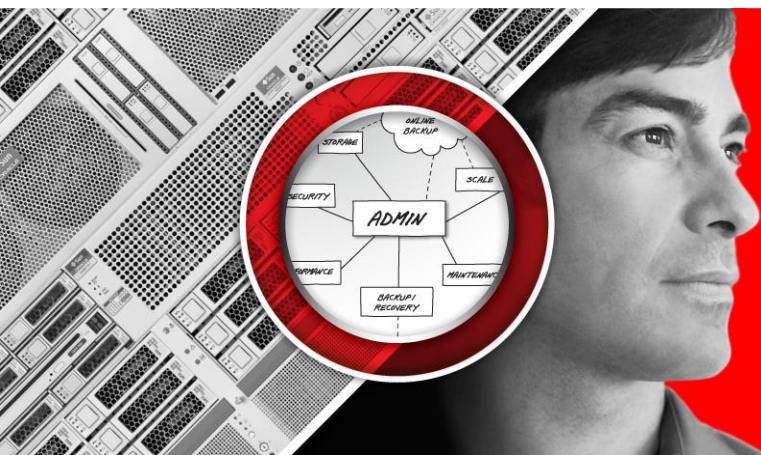


**ORACLE®**



**ENGINEERED  
FOR INNOVATION**

**ORACLE  
OPEN  
WORLD**

**ORACLE®**

## **Fast, but not Furious - ADF Task Flow in 60 Minutes**

Frank Nimphius, Senior Principal Product Manager  
Oracle Application Development Tools

# Program Agenda

- Sales Pitch
- Bounded Task Flows
- Task Flow Activities
- Exception Handling
- Memory Scopes
- ADF Region
  - Static
  - Dynamic
- Advanced Concepts
  - Train, Dialog, Remote access



# Sales Pitch

# ADF Controller

- ADF controller (ADFc) extends and supersedes JavaServer Faces navigation model
- Task flows define and process user navigation
  - Ideally suited for reuse and to partition application development
- Metadata driven
- Key concept and core knowledge in Oracle ADF
  - you can survive up to 10 days without water but you won't last more than a day without task flows

# ADF Controller Responsibility

- Perform application navigation across browser pages and within browser pages
- Navigate between activities and task flows and maintain controller state
- Execute controller logic and custom methods during navigation
- Manage task flow instance

# ADF Controller Benefits

- Additional memory scopes
  - Avoid memory attribute name conflicts
  - Avoid memory bloating
- Improved and declarative exception handling
- Supports traditional web navigation model and rich internet navigation models
- Support for nested control flows
- Configurable order of Phase Listener execution
- Save points for undo and failure recovery

# About ADF Task Flow

- Unbounded Taskflow
  - No well-defined boundary or single point of entry
  - Used to define the “top level” flow of an application
  - Used to build menu models
  - Declarative bookmarking of view activities
- Bounded Taskflow
  - Reusable "white box" navigation and processing
  - Single point of entry
  - Well-defined boundary
  - Own memory scope
  - Declarative transaction management
  - Input/output parameters



# ADF Task Flow - Key Benefits

- Help partition and modularise large business applications
- Visual task flow diagrammer provides shared language between developers and domain experts
- Reusable application flows (ADFLibrary)
- Static and dynamic configuration
- Can have isolated transactions
- Can be used as "remote services with a UI"
- On-board Security
  - ADF Security
  - Bounded task flow browser access protection



