BAdIs in WCM

Release ERP 6.0, EhP3 + EhP5

Michael Lesk
WCM GmbH

WCM Info Day
October 2010
Amsterdam, Netherlands

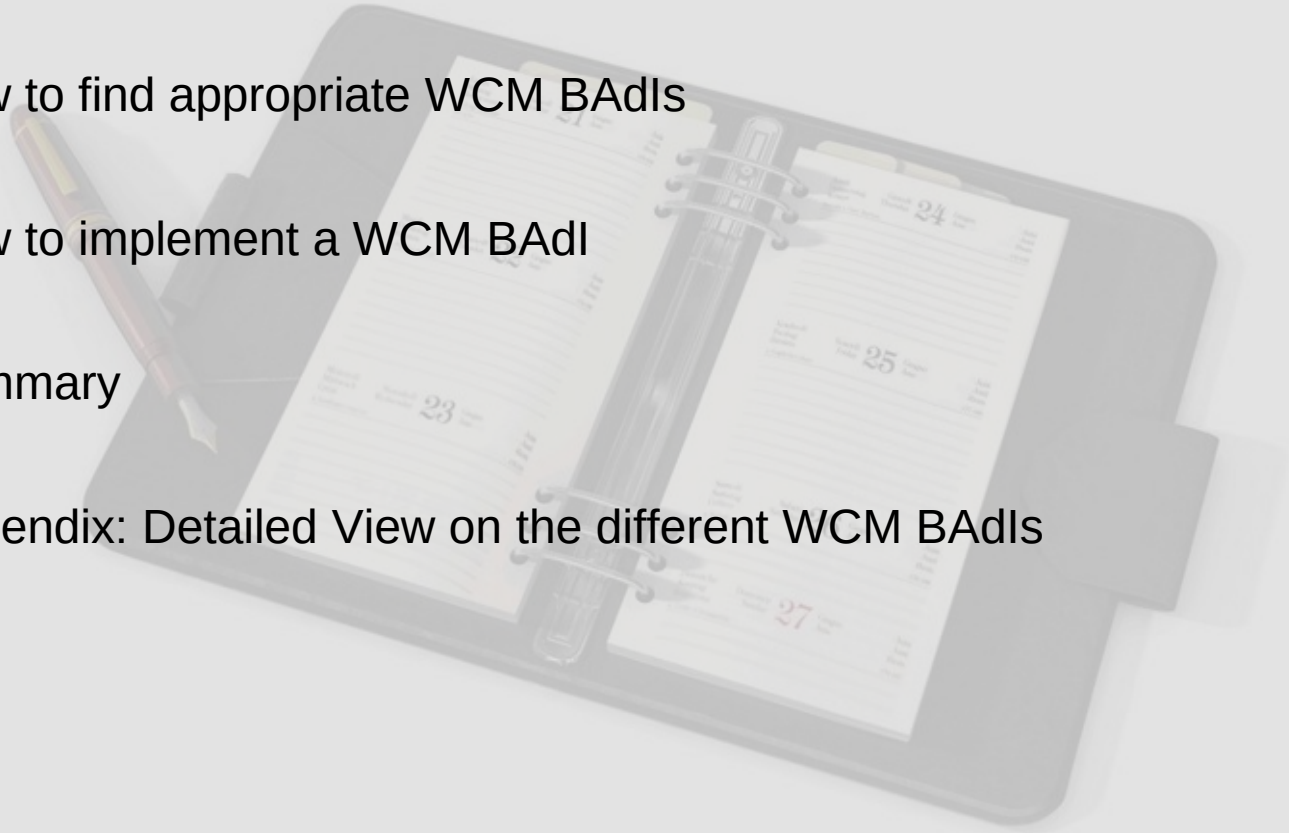
International Conference on SAP Enterprise Asset Management & SAP EHS Management 2010
05 - 07 October 2010, Amsterdam

T.A. Cook

WCM

Agenda

- **1. Introduction and rough Classification**
- 2. How to find appropriate WCM BAdIs
- 3. How to implement a WCM BAdI
- 4. Summary
- 5. Appendix: Detailed View on the different WCM BAdIs



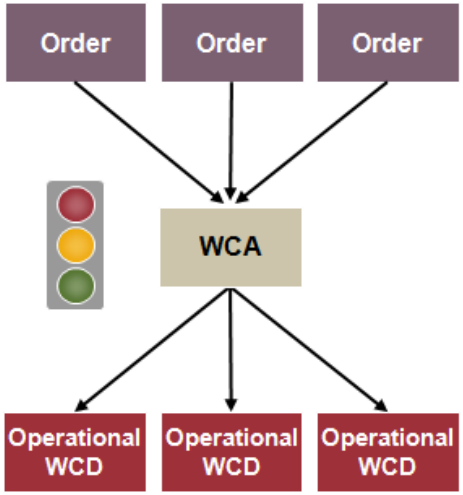
1) Motivation

- As a matter of fact, a standard software component like e.g. SAP-WCM cannot meet every individual customer requirement.
- Situation before Release ERP 6.0, EhP3:
  - To adapt SAP-WCM standard functionality to individual needs, customers needed to modify the system in many cases. However, every modification implies a couple of critical aspects, e.g.:
    - Destabilization risk for delivered standard functionality
    - Potential difficulties when importing SAP correction notes
    - Higher efforts for future software upgrades (due to the need to merge and possibly adjust the modifications)
- Follow-up action for Release ERP 6.0, EhP3:
  - Based on a roll-in of customer requirements, SAP has provided various BAdIs for processing single WCM objects with release ERP 6.0, EhP3.
- Further action for Release ERP 6.0, EhP5:
  - In addition to the BAdIs for processing single WCM objects, SAP provides further BAdIs for list editing of WCM objects with release ERP 6.0, EhP5.

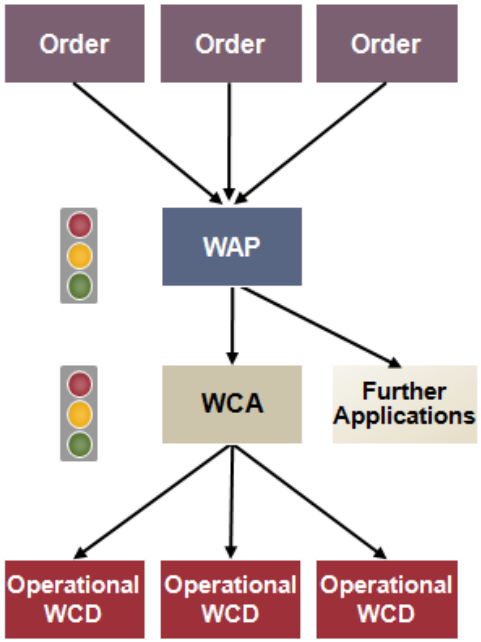
1) BAdI?

- BAdI = Business Add-In.
- A BAdI is an enhancement option providing an easy mechanism to change the functionality of a well-defined business function without modifying the delivered source code.
- Future standard upgrades can be applied without impact / follow-up investments, i.e. without losing the customer-specific enhancements and without the need to merge the changes.
- Compared to the old customer exits, BAdIs have a lot of further advantages.
  - BAdIs are object-oriented, they encapsulate a couple of methods belonging together from an intuitive business (object) perspective. Customer exits were simple function modules, they could not offer such an encapsulation by design.
  - One of the most important advantages is that BAdIs no longer assume a two-level software infrastructure (SAP standard, customer), but instead allow for a multi-level landscape (SAP standard, SAP industries, partner, customer).
    - Definitions and implementations of BAdIs can be created at each level within such a infrastructure.

1) Let's start with a look at the WCM Architecture…



The Standard Model

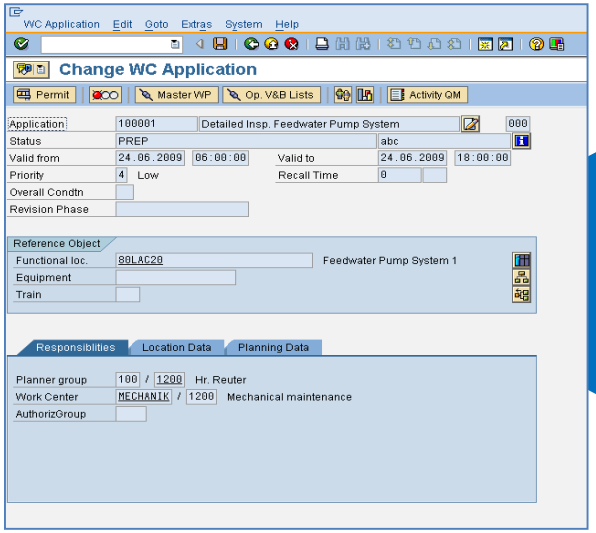


The Enhanced Model

- For the whole WCM process, from the order down to WCM and back, customers asked for the option to influence the process steps by implementing individual business checks, e.g. during status transitions of order and WCM objects.

  → SAP provides BAdIs for controlling different WCM process steps.

1) ...and continue with a look at a single WCM object...



Generic object services:
different services are possible

Status processing and audit trail:
system status and user status

Assignment of partners:
different roles possible

Long text processing:
several internal remarks

Assignment of documents:
link to document management system

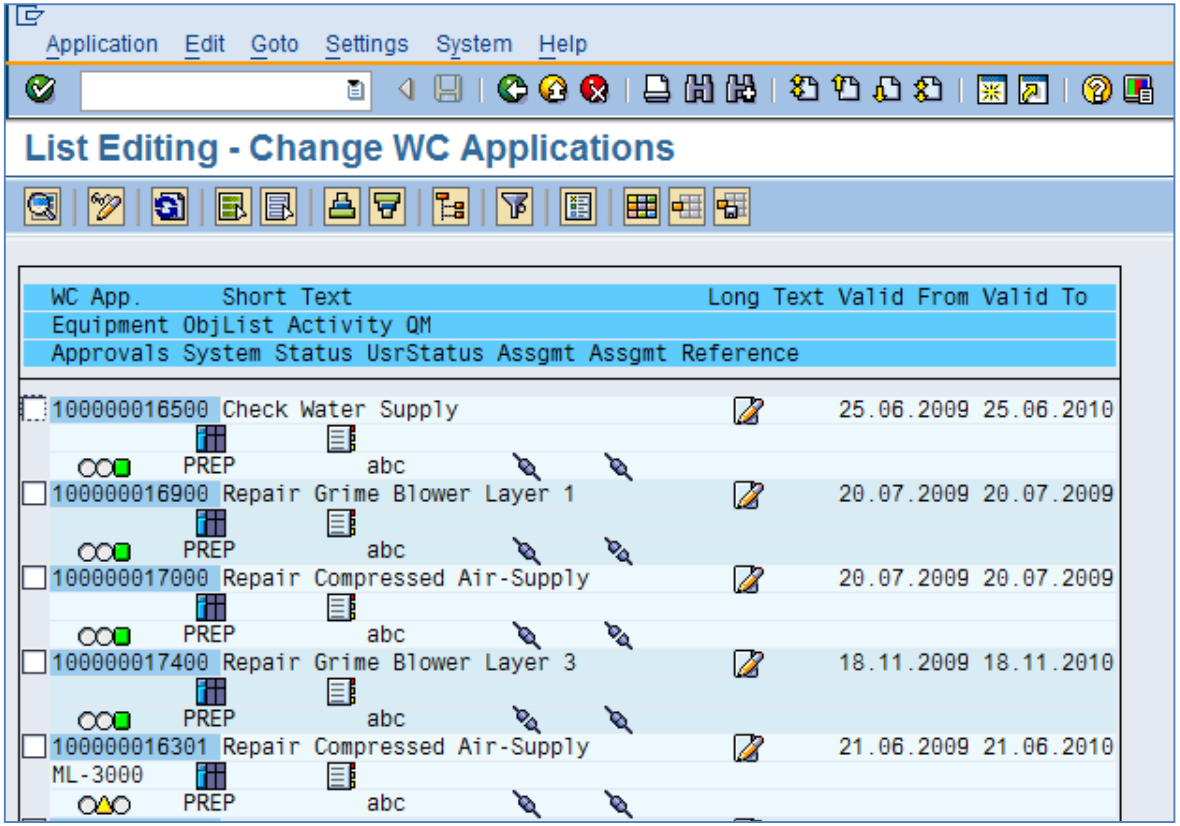
Assignment of approvals:
cross-document, interactive

Use of catalog technique:
catalog groups and codes

Assignment of technical objects:
equipment and functional locations

- For processing WCM objects like e.g. the WCA, SAP was asked for the option to
  - enhance the screens by individual fields → SAP provides screen BAdIs
  - enhance the menus by individual functions → SAP provides menu BAdIs
  - control processing of additional data → SAP provides BAdIs for additional data

