

Question and Answer Style Process Builder

Abstract

This tech note enhances the [basic guidance on this topic](#) provided in the Rational Method Composer (RMC) online help with detailed guidance and an example.

1.0 Advantages to using a question and answer style process builder

A Question and Answer style process builder allows a set of questions to determine what configuration to build. The questions and answers can guide a project manager or process engineer to create an appropriate starting configuration for his or her project, which can then be further customized using the Team Process perspective.

The answers are remembered in the configuration, so if the library changes, you can refresh the configuration by executing "Run stored queries" on the configuration.

2.0 Considerations and limitations

Process builder cannot vary the questions asked based on prior answers. Only a fixed set of questions are currently supported.

For each question, a set of answers can be defined. The user can select any number of the answers – there is no way to limit the responses to a single answer.

The IBM Practices library does not automatically populate a work breakdown structure as a result of selecting practices. It is possible to set up a standard work breakdown structure that is automatically populated as a result of process builder, but this is outside the scope of this article.

The configuration resulting from running process builder can be further customized using the team process perspective. See [Technote #7027352 - Creating a Team Process](#).

3.0 A starter template

The following XML can be used as a template for creating a "question and answer" style process builder.

```
<?xml version="1.0" encoding="UTF-8"?>
<wizards xmlns="http://www.ibm.com/rmc/library/xmldef">
  <wizard name="{some name}" id="{some id}" style="qna">
    <initial_selections>
      <selection>...</selection>
    </initial_selections>
    <page>
      <!-- this is where the questions and answer page is -->
    </page>
    <views>
      <query>...</query>
    </views>
  </wizard>
</wizards>
```

To use this template, copy the contents of this template into a file of type "xmldef", and save this file in the configuration directory (for a standard library), or a configuration project folder (for a workspace-based library). Then make the following changes:

1. Change "{some name}" to the name of your process builder wizard throughout the template, for example, you could change it to "Company ABC Process Wizard for IT" if creating a process builder to select a process for an IT project.
2. Change "{some id}" to a unique id, such as "com.abc.rmc.processbuilder.infotech".
3. Add one or more <selection> tags inside the <initial_selections> tag to define the plugins or elements that you'd like to be initially selected before going through the pages of questions and answers.
4. Add one or more <page>tags to define the questions and answers that you want to include in this process builder. Details on <page> tags is discussed in the next section.
5. Specify views
To specify the views to publish, use the RMC authoring perspective to edit each of the categories you wish to publish as views. Mark with a tag such as "abc_for_it_view", and then add a tag-based query to the <views> XML tag.

For example:

```
<views>
  <query>
    <where>
      <condition name="tag" value="abc_for_it_view"/>
    </where>
  </query>
</views>
```

In this example, all custom categories tagged as "abc_for_it_view" will publish as views.

3.0 The <page> tag

You define the questions and answers inside <page> tags. The following is an outline of the <page> tag.

```
<page name="{page name}" id="{page id}" display_condition="{TRUE|FALSE}">
  <question name="{question name}" id="{question id}" display_condition="{TRUE|FALSE}">
    <description>{some description of the question}</description>
    <answer name="{answer name}" id="{answer id}" display_condition="{TRUE|FALSE}">
      <description>{some description of this answer}</description>
      <query>
        <where>
          <group type="{AND|OR}">
            <group>
              <condition name="{PLUGIN|TAG|TYPE}"
                operator="{IS|IS NOT}" value="{text string}"/>
            </group>
            ...
            <group>...</group>
          </group>
        </where>
      </query>
      <action>{ADD|SUBTRACT}</action>
    </answer>
    ...
    <answer>...</answer>
  </question>
  ...
  <question>...</question>
</page>
```

You can have many questions inside a page, and you can have many answers for one question. Each page, question, or answer must have a name, a unique id, and display_condition. If display_condition is set to True, then<<<what happens>>>

Question and answer can have <description> tags where you can provide more information about the question or answer. Each answer have <query> tags and <action> tags.