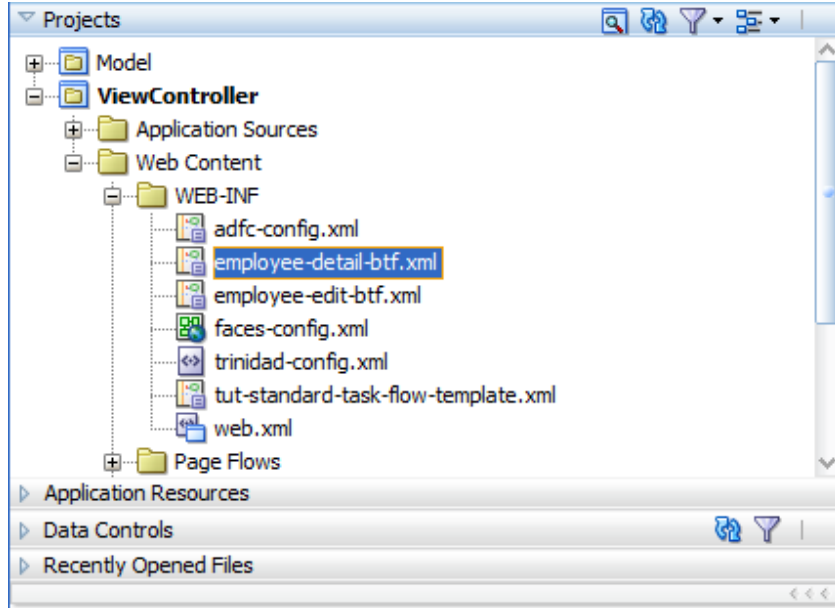
# Product Demonstration

---

## Sample Overview

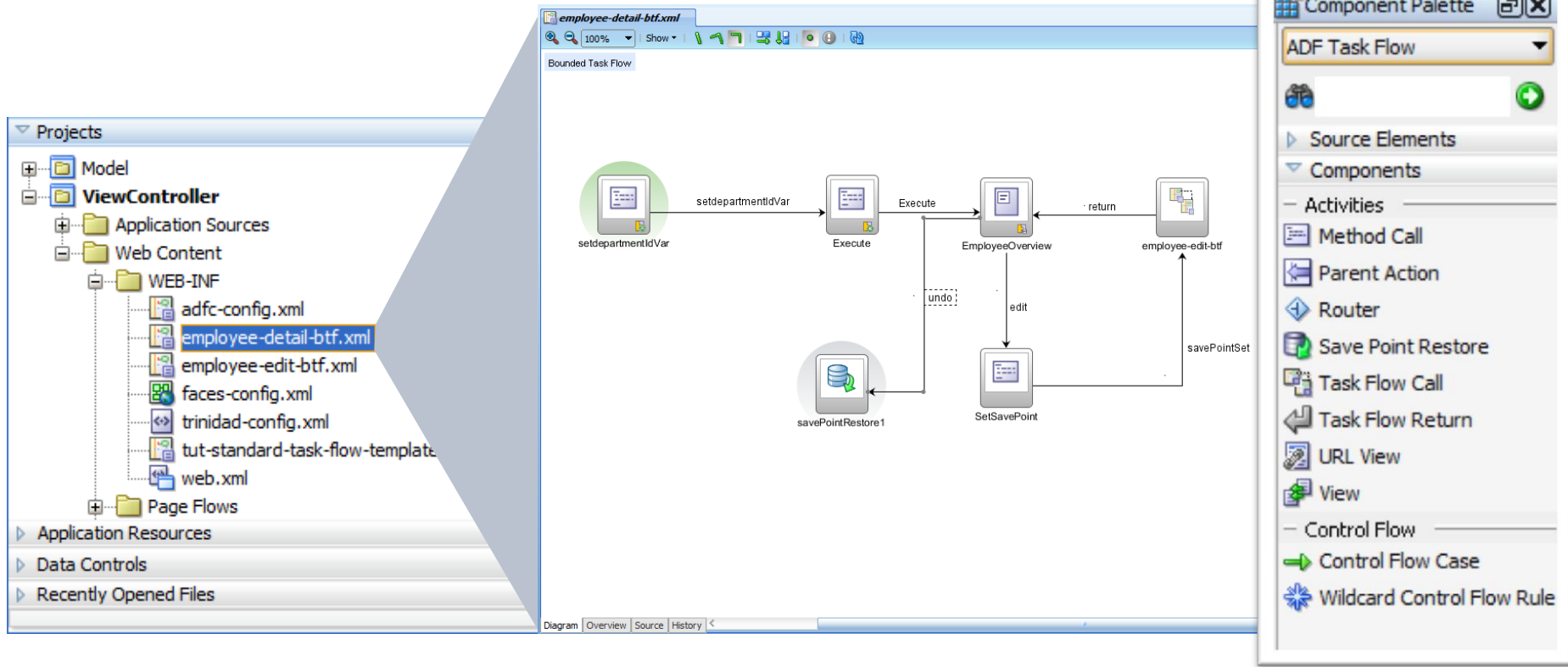
# Bounded Task Flow

# Bounded Task Flow



- XML defined in View Controller project
- Located in WEB-INF directory or (better) sub-folder(s) therein
- XML
  - Defines task flow behavior like transaction management and Data Control sharing
  - Defines activities
    - Activities reference views, methods, operations, task flows, switcher statements
  - Defines managed beans used in task flow
  - May reference task flow template

# Bounded Task Flow Diagram



# Task Flow Access Options

- From URL
  - Bounded task flows can be configured to be hidden for browser access
  - Configured through "URL Invoke" property
    - Calculated (default), Disallowed, Allow
- From task flow
  - Task Flow call activity invokes bounded task flow
  - Calling flow state is saved on controller stack
- ADF Region
