

UMBRACO I/S

Package actions

Documentation

Per Ploug Hansen

9/25/2008

Version 1.1, Rewritten documentation, added description of undoing, and added some much needed formatting

Version 1.0: Initial documentation of package actions for codegarden 2008

This document describes the package actions for umbraco 4's package installer. Package actions are small independent methods that consume small piece of XML and perform tasks based on this xml.

Package actions

Table of Contents

- Introduction 3
 - What is package actions?..... 3
 - Adding actions to a package script 3
 - Standard fields on all actions 3
 - Standard elements on all actions..... 3
 - Installing, Uninstalling and Undo 3
- Available actions: 4
 - Add Application..... 4
 - Add Application Tree..... 4
 - Add dashboard section 4
 - Add Xslt Extension..... 5
 - Add Rest Extension 5
 - Add document type 5
 - Move root document..... 5
 - Publish root document 6
 - Add string to html element..... 6
 - Publish root document 6

Introduction

What is package actions?

Package actions are a simple way to perform common tasks while installing packages / nitros. It is XML based and could be compared to f.ex. nant tasks. Actions perform small configuration related tasks, which could f.ex. be allowing a document type on another type, moving a document to another location or adding a string to an existing template.

Currently we have 10 actions available. If you need to perform more advanced installation routines, you will have to build a custom installer (.ascx based) or write your own package action using the `IPackageAction` Interface, which will then automatically be picked up by umbraco.

Adding actions to a package script

With the build-in packager, it is possible to add the actions you need directly from the UI. Just add `<Action>` nodes to the textare.

Alternatively the actions can be manually added to the package manifest by adding a `<Actions>` node containing the package actions xml.

Standard fields on all actions

- **Undo**, *optional*, is true by default.
- **Alias**, *mandatory*, The alias of the package action to be executed
- **Runat**, *mandatory*, can either be install or uninstall

Standard elements on all actions

All actions are a `<Action>` node element (notice the uppercase A)

Installing, Un-installing and Undo

In the first version of the package actions, about 10 actions were included, the high number of actions was caused by the fact that every action needed a corresponding uninstall action. So a "Insert string into template" action would also need a "Remove string from template" action to be un-installable.

This led to a lot of confusion and also opened up the possibility that the actions could be used for evil, (like removing the string "`<umbraco:macro>`" from all templates, would cause quite a bit of havoc.

So in this revised version, an undo attribute has been put on all actions. An action will always contain an undo action which is turned on by default, but can be turned off.

At the same time, all actions that was actually only there to perform uninstall actions has been removed. Depending on community feedback, some of these might be included again.

Available actions:

Add Application

Creates a new application, and adds it to the db.

Alias: addApplication

```
<Action runat="install" [undo="false"] alias="addApplication" appName="Application Name"
appAlias="myApplication" appIcon="application.gif"/>
```

Add Application Tree

Creates a new application tree, and adds it to the db.

Alias: addApplicationTree

```
<Action runat="install" [undo="false"] alias="addApplicationTree" silent="[true/false]"
initialize="[true/false]" sortOrder="1" applicationAlias="appAlias" treeAlias="myTree" treeTitle="My
Tree" iconOpened="folder_o.gif" iconClosed="folder.gif" assemblyName="umbraco"
treeHandlerType="treeClass" action="alert('you clicked')"/>
```

Add dashboard section

Creates a new dashboard section. Uses the standard dashboard xml as a child node of the action itself.

Alias: addDashboardSection

```
<Action runat="install" [undo="false"] alias="addDashboardSection" dashboardAlias="MyDashboardSection">
  <section>
    <areas>
      <area>default</area>
      <area>content</area>
    </areas>
    <tab caption="Last Edits">
      <control>/usercontrols/dashboard/latestEdits.ascx</control>
      <control>/usercontrols/umbracoBlog/dashboardBlogPostCreate.ascx</control>
    </tab>
    <tab caption="Create blog post">
      <control>/usercontrols/umbracoBlog/dashboardBlogPostCreate.ascx</control>
    </tab>
  </section>
</Action>
```

Add Xslt Extension

Publishes a document located in the root of the website

Alias: addXsltExtension

```
<Action runat="install" alias="addXsltExtension" assembly="/bin/umbraco" type="umbraco.library"
extensionAlias="umb.lib" />
```

Add Rest Extension

Adds an extension in the restExtensions.config file. For use with /base.

It uses the standard restExtensions xml format to import the extensions. It is possible to import multiple extensions at a time.

Alias: addRestExtension

```
<Action runat="install" alias="addRestExtension" documentName="News">
  <ext assembly="/bin/umbraco" type="umbracoBase.library " alias="currentMember">
    <permission method="login" allowAll="true" />
  </ext>
  <ext assembly="/bin/umbraco" type="umbracoBase.library " alias="currentPage">
    <permission method="data" allowAll="true" />
  </ext>
</action>
```

Add document type

Allows a document type to be created below another document type. Ex: allow TextPage to be allowed under HomePage.

Alias: allowDoctype

```
<Action runat="install" alias="allowDocumenttype" documentTypeAlias="MyNewDocumentType"
parentDocumentTypeAlias="HomePage"/>
```

Move root document

Moves a document located in the root of the website, into a child of another root node. This action is needed to move things around, as everything by default is placed in the root of the website

Alias: moveRootDocument

```
<Action runat="install" alias="moveRootDocument" documentName="News" parentDocumentType="Home" />
```

Publish root document

Publishes a document located in the root of the website

Alias: publishRootDocument

```
<Action runat="install" alias="publishRootDocument" documentName="News" />
```

Add string to html element

Inserts a string into a specific html element in a specific template. The undo option makes sure that the string can be removed again at uninstall.

Alias: addStringToHtmlElement

- templateAlias: name of the target template.
- htmlElementId: the target html element in the target template
- {nodevalue}: the string to be copied to the target element.
- Position: where to place the string in the element: (can only be end/ beginning)

```
<Action runat="install" alias="addStringToHtmlElement" templateAlias="news"
htmlElementId="newsSection" position="[beginning/end]"><![CDATA[hello world!]]></action>
```

Publish root document

Publishes a document located in the root of the website

Alias: publishRootDocument

```
<Action runat="install" alias="publishRootDocument" documentName="News" />
```