

**ORACLE®**

# **Mastering Customization and Personalization in Oracle ADF and WebCenter Applications**

Frank Nimphius  
Senior Principal Product Manager  
Oracle Application Development Tools

“It is not the strongest of the species that survives, nor the most intelligent that survives. It is the one that is the most adaptable to change.”

Charles Darwin

# Types of customization

- Rebranding
  - Change look and feel to match company or market identity
  - Design time
  - By Developer, Analyst or UI Designer
- Personalization
  - Runtime
  - By end user
- Seeded Customization
  - Design Time
  - By customizer
  - Using JDeveloper Customization Role

# Application Development Framework (ADF)

- Provides
  - ADF Faces Rich Client
  - Data Visualization Components
  - ADF Controller
  - ADF Binding Layer
  - ADF Business Components
- Metadata on all Tiers
  - XML is King
- Benefits of metadata frameworks
  - Customizable
  - Declarative development
  - External administration

# Oracle Metadata Services (MDS)

- The framework for handling application customizations
- Uses a single base application
- Applies customizations on top of the base.
- Customizations are created
  - At design time using JDeveloper (seeded customization)
  - At runtime by users (personalization, Web Center)
- Customizations are stored in a metadata repository and merged with base metadata at runtime
- Base application can be updated without affecting customization

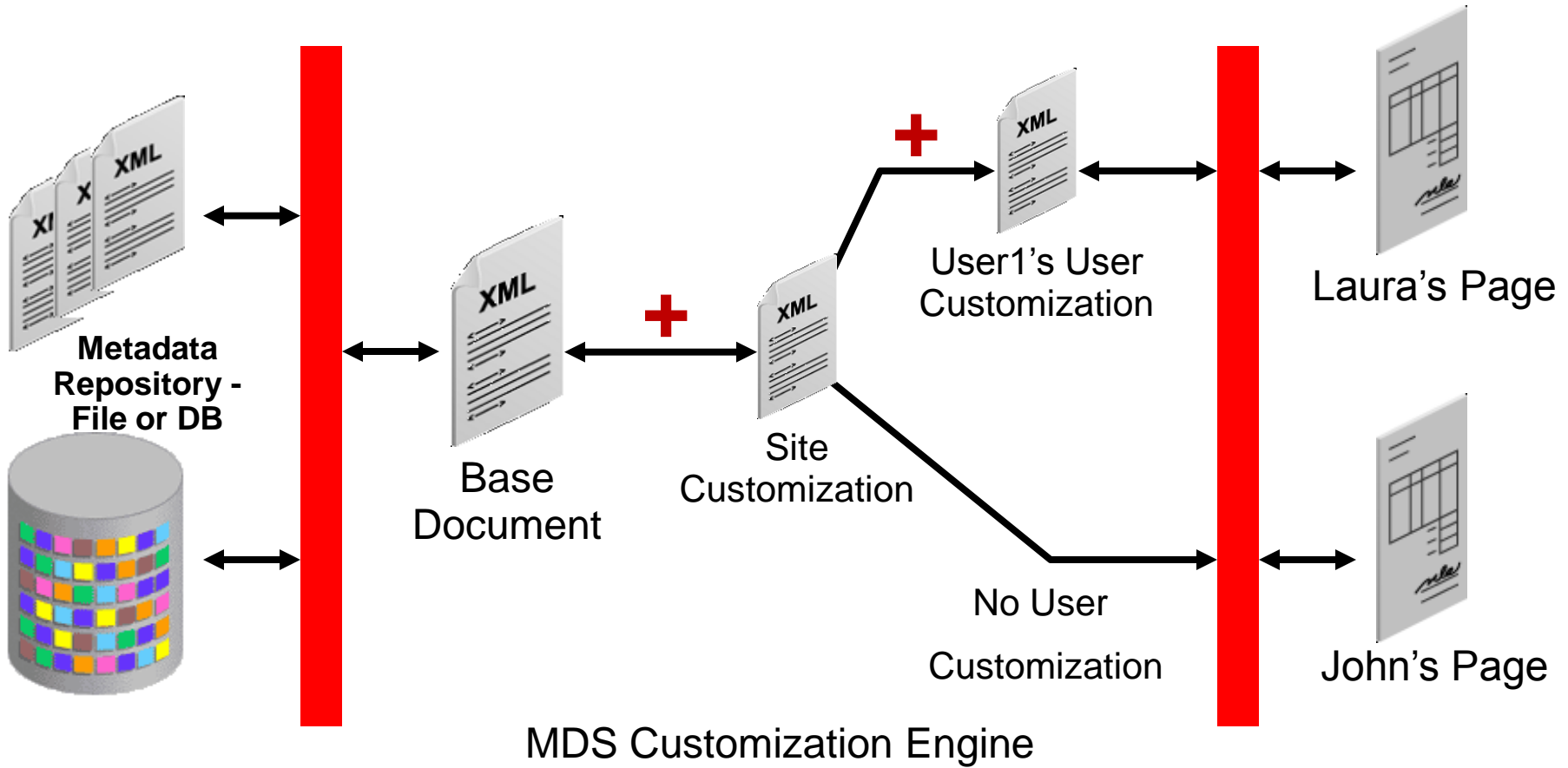
# How customization is applied

- Customizations can be applied
  - for all users
    - site level customizations
  - Dynamically based on programmatically defined criteria
    - who the user is
    - what organization users belong to
    - Current day time
    - ... just anything you can do in Java
- Customizations are a set of instructions that change the underlying metadata

