additional content items.
- **ReadProduct(id)**  
Reads the product with the given id including its template and additional content items.
- **ReadData(database, id)**  
Reads the data with the given id from the custom content database with the given database name.
- **ContentHeader(id, attribute)**  
Returns the value of the given attribute name for the content item with the given id.
- **PageDOCTYPE(id)**  
Returns the DOCTYPE HTML code for the page with the given id for output at the top of the HTML page.
- **PageHeader(id, attribute, "")**  
Returns the value of the given attribute name for the page with the given id for output in the HTML HEAD.
- **PageHeader(id, element, attribute)**  
Returns the value of the given attribute name for the additional content element of the given element class name for the page with the given id for output in the HTML HEAD.
- **PageStyleSheet(id)**  
Returns the style sheet HTML code for the page with the given id for output in the HTML HEAD.
- **ProductDOCTYPE(id)**  
Returns the DOCTYPE HTML code for the product with the given id for output at the top of the HTML page.





- **ProductHeader(id, attribute, "")**  
Returns the value of the given attribute name for the product with the given id for output in the HTML HEAD.
- **ProductHeader(id, element, attribute)**  
Returns the value of the given attribute name for the additional content element of the given element class name for the product with the given id for output in the HTML HEAD.
- **ProductStyleSheet(id)**  
Returns the style sheet HTML code for the product with the given id for output in the HTML HEAD.
- **StyleSheetHeader()**  
Outputs the HTTP style sheet header ("Content-Type: text/css").
- **DisplayContent(id, attributename)**  
Returns the value of the given attribute name for the page with the given id for output in the HTML BODY.
- **DisplayPage(id, attribute, "")**  
Returns the value of the given attribute name for the page with the given id for output in the HTML BODY.
- **DisplayPage(id, element, attribute)**  
Returns the value of the given attribute name for the additional content element of the given element class name for the page with the given id for output in the HTML BODY.
- **DisplayProduct(id, attribute, "")**  
Returns the value of the given attribute name for the product with the given id for output in the HTML BODY.
- **DisplayProduct(id, element, attribute)**  
Returns the value of the given attribute name for the additional content element of the given element class name for the product with the given id for output in the HTML BODY.
- **OutputContent(content)**  
Executes "@@extension:...@@" special codes in the content and returns the content with the output from the program extensions.
- **CMSHeader(id)**  
Returns the Browse & Edit mode HTML header code for the page with the given id.
- **CMSDisplay(id)**  
Returns the Browse & Edit mode web page header code for the page with the given id.
- **CMSStyleSheet(id)**  
Returns the Browse & Edit mode style sheet web page header code for the page with the given id.



- `CMSTemplate(id)`  
Returns the Browse & Edit mode template web page header code for the page with the given id.
- `CMSLog(id, class, "")`  
Logs a request for the content item of the given class name with the given id.
- `CMSLog(id, "data", database)`  
Logs a request for the data item with the given id from the custom content database with the given database name.

Please see the `"/page.xxx"`, `"/product.xxx"` and `"/data.xxx"` web content management system program files for examples on how the web content management system programming API functions are used.



## 8 Cloud Deployment and Media Storage

### 8.1 Media Cloud Storage API

As default the Asbru Web Content Management system stores the website images and files on the website server. If the website and the Asbru Web Content Management system run on a cluster of website servers they must be setup to use shared or mirrored/replicated file storage to make the website images and files available on all the website servers. Alternatively, you may want to use a cloud storage service (or some other type of shared storage service) for your website images and files.

When the web content management system is configured to use cloud storage through Configuration / System / Website / Media Storage / Cloud Storage the web content management system will execute a number of “/webadmin/api/” program scripts when website images and files are uploaded, copied, moved/renamed, deleted and downloaded:

- /webadmin/api/exists.xxx” (replace “xxx” with: “aspx”, “jsp” or “php”)   
The web content management system needs to know if a given website image/file exists on the cloud storage.
- /webadmin/api/upload.xxx” (replace “xxx” with: “aspx”, “jsp” or “php”)   
A new website image/file has been uploaded to the website and should be uploaded to the cloud storage.
- /webadmin/api/copy.xxx” (replace “xxx” with: “aspx”, “jsp” or “php”)   
A website image/file has been copied and should be copied on/to the cloud storage.
- /webadmin/api/move.xxx” (replace “xxx” with: “aspx”, “jsp” or “php”)   
A website image/file has been moved/renamed and should be moved/renamed on the cloud storage.
- /webadmin/api/delete.xxx” (replace “xxx” with: “aspx”, “jsp” or “php”)   
A website image/file has been deleted and should be deleted from the cloud storage.
- /webadmin/api/download.xxx” (replace “xxx” with: “aspx”, “jsp” or “php”)   
A modified or non-existing website image/file has been accessed and should be downloaded from the cloud storage.

These “/webadmin/api/” program scripts are included with the web content management system with support for a number of cloud storage service providers as well as placeholder comments in the program code for you to add your own custom programming for your own or other third-party cloud storage service providers. Please see the included “/webadmin/api/” program scripts for details.

The Media Cloud Storage API can be used on its own, or in conjunction with the Cloud Deployment API (which requires the use of the Media Cloud Storage API). Please see 8.2 Cloud Deployment API for details.



## 8.2 Cloud Deployment API

The Asbru Web Content Management system can be installed on your own servers locally or with a hosting service provider of your own choice as well as cloud hosting services.

For a cloud hosted installation of the web content management system with dynamic scalability through addition of additional web servers, the web content management system installation may need to automatically detect and connect to the configured database server and media cloud storage when a new web server is added.

As default the Asbru Web Content Management system includes support for a number of cloud hosting services and database servers with automatic detection and connection to the database server. To use the web content management with other cloud hosting services and/or databases than the ones supported as default, you may need to add your own program code to automatically detect the configured database server and generate the database connection string to be used by the web content management system to connect to the configured database server as well as the media cloud storage configuration settings.

The cloud deployment detection and database connection and media cloud storage is handled by the `"/config.cloud.xxx"` (replace `"xxx"` with: `"aspx"`, `"jsp"` or `"php"`) special configuration program script. To add support for other cloud hosting service providers and database servers, simply edit this program script and add your own program code to:

- Detect the cloud deployment settings through the server environment variables or any other way these settings are made available by the cloud hosting services.
- Set the `"database"` to the database connection string for the configured database server as it would be entered into the web content management system's Configuration / System / Database / Database Connection configuration page.
- Set the `"database_init"` configuration setting to any special SQL commands, which may be required to create and initialise the database server.
- Set the `"csservice"` configuration setting to one of the supported media cloud storage service providers or to any other unique id/name for your cloud storage service provider as also used in your Media Cloud Storage API program scripts.
- Set the `"csusername"`, `"cspassword"`, `"csrootpath"` and `"csURLrootpath"` configuration settings to the relevant values as they would be entered into the web content management system's Configuration / System / Website / Media Storage configuration page.

Please see the `"/config.cloud.xxx"` (replace `"xxx"` with: `"aspx"`, `"jsp"` or `"php"`) web content management system program files for examples on how the web content management system programming API functions are used

The Cloud Deployment API must be used in conjunction with the Media Cloud Storage API which is used to store media files. Please see 8.1 Media Cloud Storage API for details.