

# MODERNIZE YOUR REAL- WORLD APPLICATION WITH E4 AND JAVAFX

COLE MARKHAM  
SENIOR SOFTWARE ENGINEER  
THE WELLAND COMPANY

[CMARKHAM@WELLAND.COM](mailto:CMARKHAM@WELLAND.COM)



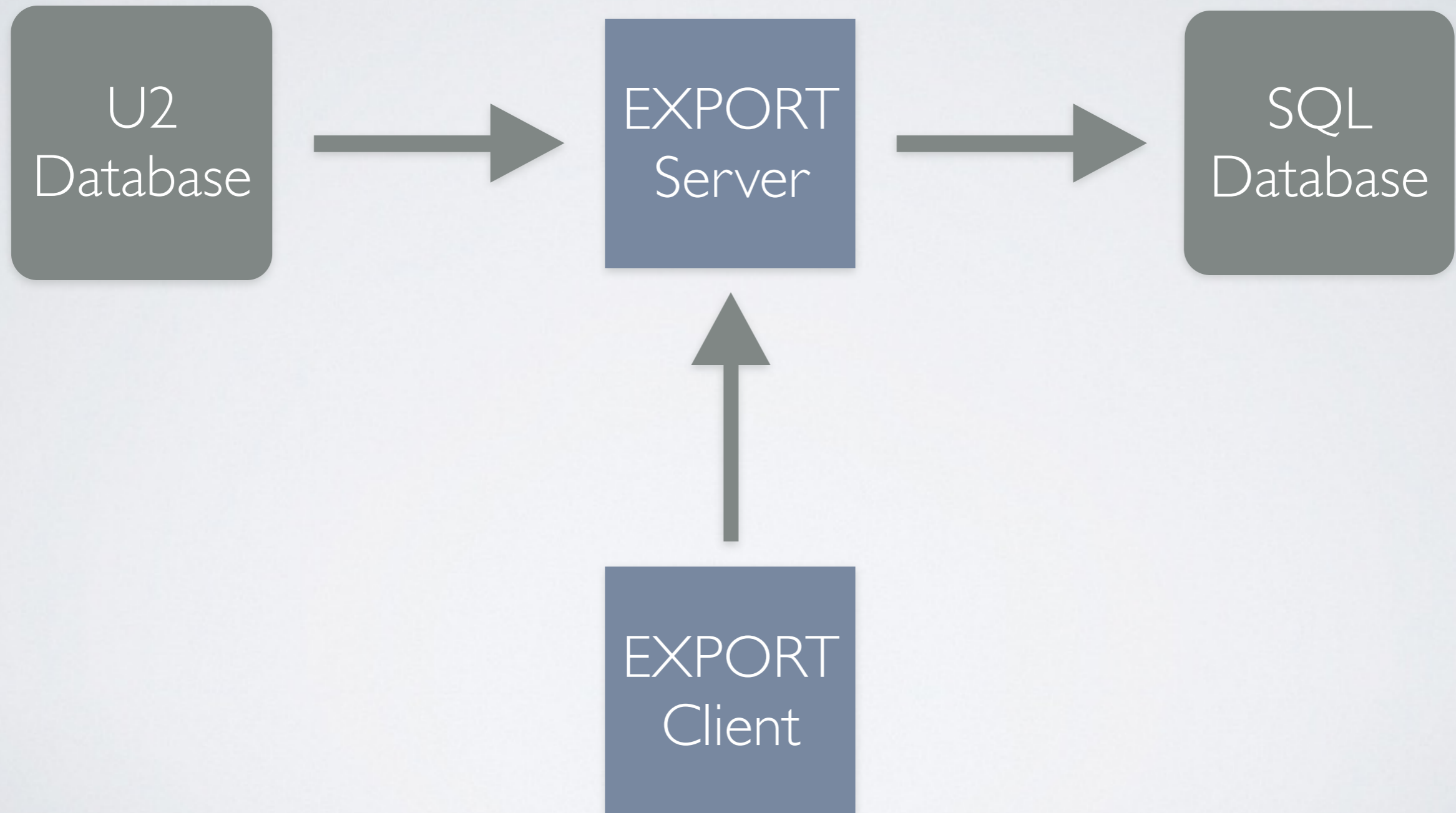
# WELLAND BACKGROUND

- Welland Company celebrating 24 years
- Flagship products
  - EXPORT
  - FORMS
- IBM / Rocket Business Partner
- Perform support for UniVerse / UniData sites

