Rhythmyx FOP Assembler

Introduction

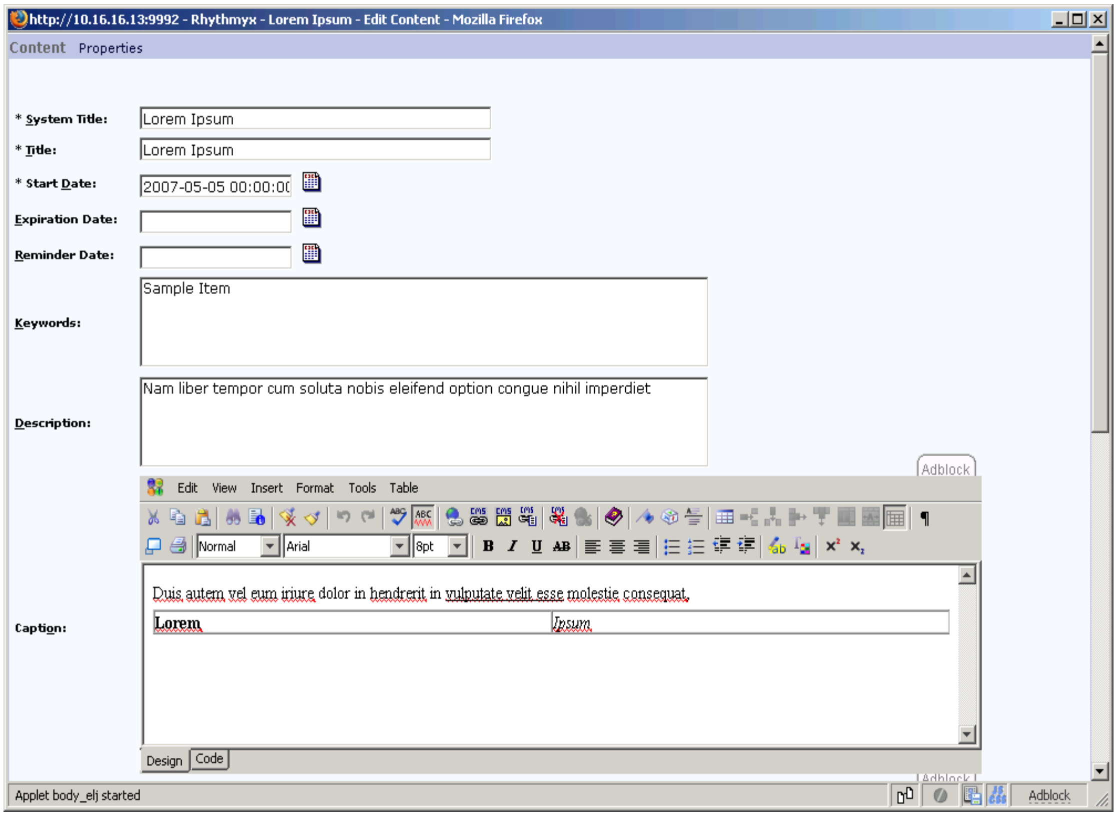
This document describes usage and installation the Rhythmyx FOP Assembler.[1]

The FOP Assembler is used to assemble content into PDF, Postscript and other common printable formats, using the Apache FOP processor. The user can create templates that take full advantage of XSL FO without having to write (or debug) large amounts of XSLT.

Contributors use normal Rhythmyx content types, and enter their content using the normal editor controls. Implementers define one or more FOP Templates that produce the proper output format for that particular content type. Since these are ordinary Rhythmyx templates, multiple templates can be created with different formats as required.

Example

We have built a simple example using the Rhythmyx *Fast Forward* content types. The contributor enters the content item using the Generic Content Editor



http://10.16.16.13:9992 - Rhythmyx - Lorem Ipsum - Edit Content - Mozilla Firefox

* System Title: Lorem Ipsum

* Title: Lorem Ipsum

* Start Date: 2007-05-05 00:00:0(

Expiration Date:

Reminder Date:

Keywords: Sample Item

Description: Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet

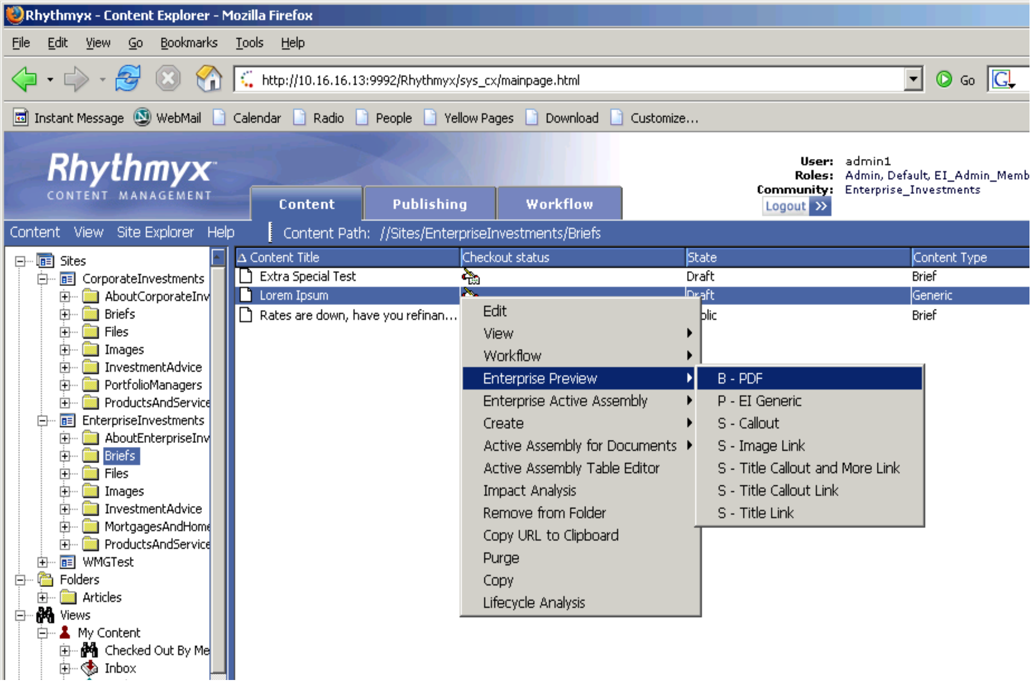
Caption: Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat.

Lorem	*Ipsum*

Applet body_elj started

[1] The FOP Assembler is a PSO Toolkit component. This means that it will be freely distributed to Rhythmyx customers, but will not be formally supported. See license.txt for details and restrictions.

Using the Content Explorer, the user can then preview the content item with the PDF template.



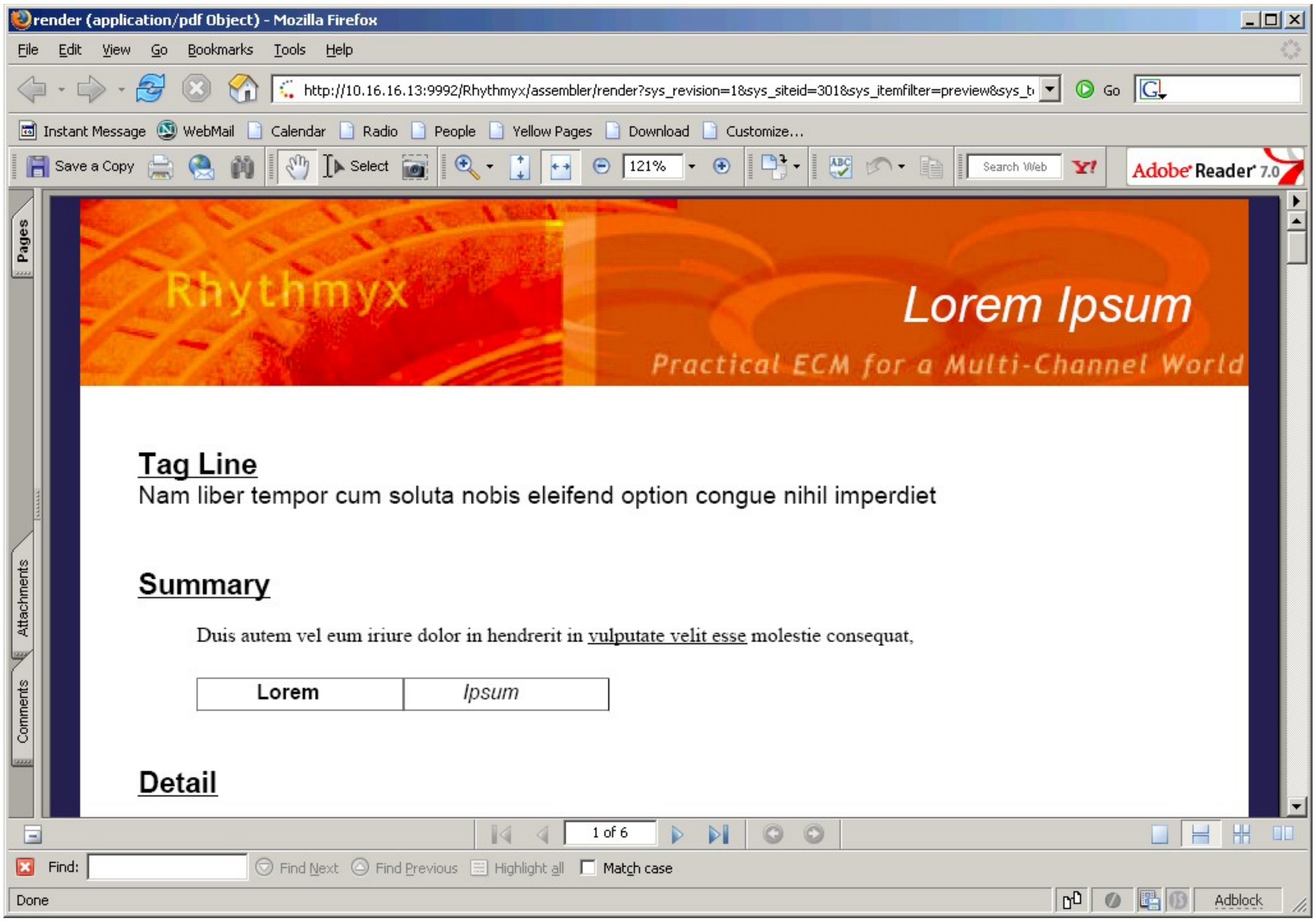
http://10.16.16.13:9992/Rhythmyx/sys_cx/mainpage.html