1) ...and finish with a look at WCM list editing



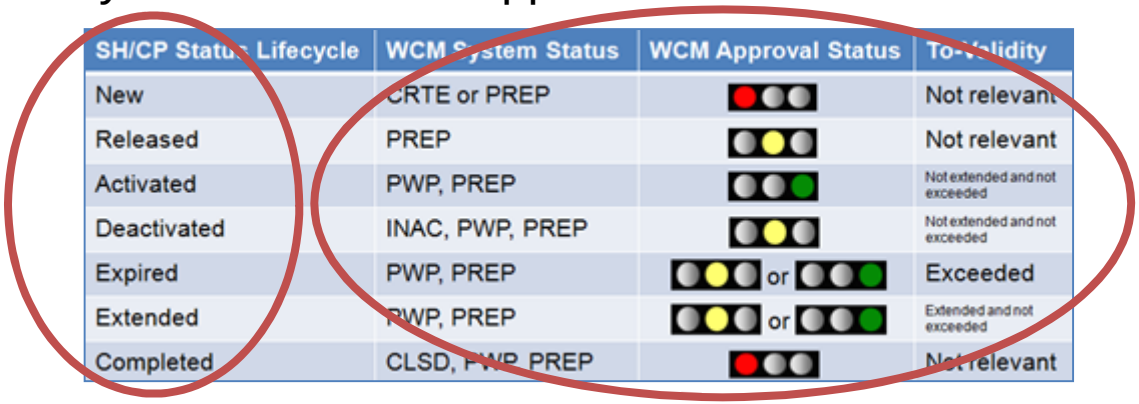
- For list editing of WCM objects like e.g. WCAs, SAP was asked for the option to
  - add individual columns to selection result → SAP provides BAdIs for list selection
  - enhance the menus by individual functions → SAP provides menu BAdIs for lists

1) WCM BAdIs – Rough Classification (1/2)

- So we have a first rough classification of WCM BAdIs:
  - BAdIs for processing single (WCM) objects (EhP3):
    - Basic process-controlling BAdIs for order and single WCM objects
      Example: Control ‘Set prepared’ within an Operational WCD by performing a customer-specific consistency check.
    - BAdIs for additional data of single WCM objects
      Example: Control assignment of an Operational WCD within a WCA based on the status of the Operational WCD.
    - BAdIs for customer-specific screen enhancements of single WCM objects
      Example: Display existing assignments of Operational WCDs immediately on the header screen of a WCA.
    - BAdIs for customer-specific menu enhancements of single WCM objects
      Example: Provide a menu function for a customer-specific consistency check within an Operational WCD (same check as processed automatically during ‘Set prepared’ above).

1) WCM BAdIs – Rough Classification (2/2)

- So we have a first rough classification of WCM BAdIs:
  - BAdIs for list editing of WCM objects (EhP5):
    - BAdIs for WCM list selection result list
      Example: Add an additional column to the result list of WCA list selection, showing the lifecycle phase for each list entry as a combination of validity, system status and approval status:

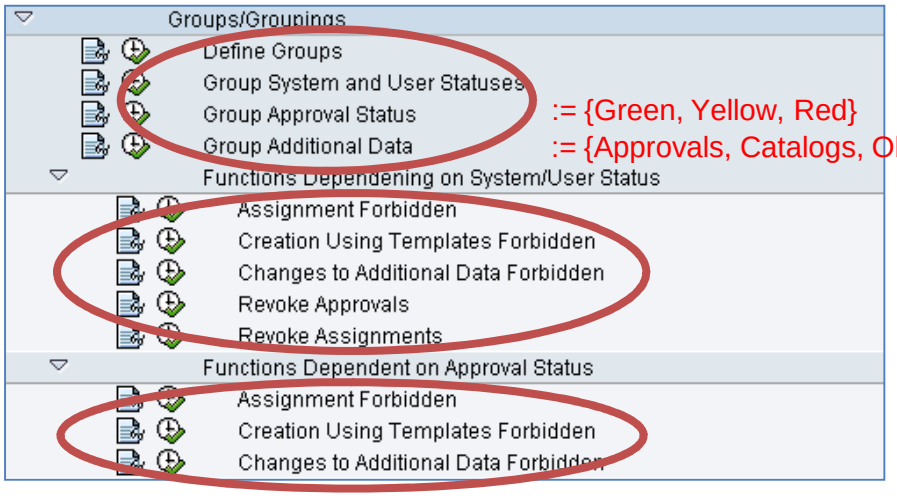


SH/CP Status Lifecycle	WCM System Status	WCM Approval Status	To-Validity
New	CRTE or PREP	🔴🟡🟡	Not relevant
Released	PREP	🟡🟡🟡	Not relevant
Activated	PWP, PREP	🟡🟡🟢	Not extended and not exceeded
Deactivated	INAC, PWP, PREP	🟡🟡🟡	Not extended and not exceeded
Expired	PWP, PREP	🟡🟡🟡 or 🟡🟡🟢	Exceeded
Extended	PWP, PREP	🟡🟡🟡 or 🟡🟡🟢	Extended and not exceeded
Completed	CLSD, PWP, PREP	🔴🟡🟡	Not relevant

    - BAdIs for menu enhancements of WCM list processing (supporting mass transactions for WCM objects)
      Example: Provide a menu function for a customer-specific consistency check for a number of selected Operational WCDs (same check as provided within Operational WCD maintenance on the previous slide).

1) Status Groups

- Before having a closer look at the WCM BAdIs, we want to note the very powerful concept of status groups:
  - Status groups allow configuration of pre-defined influence options on certain process steps, based on
    - Approval status (as of release 4.7)
    - System status (as of release 4.7)
    - User status (as of release ERP 6.0, EhP4)
  - Status groups are completely set up in Customizing, additional implementation is not required.

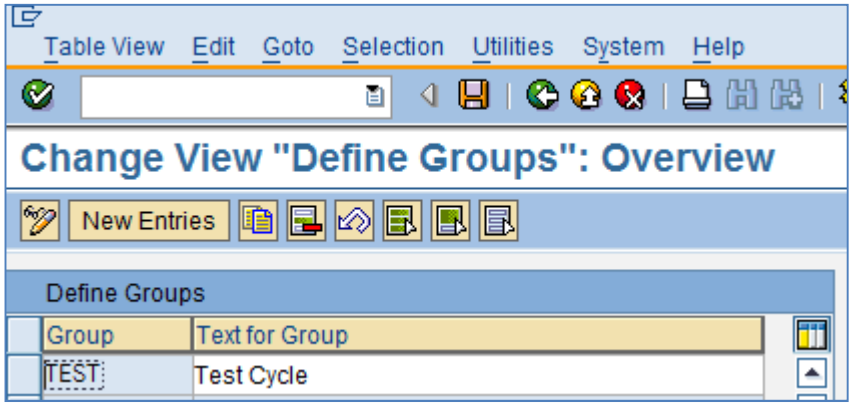


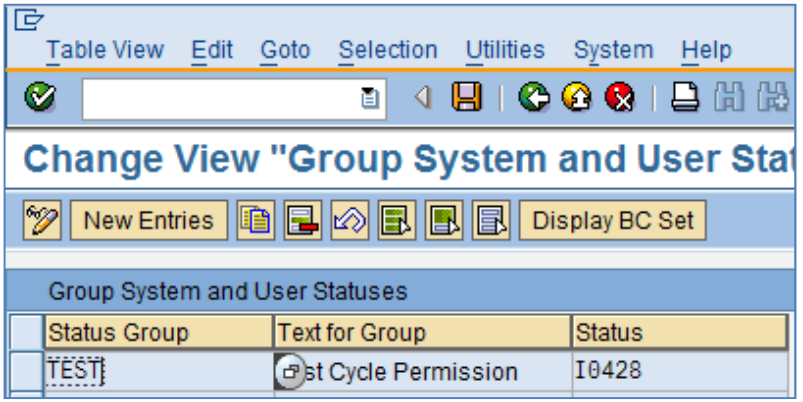
:= {Green, Yellow, Red}

:= {Approvals, Catalogs, Object List, Partners, Documents, Texts}

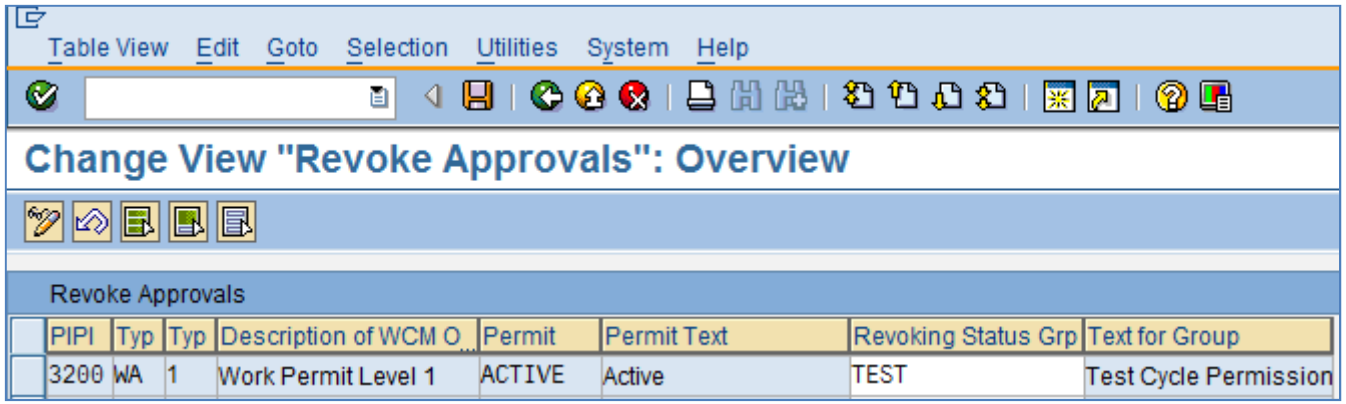
1) Status Groups – Simple Example

- Requirement: We want to revoke the final safety approval (green traffic light) of the WCA automatically as soon as the test cycle has been permitted.
  - Step 1: Define a group TEST and assign appropriate system status to it:



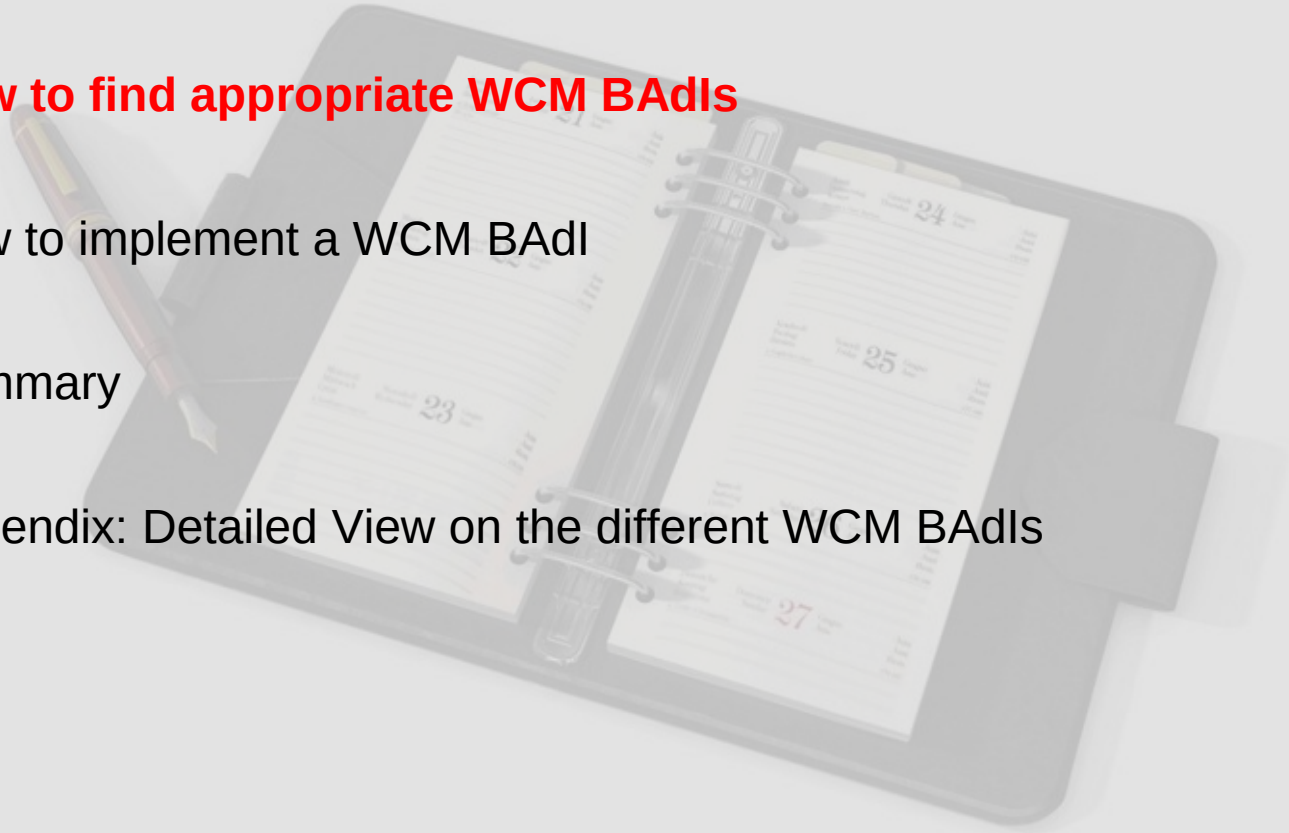


  - Step 2: Assign TEST as “revoking status group” to the final safety approval:



Agenda

- 1. Introduction and rough Classification
- **2. How to find appropriate WCM BAdIs**
- 3. How to implement a WCM BAdI
- 4. Summary
- 5. Appendix: Detailed View on the different WCM BAdIs



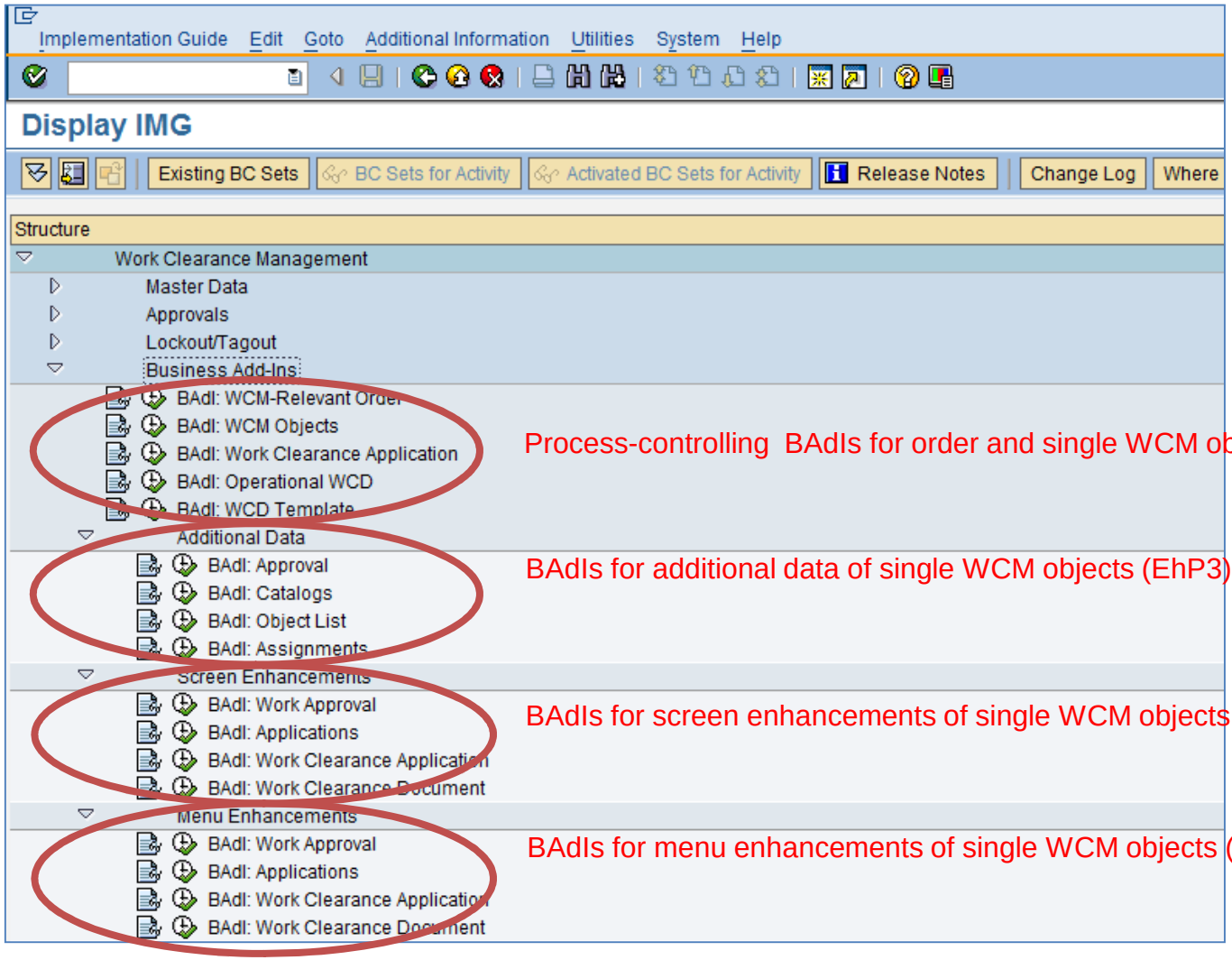
2) How to find the appropriate WCM BAdI? (1/2)

- Preparation / prerequisites:
  - Exact specification of the affected business scenario / resulting requirement(s).
  - How can the business requirements be met in SAP-WCM standard?
    - Includes check of standard IMG configuration options (e.g. status groups).
- In case of requirements that cannot be met in SAP-WCM standard:
  - Which business objects are involved?
  - Which aspects of a business object are involved?
    - Basic process steps during business object maintenance (e.g. status transitions)?
    - Additional data assigned during business object maintenance (e.g. object assignments, approvals)?
    - Data screens for business object maintenance?
    - Menu functions for business object maintenance?
    - Columns for result lists of business object list selection?
    - Menu functions for business object list processing?

2) How to find the appropriate WCM BAdI? (2/2)

- Open the IMG to search for appropriate BAdIs:
  - IMG is the central entry point for all WCM BAdIs.
  - IMG provides detailed documentation for every WCM BAdI, including the contained methods/interfaces.
  - IMG supports access to the development environment (BAdI Builder).
  - IMG enables activation and deactivation of every implemented BAdI.
- Note that it might happen that you don't see the WCM BAdIs in the IMG.
  - This is due to the fact that appearance in IMG requires EAM business function activation in the Switch Framework
    - LOG_EAM_CI_2 for the WCM BAdIs delivered with EhP3.
    - LOG_EAM_WCM_1 for the WCM BAdIs delivered with EhP5.
  - Technically, each BAdI interface could also be accessed directly via BAdI Builder (SE18). However, activation of above business functions is highly recommended in any case.

2) IMG – WCM BAdIs for single WCM objects (EhP3)



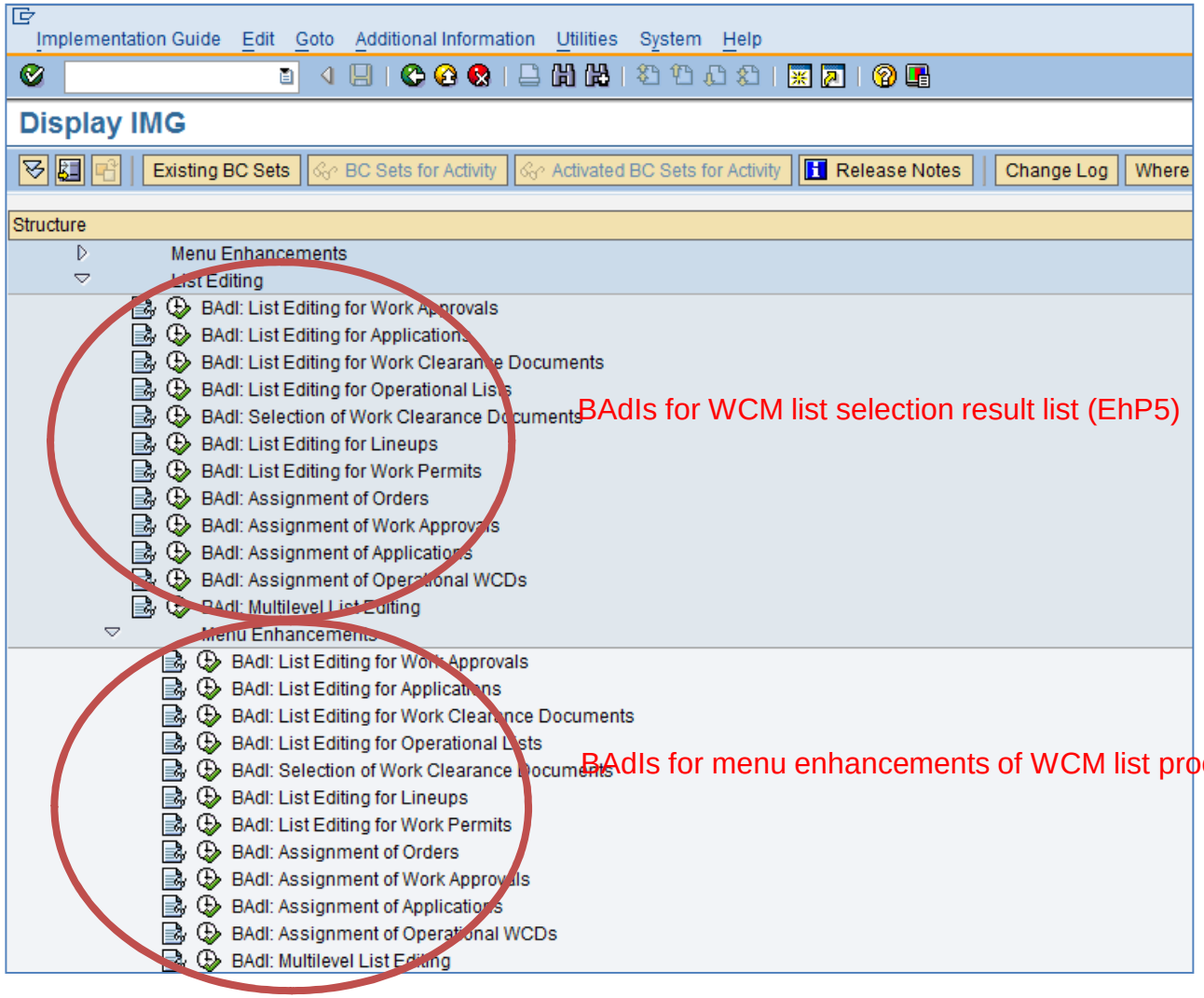
Process-controlling BAdIs for order and single WCM objects (EhP3)

BAdIs for additional data of single WCM objects (EhP3)

BAdIs for screen enhancements of single WCM objects (EhP3)

BAdIs for menu enhancements of single WCM objects (EhP3)

2) IMG – WCM BAdIs for WCM list editing (EhP5)

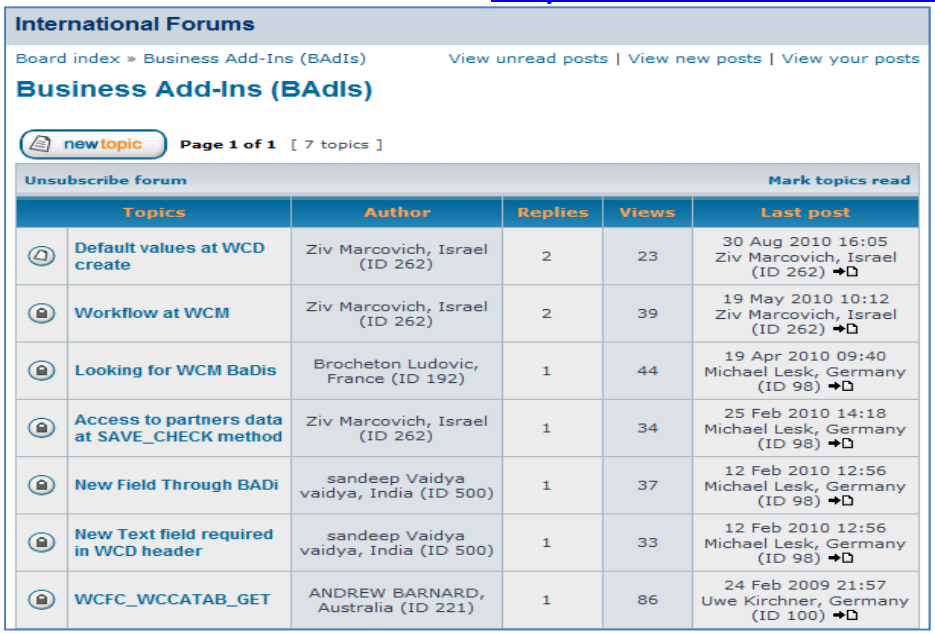


2) WCM BAdIs – Basic Facts

- Looking at the number of methods contained in each BAdI on one hand and at the reuse across several WCM objects on the other, one can calculate the number of BAdI access points provided in WCM standard:
  - EhP3: ~ 300 access points
  - EhP5: ~ 50 additional access points
- In general, the WCM BAdIs provided for customer-specific checks constitute a strengthening as they do not replace the standard checks, but rather complement them.
  - As a matter of principle, a check BAdI is only called after related standard checks have been (successfully!) processed.
- Reusability of the WCM BAdIs can be maximized when taking into account that beyond its semantic context, each BAdI interface can be simply considered as a dedicated point-of-time allowing customer-specific control.
  - For example, a BAdI interface for checking if a dialog window can be closed could be used for other purposes instead, e.g. for updating customer-specific data depending on the dialog window settings.

