



Considerations when upgrading SAP Business Workflow from SAP R/3 4.6C to SAP ERP 6.0

Content Providers

Eddie Morris: SAP NetWeaver Tools Support, SAP Ireland

Carol Thomas: SAP NetWeaver Consultant, SAP America

Bob Duffy: SAP NetWeaver Consultant, SAP America

Thomas Kosog: Platinum SAP NetWeaver Consultant, SAP America

Anna Hill: SAP NetWeaver Consultant, SAP UK

Mark Griffiths: SAP NetWeaver Consultant, SAP UK

Jocelyn Dart: SAP NetWeaver Consultant, SAP Australia

Nicole Fessler: Workflow Developer, SAP AG

Markus Kinateder: Senior Developer, SAP AG

Ginger Gatling: NetWeaver Product Manager, SAP Labs



Preparing for the workflow upgrade

During and after the upgrade

New workflow features since R/3 4.6c

Summary and Appendices

Before the upgrade

- **Complete all running workflow instances**
- **Create workflow test plans**
- **WF& T versus WS & TS**
 - ◆ **Not 'required' to switch (Workflow Tasks (WF) and Customer Tasks (T)) to workflow templates (WS) and standard task (TS).**
 - ◆ **Recommend to copy from WF/T to WS/TS.**
 - ◆ **Change can be done after the upgrade (done over time). You can use the transaction PFTC and the copy functionality to do.**

■ Clean runtime tables before the upgrade

- ◆ If the workflow runtime tables have less data in them then this will enable the upgrade to run faster. The best option is to archive as much workflow runtime data as possible according to note 573656. If you have runtime data that is not needed in the future then you can consider deleting this data via report RSWWWIDE (See note 49545). Please be aware that once deleted that data is no longer available. Various workflow database tables may contain unnecessary entries after you delete work items. It is also a good idea to run report RSWWWIDE_DEP which removes these unnecessary entries.
- ◆ 573656 - Collective note relating to Archiving in workflow
- ◆ 49545 - Deleting unnecessary work items
- ◆ 738148 - Deletion report for superfluous work item entries (46C)



Preparing for the workflow upgrade

During and after the upgrade

New workflow features since R/3 4.6c

Summary and Appendices

During the upgrade

■ Conversion of event linkage tables

- ◆ 1019080 - Poor RSWFEVTPRA performance
- ◆ 808790 - Poor RSWFEVTPRA performance
- ◆ In release 4.6C, the instance linkage for workflow steps was stored in table SWEINSTCOU. In WAS (Basis 620) onwards, this information is stored in table SWFDEVINST. During the upgrade, report RSWFEVTPRA moves the entries from the old table into the new table.

NOTE: If you do the deletion of work items these tables will be cleaned up as well.

After the upgrade

■ Composite Upgrade note

- ◆ 1068627 - Composite note about workflow upgrade
- ◆ This note lists the notes concerning the workflow that you must refer to when you upgrade to 6.40 or 7.00. Please make sure all of the notes listed are applied to your system and you have reviewed any recommendations within the notes.

■ Workflow Definition & Binding

◆ Workflow definition

- 1060762 - Container operation with date fields and time fields (date check routine in workflow builder)
- 1058159 - Elements are missing in the workflow container (DDIC elements no longer active)

◆ Binding checks are much more strict in NW04 and 7.0. Binding definitions that did not error or cause problems may now cause problems. Your workflow template containers may refer to DDIC data types that no longer exists in the upgraded system. Note 939489 discusses this issue and provides 2 solutions:

- Solution 1: The easiest solution is to recreate the missing type definition in the ABAP Dictionary.
- Solution 2: If Solution 1 is not feasible or useful, you can replace the data types in the container definitions for the relevant processes with a report. Bear in mind that you have to replace them with the same type. Apply note 939489 in order to have the report RSWF_CNT_BOR_ELEM_REPLACE.

◆ 939489 - Workflow: Container elements are missing after an upgrade

New Binding Editor

The screenshot shows the 'Change Binding For Step' dialog box in SAP. It is divided into two main panes: a 'source container' on the left and a 'target container' on the right. Both panes display a list of fields with their descriptions. A large yellow arrow labeled 'drag & drop' points from the source container to the target container. Below the panes, there are two tables of binding instructions. The top table is labeled 'import binding' and the bottom table is labeled 'export binding'. Two yellow arrows labeled 'drag & drop' point from the source and target containers down to their respective binding tables. A green callout box labeled 'binding instructions' points to the right table. The interface includes a toolbar with various icons and a status bar at the bottom.

source container

target container

drag & drop

import binding

export binding

binding instructions

Binding Expressions

Expressions now provide index access to multi-line elements

<code>&table[index]&</code>	Access to an entire table line
<code>&table[index].columnName&</code>	Access to a component of a table line
<code>&table[].columnName&</code>	Projection to a single column
<code>&customers[1].orders[2]&</code>	Second order of first customer

Expressions can contain functional method calls to BOR or ABAP Objects

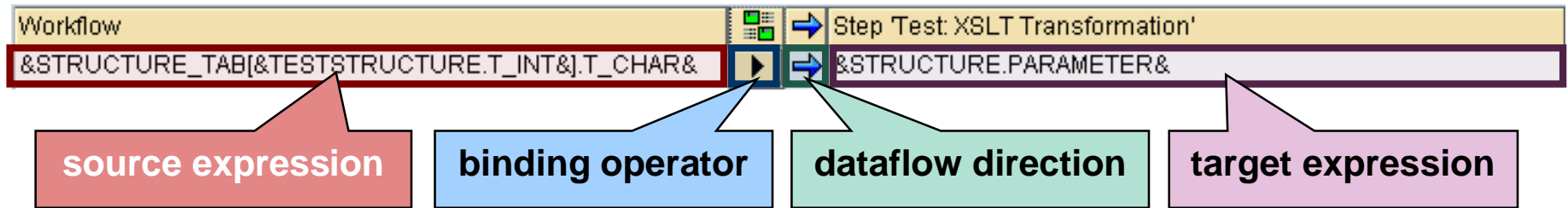
- The expression evaluates to the result value of the method
- It is possible to pass parameters
- Methods used in expressions must not have side effects
 - ◆ „Read only“ methods
 - ◆ No database changes

Instance method w/o parameters	<code>&my_object.get_value()&</code>
Instance method w parameters	<code>&my_object.methodA(param1=&exp1& param2=17)&</code>
Static method	<code>%my_class.static_method(param1=&exp2&)%</code>

No more restrictions concerning data types

- ABAP-OO object references
- Strings
- Nested structures

Binding Instructions










Source and Target of a binding instruction are expressions

- Assigning a table line to a structure component is now possible (if compatible)

Several binding operators are available

- for expression to expression binding (as above)
- for container to container binding

The  button lets you choose the right binding operator

Expression to expression		Assign value
		Append to table
		Initialize
		XSLT Transformation
Container to container		Merge (Copy all elements)
		Programmed Binding (Container-IF)
		Programmed Binding (SWCONT)

→ see Appendix for further configuration possibilities

■ Tables SWW_CONTOB & SWW_CONT

- ◆ Tables SWW_CONTOB & SWW_CONT are used in Basis 640 onwards, but only if you choose the 'old' container persistence. When using XML persistence in your workflow no entries will be written to these tables. You can change the settings for the persistence profile of a workflow via the builder => Basic Data => Version Dependent => Control tab. Look in the 'Persistence profile' tab and you can change the settings.
- ◆ Entries with XML persistence are stored in table SWWCNTP0. This should not have any affect on the system at all. Only issue will be if the customer has some custom code that reads from tables SWW_CONTOB & SWW_CONT.

Complete reimplementatation of container technology

No restrictions concerning data types

- **DDIC types with arbitrary lengths**
- **Binary data**
- **Structures with arbitrary depth and complexity**
- **ABAP OO references**

Container is by default stored as XML document

- **Facilitates diversity of supported data types**
- **Saves storage space (especially for large containers/structures)**
 - No database entry for each structure field required anymore
- **Entails serialisation / deserialisation effort**

Old container persistence (SWCONT, SWCONT OB) still supported

- **Compatibility mode**
- **Workflow Builder setting (→ Appendix)**
- **Old container persistence (SWCONT, SWCONT OB) still supported**

Parameter Container Interface

The old container was represented as an internal table

→ Manipulating the container meant using awkward container macros

The new container implementation is ABAP Objects based

A (released) API is provided for reading and manipulating container data

→ the ABAP Objects interface `IF_SWF_IFS_PARAMETER_CONTAINER`

This interface represents the container already in a number of workflow APIs (and will get more ubiquitous from release to release)

`IF_SWF_IFS_PARAMETER_CONTAINER` **contains the following methods:**

Method	Description
CLEAR	Resets an element to its type-based initial value
GET	Delivers the value of an element
GET_TYPE	Delivers the data type of an element
GET_VALUE_REF	Delivers a reference to the value of an element
LIST_NAMES	Delivers a list of names of all elements
SET	Sets the value of an element

After the upgrade

■ WEB

- ◆ BWSP -> Integrated ITS
- ◆ Universal Worklist (Recommended)
- ◆ SBWP, inbox in SAPGui

■ When moving to Unicode:

- ◆ 775111 - Unicode problem in BOR container
- ◆ 1019850 - Contents of WI_Creator field
- ◆ 967414 - Error in binding editor with double-byte characters after upgrade

After the upgrade

■ SAPGUI

- ◆ If you experience short dumps relating to `MESSAGE_TYPE_X` it may be due to your SAPGUI Release and patch level. It is a good idea to also upgrade you SAPGUI to the latest release and patch level available on SAP Service Marketplace.

■ Deadlines

- ◆ 1092157 - Deadline issues

■ Hanging Workflow

- ◆ 215753 - Upgrade: Old workflows hang

■ Transports

- ◆ 571302 - Collective note relating to transports in workflow
- ◆ 850556 - Workflow: Report for deleting the shared buffer
- ◆ 980834 - Buffering of workflow definitions
- ◆ 982352 - Updating workflows in the test system (only needed if using both WF and WS tasks)

■ Business Object Issues

- ◆ If you experience short dumps with `DATA_UC_STRUCT_C_LENGTH` or `ASSIGN_LENGTH_0` then this may be due to the business object(s) being used in your workflow.
 - Solution: Generate the object and any subtypes associated with the object via SWO1
- ◆ Transaction `SWO_ASYNC`
 - After upgrade some users find that they can no longer display the object method by clicking the link in the preview pane of SBWP. This also may affect the users ability to view work item attachments or secondary methods. This is due to missing authorization of transaction `SWO_ASYNC` which is detailed in note 1006235 - Authorization check for transaction `SWO_ASYNC`.
- ◆ Custom code
 - `RUNTIME HANDLES (TYPE SWC_OBJECT)` versus `PERSISTENT OBJECT REFERENCES (TYPE SWOTOBJID)`.
- ◆ SAP ERP 6.0 is more stringent in enforcement of ABAP syntax. Example:
 - “We had a macro call which was coded `swc_refresh_object "ZECM"`. I don't know how they ever got away with using the object name in quotes. All of the documentation pointed to defining a field to hold the object values so the call should have been `swc_refresh_object self` (self being a field holding the value in the correct format)”



Preparing for the workflow upgrade

During and after the upgrade

New workflow features since R/3 4.6c

Summary and Appendices

New workflow features since R/3 4.6c

Design Time Changes and SAP_WAPI at runtime

Delivering work items

Reporting

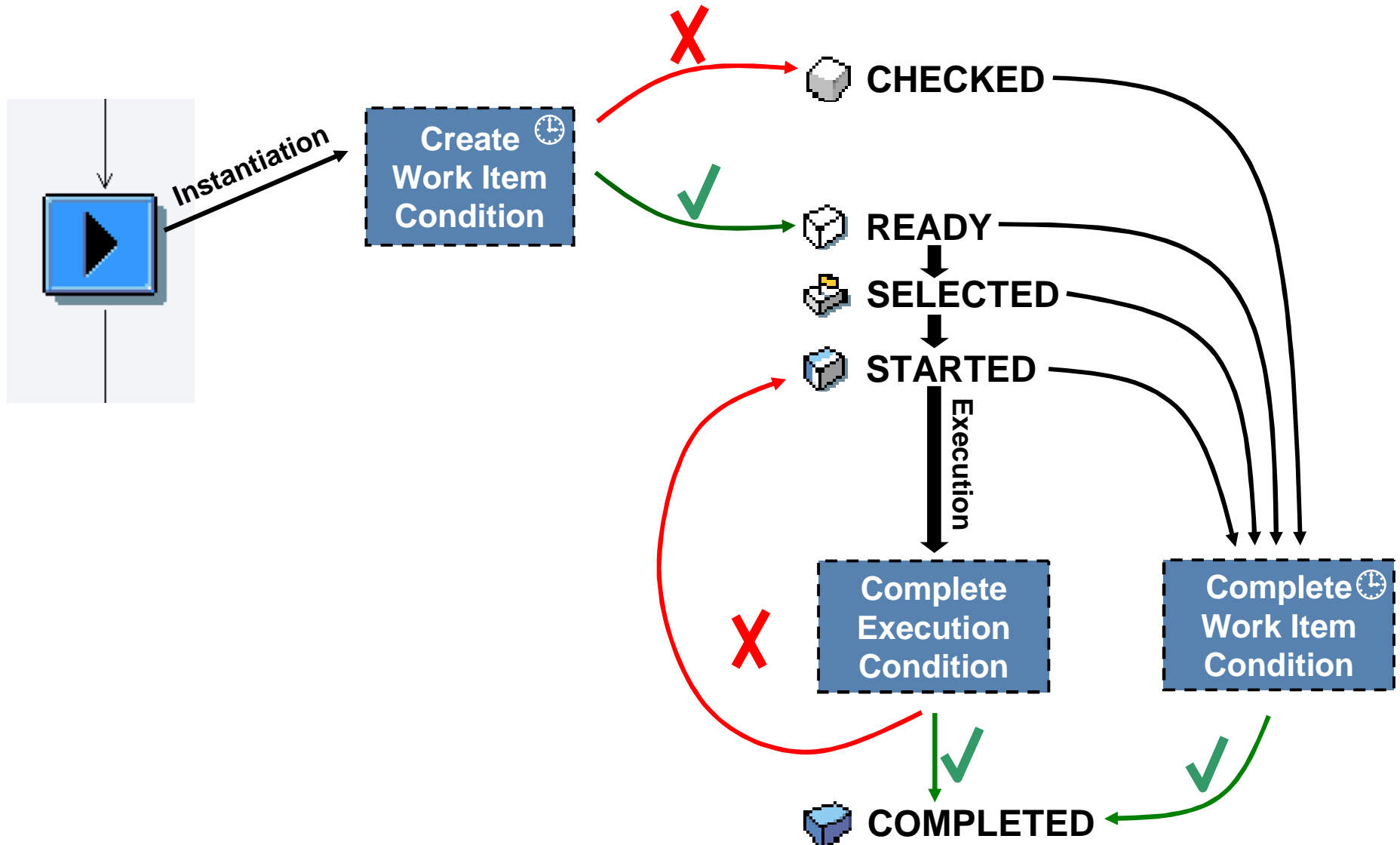
Note: We only provide overview of changes in this presentation. There is an appendix we will not cover that has more details and there is a delta class, DBITWF.

SAP_WAPI

Repository Info System: Function Modules Find (76 Hits)

Function group	Function group short text
Function Module Name	Short text for function module
SWF_API_IO	
SAP_WAPI_MESSAGE_HANDLER	
SWRA	
SAP_WAPI_ADM_WORKFLOW_CANCEL	
SAP_WAPI_ADM_WORKFLOW_RESTART	
SAP_WAPI_ADM_WORKFLOW_RESUME	
SAP_WAPI_ADM_WORKFLOW_SUSPEND	
SAP_WAPI_ADM_WORKITEM_BACK	
SAP_WAPI_ADM_WORKITEM_REDORULE	
SWRC	
Workflow interfaces: Work list client	
SAP_WAPI_ATTACHMENT_ADD	
SAP_WAPI_ATTACHMENT_ADD_REF	
SAP_WAPI_ATTACHMENT_DELETE	
SAP_WAPI_CHANGE_WORKITEM_PRIOD	Workflow Interfaces: Change Priority of Work Item
SAP_WAPI_COUNT_WORKITEMS	Workflow Interfaces: Number of Work Items for User
SAP_WAPI_END_RESUBMISSION	
SAP_WAPI_EXECUTE_WORKITEM	Workflow Interfaces: Execute Work Item
SAP_WAPI_FORWARD_WORKITEM	Workflow Interfaces: Forward Work Item
SAP_WAPI_GET_ATTACHMENTS	Workflow Interfaces: Read Attachment for Work Item
SAP_WAPI_GET_DEADLINES	Workflow Interfaces: Read Work Item Deadlines
SAP_WAPI_GET_HEADER	Workflow Interfaces: Read Work Item Header
SAP_WAPI_GET_METHODS	Workflow Interfaces: Read Work Item Methods
SAP_WAPI_GET_MULTI_EXEC_GUID	

Step Conditions



→ see Appendix for further details

SWITCH: The New Multiple Condition Flavor

1. Double-click on multiple condition icon

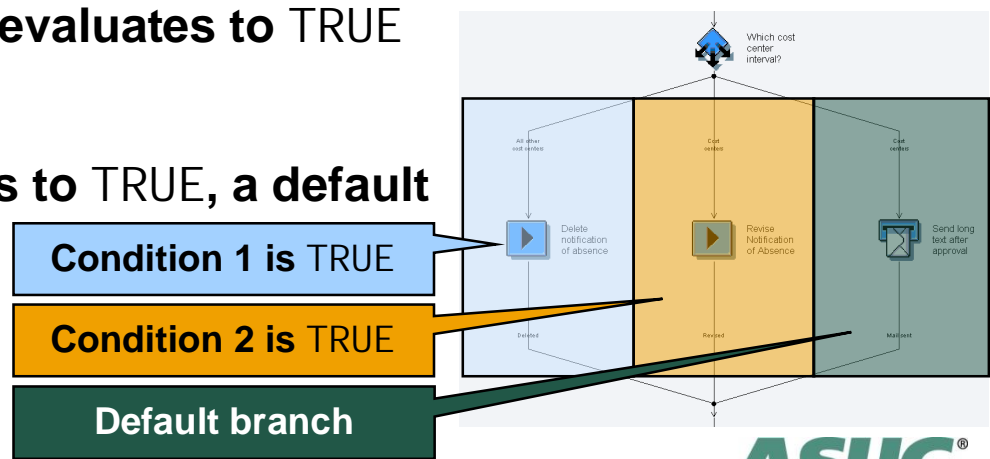
2. Select Multiple Condition Flavour

3. Enter Branches and Conditions

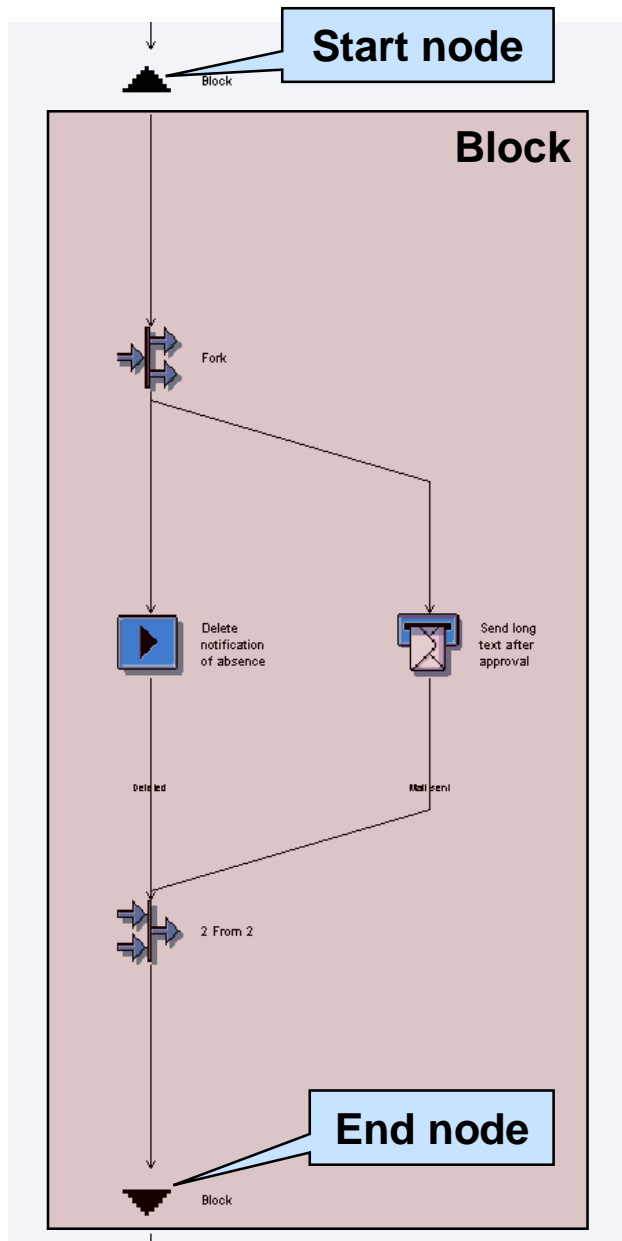
4. Name Default Branch

Condition Editor

- Each branch of the SWITCH carries a condition definition
- The conditions are evaluated in the given order
- The first branch whose condition evaluates to TRUE is taken
- If none of the conditions evaluates to TRUE, a default branch is taken



