CE206
Migrating Web Dynpro Applications From SAP NetWeaver 7.0 to CE 7.1
Contributing Speakers

Bertram Ganz
SAP NetWeaver UI, Web Dynpro Development, SAP AG

Marco Ertel
NetWeaver UI Product Management, SAP AG

Chris Whealy
NetWeaver RIG, SAP (UK) Ltd
Learning Objectives

As a result of this workshop, you will be able to:

- Understand the Web Dynpro Compatibility Contract in SAP NetWeaver CE 7.1
- Migrate existing Web Dynpro Development Components to SAP NetWeaver CE 7.1
- Understand the changes of the New Web Dynpro component model
- Understand Component Controller Patterns and Anti-Patterns
- Migrate existing Web Dynpro components to the new component model utilizing the Web Dynpro component migration tool (Cheat Sheet)
Migrating Web Dynpro Components

Web Dynpro Backward Compatibility

Migrating Web Dynpro DCs

Migrating Web Dynpro Components

Summary
Web Dynpro Backward Compatibility

Migrating Web Dynpro DCs

Migrating Web Dynpro Components

Summary
Backward Compatibility

You can still run existing Web Dynpro applications developed in SAP NetWeaver 04 and SAP NetWeaver 7.0 within an SAP NetWeaver CE 7.1 environment.

Source Compatibility

Your Web Dynpro application will run after *re-generation, re-compilation* and *re-deployment* of all Web Dynpro DCs it consists of requiring no source code adaptations.

**Binary Compatibility:** Not supported
Migrating Web Dynpro Components

Migrating Web Dynpro DCs

Web Dynpro Backward Compatibility

Summary
# Migrating Web Dynpro Applications (DCs) to SAP NetWeaver CE 7.1

## Web Dynpro DC Migration Process

<table>
<thead>
<tr>
<th>Set up Development Infrastructure</th>
<th>Migrate DC Dependencies</th>
<th>Rebuild Web Dynpro DCs</th>
<th>Deploy and Run</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP NetWeaver 7.0 SP13 NWDI</td>
<td>Create Active Projects for Web Dynpro DCs to be migrated</td>
<td>Eliminate all unsupported coding (e.g. typecasts to &quot;forbidden&quot; internal Web Dynpro FPIs).</td>
<td>Redeploy your Web Dynpro DCs</td>
</tr>
<tr>
<td>Valid Track for SAP NetWeaver CE 7.1 is set up (the required Net-Weaver SCs are added)</td>
<td>Repair Project Structure and Classpath of Web Dynpro DCs in Web Dynpro Explorer</td>
<td>Based on the new Web Dynpro Facade concept such code cannot be compiled any more.</td>
<td>Run your Web Dynpro Application</td>
</tr>
<tr>
<td></td>
<td>Refactor/Delete Broken or Deprecated Public Parts of Used Library DCs</td>
<td>Rebuild your Web Dynpro DCs</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Migration Step 1 – Set up Development Infrastructure

## Web Dynpro DC Migration Process Step

### Set up Development Infrastructure

<table>
<thead>
<tr>
<th>Required Components</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP NetWeaver 7.0 SP13 NWDI</td>
<td>Required for development infrastructure.</td>
</tr>
<tr>
<td>Valid Track for SAP NetWeaver CE 7.1 set up</td>
<td>Needed for development and production infrastructure.</td>
</tr>
</tbody>
</table>

- **A SAP NetWeaver Developer Studio for SAP NetWeaver CE 7.1** must be used as a design-time environment.
- **SAP NetWeaver CE 7.1** does not come with an installation of the SAP NetWeaver Development Infrastructure (NWDI).

Two main scenarios for a valid development and production infrastructure exit:

- Use an existing SAP NetWeaver 7.0 Development Infrastructure with valid tracks for SAP NetWeaver CE 7.1 set up.
- Use a third party development and production infrastructure.
## Migrate DC Dependencies

**Web Dynpro DC Migration Process Step**

<table>
<thead>
<tr>
<th>Migrate DC Dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Create Active Projects for Web Dynpro DCs to be migrated</td>
</tr>
<tr>
<td>- Repair Project Structure and Classpath of Web Dynpro DCs in Web Dynpro Explorer</td>
</tr>
<tr>
<td>- Refactor/Delete Broken or Deprecated Public Parts of Used Library DCs</td>
</tr>
</tbody>
</table>

- The initial Library DC dependencies for Web Dynpro DCs have changed in SAP NetWeaver CE 7.1
- A new Web Dynpro Library DC (Web Dynpro Façade) was introduced only exposing the public Web Dynpro Runtime APIs
  - Forbidden typecasts to invoice internal Web Dynpro FPI (Framework Programming Interfaces) are no longer possible
- Manually added DC dependencies might have changed also. They must be migrated to the new DC Public Parts manually
Migration Step 2 – Create Active Projects