2) WCM BAdIs – Additional Information

- In general, the WCM BAdI import interfaces are kept rather lean, i.e. they do not provide all and every data that might be required during BAdI processing, but only data that is obviously relevant in the respective semantic context.
  - Access to further data within a BAdI implementation can be gained by using the WCM function modules with suffix GLOBAL_DATA_GET.
- Additional information on WCM BAdIs is available on the WCM website.
  - Download area: http://www.wcm-it.com/cms/front_content.php?idcat=62
  - Dedicated BAdI forum: http://www.wcm-it.com/cms/front_content.php?idcat=9



International Forums

Board index » Business Add-Ins (BAdIs) — View unread posts | View new posts | View your posts

Business Add-Ins (BAdIs)

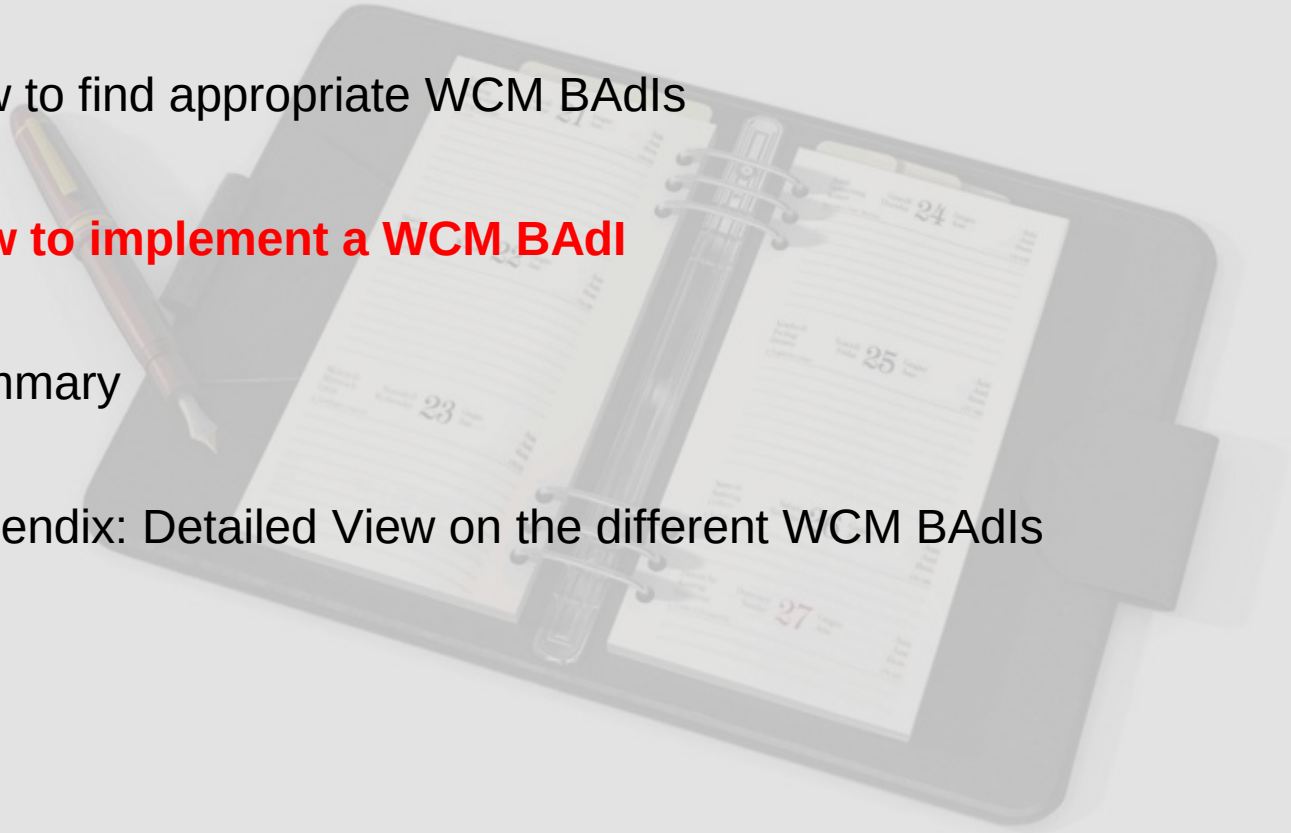
new topic — Page 1 of 1 [7 topics]

Unsubscribe forum — Mark topics read

Topics	Author	Replies	Views	Last post
Default values at WCD create	Ziv Marcovich, Israel (ID 262)	2	23	30 Aug 2010 16:05 Ziv Marcovich, Israel (ID 262)
Workflow at WCM	Ziv Marcovich, Israel (ID 262)	2	39	19 May 2010 10:12 Ziv Marcovich, Israel (ID 262)
Looking for WCM BaDis	Brocheton Ludovic, France (ID 192)	1	44	19 Apr 2010 09:40 Michael Lesk, Germany (ID 98)
Access to partners data at SAVE_CHECK method	Ziv Marcovich, Israel (ID 262)	1	34	25 Feb 2010 14:18 Michael Lesk, Germany (ID 98)
New Field Through BADi	sandeep Vaidya vaidya, India (ID 500)	1	37	12 Feb 2010 12:56 Michael Lesk, Germany (ID 98)
New Text field required in WCD header	sandeep Vaidya vaidya, India (ID 500)	1	33	12 Feb 2010 12:56 Michael Lesk, Germany (ID 98)
WCFC_WCCATAB_GET	ANDREW BARNARD, Australia (ID 221)	1	86	24 Feb 2009 21:57 Uwe Kirchner, Germany (ID 100)

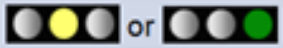
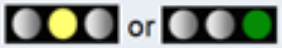
Agenda

- 1. Introduction and rough Classification
- 2. How to find appropriate WCM BAdIs
- **3. How to implement a WCM BAdI**
- 4. Summary
- 5. Appendix: Detailed View on the different WCM BAdIs



3) BAdI Implementation – Example (1/5)

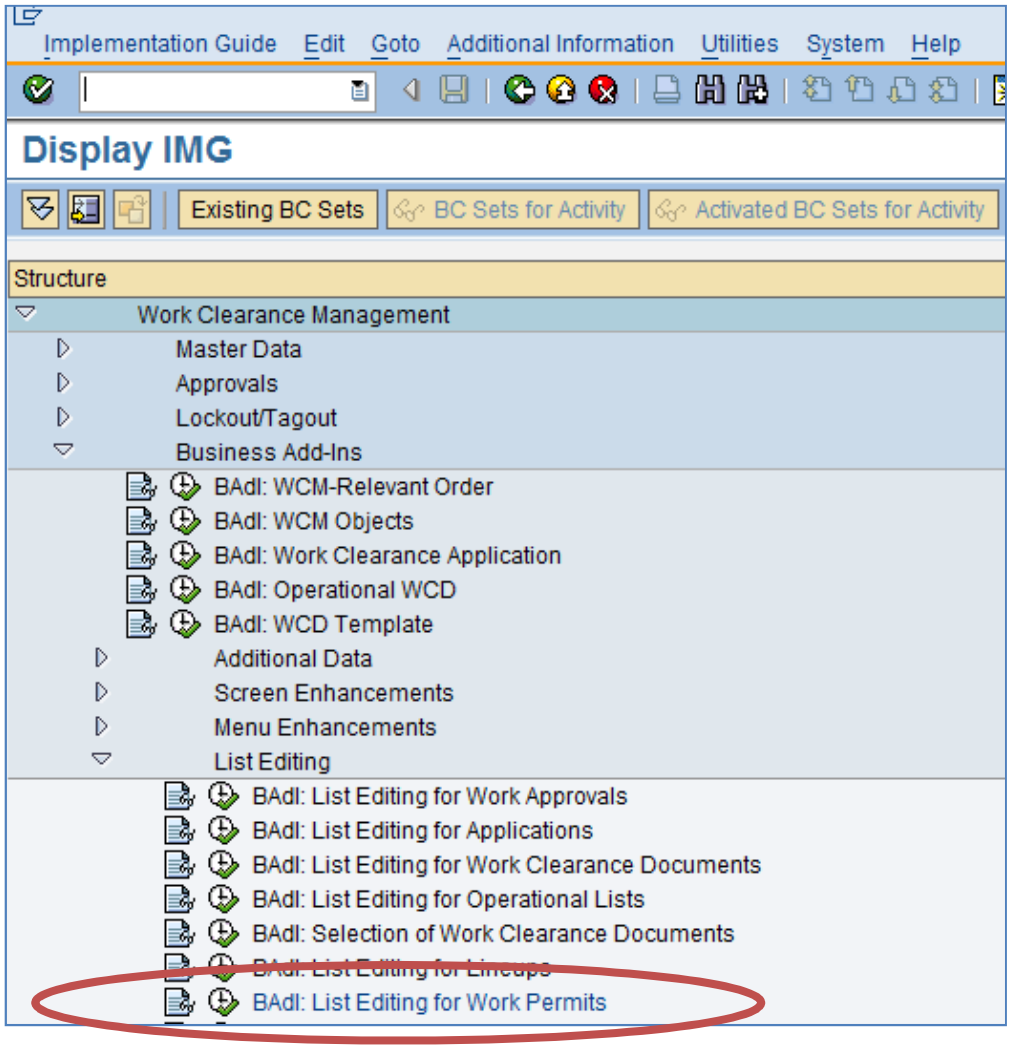
- Requirement: In the work permit/order list selection (introduced with EhP5), we want to have a column in the result list that shows the lifecycle phase for each list entry as a combination of validity, system status and approval status:

SH/CP Status Lifecycle	WCM System Status	WCM Approval Status	To-Validity
New	CRTE or PREP		Not relevant
Released	PREP		Not relevant
Activated	PWP, PREP		Not extended and not exceeded
Deactivated	INAC, PWP, PREP		Not extended and not exceeded
Expired	PWP, PREP		Exceeded
Extended	PWP, PREP		Extended and not exceeded
Completed	CLSD, PWP, PREP		Not relevant

- Obviously, this requirement is not covered within the standard list selection, as the column is not offered of the underlying ALV field catalog.

3) BAdI Implementation – Example (2/5)

- Exploring the IMG leads us to the following activity:

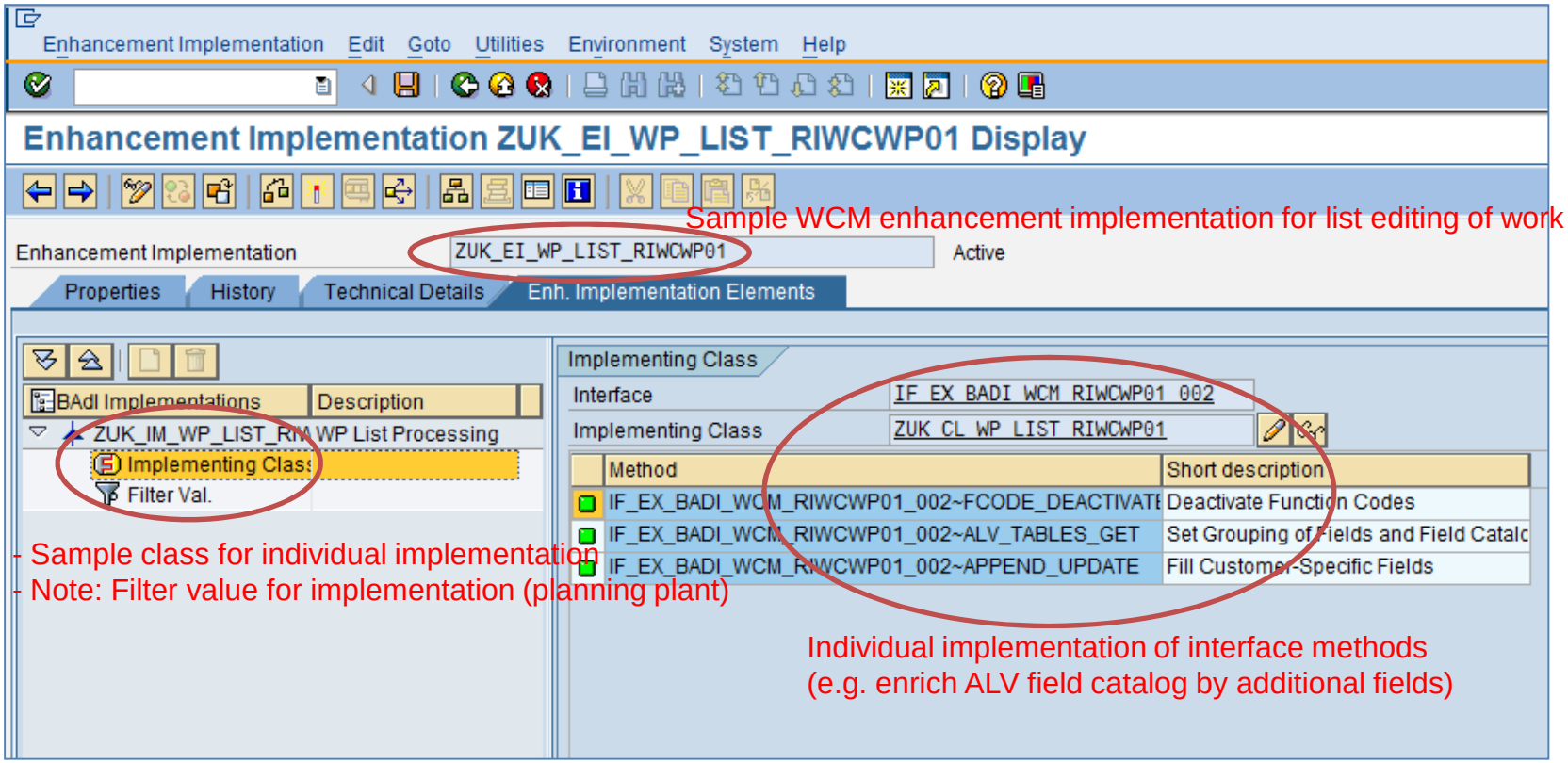


3) BAdI Implementation – Example (3/5)

