

# BOPF Determinations

SAP AG, 2012



# Disclaimer

---

This presentation outlines our general product direction and should not be relied on in making a purchase decision. This presentation is not subject to your license agreement or any other agreement with SAP. SAP has no obligation to pursue any course of business outlined in this presentation or to develop or release any functionality mentioned in this presentation. This presentation and SAP's strategy and possible future developments are subject to change and may be changed by SAP at any time for any reason without notice. This document is provided without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. SAP assumes no responsibility for errors or omissions in this document, except if such damages were caused by SAP intentionally or grossly negligent.

# Agenda

---

Introduction

Creating a Determination

Determination Configuration

- Settings
- Request, Read, and Write Nodes
- Times
- Dependencies

Determination Implementation



# Introduction

# Introduction

 SAP AG, Euroleiraste 16, 69190 Walldorf

Seth Cohen  
Newport Beach  
Orange County, California

Invoice No.: 2005-008546  
Customer No.: 15263741  
Tax No.: 12 345 67890  
Date: 07/18/2005

**Invoice**  
Your order 4711 of 07/12/2005

Item	Quantity	Art.-No.	Description	Price/U	Amount
1	1.00 PC	56452375	Sailing Boat	\$ 790.00	\$ 790.00
2	2.00 PC	75937255	Video Games	\$ 150.00	\$ 300.00
3	5.00 KG	43553452	Comics	\$ 100.00	\$ 500.00
Subtotal					\$ 1.590.00
plus Shipping Costs					\$ 8.00
plus Packing Costs					\$ 2.00
<b>Value</b>					<b>\$ 1.600.00</b>
Tax (16%) on \$ 1.600.00					\$ 256.00
<b>Total Value</b>					<b>\$ 1856.00</b>



## Motivation

- If there is a change of the quantity of video games in this invoice business object instance, the amount of this invoice has to be calculated again („side effect“)
- To calculate this amount automatically, we want to extend the standard change process for the invoice business object instances by the help of determinations

# Introduction

---

- A determination contains business-object-specific logic to perform side effects, which is executed if a business-object-internal event occurs.
- For example, determinations can react on the creation, the deletion, or the loading of an instance by deriving new values for fields.
- **Determination types:**
- **Transient Determination:** Modifies only transient fields or nodes (e.g. buffer), thus no locking is necessary.
- **Persistent Determination:** Modifies transient and persistent fields or nodes (e.g. database), thus locking is necessary.

## Hints

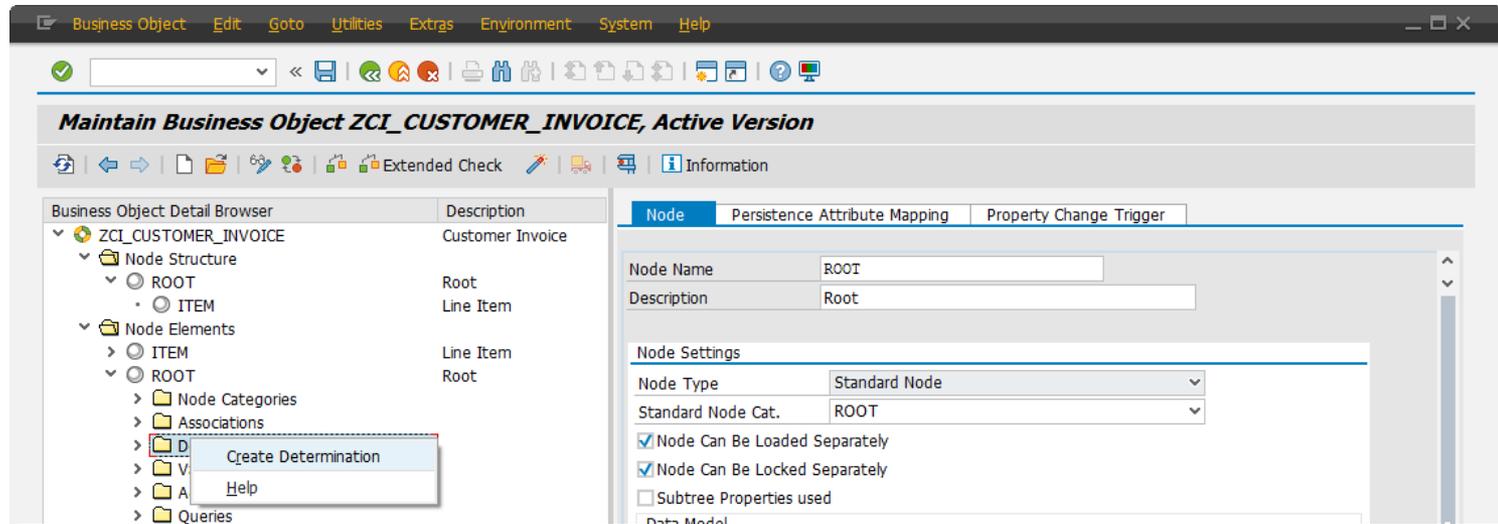
- Determinations should be designed as granular as possible for better reuse.
- Determinations shall not perform any consistency checks – therefore we use validations.
- Determinations are never executed in a nested manner to avoid endless loops.
- Determinations cannot be called directly by the consumer or out of entity implementation.



# Creating a Determination

# Creating a Determination

## Creating Determinations with the BOPF Configuration UI



Choose “Create Determination” in the context menu of the node to which the determination should be physically attached in the model. This node is called **Assigned Node**, nodes that trigger the determination are called **Triggering Node** or **Request Node**.