User: admin1
Roles: Admin, Default, EI_Admin_Memb
Community: Enterprise_Investments

Content Path: //Sites/EnterpriseInvestments/Briefs

Content Title	Checkout status	State	Content Type
Extra Special Test		Draft	Brief
Lorem Ipsum		Draft	Generic
Rates are down, have you refinan...		blic	Brief

This results in a PDF displaying on the screen.


render (application/pdf Object) - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
Go
Instant Message WebMail Calendar Radio People Yellow Pages Download Customize...
Save a Copy
Select
121%
Search Web
Adobe Reader 7.0
Pages
Attachments
Comments
Rhythmyx
Lorem Ipsum
Practical ECM for a Multi-Channel World
1 of 6
Find:
Find Next
Find Previous
Highlight all
Match case
Done
Adblock

http://10.16.16.13:9992/Rhythmyx/assembler/render?sys_revision=1&sys_siteid=301&sys_itemfilter=preview&sys_t

Tag Line

Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet

Summary

Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat,

Lorem	*Ipsum*

Detail

Similarly, this template can be published by including it in a content list in the normal fashion.

Installation

Prerequisites

The FOP Assembler requires Rhythmyx version 6.1 or later.

In addition, to enable processing of body fields, the PSOToolkit6 is required. This should be build 27 or higher. Note that the current versions PSOToolkit6 now require Rhythmyx 6.1 patch RX11247 or later (the "Item Filter" patch). If you do not have this patch, we strongly recommend that you install it (or a later patch) before deploying the new toolkit.

The PSOToolkit can be obtained from Tech Support.

You should also have the Java 1.5 JDK and Apache Ant installed and configured on the server. The deployment script depends on these components. Execute following commands:

```
>java -version
java version "1.5.0_05"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0_05-b05)
Java HotSpot(TM) Client VM (build 1.5.0_05-b05, mixed mode, sharing)
>ant -version
Apache Ant version 1.5.3 compiled on April 16 2003
```

Deployment

Unzip the distribution into a temporary directory (do **not** use the Rhythmyx home directory). Set an environment variable named RHYTHMYX_HOME. This should point to the root of the Rhythmyx installation on your server.

Run the deployment script with the following command:

```
      ant -f deploy.xml
```

This will copy the JAR files and install the configuration file. It will also register the FOP Assembler extension.

If you intend to install the Sample template, you should deploy it at this time:

```
      ant -f deploy.xml deploySamples
```

This will copy the images and XSL stylesheet used by the sample template to the correct directories in the server.