Web Dynpro DC Migration Process Step

Migrate DC Dependencies

- Create Active Projects for Web Dynpro DCs to be migrated
- Repair Project Structure and Classpath of Web Dynpro DCs in Web Dynpro Explorer
- Refactor/Delete Broken or Deprecated Public Parts of Used Library DCs

Open Development Infrastructure Perspective
Select Web Dynpro DCs of your application to be migrated
Select context menu item “Sync / Create Project – Create Active Project”

Diagram showing the steps in the migration process.
Migration Step 2 – Repair Projects

Web Dynpro DC Migration Process Step

Migrate DC Dependencies

- Create Active Projects for Web Dynpro DCs to be migrated
- Repair Project Structure and Classpath of Web Dynpro DCs in Web Dynpro Explorer
- Refactor/Delete Broken or Deprecated Public Parts of Used Library DCs
- Web Dynpro Explorer opens automatically
- Select Web Dynpro DCs of your application to be migrated
- Select context menu item “Repair – Project Structure and Classpath”

- Migrate DC Dependencies
- Web Dynpro Explorer opens automatically
- Select Web Dynpro DCs of your application to be migrated
- Select context menu item “Repair – Project Structure and Classpath”
Migration Step 2 – Repair Project Result

Web Dynpro DC Migration Process Step

Migrate DC Dependencies

- Create Active Projects for Web Dynpro DCs to be migrated
- Repair Project Structure and Classpath of Web Dynpro DCs in Web Dynpro Explorer
- Refactor/Delete Broken or Deprecated Public Parts of Used DCs

After “Repair Projects” new Library DC usages are added
- Deprecated DC usages still exist and will be deleted in next step

- Repair Project Structure and Classpath of Web Dynpro DCs in Web Dynpro Explorer
- Refactor/Delete Broken or Deprecated Public Parts of Used DCs

Required DCs
Specify the list of DCs required for this DC:
- com.sap.ai.proxy.framework
- com.sap.ai.util.misc
- com.sap.exception
- com.sap.nw.jco
- tc/cmi
- tc/ddic/ddicruntime
- tc/ddic/metamodel/content
- tc/logging
- tc/wd/webdynpro
- tc/wdp/metamodel/content

Required DCs
Specify the list of DCs required for this DC:
- com.sap.ai.proxy.framework
- com.sap.ai.util.misc
- com.sap.exception
- com.sap.nw.jco
- tc/cmi
- tc/ddic/ddicruntime
- tc/ddic/metamodel/content
- tc/ddic/metamodel/content
- tc/ddic/runtime/facade
- tc/logging
- tc/wd/spi
- tc/wd/webdynpro
- tc/wdp/metamodel/content
Migration Step 2 – Delete Deprecated Library DCs

Web Dynpro DC Migration Process Step

Migrate DC Dependencies

- Create Active Projects for Web Dynpro DCs to be migrated
- Repair Project Structure and Classpath of Web Dynpro DCs in Web Dynpro Explorer
- Delete Deprecated Public Parts of Used Library DCs

Open Development Infrastructure Perspective
- Open “Dependencies” Tab
- Repeatedly select single Web Dynpro DC
- Delete Deprecated Library DC usage relations

- Open Development Infrastructure Perspective
- Open “Dependencies” Tab
- Repeatedly select single Web Dynpro DC
- Delete Deprecated Library DC usage relations
Migration Step 2 – Set Build Time Dependency

Web Dynpro DC Migration Process Step

Migrate DC Dependencies

- Open Development Infrastructure Perspective
- Open “Dependencies” Tab
- Repeatedly select single Web Dynpro DC
- Set “Build Time” dependency of Public Part gci in required Web Dynpro Library DC tc/wd/api

Prior to SAP NetWeaver CE 7.1 SP3 Only

Set Build Time Dependency of PP gci in Web Dynpro Library DC tc/wd/api

Open Development Infrastructure Perspective

Open “Dependencies” Tab

Repeatedly select single Web Dynpro DC

Set “Build Time” dependency of Public Part gci in required Web Dynpro Library DC tc/wd/api

Open Development Infrastructure Perspective

Open “Dependencies” Tab

Repeatedly select single Web Dynpro DC

Set “Build Time” dependency of Public Part gci in required Web Dynpro Library DC tc/wd/api

Prior to SAP NetWeaver CE 7.1 SP3 Only

Set Build Time Dependency of PP gci in Web Dynpro Library DC tc/wd/api
Migration Step 2 – Migrate Manually Added DC Usages

Web Dynpro DC Migration Process Step

Migrate DC Dependencies

- Create Active Projects for Web Dynpro DCs to be migrated
- Repair Project Structure and Classpath of Web Dynpro DCs in Web Dynpro Explorer
- Refactor broken or deprecated Public Parts of used External Library DCs or used Web Dynpro DCs