# What is getting customized

- ADF Business Components
- ADF Model
- ADF Controller
- ADF View
- ADF Library Artifacts

# Metadata services customization



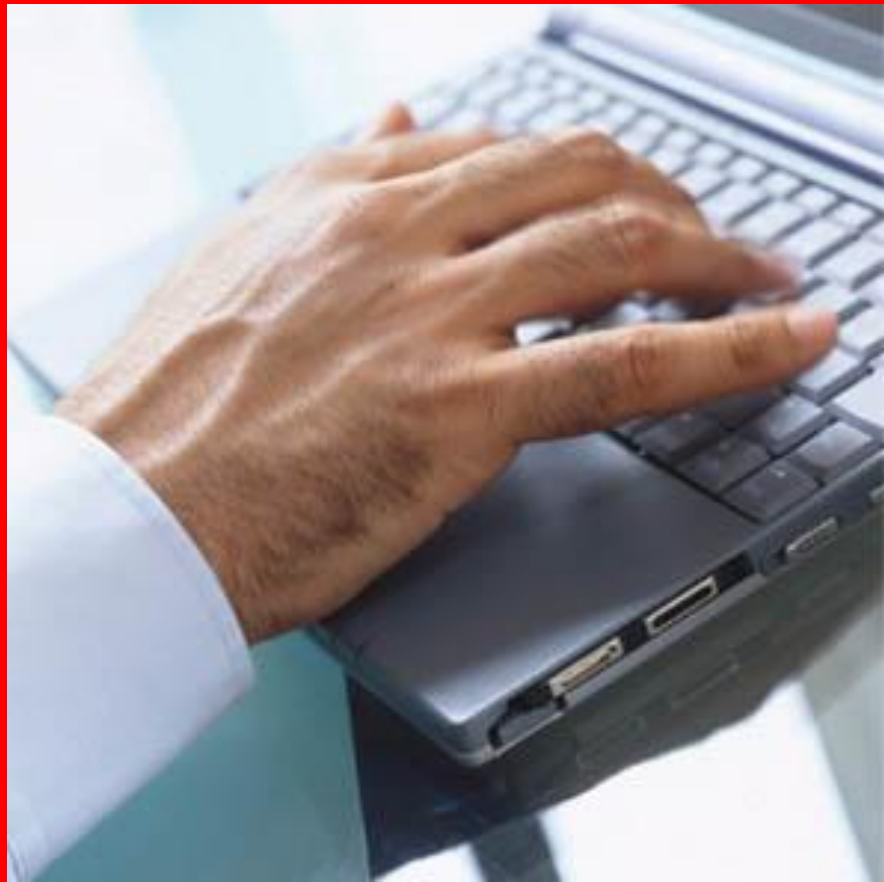




# Product Demonstration

---

## Personalization with MDS



**Let's Get Technical**

# Change Persistence Framework

- The foundation of ADF application customization and end user personalization
- Persists changes for ADF Faces components and restores them
- User personalization based on end user interaction
  - Changes to an attribute value
  - Addition or removal of a child component
  - Re-ordering of children within the same parent
  - Move of child component to a different parent

