

Enabling Binding Binary and CLOB Data to Report Controls

Design Specification

Draft 1: Sep 20, 2005

Abstract

This is the design specification of BIRT 2.0 Project BPS03: Enabling Binding Binary/Clob Data to Report Controls.

Document Revisions

Version	Author	Date	Description of Changes
Draft 1	Gary Xue	Sept 20, 2005	First draft



Contents

1. Introduction	3
2. Features and Requirements	3
2.1 Open Data Access Framework Support for BLOB and CLOB	3
2.2 Binding Binary Column to BIRT Image Control	3
2.3 Binding Clob Columns to BIRT Controls	3
2.4 Out-of-the-Box Support for JDBC BINARY/BLOB/CLOB Data Types.....	3
2.5 Samples for Binary Data Binding.....	3
3. Open Data Access Framework Support.....	3
3.1 New ODA Data Types	3
3.2 API Changes.....	4
4. Using Binary Data in BIRT Report	4
4.1 Binary Output of Data Sets	4
4.2 Binding Binary Data to Image Control	4
4.3 BIRT Scripting with Binary Data.....	5
4.4 Report Parameters	5
5. Using CLOB Data in BIRT Report.....	5
6. ODA-JDBC Driver	6
7. Model API Changes.....	6
7.1 New Data Type Constant.....	6
8. Core API Changes	6
8.1 New Data Type Constant.....	6
9. Data Engine API Changes	7
9.1 IResultIterator	7

1. Introduction

One of the limitations of BIRT 1.x is the incomplete support of database data types, including Binary, Binary Large Object (BLOB), and Character Large Object (CLOB) types. This BIRT 2.0 project seeks to provide support for these common data types in the Open Data Access (ODA) layer and in BIRT report design.

2. Features and Requirements

2.1 Open Data Access Framework Support for BLOB and CLOB

BLOB and CLOB data types will be added to the Open Data Access (ODA) framework, so that data source extensions for BIRT (and other consumers of ODA) can provide binary / large binary / large character data.

2.2 Binding Binary Column to BIRT Image Control

The BIRT Image control will enable binding to an ODA data set's Blob-typed fields or output parameter. This allows BIRT reports to display images contained in binary database columns or other binary data sources.

2.3 Binding Clob Columns to BIRT Controls

ODA Clob column or output parameter can be bound to any control that accepts String data.

2.4 Out-of-the-Box Support for JDBC BINARY/BLOB/CLOB Data Types

The ODA-JDBC driver will provide runtime support for JDBC BINARY/BLOB/CLOB data types by mapping them to corresponding ODA data types.

2.5 Samples for Binary Data Binding

The BIRT Sample Database and Sample reports will be enhanced to include sample on how SQL binary data can be used in reports.

3. Open Data Access Framework Support

3.1 New ODA Data Types

Two new scalar data types are added to ODA.

Blob – Used for all binary data. In particular, SQL BINARY and SQL BLOB data types are typically mapped to ODA Blob.

Clob – Used for large character data. In particular, SQL CLOB data type is typically mapped to ODA Clob.

3.2 API Changes

See document *BIRT API Change Control Document – Open Data Access Public Interfaces* for changes to the ODA interfaces. A link to this document is available at the *Specification Documents* section of this BPS:

<http://www.eclipse.org/birt/wiki/index.php?n=BPS.BPS3>

4. Using Binary Data in BIRT Report

4.1 Binary Output of Data Sets

An ODA data set can produce binary output in two ways:

- (1) A data set output column can have an ODA Blob data type.
- (2) A data set output parameter can have an ODA Blob data type.

BIRT report is capable of consuming binary data produced either way. In an ODA data set editor, the data type of an output column or output parameter is displayed as “Binary” if the underlying data is of type ODA Blob.

Binary is also a supported data type for columns of a Scripted Data Set and a Computed Column. However due to the limited ability of JavaScript to deal with binary data (see Section 4.3 it is generally difficult for JavaScript to generate binary data.

4.2 Binding Binary Data to Image Control

In BIRT 1.x, an image control obtains its image data from either an URI or from data embedded in the report design. In BIRT 2.0, an image control will also be able to obtain its content from the evaluation of a JavaScript expression. The JavaScript providing the image content typically returns an ODA data set column or an ODA data set output parameter.

The builder dialog of the image control will be enhanced with an additional choice, “Dynamic image”, as shown in Figure 1. When Dynamic image is chosen, the user can enter a JavaScript expression as the source of the data. This JavaScript expression is expected to refer to a data set column or output parameter with ODA data type Blob. For example, “*row.icon_data*” can be a valid JavaScript expression for dynamic image if the data set associated with the image control produces an output column named *icon_data* with Blob data type. A runtime error may result if the JavaScript expression evaluates to anything other binary data.