The <query> tags specify the selection (plugins or elements) when user chooses this answer. Inside the <query> tags, you can use the <group> tags to specify Boolean expressions to do complex selections. Ultimately the <condition> tag specifies what to select. The text string in the condition tags can include "*" for wild card expressions. You can use backslash '\' to escape special characters as needed.

The <action> tags identify whether to add or subtract the selections of this answer which is defined by the <query> tags.

4.0 Example process builder

This example QnA process builder lets a user select from alternative requirements practices, either user-story or use-case based practices. One practice, "Iterative Development" is selected automatically. This is a simplification of the situation common in many companies where some practices are mandatory, but there are some categories where the project can make a choice between alternative approaches.

4.1 Defining the process builder wizard

Using the starter template, we created the example QnA process builder. We've given it a name "Example QnA Style Process Builder", which we will use to identify the process builder when we run it in the process builder view in RMC.

The first thing to add the mandatory practices. In this case, the initial selection includes all of the plugins in core (core content is shared across practices and is published only if used by a practice), and the mandatory iterative development practice.

```
<initial_selections>
  <selection id="selection.01">
    <action>add</action>
    <query>
      <where>
        <group type="OR">
          <group>
            <condition name="plugin" operator="is"
              value="/core.*"/>
            <condition name="type" value="MethodPlugin"/>
          </group>
          <group>
            <condition name="plugin" operator="is"
              value="/practice.mgmt.iterative_dev.*"/>
          </group>
        </group>
      </where>
    </query>
  </selection>
</initial_selections>
```

```

        <condition name="type" value="MethodPlugin"/>
    </group>
</group>
</where>
</query>
</selection>

</initial_selections>

```

Next, we add pages to present the selection choices. There is one question "How would you like to capture requirements for the project?" in this page. A description of this question is also given. Two answers are provided for this question: 1) By Use Cases, and 2) By User Stories. The <query> tags define the selection of use case driven development plugins for the first answer. And another <query> tags define the selection of the user story driven development plugins for the second answer. Both answers will add the selection as specified by their <action> tags.

```

<page name="Development Preferences - Requirements" id="example.pb.page1"
display_condition="true">
    <question name="How would you like to capture requirements for the project? (Select one of
the following)" id="example.pb.page1.question1" display_condition="true">
        <description>
            <i>Requirements can be captured using different entities describing the system's
functionality, user behaviors, and active roles involved. Both user stories and
use cases are common entities that can be used to capture requirements.
Please identify which would be better for your project.</i>
        </description>
        <answer name="By Use Cases" id="example.pb.page1.question1.answer1"
display_condition="true">
            <description>
                <i>Requirements are captured as Use Cases. Use cases are preferable if a
detailed requirements specification is needed.</i>
            </description>
            <action>add</action>
            <query>
                <where>
                    <group>
                        <condition name="plugin" operator="is"
value="/practice.tech.use_case_driven_dev.*"/>
                        <condition name="type" operator="is"
value="MethodPlugin"/>
                    </group>
                </where>
            </query>
        </answer>

```

```

<answer name="By User Stories" id="example.pb.page1.question1.answer2"
display_condition="true">
  <description>
    Requirements are captured as User Stories. User stories tend to be
    relatively informal and incomplete, leaving details to be resolved during
    implementation. They work well when a complete, detailed requirements
    specification is not required or desired.
  </description>
  <action>add</action>
  <query>
    <where>
      <group>
        <condition name="plugin" operator="is"
value="/practice.tech.user_story_driven_dev.*"/>
        <condition name="type" operator="is"
value="MethodPlugin"/>
      </group>
    </where>
  </query>
</answer>
</question>
</page>