Open Development Infrastructure Perspective
Open “Dependencies” Tab
Repeatedly select single Web Dynpro DC
Migrate deprecated Public Part usages of manually added External Library DCs or other Web Dynpro DCs
Apply Quickfix (CTRL + 1) in case a Public Part successor is defined

Press CTRL + 1 to select an alternative for the deprecated PP
(The Component Properties view must have the focus for the shortcut to work)
Open Development Infrastructure Perspective

Open **Dependencies** Tab

Select Web Dynpro DCs of your application to be migrated

Select context menu item **“Build”**

- Eliminate all unsupported coding (e.g. typecasts to "forbidden" internal Web Dynpro FPIs).
- Based on the new *Web Dynpro Facade concept* such code cannot be compiled any more.
- Rebuild your Web Dynpro DCs
Migration Step 4 – Deploy and Run Application

Web Dynpro DC Migration Process Step

Deploy and Run Application

- Open Development Infrastructure Perspective
- Open “Dependencies” Tab
- Select Web Dynpro DCs of your application to be migrated
- Select context menu item “Deploy”
**Migration Step 4 – Deploy and Run Application**

**Web Dynpro DC Migration Process Step**

**Deploy and Run Application**

- **Open Web Dynpro Explorer**
- **Open Web Dynpro DC comprising your Web Dynpro application entity**
- **Select context menu item “Run” for application entity**

```
LocalDevelopment/techd/cd354/sob/vers05/custui
LocalDevelopment/techd/cd354/sob/vers05/myui
LocalDevelopment/techd/cd354/sob/vers06/locator
LocalDevelopment/techd/cd354/sob/vers06/root
```

**Diagram:**
- Web Dynpro
  - Applications
    - SalesOpportunitiesApp
  - Models
  - Component Interactions
  - Components
  - Dictionaries
  - Resources
  - Open
  - Open Data Modeler
  - Record
  - Delete
  - Copy
  - Ctrl+C
  - Deploy new Archive and Run
  - Run
  - Refactor
Migrating Web Dynpro DCs to SAP NetWeaver CE 7.1

DEMO
Migrating Web Dynpro Components

Enhanced Web Dynpro Component Model

How To Migrate Web Dynpro Components

SAP Recommendations on Component Migration

Further Migration Aspects
# Web Dynpro Component Migration at a Glance

<table>
<thead>
<tr>
<th>Question</th>
<th>Answer</th>
</tr>
</thead>
</table>
| What is Web Dynpro component migration? | The Web Dynpro component programming model was significantly changed in SAP NetWeaver CE 7.1!  
Existing Web Dynpro component created in NetWeaver 04 or 7.0 can be migrated to the new component model. |
| What about Web Dynpro component compatibility? | SAP NetWeaver CE 7.1 is **source-compatible** to all your existing Web Dynpro components created in SAP NetWeaver 04 or 7.0.  
After repair, rebuild and deployment of Web Dynpro DCs your can run existing components in CE 7.1. |
| Must all my Web Dynpro components be migrated? | No! Based on the given **source-compatibility** you can run existing Web Dynpro components without migration.  
Under certain conditions component migration should or must be applied (e.g. to use new functions). |
| How can I migrate my Web Dynpro components? | Read the *Component Migration Process Steps* described in this presentation  
Run the **Web Dynpro component migration tool** (Cheat Sheet) assisting you in successfully migrating existing components to the new component model. |
Migrating Web Dynpro Components

Enhanced Web Dynpro Component Model

How To Migrate Web Dynpro Components

SAP Recommendations on Component Migration

Further Migration Aspects
There are significant changes in the Web Dynpro component programming model in SAP NetWeaver CE 7.1.

- The changes are source-compatible. Existing Web Dynpro components can be rebuilt, rearchived, redeployed and will then run.
- Old components will look visibly different (grey component icons) and will further be based on the old programming model.
- Old Web Dynpro components will further be supported.
- All Web Dynpro components which are newly created in the SAP NetWeaver CE 7.1 Developer Studio are based on the new programming model.

The migration of existing Web Dynpro component is supported by the Web Dynpro Tools but can potentially be a complex task.

- SAP recommends not to migrate existing Web Dynpro components by default. Component migration is only recommended in specific use cases.
Modified Web Dynpro Component Architecture (2)

New Window Controller

- Replaces former Component Interface Controller class
- Implements one or many Interface Views defined in Local or Standalone Component Interface Definitions
- The 1..1 relation to former Component Interface View is no more mandatory
- New Window Plugs for Simpler Cross-Component Navigation
**Modified Web Dynpro Component Architecture (1)**

**New Local Component Interface Definition**

- **Local Component Interface Definition** is purely abstract
  - Defines IExternal-API of a component implementation
  - Component Interface Controller Class no longer exists

- Component controller implements **Local** and **Standalone Component Interface Definitions**

- View containers in component interface views
- Component interface inheritance
Web Dynpro Component Anatomy – SAP NetWeaver 7.0