# EXPORT PRODUCT OVERVIEW

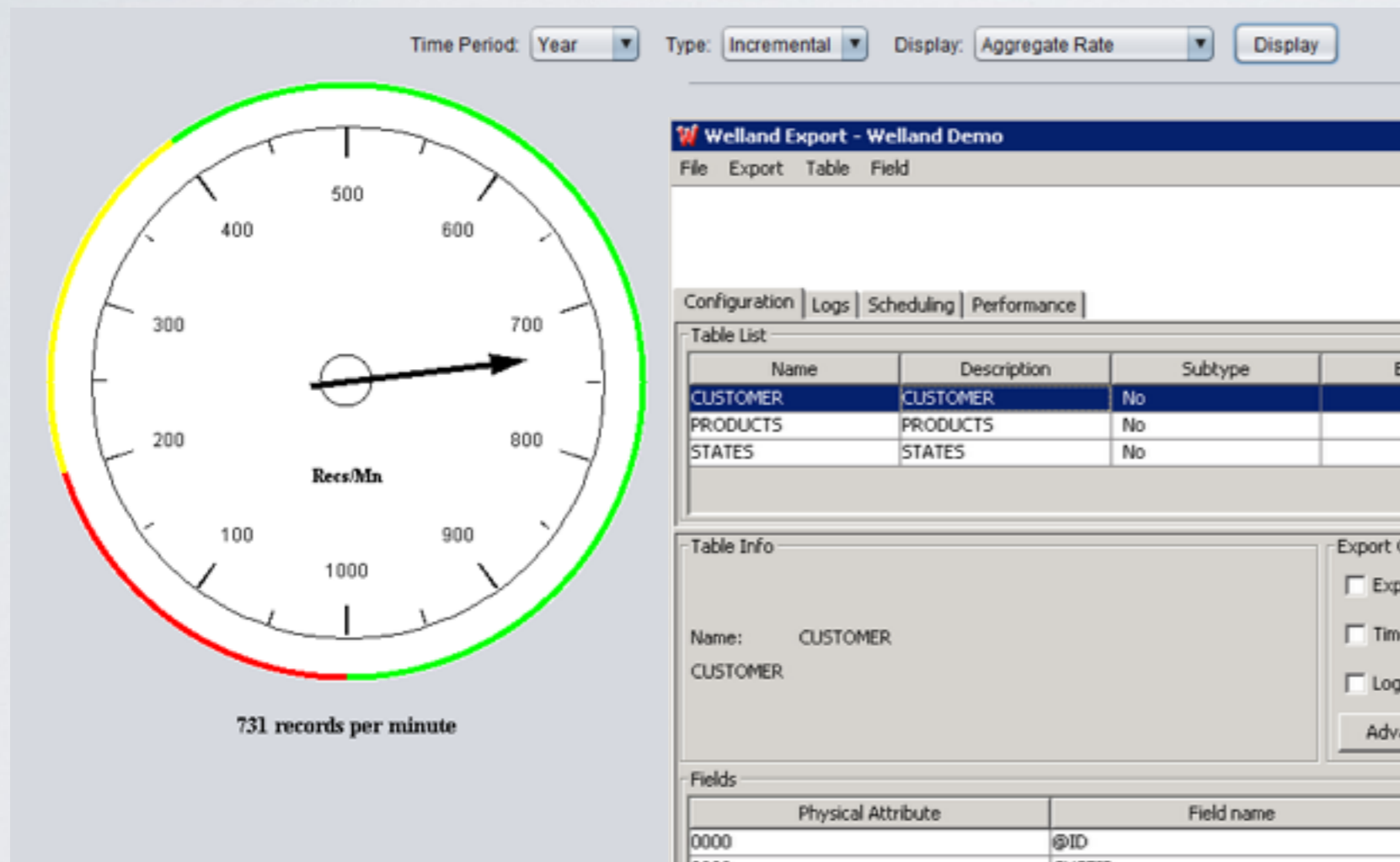
- Transfers U2 data into an SQL compliant database
- Access data for reporting, warehousing, and web services
- Empowers users to build reports and queries with industry-standard tools.

# EXPORT ARCHITECTURE





# MAKE IT LOOK LIKE IT WASN'T BUILT IN 1995!



Welland Export - Welland Demo

File Export Table Field Help

Configuration | Logs | Scheduling | Performance

Table List

Name	Description	Subtype	Export?	Time Stamp	Subtypes Only	Log Deletions
CUSTOMER	CUSTOMER	No	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PRODUCTS	PRODUCTS	No	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
STATES	STATES	No	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Table Info

Name: CUSTOMER  
CUSTOMER

Export Options

Export     Export subtypes only  
 Time Stamp     Export as Base table  
 Log Deletions

Advanced

Fields

Physical Attribute	Field name	Export	Description
0000	@ID	<input type="checkbox"/>	CUSTOMER
0000	CUSTID	<input type="checkbox"/>	Customer ID
0001	SAL	<input type="checkbox"/>	Selutation
0002	FNAME	<input type="checkbox"/>	First Name
0003	LNAME	<input type="checkbox"/>	Last Name
0004	COMPANY	<input type="checkbox"/>	Company Name
0005	ADDR1	<input type="checkbox"/>	Address line 1
0006	ADDR2	<input type="checkbox"/>	Address line 2
0007	CITY	<input type="checkbox"/>	City
0008	STATE	<input type="checkbox"/>	State
0009	ZIP	<input type="checkbox"/>	Zip

Delete Save Refresh

# BUT IT'S BETTER THAN THIS...

```
UniVerse Shell
LIST CUSTOMER 06:47:22pm 24 Mar 2014 PAGE 2
for M3000/C3000
Service..
Price.. Discount Service... Service. Service contract.
start date end date price.. paid date
$6,600 4.2 01/13/91 01/15/92 $900 02/05/91
$12,000 7.6 01/13/91 06/12/91 $500 02/05/91
$900 54.8 01/13/91 01/15/92 $150 02/12/91
```

```
CUSTOMER..... 4
Customer ID.... 4
Salutation..... Ms.
First Name..... Jill
Last Name..... Kahn
Contact Name... Ms. Jill Kahn
Company Name... Fast Copy Center
Address line 1. 12 School St.
Address line 2.
Street Address. 12 School St.
City..... Boston
State..... MA
State name..... Massachusetts
Press any key to continue...
```

```
UniVerse Shell
LA Louisiana
MA Massachusetts
ME Maine
Press any key to continue...
Quit
>LIST PRODUCTS
LIST PRODUCTS 10:10:17pm 24 Mar 2014 PAGE 1
PRODUCTS.. Product ID List Price Product Description.....
M1000 M1000 $1,990 Low cost, entry level, light
duty, monochrome copier
M2000 M2000 $4,490 Moderate duty, monochrome
copier
C2000 C2000 $6,890 Moderate duty, entry level,
color copier
M3000 M3000 $12,990 Heavy duty monochrome copier
C3000 C3000 $17,990 Heavy duty color copier
S2000 S2000 $990 Sorting attachment for
M2000/C2000
S3000 S3000 $1,990 Sorting attachment for
M3000/C3000
7 records listed.
>
```

# NEW SWING L&F?

W product.name@ - Cole's Test Server

File @product.name.verb@ Table Field Help

Configuration Logs Scheduling Performance

Table List

Name	Description	Subtype	@product.na...	Time Stamp	Subtypes Only	Log Deletions
C2	C2	No	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CUSTOMER	CUSTOMER	No	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CUSTOMERS2	CUSTOMERS2	No	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
EDA_DATASOU...	EDA_DATASOU...	No	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
EDA_DRIVER	EDA_DRIVER	No	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
EDA_EXCEPTION	EDA_EXCEPTION	No	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
HELP.FILE	HELP.FILE	No	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Table Info @product.name.verb@ Options