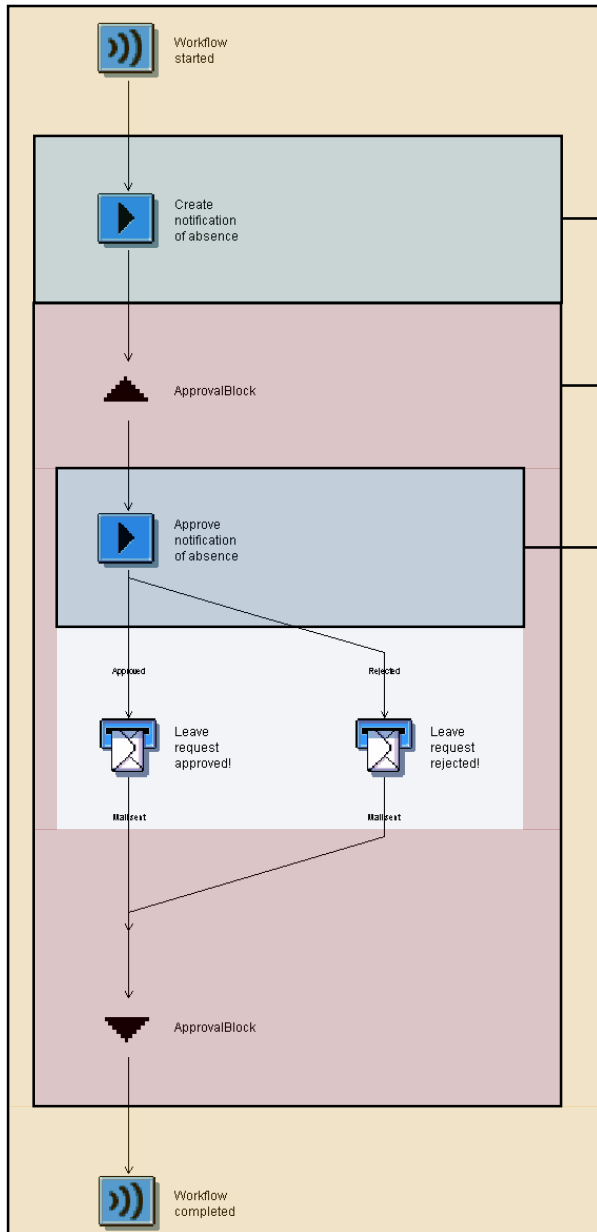
Blocks: Basic Features



Blocks...

- are modeling elements
- can contain other modeling elements
- represent a data sub-context
 - all modeling elements within the block refer to this data context (binding, conditions, ...)
- have a data interface
- have one start and one end node
- are represented by block work items
 - new work item type
- can be deadline monitored (latest end only)
- can catch exceptions (→ see Appendix)

Blocks: Work Item Hierarchy

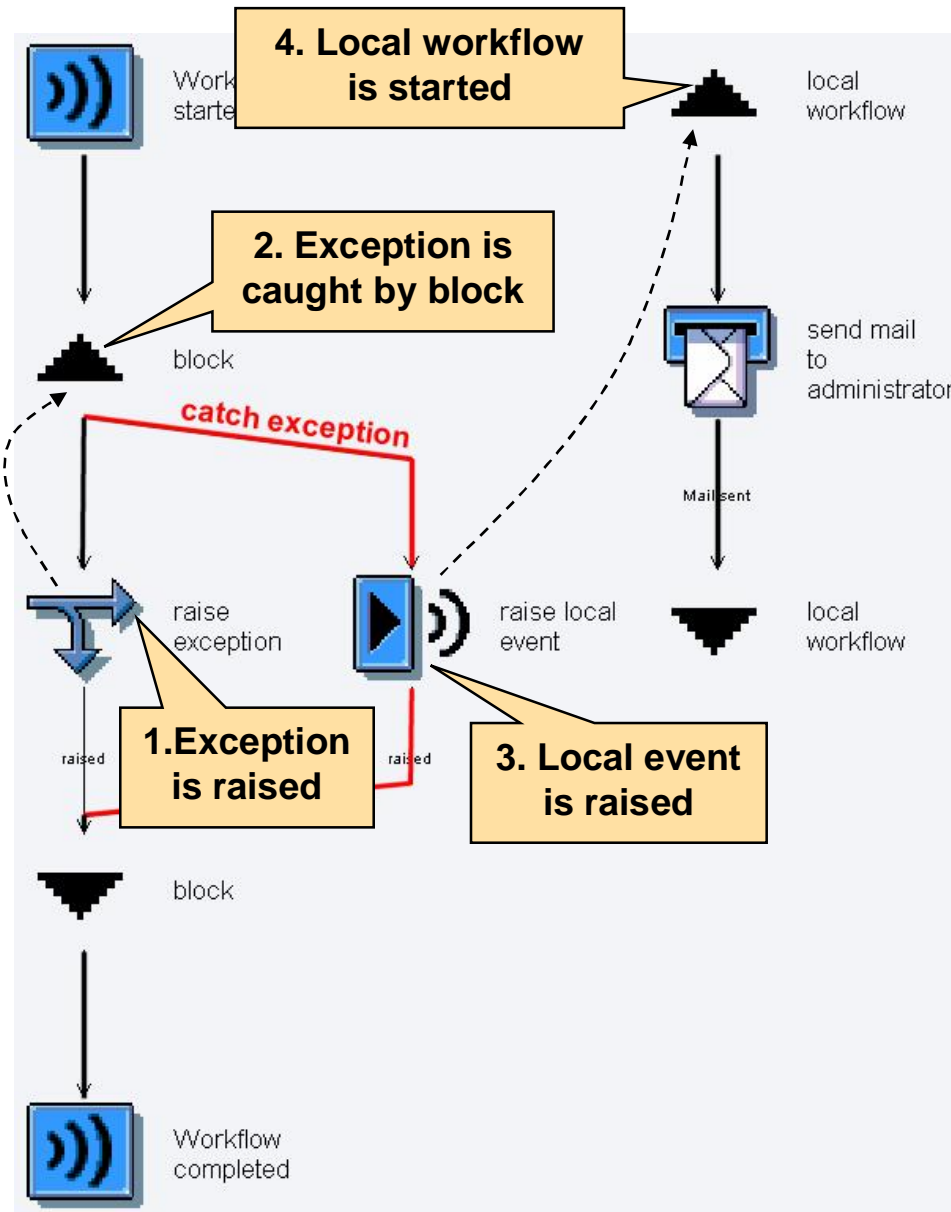


Workflow and task	Details	Graphic	Agent	Status	Result	Date	Time
Approval with Block				In Process	Workflow started	22.08.2005	12:32:23
Create Notification of Absence				Completed		22.08.2005	12:32:23
ApprovalBlock				In Process		22.08.2005	12:32:25
Employee D034138 : Approve notification of absence				In Process		22.08.2005	12:32:25

Details for step: ApprovalBlock						
Agent	Executed Action	Date	Time	Object	Object Name	
D034138	Block item (local Workflow) created	22.08.2005	12:32:25			

- **Block items introduce new levels in work item hierarchies**
 - ◆ **Blocks can contain dependent work items**
- **Blocks are represented as folders in the workflow log**
- **Block structure is sustained at runtime**
 - ◆ **Block is completed only if dependent work items are in a final state**
 - ◆ **If a block is cancelled, all its dependent work items are cancelled, too**

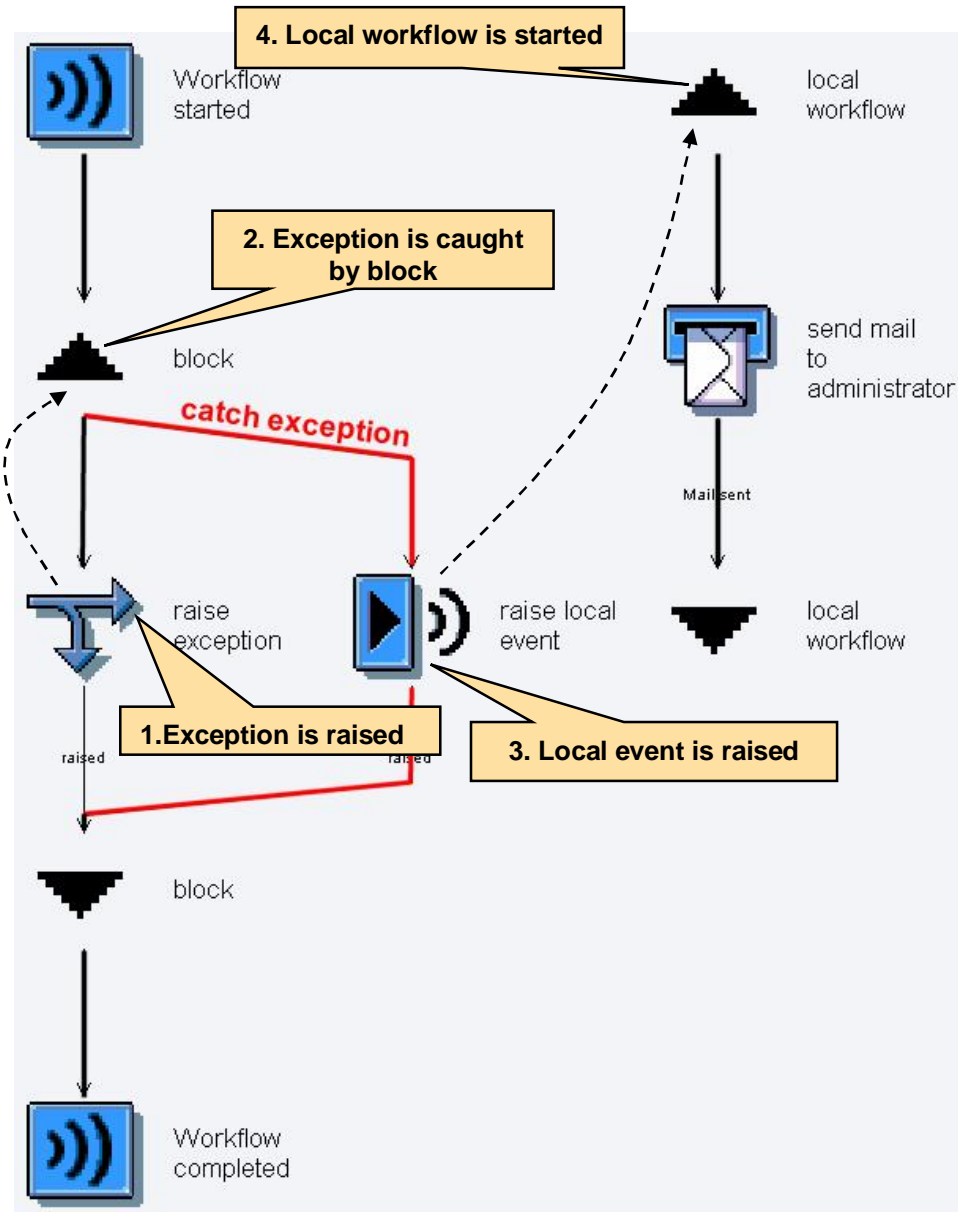
Local Workflows



Local workflows...

- are control flow snippets
 - not part of the main process sequence
- are just special blocks
 - represented by block work items
- are started via local events
 - data binding event – local flow
- have full access to process data
- can be instantiated an arbitrary number of times
- are cancelled if still active when main process completes

Local Workflows



Workflow Header Events (1)

1. Select „Basic Data“

2. Select „Version-Dependent“

3. Select „Events“

4. Enter event data

5. Choose receiver type

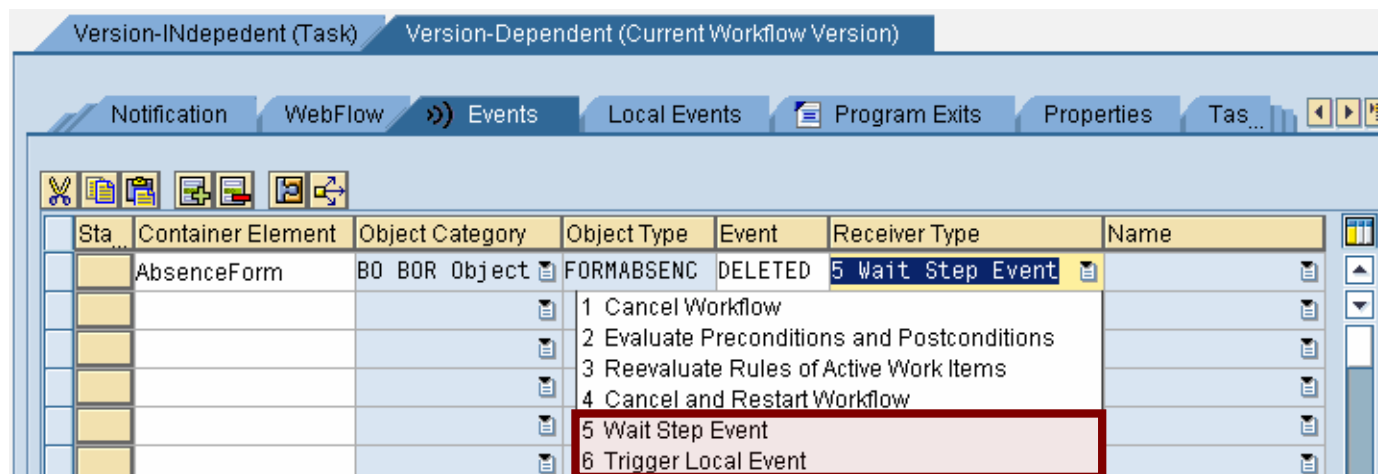
Container Element	Object Category	Object Type	Event	Receiver Type
AbsenceForm	BO BOR Object	FORMABSENC	DELETED	1 Cancel Workflow

- 1 Cancel Workflow
- 2 Evaluate Preconditions and Postconditions
- 3 Reevaluate Rules of Active Work Items
- 4 Cancel and Restart Workflow

- The workflow instance can listen to events
- Event couplings only exist if the corresponding object exists in the workflow container
- The receiver type tells you how the workflow processes the event

Receiver Type	Reaction of Workflow to Event
Cancel Workflow	<ul style="list-style-type: none"> ■ The workflow instance is set to status CANCELLED
Cancel and Restart Workflow	<ul style="list-style-type: none"> ■ The workflow instance is set to status CANCELLED ■ A new instance is started with the same data
Evaluate Pre- and Postconditions	<ul style="list-style-type: none"> ■ Conditions „Create Work Item“ and „Complete Work Item“ are evaluated
Reevaluate Rules	<ul style="list-style-type: none"> ■ Agent determination of all active dialog work items is repeated

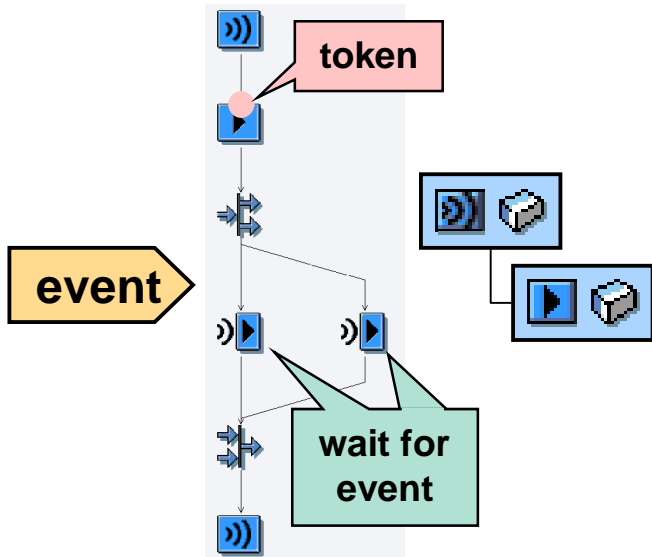
Workflow Header Events (2)



■ WAS 640 (NW04) offers you two new receiver types

Receiver Type	Reaction of Workflow to Event
Wait Step Event	<ul style="list-style-type: none"> ■ Event is delivered to existing appropriate wait step (event work item) ■ If no active event work item exists, the event is parked ➔ see one of the next slides for parked events and the extended wait step!
Trigger Local Event	<ul style="list-style-type: none"> ■ A specified local event is triggered ■ This local event could <ul style="list-style-type: none"> ◆ start local workflows ◆ complete wait steps (event items) ➔ see one of the next slides for the extended wait step!