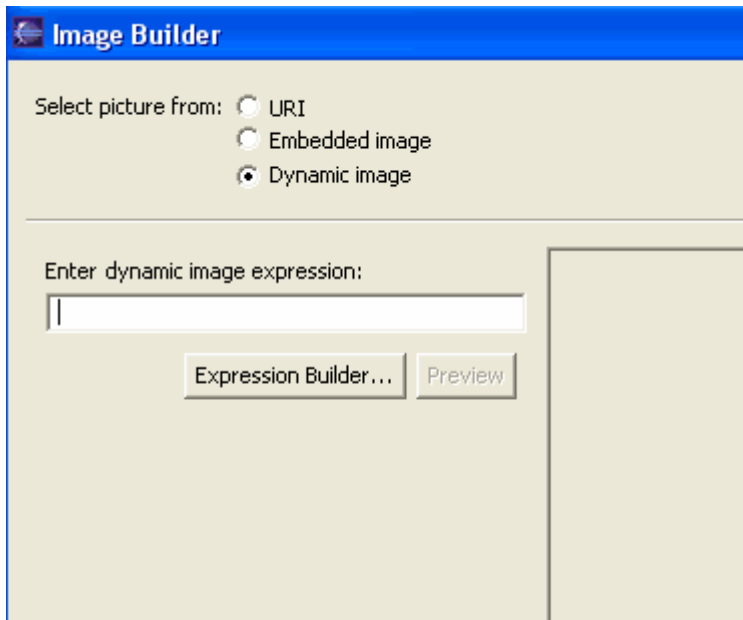


Figure 1 Image Builder Dialog

4.3 BIRT Scripting with Binary Data

Due to the lack of support of binary data types in JavaScript, the ability to work with binary data in BIRT scripting is limited. A binary column or parameter value is passed to JavaScript engine as a Java *byte[]* object, which cannot be manipulated directly in JavaScript. The script however can pass binary data obtained from a data set to external Java code that can operate on such data.

4.4 Report Parameters

BIRT 2.0 does not support binary data type for report parameters.

5. Using CLOB Data in BIRT Report

The ODA Clob data type (which is typically mapped to SQL CLOB data type in a JDBC data source) is handled the same way as String data type throughout the BIRT report designer and report engine. This means that

- (1) If a data set output column or output parameter has an ODA Blob data type, it is shown as having a String type in the data set editor.
- (2) Such a column or output parameter can be bound to any control that accepts String data.
- (3) In BIRT scripts, values for an ODA Blob column or parameter is processed as JavaScript strings.

The BIRT data engine handles the necessary conversion between ODA Clob and String data types.

6. ODA-JDBC Driver

The BIRT-supplied ODA-JDBC bridge driver will handle SQL BLOB, BINARY and CLOB data types according to the following rules:

- (1) SQL BINARY and SQL BLOB data types are mapped to ODA Blob data type
- (2) SQL CLOB data type is mapped to ODA Clob data type.

7. Model API Changes

7.1 New Data Type Constant

Interface *org.eclipse.birt.report.model.api.elements.DesignChoiceConstants*, which defines various enumeration constants used in BIRT model API, will be augmented with the following constant value:

```
public static final String COLUMN_DATA_TYPE_BINARY = "binary";
```

This constant value is used indicate ODA Blob as the type of data set column or data set parameters in the model API.

8. Core API Changes

8.1 New Data Type Constant

In BIRT core API class *org.eclipse.birt.core.data.DataType*, new constants (highlighted below) are added to represent the new binary data type

```
public abstract class DataType
{
    public static final int UNKNOWN_TYPE = -1;
    public static final int ANY_TYPE = 0;
    public static final int BOOLEAN_TYPE = 1;
    public static final int INTEGER_TYPE = 2;
    public static final int DOUBLE_TYPE = 3;
    public static final int DECIMAL_TYPE = 4;
    public static final int STRING_TYPE = 5;
    public static final int DATE_TYPE = 6;
    public static final int BLOB_TYPE = 7;
    public static final int BINARY_TYPE = 8;
    ...
    public static final String BINARY_TYPE_NAME = "Binary";
    ...
}
```

Note that the existing BLOB_TYPE was not supported in BIRT 1.x. It will remain unsupported in BIRT 2.0.

9. Data Engine API Changes

9.1 IResultIterator

Data Engine API interface *org.eclipse.birt.data.engine.api.IResultIterator* will have one additional method *getBytes()*. In addition, the *getValue()* method may now return an *byte[]* object.

```
public interface IResultIterator
{
    ...

    /**
     * Returns the value of a query result expression as a
     * byte[] array,
     * <br>
     * If the expression value has an incompatible type,
     * a <code>DataException</code> is thrown at runtime.
     * @param dataExpr      An <code>IBaseExpression</code>
     * object provided in the <code>IQueryDefinition</code>
     * at the time of prepare.
     * @return      The value of the given expression as byte[].
     *              It could be null.
     * @throws BirtException if error occurs in Data Engine
     */
    public byte[] getBytes( IBaseExpression dataExpr )
        throws BirtException;

    ...
}
```