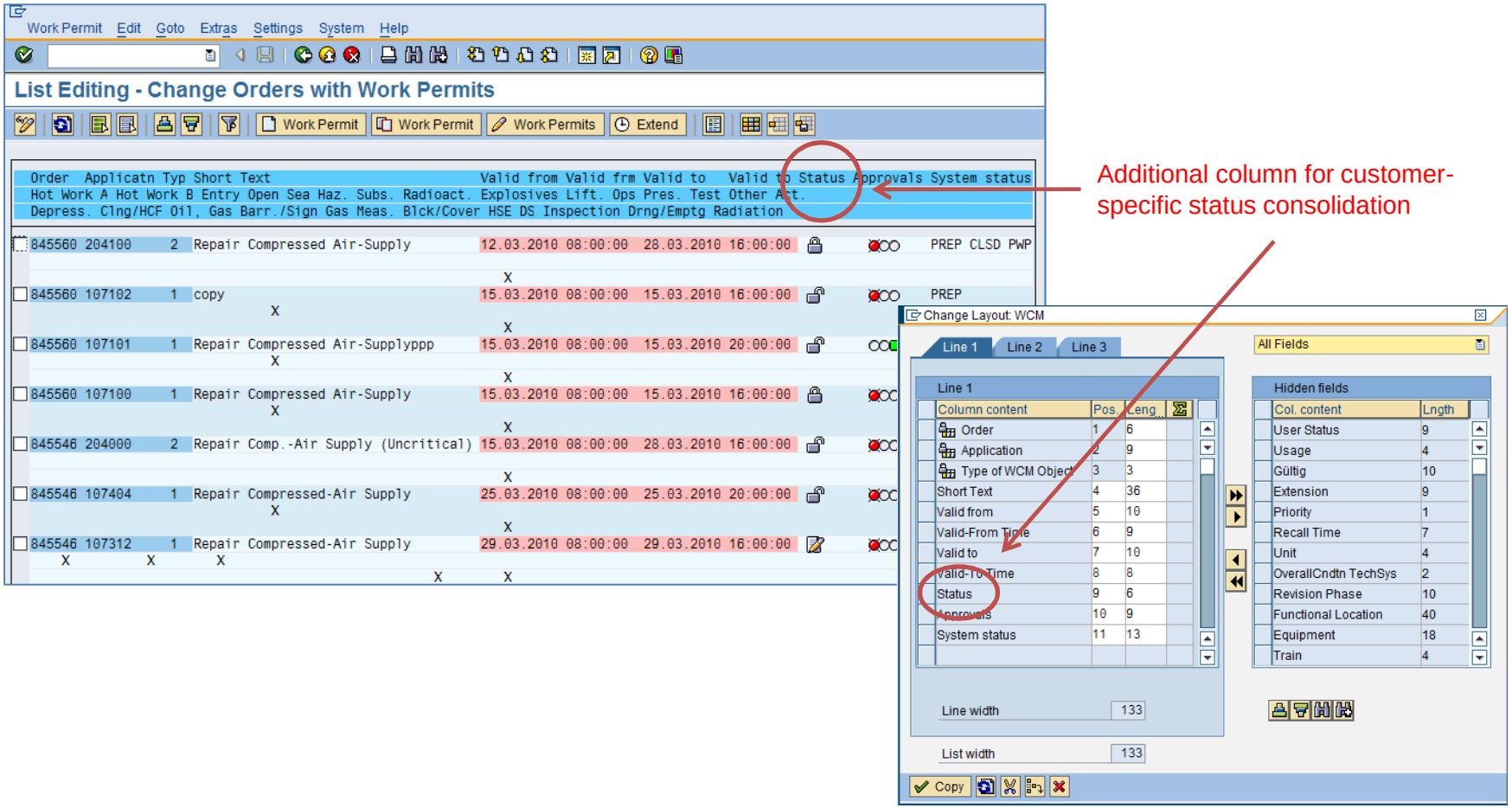
- Reading the documentation of the IMG activity, we learn that the BAdI for the work permit/order list selection provide the following interface methods:
  - ALV_TABLES_GET: Set ALV field catalog and grouping of fields for the result list
  - APPEND_UPDATE: Fill customer-specific fields in the result list
  - FCODE_DEACTIVATE: Deactivate function codes for the result list
- Note that the BAdIs for list selection are (nearly) identical for all WCM objects:
  - All BAdIs (except the one for the multilevel list) provide above interface methods.
- So to meet our requirement, we simply have to do two things:
  - We have to add the new field for the lifecycle phase to the ALV field catalog.
    → Implementation of method ALV_TABLES_GET.
  - We have to provide a logic filling the new field with appropriate values.
    → Implementation of method APPEND_UPDATE.

3) BAdI Implementation – Example (4/5)

- Clicking on the IMG activity leads us to the BAdI Builder for creation of an enhancement implementation for work permit/order list selection.



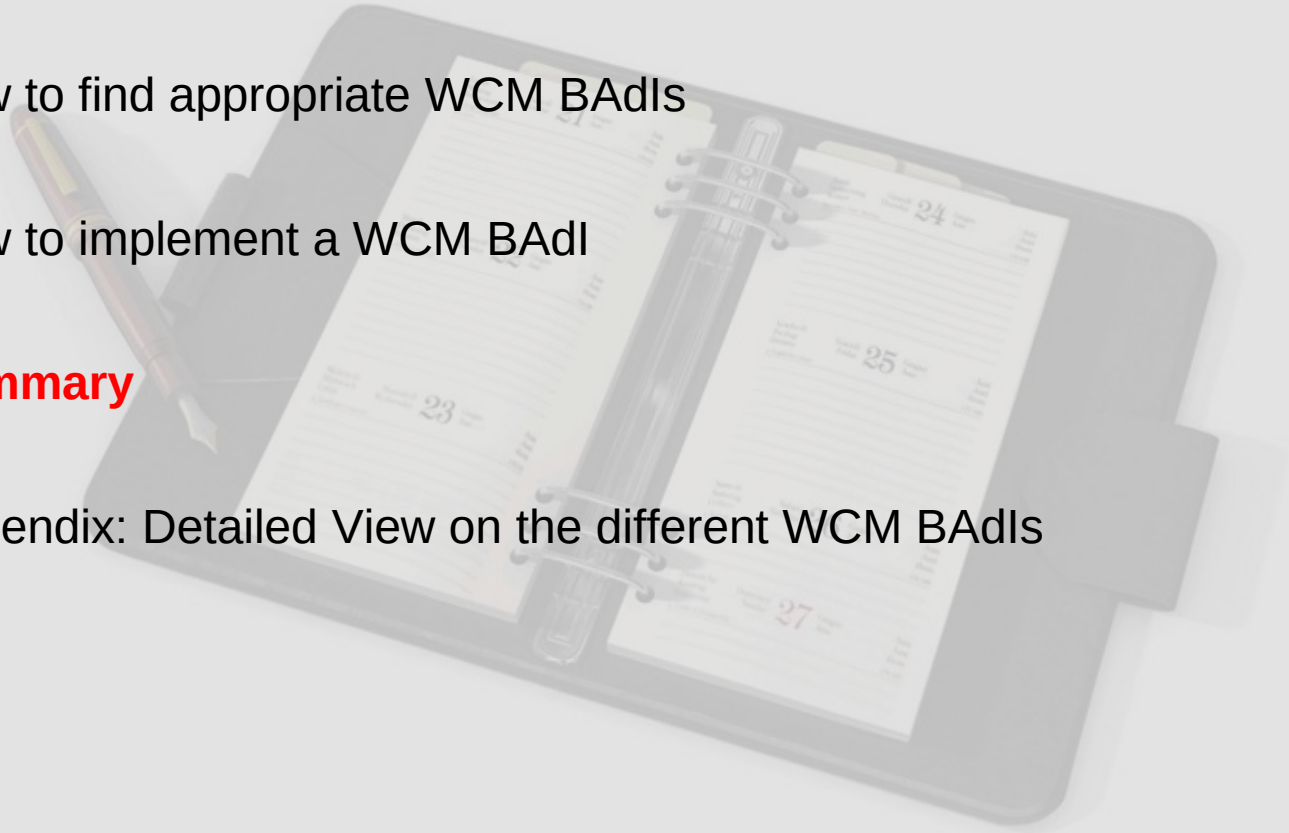
3) BAdI Implementation – Example (5/5)



- Method ALV_TABLES_GET adds customer-specific field to ALV field catalog.
- Method APPEND_UPDATE calculates the field value (not stored on DB).

Agenda

- 1. Introduction and rough Classification
- 2. How to find appropriate WCM BAdIs
- 3. How to implement a WCM BAdI
- **4. Summary**
- 5. Appendix: Detailed View on the different WCM BAdIs



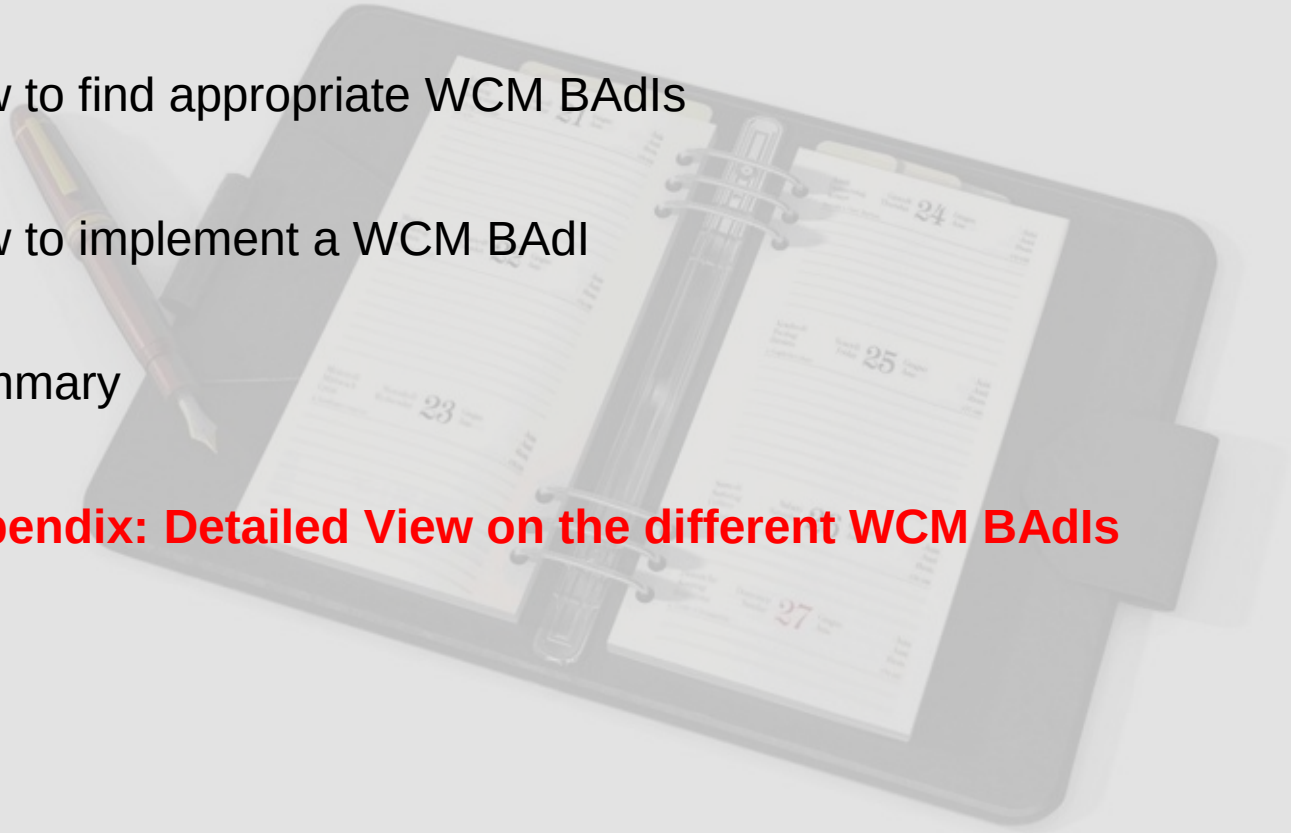
4) Key Messages

- General benefits of BAdIs:
  - Modification-free enhancement of SAP standard.
  - No impact on imports of SAP correction notes and future release upgrades.
- Additional benefits of WCM BAdIs:
  - Huge number of access points (~ 350) for customer-specific logic.
  - For every WCM object, enhancements are available with regard to:
    - Basic processing for single WCM objects.
    - Additional data assigned to single WCM objects.
    - Customer-specific screen enhancements for single WCM objects.
    - Customer-specific menu enhancements for single WCM objects.
    - Additional columns in result lists of WCM object list selection.
    - Customer-specific menu enhancements for WCM object list processing.
  - IMG serves as central entry point to all WCM BAdIs.
  - Availability of extensive documentation (SAP system, WCM website).

