ODA API Change Control Document

DTP Open Data Access Public Interfaces

Last Updated: March 12, 2007

Abstract

This document tracks the change requests to the public API of the Open Data Access (ODA) framework. For each change request, the document describes the new requirement, proposed solution and follow-up actions.

Document Revisions

Version	Date	Description of Changes
1.0	3/12/2007	Initial Version

1. Introduction

The Open Data Access (ODA) framework defines run-time and design-time interfaces for accessing data from both standard and custom data sources. A data source provider implements such interfaces for consumption by any ODA data consumer applications. A number of minor enhancements are identified for the ODA interfaces.

Below sections describe the enhancements and corresponding proposed interface changes in ODA version 3.1, for release in DTP 1.5 (Europa).

1.1 Additional data type support of "boolean" in result set columns and output parameter – Bugzilla 127234, 152443

https://bugs.eclipse.org/bugs/show_bug.cgi?id=127234

The "boolean" data type is added as a new ODA scalar data type. The new data type is supported in an ODA result set column, and in an ODA parameter for both input/output and scalar/complex types.

See the Changed APIs section below for the corresponding new methods in **`IResultSet`** (for output columns), **`IQuery`** (for scalar input parameters), **`IAdvancedQuery`** (for scalar output parameters) and **`IParameterRowSet`** (for complex input/output parameters).

The "Boolean" data type will also be added to the list of valid values for an ODA Scalar Data Type in the ODA dataSource extension point and the ODA Design model.

Note: An ODA run-time driver may choose to add support for the Boolean data type through any of these new methods, or simply throws an UnsupportedOperationException.

1.2 Provides explicit setting of a null value for input parameter – Bugzilla 122121, 142392

https://bugs.eclipse.org/bugs/show_bug.cgi?id=122121

From the perspective of an ODA data provider, an input parameter is assumed to have a default value, normally "null", if no value is explicitly set by a consumer application. It is up to individual ODA data providers to determine the default value for an input parameter. Since the default value may not necessarily be "null" for no value, an ODA consumer application may want to explicitly specify a "null" value for an input parameter.

See the Changed APIs section below on the new interface methods added in **`IQuery`** and **`IParameterRowSet`** to explicitly assign a "null" value to an input parameter.

1.3 Exposes the Name of a Parameter in its Runtime Meta Data – Bugzilla 177266, 122194

https://bugs.eclipse.org/bugs/show_bug.cgi?id=177266

The name of a parameter is not currently exposed in a parameter's metadata in the ODA runtime API, as the ODA runtime framework does not directly require a parameter's name. A parameter name is normally used at design-time, and is thus part of the parameter metadata defined in the ODA Design model.

However, a custom ODA designer may want to use its runtime driver to retrieve all the metadata of a data set parameter. It would thus be more convenient to expose a parameter's name as part of its metadata in the ODA runtime API.

See the Changed APIs section below on the new interface methods added in **`IParameterMetaData`** to retrieve the name of a parameter, if available.

1.4 Backward Compatibility Support

The ODA framework will provide binary backward compatibility support of any ODA extension compiled with a prior version of the ODA 3.0.x API interfaces. Through the use of the ODA Consumer Helper framework, if an ODA extension has not implemented a new interface method called by an ODA consumer application, an UnsupportedOperationException will be thrown on its behalf to the caller.

2. Changed APIs:

2.1 IResultSet

Component name = ODA Run-time Interfaces

Package name = org.eclipse.datatools.connectivity.oda

2.1.1 Change Request: Support boolean data type in a result set column.

Proposed Solution:

Add new getter methods to retrieve a primitive boolean value from a result set column.

```
/**
 * Gets the value of the designated column in the current row
 * as a boolean.
 * @param index column number (1-based)
 * @return      the boolean value in the specific column of the
                current row
 * @throws OdaException     if data source error occurs
 * @since       3.1
 */
public boolean getBoolean( int index ) throws OdaException;

/**
 * Gets the value of the designated column in the current row
 * as a boolean.
 * @param columnName    column name
 * @return       the boolean value in the specific column of the
                current row
 * @throws OdaException     if data source error occurs
 * @since       3.1
 */
public boolean getBoolean( String columnName ) throws
OdaException;
```

2.2 IQuery

Component name = ODA Run-time Interfaces

Package name = org.eclipse.datatools.connectivity.oda

2.2.1 Change Request: Support boolean data type in a parameter.

Proposed Solution:

Add new setter methods to assign a primitive boolean value to a scalar input parameter.

```
    /**
     * Sets the designated parameter to the given boolean value.
     * @param parameterName     name of the parameter.
     * @param value             boolean value.
     * @throws OdaException     if data source error occurs
     * @since 3.1
     */
    public void setBoolean( String parameterName, boolean value )
throws OdaException;

    /**
     * Sets the designated parameter to the given boolean value.
     * @param parameterId       id of the parameter (1-based).
     * @param value             boolean value
     * @throws OdaException     if data source error occurs
     * @since 3.1
     */
    public void setBoolean(int parameterId, boolean value ) throws
OdaException;
```

2.2.2 Change Request: Support explicit setting of a null value to an input parameter.

Proposed Solution:

Add new setter methods to explicitly assign a null value to a scalar input parameter.

```
    /**
     * Sets the designated parameter to a null value.
     * @param parameterName     name of the parameter.
     * @throws OdaException     if data source error occurs
     * @since 3.1
     */
    public void setNull( String parameterName ) throws OdaException;

    /**
     * Sets the designated parameter to a null value.
     * @param parameterId       id of the parameter (1-based).
     * @throws OdaException     if data source error occurs
     * @since 3.1
     */
    public void setNull( int parameterId ) throws OdaException;
```

2.3 IAdvancedQuery

Component name = ODA Run-time Interfaces

Package name = org.eclipse.datatools.connectivity.oda

2.3.1 Change Request: Support boolean data type in a parameter.

Proposed Solution:

Add new getter methods to retrieve a primitive boolean value from a scalar output parameter.

```
    /**
     * Returns the boolean value from the designated output
parameter.
     * @param parameterName     name of the parameter.
     * @return                  the boolean value.
     * @throws OdaException     if data source error occurs.
     * @since       3.1
     */
    public boolean getBoolean( String parameterName ) throws
OdaException;

    /**
     * Returns the boolean value from the designated output
parameter.
     * @param parameterId       id of the parameter (1-based).
     * @return                  the boolean value.
     * @throws OdaException     if data source error occurs.
     * @since       3.1
     */
    public boolean getBoolean( int parameterId ) throws OdaException;
```

2.4 IParameterRowSet

Component name = ODA Run-time Interfaces

Package name = org.eclipse.datatools.connectivity.oda

2.4.1 Change Request: Support boolean data type in a parameter.

Proposed Solution:

Add new setter methods to assign a primitive boolean value to a column of a complex input parameter.

```
    /**
     * Sets the boolean value at the designated column.
     * @param columnIndex   index of the column.
     * @param value         the boolean value.
     * @throws OdaException     if data source error occurs.
     * @since 3.1
     */
    public void setBoolean( int columnIndex, boolean value ) throws
OdaException;

    /**
     * Sets the boolean value at the designated column.
```

```
     * @param columnName    name of the column.
     * @param value         the boolean value.
     * @throws OdaException     if data source error occurs.
     * @since 3.1
     */
    public void setBoolean( String columnName, boolean value ) throws
OdaException;
```

2.4.2 Change Request: Support explicit setting of a null value to an input parameter.

Proposed Solution:

Add new setter methods to explicitly assign a null value to a column of a complex input parameter.

```
    /**
     * Sets a null value at the designated column.
     * @param columnIndex   index of the column.
     * @throws OdaException     if data source error occurs
     * @since 3.1
     */
    public void setNull( int columnIndex ) throws OdaException;

    /**
     * Sets a null value at the designated column.
     * @param columnName    name of the column.
     * @throws OdaException     if data source error occurs
     * @since 3.1
     */
    public void setNull( String columnName ) throws OdaException;
```

2.5 IParameterMetaData

Component name = ODA Run-time Interfaces

Package name = org.eclipse.datatools.connectivity.oda

2.5.1 Change Request: Exposes the name of a parameter in its runtime metadata.

Proposed Solution:

Add a new getter method to retrieve the name in a scalar parameter's metadata.

```
    /**
     * Returns the name of the specific parameter.
     * Optional method; a parameter may or may not have a name.
     * @param param 1-based index of the parameter.
     * @return      the parameter name, or
     *              null if the name is not available or this
parameter is not named.
     * @throws OdaException     if data source error occurs.
     * @since 3.1
     */
    public String getParameterName( int param ) throws OdaException;
```

2.6 OdaScalarDataType

Component name = ODA Design Model Interfaces

Package name = org.eclipse.datatools.connectivity.oda.design

2.6.1 Change Request: Support boolean data type in a result set column and parameter.

Proposed Solution:

Add a new data type code and literal string for the boolean data type.

```
public static final int BOOLEAN;
public static final OdaScalarDataType BOOLEAN_LITERAL;
```

3. Added APIs:

None.

4. Removed APIs:

None.

5. Miscellaneous Change Requests

5.1 datasource.exsd

Component name = ODA Plug-in Extension Point Schema Definition

Package name = org.eclipse.datatools.connectivity.oda.dataSource

Change Request:

Defines the mapping of an ODA data source's native data type to the ODA Boolean data type.

Proposed Solution:

Add Boolean data type to the list of ODA scalar data type names.

```
<attribute name="odaScalarDataType" use="required" >
    <simpleType>
       <restriction base="string">
          <enumeration value="Date">
          </enumeration>
          <enumeration value="Double">
          </enumeration>
          <enumeration value="Integer">
          </enumeration>
          <enumeration value="String">
          </enumeration>
          <enumeration value="Time">
```

```
            </enumeration>
            <enumeration value="Timestamp">
            </enumeration>
            <enumeration value="Decimal">
            </enumeration>
            <enumeration value="Blob">
            </enumeration>
            <enumeration value="Clob">
            </enumeration>
            <enumeration value="Boolean">
            </enumeration>
        </restriction>
    </simpleType>
</attribute>
```