Name:

@product.name.verb@  @product.name.verb@ subtypes only

Time Stamp  @product.name.verb@ as Base table

Log Deletions

Advanced

Fields

Physical Attribute	Field name	@product.name.verb@	Description
--------------------	------------	---------------------	-------------

Delete Save Refresh

# DESIGNER'S VISION

**W** Welland Export - EXPORT FILE EXPORT TABLE FIELD □ × ?

LOGO

Configuration  
Logs  
Scheduling  
Performance

### Table List

DATA	DESCRIPTION	SUBTYPE	EXPORT	TIME STAMP	SUBTYPES ONLY	LOG DEL...
Field Data	Field Data	No	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Field Data	Field Data	No	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Field Data	Field Data	No	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Field Data	Field Data	No	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Field Data	Field Data	No	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Field Data	Field Data	No	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Field Data	Field Data	No	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Field Data	Field Data	No	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

### Table Info

Batch\_Control

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna.

Quarter	Q1	Q2	Q3	Q4
Value	\$0.5M	\$2.5M	\$3.0M	\$2.8M

### Fields

Export  Export subtypes only  
 Time Stamp  Export as Base table  
 Log Deletions

ADVANCED

### Pie Chart

Title

Category	Percentage
45% Title	45%
8% Title	8%
10% Title	10%
12% Title	12%
25% Title	25%

### Export Options

DATA	DESCRIPTION	SUBTYPE	EXPORT	TIME STAMP	SUBTYPES ONLY	LOG DEL...
Field Data	Field Data	No	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Field Data	Field Data	No	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Field Data	Field Data	No	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Field Data	Field Data	No	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Field Data	Field Data	No	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Field Data	Field Data	No	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Field Data	Field Data	No	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Field Data	Field Data	No	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

DELETED SAVE REFRESH



# BACK TO THE DRAWING BOARD

- UI Frameworks

- Swing
- SWT
- JavaFX
- HTML5
- .NET

- Application Frameworks

- RCP 3.x
- E4
- Scout
- Riena
- .NET

# Why JavaFX?

- FXML using SceneBuilder
- CSS Styling
- Chart library
- e(fx)clipse tooling and E4 Renderers
- Deployment via packaged application with JRE

# CSS STYLING

- 500 Lines of CSS for our custom theme
- E4 Renders add CSS classes

```
.MWindow {  
  -fx-border-width: 3.0px;  
  -fx-border-color: darkgray  
}  
.MPartSashContainer {  
  -fx-background-color: transparent;  
  -fx-effect:null;  
  -fx-padding: 0.0;  
}  
.MPartSashContainer .split-pane-divider {  
  -fx-border-style: none;  
  -fx-background-color: transparent;  
  -fx-effect:null;  
  -fx-padding: 20.0px;  
}  
.MPerspective {  
  -fx-padding: 20.0 20.0 20.0 20.0;  
}
```

```
.menu-bar {  
  /*-fx-padding: 6 20 6 20;*/  
  -fx-background-color: transparent;  
}  
.menu-bar .menu {  
  -fx-padding: 0.0 10.0 0.0 0.0;  
  -fx-font-size:12.0;  
}  
.menu:hover {  
  -fx-background-color: rgba(198.0,39.0,48.0,0.6);  
}  
.menu:showing {  
  -fx-background-color:rgb(198.0,39.0,48.0);  
}  
#menupane {  
  -fx-background-color: linear-gradient(gray, black)  
}  
.menu-text {  
  -fx-text-fill: white;  
}
```