```

To finish the process builder, we select a view. For simplicity, we just used the default IBM view which is already tagged as "nav_wizard_view".

```

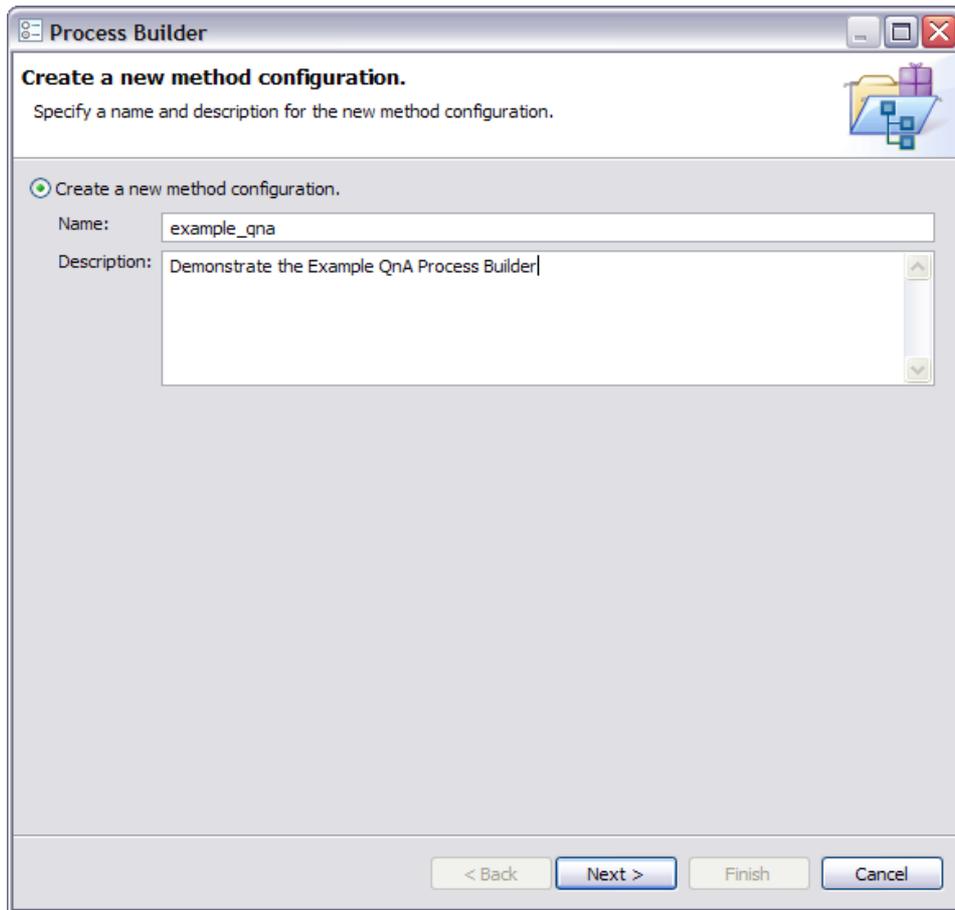
<views>
  <query>
    <where>
      <condition name="tag" value="nav_wizard_view"/>
    </where>
  </query>
</views>

