BIRT Extensible Data Explorer View Spec

Author: Yuejie Chen, Qiangsheng Wang

1. Introduction

The BIRT tress view provides a tree view to show the user of report design outline, data sources, data sets and report parameters. The spec defines the new structure that can help user to customize and extends all the tree views in BIRT perspective.

The requirement includes:

- **Define the data source and data set as extension items.** The data related feature of BIRT are defined as extension items at API level. The data source type definition and data access features are part of the DTP project. Theoretically BIRT GUI should be working all right without the data related features. This spec try to separate the data access related feature into a new plug-in.
- **Customize the tree structure.** This spec provides the way let user customize the tree structure of this view. User can add new node or provides new data set builder and etc.
- **No directly depends on data plug-in.** By the changes, which define in this spec, the designer doesn't explicit data APIs any more.
- **Supporting other extended item of BIRT.**

2. APIs

2.1 Overview

We really don't change any API of current implementation. The idea is to change the way that the user register the INodeProvider for every node on the tree.

Current implementation depends on a hard code NodeProviderFactory, but this spec defines an extension point. By using this extension point, user gets the flexible to customize the existing behavior or contribute new type of node.

2.2 Extension point definition

This extension point is really a general mechanism to mapping a class type to an adapter class. Ideally, all kind of service register can be defined by this definition. Currently BIRT uses it to mapping the model element type to a specified node provider class.

2.2.1 Extension point schema

This is a brief schema of adaptable node. To get the full schema definition, please visit the Appendix A.

```
<element name="adaptable">
    <attribute name="class" type="string" use="required" />
</element>
```

By using the, user can define the class, which wants to adapt.

This is a brief schema of adapter node. To get the full schema definition, please visit the Appendix A.

```
<element name="adapter">
  <attribute name="id" type="string" use="required" />
```

The id defines the unique identity of this element.

```
  <attribute name="type" type="string" use="required" />
```

The type defines the type of Class attribute.

```
  <attribute name="class" type="string" />
```

The class defines the Class name, which adapter to the adaptable class.

```
  <attribute name="factory" type="string" />
  <attribute name="singleton" type="boolean" />
  <attribute name="priority" type="string" />
</element>
```

2.2.2 An example

```
<extension point="org.eclipse.birt.report.designer.ui.elementAdapters">
        <adaptable class="org.eclipse.birt.report.model.api.ReportDesignHandle">
        <adapter
                class="org.eclipse.birt.report.designer.internal.ui.views.outline.providers.ReportDesignNodeProvider"
                id="ReportDesign.ReportDesignNode"
                priority="2"
                type="org.eclipse.birt.report.designer.ui.views.INodeProvider" />
          </adaptable>
</extension point>
```

3. Appendix

A. Extension point schema

```
<?xml version='1.0' encoding='UTF-8'?>
<!-- Schema file written by PDE -->
<schema targetNamespace="org.eclipse.birt.report.designer.ui.views">
<annotation>
    <appInfo>
```

```
      <meta.schema plugin="org.eclipse.birt.report.designer.ui.views" id="elementAdapters"
name="Element Adatpers Extension"/>
   </appInfo>
   <documentation>
      [Enter description of this extension point.]
   </documentation>
  </annotation>

  <include schemaLocation="schema://org.eclipse.core.expressions/schema/expressionLanguage.exsd"/>

  <element name="extension">
   <annotation>
      <documentation>
        (no description available)
      </documentation>
   </annotation>
   <complexType>
      <sequence>
        <element ref="adaptable" minOccurs="1" maxOccurs="unbounded"/>
      </sequence>
      <attribute name="point" type="string" use="required">
        <annotation>
          <documentation>


          </documentation>
        </annotation>
      </attribute>
      <attribute name="id" type="string">
        <annotation>
          <documentation>


          </documentation>
        </annotation>
      </attribute>
      <attribute name="name" type="string">
        <annotation>
          <documentation>
```

```
            </documentation>
            <appInfo>
              <meta.attribute translatable="true"/>
            </appInfo>
          </annotation>
        </attribute>
      </complexType>
    </element>

    <element name="adaptable">
      <annotation>
        <appInfo>
          <meta.element labelAttribute="class"/>
        </appInfo>
        <documentation>
          (no description available)
        </documentation>
      </annotation>
      <complexType>
        <sequence>
          <element ref="adapter" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
        <attribute name="class" type="string" use="required">
          <annotation>
            <documentation>

            </documentation>
            <appInfo>
              <meta.attribute kind="java"/>
            </appInfo>
          </annotation>
        </attribute>
      </complexType>
    </element>
```

```
<element name="adapter">
  <annotation>
    <appInfo>
      <meta.element labelAttribute="ID"/>
    </appInfo>
    <documentation>
      (no description available)
    </documentation>
  </annotation>
  <complexType>
    <sequence minOccurs="0" maxOccurs="1">
      <element ref="enablement"/>
    </sequence>
    <attribute name="id" type="string" use="required">
      <annotation>
        <documentation>
          (no description available)
        </documentation>
      </annotation>
    </attribute>
    <attribute name="type" type="string" use="required">
      <annotation>
        <documentation>
          (no description available)
        </documentation>
        <appInfo>
          <meta.attribute kind="java"/>
        </appInfo>
      </annotation>
    </attribute>
    <attribute name="class" type="string">
      <annotation>
        <documentation>
          (no description available)
        </documentation>
        <appInfo>
```

```
      <meta.attribute kind="java"/>
    </appInfo>
  </annotation>
</attribute>
<attribute name="factory" type="string">
  <annotation>
    <documentation>
      (no description available)
    </documentation>
    <appInfo>
      <meta.attribute kind="java"
            basedOn="org.eclipse.core.runtime.IAdapterFactory"/>
    </appInfo>
  </annotation>
</attribute>
<attribute name="singleton" type="boolean">
  <annotation>
    <documentation>
      (no description available)
    </documentation>
  </annotation>
</attribute>
<attribute name="priority" type="string">
  <annotation>
    <documentation>
      (no description available)
    </documentation>
  </annotation>
</attribute>
<attribute name="overwrite" type="string">
  <annotation>
    <documentation>
      (no description available)
    </documentation>
  </annotation>
</attribute>
```

```
  </complexType>
</element>

<annotation>
  <appInfo>
    <meta.section type="since"/>
  </appInfo>
  <documentation>
    [Enter the first release in which this extension point appears.]
  </documentation>
</annotation>

<annotation>
  <appInfo>
    <meta.section type="examples"/>
  </appInfo>
  <documentation>
    [Enter extension point usage example here.]
  </documentation>
</annotation>

<annotation>
  <appInfo>
    <meta.section type="apiInfo"/>
  </appInfo>
  <documentation>
    [Enter API information here.]
  </documentation>
</annotation>

<annotation>
  <appInfo>
    <meta.section type="implementation"/>
  </appInfo>
  <documentation>
    [Enter information about supplied implementation of this extension point.]
```

```
    </documentation>
  </annotation>

  <annotation>
    <appInfo>
      <meta.section type="copyright"/>
    </appInfo>
    <documentation>

    </documentation>
  </annotation>
</schema>
```