BIRT API Change Control Document

1. Introduction

This is an enhancement for BIRT to use the new syntax for expressions. It is requested by Bugzilla 191445. In short, after the change, each expression value consists of two parts: the raw value and its type. So that, other scripting languages can be supported.

API changes are in org.eclipse.birt.report.model plug-ins. More specified, all changes are in org.eclipse.birt.report.model.api package or its sub-packages.

2. Changed APIs:

2.1 Name = getAllTocs, getAllBookmarks

Component name = ReportDesignHandle

Package name = org.eclipse.report.model.api

Change Request:

Since bookmark and TOC are in the expression type, the return values become expression lists. Previously, the return values are string lists.

3. Added APIs:

3.1 Name = Expression

Component name = Expression

Package name = org.eclipse.report.model.api

Change Request:

Add a new class to represent the expression data:

```
 /**
  * Represents the object for the expression. The expression has an
    expression value and its type.
 */
```

Class Expression

{

```
/**
 * Returns the raw value if the type is not constant. If the type is
 * "constant", return the value.
 *
 * @return the raw expression or the value
 */
      public Object getExpression( )


/**
 * Returns the raw expression if the type is not constant. If the type
 * is "constant", get the value in string format.
 *
 * @return the raw expression or the value in string
 */
      public String getStringExpression( )


/**
 * Return the type of the expression. It can be one of defined in
 * <code>ExpressionType</code>. For the compatibility issue, in default,
 * it is <code>ExpressionType.JAVASCRIPT</code>.
 *
 * @see ExpressionType
 *
 * @return the type
 */
      public String getType( )


/**
 * Return the type of the expression set by the user. This method ignore
 * the compatibility issue.
 *
 */
```

```
        public String getUserDefinedType( )
}
```

Proposed Solution:

Uses them like this:

Expression expression = (Expression) element.getProperty(exprPropName);

Hence, such codes must be revised:

(String) element.getProperty(propName);

3.2 Name = ExpressionHandle

Component name = ExpressionHandle

Package name = org.eclipse.report.model.api

Change Request:

Add a new class to represent the expression:

Class ExpressionHandle

{

```
/**
 * Return the raw expression if the type is not constant. If the type is
 * constant, returns the value.
 *
 * @return the raw expression
 */

public Object getExpression( )

/**
 * Sets the raw expression if the type is not constant. If the type is
 * constant, sets the value.
 *
 * @param expr
 *            the raw expression or the value
 * @throws SemanticException
 *
 */

public void setExpression( Object expr ) throws SemanticException

/**
 * Return the type of the expression.
 *
 * @return the expression type
 */

public String getType( )
```

```
/**
 * Sets the type of the expression.
 *
 * @param type
 *            the expression type.
 * @throws SemanticException
 *
 */

public void setType( String type ) throws SemanticException

/**
 * Return the expression in string format.
 * <p>
 * <ul>
 * <li>if the type is not constant, return the raw expression;
 * <li>if the type is constant, return the value in string.
 * </ul>
 *
 * @return the raw expression or the value in string
 */

public String getStringExpression( )

}
```

Uses them like this:

ExpressionHandle exprHandle = element.getExpressionProperty(exprPropName);

3.3 Name = ExpressionType

Component name = ExpressionType

Package name = org.eclipse.report.model.api

```
public static final String CONSTANT = "constant";

public static final String JAVASCRIPT = "javascript";


/**
 * Gets possible types for the expression.
 *
 * @return an iterator to enumerate all possible types.
 */

public static Iterator<String> typeIterator( )
```

3.4 Name = DesignElementHandle.getExpressionProperty/setExpressionProperty

Component name = DesignElementHandle

Package name = org.eclipse.report.model.api

```
/**
 * Sets the value of a property to an expression.
 *
 * @param propName
 *            the property name
 * @param expression
 *            the value to set
 * @throws SemanticException
 */
```

public void setExpressionProperty(String propName, Expression expression) **throws** SemanticException

```
/**
 * Returns a handle to work with an expression property. Returns null if
 * the given property is not defined or cannot be set with expression
 * value.
 * @param propName
 *            name of the property.
 * @return a corresponding ExpressionHandle to with with the expression
 *         property.
 *
 */

public ExpressionHandle getExpressionProperty( String propName )
```

3.5 Name = StructureHandle.getExpressionProperty/setExpressionProperty

Component name = StructureHandle

Package name = org.eclipse.report.model.api

```
/**
 * Sets the value of the member as an expression.
 *
 * @param memberName
 *            name of the member to set.
 * @param value
 *            the expression to set
 * @throws SemanticException
 *             if the member name is not defined on the structure or the
 *             value is not valid for the member.
 */

public void setExprssionProperty( String memberName, Expression value )
            throws SemanticException

/**
 * Gets the value of the member as an expression.
 *
 * @param memberName
 *            name of the member to set.
 * @return the expression
 * @throws SemanticException
 *             if the member name is not defined on the structure or the
 *             value is not valid for the member.
 */

public ExpressionHandle getExprssionProperty( String memberName )
```

3.6 Name = Structure.getExpressionProperty/setExpressionProperty

Component name = Structure

Package name = org.eclipse.report.model.api.elements.structures

New methods apply to all classes in org.eclipse.birt.report.model.api.elements.structures.

```
/**
 * Sets the value of the member as an expression.
 *
 * @param memberName
 *            name of the member to set.
 * @param value
 *            the expression to set
 * @throws SemanticException
 *             if the member name is not defined on the structure or the
 *             value is not valid for the member.
 */

public void setExprssionProperty( String memberName, Expression value )

/**
 * Gets the value of the member as an expression.
 *
 * @param memberName
 *            name of the member to set.
 * @return the expression
 * @throws SemanticException
 *             if the member name is not defined on the structure or the
 *             value is not valid for the member.
 */

public Expression getExprssionProperty( String memberName )
```

4. Removed APIs:

N.A.

5. Miscellaneous Change Requests

N.A.