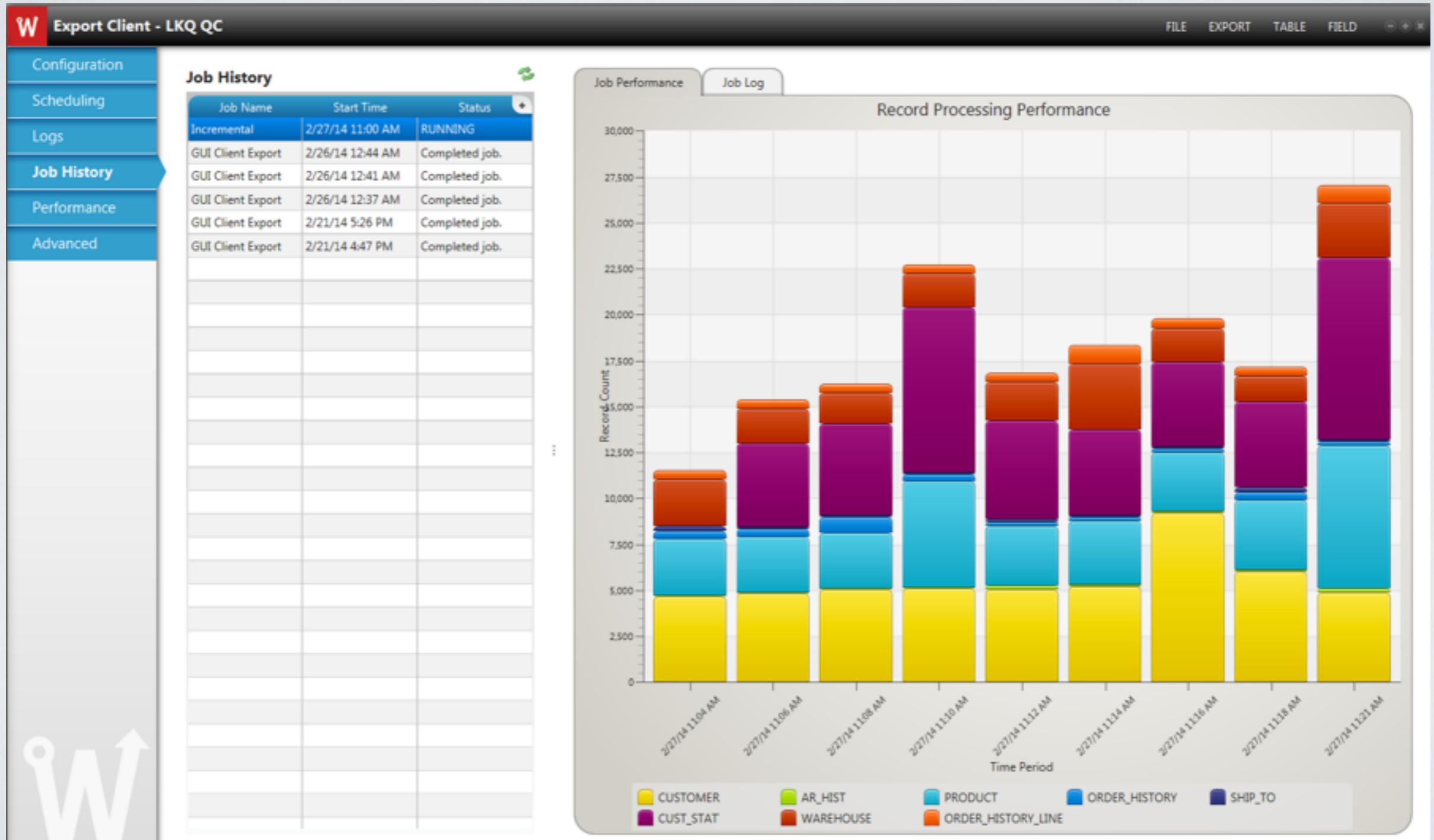
# SCENE BUILDER

The image displays the Scene Builder interface, which is used for designing user interfaces. It is divided into several main sections:

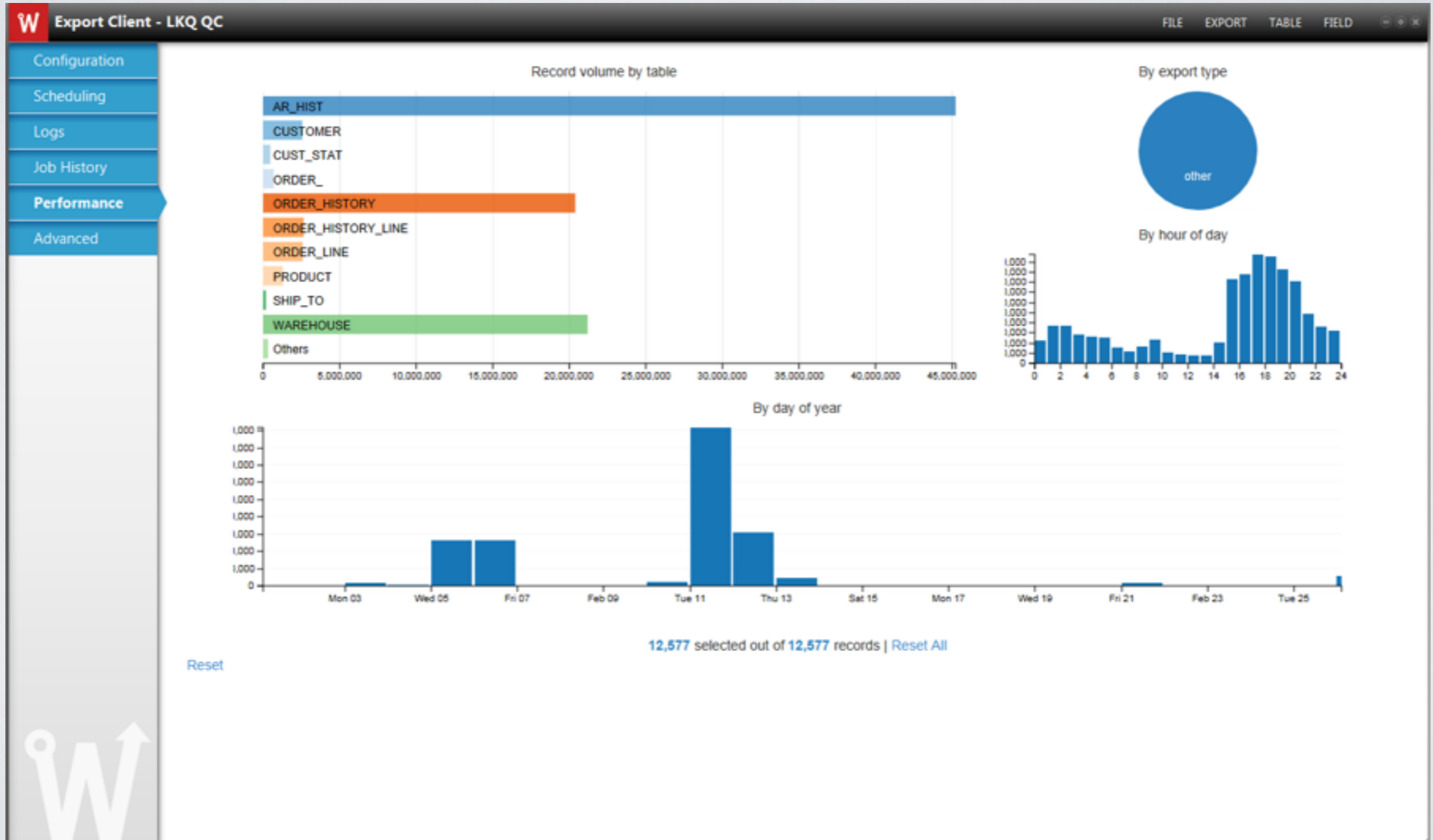
- Library:** A sidebar on the left containing a search bar and a list of UI components categorized into Containers, Controls, Menu, Miscellaneous, Shapes, Charts, and 3D. The Containers list includes Accordion, AnchorPane, BorderPane, FlowPane, GridPane, HBox, Pagination, and Pane.
- Document:** A central area showing a hierarchy of the current scene. It includes a 'Hierarchy' tree where a 'Tab Field Configuration' is expanded to show an 'AnchorPane' containing a 'BorderPane' with 'insert TOP' and 'insert LEFT' buttons, an 'HBox', and a 'VBox' containing a 'Label Identity' and a 'GridPane (2 x 4)'. The GridPane contains several 'Label' elements with coordinates and names, and a 'CheckBox (1, 2)'. A 'Controller' section is visible at the bottom.
- Configuration Panel:** A large panel on the right for configuring the selected element. It has two tabs: 'Field Configuration' and 'Post-processing rules'. The 'Field Configuration' tab is active and shows various settings for a field, including:
  - Identity:** Table (Label), Name (Label), Derived Field (checkbox), Phys Attr (text input), Format (text input), I-Type (text input), Format (dropdown), Lenient (checkbox), Use time file (checkbox), Length (text input), Decimals (text input), Justification (dropdown), Cov.Code (text input), SMV (dropdown), Custom MV Sep. (text input), Cross-Validation t... (text input), Cross-Validation f... (text input), Indexed MV offset (text input), Foreign Key (checkbox), and Validate tot (checkbox).
  - Misc.:** Add to existing subtable (checkbox), a dropdown menu, a text input, SQL Field Name (text input), Description (text area), Long Description (text area), and Post proces (checkbox).