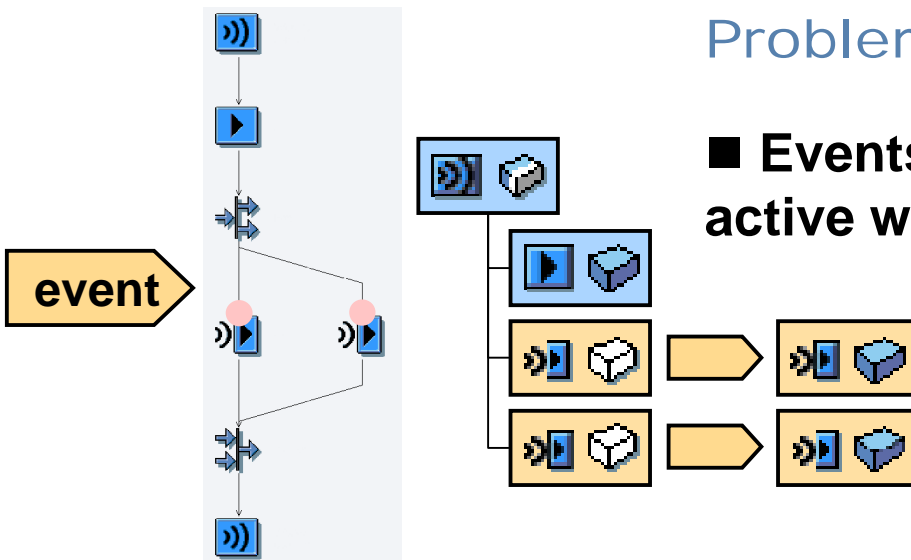
Parked Events (1)



Problem/Feature 1:

- Workflow can only react on events if appropriate wait step has already been instantiated

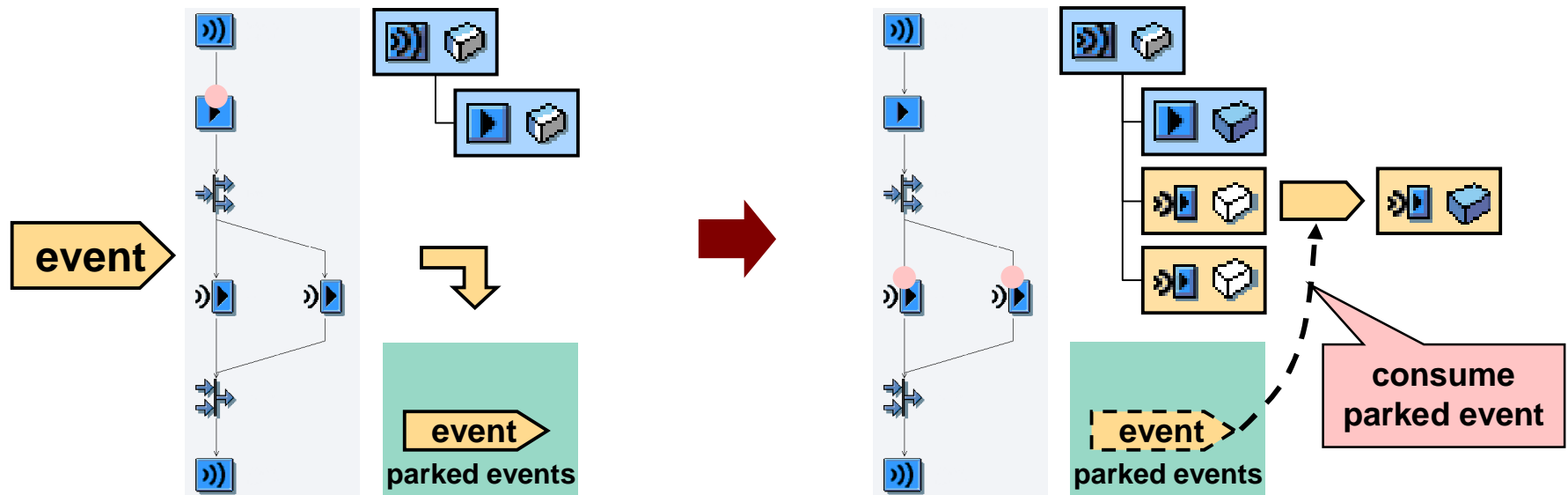
- If event is raised „too early“, it is lost for the process



Problem/Feature 2:

- Events always complete all corresponding active wait step instances (event items)

Parked Events (2)



The process instance can act as an intermediate event storage

- If active receiver event items exists, the one created first receives the event
- If no active receiver event item exists, the event is parked
- The first matching event item created consumes the first event parked
- In any case: One event is delivered to exactly one event item

→ See Appendix for further details

Wait Step: Four Different Flavours

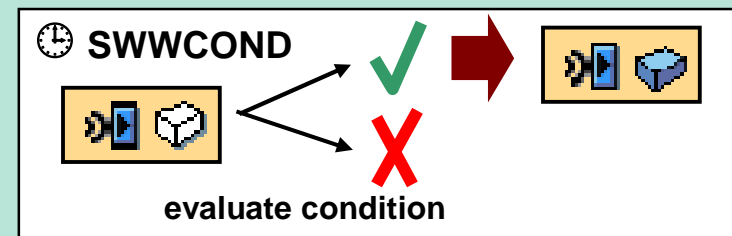
Wait for event

- Conventional wait step
- Wait for global event
- Correlations can be used



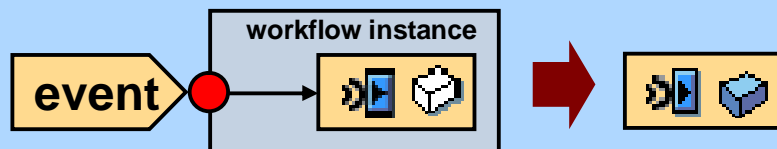
Wait for condition

- Condition specified in wait step
- Evaluated by periodic background job SWWCOND



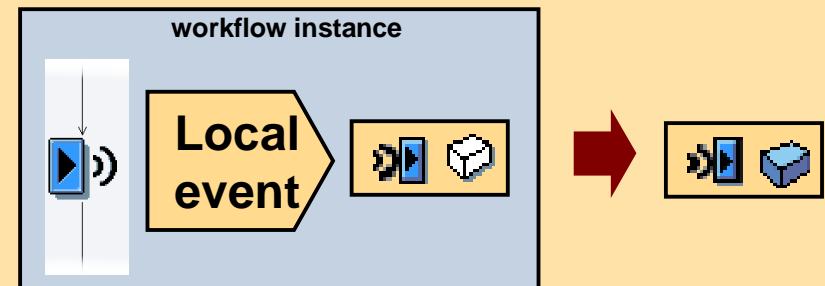
Wait for event using workflow

- Wait for global event
 - ◆ received by workflow instance
 - ◆ dispatched to event item
- Correlations can be used



Wait for local event

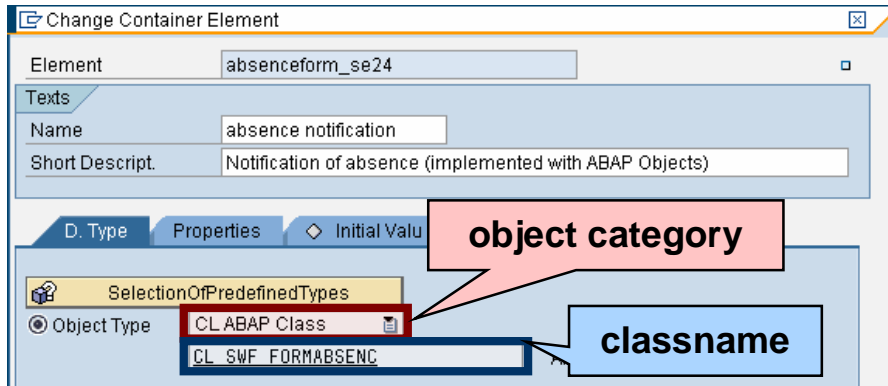
- Wait for local event raised by trigger step within the same workflow instance



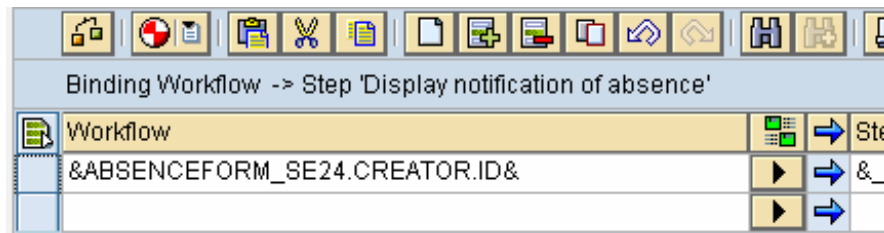
Using ABAP Objects in Workflows (1)

- **SAP Business Workflow now supports two object repositories**
 - ◆ BOR (Business Object Repository)
 - ◆ SE24 (ABAP Objects)
- **Persistent object keys now have three components**
 - ◆ Object category (CATID) (→ BOR or ABAP Objects)
 - ◆ Object type (TYPEID) (→ BOR object type or ABAP Objects classname)
 - ◆ Object key (INSTID)
- **Both repositories equally well supported**
 - ◆ But: No delegation feature in SE24
- **SE24 support covers the same aspects as BOR support**
 - ◆ Container
 - ◆ Expressions
 - ◆ Methods
 - ◆ Events

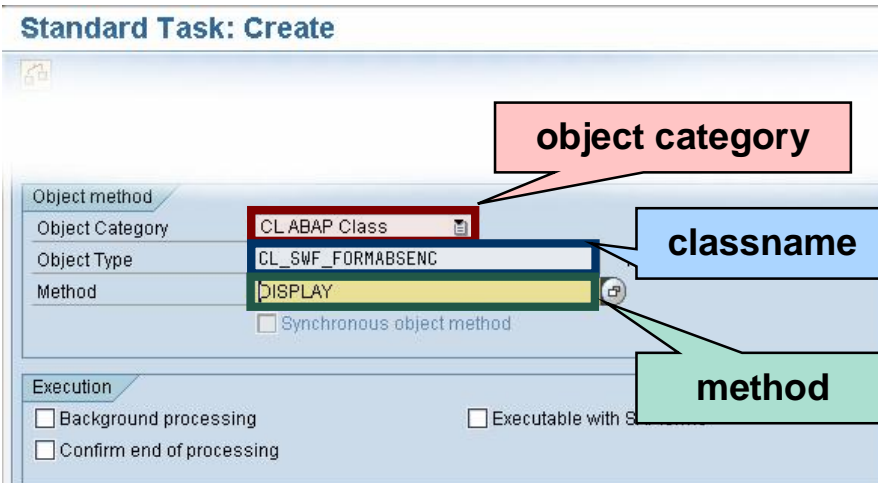
Using ABAP Objects in Workflows (2)



- Container can reference ABAP OO objects



- ABAP OO objects can be used in expressions
 - ◆ just as BOR objects are used



- ABAP OO object methods can be used in standard task definitions
 - ◆ just as BOR object methods are used

New workflow features since R/3 4.6c

Design Time Changes and SAP_WAPI at runtime

Delivering work items

Reporting


Note: We only provide overview of changes in this presentation. There is an appendix we will not cover that has more details and there is a delta class, DBITWF.

Universal Worklist

- Give users a **unified and centralized access** to their work and the relevant information
- Aggregate task items from **multiple** and different **systems** of the system landscape - in one list for one-stop access
 - ◆ Business Workflow
 - ◆ Collaboration Task
 - ◆ Alert Management System
 - ◆ KM Content Management Notifications
- Enable users for direct **decisions** and **actions**
- Support users in **personalization** of the presentation
- Allow extensive **customization** for specialized work lists in Work Centers, including custom attributes, actions and bulk processing

UWL 7.0 – Task list

Welcome Matthias Kruse



Help | Personalize | Log Off 

Collaboration | Search [Advanced Search](#)

Home | [Content Administration](#) | [User Administration](#) | [System Administration](#)











Company | **Work** | [Teams](#) | [Documents](#) | [Portal Information](#)

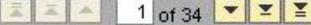
Overview | History [Back](#) [Forward](#)

Work Inbox: Tasks in "New and in progress tasks (11 / 27)" view  

Tasks (11 / 27) | Alerts | Notifications | Tracking

Show [New and in progress tasks \(11 / 27\)](#) <select a subview> All [Create Task](#) [Show Filters](#) [Hide Preview](#)

Subject	From	Sent	Priority	Due	Status
 JDK1.3/1.4 problem for NW 2004s/7.0 (Downport)	Cartman, Eric	Feb 14, 2005	High	Yesterday	In Progress
 Provide UWL status	Kruse, Matthias	Today	High	Tomorrow	New
 Demo NW Team PA	Kruse, Matthias	Feb 15, 2005	High	Feb 22, 2005	New
 Initiate Employee Performance Management	Kruse, Matthias	Feb 14, 2005	High	Mar 1, 2005	New
 Notification of absence number 0000002052 rejected. What do you want to do?	Administrator, Admin	Yesterday	Normal		New
 Notification of absence number 0000002092 rejected. What do you want to do?	Workflow System	Yesterday	Normal	3	New
 Notification of absence number 0000002081 rejected. What do you want to do?	Workflow System	Yesterday	Normal		New
 Notification of absence from 16.10.2003: Revise	Workflow System	Feb 15, 2005	Normal	2	New
 Notification of absence from : Revise	Workflow System	Feb 15, 2005	Normal	1	New
 Employee CARTMAN: Approve notification of absence	Workflow System	Feb 11, 2005	Normal	1	In Progress


1 of 34 

[Edit](#) [Forward](#)


JDK1.3/1.4 problem for NW 2004s/7.0 (Downport)

Due: Yesterday by Kruse, Matthias

Progress: 50%

Escalated: 

Tracked By: Cartman, Eric

Priority: High 

Sent: Feb 14, 2005 by Cartman, Eric

Status: In Progress

As you may know parts of NW04 must be build with J2SE 1.3. Upgrading these parts in NW 2004s/7.0 in such a way that they can be build with J2SE 1.4 seems too much effort and would bring too much instability in the NW 2004s codeline. On the other hand Sun does not support that 1.3 compliant Java sources are built with an 1.3 compliant compiler (not even by using cross-compilation) if they have to be compiled against 1.4 class files as indicated in the following picture. Action Item: If your component does currently not comply with the above rule and you are compiling with 1.4, we recommend that you also check immediately how your project is affected by doing a local build with a J2SE 1.3.

UWL 7.0 – Launch Work Transaction

Tasks (11 / 27) Alerts Notifications Tracking

Show <select a subview> All [Show Filters](#) [Hide Preview](#)

Subject	From	Sent	Priority	Due	Status
<input checked="" type="checkbox"/> Notification of absence from 16.10.2003: Revise	Workflow System	Feb 15, 2005	Normal	1	New
<input type="checkbox"/> Notification of absence from : Revise	Workflow System	Feb 15, 2005	Normal	1	New
<input type="checkbox"/> Employee CARTMAN : Approve notification of absence	Workflow System	Feb 11, 2005	Normal	1	In Progress

Welcome Matthias Kruse [Help](#) [Personalize](#) [Log Off](#)

Collaboration | [Advanced Search](#)

Home [Content Administration](#) [User Administration](#) [System Administration](#)

Company | **Work** | Teams | Documents | Portal Information

UWL Launch SAP Transaction [History](#) [Back](#) [Forward](#)

Form Edit Goto System Help Function

Change Notification of Absence

Application Data

Number Status

Personal Data

Name

Department

Personnel no. Cost center

Absence Data

	By	to	Hours	Leave Type
Leave 1	16.10.2003	20.10.2003		<input type="text" value="Vacation"/>
Leave 2				<input type="text" value="Vacation"/>
Leave 3				<input type="text" value="Vacation"/>

Reason

Contact at

Entry and Approval

Date Date

Issuer Approver

- Parameterized launching for
- BSP
- iView
- URL
- Web Dynpro
- SAP GUI (for HTML)

UWL 7.0 – Standard Task Views & Filters

Tasks (11 / 27) Alerts Notifications Tracking

Show **New and in progress tasks (11 / 27)** <select a subview> All

Subject	From	Sent
Tasks forwarded to someone	!	
JDK1_3M_4 problem for NW 2004s/7.0 (Downport)	Cartman, Eric	Feb 14, 2005
Tasks for resubmission	Kruse, Matthias	Today
Completed Tasks	Kruse, Matthias	Feb 15, 2005
Demo NW Team PA	Kruse, Matthias	Feb 15, 2005
Initiate Employee Performance Management	Kruse, Matthias	Feb 14, 2005
Notification of absence number 0000002052 rejected. What do you want to do?	Administrator, Admin	Yesterday
Notification of absence number 0000002092 rejected. What do you want to do?	Workflow System	Yesterday

Tasks (11 / 27) Alerts Notifications Tracking

Show **New and in progress tasks (11 / 27)** <select a subview> **Overdue** [Create Task](#) [Show Filters](#) [Hide Preview](#)

Subject	From	Sent	Priority	Due	Status
JDK1_3M_4 problem for NW 2004s/7.0 (Downport)	Cartman, Eric	Feb 14, 2005	High	Yesterday	In Progress
Edit maintenance notification 10001257 / M2	Workflow System	Aug 26, 2004	Normal	Aug 26, 2004	In Progress
Edit maintenance notification 10001258 / M1	Workflow System	Aug 26, 2004	Normal	Aug 26, 2004	In Progress
Edit maintenance notification 10001259 / M1	Workflow System	Aug 26, 2004	Normal	Aug 26, 2004	In Progress
Edit maintenance notification 10001260 / M1	Workflow System	Aug 26, 2004	Normal	Aug 26, 2004	In Progress

Launch Work Transaction

Subject	From	Sent	Priority	Due	Status
Notification of absence from 16.10.2003: Revise	Workflow System	Feb 15, 2005	Normal	1	New
Notification of absence from : Revise	Workflow System	Feb 15, 2005	Normal	1	New
Employee CARTMAN : Approve notification of absence	Workflow System	Feb 11, 2005	Normal	1	In Progress

Application Data
Number: 2052 Status: Rejected

Personal Data
Name: CARTMAN
Department:
Personnel no.:
Cost center: 4567

Absence Data

	By	to	Hours	Leave Type
Leave 1	16.10.2003	20.10.2003		Vacation
Leave 2				Vacation
Leave 3				Vacation

Reason: Need some time off
Contact at:

Entry and Approval
Date: 16.10.2003 Date:
Issuer: CARTMAN Approver:

- Parameterized launching for**
- BSP
 - iView
 - URL
 - Web Dynpro (Java or ABAP)
 - SAP GUI (for HTML)

Launch Customization: ABAP Web Dynpro

The screenshot shows the SAP Web Dynpro interface. At the top, there are tabs for 'Tasks (11 / 12)', 'Alerts', 'Notifications', and 'Tracking'. Below the tabs, there are dropdown menus for 'Show: New and In Progress Tasks (11 / 12)' and 'my ABAP web dynpro (2)'. A table lists tasks, with the first task 'select flight' highlighted and circled with a '1'. Below the table, a detailed view of the 'select flight' task is shown. It includes a 'Return' button, the title 'select flight', and details: 'Sent: Today by Green, Gretchen', 'Status: New', 'Escalated:', 'Priority: Normal'. A message says 'Select your flight carefully.' and there is a link 'WFT synchronous: 12' circled with a '2'. Below this, there are buttons for 'Follow-up', 'Forward', 'Claim', and 'launchWebDynProABAP'. An orange callout box with an arrow points to the 'launchWebDynProABAP' button, containing the text 'Launches an ABAP Web Dynpro'. To the right, there is a section 'You can also' with links: 'View Preview', 'View Detail in SAP Gui', 'Manage Attachments', and 'Create Ad Hoc Request'. Below this, there is a form with an 'Airline:' label and a text input field. At the bottom right, there is a table with columns 'Airline', 'Flight Number', 'Depart.city', and 'Arrival city'. The table contains five rows of flight data. Below the table, there are navigation controls and a 'Go Complete' button circled with a '3'.