Extermally Visible

Component Interface View

Internally Visible

Component Interface Controller

IExternal-API

Interface View Controller

Interface Controller

Component Controller

Custom Controller

View Controller

Message Pool

Web Dynpro Component

View Layout (UI Elements)

Window

embeds

Visual Entities

Programmatic Entities
Web Dynpro Component Anatomy – SAP NetWeaver CE 7.1

Visual Entities

- Component Interface View
- Interface View Controller
- View Layout (UI Elements)
- View Controller
- Web Dynpro Component
- Message Pool

Programmatic Entities

- Component Interface Controller
- Interface Controller
- Component Controller
- Custom Controller

External Visible

Internally Visible

embeds
Migrating Web Dynpro Components

Enhanced Web Dynpro Component Model

How To Migrate Web Dynpro Components

SAP Recommendations on Component Migration

Further Migration Aspects
Components which are based on the old component model are displayed with a *grey component icon*.

Old Web Dynpro components can be migrated with the *semi-automatic Web Dynpro Component Migration Tool* based on a *user assistance cheat sheet*.

Start the Component Migration Tool with context menu item „*Migrate …““
The **Component Migration Cheat Sheets** assists the developer in preparing a Web Dynpro component for migration.

Migration can be started after successful component preparation.
**Potential Side Effects of Web Dynpro Component Migration**

<table>
<thead>
<tr>
<th>Irreversible Process</th>
<th>The Component Migration Process <strong>cannot</strong> be reverted.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Behavioral Changes</td>
<td>Migrated Components might change behaviorally → <strong>post-migration testing required.</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Carefully migrate heavily used components!</strong></td>
</tr>
<tr>
<td>Hook Method Invocation Changes</td>
<td>Hook methods <code>doBeforeNavigation()</code> and <code>doPostProcessing()</code> are only called with pending action → <strong>might affect application logic</strong></td>
</tr>
<tr>
<td>Context Specific Changes</td>
<td>Invocation of <code>IWDNode.bind()</code> now throws a runtime exception when a corresponding supply function is defined.</td>
</tr>
<tr>
<td>Table UI Element Changes</td>
<td>Table Selection Change Behavior will change when <code>selectionChangeBehavior</code> was set to values <code>auto</code> or <code>nw04</code> <em>(not nw04splus)</em> before.</td>
</tr>
<tr>
<td>Naming UI Element IDs</td>
<td>UI element IDs MUST NOT start with ‘_’</td>
</tr>
</tbody>
</table>
How to Simplify and Accelerate Component Migration

RESOLVE all Component Controller Bypass Anti-Patterns in Your component implementation before migrating it!

ADHERE to the Component Controller Delegation Patterns! This accelerates and simplifies Your Component Migration Process!

Patterns

ONLY Define Indirect Usage Relations To the Interface Controller using the Component Controller as Delegator!

Anti-Patterns

NEVER Define a Usage Relation from a Non-Component Controller to the Interface Controller directly!

NEVER Define a Usage Relation from a Component Interface Controller to a Non-Component Controller!
NEVER Define a Usage Relation from a Non-Component Controller to the Interface Controller!
Detecting Anti-Patterns in Your Web Dynpro Component

To easily detect the Component Controller Bypass Anti-Pattern, you can apply the Inverse Search Function provided by the Web Dynpro Tools:

- Select Component Interface Controller
- Open context menu
- Select Menu Item “References – Project”
ONLY Define *Indirect* Usage Relations to the Interface Controller using the Component Controller as Delegator!
NEVER Define a Usage Relation from a Component Interface Controller to a Non-Component Controller!
**Component Controller Delegation Pattern A**

**ALWAYS** Define Indirect Controller Usage Relations from the Interface Controller to Non-Component Controllers using the Component Controller as a Delegator!

 Applies to SAP NetWeaver 04 and 7.0
Apply **Quick Fix** to resolve a broken implementation relation from the Component Controller to a Local or Standalone Component Interface Definition.
How to Migrate a Web Dynpro Component – Process Steps

<table>
<thead>
<tr>
<th>Preparative Steps before Component Migration</th>
<th>Component Migration</th>
<th>Post-Processing Steps after Component Migration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check Package Namespaces of Component Entities</td>
<td>Start Web Dynpro Component Migration</td>
<td>Implement Window Controllers</td>
</tr>
<tr>
<td>Analyze and Prepare Component Interface Controller</td>
<td></td>
<td>Invoke IPublic-API of Window Controllers</td>
</tr>
<tr>
<td>Analyze and Prepare Component Interface Views and Controllers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analyze and Prepare Implemented Interfaces</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Web Dynpro Component Migration Cheat Sheet
The component migration process steps described in this presentation are **all potential** steps to be applied.

Most Web Dynpro component implementations can be migrated to the new component model **more easily** by applying a **small subset** of component migration process steps!