**Recommendation:** Choose an assigned node that is reachable from all the request nodes of the determination by associations:

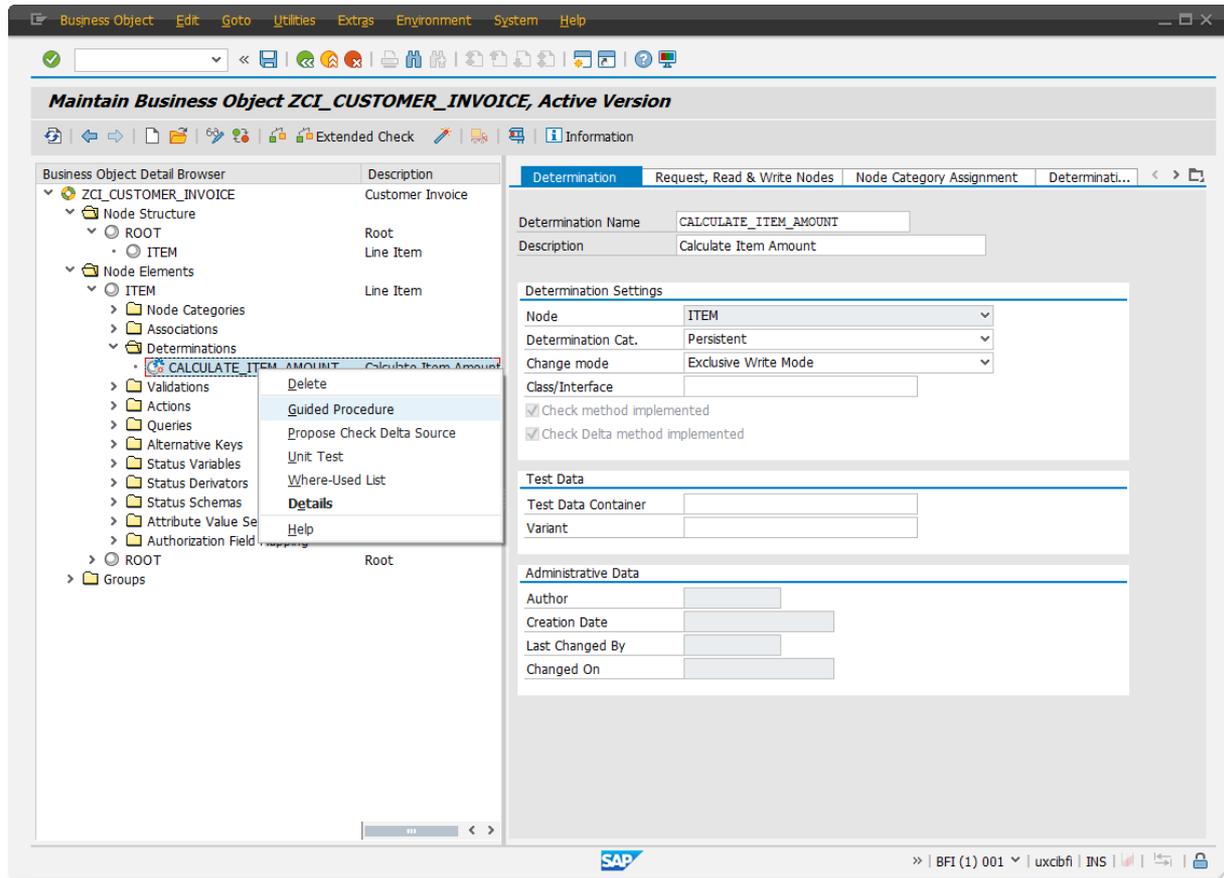
- **Example 1:** If there is only one request node and this node contains all the fields that are changed by the determination, take this node as assigned node.
- **Example 2:** If there are more request nodes, you choose one of each common parent node of them (because of the parent association connection).



# Determination Configuration

# Determination Configuration

## Guided Procedures



Use the „Guided Procedure“ to configure the following determination settings.

# Determination Configuration Settings

Maintain Business Object ZCI\_CUSTOMER\_INVOICE, Active Version

Determination Name: CALCULATE\_ITEM\_AMOUNT  
Description: Calculate Item Amount

**Determination Settings**

Node	ITEM
Determination Cat.	Persistent
Change mode	Exclusive Write Mode
Class/Interface	

Check method implemented  
 Check Delta method implemented

**Test Data**

Test Data Container	
Variant	

**Administrative Data**

Author	
Creation Date	
Last Changed By	
Changed On	

Navigation icons: [OK] [Cancel] [Back] [Forward] [Help]

Define the basic properties of the new determination

**Determination Category:** „Persistent“ or „Transient“ (if there is no change of persistent fields)

**Change Mode** (Rule of Thumb):

- Persistent Determination => „Exclusive Write“
- Transient Determination => „Only Read Mode“