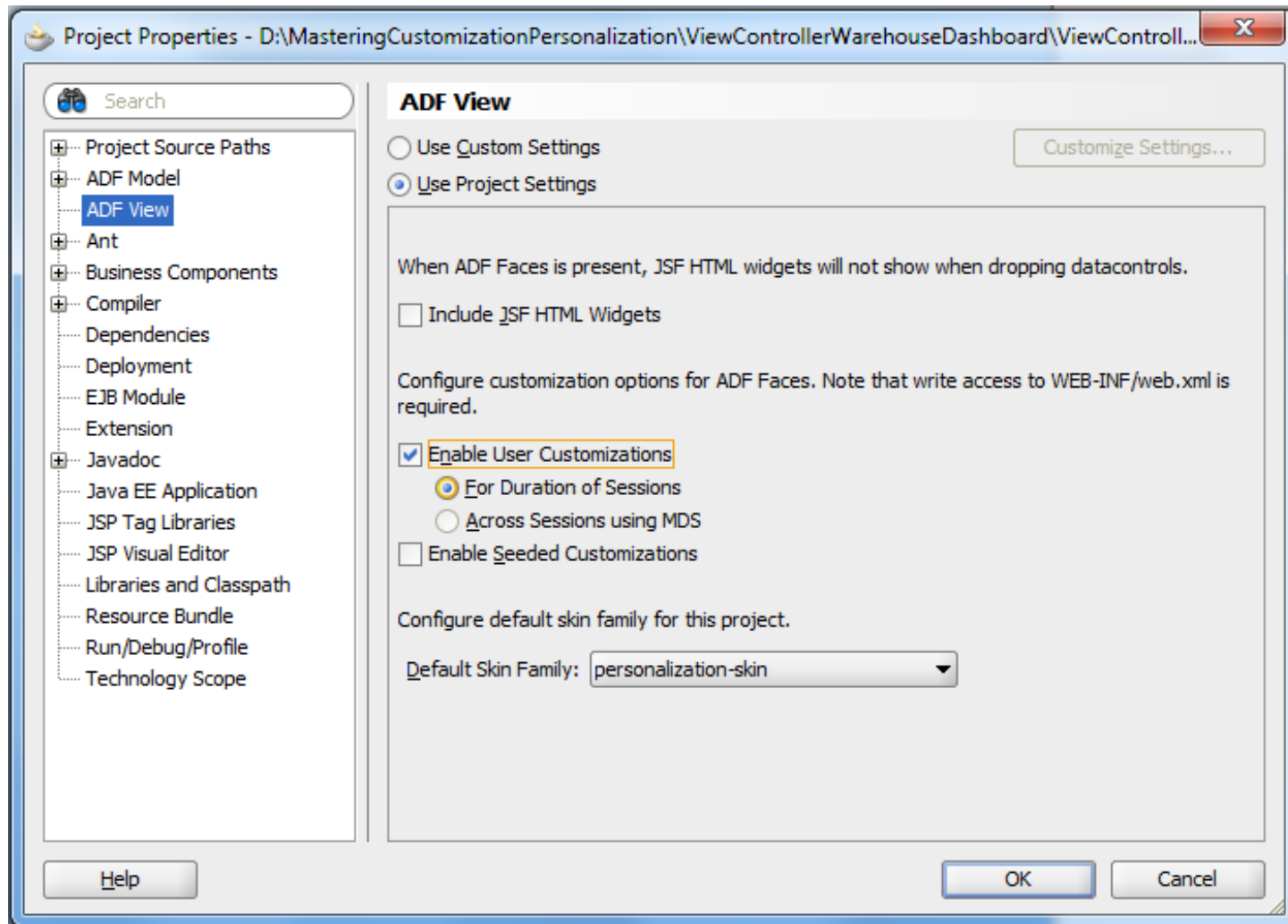
# Change Persistence Framework

- Changes are persisted as
  - Component changes in the session
- Types of Change Persistence
  - Implicit
    - Attribute changes are handled by ADF Faces runtime code
    - Applies to component specific list of attributes
  - Explicit
    - Developer could explicitly add any attribute changes to the Change Manager
    - For example in response to an event from a different component

# Change Persistence API

- Base packages
  - org.apache.myfaces.trinidad.change
  - oracle.adf.view.rich.change
- Change Manager
  - Accessible from AdfFacesContext
  - SessionChangeManager
  - MDSDocumentChangeManager
- Changes can be applied anywhere in the JSF lifecycle
  - UI tree must be available
  - Create instance of ComponentChange and DocumentChange and add it to ChangeManager

# Enabling Change Persistence



# Personalization with MDS

- Extends the ChangePersistence functionality, adding metadata repository support
- Persists a selected set of components and attributes to save changes beyond the end of a session
- Allows developers to exclude attribute from getting persisted
  - Globally
  - On a per component base

# MDS Persistence at Runtime

- Any changes made during a session are recorded in a session variable
  - Data structure indexed according to the view ID
- Subsequent Create View or Restore View apply changes the order as they were added
  - Persistence is "active" after subsequent request
- Changes in the session that are persisted to MDS mutate the MDS Document Object Model
- Phase listener triggers a commit on the MDS session
- Oracle's JSP engine merges stored changes to the base document



# Configuring MDS Persistence

The screenshot shows the Oracle JDeveloper IDE interface. On the left, the 'Application Resources' tree is visible, with 'ADF META-INF' expanded and 'adf-config.xml' selected. The main editor displays the 'adf-config.xml' file. The 'MDS Configuration' section is active, showing the 'Tag Configuration' tab. The 'Tag Library URI' is set to 'http://xmlns.oracle.com/ADF/faces/rich'. The 'Tags' list includes 'column', which is selected. The 'Tag Attributes: column' table shows attributes like 'displayIndex', 'frozen', 'noWrap', 'selected', 'visible', and 'width', with checkboxes for 'Persist Changes'. A dropdown menu is open, showing a list of ADF tags.