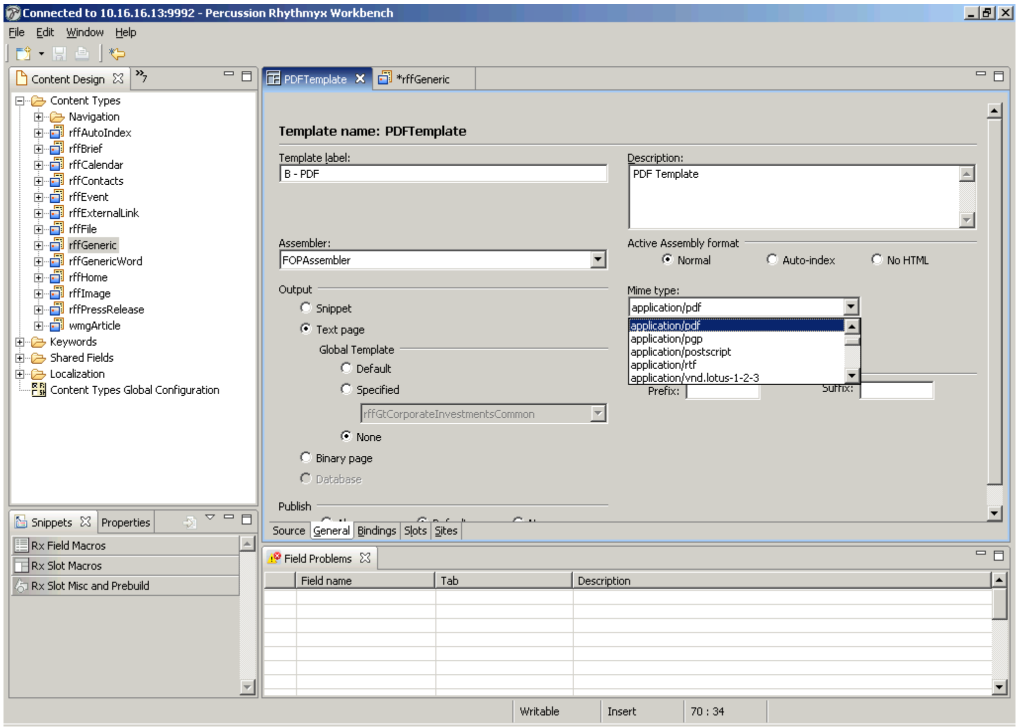
Adding MIME Types to the Workbench

The FOP Assembler as deployed by the deployment script supports the following MIME types:

```
application/pdf
application/postscript
application/x-pcl
text/plain
```

```
text/richtext
text/rtf
application/rtf
```

The standard Workbench installation does not include all of these types in the "MIME Type" drop down control.



Connected to 10.16.16.13:9992 - Percussion Rhythmyx Workbench

Template name: PDFTemplate

Template label: B - PDF

Description: PDF Template

Assembler: FOPAssembler

Active Assembly format: ● Normal ○ Auto-index ○ No HTML

Output: ○ Snippet ● Text page

Global Template: ○ Default ○ Specified (rffGtCorporateInvestmentsCommon) ● None

○ Binary page ○ Database

Mime type: application/pdf

There are 2 files in the Workbench installation that need to be modified. Both of them are named `mimemapwb.properties`. The "Default" properties are located in the Workbench plugin folder:

```
\Rhythmyxl\eclipse\plugins\com.percussion.workbench_1.0.0\default-config\rxconfig\Workbench
```

The "User" properties are found in the Eclipse Workspace folder.

\Rhythmyx\rxconfig\Workbench\users\<username>\workspaces\workspace\.metadata\.plugins\com.percussion.workbench.ui\rxconfig\Workbench