Questions and Answers

Agenda

- 1. Introduction and rough Classification
- 2. How to find appropriate WCM BAdIs
- 3. How to implement a WCM BAdI
- 4. Summary
- **5. Appendix: Detailed View on the different WCM BAdIs**

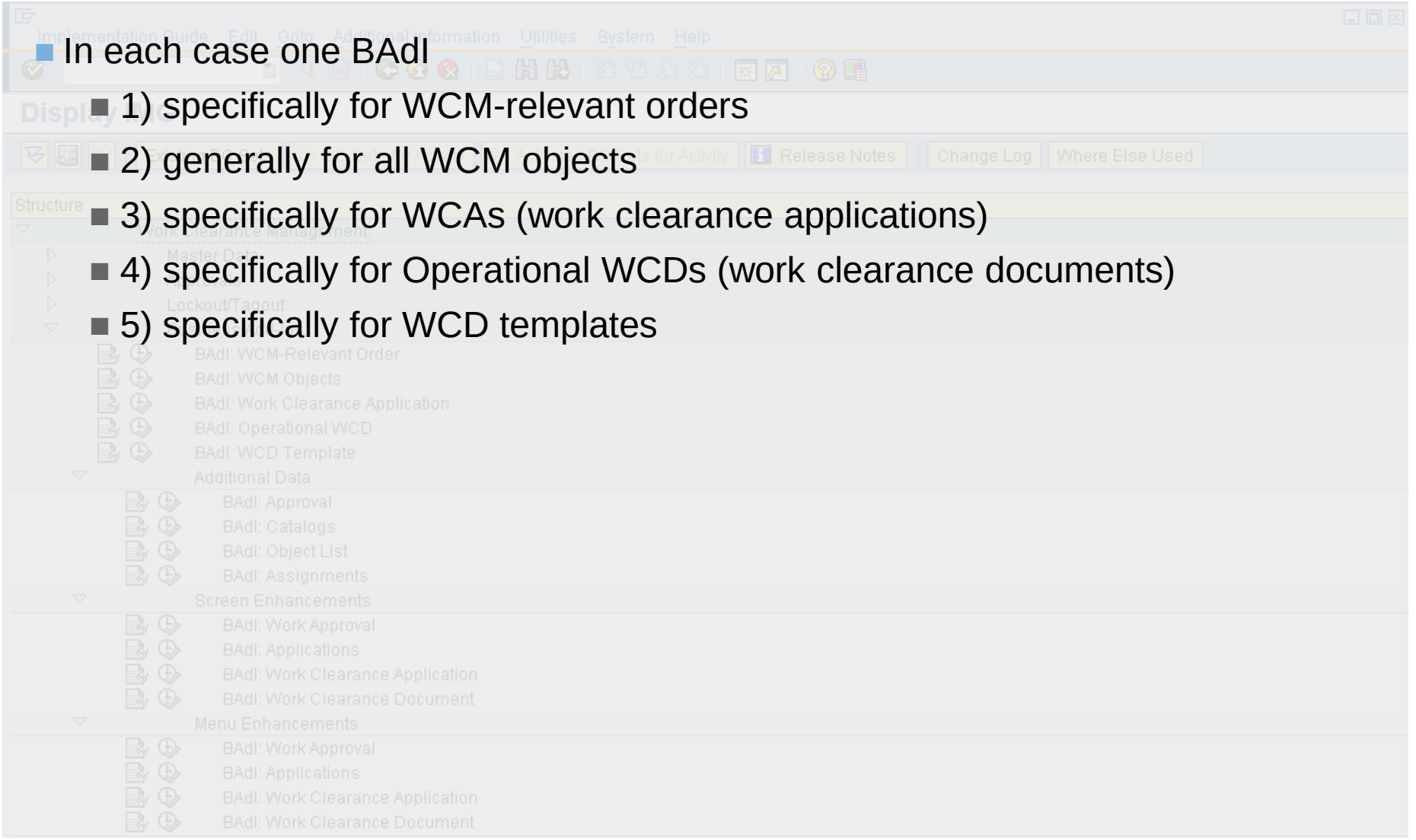


5) Appendix: Detailed View on the different WCM BAdIs

- **5.1. Process-controlling BAdIs for Order and single WCM Objects (EhP3)**
- 5.2. BAdIs for Additional Data of single WCM Objects (EhP3)
- 5.3. BAdIs for Screen Enhancements of single WCM Objects (EhP3)
- 5.4. BAdIs for Menu Enhancements of single WCM Objects (EhP3)
- 5.5. BAdIs for WCM List Selection Output (EhP5)
- 5.6. BAdIs for Menu Enhancements of WCM List Processing (EhP5)

5.1) Process-controlling BAdIs for Order and single WCM Objects

- In each case one BAdI
  - 1) specifically for WCM-relevant orders
  - 2) generally for all WCM objects
  - 3) specifically for WCAs (work clearance applications)
  - 4) specifically for Operational WCDs (work clearance documents)
  - 5) specifically for WCD templates



5.1.1) Process-controlling BAdI for WCM-relevant Orders Header Level

- General BAdI methods for WCM-relevant orders:
  - Change color or short text (e.g. in multi-level list)

- Methods for valuation of WCM-relevant orders:
  - Deactivate function codes in the valuation dialog
  - Check if application can be valuated as requested (“Yes”, “No”)
  - Check if the valuation dialog can be closed as requested (by the green check)

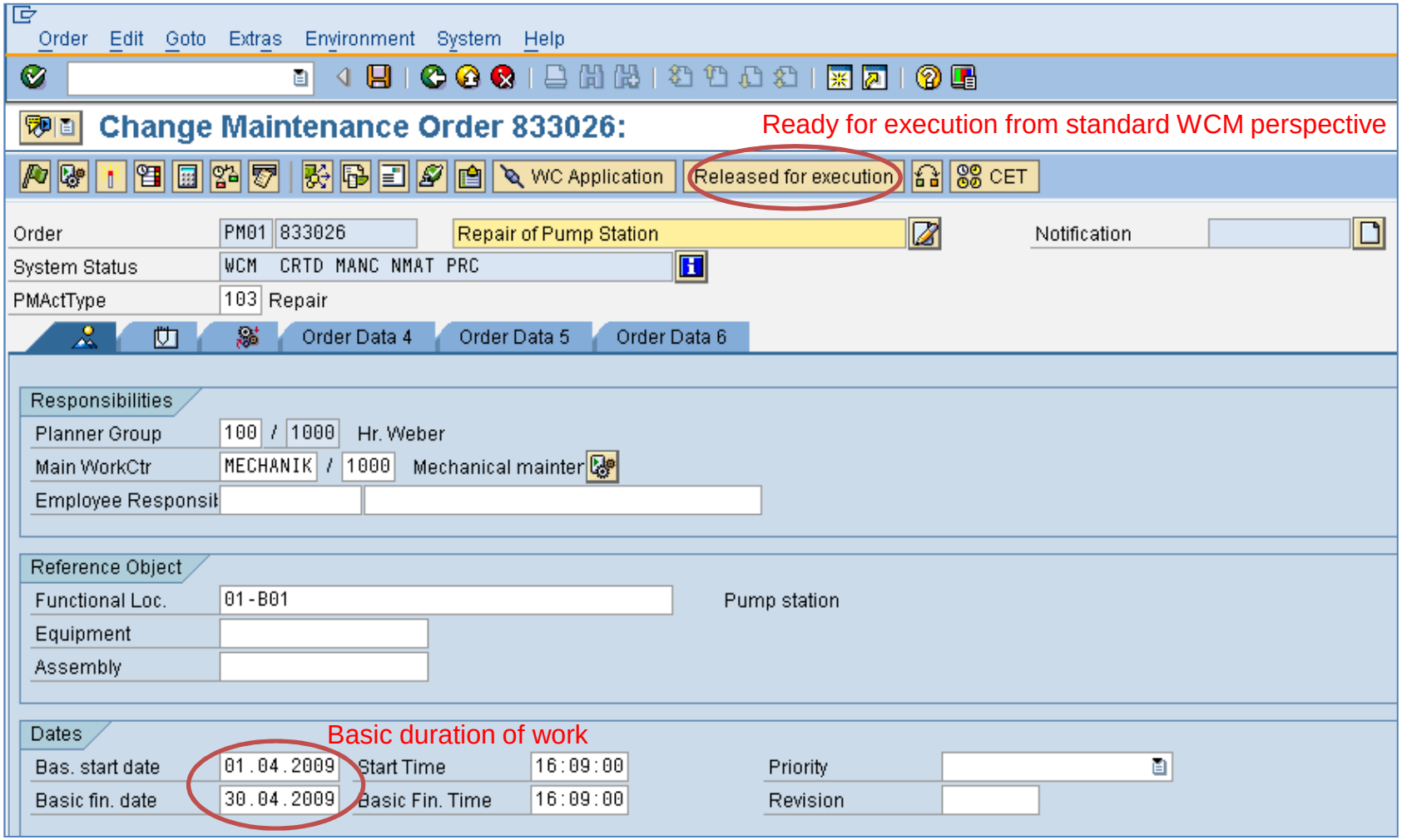
- Methods for work release/completion in a WCM-relevant order:
  - Check if “Release for Execution” can be issued as requested
  - Check if “Work Completed” can be confirmed as requested

5.1.1) Process-controlling BAdI for WCM-relevant Orders Operation Level

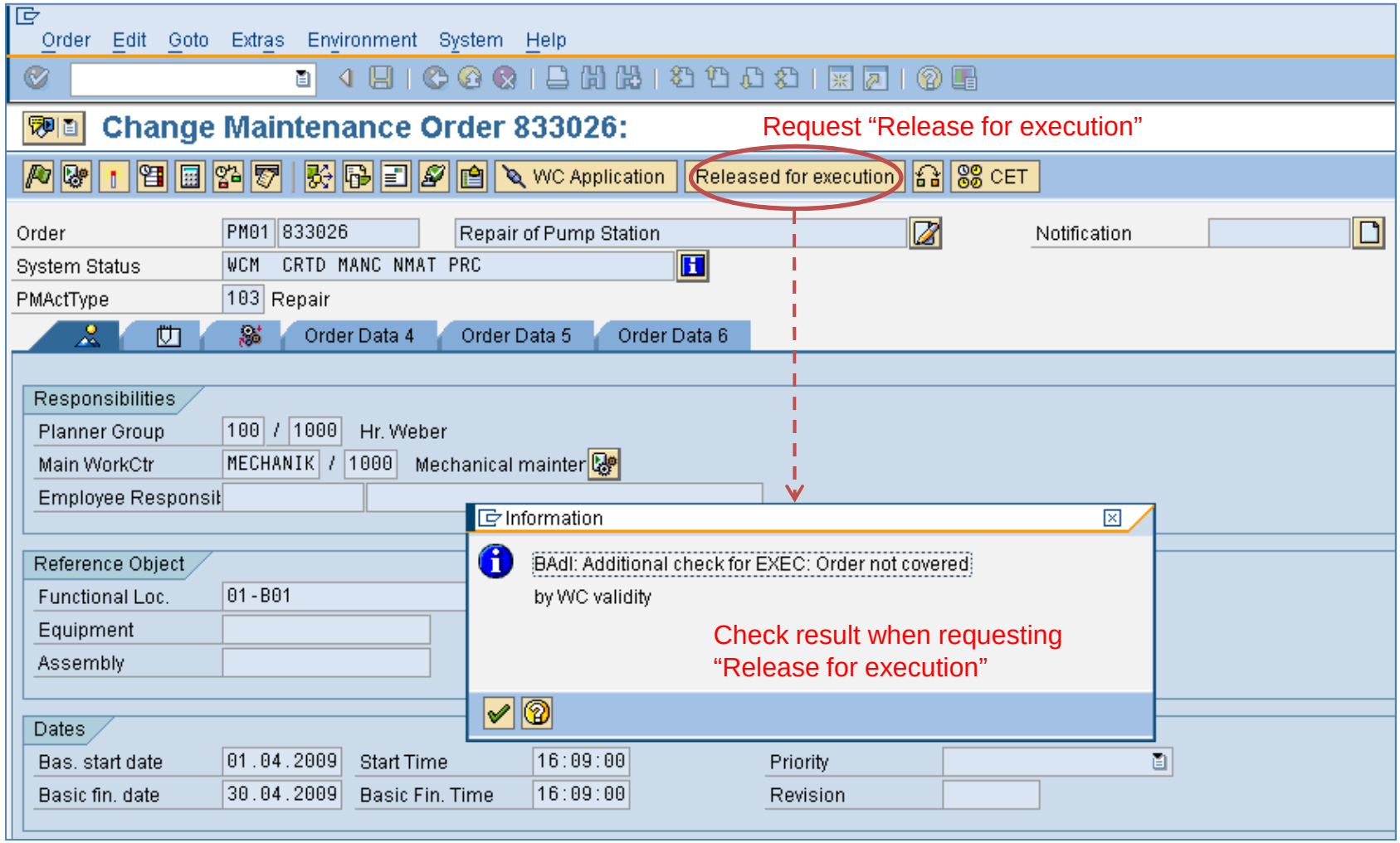
- Methods for maintaining WCM-relevant operations:
  - Deactivate function codes in the operation dialog
  - Check if approval can be valuated as requested (“Yes”, “No”)
  - Check if the operation dialog can be closed as requested (by the green check)