**Tag Configuration**

Tag Library URI: `http://xmlns.oracle.com/ADF/faces/rich`

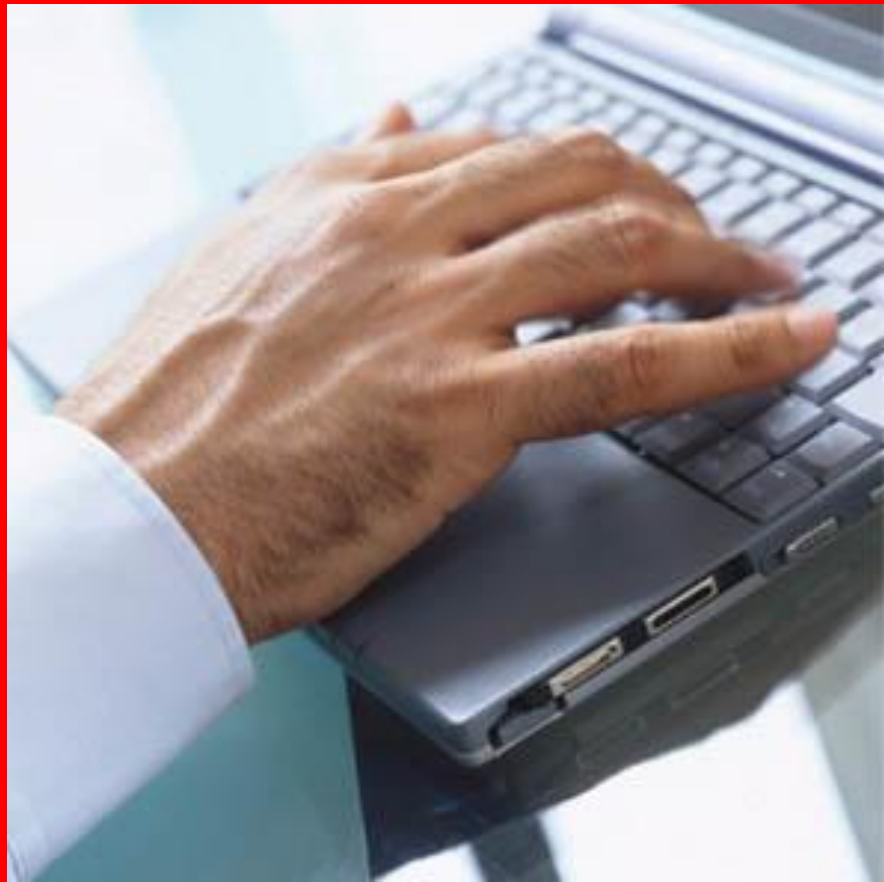
**Tags**

Name
table
panelSplitter
column
panelAccordion
showDetailItem
panelTabbed

**Tag Attributes: column**

Name	Persist Changes
displayIndex	<input checked="" type="checkbox"/>
frozen	<input checked="" type="checkbox"/>
noWrap	<input checked="" type="checkbox"/>
selected	<input type="checkbox"/>
visible	<input checked="" type="checkbox"/>
width	<input checked="" type="checkbox"/>

activeCommandToolBarButton  
calendar  
commandButton  
commandImageLink  
commandLink  
commandMenuItem  
commandNavigationItem  
commandToolBarButton  
dialog  
panelBox  
panelWindow  
richTextEditor  
showDetail  
showDetailHeader