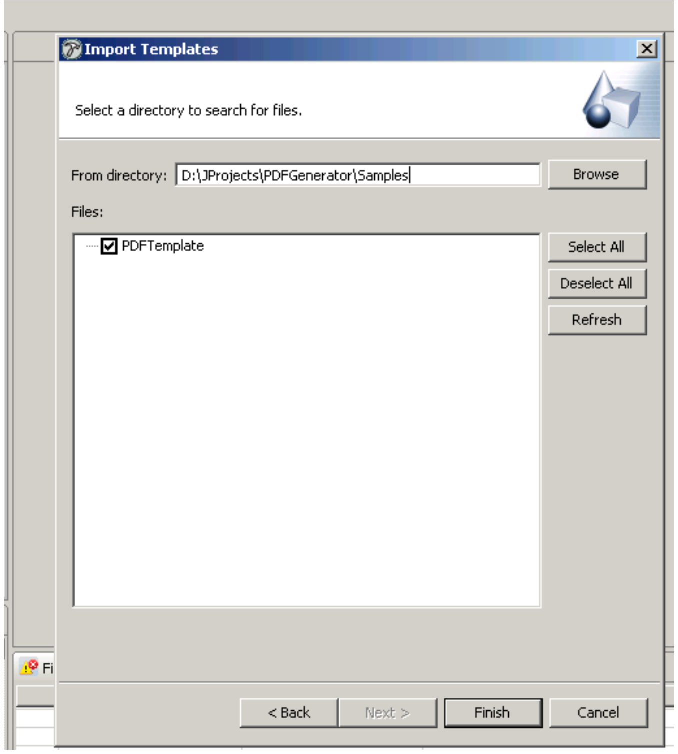
Shut down the Workbench if it is running and edit the files with your favorite text editor. In both files, add the following lines to the edit of the file:

```
pdf=application/pdf
eps=application/postscript
pcl=application/x-pcl
rtf=application/rtf
rtf2=text/rtf
```

Restart the Workbench and you should see the new types added to the MIME type dropdown.

Importing the Sample Template

The Sample Template is included in the distribution zip file. Open the Workbench and use the “File – Import” menu to import Templates. Use the browse button to locate the directory where you unpacked the zip file. (Note that if you are using 2 different machines, this is relative to the workbench, not the Server)



Import Templates

Select a directory to search for files.

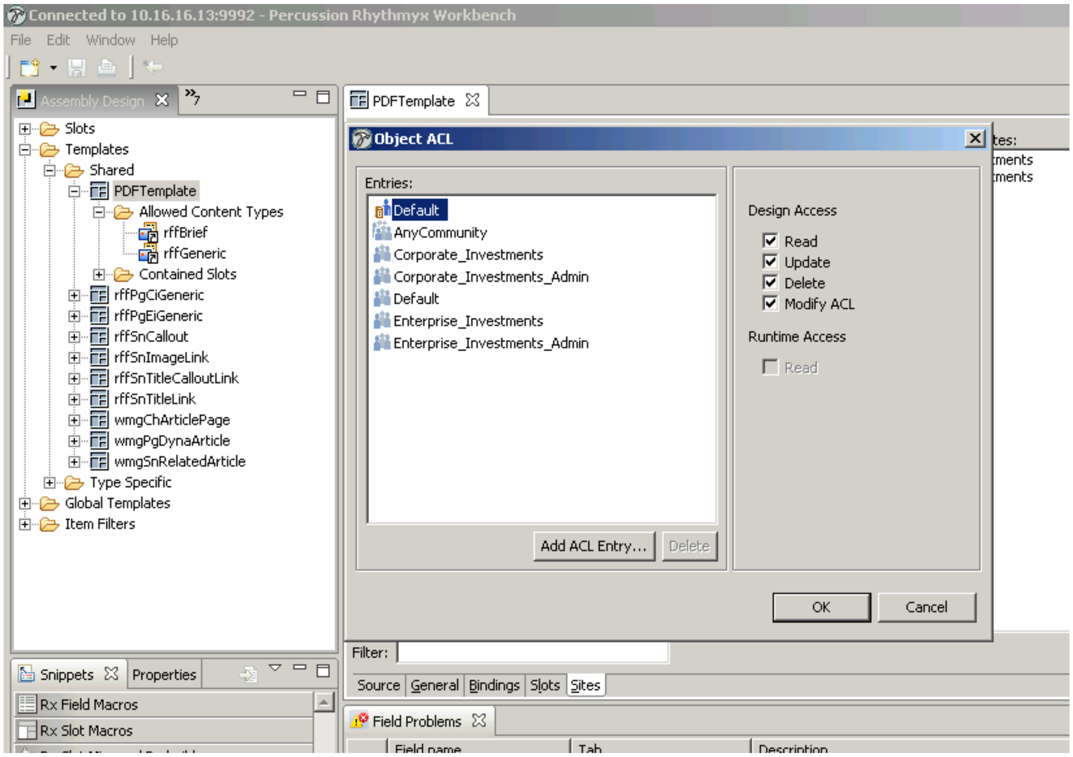
From directory: D:\JProjects\PDFGenerator\Samples

Files:

☑ PDFTemplate

Select the “PDFTemplate” entry. Once you press the “Finish” button, the template will appear in the Assembly Design view.

