



Introducing Open Data Access Framework in Data Tools Platform

Linda Chan – Actuate[®] Corporation
Open Data Access Component Lead
DTP PMC Member

Topics

- What is Open Data Access₁ (ODA)? Why?
- Overview – ODA Framework
- Build a Custom ODA Data Provider
- Consume any ODA Data Providers
- Q & A

₁ Actuate® patent pending

Business Requirements – Getting at the Data

Variety of Data Sources

- Standard sources such as RDBMS, packaged applications
- Non-standard sources such as custom & legacy applications

Use of an application layer

- Application systems often use an application layer to manage business rules and security
- Application data is typically very complex and hard to reach
 - ◆ Proprietary data storage mechanism
 - ◆ Complex APIs
 - ◆ XML-based storage
 - ◆ Web Services

Solving Application Data Access Problem

- Typical Approach:
 - ◆ Export data to a “Data Mart” for easy access
- But...
 - ◆ *Extra complexity of additional database and exports*
 - ◆ *Indirect mapping of application terminology to database*
 - ◆ *Exported data does not reflect current data*

Solving Application Data Access Problem

- Better approach:
 - ◆ Enable users to access data directly
 - Eliminates data latency
 - ◆ Use a friendly Query Builder for direct data access
 - Customized UI to adopt application brand and terminology
 - Query UI can mask underlying data access complexity

Solution – Open Data Access

Enables data access to any data source

- Provides an abstraction for accessing heterogeneous data sources
- Highly scalable data retrieval and easy end-user experience
- Built using familiar Eclipse extension points

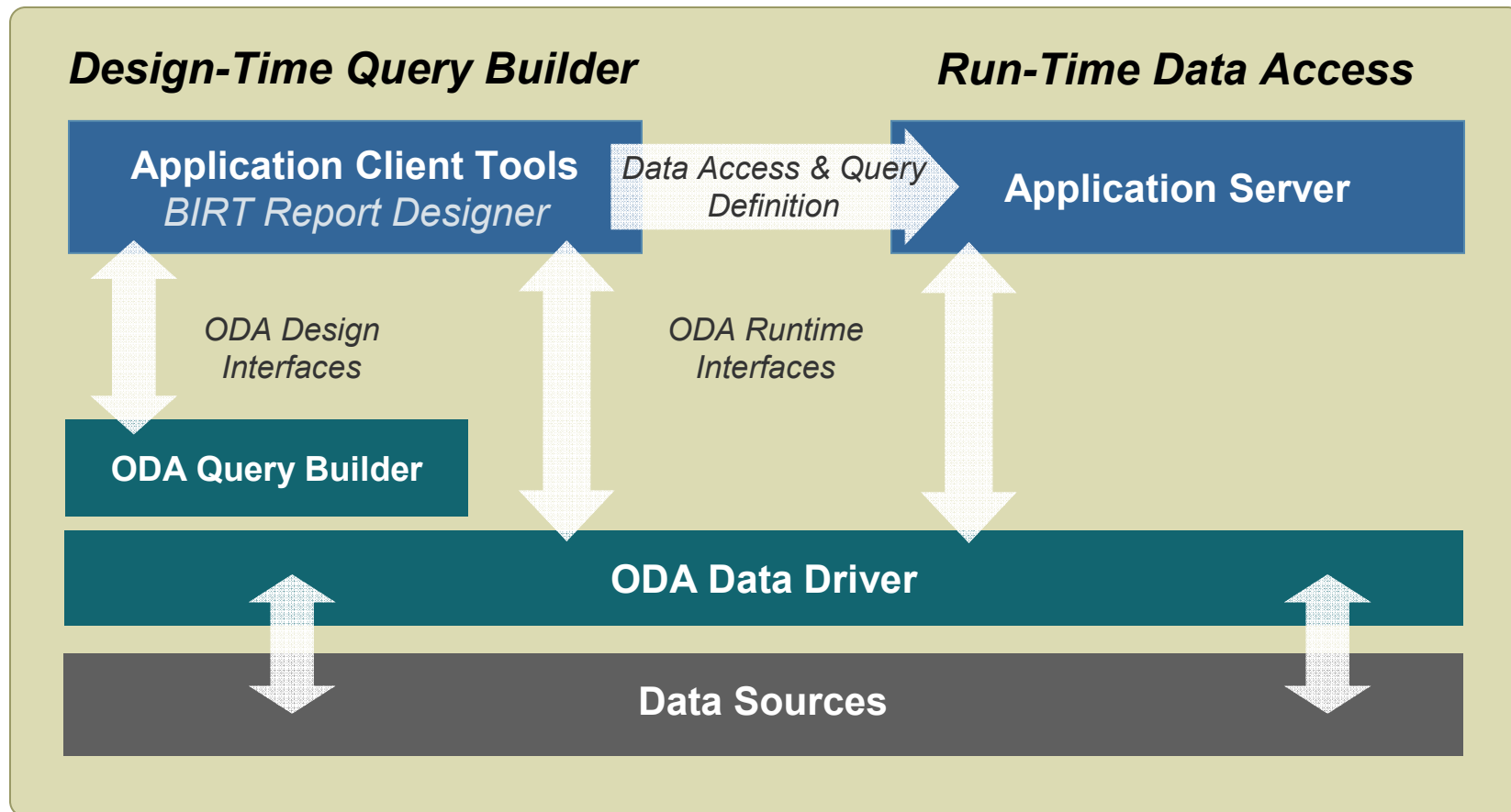
Extends & customizes Applications Systems