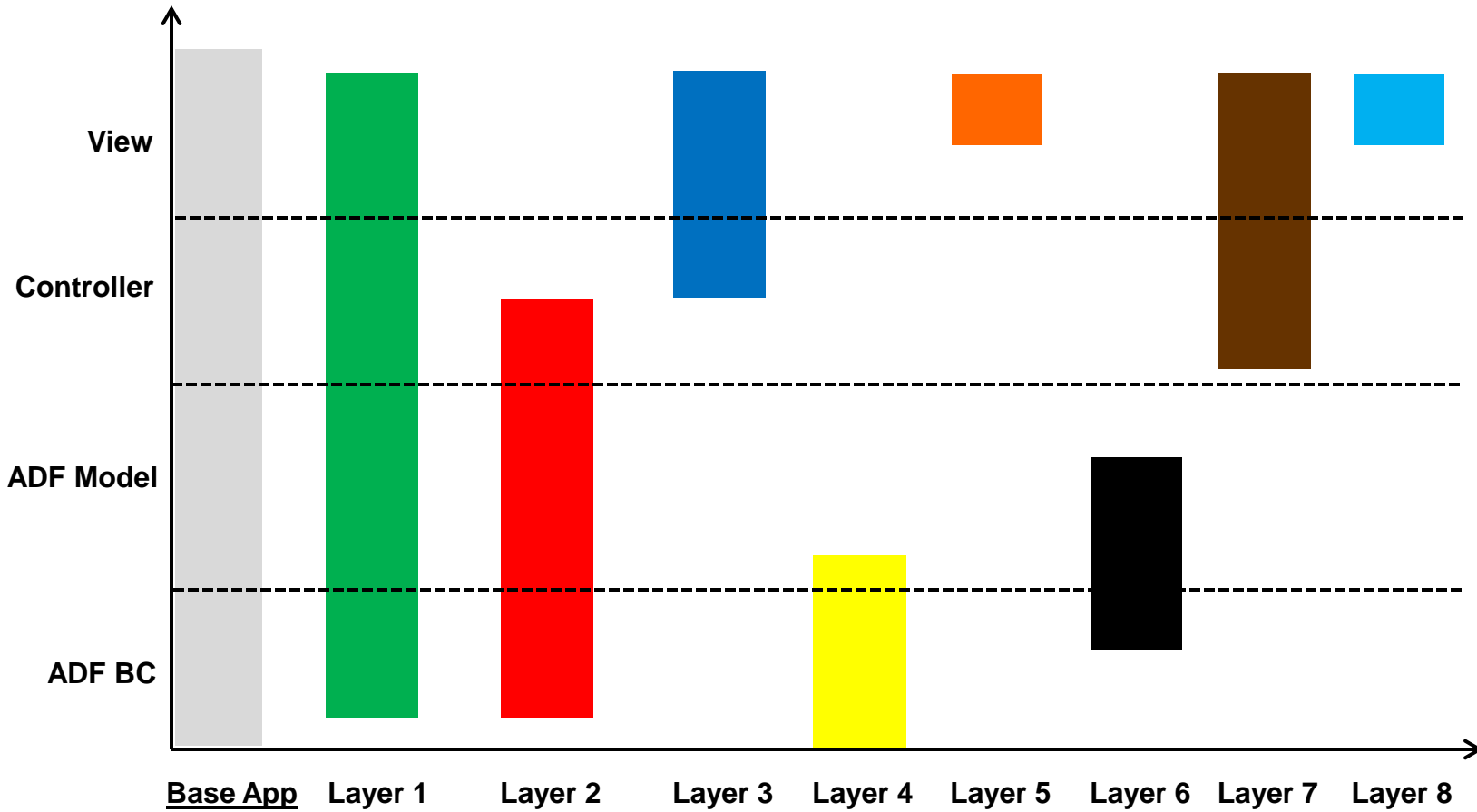
# How-to: The Seeded Thing

# Steps to build seeded customizations

- Build the base application
- Enable MDS and seeded customization
- Optionally, create your own customization classes
  - Name
  - Cache hint
  - Layer
  - Has access to MDS, JSF, Servlet and Request context
  - May leverage ADF Security
  - A lot of fun
- Configure customization classes in `adf-config.xml`

# Technical at Designtime ...

## Application



# Beautiful at Runtime !



# Customization Classes already Available in ADF

- UserCC
  - Returns the Principal name for authenticated user
  - Use for end user personalization
- SiteCC
  - Returns "site" as layer value
- ADFRolesCC
  - Returns enterprise and application role names
    - Use enterpriseroles for site level customization
    - Use application roles for seeded customizatio

# Custom Customization Classes

- Have access to the Http and MDS Session object
- Can use ADF Security to check user permission before applying customization
- Return a layer name and one to many values
  - Value(s) determine(s) the customization layer to apply for the customization layer
- Extend abstract class CustomizationClass

# Example: PreferencesCC

```
public class PreferencesCC extends CustomizationClass {
    public static final String ADVANCED_MODE = "advanced";
    public static final String NORMAL_MODE = "normal";

    public PreferencesCC() {
        super();
    }
    public CacheHint getCacheHint() {
        return CacheHint.MULTI_USER;
    }

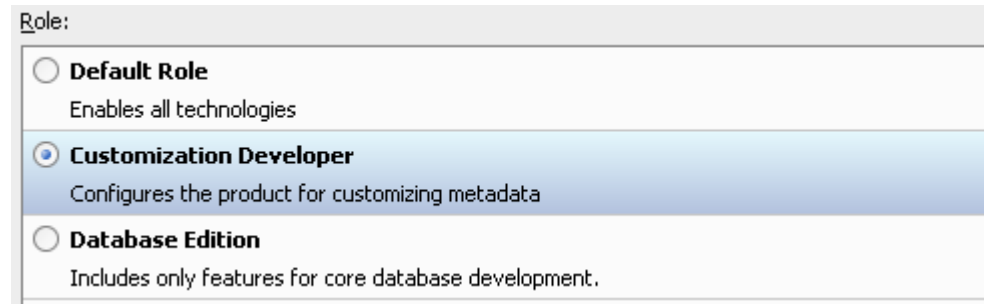
    public String getName() {
        return PreferenceLayerName;
    }

    public String[] getValue(RestrictedSession restrictedSession,
                             MetadataObject metadataObject) {
        //returns new String[] {"normal"}, String[] {"advanced"}
        //or new String[0]
        return getUserPreference();
    }
}
```



