

Stellent[®] Content Server

Sample RSS Feeds



STELLENT™



Copyright

© 1996–2006 Stellent, Inc. All rights reserved. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system without written permission from the owner, Stellent, Inc., 7500 Flying Cloud Drive, Suite 500, Eden Prairie, Minnesota 55344 USA. The copyrighted software that accompanies this manual is licensed to the Licensee for use only in strict accordance with the Software License Agreement, which the Licensee should read carefully before commencing use of this software.

Stellent is a registered trademark, and the Stellent logo, Stellent Content Server, Stellent Content Management, Stellent Site Studio, Stellent Content Integration Suite, Stellent Content Portlet Suite, Stellent Desktop Integration Suite, Stellent Dynamic Converter, Stellent Content Publisher, Stellent Inbound Refinery, and Stellent Image Server are trademarks of Stellent, Inc. in the USA and other countries.

Adobe, Acrobat, the Acrobat Logo, Acrobat Capture, Distiller, Frame, the Frame logo, and FrameMaker are registered trademarks of Adobe Systems Incorporated.

Apache is a registered trademark of the Apache Software Foundation. Stellent Content Integration Suite includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

BEA is a registered trademark, and WebLogic is a trademark of BEA Systems, Inc.

HP-UX is a registered trademark of Hewlett-Packard Company.

IBM, Informix, and WebSphere are registered trademarks of IBM Corporation.

Kofax is a registered trademark, and Ascent and Ascent Capture are trademarks of Kofax Image Products.

Linux is a registered trademark of Linus Torvalds.

Microsoft is a registered trademark, and Windows, Word, and Access are trademarks of Microsoft Corporation.

Oracle is a registered trademark of Oracle Corporation.

Plumtree is a registered trademark of Plumtree Software, Inc.

Portions Copyright © 1991-1997 LEAD Technologies, Inc. All rights reserved.

Red Hat is a registered trademark of Red Hat, Inc.

Sun is a registered trademark, and Solaris, Sun ONE, iPlanet, Java, and all Java-based trademarks are trademarks of Sun Microsystems, Inc.

Sybase is a trademark of Sybase, Inc.

UNIX is a registered trademark of The Open Group.

Verity is a registered trademark of Verity, Incorporated.

All other trade names are the property of their respective owners.

Sample RSS Feeds

This component adds XML-based RSS feed support to the Content Server, and Site Studio. It includes several sample RSS feeds for searches, workflows, error logs, dynamic lists, and static lists.

What is RSS?

RSS is an acronym for **Really Simple Syndication**. It allows web sites to display their content in the form of a simple XML file, instead of a complex HTML page. These files are commonly used to create 'web feeds,' which allows people to directly subscribe to any content on a web site.

Several popular web sites offer their content as an RSS web feed. Most news sites allow people to subscribe to RSS feeds, as do most blogs. Some examples are below:

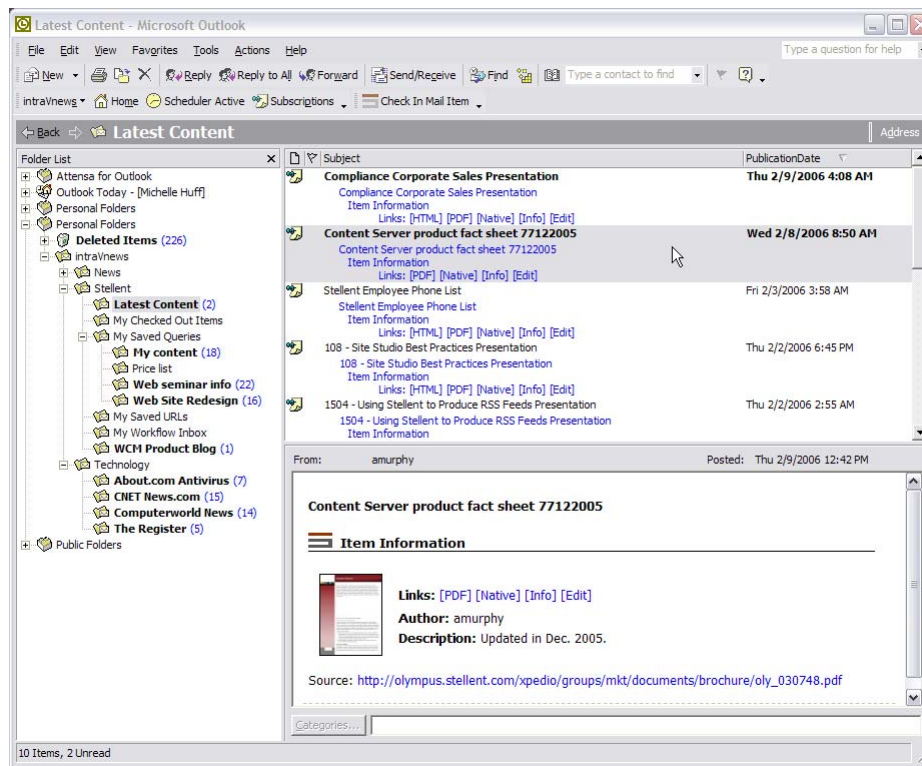
Web Site	Example RSS Feed
LA Times	http://feeds.latimes.com/latimes/news/nationworld/world
New York Times	http://www.nytimes.com/services/xml/rss/nyt/HomePage.xml
Yahoo! News	http://rss.news.yahoo.com/rss/topstories