---

**Scenario 1**

- **Preparative Steps before Component Migration**
  - Analyze and Prepare Component Interface
  - Analyze and Prepare Component Interface Views and Controllers
  - Model Component Migration

- **Component Migration**
  - Start Web Dynpro Component Migration

- **Post-Processing Steps after Component Migration**
  - Implement Window Controls
  - Implement Window Controls

---

**Scenario 2**

- **Preparative Steps before Component Migration**
  - Analyze and Prepare Component Interface
  - Analyze and Prepare Component Interface Views and Controllers
  - Model Component Migration

- **Component Migration**
  - Start Web Dynpro Component Migration

- **Post-Processing Steps after Component Migration**
  - Implement Window Controls
  - Implement Window Controls

---

**Scenario 3**

- **Preparative Steps before Component Migration**
  - Analyze and Prepare Component Interface
  - Analyze and Prepare Component Interface Views and Controllers
  - Model Component Migration

- **Component Migration**
  - Start Web Dynpro Component Migration

- **Post-Processing Steps after Component Migration**
  - Implement Window Controls
  - Implement Window Controls
Component Preparation: Names and Packages

Web Dynpro Component Migration Cheat Sheet

Preparative Steps before Component Migration

Check Names and Packages of Component Entities

- Analyze and Prepare Component Interface Controller
- Analyze and Prepare Component Interface Views
- Analyze and Prepare Implemented Interfaces

Move Views, Windows and Custom Controllers
* dynamic cheat sheet step

Check Conflicting Window Names
* dynamic cheat sheet step

Check Component Package

© SAP AG 2007, SAP TechEd '07 / CE206 / 43
Component Preparation: Move Component Entities

Web Dynpro Component Migration Cheat Sheet

Preparative Steps before Component Migration

- Check Names and Packages of Component Entities
  - Analyze and Prepare Component Interface Controller
  - Analyze and Prepare Component Interface Views
  - Analyze and Prepare Implemented Interfaces

- Move Views, Windows and Custom Controllers
  * dynamic cheat sheet step
  - Check Conflicting Window Names
  - Check Component Package

© SAP AG 2007, SAP TechEd '07 / CE206 / 44
Component Preparation: Check Conflicting Window Names

Preparative Steps before Component Migration

- Check Names and Packages of Component Entities
  - Analyze and Prepare Component Interface Controller
  - Analyze and Prepare Component Interface Views
  - Analyze and Prepare Implemented Interfaces

- Move Views, Windows and Custom Controllers

- Check Conflicting Window Names

- Check Component Package
Preparative Steps before Component Migration

- Check Names and Packages of Component Entities
  - Analyze and Prepare Component Interface Controller
  - Analyze and Prepare Component Interface Views
  - Analyze and Prepare Implemented Interfaces

- Move Views, Windows and Custom Controllers

- Check Conflicting Window Names

* dynamic cheat sheet step

- Check Component Package
Component Preparation: Component Interface Controller

Web Dynpro Component Migration Cheat Sheet

Preparative Steps before Component Migration

- Check Package Namespaces of Component Entities
- Analyze and Prepare Component Interface Controller
- Analyze and Prepare Component Interface Views
- Analyze and Prepare Implemented Interfaces
- Analyze and Prepare Interface Context
- Analyze and Prepare Interface Events and Event Handlers
- Analyze and Prepare Public Interface Methods
- Analyze and Transfer Interface Controller Coding
- Implement Interface Controller in Component Controller
- Re-Define Controller Usage Relations

© SAP AG 2007, SAP TechEd '07 / CE206 / 47
STRICTLY Adhere to the Component Controller Delegation Patterns! RESOLVE all Bypass Anti-Patterns before!
This highly accelerates Your Component Migration Process!
Web Dynpro Component Migration Cheat Sheet

Preparative Steps before Component Migration

1. Check Package Namespaces of Component Entities
2. Analyze and Prepare Component Interface Controller
3. Analyze and Prepare Component Interface Views
4. Analyze and Prepare Implemented Interfaces
5. Analyze and Prepare Implemented Interfaces
6. Analyze and Prepare Interface Context
7. Implement Interface Controller in Component Controller
8. Re-Define Controller Usage Relations
9. Analyze and Prepare Interface Events and Event Handlers
10. Analyze and Prepare Public Interface Methods
11. Analyze and Transfer Interface Controller Coding
Web Dynpro Component Migration Cheat Sheet

Preparative Steps before Component Migration

- Check Package Namespaces of Component Entities
- Analyze and Prepare Component Interface Controller
- Analyze and Prepare Component Interface Views
- Analyze and Prepare Implemented Interfaces
- Analyze and Prepare Interface Context
- Analyze and Prepare Interface Events and Event Handlers
- Analyze and Prepare Public Interface Methods
- Analyze and Transfer Interface Controller Coding
- Implement Interface Controller in Component Controller
- Re-Define Controller Usage Relations