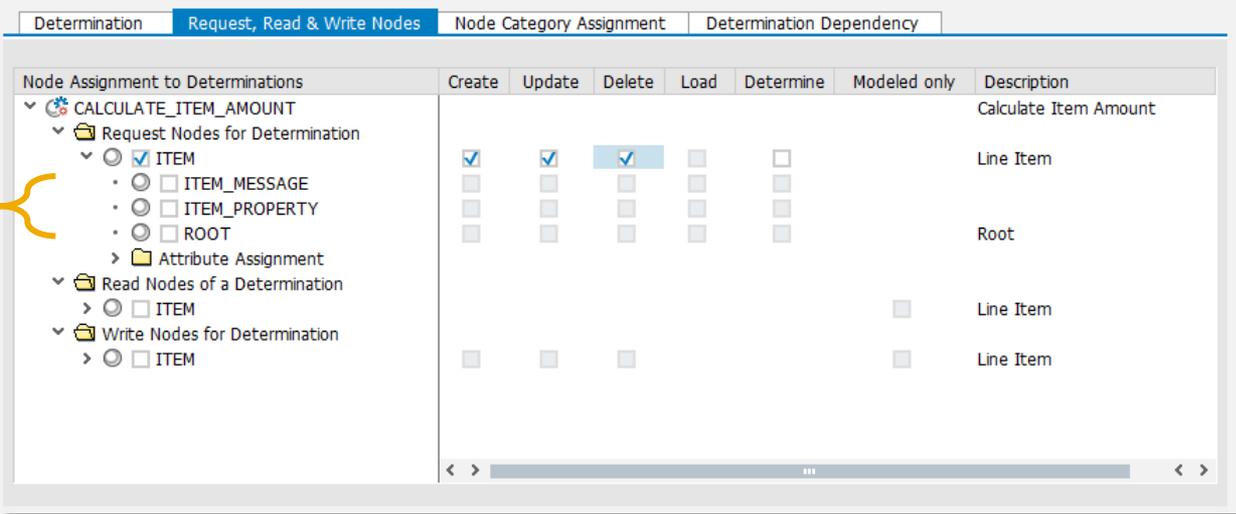
**Class/ Interface:** Choose a class from the BOPF Library or create a new class by forward navigation to implement an individual determination

# Determination Configuration

## Request, Read and Write Nodes

Nodes, which have associations to the Assigned Node

(<Node>~<Association>)



The screenshot shows the 'Request, Read & Write Nodes' tab in the SAP Determination Configuration tool. The left pane shows a tree view for the determination 'CALCULATE\_ITEM\_AMOUNT'. Under 'Request Nodes for Determination', the 'ITEM' node is selected with a checkmark. A yellow bracket highlights the 'ITEM' node and its sub-nodes: 'ITEM\_MESSAGE', 'ITEM\_PROPERTY', and 'ROOT'. The right pane is a table with columns: Create, Update, Delete, Load, Determine, Modeled only, and Description. The 'ITEM' row has checked boxes for Create, Update, and Delete. The 'ITEM\_MESSAGE', 'ITEM\_PROPERTY', and 'ROOT' rows have unchecked boxes for all columns. The 'Read Nodes of a Determination' section has an unchecked 'ITEM' checkbox. The 'Write Nodes for Determination' section has an unchecked 'ITEM' checkbox.

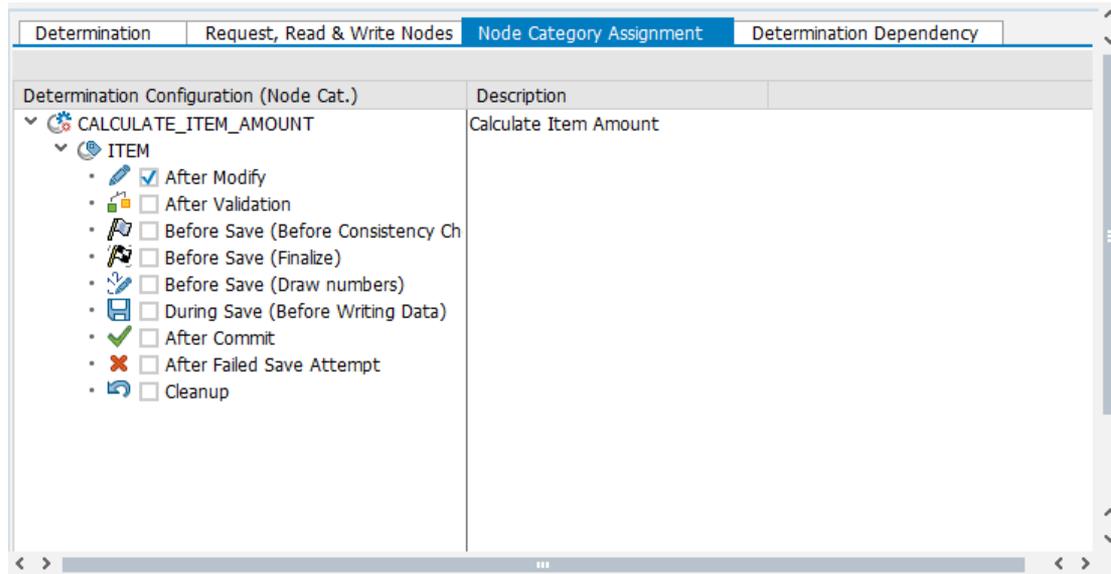
Node Assignment to Determinations	Create	Update	Delete	Load	Determine	Modeled only	Description
▼ CALCULATE_ITEM_AMOUNT							Calculate Item Amount
▼ Request Nodes for Determination							
<input checked="" type="checkbox"/> ITEM	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		Line Item
<input type="checkbox"/> ITEM_MESSAGE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/> ITEM_PROPERTY	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/> ROOT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		Root
> Attribute Assignment							
▼ Read Nodes of a Determination							
<input type="checkbox"/> ITEM						<input type="checkbox"/>	Line Item
▼ Write Nodes for Determination							
<input type="checkbox"/> ITEM						<input type="checkbox"/>	Line Item

Choose request nodes / associations and trigger condition

- **Request Nodes:** Nodes that trigger the determination whenever instances of this node fulfill the trigger condition (create, update, delete, load, determine)
- **Read Nodes:** Nodes that have to be all loaded into the buffer before the determination can start - instead of loading each on demand. Don not use them because they represent obsolete implementing mass-enabled data access (see performance guideline).
- **Write Nodes:** Only use write nodes in case of After Loading Determinations in order to specify the transient attributes that are derived by that determination.
- **Modeled Only:** If this checkbox is selected, the whole configuration of that row has no run-time effect (but could be useful to generate documentation or model checks).