  - Special task flow use case: executes task flow as part of a page

# Inter Task Flow Communication

- Input Parameter

Parameters



**Input Parameter Definitions**  

Name *	Class	Value	Required
departmentIdParam		<code>#{pageFlowScope.employeeDetailProcessBean.departmentIdParam}</code>	<input checked="" type="checkbox"/>

- Return values

Parameters

Behavior

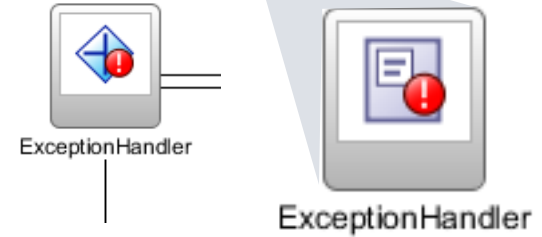
**Return Value Definitions**  

Name *	Class	Value *
selectedEmployeeId		<code>#{pageFlowScope.employeeDetailProcessBean.returnValue}</code>

# Exception Handling



- Exceptions can be mapped to an activity
- Called when exceptions are thrown
  - Not a “normal” part of a flow
- Best practice: Add error handler to each flow
  - Handler could be any task flow activity
  - Use method activity or a router activity to handle errors based on their error message



```
<exception-handler>unexpectedErrorPage</exception-handler>
```