- Applications can provide custom data driver and query builder for accessing their data

Provides proven framework

- BIRT's built-in data connectors use ODA
- Commercial products' data connectors use ODA

Overview – Open Data Access Architecture



Complete control of data access and branding

Overview – Open Data Access Framework

- Run-time
 - ◆ Public API defines a set of Java™ interfaces that allow heterogeneous data to be accessed in a generic way
- Design-time
 - ◆ Allows data source-specific user interface to be embedded in any ODA-compliant consumer application
- ODA Consumer Helper
 - ◆ Built-in support for any ODA-compliant consumer application to consume data from any ODA data provider

ODA Run-time Framework

- Stable run-time API; no change since DTP 1.5 (Europa) release
- Integrates with DTP Connection Profile and JDBC™ Databases framework
- Implementation wraps data source-specific APIs
- Run-time plug-in implements
 - ◆ extension points
 - *org.eclipse.datatools.connectivity.oda.dataSource*
 - *org.eclipse.datatools.connectivity.connectionProfile*
 - ◆ ODA run-time API interfaces
 - *org.eclipse.datatools.connectivity.oda* package

ODA Run-time API Interfaces

- Defines the primary run-time operations needed from an ODA data provider to access and retrieve data from a data source
- Java interfaces
 - ◆ JDBC-like, extended to support additional capabilities of non-RDBMS data sources
 - ◆ Emphasis on scalable data retrieval
- Main run-time operations
 - ◆ Data Source Connection – *IConnection*
 - Establishes a live connection to any type of data source
 - Obtains provider capabilities for each type of data set queries
 - Creates one, or multiple, concurrent data set-specific queries