# CHART LIBRARY



# HTML5 SUPPORT



# Why Eclipse 4?

- Modern architecture
- Modularity
- Application model
- Custom Renderers
- Dependency Injection
- OSGi

# RAPID PROTOTYPING

The screenshot displays the 'Export Client - Cole's Test Server' application. The main window is titled 'Export Client - Cole's Test Server' and features a menu bar with 'File' and a search bar for 'Search Help'. Below the menu bar are tabs for 'Configuration', 'Scheduling', 'Logs', and 'Performance'. The 'Tables' section is active, showing a table with columns: NAME, DESCRIPTION, EXPORT?, TIME STAMP, and LOG DELETES. Below this is a 'Fields' section with a table of physical attributes and field names.

**Tables**

NAME	DESCRIPTION	EXPORT?	TIME STAMP	LOG DELETES
C2	C2			
CUSTOMER	CUSTOMER			
CUSTOMERS2	CUSTOMERS2			
EDA_DATASOURCE	EDA_DATASOURCE			
EDA_DRIVER	EDA_DRIVER			
EDA_EXCEPTION	EDA_EXCEPTION			
HELP.FILE	HELP.FILE			
HS_FILE_ACCESS	HS_FILE_ACCESS			
LKQ.CUSTOMER	LKQ.CUSTOMER			
M2	M2			
MEMOS	MEMOS			
PRODUCTS	PRODUCTS			

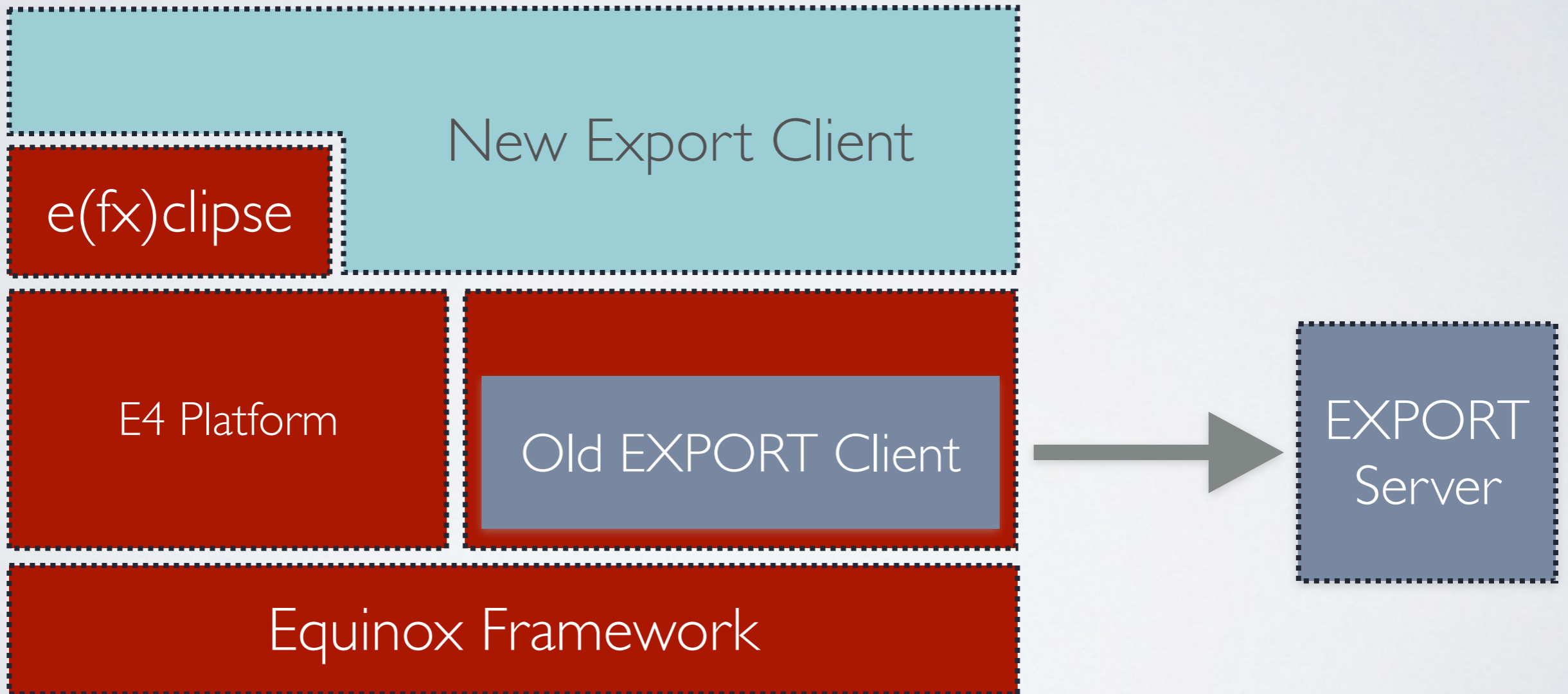
**Fields**

Physical Attribute	Field Name	
0000	BASE_CUSTOMER	
0000	CUSTID	Customer I
0001	SAL	Salutation
0002	FNAME	First Name
0003	LNAME	Last Name
0004	COMPANY	Company M
0005	ADDR1	Address lin
0006	ADDR2	Address lin
0007	CITY	City
0008	STATE	State
0009	ZIP	Zip
0010	PHONE	Telephone

The 'Advanced' tab is also visible, showing a 'Table List' with a list of table names and an 'Excluded Records' section with columns: Base Table Name, Table Name, Account Name, Prefix, and Suffix. The 'Excluded Records' section currently displays 'No table selected.'