These feeds offer the same content as their web sites, but in a terse XML format. It is not easier for people to read RSS content directly, but it is very easy for other applications to import the data. Below is an example of how an RSS feed is formatted:

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<rss version="2.0">
  <channel>
    <title>News Stories</title>
    <copyright>Copyright (c) 2006 Yahoo! Inc. All rights reserved.</copyright>
    <link>http://news.yahoo.com/i/716</link>
    <description>Top Stories</description>
    <item>
      <title>Stellent Releases RSS Samples </title>
      <link>http://www.stellent.com/en/news/releases/product/</link>
      <pubDate>Tues, 14 Feb 2006 18:58:46 GMT</pubDate>
      <description>Stellent, Inc. (Nasdaq: STEL), a global provider of content
        management solutions, announced today a new strategy for leveraging really
        simple syndication (RSS) within an enterprise content management (ECM)
        framework.</description>
    </item>
  </channel>
</rss>
```

In order to consumer an RSS feed, the user must have a RSS Reader. Most email software can subscribe to RSS feeds, but not all. If you are using Outlook, you will need to install a third-party plug-in to support RSS. One option is the IntraVNews plugin (<http://www.intravnews.com>).

When the content on the web sites change, the RSS feeds are updated as well. Your RSS reader will download the feeds periodically, to check for any new content. When something is available, you will receive a notification that looks similar to an email:



In effect, you will be quickly notified anytime the content on the web site has changed. RSS Feeds are an excellent alternative to sending emails, or manually reloading web sites to find new content. In short, RSS feeds are currently the most efficient way to keep up-to-date on content on multiple web sites.

How does RSS Work with the Content Server?

In a Content Server application RSS feeds can be used to display lists of simple information. This sample includes RSS feeds for search results, lists of items checked-out, lists of items in workflow, as well as the Content Server logs.

Placing this data in a RSS feed makes it easy to be updated when the lists change. In the case of saved searches, this allows a user to be instantly notified when a new content item matching a search query is added to the repository. In the case of Content Server logs, an administrator will be instantly notified when a serious error occurred. The latter is particularly useful when managing multiple Content Servers.

Instating the Samples

This component can be installed with the Admin Server, or the Component Wizard. After installation, you may optionally adjust the configuration settings for your Content Server. See the sections below on RSS Feeds for Administrators for more information.

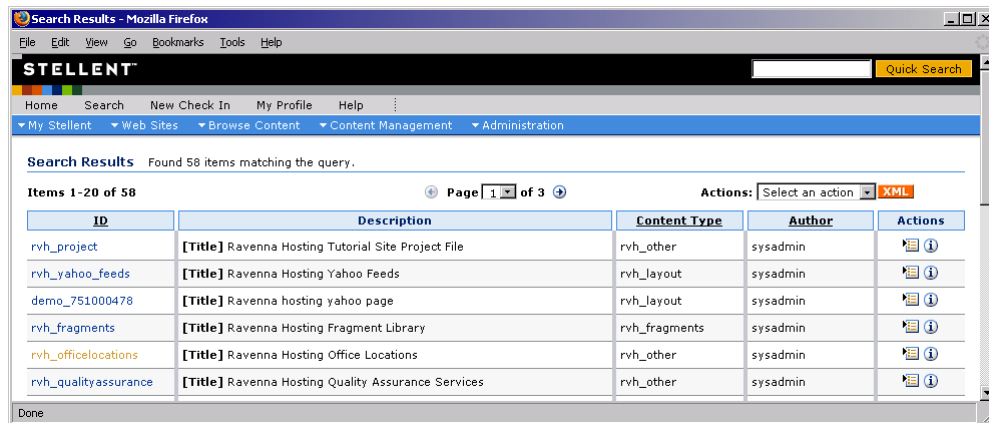
Using the Sample RSS Feeds