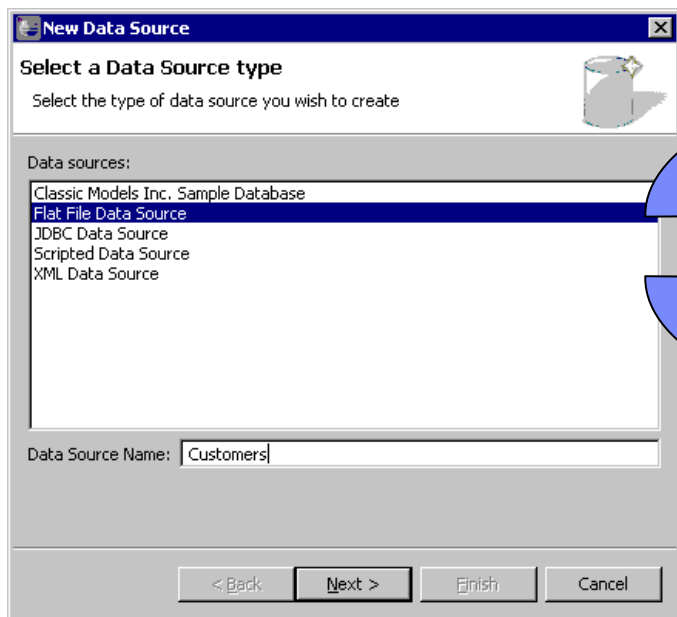
ODA Run-time API Interfaces

- Main run-time operations (cont'd)
 - ◆ Data Set Query – *IQuery, IAdvancedQuery*
 - Prepares and executes a data set-specific query text command, e.g. XPath, MDX, SQL, Stored Procedure calls
 - Handles one or more sets of data rows, i.e. result sets, retrieved by a single data set query
 - ◆ Query Parameters – *IParameterMetaData, IParameterRowSet*
 - Provides run-time metadata of parameters specified in a prepared query
 - Handles scalar and complex input/output parameters
 - ◆ Result Sets – *IResultSet, IResultSetMetaData*
 - Fetches tabular data rows
 - Allows sequential or concurrent access to multiple result sets

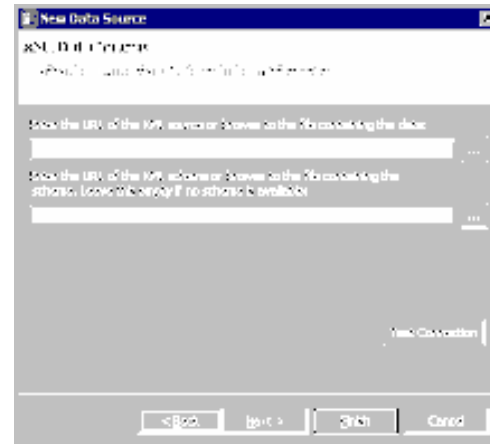
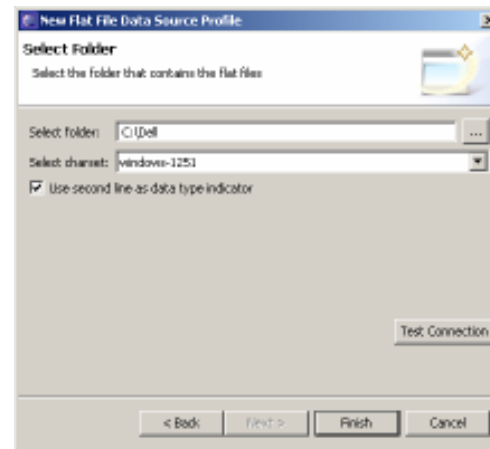
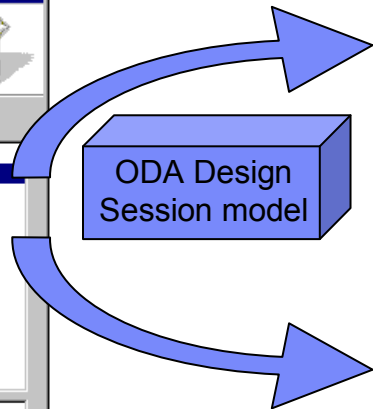
ODA Design-time Framework

- Eclipse Modeling Framework (EMF) model-based design-time interfaces
 - ◆ communicate connection properties, query and parameter definitions to an ODA consumer application
- Integrates with DTP Data Source Explorer view

ODA Design-time Framework



ODA Consumer Designer
(e.g. BIRT Data Source Wizard)



Customized Wizard Page
contributed by ODA Data Providers

ODA Designer UI Plug-in

- Implements
 - ◆ extension points
 - *org.eclipse.datatools.connectivity.connectionProfile*
 - *org.eclipse.ui.propertyPages*
 - *org.eclipse.datatools.connectivity.oda.design.ui.dataSource*
 - *org.eclipse.datatools.connectivity.ui.connectionProfileImage*
(optional)
 - ◆ customizes ODA Designer UI pages
 - *org.eclipse.datatools.connectivity.oda.design.ui.wizards* package
- Communicates its Data Source and Data Set design definitions in an *ODA Design Session* model