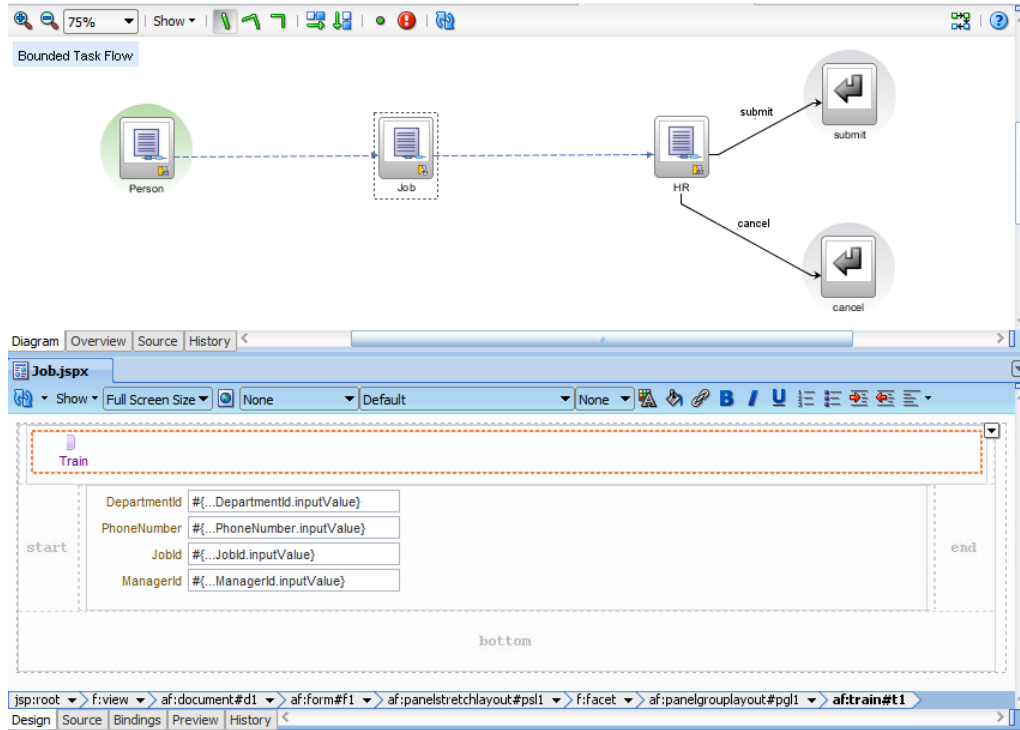
# Product Demonstration

---

## Bounded Task Flow

# Trains

# Example



The diagram shows a flow from 'Person' to 'Job' to 'HR'. Below it is a form with the following fields:

DepartmentId	<input type="text" value="20"/>
PhoneNumber	<input type="text" value="603.123.6666"/>
* JobId	<input type="text" value="MK_REP"/>
ManagerId	<input type="text" value="201"/>

# About Trains

- `af:train`
  - `org.apache.myfaces.adf.model.MenuModel` **extending** `TreeModel`
  - Renders train stops in a sequence
  - Trainstops can be skipped
- `af:trainButtonBar`
  - `org.apache.myfaces.adf.model.MenuModel` **extending** `TreeModel`
  - Renders train stops in a sequence plus additional navigation button that avoid the confusion of skipped stops
- Bounded task flows can be configured to expose a train model
  - `#{ControllerContext.currentViewPort.taskFlowContext.trainModel}`



# Product Demonstration

---

## Train Navigation

# Memory Scopes

# ADF Controller Memory Scopes

- Application (Servlet Scope)
  - Attributes and managed beans are available for an entire application and shared among users
  - Information should not have session dependency
- Session (Servlet Scope)
  - Attributes and beans are user instance specific, and exist from the time of their creation to the time they are deleted or the session ends
  - Example usecase is a user info bean that stores information about a user to avoid unnecessary queries

# ADF Controller Memory Scopes

- PageFlow (ADF Controller Scope)
  - Available for the duration of a task flow
  - If used with nested task flows, the PageFlow scope of the caller flow is suspended and activated upon return from the called Task Flow
- View (ADF Controller Scope)
  - Stores objects used by a single view
  - Automatically released when the user leaves the view



# ADF Controller Memory Scopes

- Request (Servlet Scope)
  - Represents the smallest available scope
  - Lasts for the time it takes to transition from one view activity to the next
- BackingBean (ADF Controller Scope)
  - Special case of request scope
  - Associated with a specific managed bean instance
  - Used with declarative components, regions and page fragments that use a managed bean to hold view state information
- None
  - Use for beans that don't require to keep state so that bean does not need to remain in memory

# ADF Region

# About ADF Regions

- Display task flows in independent areas of a page or a page fragment
- Executes task flow without requiring the parent view to refresh
- Key enabler of enterprise Web 2.0 development with the Oracle Fusion development platform
- Allows building desktop like web user interfaces that unveil the real power of Ajax
- Basis of UI Shell layout pattern