# EXPORT ARCHITECTURE



# FXML LOADING

- Custom factory class used for most parts
  - Handles loading FXML resource defined in model
- Controller class specified in FXML file
  - Inject @FXML variables into controller
  - Full Eclipse dependency injection support

# FXML LOADING

- Custom factory class used for most parts

```
public class FxmlLoadedPart{
    // Loaded from the FXML
    private Node mainControl;
    @PostConstruct
    public void postConstruct(
        @org.eclipse.fx.ui.di.FXMLLoader
        FXMLLoaderFactory factory,
        BorderPane parent,
        MPart part){
        try{
            String partFXML = part.getProperties().get("welland.part.fxml");
            if (partFXML == null){
                getLogger(this).warning("welland.part.fxml was null for part: " + part.getLabel() );
                return;
            }
            FXMLLoader<Node> builder = factory.loadBundleRelative(partFXML);
            mainControl = builder.load();
            parent.setCenter(mainControl);
        }catch(IOException e){
            LoggerFactory.getFactory().getLogger(this).log(Level.WARNING,
                "Error loading fxml", e);
        }
    }
    @Focus
    public void onFocus(){
        mainControl.requestFocus();
    }
}
```

# IMPLEMENTING A LOGIN DIALOG

The screenshot displays the 'Export Client' application interface. A 'Login' dialog box is open, featuring the 'Export' logo and the following fields:

- Connection:** A dropdown menu set to 'Local' with a green '+' icon to its right.
- User Name:** A text input field containing the text 'export'.
- Password:** An empty text input field.

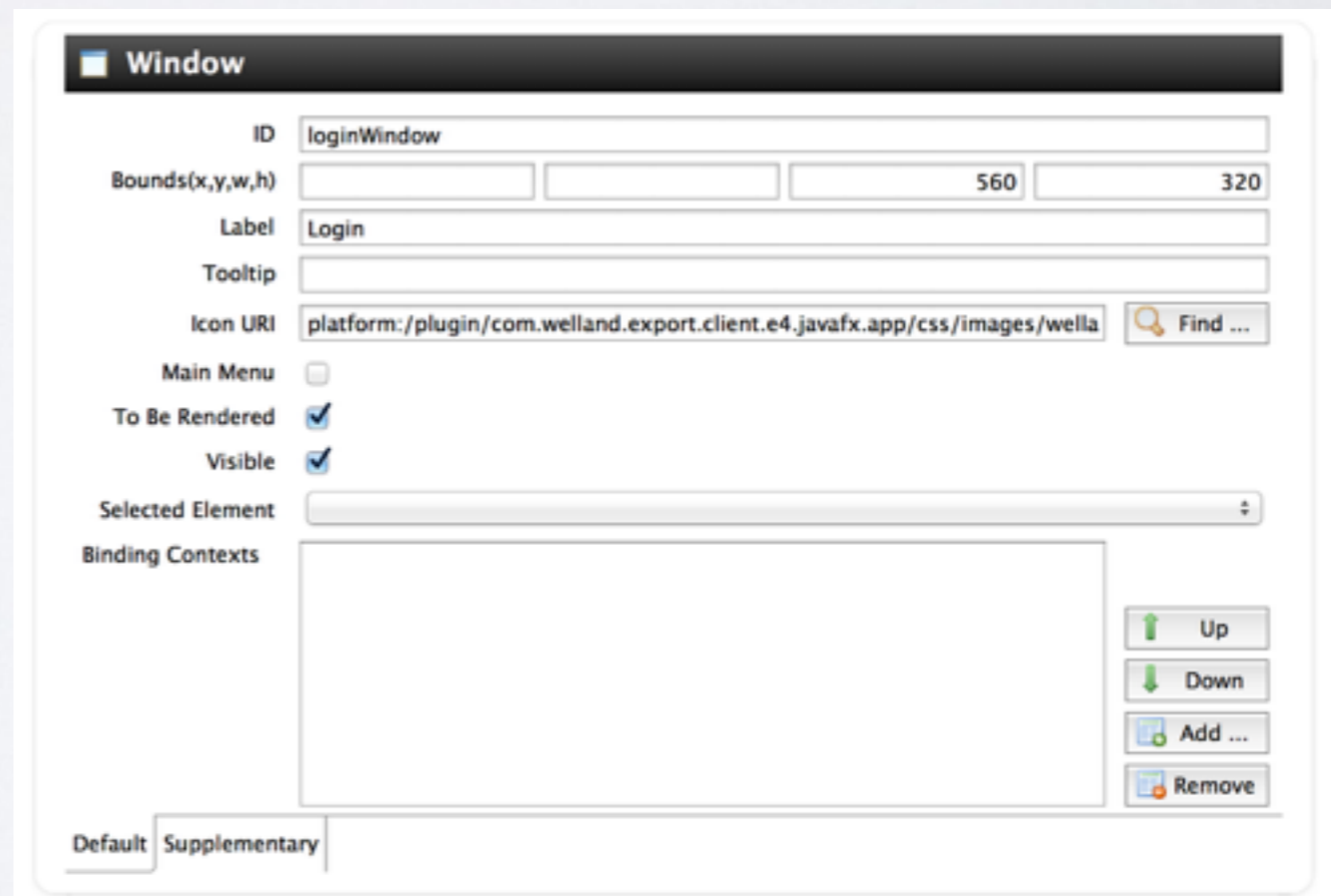
At the bottom of the dialog are two buttons: 'Cancel' and 'Login'.