```

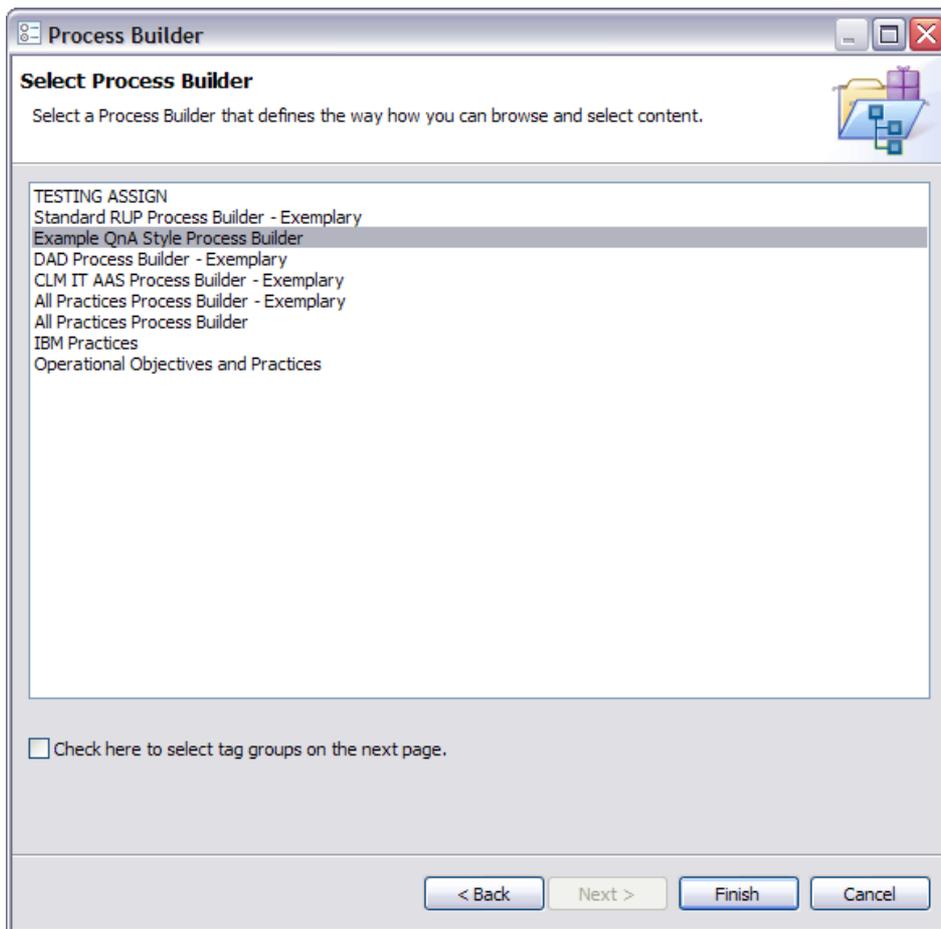
4.2 Running the process builder wizard

Run the example process builder as follows.

1. Go to the Process Builder Perspective in RMC.
2. Click on "Build a new Method Configuration".
3. Type in Name and Description for the configuration. This example has the name "example_qna" and a description. Click Next.