You will use RSS Feeds differently, depending on if you are a consumer, a contributor, or an administrator. In most situations, you will want your site designer to create more RSS Feeds than the ones supplied in this sample. In general, this can be accomplished with a new Site Studio page, or by placing a link on specific Content Server pages. See **Creating New RSS Feeds** below for information on how to accomplish this.

Consumers

If you are a consumer, you may only be interested in a handful of RSS feeds. For example, the feed that contains information about news, new products, or press releases. As mentioned above, the site designer will need to anticipate the needs of the audience, and create RSS Feeds when appropriate.

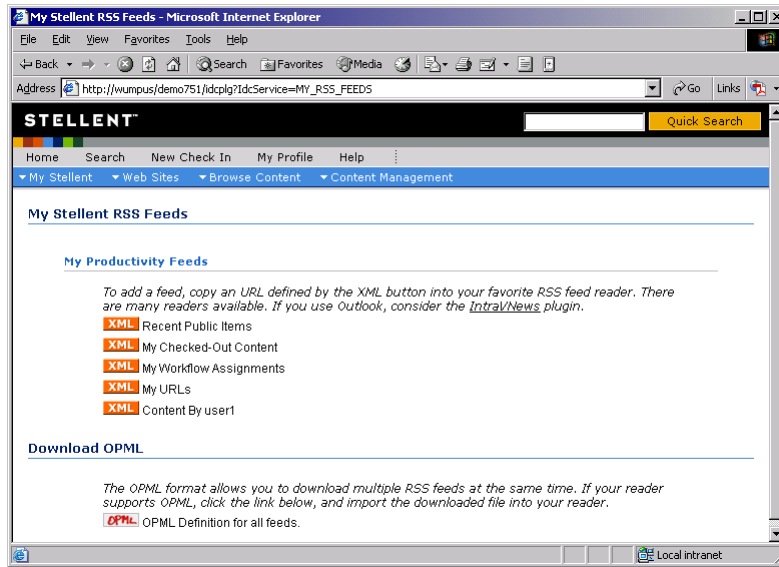
Alternatively, any Content Server user can create their own feeds by running a search, and subscribing to the feed on the Search Results page:



This allows users to set up their own feeds, according to their specific needs.

Content Contributors

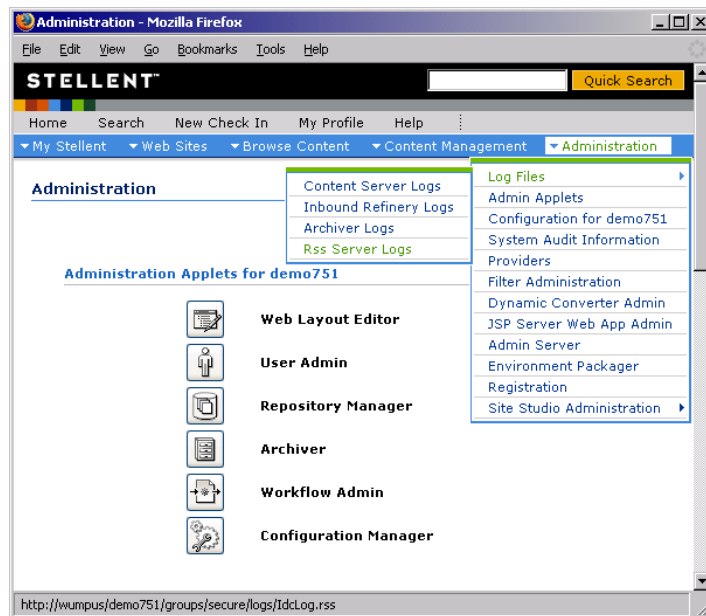
If you are a contributor, you will also be interested in feeds that help you contribute new content. This sample adds the **My Rss Feeds** page, which contains a collection of feeds useful for a contributor. These include lists of your checked-out content, items in workflow, and the feeds for any saved search.



In addition, it contains a list of administrator-defined items, such as Recent Public Items, and content authored by the system administrator.

Administrators

If you are an administrator, you will also be interested in RSS Feeds that help you maintain the Content Server. For example, you may want instant notification when a serious error occurs in the Content Server. This can be accomplished with expensive server monitoring solutions, but it can also be done by publishing error logs as RSS Feeds, and then subscribing to those feeds.



This sample includes RSS Feeds for the Content Server error logs. You can enable them by setting the configuration flag **ExtraLogWriterPath**, mentioned at the end of this document. After which you will see another link for administrative log files:

Subscribe to this feed to have your RSS reader notify you when an Content Server error occurs.

Creating New RSS Feeds

In addition to the out-of-the-box RSS feeds, this sample includes tools that allow you to create new feeds. You can create feeds of other Content Server pages, Site Studio dynamic lists, or Site Studio static lists. You can also consume a remote RSS feed, and display the results as HTML on a web page.

Below are a handful of tutorials about how to set up custom RSS feeds with the existing tools. These tools are described in detail in the Developer's Reference section.

Site Studio Feeds

There are several ways to add RSS support to an existing Site Studio page. The examples below demonstrate how to add them to the standard Ravenna Hosting Tutorial for Site Studio 7.6. If you are using an older version of Site Studio, or a different web site, these step may be different.

To begin, you should upload the sample fragment library that ships with this component. It should be located here:

```
stellent/custom/RSSFeeds/samples/Site Studio/rssfeeds_fragment_lib.zip
```

You can deploy this from Site Studio Designer, by selecting **File > Fragments > Upload Fragment Library**.

You should also check-in the RSS layouts. These are located here:

```
stellent/custom/RSSFeeds/samples/Site Studio/rss_dynamic_list_layout.hcsp
stellent/custom/RSSFeeds/samples/Site Studio/rss_static_list_layout.hcsp
stellent/custom/RSSFeeds/samples/Site Studio/rss_office_loc_static_list.hcsp
```

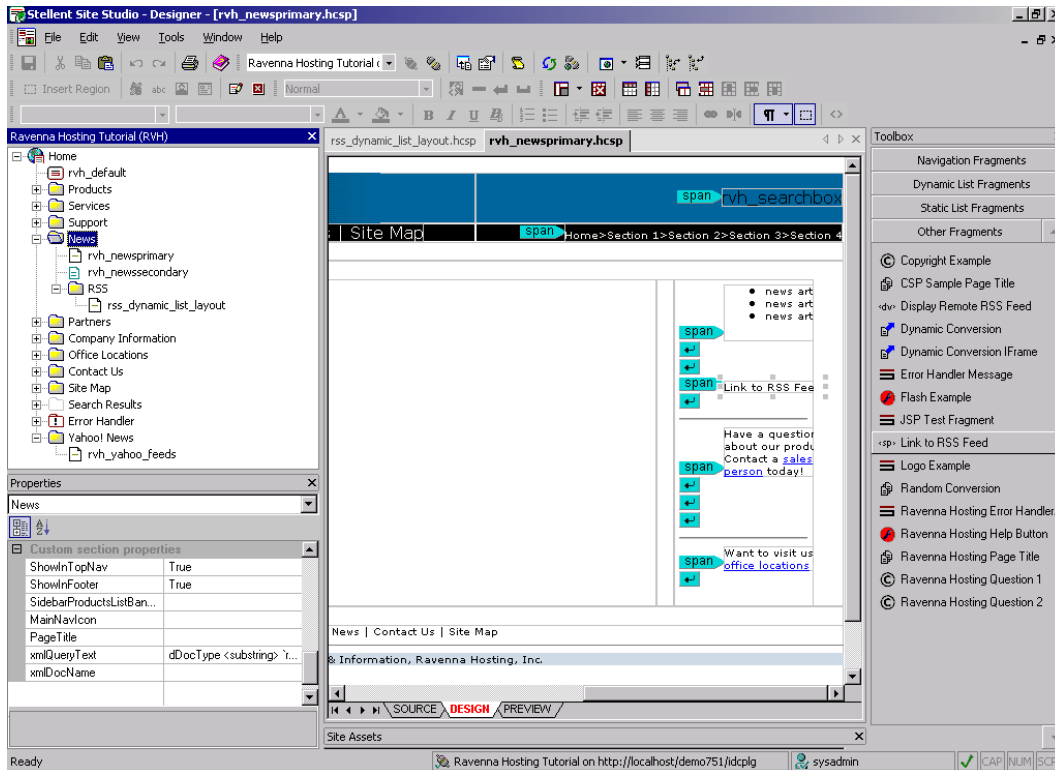
Check them in with the Content Names **rss_dynamic_list_layout**, **rss_static_list_layout**, and **rss_office_loc_static_list** respectively. This can also be done from the Site Studio Designer when you create the specific RSS sections.

RSS for Dynamic Lists

The Ravenna Hosting tutorial has several Dynamic Lists to display content. The **News** page is a good example of a page that should have an RSS feed. This will allow you to keep up-to-date with the news articles. To add a RSS feed, follow the steps below:

1. Create a new child section to the **News** section, and name it **RSS**
2. Set the primary page for the RSS section to be the layout **rss_dynamic_list_layout**.
3. Click the **News** section again, then select **Tools > Define Custom Section Properties**.

4. Create a new property named **xmlQueryText**. You can make this a **text** field, or a **managedquery** field.
5. Set this property to the same query that is run on the **News** page. It should look something like this:
 - `dDocType <substring> `rvh_newsstory``
6. Open the primary page for the **News** section, which should be `rvh_newsprimary`

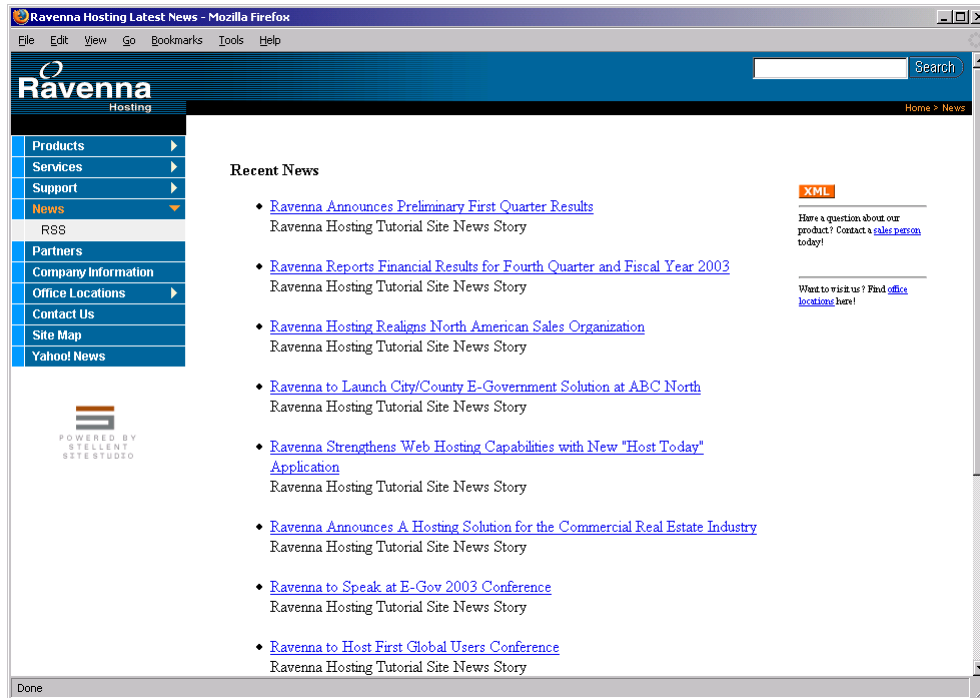


7. On the right-hand side of the page, add the fragment **Link to RSS Feed**, as in the figure below
8. Save all pages, and republish the navigation.

After adding the RSS feed, the Ravenna News page now has an orange XML link on the right hand side. Clicking on that link will take you to the RSS section, which displays the results in RSS format.

This RSS feed is dynamically generated based on the query of the parent section. When a new news article is posted on the Ravenna Hosting Tutorial site, the RSS feed is automatically updated.

Instead of creating RSS sub-sections for every page, you can also have one since RSS section for the entire web site. See the Developer's Reference section below for more details.



RSS for Static Lists