# Determination Configuration

## Times

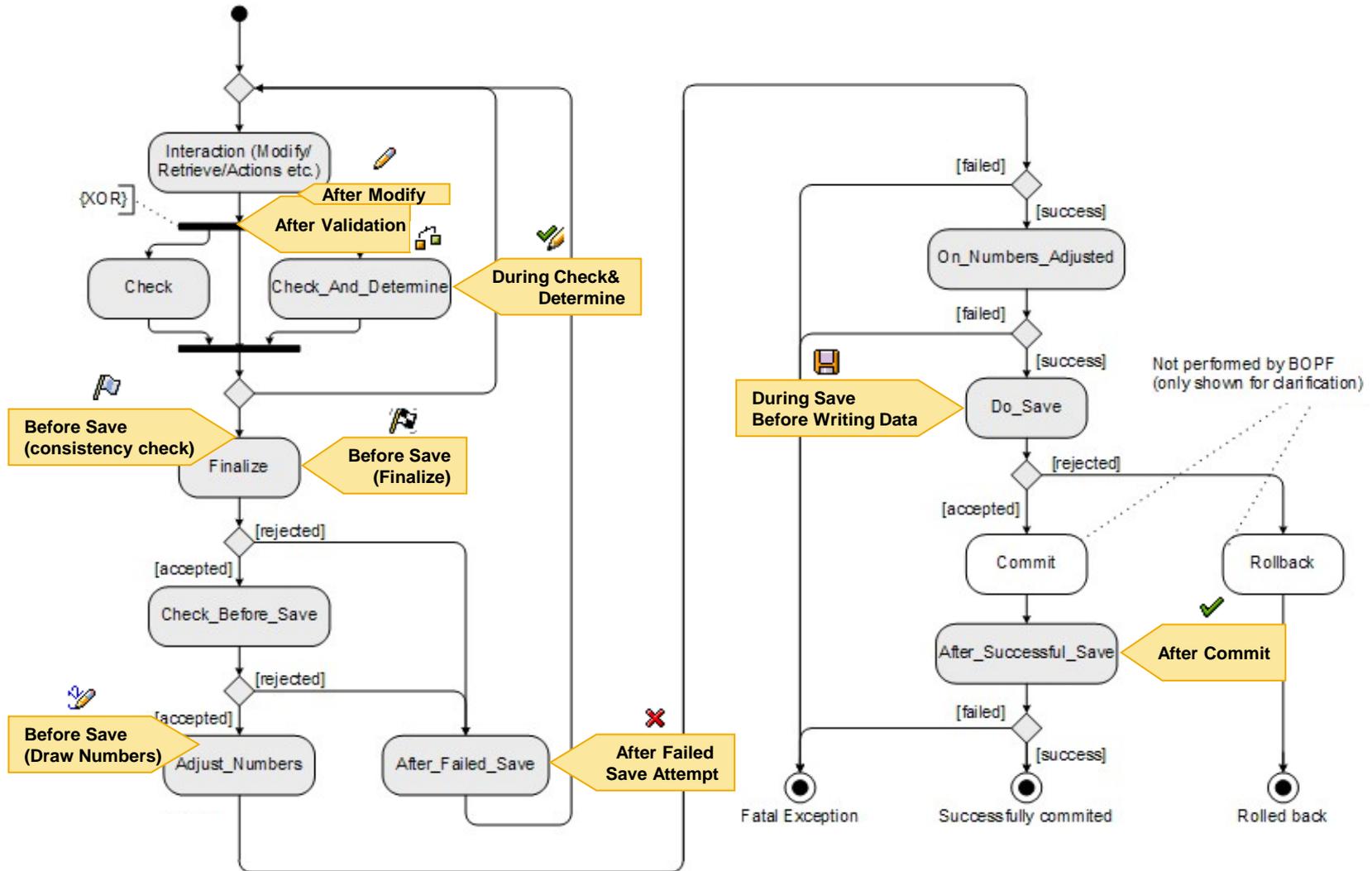


A determination time defines at what time in the transaction cycle the trigger condition of that determination should be evaluated.

- **Example:** The recalculation of the invoice amount should take place every time after a modification is done (=> determination time „after modify“), but only if there are instances of the `ITEM` node (=request node) that were updated (= trigger condition „update“).