ODA Design Session model

- Eclipse Modeling Framework (EMF) model based
 - ◆ *org.eclipse.datatools.connectivity.oda.design* package
- Allows customized data source and query builders to design the slice of data to access at run-time
 - ◆ Communicates connection information, query and parameter definitions to an ODA consumer application
 - ◆ Transient Objects
 - root element: `org.eclipse.datatools.connectivity.oda.design.OdaDesignSession`
- ODA design-time consumer application, e.g. BIRT Report Designer, initiates an ODA design session
 - ◆ Consumes a data provider's UI page contributions
 - ◆ Adapts an edited data access design to host-specific design components
 - ◆ Provides persistent services for editing designs

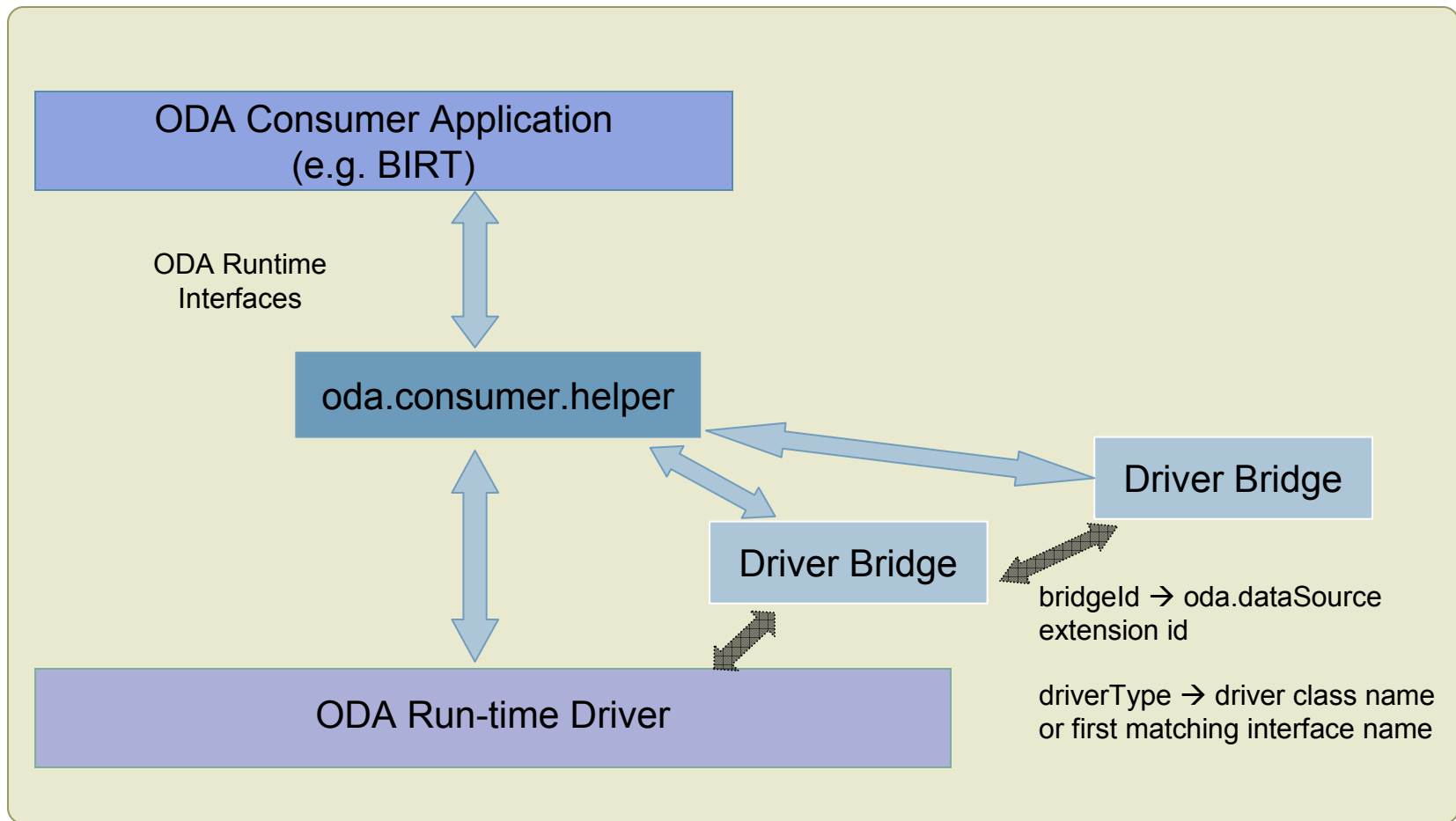
Build a Custom ODA Data Provider

- From scratch
 - ◆ PDE New Plug-in Project Wizard
 - Demo
- Enhance an existing ODA provider
 - ◆ driverBridge extension point
 - *org.eclipse.datatools.connectivity.oda.consumer.driverBridge*

Enhance an Existing ODA Data Provider

- Modify/enhance behavior of an existing ODA data provider
 - ◆ *org.eclipse.datatools.connectivity.oda.consumer.driverBridge* extension point
- Ideal for minor enhancements
 - ◆ preserves existing ODA extension ID
 - ◆ Supports “chained” bridges

DriverBridge Extension Point



Enhance an Existing ODA Data Provider

- DriverBridge extension implements
 - ◆ Extension Point
 - `org.eclipse.datatools.connectivity.oda.consumer.driverBridge`
 - ◆ Sample driverBridge extension element

```
<bridge
  driverType="org.eclipse.birt.report.data.oda.jdbc.OdaJdbcDriver"
  bridgeId="org.eclipse.birt.report.data.testjdbc">
</bridge>
```
 - ◆ Bridge driver is handed its underlying driver instance
 - `IDriver.setAppContext(Map)`
 - ✦ Key: `OdaDriver.ODA_BRIDGED_DRIVER`
 - ✦ Value: underlying driver instance
 - May use own API between Bridge and underlying Driver instances
 - ◆ Sample code attached in Bugzilla 135246