Component Preparation: Interface Context
Component Preparation: Interface Event Handlers

Web Dynpro Component Migration Cheat Sheet

Preparative Steps before Component Migration

- Check Package Namespaces of Component Entities
- Analyze and Prepare Component Interface Controller
- Analyze and Prepare Component Interface Views
- Analyze and Prepare Implemented Interfaces
- Analyze and Prepare Interface Context
- Analyze and Prepare Interface Events and Event Handlers
- Analyze and Prepare Public Interface Methods
- Analyze and Transfer Interface Controller Coding

Implement Interface Controller in Component Controller
Re-Define Controller Usage Relations

© SAP AG 2007, SAP TechEd '07 / CE206 / 51
Component Preparation: Interface Events

Web Dynpro Component Migration Cheat Sheet

Preparative Steps before Component Migration

- Check Package Namespaces of Component Entities
- Analyze and Prepare Component Interface Controller
- Analyze and Prepare Component Interface Views
- Analyze and Prepare Implemented Interfaces
- Analyze and Prepare Interface Context
- Analyze and Prepare Interface Events and Event Handlers
- Analyze and Prepare Public Interface Methods
- Analyze and Transfer Interface Controller Coding
- Implement Interface Controller in Component Controller
- Re-Define Controller Usage Relations
Component Preparation: Public Interface Methods

Web Dynpro Component Migration Cheat Sheet

Preparative Steps *before* Component Migration

- Check Package Namespaces of Component Entities
- Analyze and Prepare Component Interface Controller
- Analyze and Prepare Component Interface Views
- Analyze and Prepare Implemented Interfaces

Implement Interface Controller in Component Controller

- Analyze and Prepare Interface Context
- Analyze and Prepare Interface Events and Event Handlers
- Analyze and Prepare Public Interface Methods
- Analyze and Transfer Interface Controller Coding

Re-Define Controller Usage Relations
Component Preparation: Interface Controller Coding

Web Dynpro Component Migration Cheat Sheet

Preparative Steps before Component Migration

- Check Package Namespaces of Component Entities
- Analyze and Prepare Component Interface Controller
- Analyze and Prepare Component Interface Views
- Analyze and Prepare Implemented Interfaces

Implement Interface Controller in Component Controller

Re-Define Controller Usage Relations

Analyze and Prepare Interface Context

Analyze and Prepare Interface Events and Event Handlers

Analyze and Prepare Public Interface Methods

Analyze and Transfer Interface Controller Coding

© SAP AG 2007, SAP TechEd '07 / CE206 / 54
Component Preparation: Component Interface Views

Web Dynpro Component Migration Cheat Sheet

Preparative Steps before Component Migration

- Check Package Namespaces of Component Entities
- Analyze and Prepare Component Interface Controller
- Analyze and Prepare Component Interface Views and their Controllers

- Analyze and Copy Interface View Controller Code Handling Plugs
- Note Outbound, Exit and Suspend Plugs
- Analyze and Copy Interface View Controller Coding Invoking IPublic-APIs

See Step Post-Processing Steps after Component Migration
Component Preparation: Outbound, Exit and Suspend Plugs

Web Dynpro Component Migration Cheat Sheet

Preparative Steps before Component Migration

- Check Package Namespaces of Component Entities
- Analyze and Prepare Component Interface Controller
- Analyze and Prepare Component Interface Views and their Controllers

See Step Post-Processing Steps after Component Migration

- Note Outbound, Exit and Suspend Plugs
- Analyze and Copy Interface View Controller Code Handling Plugs
- Analyze and Copy Interface View Controller Coding Invoking IPublic-APIs
Component Preparation: Interface View Controller Coding

Web Dynpro Component Migration Cheat Sheet

Preparative Steps **before** Component Migration

- Check Package Namespaces of Component Entities
- Analyze and Prepare Component Interface Controller
- Analyze and Prepare Component Interface Views and their Controllers

- Note Outbound, Exit and Suspend Plugs
- Analyze and Copy Interface View Controller Code Handling Plugs
- Analyze and Copy Interface View Controller Coding Invoking IPublic-APIs

See Step Post-Processing Steps **after** Component Migration
Component Preparation: Interface View Controller Coding

Web Dynpro Component Migration Cheat Sheet

Preparative Steps before Component Migration

- Check Package Namespaces of Component Entities
- Analyze and Prepare Component Interface Controller
- Analyze and Prepare Component Interface Views and their Controllers
- Note Outbound, Exit and Suspend Plugs
- Analyze and Copy Interface View Controller Code Handling Plugs
- Analyze and Copy Interface View Controller Coding Invoking IPublic-APIs

See Step Post-Processing Steps after Component Migration
Component Preparation: Implemented Interfaces

Web Dynpro Component Migration Cheat Sheet

Preparative Steps before Component Migration

- Check Package Namespaces of Component Entities
- Analyze and Prepare Component Interface Controller
- Analyze and Prepare Component Interface Views and their Controllers
- Analyze and PrepareImplemented Interfaces

Check Native Interface Inheritance

* dynamic cheat sheet step
Start Component Migration

Web Dynpro Component Migration Cheat Sheet

Component Migration

Start
Web Dynpro
Component Migration

Start Migration

This step will complete the migration. After the step has completed the component will have the current interface programming model.