# Determination Configuration

## Times Overview



# Determination Configuration

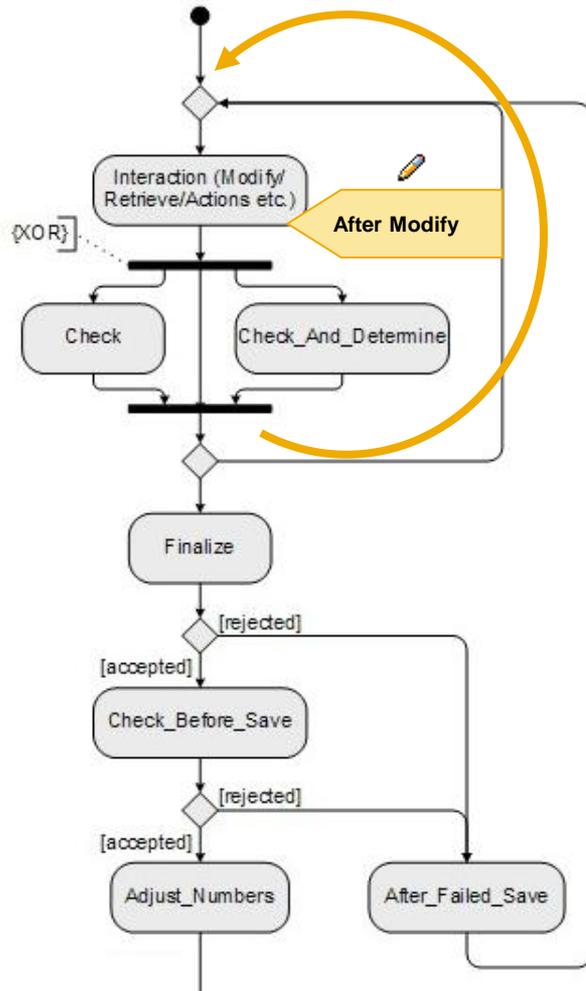
## Times & Trigger Conditions

Trigger condition ▶		Create	Update	Delete	Load*	Determine
▼ Determination Time						
	After Modfiy	X	X	X		
	During Check&Determine					X
	After Validation	X	X	X		
Before Save:						
	Before Consistency Check	X	X	X		
	Finalize	X	X	X		
	Draw numbers	X				
	During Save: Before Writing Data	X	X	X		
	After Commit	X	X	X		
	After Failed Save Attempt	X	X	X		
	After Loading				X	
	Before Retrieve					

\*) Trigger Condition „Load“ is only available if at least one of the request nodes is a „transient“ one.

# Determination Configuration

Determination Time: After Modify



Reaction on modification of a business object node (create, update, delete), which need not very performance

These modifications can occur during Interaction Phase because of

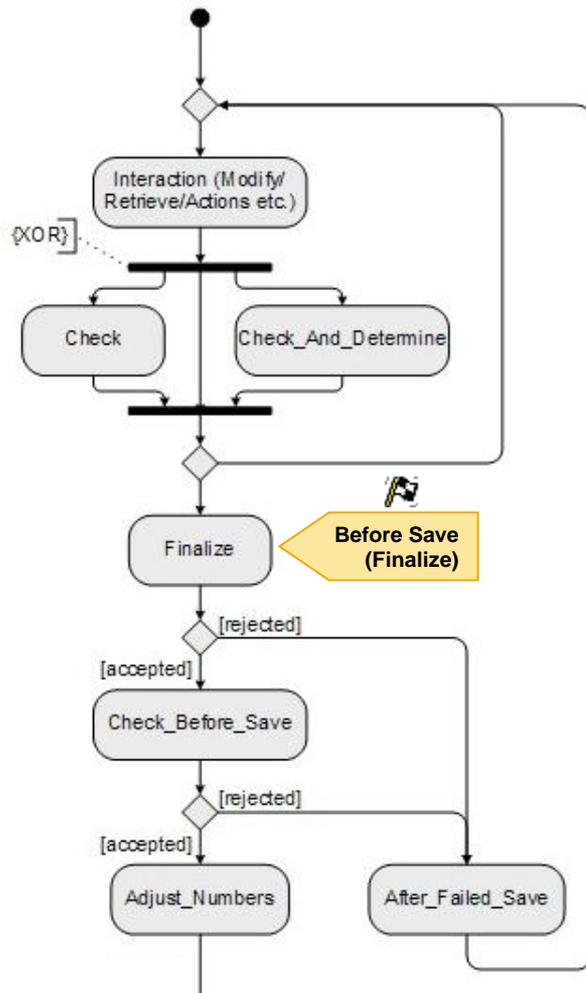
- an Service Manager call (external)
- an action implementation (internal)
- another determination at „After Modify“ that was executed before

**Main Use Case:** Calculate data that is derivable from the change

**Example:** Aggregating the item prices to the root level of an invoice after changing the quantity

# Determination Configuration

## Determination Time: Before Save (Finalize)



Determinations that are assigned to the determination time „Finalize“ are called during the save process of the transaction manager.

**Hint:** The core service „Finalize“ is potentially called more than once during one save cycle – the determination implementation must not rely on the assumption that it is called only once.

**Main Use Case:** Deriving data that is more costly

**Example:** Setting of the last change user, date, and time

# Determination Configuration

## Determination Time: After Loading

---

- Available only at trigger condition „Load“
- Available only for transient determinations, therefore only transient attributes shall be written
- Determinations assigned to the determination time „After Loading“ are called at every load and reload of node instances from the database.