# Use it inside another View

\* DepartmentId

\* DepartmentName

ManagerId

LocationId

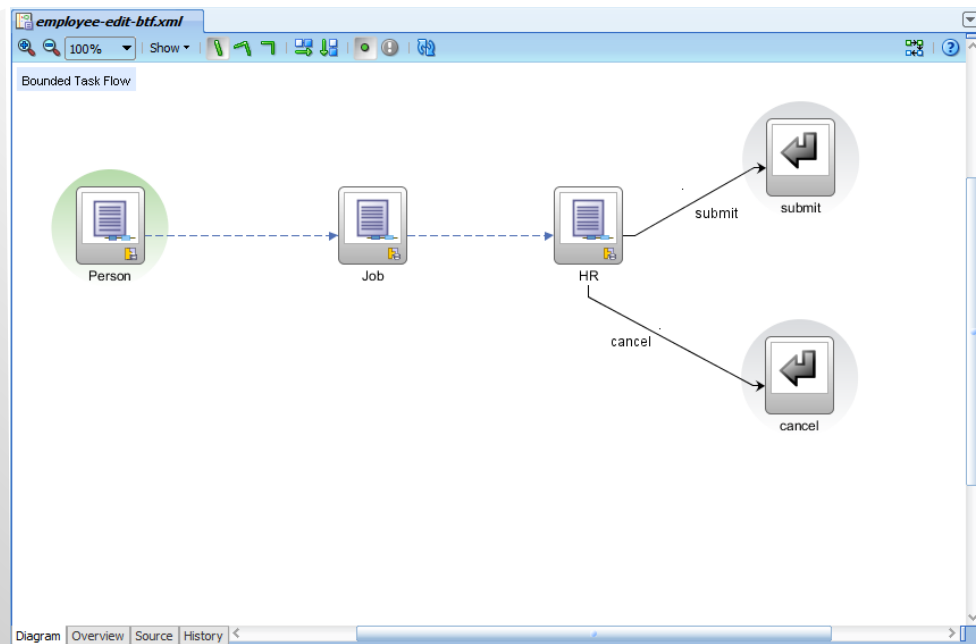
First Previous Next Last

Submit

View ▾  Detach

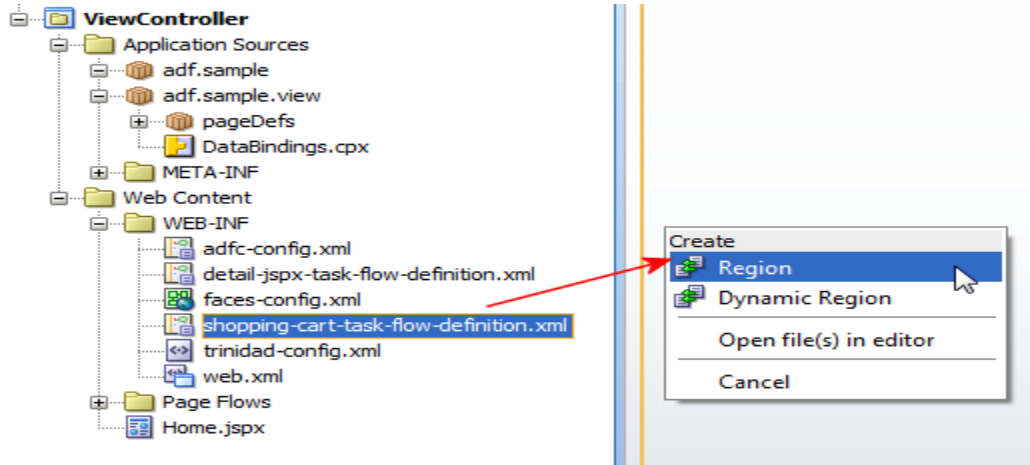
DepartmentId	EmployeeId	FirstName	LastName	Email
20	201	Michael	Hartstein	MHARTSTE
20	202	Pat	Fay	PFAY

Edit Current Row Commit Undo Last Change



# How-to create an ADF Region

- Drag bounded task flow to visual page editor
  - Task flow must use page fragments