The background interface includes a sidebar with navigation options: Configuration, Scheduling, Logs, Performance, and Advanced. The main area is divided into sections: 'Table List' (with a table header: NAME, DESCRIPTION, EXPORT?, TIME STAMP, LOG DELETES), 'Table Info' (with a 'Batch Control' section and placeholder text), and 'Fields' (with a table header: Physical ..., Field Name, Description, EXPORT?, SQL Name, SMV). The 'Fields' table is currently empty, displaying 'No content in table'. At the bottom of the main area are buttons for 'Delete', 'Save', and 'Refresh'.



# IMPLEMENTING A LOGIN DIALOG

- Defined as a *Window* in the application model
- Window contains a single *Part*
- Custom *Window* *Renderer* to set modality



# CUSTOM PERSPECTIVE SWITCHER

**Export Client - Cole's Local With AWS**

FILE EXPORT TABLE FIELD

**Configuration**  
Scheduling  
Logs  
Job History  
Performance  
Advanced

### Table List

NAME	DESCRIPTION	EXPORT?	TIME STAMP	LOG DELETES
CUSTOMER	CUSTOMER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PRODUCTS	PRODUCTS	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
STATES	STATES	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

### Table Info

Category	Count
Single	115
Multivalue	85
Associations	10

Single Multivalue Associations

### Export Options

Time Stamp  Log Deletions

ADVANCED

Character Number

### Fields

PHYSICAL ATTRIBUTE	FIELD NAME	DESCRIPTION	EXPORT?	CREATE INDEX	SQL NAME	SMV
0001	SAL	Salutation	<input type="checkbox"/>	<input type="checkbox"/>	SAL	S
0002	FNAME	First Name	<input type="checkbox"/>	<input type="checkbox"/>	FNAME	S
0002	FULL_NAME		<input type="checkbox"/>	<input type="checkbox"/>	FULL_NAME	S
0003	LNAME	Last Name	<input type="checkbox"/>	<input type="checkbox"/>	LNAME	S
0004	COMPANY	Company Name	<input type="checkbox"/>	<input type="checkbox"/>	COMPANY	S
0005	ADDR1	Address line 1	<input type="checkbox"/>	<input type="checkbox"/>	ADDR1	S
0006	ADDR2	Address line 2	<input type="checkbox"/>	<input type="checkbox"/>	ADDR2	S

DELETE SAVE REFRESH

# CUSTOM PERSPECTIVE SWITCHER

- Prototyped in SceneBuilder, defined in FXML
  - Radio Buttons with custom CSS
- Implemented as a ToolControl, in Window TrimBar
  - Simpler implementation than MPerspectiveStack renderer
- Listens to model for perspective changes
- Updates model when buttons are pressed

# CUSTOM PERSPECTIVE SWITCHER

Simple yet powerful CSS



```
-----
LeftTabs RadioButton
-----
**/
#perspectiveTabs .radio-button {
    -fx-font-size:15.0;
    -fx-alignment:center-left;
    -fx-text-fill:white ;
    -fx-background-color: linear-gradient(#1c689f 0.0%, #39a0cb 25.0%, #39a0cb 100.0%);
    -fx-background-radius: 0.0 0.0 0.0 0.0;
    -fx-background-insets: 1.0 10.0 0.0 0.0;
    -fx-cursor:hand;
}

#perspectiveTabs .radio-button:selected {
    -fx-font-size:15.0;
    -fx-font-weight: bold;
    -fx-alignment:center-left;
    -fx-text-fill:white ;
    -fx-background-color: linear-gradient(#1c689f 0.0%, #39a0cb 25.0%, #39a0cb 100.0%);
    /*-fx-background-radius: 0.0 0.0 0.0 0.0;*/
    -fx-background-insets:1.0 0.0 0.0 0.0;
    -fx-shape:"M0,0 L100,0 L106,50 L100,100 L0,100 z";
}

#perspectiveTabs .radio-button .radio{
    visibility:false;
}

#leftbg{
    -fx-background-color:linear-gradient(#ffffff 0.0%, #eeeeee 60.0%, #dddddd 100.0%);
    -fx-background-image: url("images/WellandLogo_24.png");
    -fx-background-repeat:no-repeat;
    -fx-background-position:bottom;
    -fx-background-insets: 0.0 10.0 0.0 0.0;
}
-----
```



# CUSTOM TITLE BAR AND MENUS

**Export Client - Cole's Local With AWS** FILE EXPORT TABLE FIELD

Configuration  
Scheduling  
Logs  
Job History  
Performance  
Advanced

### Table List

NAME	DESCRIPTION	EXPORT?	TIME STAMP	LOG DELETES
CUSTOMER	CUSTOMER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PRODUCTS	PRODUCTS	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
STATES	STATES	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

### Table Info

Category	Count
Single	11
Multivalue	9
Associations	1

### Export Options

Time Stamp  Log Deletions

ADVANCED

Category	Count
Character	95
Number	5

### Fields

PHYSICAL ATTRIBUTE	FIELD NAME	DESCRIPTION	EXPORT?	CREATE INDEX	SQL NAME	SMV
0001	SAL	Salutation	<input type="checkbox"/>	<input type="checkbox"/>	SAL	S
0002	FNAME	First Name	<input type="checkbox"/>	<input type="checkbox"/>	FNAME	S
0002	FULL_NAME		<input type="checkbox"/>	<input type="checkbox"/>	FULL_NAME	S
0003	LNAME	Last Name	<input type="checkbox"/>	<input type="checkbox"/>	LNAME	S
0004	COMPANY	Company Name	<input type="checkbox"/>	<input type="checkbox"/>	COMPANY	S
0005	ADDR1	Address line 1	<input type="checkbox"/>	<input type="checkbox"/>	ADDR1	S
0006	ADDR2	Address line 2	<input type="checkbox"/>	<input type="checkbox"/>	ADDR2	S

DELETE SAVE REFRESH

# CUSTOM TITLE BAR AND MENU



- e(fx)clipse support for custom window decoration
  - Set `efx.window.decoration.fxml` in model
- Custom Window renderer to populate icon, title and menus
- Application Addon to bind menus to perspectives

# CUSTOM TITLE BAR AND MENU



- Set tag on Menu in application model

The screenshot shows the application model tree on the left and the 'Menu' configuration panel on the right.