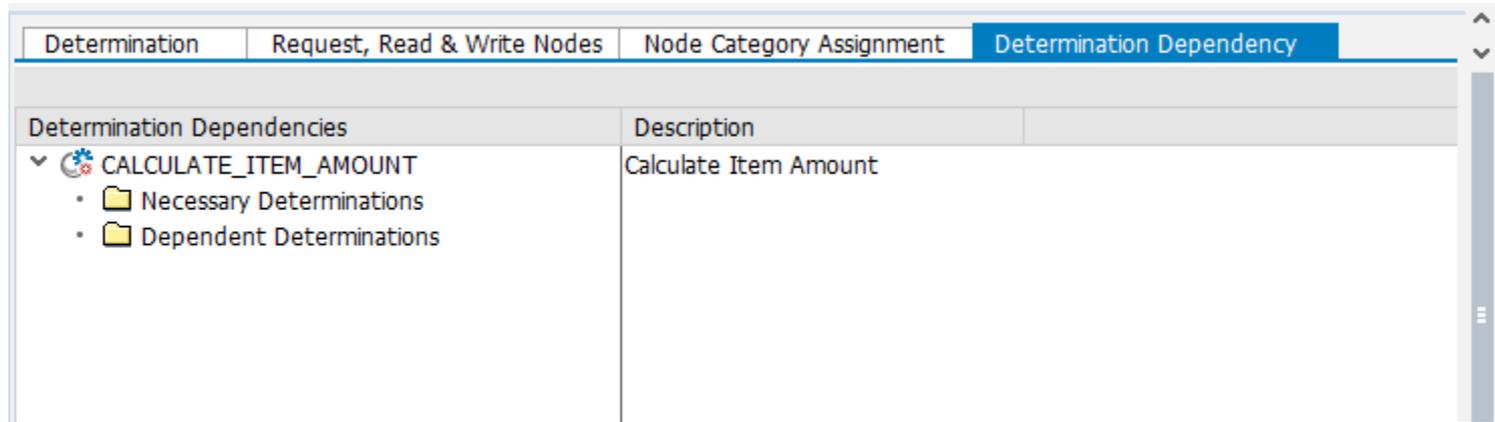
**Attention:** There is no before image available at this determination time, therefore the `check_delta` method of the determination is usually not implemented

**The One and Only Use Case:** Derivation of transient fields!

**Example:** Only the UUID is persistent, the product ID is derived from product master after each loading from database and put into a transient field. (Because only the UUID is always unique, product IDs might change.)

# Determination Configuration

## Dependencies



The screenshot shows the 'Determination Dependency' tab in the SAP configuration interface. It displays a table with two columns: 'Determination Dependencies' and 'Description'. The 'Determination Dependencies' column is expanded to show a tree structure for the determination 'CALCULATE\_ITEM\_AMOUNT'. Under this determination, there are two sub-items: 'Necessary Determinations' and 'Dependent Determinations', each represented by a folder icon. The 'Description' column contains the text 'Calculate Item Amount'.

Determination Dependencies	Description
▼  CALCULATE_ITEM_AMOUNT	Calculate Item Amount
•  Necessary Determinations	
•  Dependent Determinations	

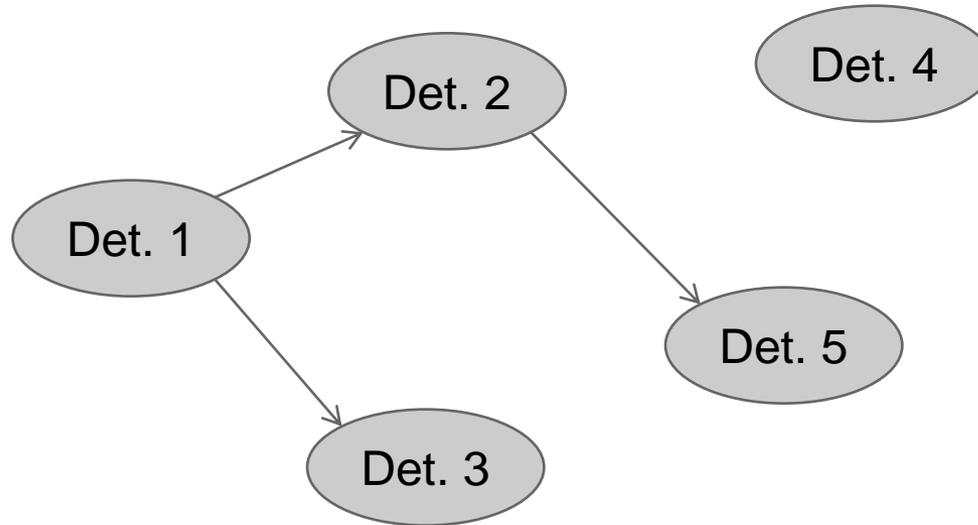
Determination dependencies define the order in which the trigger conditions are evaluated. The standard order is a linear execution, thus no loops are possible.

- **Necessary Determinations:** Determinations that are executed before this determination (if their trigger condition is satisfied).
- **Dependent Determinations:** Determinations that are triggered by changes made by this determination.

# Determination Configuration

## Dependencies

### Viewpoint: Business Object Model



### Viewpoint: Examples of Valid Execution Orders of Determination Instances

- 4,1,3,2,5 (if a determination instance of type “2” is executed, then always before “5”)
- 1,2,5,4,3 (if a determination instance of type “1” is executed, then always before “2”)

- **Attention:**

If determination 2 creates a node instance that triggers determination 5, it is necessary to define determination 2 as a necessary determination of determination 5. Otherwise – due to the linear evaluation order – determination 5 might be evaluated before the execution of determination 2 – and thus would not be triggered!