Click to perform
Post-Processing Steps after Component Migration

- Implement Window Controllers
- Invoke IPublic-API of Window Controllers

See Step Component Preparation: Component Interface Views

- Implement Inbound-, Resume- and Startup-Plug Event handlers

© SAP AG 2007, SAP TechEd '07 / CE206 / 61
Complete Web Dynpro Component Migration Process

Preparing your Web Dynpro Components for Migration

- Refactor Component Controller Bypass Anti-Patterns
- Make sure, that the Component Controller Delegation Pattern is strictly applied!
- Copy Component Interface View Controller Implementation to Notepad

Applying the Web Dynpro Component Migration Tool

Component Migration Post-Processing

- Paste Interface View Controller Implementation from Notepad to Window Controller
- Redefine broken interface controller usages pointing to new window controller
- Invoke the window controller's IPublic-API to fire exit, outbound and suspend plugs

Migration Process Summary
Migrating Web Dynpro Components

- Enhanced Web Dynpro Component Model
- How To Migrate Web Dynpro Components
- SAP Recommendations on Component Migration
- Further Migration Aspects
SAP Recommendation on Component Migration at a Glance

Do not Migrate Web Dynpro Components *By Default!*

- You can still run existing Web Dynpro Components in CE 7.1
- SAP recommends *not* to migrate existing Web Dynpro components by default.
- In many existing Web Dynpro application scenarios there is no mandatory necessity for component migration.

Do Migrate Web Dynpro Components *In Specific Use Cases Only!*

- Existing Web Dynpro component should only be migrated to the new component model in specific use cases:
  - Ease of maintenance: Developers only deal with new component model
  - Simplified cross-Component Navigation
  - Apply inheritance of Standalone Component Interfaces
  - Implementation of Standalone Component Interface using new function
  - Development of pure layout components referencing UI components
## SAP Recommendations on Web Dynpro Component Migration

<table>
<thead>
<tr>
<th>Mandatory Migration</th>
<th>Optional Migration</th>
<th>Not Necessary Migration</th>
</tr>
</thead>
</table>
| - An existing Web Dynpro component must implement a new standalone component interface definition utilizing new functions  
- An existing Web Dynpro component must utilize component interface inheritance which is only supported in the new component model  
- An existing Web Dynpro component must support the generic secondary help service | - Web Dynpro component developers are not familiar with the old Web Dynpro component model  
- You want to simplify your cross-component navigation by applying the new window navigation plug concept | - An existing Web Dynpro component must implement a new standalone component interface definition not utilizing new functions  
- Utilizing View Containers in Component Interface Views  
- You want to apply the new default plug property for Web Dynpro view usages  
- You want to use the new universal context nodes |
SAP Recommendations on Web Dynpro Component Migration

Mandatory Migration

- An existing Web Dynpro component must implement a new standalone component interface definition utilizing new functions.
- An existing Web Dynpro component must utilize component interface inheritance which is only supported in the new component model.
- An existing Web Dynpro component must support the generic secondary help service.
SAP Recommendations on Web Dynpro Component Migration

- Web Dynpro component developers are not familiar with the old Web Dynpro component model
- You want to simplify your cross-component navigation by applying the new window navigation plug concept
SAP Recommendations on Web Dynpro Component Migration

- An existing Web Dynpro component must implement a new standalone component interface definition not utilizing new functions
- Utilizing View Containers in Component Interface Views to develop **Layout Components** referencing UI Components
You want to apply the new default plug property for Web Dynpro view usages
You want to use the new universal context nodes
Migrating Web Dynpro Components from SAP NetWeaver 7.0 to CE 7.1

DEMO
Sample Component Migration Scenario

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Window Name Conflicts with Component Controller Name</td>
</tr>
<tr>
<td>2</td>
<td>Exit plug defined in Component Interface View</td>
</tr>
<tr>
<td>3</td>
<td>View controller fires exit plug invoking the interface view controllers IPublic-API</td>
</tr>
<tr>
<td>4</td>
<td>View controller bypasses component controller</td>
</tr>
<tr>
<td>5</td>
<td>Component context is mapped to interface data context</td>
</tr>
<tr>
<td>6</td>
<td>Interface context is mapped to the component context</td>
</tr>
</tbody>
</table>
How to Migrate a Web Dynpro Component – Process Steps