ODA Data Providers

- DTP Incubation Project
 - ◆ Welcomes contribution – custom ODA data providers
 - ◆ Initial contribution – ECore ODA Data Provider (Bugzilla 132958)

ODA Data Consumers

- ODA Consumer Helper Run-time component
 - ◆ Manages the diverse behavior and capabilities of individual ODA data providers
 - ◆ Supports driverBridge extension point
 - ◆ Adopts the same ODA runtime interfaces
- ODA Designer Host API
 - ◆ Manages custom wizard and property page contributions of an ODA data source and data set
 - ◆ Collects a data source and data set design defined by custom ODA page contributions
 - *org.eclipse.datatools.connectivity.oda.design.ui.designsession* package
 - *org.eclipse.datatools.connectivity.oda.design* package

ODA Framework Enhancements

- Ganymede – DTP 1.6 release
 - ◆ Better Integration with the Connection Profile framework
 - ◆ Integrates with
 - DTP JDBC Databases as an ODA data source
 - DTP SQL Query Builder
- Future
 - ◆ Multi-dimensional data sources

Resources

- ODA Overview document
 - ◆ link on DTP Connectivity home page
<http://wiki.eclipse.org/Connectivity>
- Developers' Guide for Connectivity Frameworks
- DTP Help Contents – Javadoc™ and Extension Points' Schema
- Out-of-the-Box ODA Data Providers
 - ◆ *org.eclipse.datatools.connectivity.oda.flatfile** plug-ins
 - ◆ *org.eclipse.datatools.enablement.oda.xml** plug-ins
 - ◆ *org.eclipse.datatools.enablement.oda.ws** plug-ins
 - ◆ *org.eclipse.birt.report.data.oda.jdbc** plug-ins
- Presentation slides
 - ◆ <https://eclipsecon.greenmeetingsystems.com/submissions/view/4>

Open Data Access Framework

- Discussions, Feedbacks, Q&As

Legal Notices

- Actuate is a registered trademark of Actuate Corporation and/or its affiliates in the U.S. and certain other countries.
- Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
- Other company, product, or service names may be trademarks or service marks of others.