You can create a RSS feeds for a static lists as well as dynamic lists. The process is very similar. The primary differences are in the primary page for the RSS section, and the name of the custom section property.

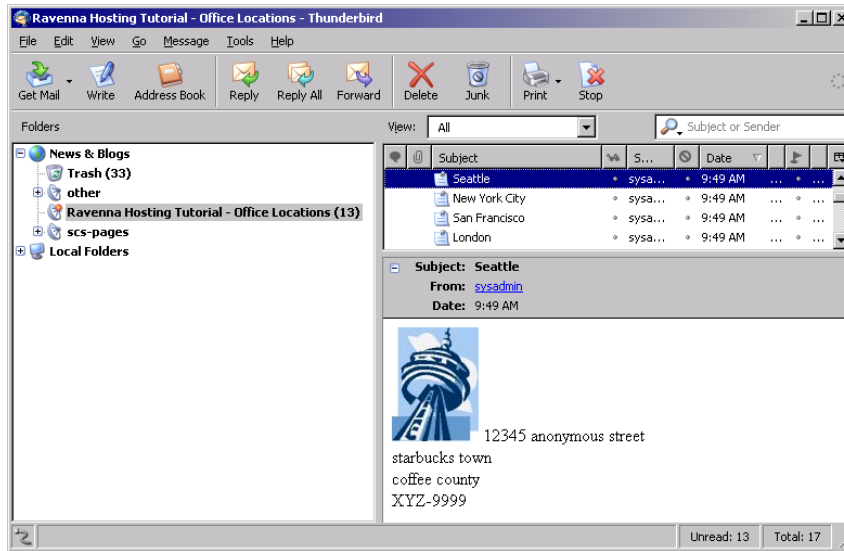
In the Ravenna Hosting Tutorial, the web page Office Locations contains a static list of offices. To create an RSS feed to this page, follow the steps below:

1. Create a new child section to the **Office Locations** section, and name it **RSS**
2. Set the primary page for the RSS section to be the layout **rss_static_list_layout**.
3. Click the **Office Locations** section again, then select **Tools > Define Custom Section Properties**.
4. Create a new property named **xmlDocName**. You can make this a **text** field, or a **manageddoc** field.
5. Set this property to the Content Name of the contribution region on the Office Locations page. In the tutorial, the contribution region is assigned to the item **rvh_officelocations**.
6. Open the primary page for the Office Locations section, which should be **rvh_officelocationsprimary**
7. On the right-hand side of the page, add the fragment **Link to RSS Feed**, similar to the example above.
8. Save all pages, and republish the navigation.

Unlike a dynamic list, there is no universal way to map a static list into a RSS feed. Each static list is a specific XML file with its own structure and schema. The template **rss_static_list_layout** is only a very basic example of how you can turn a static list into a feed.

A better example is the **rss_office_loc_static_list** template. This one is specialized to the Office Locations static list for the Ravenna Hosting Tutorial. The XML nodes named **guid** and

title are very important for RSS feeds, so that the readers know when the feed items have been updated. This template makes certain to set them properly for this specific static list. Additionally, the description field contains the image associated with the entry, in addition to the description:

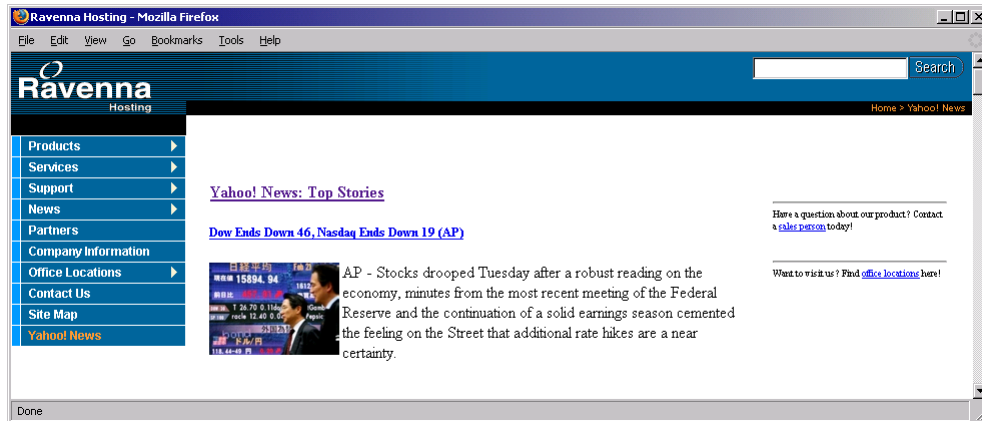


You will almost certainly need to create a custom layout for a static list. Use the [rss_static_list_layout](#) page for a starting point, and the [rss_office_loc_static_list](#) page as an example of what else is possible.

Displaying Remote RSS Feeds