Airline	Flight Number	Depart.city	Arrival city
AA	0017	NEW YORK	SAN FRANCISCO
AA	0064	SAN FRANCISCO	NEW YORK
AZ	0555	ROME	FRANKFURT
AZ	0788	ROME	TOKYO
AZ	0789	TOKYO	ROME

XML Example

```
<Action name="launchWebDynProABAP"  
handler="SAPWebDynproABAPLauncher">
```

Means to improve SBWP performance have been provided

Scenarios that benefit most are

- **Many agents work on large pools of work items (e.g. call center scenario)**
 - **Large, slow work lists**
 - **Necessity to often refresh the work list (due to work item „overlaps“)**
- **Extensive usage of dynamic columns**
 - **particularly with large work lists**
- **Grouping of work items in large work lists using container data**
 - **Grouped according to content**
 - **Grouped according to content type**
 - **Grouped according to sort key**

To mitigate SBWP performance issues, BAdIs have been defined

Replacement for SAP MAPI is provided

- **Client-based MS Outlook integration of SAP Business Workflow**
- **Support ends 10/2005**

New solutions

- **Server-based (zero footprint)**
- **Mail client independent**
- **Covers 90% of workflow-related MAPI customer scenarios**

Different replacements for different SAP releases

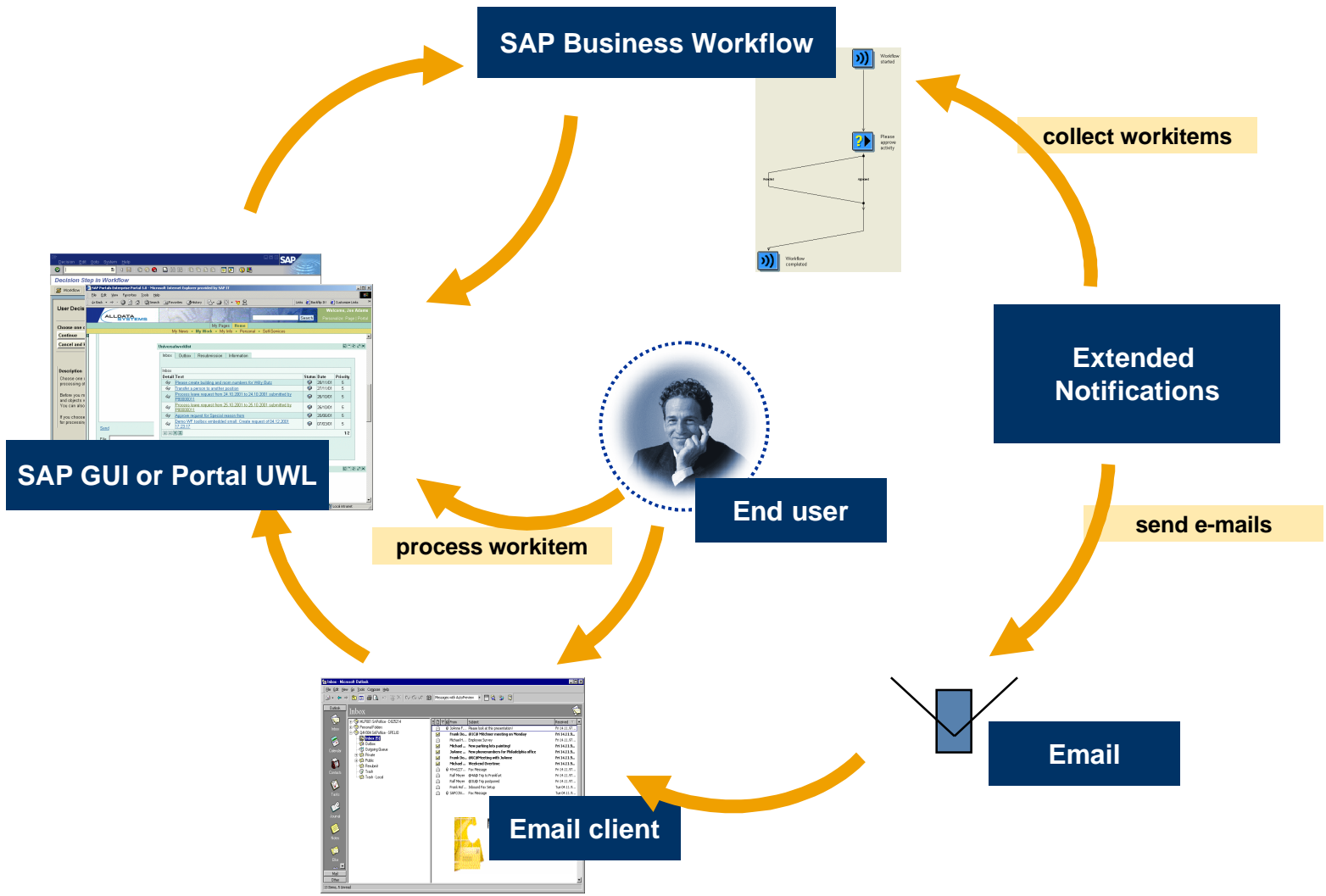
Report RSWUWFML2

- **Polling report sending email notifications**
- **Releases 4.6C – 620**

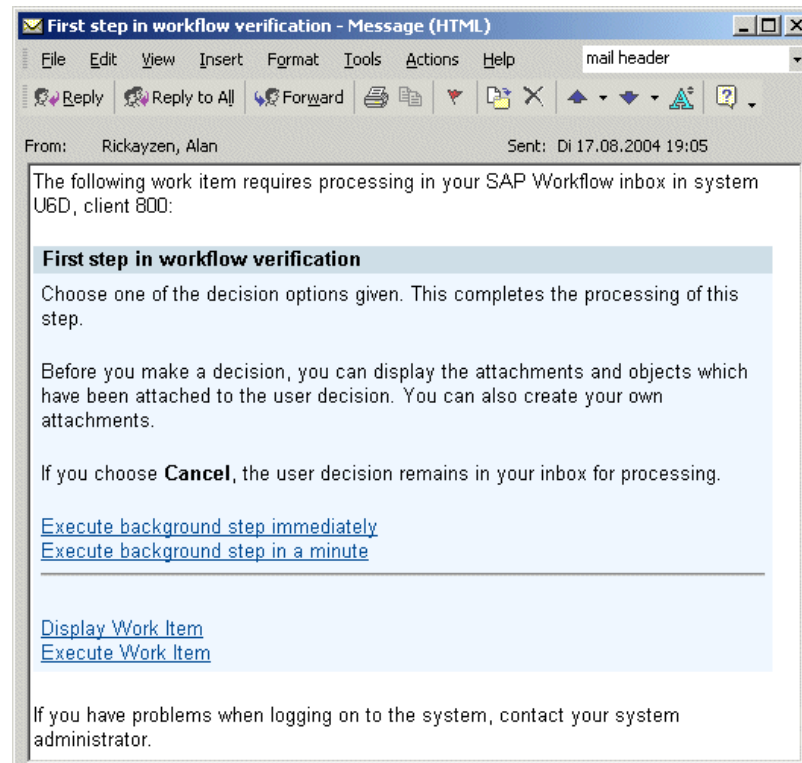
Extended Notifications

- **Server notification framework**
- **Releases > 640**

Extended Notifications: Overview



Extended Notifications: Basic Features



Notify users about work items that need to be processed

Send workitems to groupware (e.g. MS Outlook or Lotus Notes) carrying direct callbacks to backend transactions

- Execution of web-based work items (Web Dynpro, People centric UI, ...)
- Execution of work items via SAP GUI for Windows

New workflow features since R/3 4.6c

Design Time Changes and SAP_WAPI at runtime

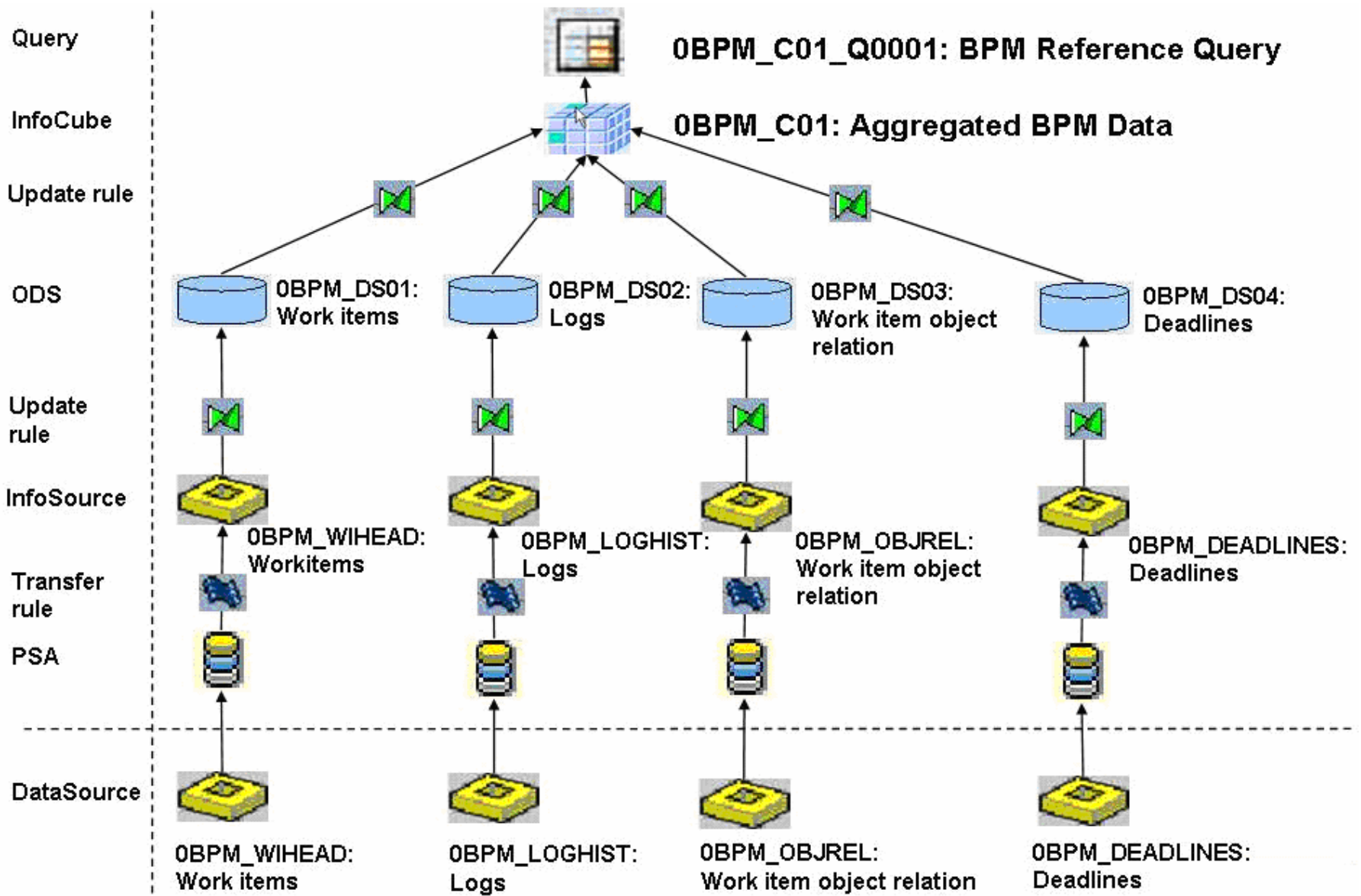
Delivering work items

Reporting

Note: We only provide overview of changes in this presentation. There is an appendix we will not cover that has more details and there is a delta class, DBITWF.

- **The Workflow Information System (WIS) has become obsolete with WAS release 610**
- **Customers do their reporting using BI**
- **Workflow projects with BI requirements**
 - ◆ **could not use standard workflow-related data extraction**
 - ◆ **had to implement their own extraction logic for workflow data**
- **This gap has now been closed**
 - ◆ **Technical** work item data is available in BI as infocube
 - **How many workflows of type 'xy' were executed?**
 - **What is the average processing time for a given workflow (step)?**
 - **How many steps of type 'xy' were executed by org unit 'abc'?**
- **Workflow and Business Objects worlds are linked by work item to object relation information available**

BW Integration for SAP Business Workflow (2)



- **Practical Workflow for SAP** from SAP Press
- **Classroom Education:**
 - DBITWDF – 2 day workflow delta class,
 - BC401 – 3 days ABAP objects class
 - <http://www.sap.com/usa/education>
- **SDN workflow forum and important blogs**
 - <http://sdn.sap.com> Forums -> SAP NetWeaver -> BPM and workflow
 - Why use ABAP OO with workflow?
<https://weblogs.sdn.sap.com/pub/wlg/3858>
 - Getting started with ABAP OO for workflow
<https://weblogs.sdn.sap.com/pub/wlg/3907>
- **SDN workflow wiki FAQ maintained by Mike Pokraka (and you)**
 - <http://sdn.sap.com> Wiki -> Wiki (home) -> BPM and Workflow -> SAP Business Workflow FAQ
<https://wiki.sdn.sap.com/wiki/display/HOME/SAP+Business+Workflow+FAQ>

Questions?

Q&A



Feedback

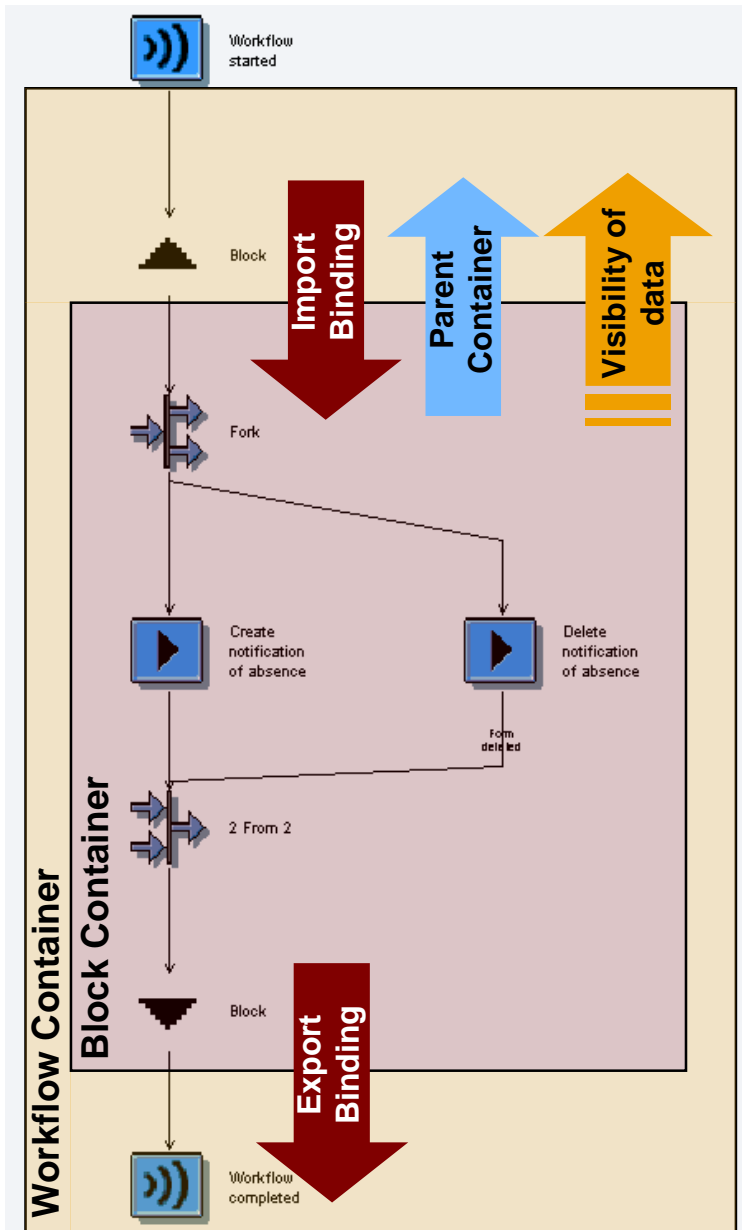
Please complete your session evaluation.

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

Thank You !

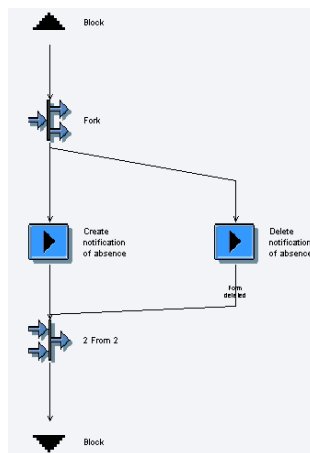


Appendix



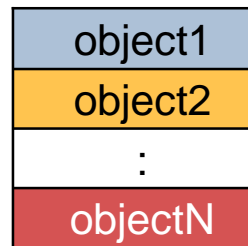
- Every block has a container
- Nested blocks form nested data contexts
 - ◆ Parent relationships between containers
 - ◆ Workflow is the top level block
- Parent context is visible within sub context
- Bindings can be defined from
 - ◆ the parent context to the sub context (Import binding at block instantiation)
 - ◆ the sub context to the parent context (Export binding after block completion)

- **Block instantiation can be dynamically controlled by multiline container elements (ParForEach)**
 - ➔ **Similar to the „ParForEach“ dynamic parallel processing for activities!**
- **For each entry in the multiline element, one block instance is created**
 - ◆ **Corresponding line data transported to the block context via binding**
- **Process continues after the ParForEach block if either**
 - ◆ **all instantiated parallel blocks are completed or**
 - ◆ **a special join condition evaluates to TRUE after completion of a block**



Block definition

+



Multiline container element

=

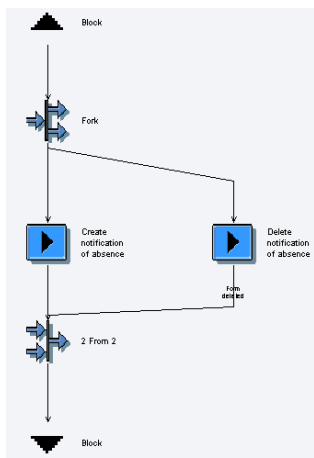


N parallel block instances

■ Block execution can be dynamically controlled by multiline container elements (ForEach)

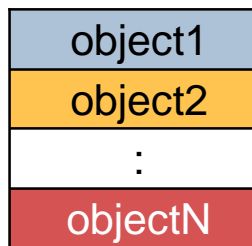
For each entry in the multiline element, the sequence of steps defined within the block is executed once

- Only one block item is created!
- ForEach loop semantics
- Process continues after the ForEach block if either
 - all table entries are processed or
 - a special condition evaluates to TRUE after completion of a loop sequence



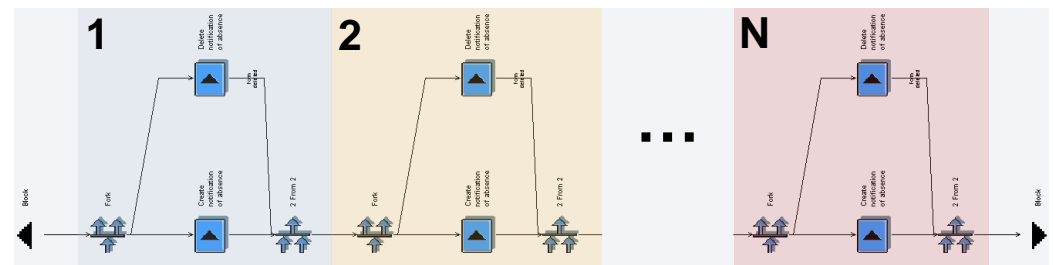
Block definition

+

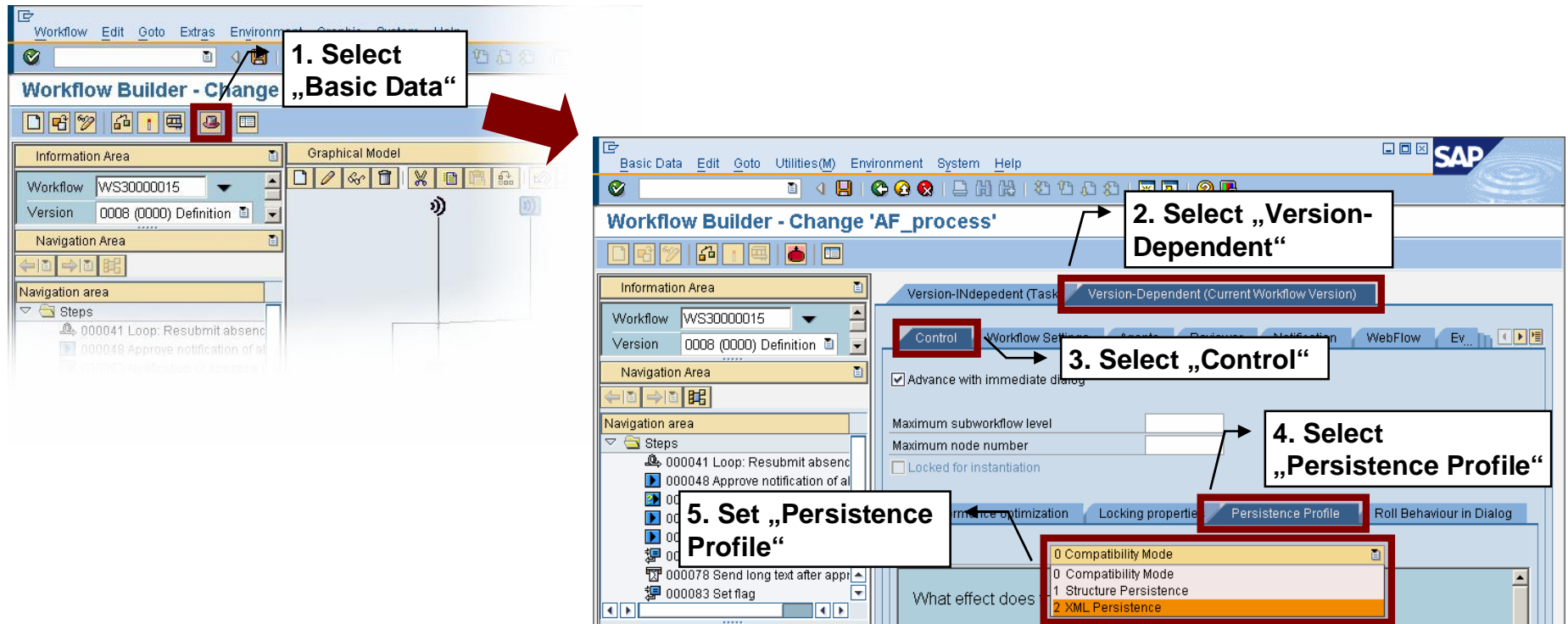


Multiline container element

=



N block sequence cycles



Each Workflow instance has a persistence profile

- It is set in the basic data of the workflow definition
- The default value is
 - ◆ „XML Persistence“ for new workflow definitions
 - ◆ „Compatibility Mode“ for legacy workflow definitions

Every binding instruction can be configured

The available configuration options depend on the operator

Option	Available for
Handle errors as warnings	All binding operators
Move-corresponding	Assign, Append
Overwrite empty target elements only	Assign, Merge, XSLT
No action if source is initial	Assign, XSLT
No action if source is empty	Assign, XSLT

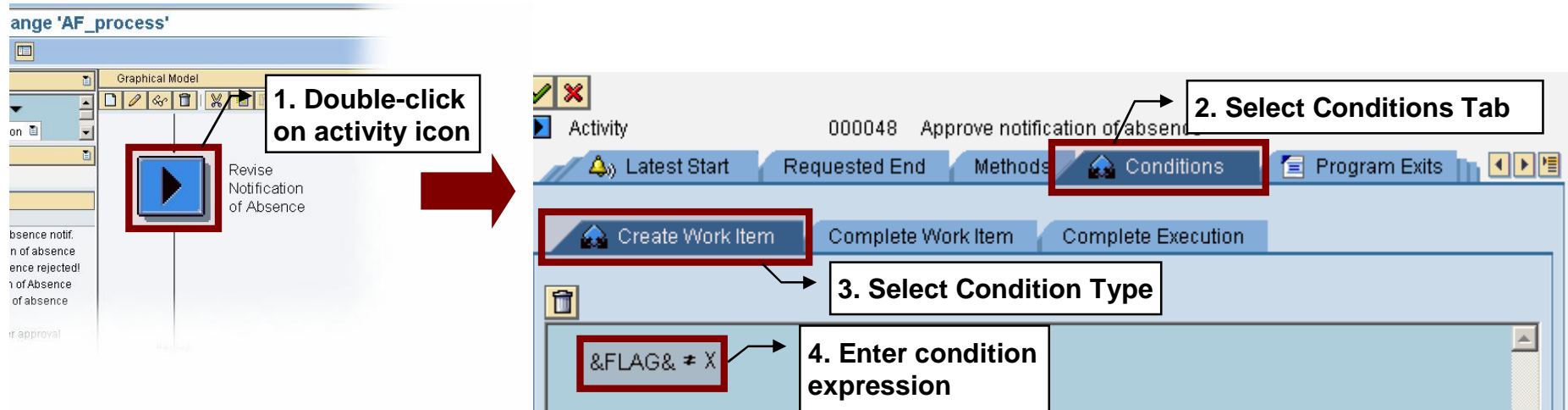
■ **The set of binding operators is extensible**

■ **You can program your own binding operators**

- **Use `IF_SWF_IFS_BINDING_TRANSFORM` for expression to expression operators**

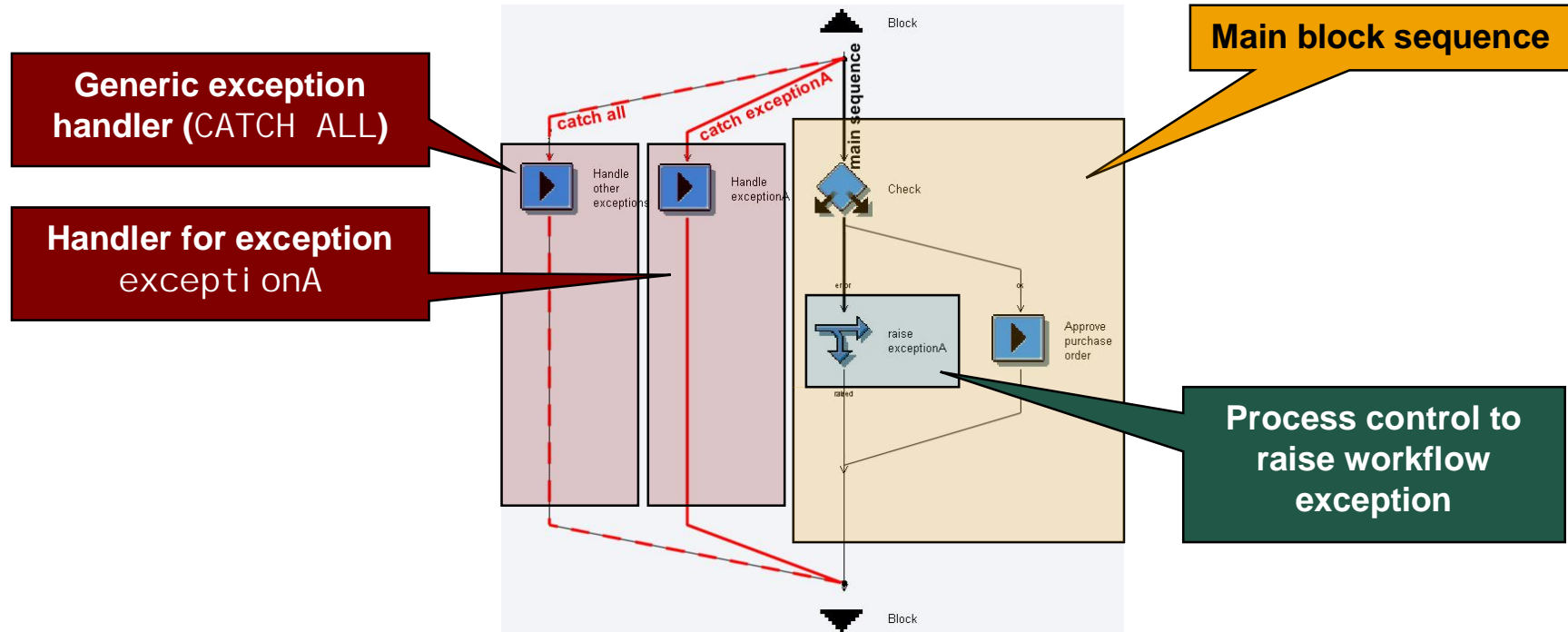
- **Use `IF_SWF_IFS_BINDING_TRANSFORM_CONTAINER` for container to container operators**

Condition Type	Behavior
Create Work Item	<ul style="list-style-type: none"> ■ This condition is evaluated on the surrounding container ■ Work item remains in WAITING state until condition evaluates to TRUE ➔ It can not be executed until condition is fulfilled
Complete Work Item	<ul style="list-style-type: none"> ■ This condition is evaluated on the surrounding container ■ Work item is immediately set to COMPLETED when condition evaluates to TRUE ■ Result branch „Complete condition true“ is taken
Complete Execution	<ul style="list-style-type: none"> ■ This condition is evaluated on the step container itself ■ Condition is evaluated after work item execution ■ If the condition evaluates to TRUE, the work item is set to COMPLETED ■ If the condition evaluates to FALSE, the work item remains IN PROCESS or is even set back to READY (➔ see slide notes for details)

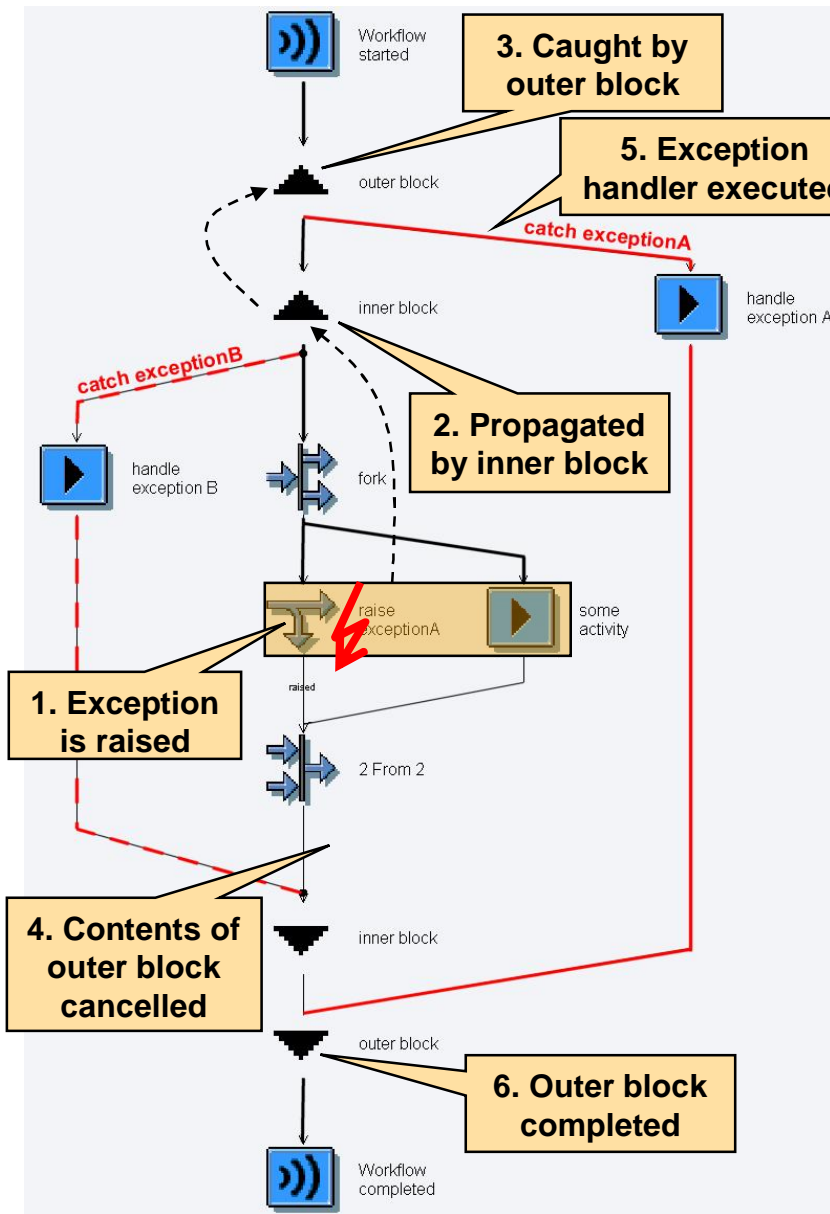


■ Step conditions are

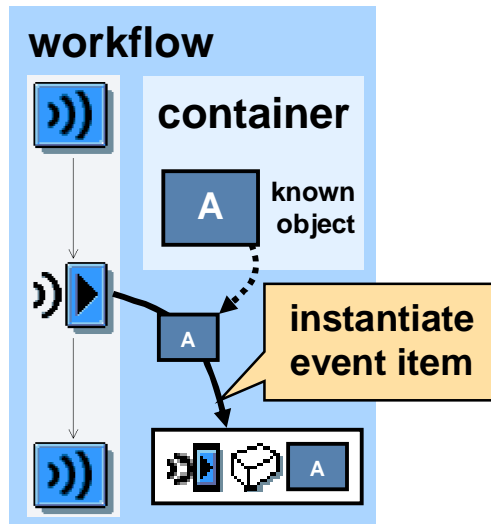
- formulated on the surrounding container (data context) or the step container
- evaluated either by the workflow engine, a periodic background job (SWWCOND) or event driven
- available for the following step types (with work item representation)
 - ◆ Activity
 - ◆ User Decision
 - ◆ Wait Step
 - ◆ Web Activity
- The state of the corresponding work item can be influenced by
 - the fact that a condition is defined
 - the result of condition evaluation



- Blocks can handle exceptions (TRY-CATCH mechanism)
- Exceptions can be raised by a Process Control step (→ workflow exceptions)
- For every workflow exception defined within the process, exception handlers can be defined on every block
- A generic (CATCH ALL) exception handler is also available



- Exceptions are propagated
 - From their origin up the block hierarchy
- Blocks either catch exceptions...
 - ◆ in case appropriate handler is defined
- ... or propagate them further
 - ◆ in case no appropriate handler defined
 - ◆ or
 - ◆ block already caught exception
- If exception is caught
 - ◆ everything inside block is cancelled
 - ◆ exception handler is executed
 - ◆ block is completed normally afterwards
- If exception is not caught, process is set to an ERROR state



Event couplings are created using

- event name (design time information)
- event object instance (run time information)

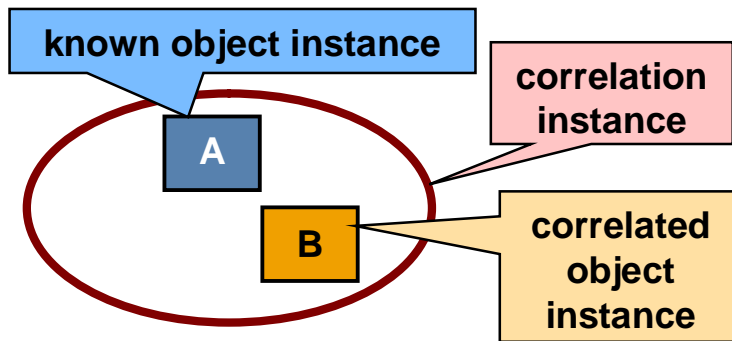
The object instance has to be known to the process

Problem: How to wait for events of yet unknown objects?

Knowing an object means knowing its key (→ unique identifier)

But: objects can also be identified using semantic information

Goal: Wait for events of objects semantically coupled to objects known



Correlations are objects which group other objects by semantic criteria (correlation key)

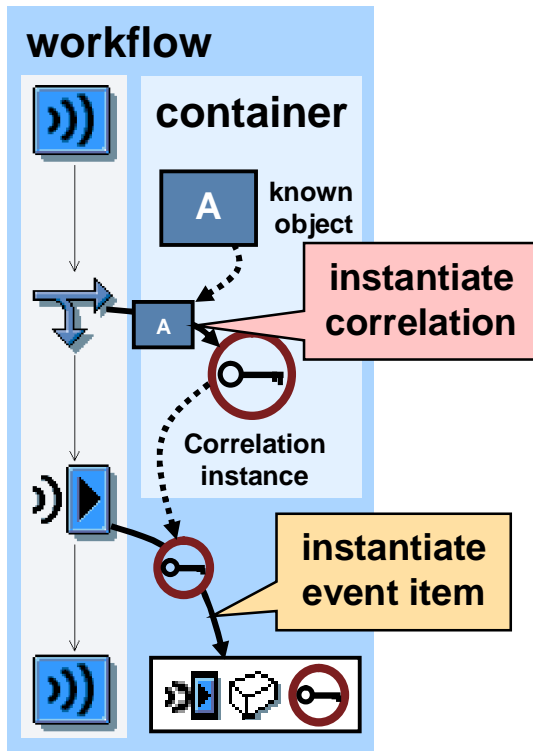
→ “objects having to do with each other”

Event couplings can also be created using

- **event name (design time information)**
- **event object type (design time information)**
- **correlation instance (run time information)**

The object instance does not have to be known to the process

- **but at least one correlated object instance has to**
- Example: know order to identify correlated invoice**



Correlations are instantiated using one participating object instance

- (semantic) correlation key has to be determined

→ Example: order ID

Wait step (event item) is instantiated using

- correlation instance (semantic key)

- object type

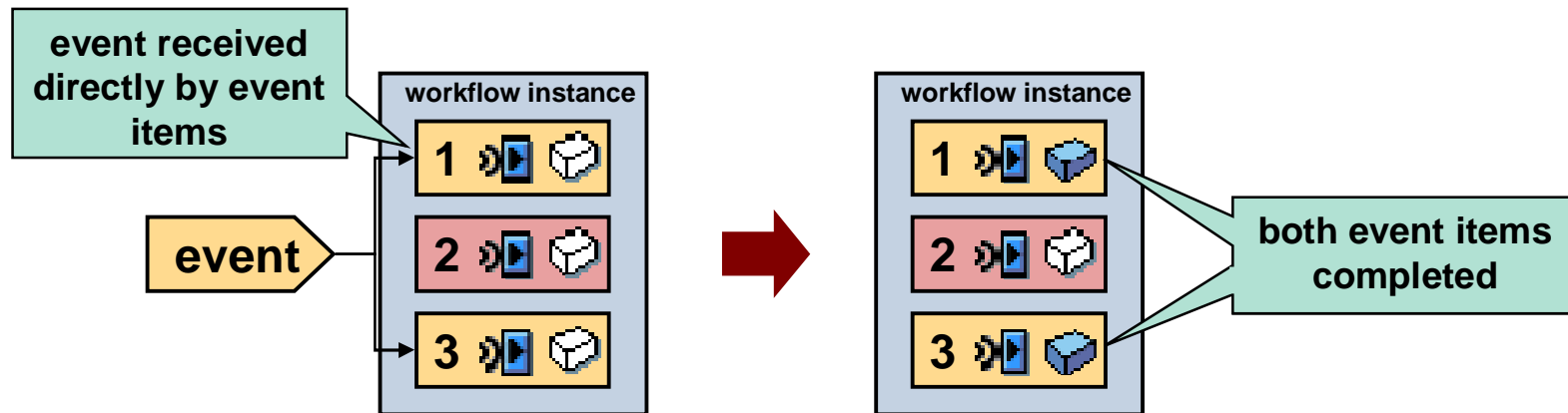
- event

→ Example: wait for creation of invoice referring to known order

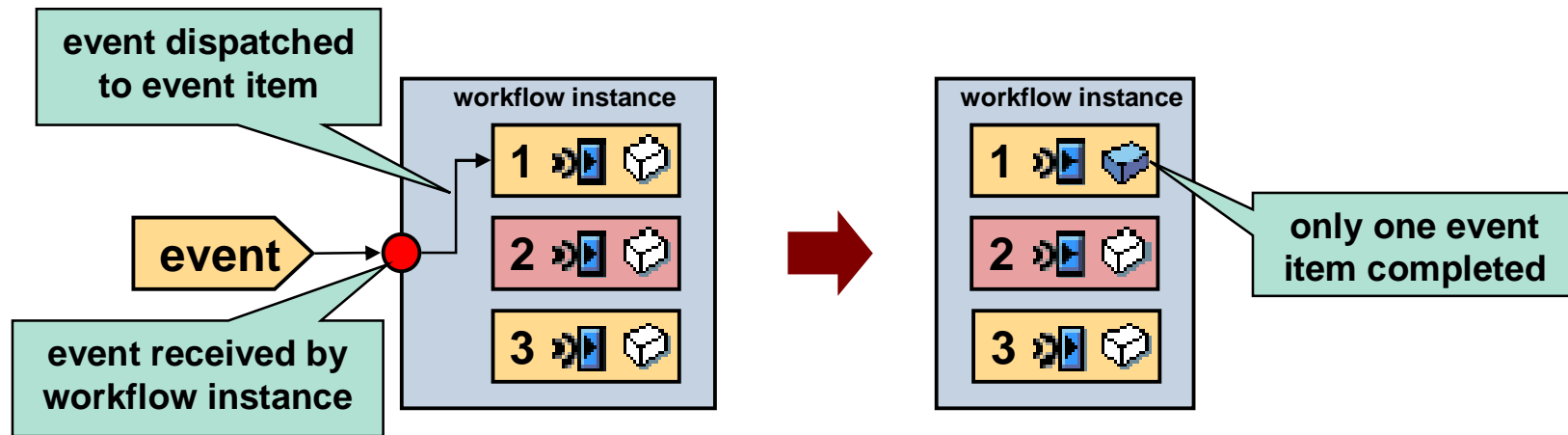


Event of unknown, yet correlated object is delivered to event item via the correlation instance

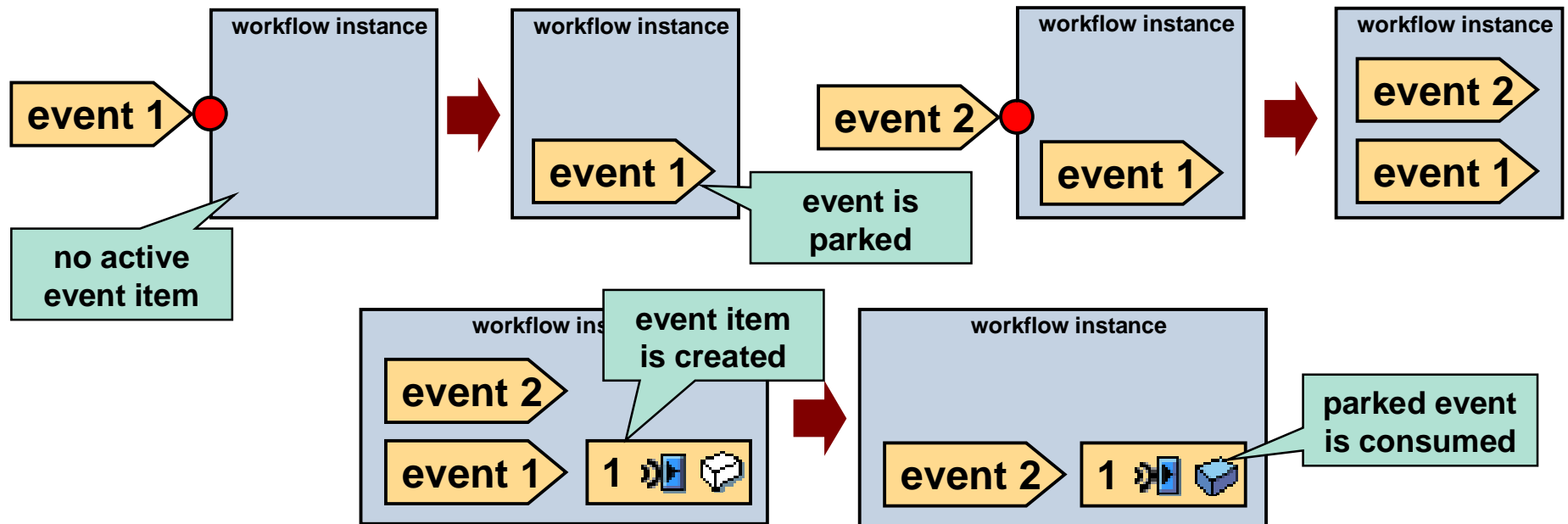
→ see Appendix for further details



- When a wait step is instantiated,
 - ◆ an event work item is created acting as event listener
 - ◆ an event instance coupling (subscription) is written for this event work item
- Every event triggered completes all active event work items subscribed to it
- The process (workflow instance) plays no role here



- The workflow instance can act as an event dispatcher
 - ◆ The workflow instance acts as the event receiver
 - ◆ It dispatches those events to its dependent event items
- Event work items receive events via the workflow
- Only one event work item is completed per event
 - ◆ If more than one event work items in question exist, the oldest is taken
 - ◆ This mechanism implements an event item queue within the process



- If no active event items exist when workflow receives event, the event is *parked*
 - ◆ Event waits for the next corresponding event item to be created within the process
 - ◆ This mechanism implements a *queue of parked events*
- New event items consume corresponding parked events immediately after their creation

Conventional event couplings:

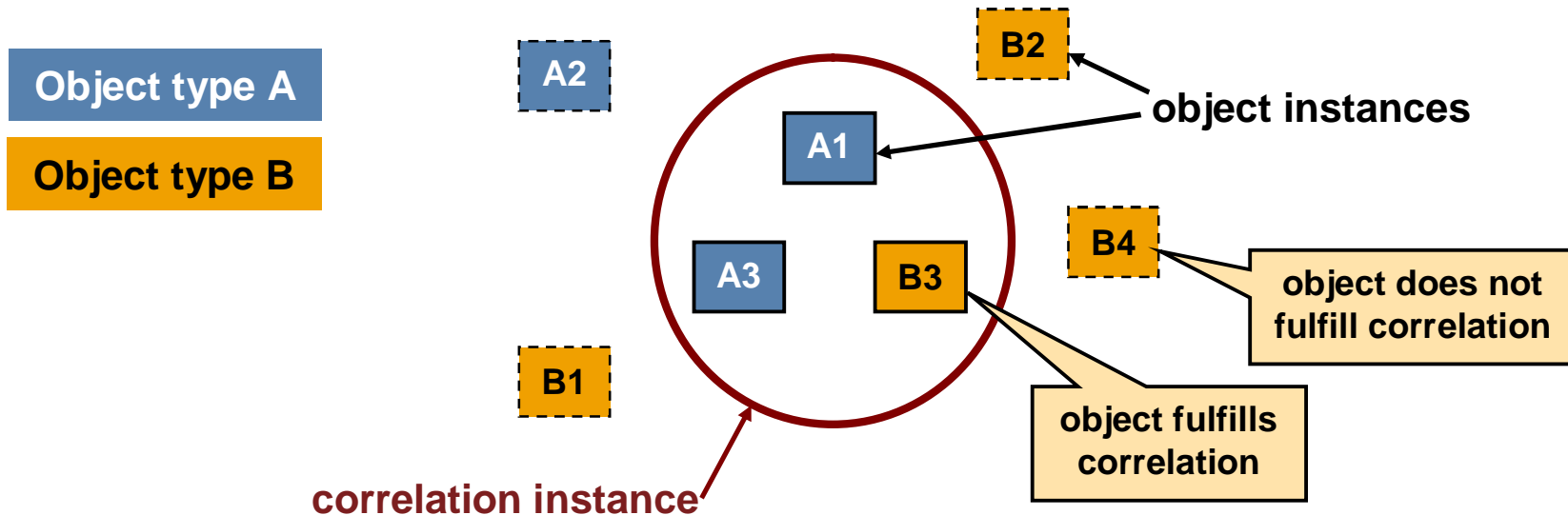
- **Wait steps are instantiated for existing and known object instances**
- **The object key is used to connect event and wait step (coupling)**

Requirement:

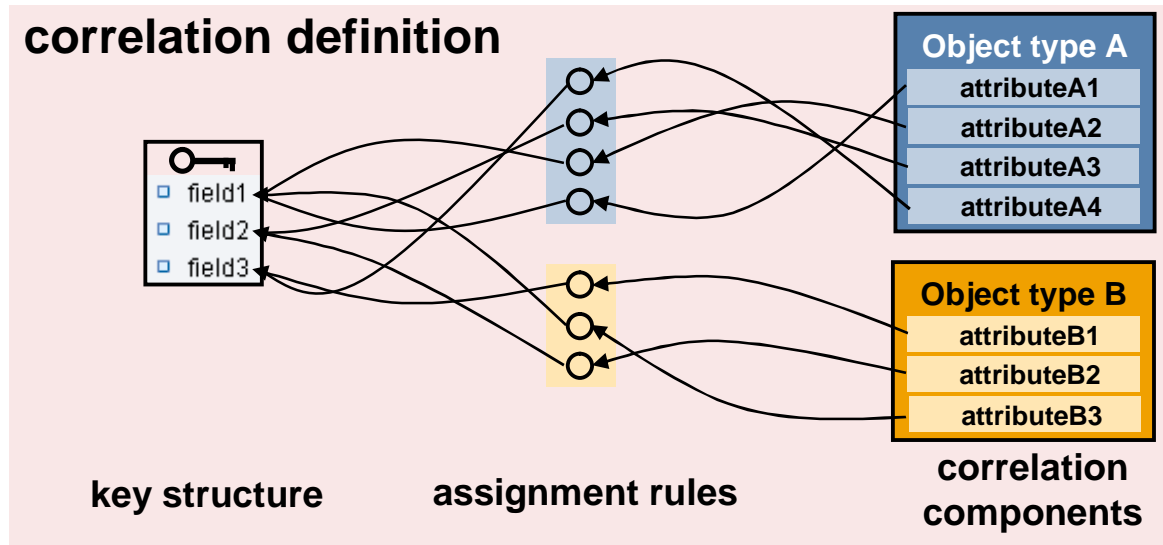
- **Wait for events of unknown objects**
 - ◆ **Object key is not available at wait step creation time**
- **Object key cannot be used to link event to wait step**

Solution:

- **Use semantic coupling between object instances (correlation)**
 - ◆ **One object instance leads to several correlated object instances**
- **Define wait step not by object key but by **object type + correlation instance****

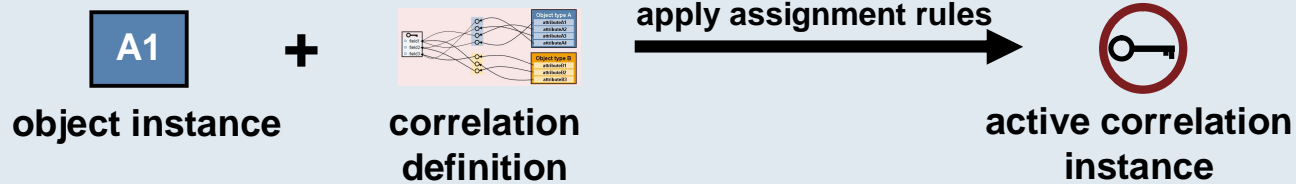


- Correlation instances group object instances by semantic criteria
- The objects are pulled together by a specially defined correlation key
 - ◆ Those object instances **fulfill the correlation instance**
 - ◆ They can have different object types
- A correlation instance can not be created until at least one fulfilling object instance exists
 - ◆ A correlation does not make sense without its objects!

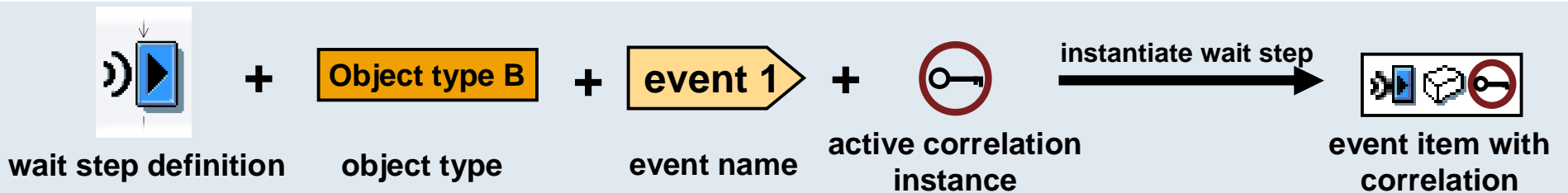


- Every correlation instance is based on a correlation definition
- The correlation definition consists of
 - ◆ A set of key fields (key definition)
 - ◆ A set of object types that shall be correlated with each other (correlation components)
 - ◆ A set of assignment rules between object type data and correlation key fields (one for each correlation component)
- For every correlation component, those assignment rules have to fully specify the correlation key!

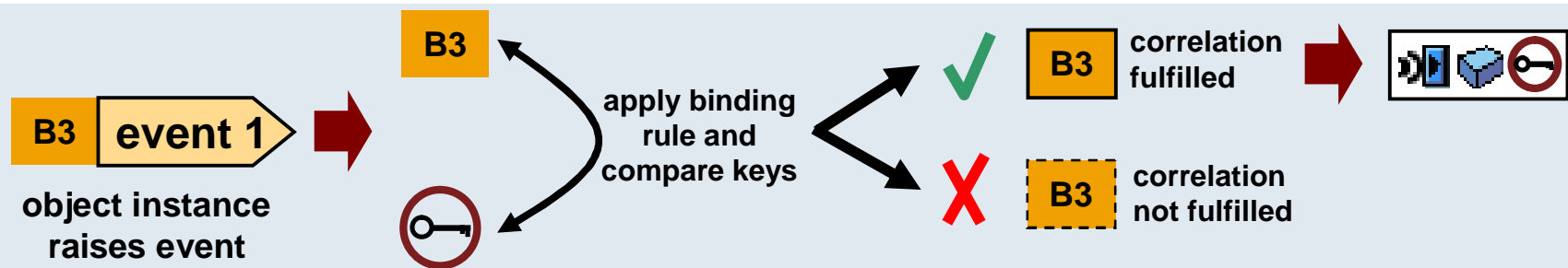
Correlations are activated with the help of an object instance
→ Object type has to belong to the correlation components



Work items listening to events can use correlations



If the event is raised by any correlation component, the correlation is checked



Report RSWUWFML

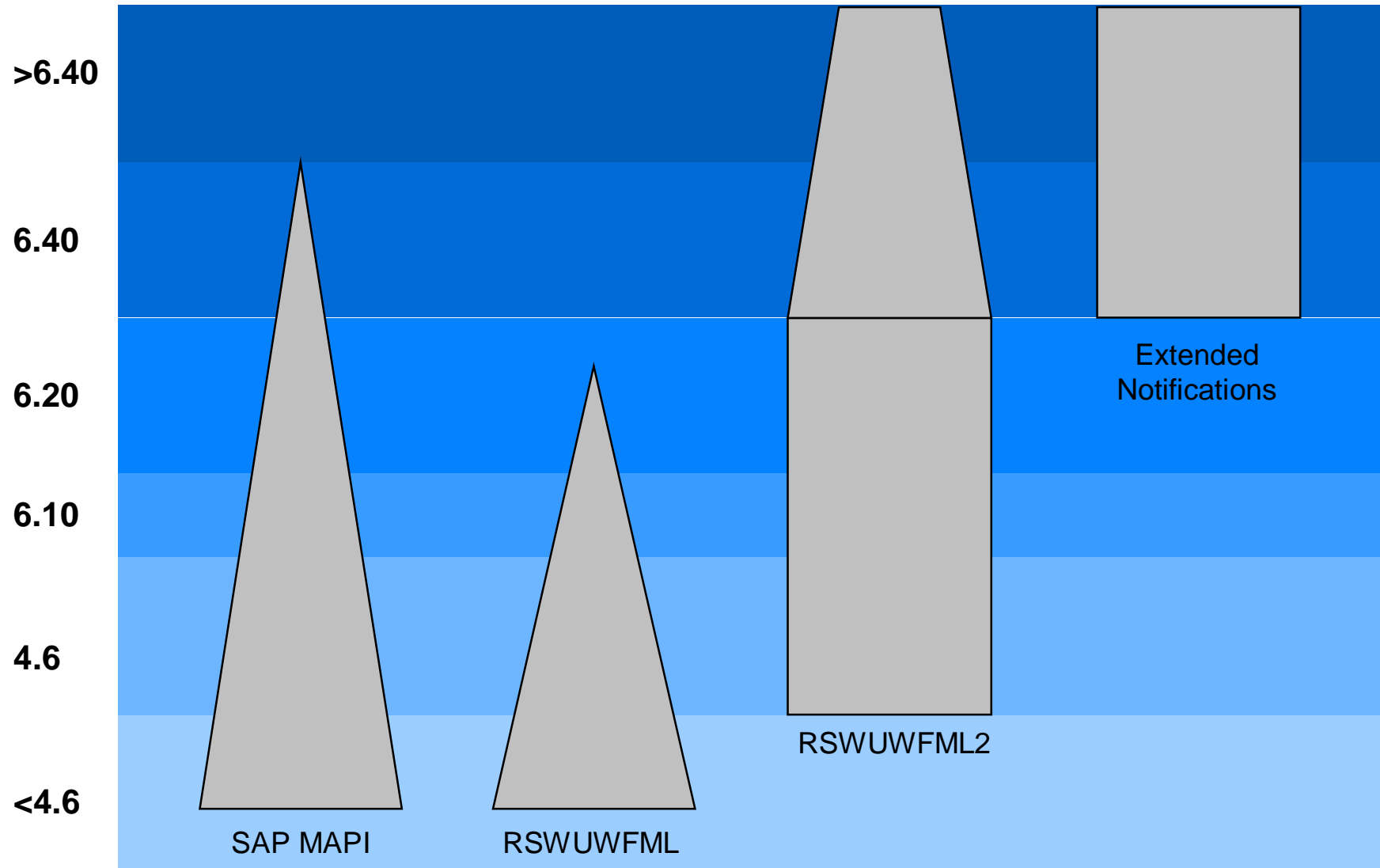
- **Polling report**
- **Uses SMTP/SAPconnect mail connection**
- **Uses SAP Office's auto-forward address**
- **Availability: Release 3.1I**

E-Mail Notifications for Business Workflow

- **Report RSWUWFML2**
- **Successor of RSWUWFML**
- **WinGui shortcuts instead of R3F attachments**
- **Availability: Release 4.6C**

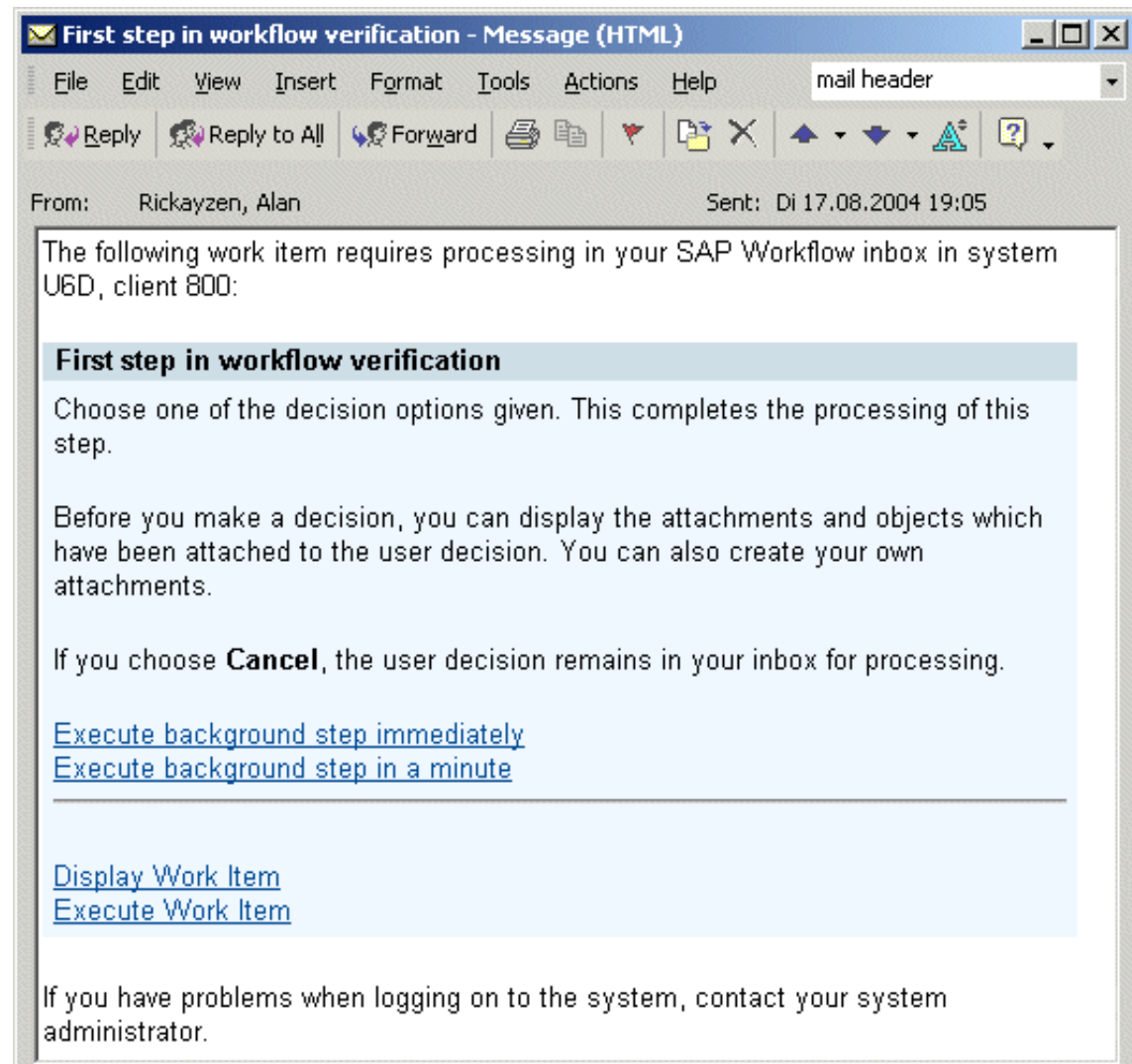
Extended Notifications for SAP Business Workflow

- **Server notification framework**
- **Availability: WAS 6.40**



Individual mails possible on a work item by work item basis

Or....



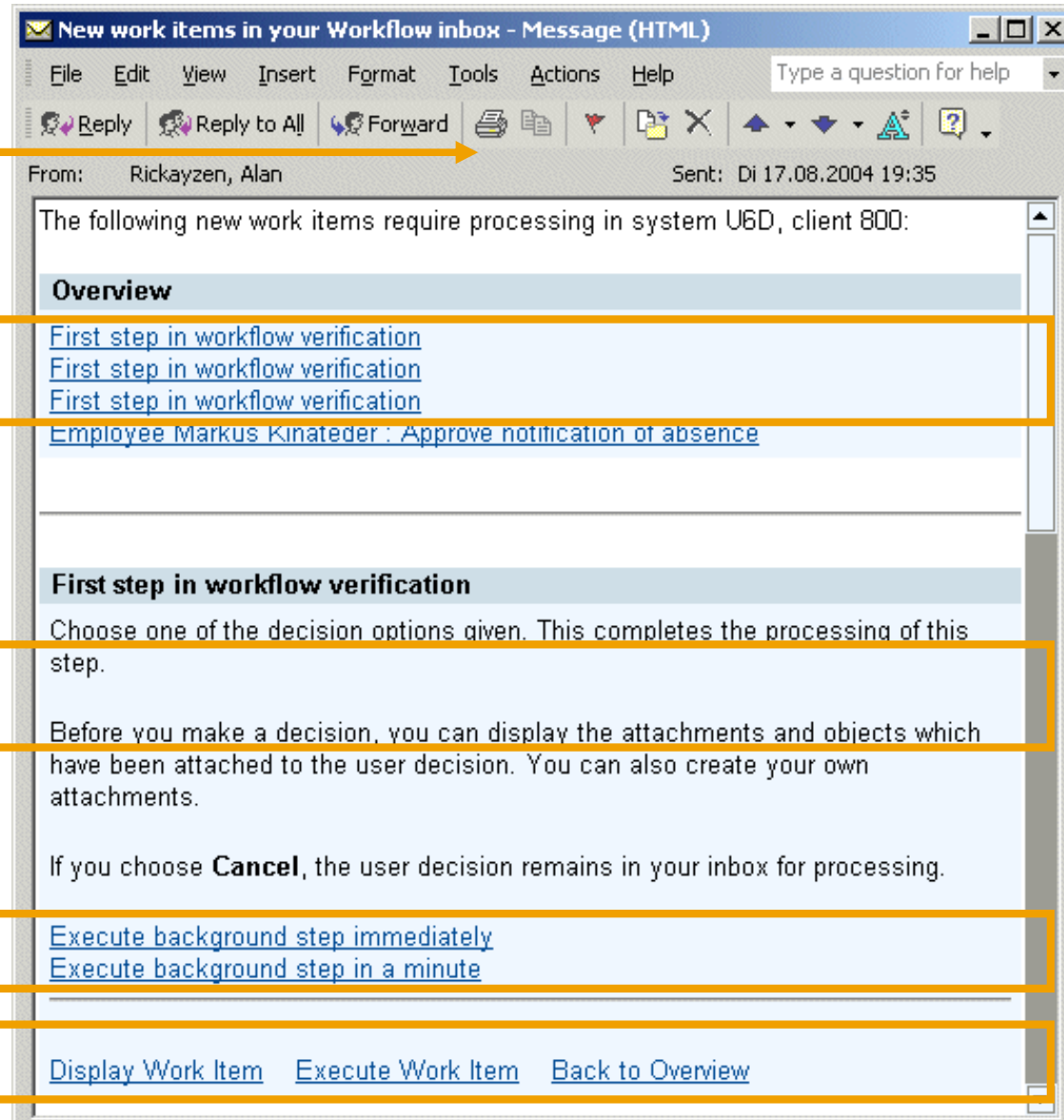
Shortcut Attachments possible

Index of work items

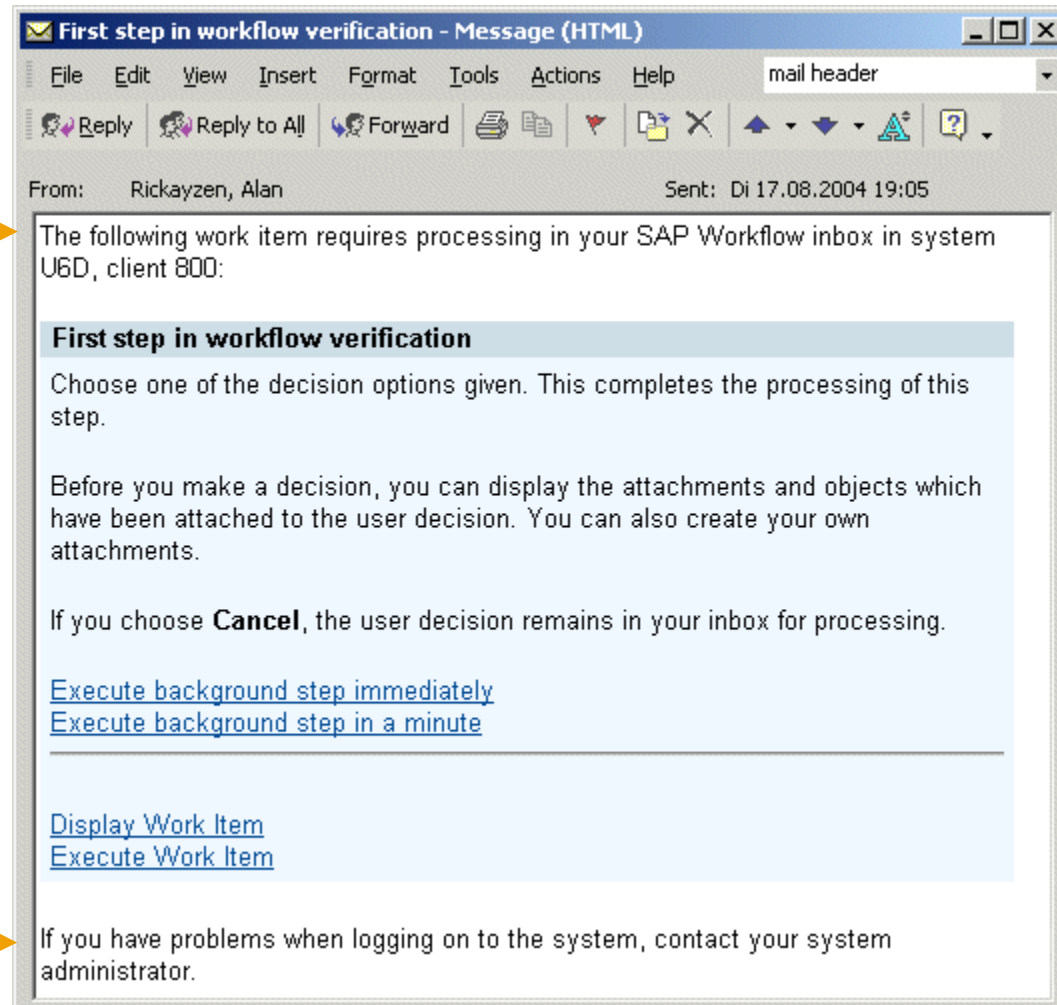
Workitem Description

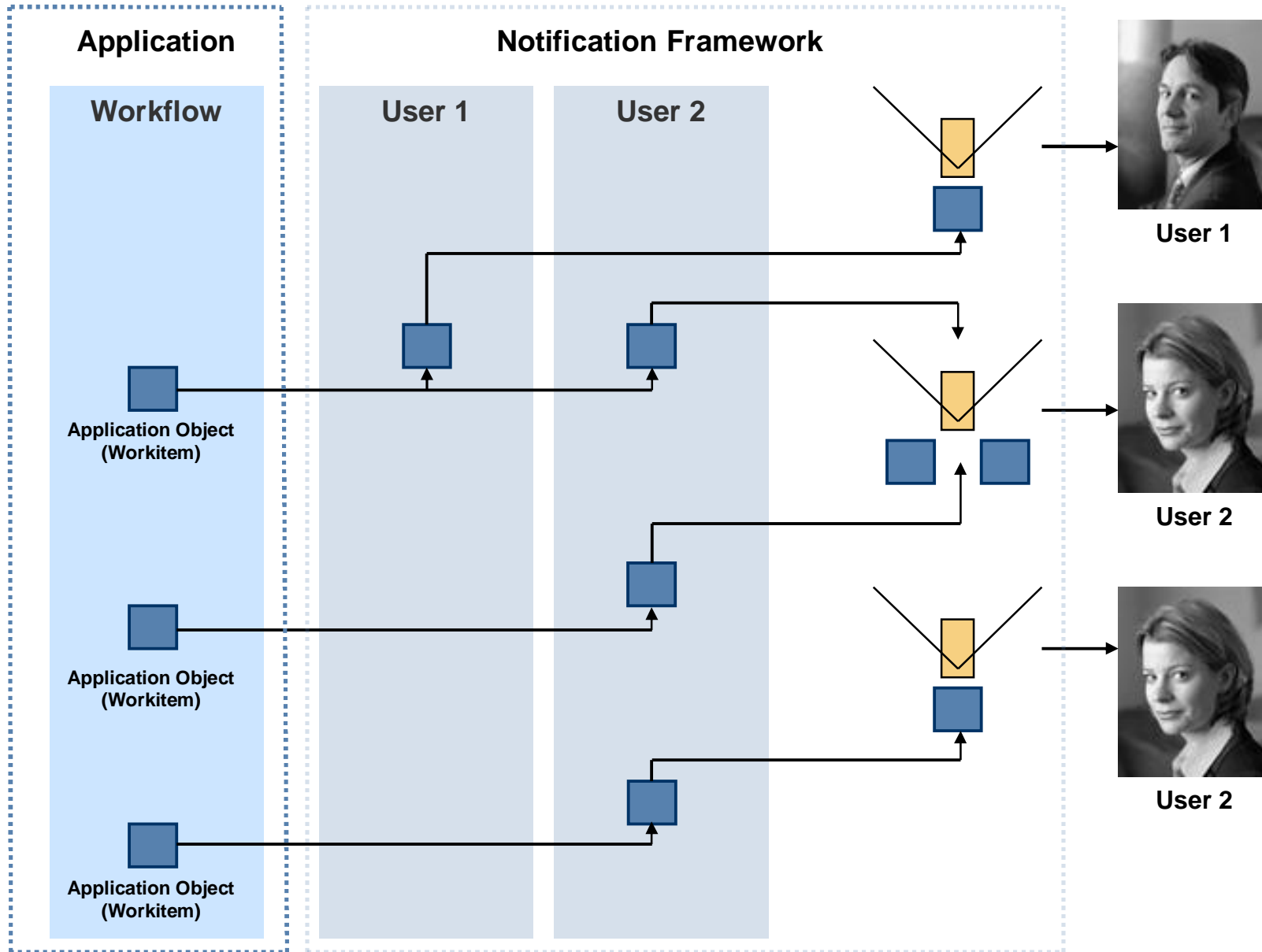
Generic Decision

Shortcut links



Redefinable Text





Feature	SAP MAPI	RSWUWFML	RSWUWML2	Extended Notifications
Deliver workitems to any e-mail client	-	X	X	X
SAP Shortcut attachments	-	-	X	X
Prefix AND suffix text (surrounding workitem description)	-	-	X	X
Enhanced address determination	-	-	X	X
Enhanced substitution handling	-	-	X	X
HTML e-mail (Outlook only)	-	-	-	X
Generic decisions	-	-	-	X
Links to new Web UIs (WebDynpro, etc)	-	-	-	X
Lists of workitems in one e-mail	-	-	-	X
Support workitem forwarding via e-mail	-	-	-	X

	Method	Level	Visibility	Method type	Description
Methods needed for persistence handling	<BI_PERSISTENT>				
	FIND_BY_LPOR	Static Method	Public		Find Using Local Persistent Object Reference
	LPOR	Instance Method	Public		Local Persistent Object Reference
	REFRESH	Instance Method	Public		Flag to Reload from Database
Standard methods used by runtime	<BI_OBJECT>				
	DEFAULT_ATTRIBUTE_VALUE	Instance Method	Public		Value of Default "Attribute" (as Data Reference)
	EXECUTE_DEFAULT_METHOD	Instance Method	Public		Execute Default Methods
	RELEASE	Instance Method	Public		Release for Garbage Collector to Delete
	CONSTRUCTOR:	Instance Method	Public	☞	Constructor
	CREATE	Static Method	Public		
	CREATE_VIA_API	Static Method	Public		
Application specific methods	APPROVE	Instance Method	Public		
	APPROVE_ASYNCHRON	Instance Method	Public		
	DELETE	Instance Method	Public		
	DISPLAY	Instance Method	Public		
	EXISTENCECHECK	Instance Method	Public		
	UPDATE	Instance Method	Public		

- ABAP OO classes are built with the class builder (SE24)
- ABAP OO classes used in workflows have to implement the interface IF_WORKFLOW
- IF_WORKFLOW includes two interfaces
 - ◆ BI_PERSISTENT (methods for object persistence)
 - ◆ BI_OBJECT (methods for object runtime)

The methods of BI_PERSISTENT implement persistence handling

Method	Explanation
FIND_BY_LPOR	<ul style="list-style-type: none"> ■ Factory method ■ Converts a persistent object reference (POR) to a ABAP OO object instance
LPOR	<ul style="list-style-type: none"> ■ Returns the persistent object reference (POR) of the ABAP OO object instance
REFRESH	<ul style="list-style-type: none"> ■ Tells the object to reload its state from the database (synchronization)

The methods of BI_OBJECT are needed by the workflow runtime

Option	Explanation
DEFAULT_ATTRIBUTE_VALUE	<ul style="list-style-type: none"> ■ Returns the value of the default attribute used for displaying the object instance
EXECUTE_DEFAULT_METHOD	<ul style="list-style-type: none"> ■ Tells the object instance to call its default method ■ Normally, this is a method which displays the object
RELEASE	<ul style="list-style-type: none"> ■ Tells the object instance that it is no longer needed ■ Possibility for controlled cleanup

Class Interface: CL_SWF_FORMABSENCE Implemented / Active

Properties Interfaces Friends **Attributes** Methods Events Types

Attribute	Level	Visibility	Key...	Re...	Typing	Associated Type
CREATOR	Instance Attribute	Public	<input type="checkbox"/>	<input type="checkbox"/>	Type	SIBFLPORB
DESCRIPTION	Instance Attribute	Public	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Type	SYST-TITLE
NUMBER	Instance Attribute	Public	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Type	SWXFORMABS-FORMNUMBER
MS_TYPEID	Static Attribute	Protected	<input type="checkbox"/>	<input type="checkbox"/>	Type	SIBFTYPEID
M_POR	Instance Attribute	Protected	<input type="checkbox"/>	<input type="checkbox"/>	Type	SIBFLPOR
M_SWXFORMABS	Static Attribute	Private	<input type="checkbox"/>	<input type="checkbox"/>	Type	SWXFORMABS
MST_INSTANCES	Static Attribute	Private	<input type="checkbox"/>	<input type="checkbox"/>	Type	T_INSTANCES

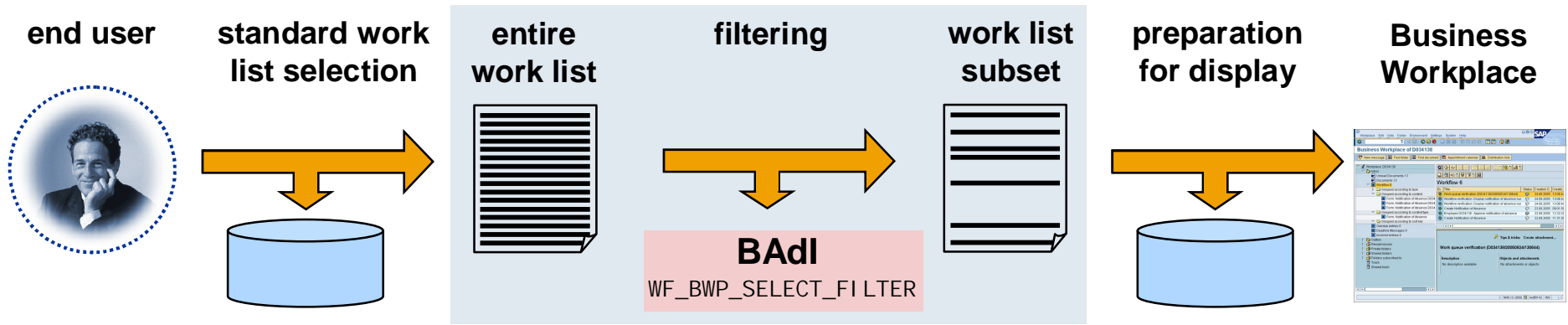
Workflow Container

Element	M	Description
▸ _Adhoc_Objects		Ad Hoc Objects of Workflow Instance
▸ _Attach_Objects		Attachments of Workflow Instance
▸ _Wf_Initiator		Initiator of Workflow Instance
▸ _Wf_Priority		Priority of Workflow Instance
▸ ▸ _Wi_Group_ID		Grouping Characteristic for Workflow
▸ ▸ _Workitem		Workflow Instance
▸ _Wf_Version		Definition Version of this Workflow I
▾ ▢ absenceform		absenceform
▸ ⚙ CREATOR		CREATOR
▾ ▢ DESCRIPTION		Object Description
▾ ▢ NUMBER		Proposal Number
▾ ▢ NAME		Name of applicant