# Seeded Customizations

- Customizations are made with Oracle JDeveloper Customization role



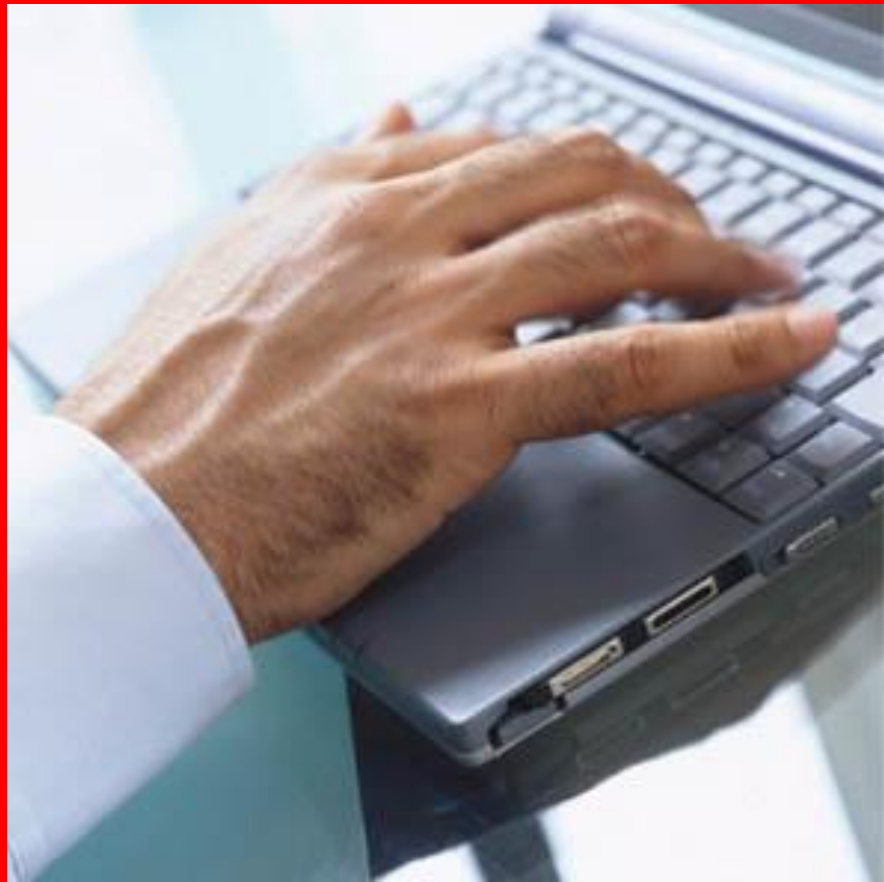
- Customization are made for each defined layer
- Developer experience is the same as creating base metadata
- Only existing metadata is customizable
  - No Java objects



# Product Demonstration

---

## Seeded Customization



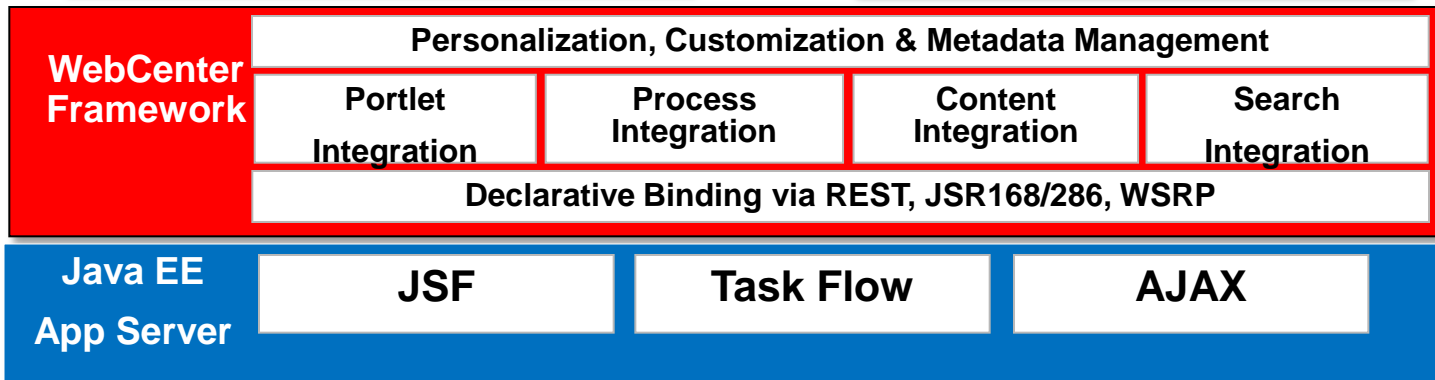
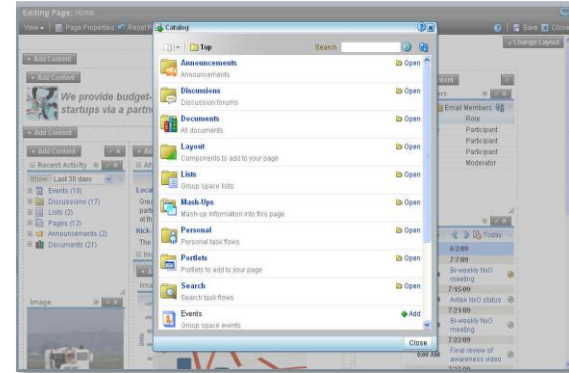
# The Bang for the Bucks WebCenter