Another use of RSS feeds is to display content from other sites on yours. Since RSS data can be parsed easily, it is usually a simple task to download a remote feed, format it, and display it on your page. This example will create a page on the Ravenna Hosting Tutorial to download and render a feed from Yahoo! News.

1. Create a new Section to the web site, named **Yahoo! News**.
2. Select a primary page for this section. Create a new layout based on the **rvh_default** template.
3. Click on the center of the page in Design View.
4. Insert the Display Remote RSS Feed fragment by clicking on its button on the right toolbar.
5. When prompted, type in the URL to a RSS feed. In this case, enter this URL:
 - `http://rss.news.yahoo.com/rss/topstories`
6. Click OK
7. Republish the navigation



When you navigate to the Yahoo! News link on your site, you should see a HTML representation of the Yahoo! RSS feed. Edit the Display Remote RSS Feed fragment to change how the HTML is displayed for the feed. You may also specify a XSL stylesheets to use to render the HTML.

Content Server Feeds

You can use the same tools described above to generate feeds for Content Server pages, or render remote RSS pages as HTML. Samples for this are included in the directory:

stellent/custom/RSSFeeds/samples/

Adding a RSS Feed to a Page

This sample comes with RSS Feeds for search results, workflow in queue, personal URLs, and checked-out content. If you have a custom page that displays results like these, you should be able to leverage the same includes to generate your RSS feeds.

If you need to add an RSS feed to a different kind of page, for example collaboration projects, folders, or records management pages, you will need to create custom includes. See the include **search_results_rss_feed** for a sample.

If you would like to add another feed to the **My RSS Feeds** page so that it is available for everybody, override the include **my_productivity_rss_feeds**, or **global_rss_search_feeds**. Also override the **opml_rss_links** include if you wish to have your links appear in the OPML file.

For maximum forwards compatibility, it is advisable for you to create a separate component for your modified includes. That way, when you install a newer version of this sample, your modifications will not be lost.

Displaying Remote RSS Feeds

You can use the **GET_REMOTE_FEED** service to download a RSS feed from another server. You can then use JSP or IdocScript to render the results. For example, to display the Yahoo! News feed on a template page, use the following IdocScript:

```

<${feedUrl}="http://rss.news.yahoo.com/rss/topstories",
  parseXml="true"$>
<${executeService("GET_REMOTE_FEED")}$>
<a href="<${channelLink$}>"><h1><${channelTitle$}></h1></a>
<${loop RSS_ENTRIES$>
  <h3><a href="<${link$}>"><${title$}></a></h3>
  <${description$>
<${endloop$>

```

If you prefer XSL to IdocScript or JSP, you can also use the **rssTransform** function with an XSL to render the HTML:

```

<${isAbsoluteWeb=1,
  feedUrl="http://rss.news.yahoo.com/rss/topstories",
  xslUrl= HttpWebRoot & "resources/xsl/yahoo-news-rss.xsl"$>
<${htmlOutput = rssTransform(feedUrl, xslUrl)$>
<${xmlDecode(htmlOutput)$>

```

The yahoo-news-rss.xsl ships with this sample. You can modify it as needed.

Developer's Reference

This section contains information about the tools provided by this sample that allow developers to create new RSS Feeds.

Configuration Settings

These settings will affect the behavior of how RSS feeds are published and processed. Typically, these must be manually placed into the config.cfg file for the Content Server.

RSS Feed Configuration Settings

Name	Default	Description
ExtraLogWriterPath	none	This flag enables the Content Server to output its logs in RSS format. Set it to rssfeeds.RssLogWriter to allow RSS feeds of the Content Server logs. Since logging begins very early on, this flag can only be set in the config.cfg for the server.
RssLogWriterLogLevel	1	Controls which errors are logged to the RSS Feeds. If set to 0, all errors will be logged. When set to 3, only fatal errors will be logged.
RssLogWriterNumEntries	30	The maximum number of error messages in the RSS Feed.

Name	Default	Description
RssFeedMaxAge	600	Maximum age of a remote RSS feed, in seconds. This value is used if a max age is not passed in the GET_REMOTE_FEED service, or the xslTransform IdocScript function.
RssDateFormat	EEE, dd MMM yyyy HH:mm:ss	The date format used for RSS feeds. Some RSS readers require special formatting. However, the default should work most of the time. Defaults to:
UseSSLForRSS	false	Available on the Stellent Support site: http://www.stellent.com/support/

IdocScript Functions

These new functions allow developers to consume remote RSS feeds, and display the results on a JSP or HCSP page. Also note that the service **GET_REMOTE_FEED** is scriptable, so it can be called with the standard IdocScript function **executeService**.

rssTransform(urlToRSS, urlToXSL [, feedMaxAge])

This function downloads a RSS feed from another web site, renders it with an XSL stylesheet. The results are cached in the Content Server to boost performance.

urlToRSS

This is the full URL to an RSS feed.

urlToXSL

This is the full URL to an XSL stylesheet.

feedMaxAge

This is the amount of time to cache the results of the XSL transformation. If not passed, it will default to the **RssFeedMaxAge** configuration setting.

Results

Returns a HTML string which can be displayed on the page, or processed further.

xmlDecode(xmlString [, beginIndex [, endIndex]])

This function is the opposite of the standard IdocScript function `xml`. It decodes an XML encoded string, so it can be displayed on a HTML page. This is sometimes needed in RSS feeds that contain HTML content.

xmlString

The XML-encoded string to decode.

beginIndex

The place to start decoding in the `xmlString`. If not specified, it defaults to the beginning of the string.

endIndex

The place to stop decoding in the `xmlString`. If not specified, it will stop at the last character in the string.

Results

The decoded XML string. An example is shown below:

```
<$string = "<p>", xmlString = xml(string)$>
<$xmlString$>    <!-- outputs "&lt;p&gt;" -->
<$xmlDecode(xmlString)$>    <!-- output "<p>" -->
```

Site Studio Fragments

These fragments are included in the fragment library located here:

```
stellent/custom/RSSFeeds/samples/Site Studio/rssfeeds_fragment_lib.zip
```

Upload this fragment library using Site Studio Designer before use.

Display Remote RSS Feed

This fragment uses two fragment parameters to download a remote RSS feed, and render it into HTML.

feedUrl

This is the full URL to an RSS feed. Required parameter.

urlXSL

This is the full URL to an XSL stylesheet. This is optional. If not specified, the RSS data is displayed with IdocScript in the fragment.

Results

Outputs a HTML representation of the remote RSS feed. The data is cached locally in the Content Server to boost performance.

Link To RSS Feed

This fragment creates a link to a RSS feed for the current page. The current section must have a child section named RSS. The RSS section must also have a primary page based on one of the RSS templates included with this sample.

Site Studio Layouts

Since a RSS feed must be pure XML, you need specially formatted Site Studio layouts to generate them. Two are included with this sample, one for dynamic lists, and another for static lists. They are located here:

```
stellent/custom/RSSFeeds/samples/Site Studio/rss_dynamic_list_layout.hcsp
stellent/custom/RSSFeeds/samples/Site Studio/rss_static_list_layout.hcsp
stellent/custom/RSSFeeds/samples/Site Studio/rss_office_loc_static_list.hcsp
```

To use them, create a new section, and use these templates as the primary page.

rss_dynamic_list_layout

This layout renders a dynamic list based on the custom section property **xmlQueryText**. This property can be set on the page, or on this page's immediate parent. If the property is not set on either page, then one of the other parameters below must be set.

sourceId

This is the **nodeId** for a Site Studio page to use to generate the RSS feed. The corresponding page must have the custom section property **xmlQueryText** set.

xmlQueryText

As a last resort, you can pass a full **xmlQueryText** to this page to run any query on the fly. This is discouraged, since it can lead to extremely long URLs.

Results

A RSS formatted XML page, displaying the results of the dynamic list.

rss_static_list_layout

This layout renders a static list based on the custom section property **xmlDocName**. This property can be set on the page, or on this page's immediate parent. If the property is not set on either page, then one of the other parameters below must be set. The **xmlDocName** should be set to the Content Name of the static list for the page. This is set when assigning region content for a page.

sourceId

This is the **nodeId** for a Site Studio page to use to generate the RSS feed. The corresponding page must have the custom section property **xmlDocName** set.

xmlDocName

As a last resort, you can pass a Content Name to this page to load any static list on the fly.

Results

A RSS formatted XML page, displaying the results of the static list.

rss_office_loc_static_list

This is a specialization of the `rss_static_list_layout` for the Ravenna Hosting Tutorial Office Locations page. Instead of a direct mapping from a static list to a RSS feed, this layout sets the **guid** and **title** elements to values specific to this static list. It also creates a richer **description** element that contains both text and graphics. It can be used as an example of how to specialize a RSS feed for your specific static list.

Results

A RSS formatted XML page, displaying the results of the Office Locations page.

Custom Services

These new Content Server services are available for people.

GET_REMOTE_FEED

This service downloads a public RSS feed from a web server. It will return either the raw RSS data, or it will parse it. After parsing, it can be displayed on a page with IdocScript or JSP. This service is scriptable, so it can be run with the `executeService` IdocScript function:

```
<!--$feedUrl="http://rss.news.yahoo.com/rss/topstories", parseXml = "true"-->
<!--$executeService("GET_REMOTE_FEED")-->
<a href="<!--$channelLink-->"><h1><!--$channelTitle--></h1></a>
<!--$loop RSS_ENTRIES-->
  <h3><a href="<!--$link-->"><!--$title--></a></h3>
  <!--$description-->
<!--$endloop-->
```

The remote RSS feed is cached locally inside the Content Server for a configurable period of time. By default, this is 10 minutes. See the **RssFeedMaxAge** configuration flag for more information. You can adjust the age of the cache on the fly by setting the **feedMaxAge** parameter in the request.

feedUrl

This is the full URL to an RSS feed.

parseXml

Defaults to false. This will cause the entire RSS feed to be returned. This is the full URL to an XSL stylesheet.

feedMaxAge

This is the amount of time to cache the results of the RSS feed. If not specified, it will default to the **RssFeedMaxAge** configuration setting.

Results

Returns the RSS data from the feed. If the feed is not parsed, the results are available in the parameter **rawXmlData**. If the RSS data is parsed, then the results are available in the Result Set named **RSS_ENTRIES**. Additional RSS data is available in the parameters **channelTitle**, **channelLink**, **channelDescription**, and others.

GET_SCS_FEED

This service renders the **STELLENT_RSS_FEED** template. This template will render different includes based on the parameters in the request.

feedName

This is the name of the feed to use. Currently supported feed names are **search_results**, **checked_out_list**, **workflow_queue**, and **personal_urls**. Each feed name corresponds to a resource include that executes a service call, and renders the results in RSS. For example, the **search_results** feed renders the include **search_results_rss_feed**, and caches the results.

feedMaxAge

This is the amount of time to cache the results of the RSS feed. If not specified, it will default to the **RssFeedMaxAge** configuration setting.

rssCacheScope

This defaults to **session.**, which means that the cache is specific to the user who rendered it. If set to **application**, all users will use the same cache when downloading this feed. This is mainly useful for public RSS feeds generated from searches.

Results

Returns the RSS data from the specific feed.