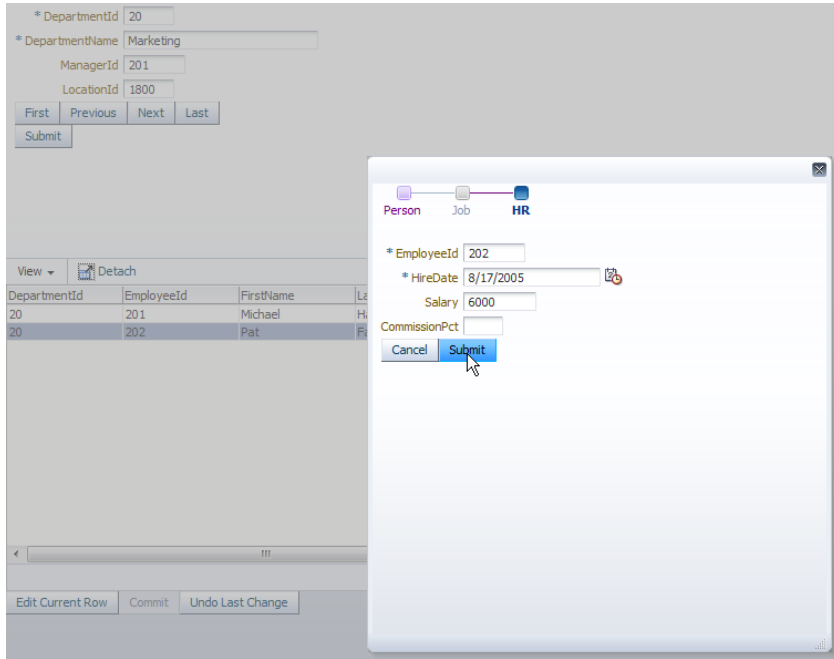
# Product Demonstration

---

## ADF Region

# Task Flow as Dialog

# About Task Flow as Dialog



- Allows you to build custom and complex lookup screens
  - List-of-values
- Return value copied to calling task flow if DataControl not shared
- Command component that launches dialog receives return event



# Configure Bounded Task Flow

The screenshot displays the Oracle JDeveloper IDE interface. The main window shows a Bounded Task Flow diagram for the file `employee-edit-btf.xml`. The diagram consists of several tasks connected by arrows:

- `setdepartmentIdVar` (highlighted with a green circle) connects to `Execute` via the arrow `setdepartmentIdVar`.
- `Execute` connects to `EmployeeOverview` via the arrow `Execute`.
- `EmployeeOverview` connects to `SetSavePoint` via the arrow `edit`.
- `SetSavePoint` connects to `savePointRestore1` via the arrow `savePointSet`.
- `savePointRestore1` connects back to `EmployeeOverview` via the arrow `undo`.
- `EmployeeOverview` connects to `employee-edit-btf` (highlighted with a red circle and a dashed border) via the arrow `return`.

The right-hand side of the IDE shows the configuration panel for the `employee-edit-btf` task. The **General** section is expanded, showing the following configuration:

- Activity ID \*:** `employee-edit-btf`
- Task Flow Reference:** Static
- Document:** `/WEB-INF/employee-edit-btf.xml`
- ID:** `employee-edit-btf`
- Remote Application URL:** (empty)
- Description:** (empty)
- Parameters:** (empty)
- Listeners:** Before Listener: (empty), After Listener: (empty)
- Behavior:** Run As Dialog: true, Display Type: inline-popup, Dialog Return Value: (empty)
- Customization:** (empty)

# Remote Task Flow

# About Remote Task Flows

- Bounded Task Flow deployed on remote server
- Use case: shared task flow instance
- External Applications
  - Called using a GET request
  - Don't share the same session with caller
  - Return value copied to memory attribute by task flow call activity
  - Outcome determines next navigation in calling flow
  - Current restriction: Doesn't work when called from ADF region

# Q&A

# Join the JDeveloper/ADF Community



**Twitter**

[twitter.com/JDeveloper](https://twitter.com/JDeveloper)



**Facebook**

[facebook.com/JDeveloper](https://facebook.com/JDeveloper)



**Oracle's JDeveloper PM blog**

[blogs.oracle.com/ JDeveloperpm](https://blogs.oracle.com/JDeveloperpm)

[oracle.com/technetwork/jdev](https://oracle.com/technetwork/jdev)



**ORACLE®**