# Determination Implementation

# Determination Implementation

## Overview

1. CHECK\_DELTA  
(optional)

2. CHECK  
(optional)

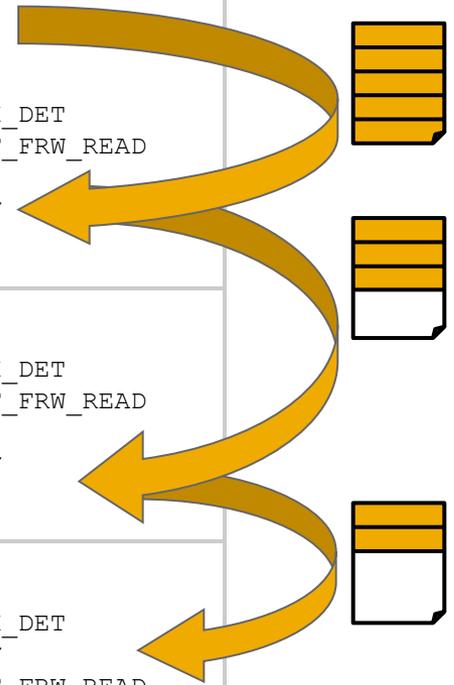
3. EXECUTE

Separation of Concerns

```
methods CHECK_DELTA
importing
  IS_CTX          type /BOBF/S_FRW_CTX_DET
  IO_READ        type ref to /BOBF/IF_FRW_READ
changing
  CT_KEY         type /BOBF/T_FRW_KEY
raising
  /BOBF/CX_FRW .

methods CHECK
importing
  IS_CTX          type /BOBF/S_FRW_CTX_DET
  IO_READ        type ref to /BOBF/IF_FRW_READ
changing
  CT_KEY         type /BOBF/T_FRW_KEY
raising
  /BOBF/CX_FRW .

methods EXECUTE
importing
  IS_CTX          type /BOBF/S_FRW_CTX_DET
  IT_KEY         type /BOBF/T_FRW_KEY
  IO_READ        type ref to /BOBF/IF_FRW_READ
  IO_MODIFY      type ref to /BOBF/IF_FRW_MODIFY
exporting
  EO_MESSAGE     type ref to /BOBF/IF_FRW_MESSAGE
  ET_FAILED_KEY type /BOBF/T_FRW_KEY
raising
  /BOBF/CX_FRW .
```



Determinations are classes implementing the `/BOBF/IF_FRW_DETERMINATION` interface. The three interface methods are executed in sequence to reduce the amount of node instance keys for which the determination logic is executed. (**Goal:** Separation of concerns!)

# Determination Implementation

Method: CHECK\_DELTA

---

```
methods CHECK_DELTA
importing
  IS_CTX type /BOBF/S_FRW_CTX_DET
  IO_READ type ref to /BOBF/IF_FRW_READ
changing
  CT_KEY type /BOBF/T_FRW_KEY
raising
  /BOBF/CX_FRW .
```

The CHECK\_DELTA method can be implemented to reduce the set of changed node instances of the assigned node to a subset of instances that are relevant.

therefore, it is possible to compare the node attributes of the assigned node instances before and after the trigger event takes place to get only the instances, on which a relevant attribute was changed.

- **IS\_CTX**: Model information about the determination (Determination Context)
- **IO\_READ**: A read object reference to read business object data
- **CT\_KEY**: Table of the node instance keys of the node to which the determination is assigned

# Determination Implementation

Method: CHECK

---

```
methods CHECK
  importing
    IS_CTX type /BOBF/S_FRW_CTX_DET
    IO_READ type ref to /BOBF/IF_FRW_READ
  changing
    CT_KEY type /BOBF/T_FRW_KEY
  raising
    /BOBF/CX_FRW .
```

The `CHECK` method reduces the amount of node instances for which the determination should be executed.

In contrast to the `CHECK_DELTA` method, the semantic of instances is taken into account. Typically, the attributes of the instances are checked against special values with this method.

**Example:** Reduces the amount of item keys in `CT_KEY` that belong to text items.

# Determination Implementation

Method: EXECUTE

---

## methods EXECUTE

### importing

IS\_CTX **type** /BOBF/S\_FRW\_CTX\_DET

IT\_KEY **type** /BOBF/T\_FRW\_KEY

IO\_READ **type ref to** /BOBF/IF\_FRW\_READ

IO\_MODIFY **type ref to** /BOBF/IF\_FRW\_MODIFY

### exporting

EO\_MESSAGE **type ref to** /BOBF/IF\_FRW\_MESSAGE

ET\_FAILED\_KEY **type** /BOBF/T\_FRW\_KEY

### raising

/BOBF/CX\_FRW.

The EXECUTE method contains the main determination business logic.

- **IO\_MODIFY**: Reference to a modify object to modify instance data.
- **EO\_MESSAGE**: Message object that contains information, error or warning messages
- **ET\_FAILED\_KEY**: Keys of IT\_KEY for which the execution of the business logic fails. (e.g. if the item quantity of one instance is negative)

# Determination Implementation

## Context

---

The Determination Context (`IS_CTX`) is input parameter of all determination methods and provides static model information about the determination.

Keys from the constants' interface of the business object that is parent of the determination in the model:

- **BO\_KEY**: Model key of the business object in which the determination is contained
- **ROOT\_NODE\_KEY**: Model key of the root node of that business object
- **NODE\_KEY**: Model key of the node that is parent of the determination
- **DET\_KEY**: Model key of this determination

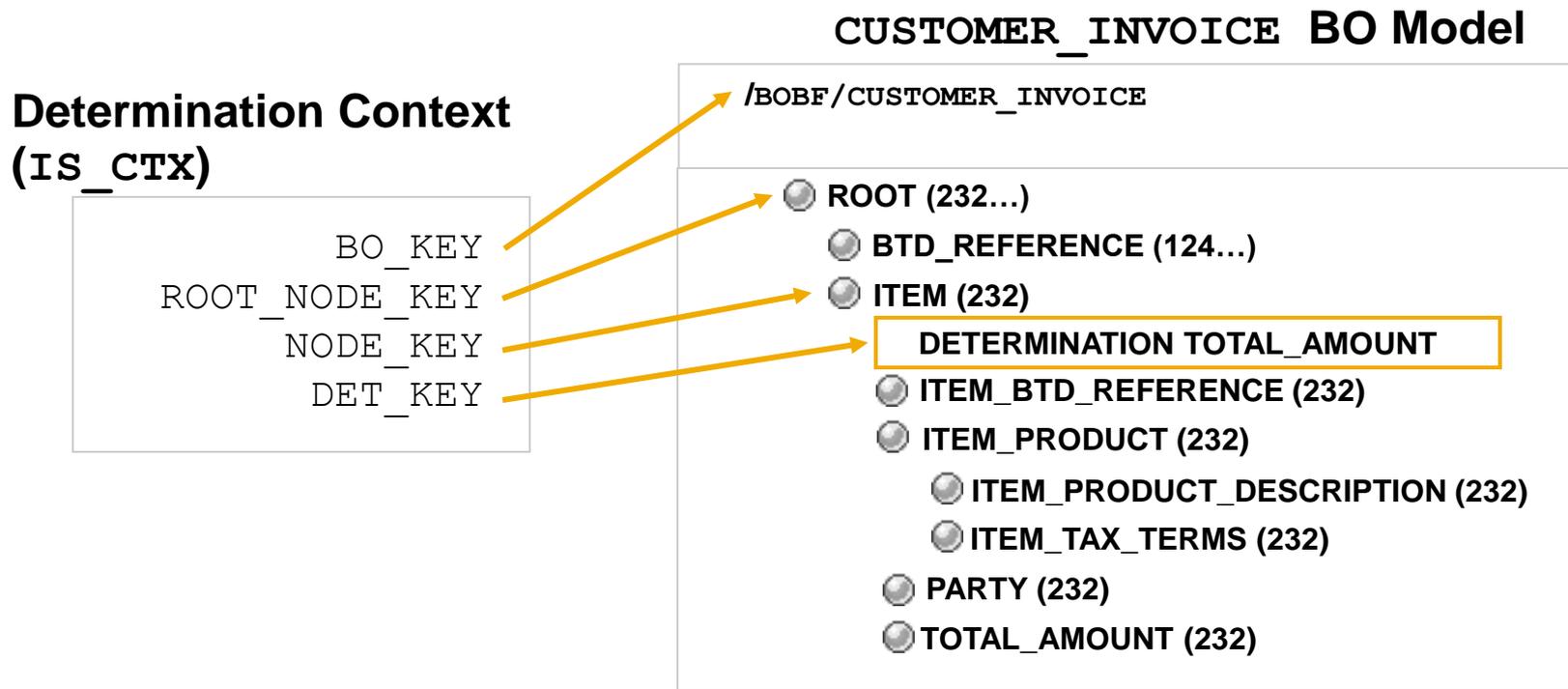
Additional information

- **EXEETIME**: Determination time at which this determination was currently triggered
- **ASSOC\_PARAMETERS**: Pointer to a filter structure
- **FACADE**: Reference

**Attention:** This information derives from the business object model  
- there is no instance information located in the `IS_CTX`!

# Determination Implementation

## Context Example





**Thank you**

# © 2012 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Excel, Outlook, PowerPoint, Silverlight, and Visual Studio are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, z10, z/VM, z/OS, OS/390, zEnterprise, PowerVM, Power Architecture, Power Systems, POWER7, POWER6+, POWER6, POWER, PowerHA, pureScale, PowerPC, BladeCenter, System Storage, Storwize, XIV, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, AIX, Intelligent Miner, WebSphere, Tivoli, Informix, and Smarter Planet are trademarks or registered trademarks of IBM Corporation.

Linux is the registered trademark of Linus Torvalds in the United States and other countries.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are trademarks or registered trademarks of Adobe Systems Incorporated in the United States and other countries.

Oracle and Java are registered trademarks of Oracle and its affiliates.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems Inc.

HTML, XML, XHTML, and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Apple, App Store, iBooks, iPad, iPhone, iPhoto, iPod, iTunes, Multi-Touch, Objective-C, Retina, Safari, Siri, and Xcode are trademarks or registered trademarks of Apple Inc.

IOS is a registered trademark of Cisco Systems Inc.

RIM, BlackBerry, BBM, BlackBerry Curve, BlackBerry Bold, BlackBerry Pearl, BlackBerry Torch, BlackBerry Storm, BlackBerry Storm2, BlackBerry PlayBook, and BlackBerry App World are trademarks or registered trademarks of Research in Motion Limited.

Google App Engine, Google Apps, Google Checkout, Google Data API, Google Maps, Google Mobile Ads, Google Mobile Updater, Google Mobile, Google Store, Google Sync, Google Updater, Google Voice, Google Mail, Gmail, YouTube, Dalvik and Android are trademarks or registered trademarks of Google Inc.

INTERMEC is a registered trademark of Intermec Technologies Corporation.

Wi-Fi is a registered trademark of Wi-Fi Alliance.

Bluetooth is a registered trademark of Bluetooth SIG Inc.

Motorola is a registered trademark of Motorola Trademark Holdings LLC.

Computop is a registered trademark of Computop Wirtschaftsinformatik GmbH.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP BusinessObjects Explorer, StreamWork, SAP HANA, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects Software Ltd. Business Objects is an SAP company.

Sybase and Adaptive Server, iAnywhere, Sybase 365, SQL Anywhere, and other Sybase products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Sybase Inc. Sybase is an SAP company.

Crossgate, m@gic EDDY, B2B 360°, and B2B 360° Services are registered trademarks of Crossgate AG in Germany and other countries. Crossgate is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

The information in this document is proprietary to SAP. No part of this document may be reproduced, copied, or transmitted in any form or for any purpose without the express prior written permission of SAP AG.