**Application Model Tree:**

- Application
  - Addons
  - Binding Contexts
    - BindingTables
  - Handlers
  - Commands
    - Command Categories
  - Windows and Dialogs
    - Trimmed Window - Export Client
      - Main Menu
        - Menu - FILE
        - Menu - EXPORT
        - Menu - TABLE
        - Menu - FIELD
      - Handlers
      - Windows and Dialogs
      - Controls
      - Shared Elements
      - TrimBars
    - Part Descriptors
    - Menu Contributions
    - Toolbar Contributions
    - Trim Contributions
    - Snippets

**Menu Configuration Panel:**

**Accessibility Phrase:**

**Tags:**

- limitPerspectives
- com.welland.export.client.perspectives.configuration
- com.welland.export.client.perspectives.advanced

**Persisted State:**

Key	Value

Default Supplementary

# CUSTOM TITLE BAR AND MENU

W Export Client - Cole's Local With AWS

FILE EXPORT TABLE FIELD

- Application Addon to bind menus to perspectives

```
public class PerspectiveMenuProcessingAddon {  
  
    @Inject @Optional  
    public void perspectiveChanged(  
        @UIEventTopic(UIEvents.ElementContainer.TOPIC_SELECTEDELEMENT) Event event) {  
        Object changedObj = event.getProperty(UIEvents.EventTags.ELEMENT);  
        if(changedObj instanceof MPerspectiveStack){  
            MPerspectiveStack perspectiveStack = (MPerspectiveStack)changedObj;  
            MPerspective perspective = perspectiveStack.getSelectedElement();  
            if (perspective != null && perspective.getContext() != null) {  
                MWindow window = perspective.getContext().getActive(MWindow.class);  
                MMenu mainMenu = window.getMainMenu();  
                for (MMenuElement element: mainMenu.getChildren()) {  
                    List<String> tags = element.getTags();  
                    if (tags.contains("limitPerspectives")) {  
                        element.setVisible(tags.contains(perspective.getElementId()));  
                    }  
                }  
            }  
        }  
    }  
}
```



# DEPENDENCY INJECTION TIPS

- Avoid generics
- Use `@Optional` when appropriate
- Use method injection with `@Optional` or dynamic objects
- Active selection  
`@Named(IServiceConstants.ACTIVE_SELECTION)`
- Use `@UIEventTopic` to receive notifications on the UI thread

# DEPENDENCY INJECTION TIPS

- Avoid generics

```
public class ExampleTableHandler {  
    @Execute  
    public void execute(  
        @Named(IServiceConstants.ACTIVE_SELECTION)  
        final List<Table> selectedTables) {  
        // this will break if the active selection becomes List<Foo>  
        for (Table table: selectedTables) {  
            // do something with the table  
        }  
    }  
}
```

```
public class ExampleTableHandler {  
    @Execute  
    public void execute(  
        @Named(IServiceConstants.ACTIVE_SELECTION) final Table[] selectedTables) {  
        // do something with the tables  
    }  
}
```

# DEPENDENCY INJECTION TIPS

- Use `@Optional` when injected object may not be available at construction
- Use methods with `@Optional`

```
@Inject
@Optional
public void populate(ExportOperations newOperations){
    // Save for later
    this.operations = newOperations;
    // Do something useful with the operations
}
```

# DEPENDENCY INJECTION TIPS

- Active selection

`@Named(IServiceConstants.ACTIVE_SELECTION)`

- Different semantics than RCP 3.x
- Useful for popup menus and master/detail views
- Consider custom `@Named` variables
- Only one part's selection can be active
- `@Inject` method not called when active part changes



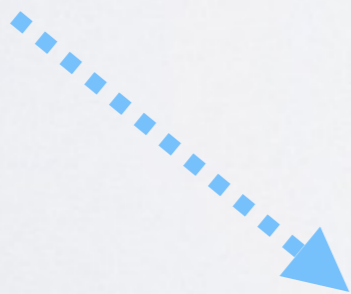
# DEPENDENCY INJECTION TIPS

- Use `@UIEventTopic` to receive notifications on the UI thread
  - Post notifications using `IEventBroker`
  - Pub-Sub pattern enables decoupled code
  - Platform handles calling subscribers on the UI Thread

```
@Inject
IEventBroker eventBroker;

protected IStatus runInJobThread(IProgressMonitor monitor) {
    operations.add(table);
    eventBroker.post(EventConstants.TOPIC_NEW, table);
    return Status.OK_STATUS;
}
```

**Publish**



**Subscribe**

```
@Inject
@Optional
public void insert(@UIEventTopic(EventConstants.TOPIC_NEW) Table table) {
    // New data added, add to our TableView data to refresh UI
    data.add(table);
}
```

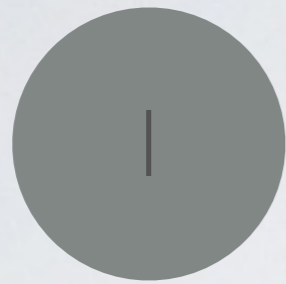
# RESOURCES/ACKNOWLEDGMENTS

- <http://eclipse.org/efxclipse>
- <http://wiki.eclipse.org/Efxclipse/Tutorials>

QUESTIONS?



# EVALUATE THIS SESSION



Sign-in: [www.eclipsecon.org](http://www.eclipsecon.org)



Select session from schedule



Evaluate:

