

# Document generation in IBM® Rational Publishing Engine 2.0 beta

Many organizations must produce documentation from their systems and software engineering data subject to complex style and format requirements imposed by different groups such as clients, business partners, or government or industry regulatory bodies. With Rational® Publishing Engine 2.0 , everyday users can easily produce these types of documents using pre-configured reports, with no coding knowledge or additional configuration required.

Two roles are usually involved in creating a report: the report designer and the everyday user. In this example, Ryan the report designer creates reports from templates that he designed in Document Studio. Susan the everyday user generates the documents from the reports that Ryan developed. Ryan and Susan's manager wants a daily digest of the top 25 headlines from various news sites. Ryan will create the templates and reports and Susan will generate the documents.

## Tip: What's the difference between templates, reports, and documents?

A *template* is a resource that a report designer creates in Rational Publishing Engine Document Studio using data source schemas, variables and styles. Templates have the file extension .dta.

A *report* is created from one or more templates or stylesheets. Templates correspond to sections of a report. For example, you might have different templates for table of contents, front cover, body, index, and back cover. These templates can be reused in different documents.

A *document* is generated from a report. Users can find reports that the report designer created and then generate documents from a single click. Documents can be in Microsoft Word, HTML, PDF or XSL-FO formats.

- [Report designer: Designing reports from templates](#)
- [Everyday user: Generating documents](#)

## Report designer: Designing reports from templates

This scenario focuses on the role of Ryan the report designer. Report designers upload templates that they have already created in Document Studio into the Rational Publishing Engine web application, where they can assemble the templates into a report that any user on their team can generate a document from.

### Before you begin

In this example, Ryan creates two templates (.dta files) in Document Studio. The first template is a cover page and the second template produces the first 25 entries of an RSS 2.0 feed. He has also designed a stylesheet (.css file) to apply to the HTML output of this document.

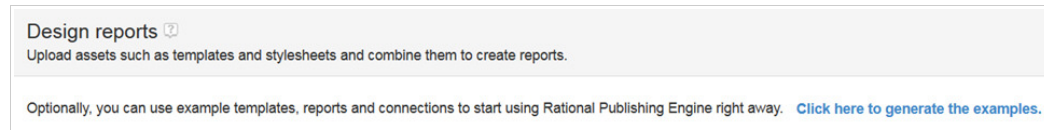
### About this task

Although it is possible to create one large template with multiple sections, Ryan prefers to create smaller templates for each section. That way he can reuse the templates and quickly assemble them into different reports whenever his manager asks him for a new document. For example, he might have templates for the front cover, table of contents, introduction, body, index and back cover.

### Procedure

1. Go to the Design page of the Rational Publishing Engine web application .
2. First Ryan must upload the templates that he created into the list of templates.

**Tip:** You can import sample templates to work with by clicking the **Click here to generate the examples** link.



- a. Click **Upload Asset**.
- b. In the Name field, enter *News feed*.
- c. Accept the default template type.
- d. Browse to select the file.
- e. Click **Upload** .
- f. Repeat the above steps to add the template for the front cover.

The RSS feed template and the front cover template are added to the list.

3. Next Ryan wants to add a style sheet to work with.

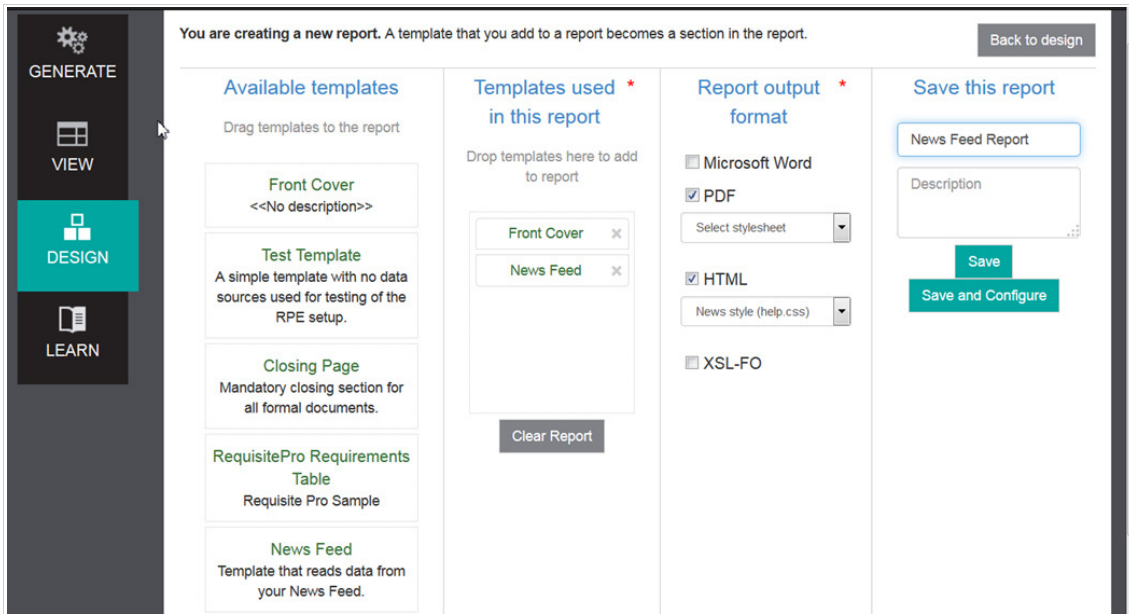
- a. Click **Upload Asset**.
- b. In the Name field, enter *News style*.
- c. In the **Type** field, select **Stylesheet**.
- d. Browse to select the file.
- e. Click **Upload** .

4. Now Ryan can assemble the templates into a report. For the beta, it is important to put the templates in the order that you want them to appear in the finished document. Click **Create Report**.

**Tip:** The order of templates in the **Templates used in this report** section is the order that the templates appear in the report. Once you have dragged the templates into the report, you can change the order. You can also add the same template multiple times in one report.

- a. Drag the Front Cover template from the Available templates list to the Templates used in this report list.
- b. Repeat the above step to add the News Feed template to the Templates used in this report list.

5. Ryan's manager has asked him to provide both HTML and PDF output, so Ryan selects these options.
6. Ryan wants to apply the style sheet to the HTML output, so he selects News Style under the **HTML** check box.
7. Ryan names the report *News Feed Report* and saves the report.



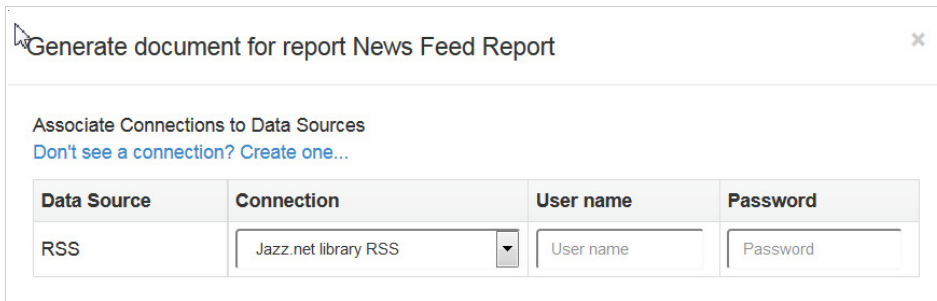
Ryan's News Feed Report is added to the list of reports. If Ryan wanted to edit additional properties for the report, he would click **Save and Configure**.

8. Ryan wants to quickly test the document before he makes the report available to his team.

- Click **Back to design** to return to the Design page.
- Click the Reports tab on the Design page.
- In the Actions column of the list of reports, select **Preview Document**.
- Ryan is prompted to create a data source connection. He wants to use an RSS feed from jazz.net and clicks the **Don't see a connection? Create one** link.
  - Enter a title and description for the feed.
  - In the URL field, add the URL to the data source. This example uses RSS feeds so Ryan enters `https://jazz.net/library/rss/`.
  - This particular data source is publicly accessible so Ryan leaves the authentication fields blank.
  - Save the connection.

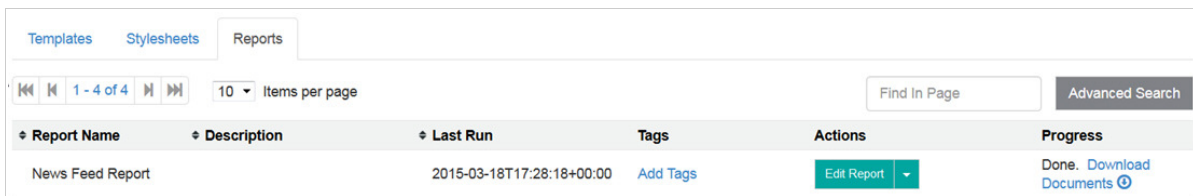
**Note:** Ryan could have also defined the data source connections before he created any reports.

e. In the **Connection** list, Ryan selects the Jazz.net connection that he just created and clicks **Preview** to preview the document.



**Preview** will take the first 10 records from each query and **Generate** creates the document with all the records from each query.

f. When the job finishes, click the **Download Documents** link.



g. Ryan is prompted to open or save the zip file. He opens it to check the contents.

h. He is satisfied with the results of the generated preview document. Now he will share the report with his team.

## Everyday user: Generating documents

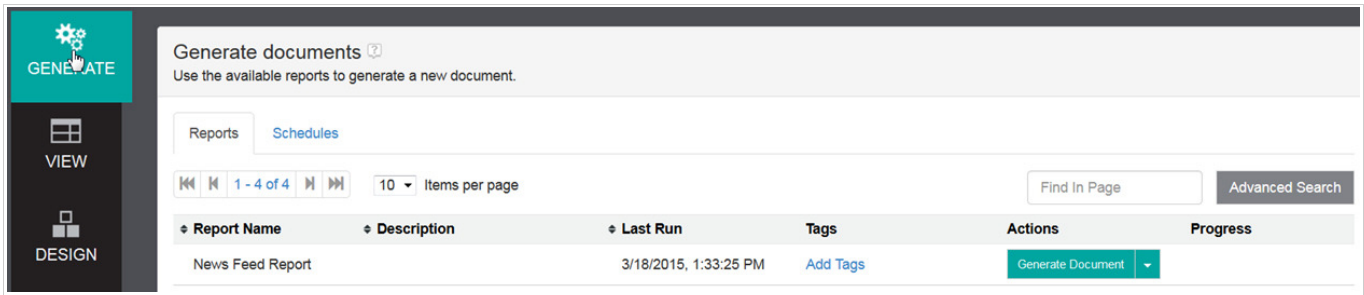
This scenario focuses on the role of Susan the everyday user. She can generate documents with minimal steps without any knowledge of document design, templates, or data source configurations. She uses the reports that Ryan the report designer created to generate her documents.

### About this task

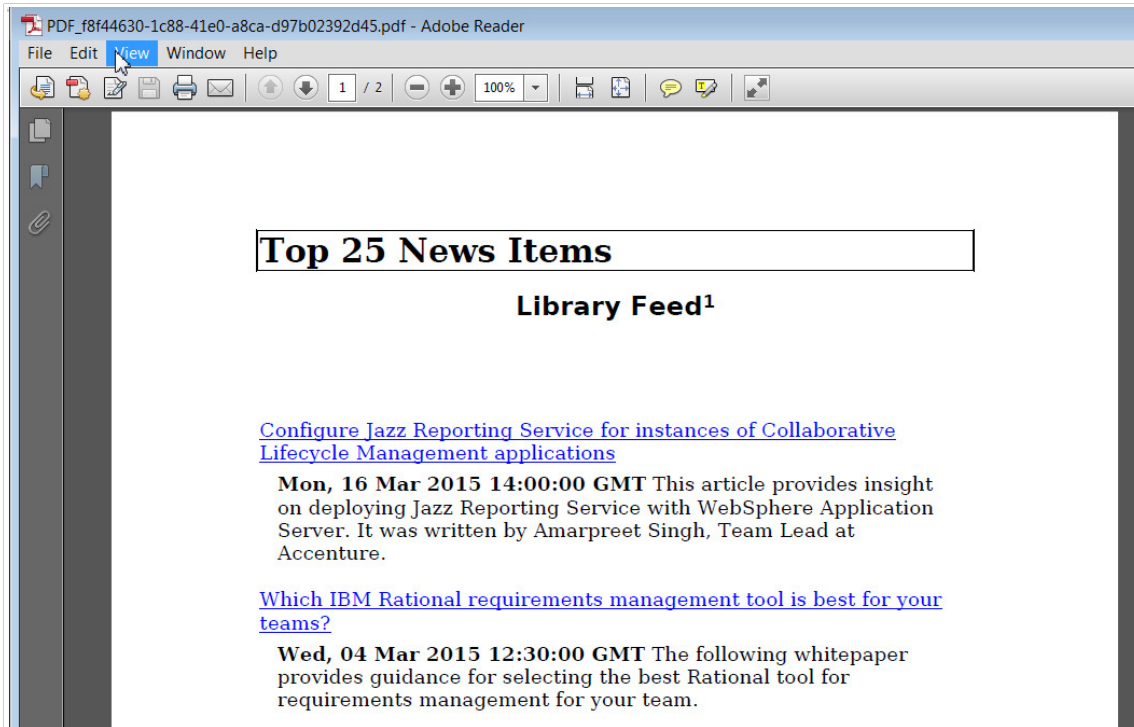
Susan sees the reports that Ryan has made available and wants to generate the News Feed report.