- Methods for work release/completion of WCM-relevant operations:
  - Check if “Release for Execution” can be issued as requested
  - Check if “Work Completed” can be confirmed as requested

5.1.1) Example
Check if Release for Execution can be issued as requested (1/2)



5.1.1) Example
Check if Release for Execution can be issued as requested (2/2)



5.1.2) Process-controlling BAdI for all WCM Objects General Functions

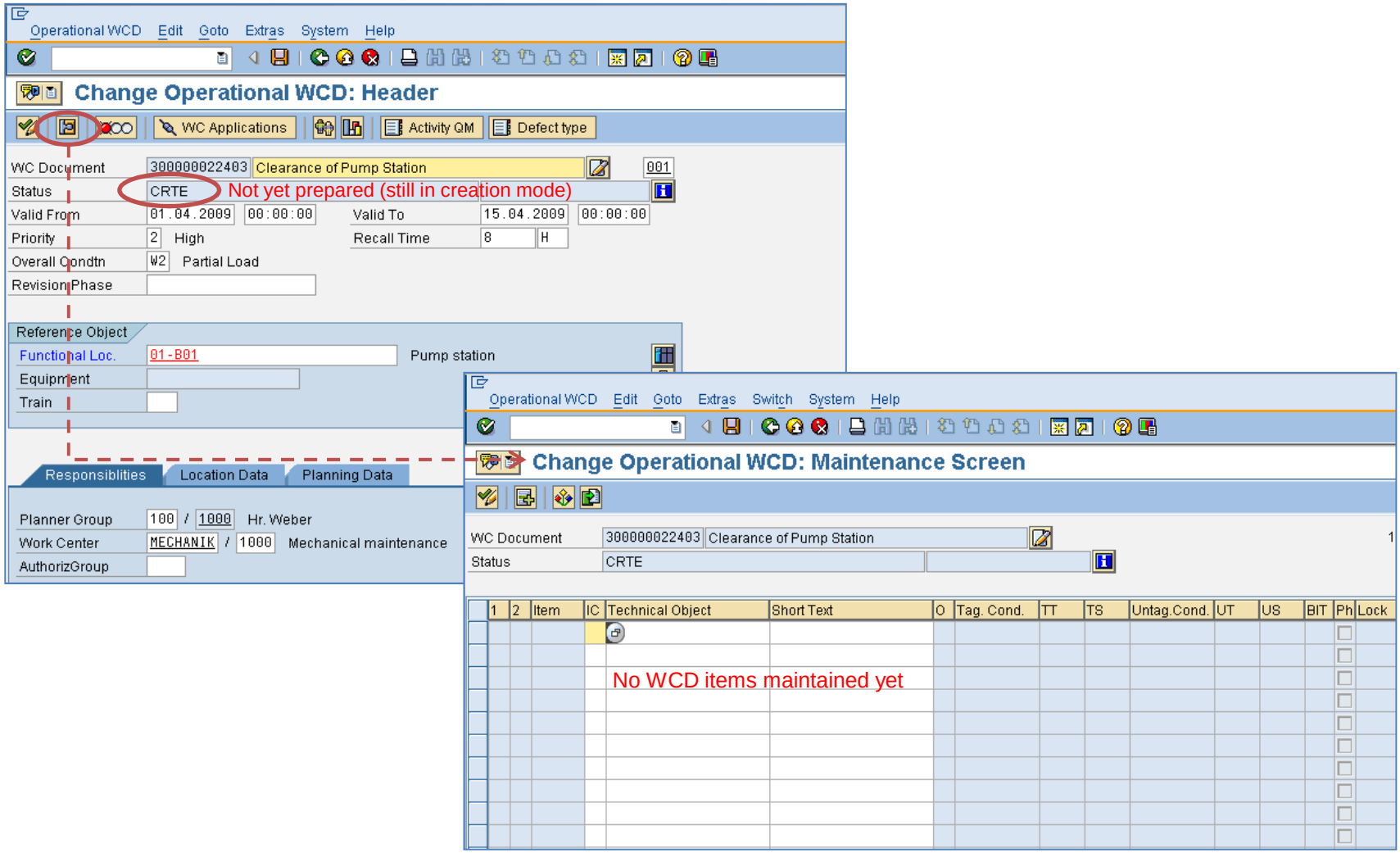
- General BAdI methods for WCM objects:
  - Change color or short text (e.g. in multi-level list)
  - Set user field for influencing standard field selection
  - Set change indicator for WCM object (e.g. for confirmation prompt upon exit)
  - Check WCM object in case of requested save

- Methods for basic functions of WCM objects:
  - Check if copy template can be used when creating WCM object (with template)
  - Set planning data of WCM object (start/end of basic, scheduled & actual dates)
  - Deactivate function codes for the WCM object being processed
  - Check requested print of WCM object header data
  - Check requested print of work permit

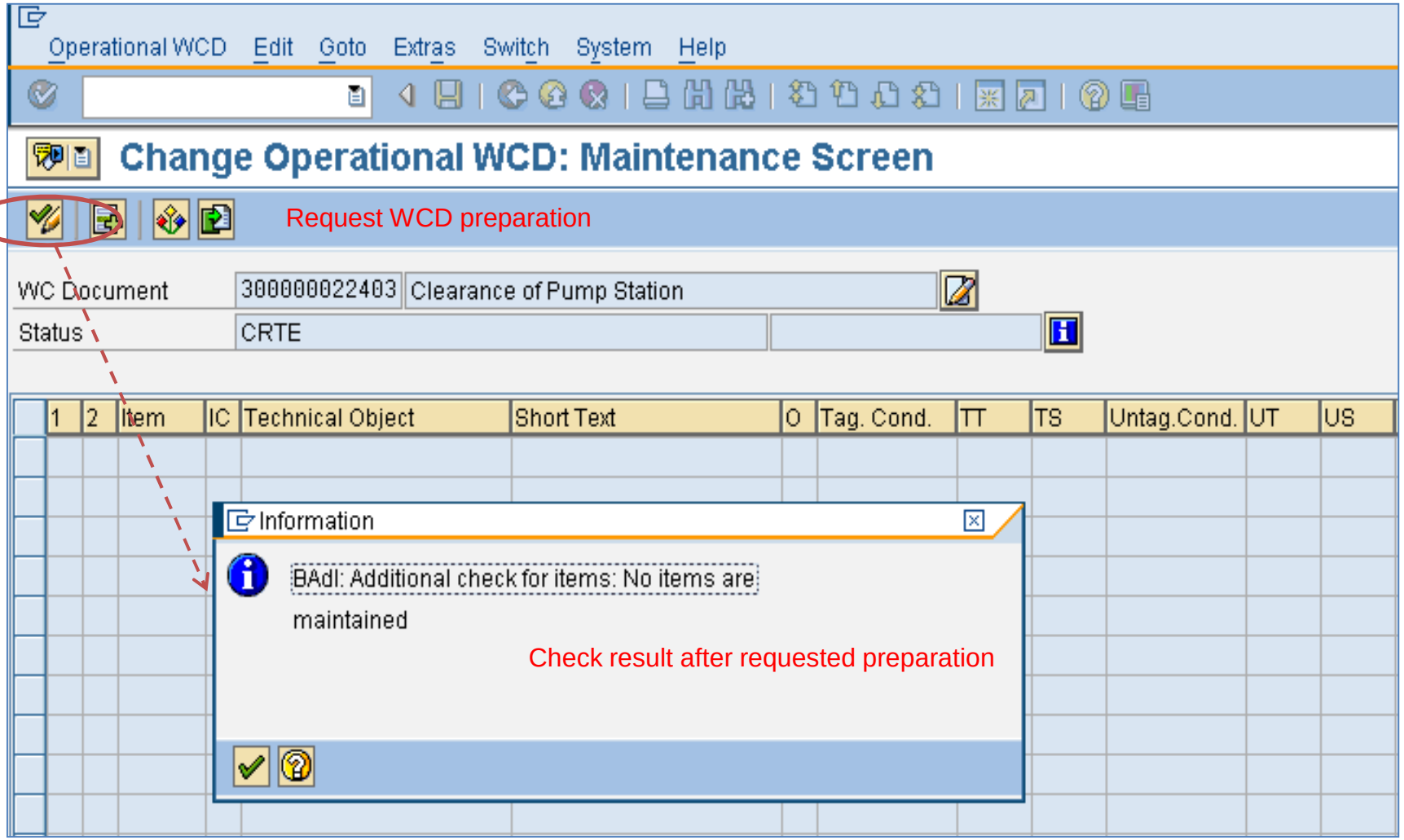
5.1.2) Process-controlling BAdI for all WCM Objects Status Changes

- Methods for status changes of WCM objects:
  - Check if the status of the WCM object can be changed as requested. The following status changes can be checked:
    - Preparation and change mode
    - Completion and rejection
    - Set and reset inactivation flag
    - Set and reset deletion flag

5.1.2) Example
Check if Operational WCD can be prepared as requested (1/2)



5.1.2) Example
Check if Operational WCD can be prepared as requested (2/2)



5.1.3) Process-controlling BAdI for WCAs

- Methods for the test cycle controlled by WCAs:
  - Check if test cycle can be permitted as requested
  - Check if existing permit for test cycle can be revoked as requested
  - Rename the functions permitting a test cycle and revoking the permit

5.1.4) Process-controlling BAdI for Operational WCDs General Functions

- General methods for Operational WCD items:
  - Change logical destination of RFC server connecting a graphical system
  - Revise data imported from graphical system
  - Revise technical objects selected via multiple selection
  - Input help and input check for lock (physical blocking)
  - Set user field for influencing standard field selection
  - Default sort sequence of the items of an Operational WCD on the maintenance screen as well as on the switching screen
  - Extend the standard simulation on the maintenance screen (→ internal checks) and on the switching screen (→ external checks) by individual checks
  - Rename the functions (pushbuttons, tooltips) within the operational cycle

- Methods for basic functions of Operational WCDs:
  - Check requested print of a tagging list or an untagging list for selected items
  - Check requested print of tags or test tags for selected items
  - Check if the WCD is untaggable

5.1.4) Process-controlling BAdI for Operational WCDs Status Changes

- Methods for status changes of Operational WCD items:
  - Check if the status of a WCD item can be changed as requested. The following status changes can be checked:
    - Inactivation
    - Set and reset of operational protection
    - Setting of all (applicable) status of the operational cycle, i.e.:
      - “Tag” (:= to be tagged)
      - “Tag Printed”
      - “Tagged”
      - "Untag Temporarily“ (:= to be temporarily untagged)
      - "Test Tag Printed“
      - "Temporarily Untagged“
      - "Untag“ (:= to be untagged)
      - "Untagged“