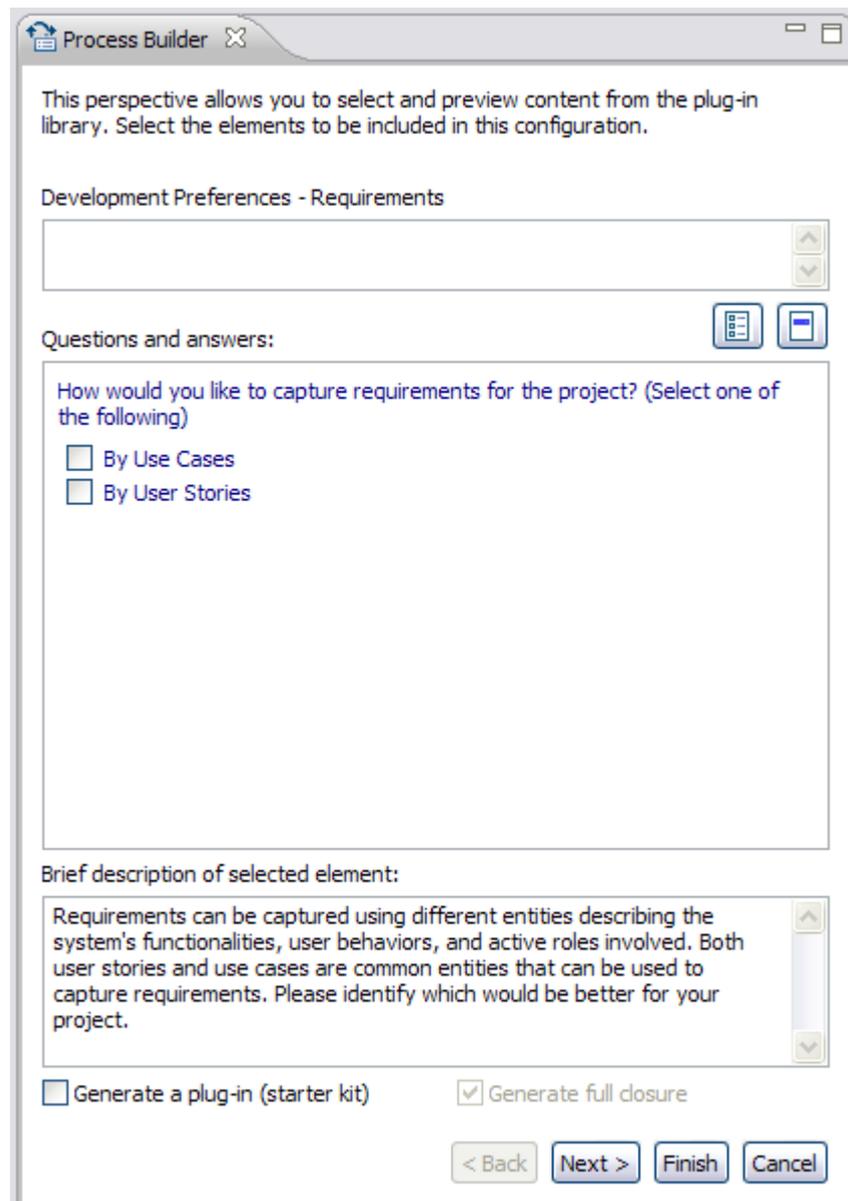


4. Then, select the process builder wizard that you defined earlier. In this example, we will select "Example QnA Style Process Builder" (the name of the process builder in the <wizard> tag). Then click Finish.



5. The first page of the process builder is displayed. Note that it may take some time to display the first page, depending on the numbers of plugins selected in the initial selection.

The questions and answers are displayed in blue text, and you can click on the text to see the associated brief description. After clicking on the text, you can also see the elements that are added by this answer by clicking on the "Preview highlighted element" and then browsing the configuration view at the bottom.

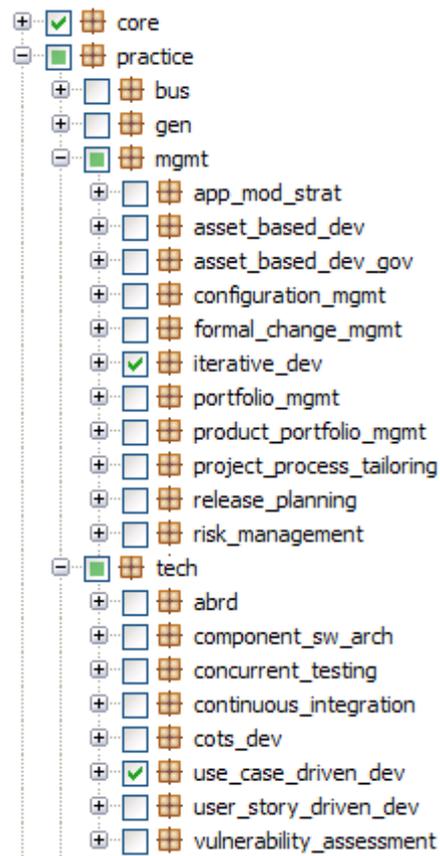


Choose your answer by clicking on the checkbox. Note that the question asks for a single answer, but there is no way to enforce this.

6. To see the results of your selections so far, you can click on the "Preview configuration" button and browse the configuration view at the bottom.
7. If you have defined additional pages with additional questions, click Next to display the next page and answer additional questions.
8. When you are finished answering all the questions, click Finish to generate the configuration.

To view the results, go back to the Authoring Perspective, and double click on the configuration to edit it. Click the Plug-in and Package selection tab to review the plugins selected based on the answers you made in the process builder.

Note that only plug-in and package selections will be visible. The syntax supports element level selections that are not visible.



5.0 More information

The syntax for selections is covered in detail in the technote “Defining a configuration using XML”.