### Procedure

- Susan opens the Generate page of the Rational Publishing Engine web application .
- She finds the report that she wants by entering `News` in the search field.
- Susan clicks **Generate Document** in the Actions column.

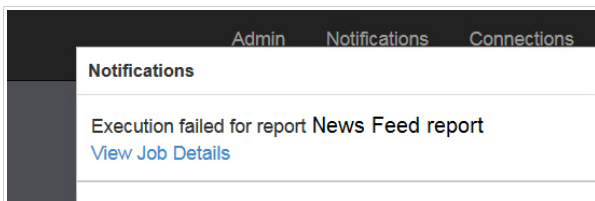


4. When prompted for the data source connection, she selects the Jazz.net RSS connection that the report designer created and then clicks **Generate**.
5. When the document generation job is complete, she goes to the View page of the Rational Publishing Engine web application.
6. She clicks **Download Documents** in the Actions column.
7. Susan is prompted to open or save the zip file. She opens the zip file and sees the generated HTML file, PDF file and log.zip.
8. Susan opens the PDF file to verify that the document was generated properly.



**Note:** Susan can also download individual files instead of the entire zip file. For example, from the Actions column she can click the down arrow and select **Download PDF**.

9. Susan clicks the **Add Tags** link in the Tags column and enters `news` so that she can easily find the report later.
10. Susan's manager has requested that this document be run every day at 10 AM. Susan schedules the document generation:
  - a. Go to the Generate page.
  - b. Find the report that you want.
  - c. In the Actions column, click the down arrow and select **Generate Later**.
  - d. Enter the name and description of the job schedule.
  - e. Set the frequency to **Daily** and use the calendar to select the start and end dates of the job.
  - f. Click **Schedule**.
 Susan clicks the Schedule tab to see that the schedule was added.
11. The next day, Susan logs in to the Rational Publishing Engine web application. She sees a Notification in the product banner. When she clicks the Notification link she sees that a recent job failed.



12. She wants to see the high level summary of the document generation job and clicks **View Job Details**

### Job details for report News Feed Report

Event time	Event description
2015-02-11 03:02:51	Job created with ID 42201
2015-02-11 03:02:51	Calling DGaaS service. The URL is https://rpe.stage1.mybluemix.net/dgaas
2015-02-11 03:02:51	Call to DGaaS service succeeded
2015-02-11 03:02:52	Starting run...
2015-02-11 03:02:52	Processing document template 1 of 2.
2015-02-11 03:02:53	3 publishing commands are processed. 0 publishing commands remain to be processed.
2015-02-11 03:02:53	Processing document template 2 of 2.

13. Now Susan wants to see the recent history of documents that she generated. She opens the View page and sees that the most recent job completed successfully.

The screenshot shows a web application interface with a sidebar on the left containing 'GENERATE', 'VIEW', 'DESIGN', and 'LEARN' buttons. The main content area is titled 'View generated documents' and includes a search bar and a table of generated reports.

Report Name	Description	Run at	Status	Tags	Actions
News Feed Report		3/18/2015, 1:42:42 PM	FINISHED	news	Download Documents
News Feed Report		3/18/2015, 1:33:25 PM	FINISHED	Add Tags	Download Documents
News Feed Report (deleted)		3/18/2015, 1:18:45 PM	FINISHED	Add Tags	Download Documents

#### Results

Now that Susan has generated the daily digest document that she needs, she can pass the document to her manager.

## Limitations

### Limitations of the Rational Publishing Engine web application

The Rational Publishing Engine web application supports data sources that use REST API or XML data, for example Rational solution for CLM. Rational DOORS® 9 will not be supported.

Copyright © 2015 IBM Corporation