# WebCenter Framework

## JDeveloper



## Browser



**Databases Systems**



**Business Process**



**Content Repositories**



**Custom/LOB Apps**



**Web Services**



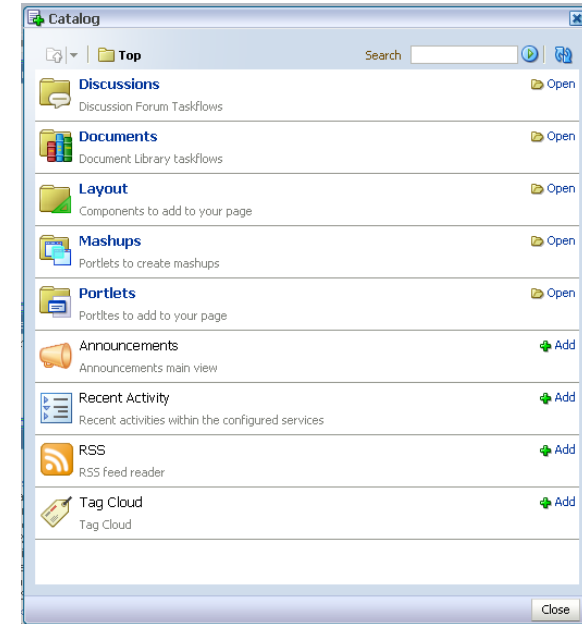
**Business Intelligence**



**Business Applications**

# Oracle Composer

- Enables Runtime Editing
- Runtime Customization & Personalization
- Design Time at Runtime (DT@RT)
- Manage Page Content with Resource Catalog
- Layered Customization with MDS
- Sandbox: support for transaction, concurrency