You must add your new template to the appropriate Communities. Right click on the template and open the Security dialog



Entries:

- Default
- AnyCommunity
- Corporate_Investments
- Corporate_Investments_Admin
- Default
- Enterprise_Investments
- Enterprise_Investments_Admin

Design Access

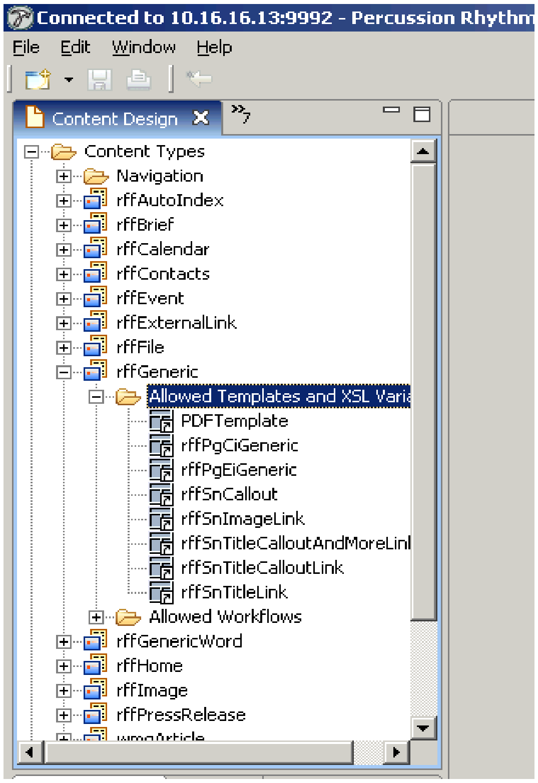
☑ Read
☑ Update
☑ Delete
☑ Modify ACL

Runtime Access

□ Read

Add any communities, and check the “Read” box.

You must also add the template to the “Allowed Templates” of any Content Types (in this example rffGeneric). In the Assembly design view, Select “PDFTemplate” and “Copy” it (with ctrl-C). Switch to the Content Design view, and expand “Generic”. Click on the “Allowed Templates and XSL Variants” and Paste. This will add PDF Template to the list of allowed templates.



You are now ready to use the Sample template.

Extending the Sample Template

Changing the MIME Type

The sample template is configured to supply PDF output. You can change this by selecting a different type (such as application/postcript) and then previewing a Generic content item. Save the output to disk and open it in your favorite text editor.

Template Source

Examine the template source. You will notice that it returns XML using the FO namespace. Apache and the W3C maintain extensive references and tutorials on FO. We are using FOP version 0.93, but expect that the FOP Assembler will be upward compatible with future versions.

Notice that Static Images can be referenced in the template. Place your images in the rx_resources folder.

```
      <fo:static-content flow-name="xsl-region-before">
            <fo:block>
                  <fo:external-graphic src="rx_resources/images/PDFmarginTop.jpg"
content-width="20cm"/>
            </fo:block>
      </fo:static-content>
```

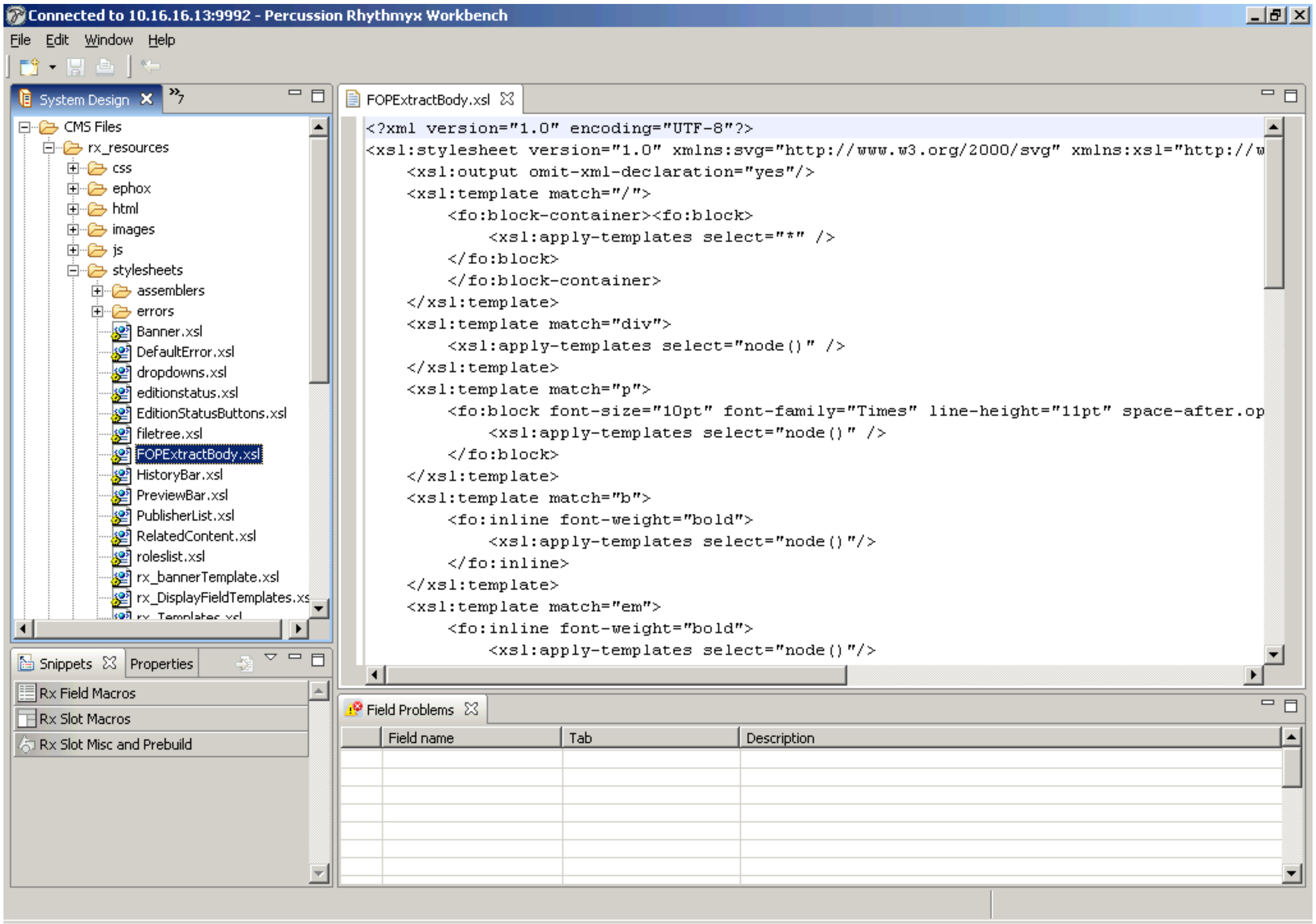
Bindings

Examine the bindings of the Sample template. You will notice that both the “callout” and “body” fields are processed with psoTransform. If your content types have additional fields, you can add them to the bindings. Give each new field a unique name.

Editing XSL Stylesheets

The XSL Stylesheet provided with the sample supports simple formatting (bold, italics, underline), as well as tables and images. It can be extended to include other types of processing.

You can edit the XSL directly within the Workbench. Go to the System Design view, expand “CMS Files” and then “rx_resources” and then “stylesheets”. By default this will open in the Text editor, but you can configure your workbench to use any editor for XSL files using the Eclipse Preferences menu.



```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:svg="http://www.w3.org/2000/svg" xmlns:xsl="http://w
    <xsl:output omit-xml-declaration="yes"/>
    <xsl:template match="/">
        <fo:block-container><fo:block>
            <xsl:apply-templates select="*" />
        </fo:block>
        </fo:block-container>
    </xsl:template>
    <xsl:template match="div">
        <xsl:apply-templates select="node()" />
    </xsl:template>
    <xsl:template match="p">
        <fo:block font-size="10pt" font-family="Times" line-height="11pt" space-after.op
            <xsl:apply-templates select="node()" />
        </fo:block>
    </xsl:template>
    <xsl:template match="b">
        <fo:inline font-weight="bold">
            <xsl:apply-templates select="node()"/>
        </fo:inline>
    </xsl:template>
    <xsl:template match="em">
        <fo:inline font-weight="bold">
            <xsl:apply-templates select="node()"/>
```

References