Web Dynpro Component Migration Cheat Sheet

<table>
<thead>
<tr>
<th>Preparative Steps before Component Migration</th>
<th>Component Migration</th>
<th>Post-Processing Steps after Component Migration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check Package Namespaces of Component Entities</td>
<td>Start Web Dynpro Component Migration</td>
<td>Implement Window Controllers</td>
</tr>
<tr>
<td>Analyze and Prepare Component Interface Controller</td>
<td></td>
<td>Invoke IPublic-API of Window Controllers</td>
</tr>
<tr>
<td>Analyze and Prepare Component Interface Views and Controllers</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Migrating Web Dynpro Components

- Enhanced Web Dynpro Component Model
- How To Migrate Web Dynpro Components
- SAP Recommendations on Component Migration
- Further Migration Aspects
Migrating Component Interface Definitions

1. SAP recommends **not to migrate** existing Web Dynpro component interface definitions by default. In many existing Web Dynpro application scenarios there is no mandatory necessity for component interface definition migration.

2. Existing Web Dynpro component interface definitions should only be migrated to the new component model in specific use cases.

SAP Recommendation
Migrating Web Dynpro Controller Contexts

Does the component migration tool also convert the former context model and value elements to the new universal elements?

- I know that there is no difference between context model versus value elements (nodes and attributes) any more in SAP NetWeaver CE 7.1.
- Both element types are merged in the new universal context nodes and attributes.

No, the component migration tool **does not** migrate former context model and value elements to the new universal context elements itself.

You must explicitly trigger this context migration in a separate step.

Migrate complete controller context hierarchy separately via context menu item **Migrate Context Tree** on context root node level.
**Migrating Deprecated UI Element Associations**

In SAP NetWeaver CE 7.1 the following **UI element associations** of the *Table UI element* were **deprecated**:

<table>
<thead>
<tr>
<th>Deprecated Association</th>
<th>New Association</th>
<th>Association Target UI Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column</td>
<td>GroupedColumn</td>
<td>□ IWDTableColumn □ IWDTableColumnGroup</td>
</tr>
<tr>
<td>MasterColumn</td>
<td>RowArrangement</td>
<td>□ IWDTreeByNestingTableColumn □ IWDTableRowGrouping</td>
</tr>
</tbody>
</table>

Migrate deprecated UI element associations of the Table UI element with context function „Migrate Columns“
To run Web Dynpro Applications using Adaptive RFC Models in SAP NetWeaver CE 7.1 a **System Landscape Directory (SLD)** is required.

In SAP NetWeaver CE 7.1 the SLD combined with the SLD Administration UI is first available in **CE 7.1 SP Stack 3** together with the **ESR (Enterprise Service Repository) Add-On**.

- Regarding the compatibility of an existing SLD running on a separate SAP NetWeaver Application Server Java or on an SAP Backend System with Web Dynpro Adaptive RFC CE 7.1 see SAP Note 954820.
Summary

- The Web Dynpro Foundation for Java in SAP NetWeaver CE 7.1 is backward source-compatible with SAP NetWeaver 04 and 7.0.

- The migration of Web Dynpro applications to SAP NetWeaver CE 7.1 is made very simple: *just repair, build and deploy related Web Dynpro DCs and then run your Web Dynpro application*.

- The Web Dynpro component model was subject to significant changes in SAP NetWeaver CE 7.1.
  - Abstract component interfaces (views, controller) are now implemented by component and new window controllers.

- Existing Web Dynpro components can be migrated with the semi-automatic Web Dynpro Component Migration Tool.

- SAP Recommendation on Web Dynpro component migration:
  - Do not migrate components by default!
  - Migrate components in specific use cases only!
Further Information

SAP Public Web:

Related SAP Education and Certification Opportunities
http://www.sap.com/education/

Related Workshops/Lectures at SAP TechEd 2007
CE110, “Web Dynpro Java – What’s New in SAP NetWeaver 7.1 Composition Environment”, 1 hour lecture
CE351, “Designing Low TCO Applications in Web Dynpro Java”,
CE352, “Getting the best from the Web Dynpro Framework”,
CE252, “Moving J2EE Applications to SAP NetWeaver Composition Environment Made Easy”
The SAP NetWeaver, Development Subscription offers a cost effective total solution for developers to build applications for the SAP NetWeaver platform.

Subscription gives you one year access to ...

- SAP NetWeaver platform software, patches, and updates
- Development license for SAP NetWeaver to evaluate, develop and test
- Standard software maintenance
- Online sessions from SAP TechEd
- Access to SAP Enterprise Services Workplace for testing
- Premium presence in forums

Purchase the SAP NetWeaver, Development Subscription today at the TechEd Community Clubhouse, or online at https://www.sdn.sap.com/irj/sdn/subscriptions

Show us you are a subscriber and get a reward!
THANK YOU FOR YOUR ATTENTION!

QUESTIONS – SUGGESTIONS – DISCUSSION
Feedback

Please complete your session evaluation.

Be courteous — deposit your trash, and do not take the handouts for the following session.

Thank You!