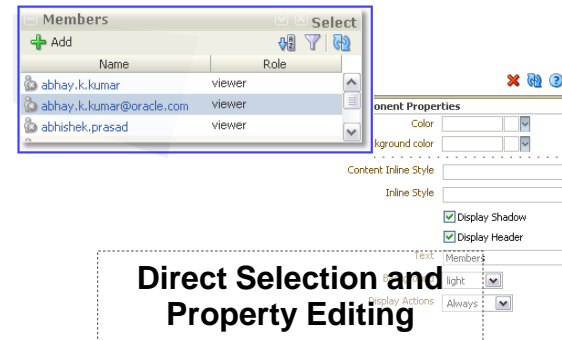


Search/Browse and Add Content

Submit to  
In-line Edit



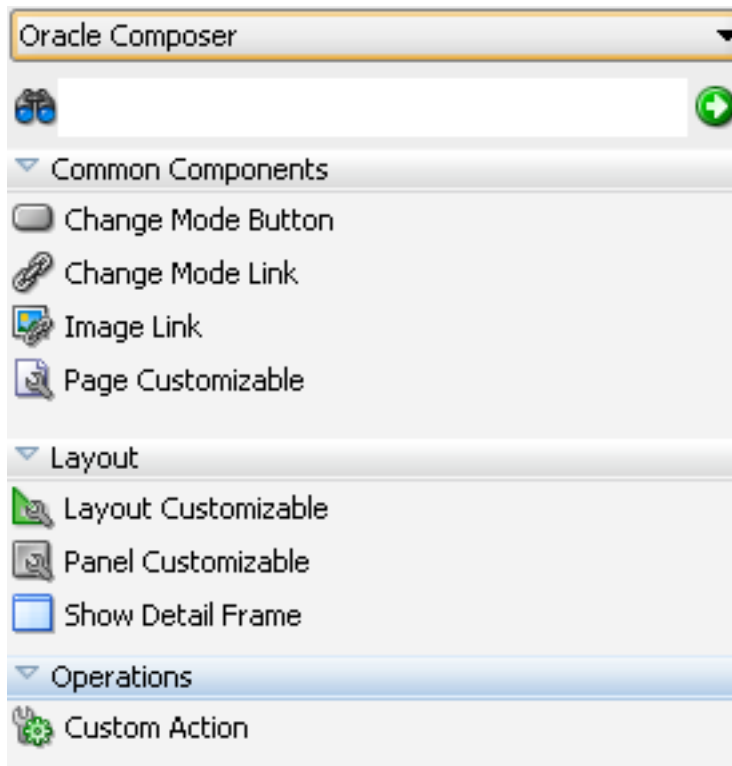
Drag and Drop



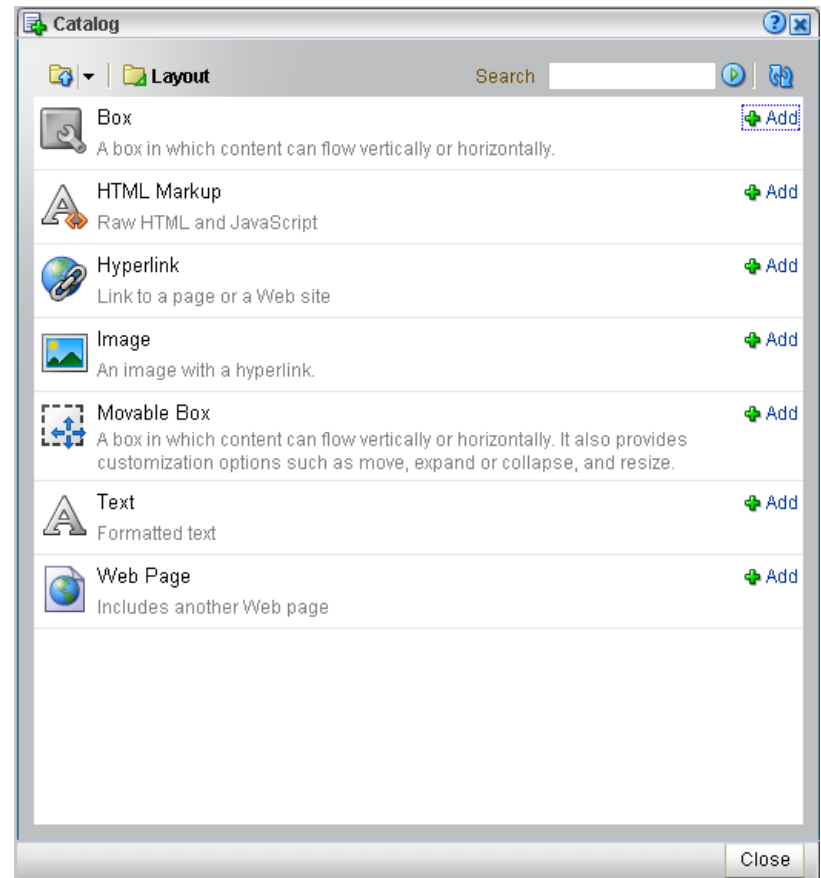
Direct Selection and Property Editing

# Customizable Components

## Design Time



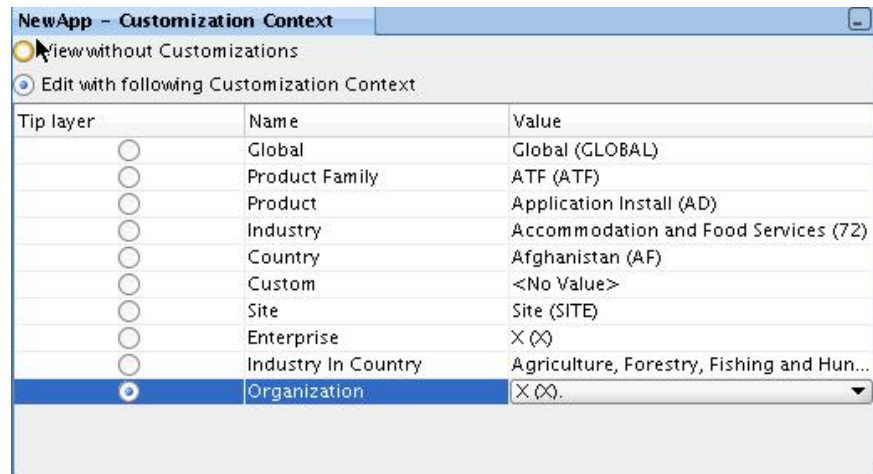
## Run Time



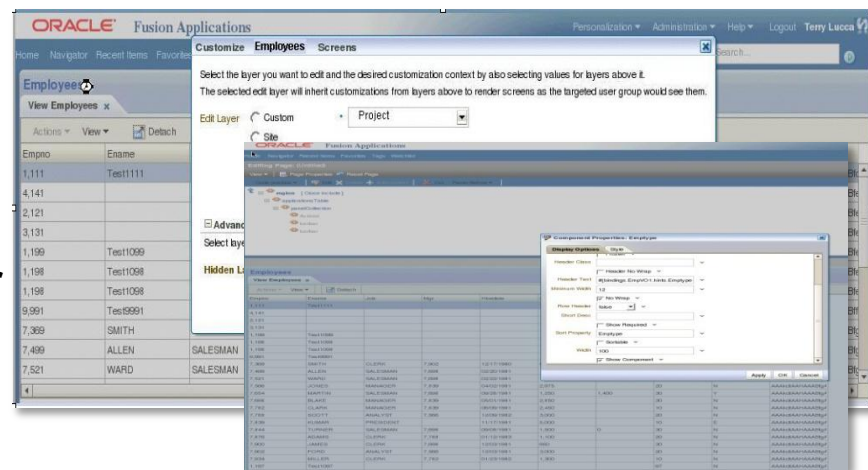
# Customizations, MDS, and Fusion Applications

- All pages are Composer/MDS-enabled for upgrade-safe customizations
  - administrator/business-user/end-user
- 11 different customization layers
  - Some layers used internally in development to provide seeded customizations (ex: “country” customizations for HCM modules, “product” customizations for common modules)
  - Admins can choose which layer to customize in both JDev and Composer
  - Ex: Create industry specific customizations for a CRM module
- Want to know more?
  - **S315481: Customizing Fusion Applications with WebCenter Composer**  
Wednesday, 10-11AM, Hotel Nikko

## Customizations in JDeveloper



## Customizations in Composer / Browser





# **Product Demonstration**

---

## **WebCenter Personalization and Customization**



# Summary

- Common misconception corrected: It is not the strongest that survives but the fittest
- Personalization and Customization is a necessity in Web 2.0
- ADF And WebCenter have it all!
  - Implicit and explicit UI personalization
  - Seeded customization

**ORACLE®**

# Related Oracle Press Books



- Quick Start Guide to **Oracle Fusion Development**
- **Oracle JDeveloper 11g** Handbook
- **Oracle Fusion** Developer Guide



**ORACLE®**