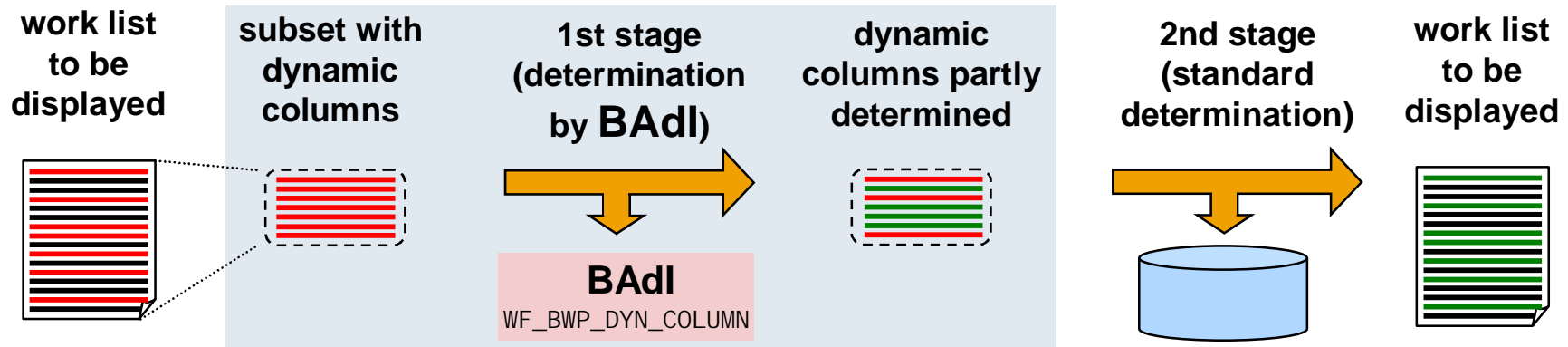
- All public attributes of an ABAP OO class can be accessed directly within workflow
- Method parameters can be of any datatype
 - ◆ Parameters of BOR object types had to be character based
 - ◆ XML Persistence profile has to be used for container

Event	Type	Visibility	Description
CREATED	Instance Event	Public	Notification of absence created
CHANGED	Instance Event	Public	Notification of absence changed
APPROVED	Instance Event	Public	Notification of absence approved
REJECTED	Instance Event	Public	Notification of absence rejected
DELETED	Instance Event	Public	Notification of absence deleted
DELETEERROROCCURED	Instance Event	Public	Notification of absence deleted (error occurred)

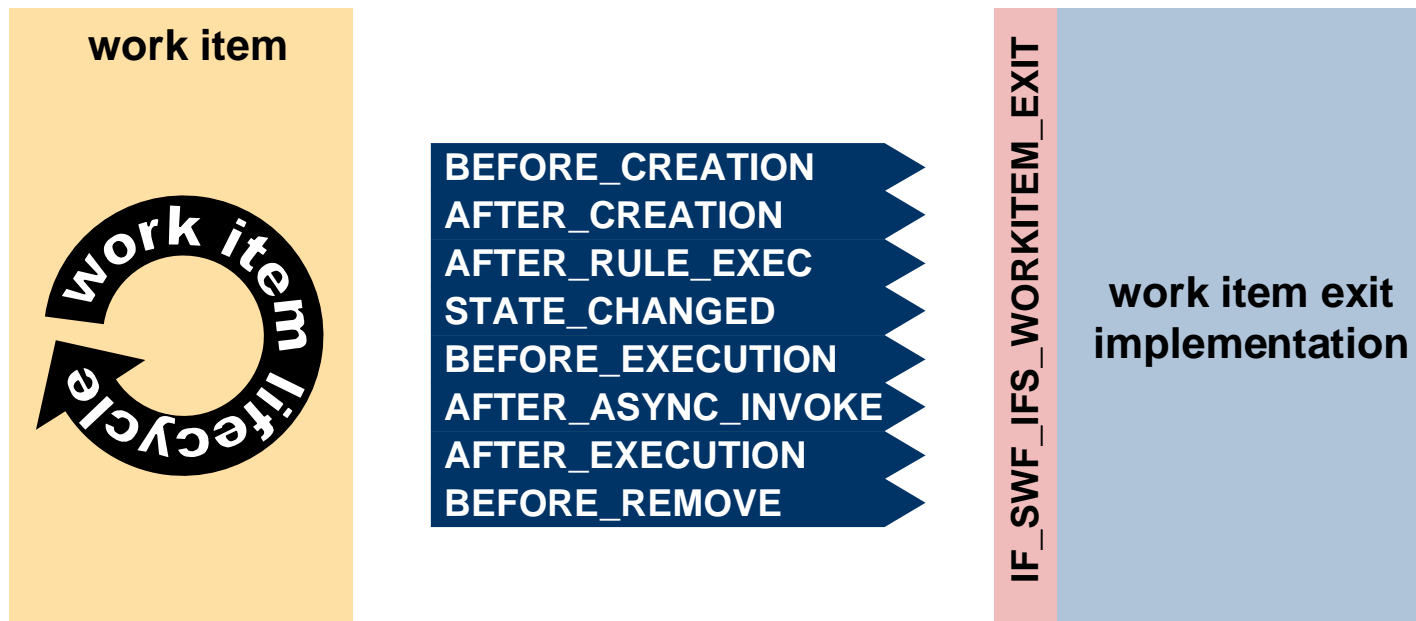
- All public events defined for the ABAP OO class can be used like BOR object events
 - ◆ as start events
 - ◆ as completion events
 - ◆ as workflow header events
 - ◆ in wait steps
 - ◆ in event trigger steps



- **BAdI `WF_BWP_SELECT_FILTER` enables the reduction of the number of work items displayed in the Business Workplace workflow inbox**
 - ◆ It imports the entire work list (work item header information)
 - ◆ It exports the (reduced) work list to be displayed as inbox
- **Custom filtering algorithms can be implemented**
 - Example implementation based on random numbers available
- **Performance critical operations applied to reduced work list**
 - Default attributes, dynamic columns, work item text in different languages, ...
- **Availability: WAS 620 SP 44, WAS 640 SP 9 (see note 765783)**



- **Dynamic columns can be customized on task level**
 - ◆ Display work item container information in Business Workplace inbox
- **Usage of dynamic columns is performance critical**
 - ◆ Read container for every single work item
 - ◆ Instantiate objects and read object attributes
 - ◆ Container and Business Object Repository are not mass-enabled
- **BAdI `WF_BWP_DYN_COLUMN` can help to improve performance**
- **The BAdI implementation facilitates**
 - ◆ custom buffering
 - ◆ mass selection from database
- **Availability: Future Support Packages for WAS 620 and WAS 640**
 - ◆ see note 848382 (not released yet)

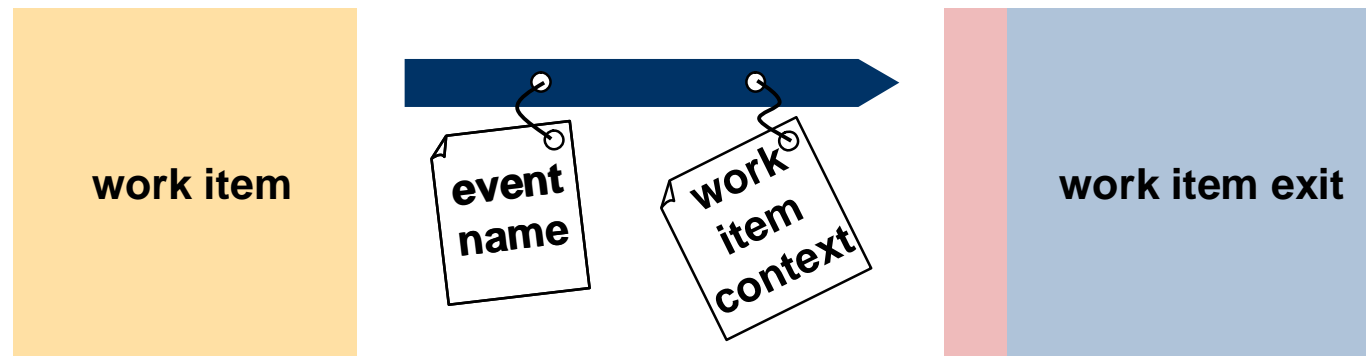


Work item exit classes have to implement the ABAP Objects interface IF_SWF_IFS_WORKITEM_EXIT

Exit classes can be attached to

- Steps (activity, decision, wait, web activity) → corresponding work items
- Blocks → block items
- Workflows → flow items

Exit classes are called at particular times in the work item lifecycle

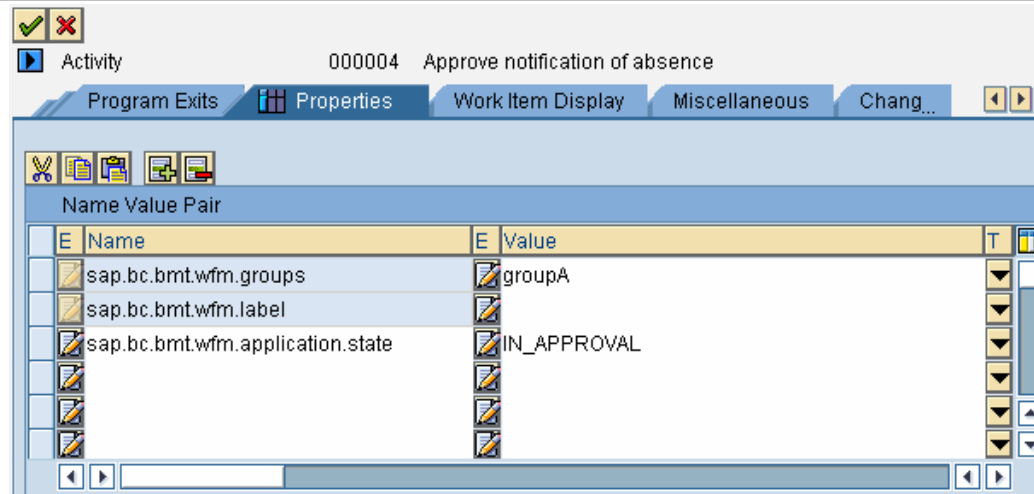


The workflow runtime provides the work item exit with

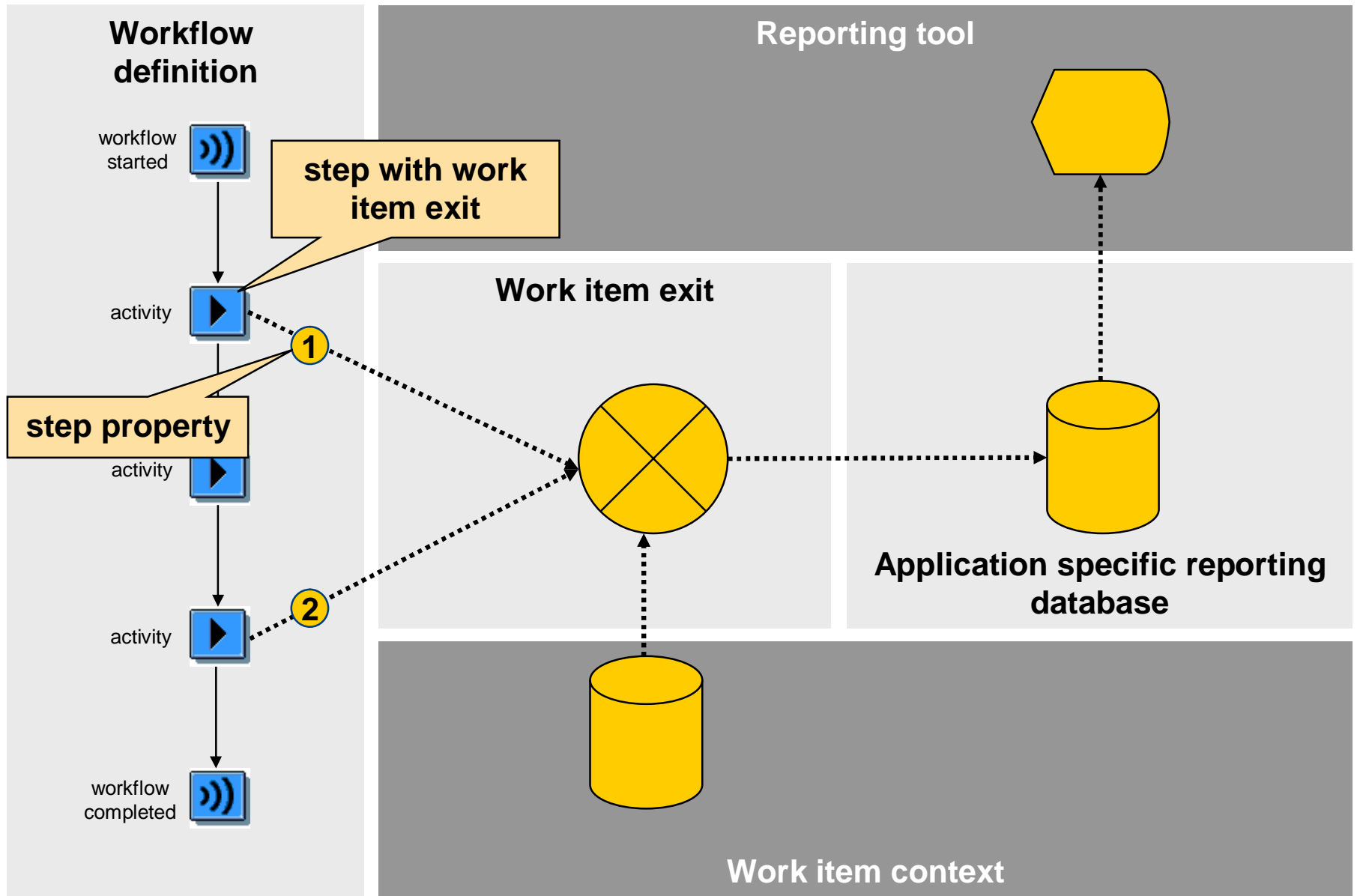
- the name of the lifecycle event
- a work item context object (reference to `IF_WAPI_WORKITEM_CONTEXT`)

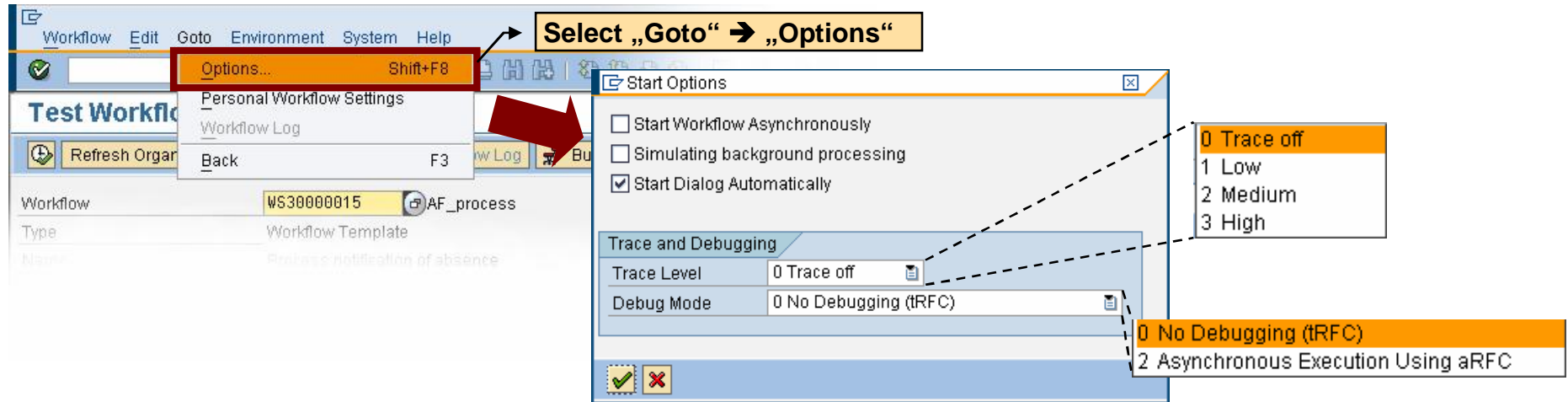
The work item context provides methods to

- read work item data (header, task, agents, texts, ...)
- get container handles (`IF_SWF_IFS_PARAMETER_CONTAINER`)
 - ◆ of the work item itself
 - ◆ of the super ordinate workflow
- read properties (→ see next slide)
- write messages to the workflow log



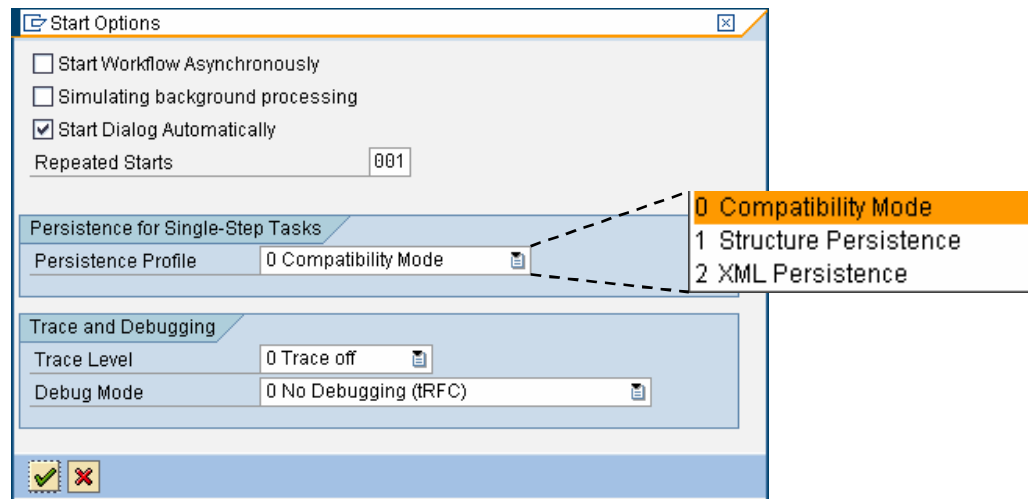
- **Properties are custom labels for workflow definition entities**
 - ◆ steps (activities, decisions, wait steps, web activities)
 - ◆ blocks
 - ◆ workflows
- **Properties can be used to**
 - ◆ categorize steps
 - ◆ mark process states (milestones)
- **Properties are just name value pairs**
- **They can be queried in work item exits (→ custom process reporting)**
 - ◆ „Is the current work item an approval work item?“
 - ◆ „Is the current process still in negotiation phase?“





- Start options can be specified when testing workflows (SWUS)
- Options enable better error analysis
 - ◆ Debugging
 - ◆ Tracing

Option	Explanation
Start Workflow Asynchronously	<ul style="list-style-type: none"> ■ The workflow instance is started via aRFC
Simulate background processing	<ul style="list-style-type: none"> ■ Background steps are executed synchronously, not via tRFC ■ Valuable for debugging background steps
Start Dialog Automatically	<ul style="list-style-type: none"> ■ Switch off synchronous dialog chains
Trace Level	<ul style="list-style-type: none"> ■ Switch on workflow trace and set trace level
Debug Mode	<ul style="list-style-type: none"> ■ Use aRFC instead of tRFC



Extended start options with release 700

Option	Explanation
Repeated Starts	<ul style="list-style-type: none"> Instantiate and start workflow a specified number of times
Persistence Profile	<ul style="list-style-type: none"> Specify container persistence profile for single step tasks (TS...) The persistence profile for workflows (WS...) is specified in the workflow definition

Further Information



Public Web:

www.sap.com

NetWeaver Developer's Guide: www.sdn.sap.com/sdn/developersguide.sdn

SAP Customer Services Network: www.sap.com/services/



Related SAP Education Training Opportunities

<http://www.sap.com/education/>

- 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, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.
 - IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, and Informix are trademarks or registered trademarks of IBM Corporation.
 - Oracle is a registered trademark of Oracle Corporation.
 - 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.
 - Java is a registered trademark of Sun Microsystems, Inc.
 - JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.
 - MaxDB is a trademark of MySQL AB, Sweden.
 - SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, 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 in several other countries all over the world. 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.
 - This document is a preliminary version and not subject to your license agreement or any other agreement with SAP. This document contains only intended strategies, developments, and functionalities of the SAP® product and is not intended to be binding upon SAP to any particular course of business, product strategy, and/or development. Please note that this document is subject to change and may be changed by SAP at any time without notice.
 - SAP assumes no responsibility for errors or omissions in this document. SAP does not warrant the accuracy or completeness of the information, text, graphics, links, or other items contained within this material. 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 shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials. This limitation shall not apply in cases of intent or gross negligence.
 - The statutory liability for personal injury and defective products is not affected. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third-party Web pages nor provide any warranty whatsoever relating to third-party Web pages.