5.1.5) Process-controlling BAdI for WCD Templates

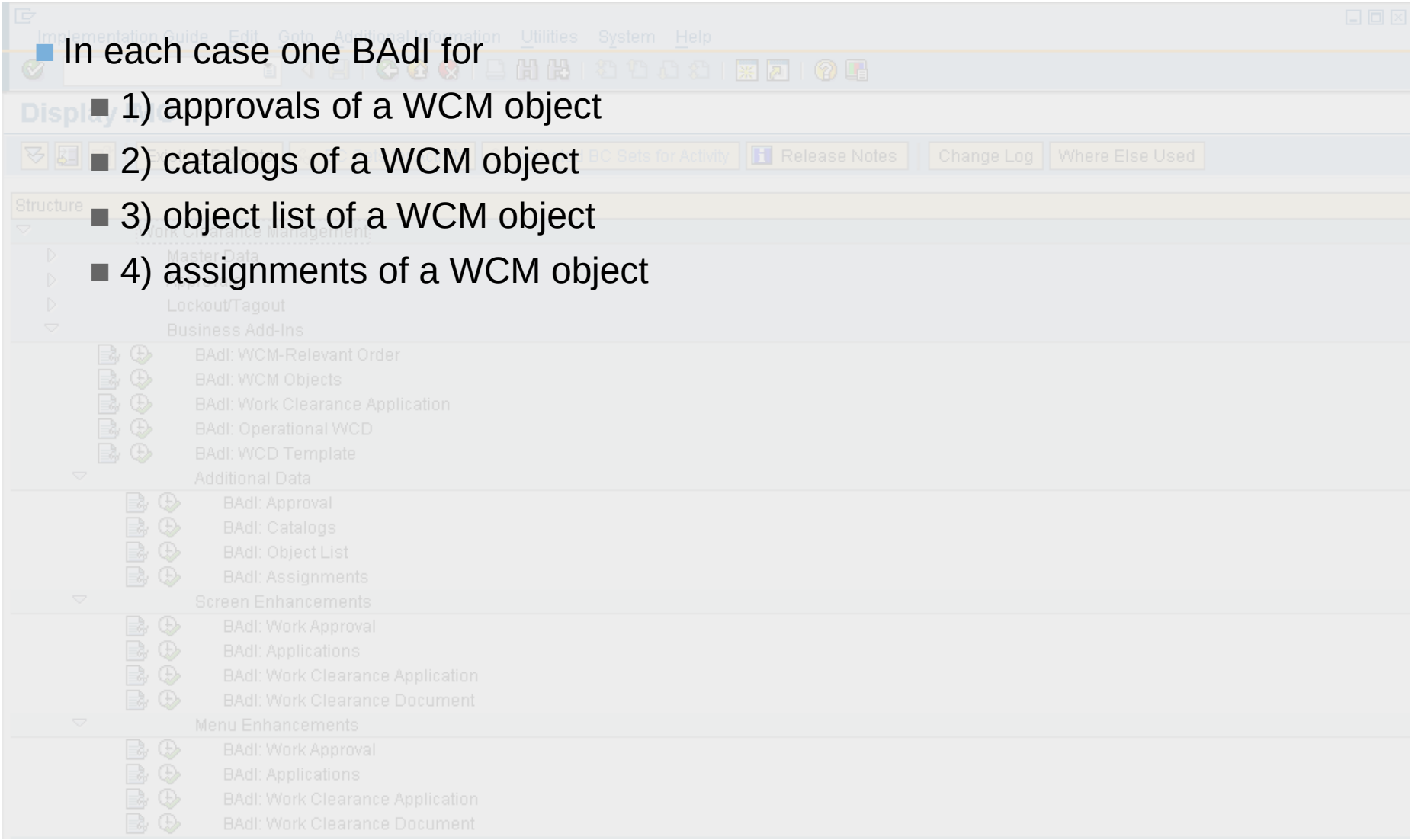
- Methods for WCD templates:
  - Default sort sequence of the items of a WCD template on the maintenance screen
  - Extend the standard simulation on the maintenance screen by individual checks
  - Check if the status of a WCD item can be set to “Inactive” as requested

5) Appendix: Detailed View on the different WCM BAdIs

- 5.1. Process-controlling BAdIs for Order and single WCM Objects (EhP3)
- **5.2. BAdIs for Additional Data of single WCM Objects (EhP3)**
- 5.3. BAdIs for Screen Enhancements of single WCM Objects (EhP3)
- 5.4. BAdIs for Menu Enhancements of single WCM Objects (EhP3)
- 5.5. BAdIs for WCM List Selection Output (EhP5)
- 5.6. BAdIs for Menu Enhancements of WCM List Processing (EhP5)

5.2) BAdIs for additional data of single WCM Objects

- In each case one BAdI for
  - 1) approvals of a WCM object
  - 2) catalogs of a WCM object
  - 3) object list of a WCM object
  - 4) assignments of a WCM object



5.2.1) BAdI for approvals of a WCM Object

- The BAdI provides the following implementation options:
  - Deactivate functions for approval assignment
  - Check if the assignment of an approval can be removed as requested
  - Check if the assigned approval can be issued as requested
  - Check if the issue of an assigned approval can be revoked as requested
  - Check if the dialog window for approval assignment can be closed as requested

5.2.2) BAdI for Catalogs of a WCM Object

- The BAdI provides the following implementation options:
  - Deactivate functions for catalog maintenance
  - Default sort sequence for the maintained entries of the assigned catalog
  - Check if a maintained entry for the assigned catalog can be removed as requested
  - Check if a maintained entry for the assigned catalog can be valuated as requested
  - Check if the dialog window for catalog maintenance can be closed as requested

5.2 3) BAdI for Object List of a WCM Object

- The BAdI provides the following implementation options:
  - Deactivate functions for object list maintenance
  - Default sort sequence for the maintained entries of the object list
  - Check if a maintained entry of the object list can be removed as requested
  - Check if the dialog window for object list maintenance can be closed as requested

5.2.4) BAdI for (superior/subordinate) Assignments of a WCM Object

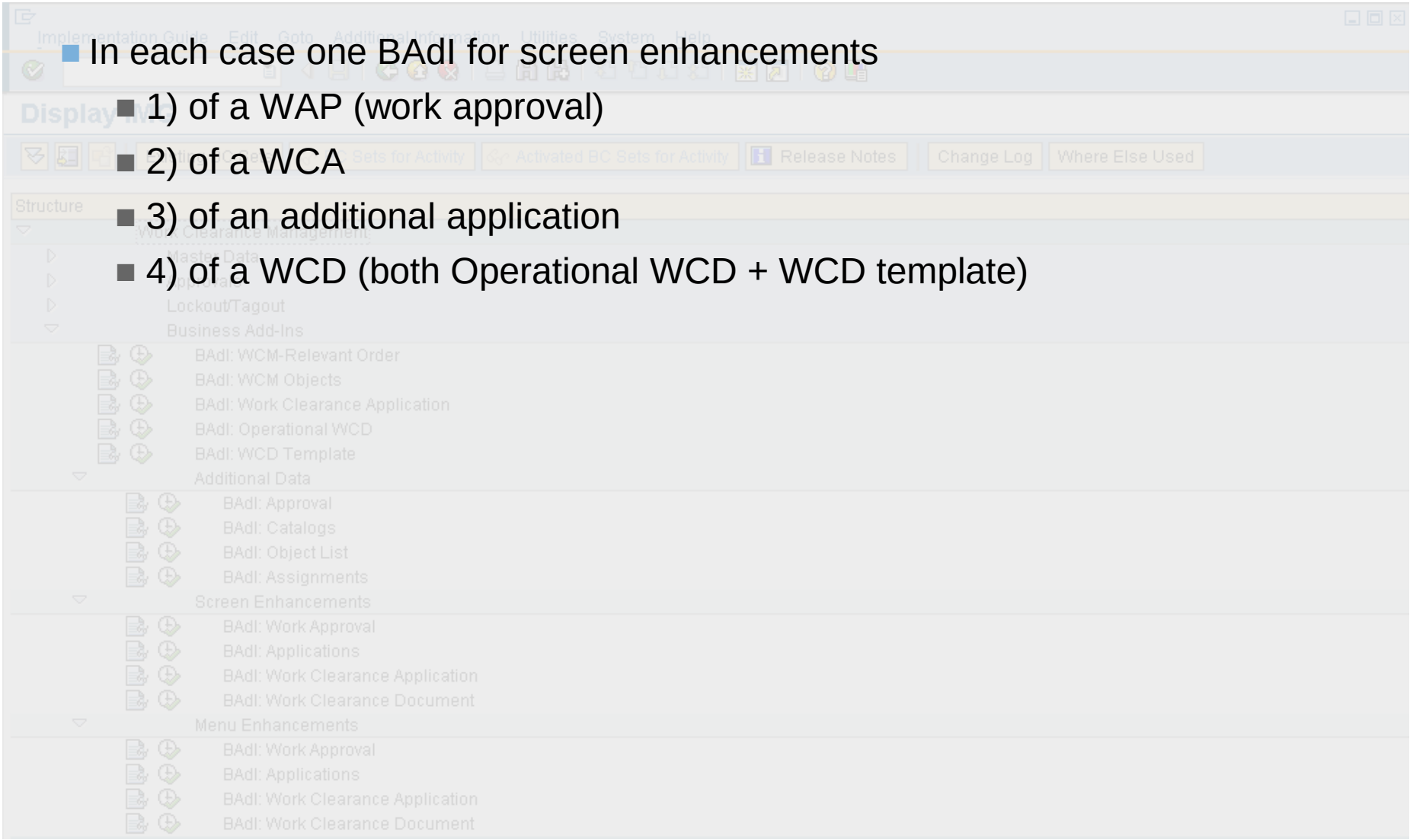
- The BAdI provides the following implementation options:
  - Deactivate functions for maintaining assignments
  - Check if a maintained assignment can be removed as requested
  - Check if the dialog window for maintaining superior and subordinate assignments can be closed as requested
  - Restrict the set of applicable objects for an assignment
    - Example (Enhanced Model): Restriction of work approvals when assignment is requested during maintenance of an order (subordinate assignment) or of an application (superior assignment)

5) Appendix: Detailed View on the different WCM BAdIs

- 5.1. Process-controlling BAdIs for Order and single WCM Objects (EhP3)
- 5.2. BAdIs for Additional Data of single WCM Objects (EhP3)
- **5.3. BAdIs for Screen Enhancements of single WCM Objects (EhP3)**
- 5.4. BAdIs for Menu Enhancements of single WCM Objects (EhP3)
- 5.5. BAdIs for WCM List Selection Output (EhP5)
- 5.6. BAdIs for Menu Enhancements of WCM List Processing (EhP5)

5.3) BAdIs for Screen Enhancements of single WCM Objects

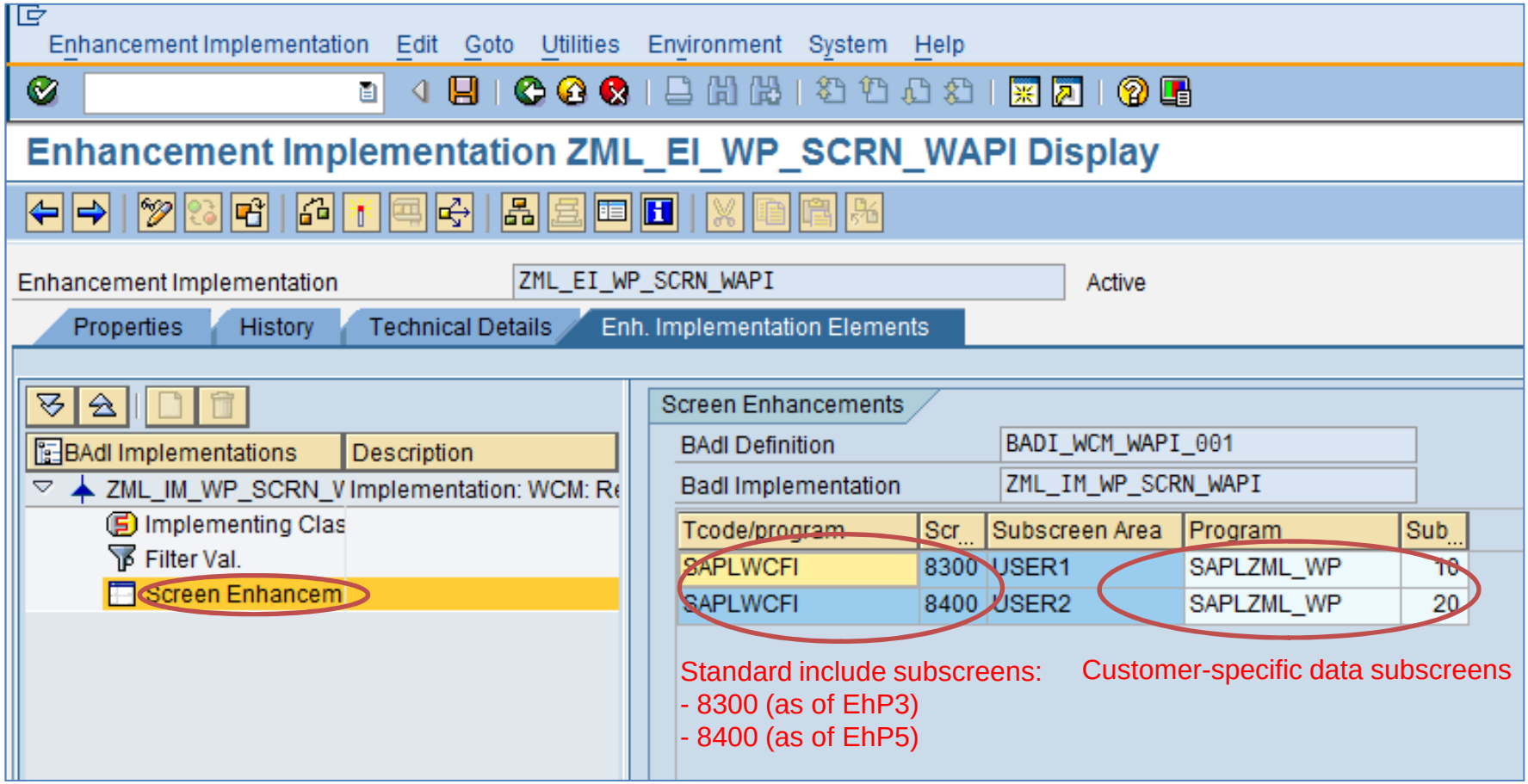
- In each case one BAdI for screen enhancements
  - 1) of a WAP (work approval)
  - 2) of a WCA
  - 3) of an additional application
  - 4) of a WCD (both Operational WCD + WCD template)



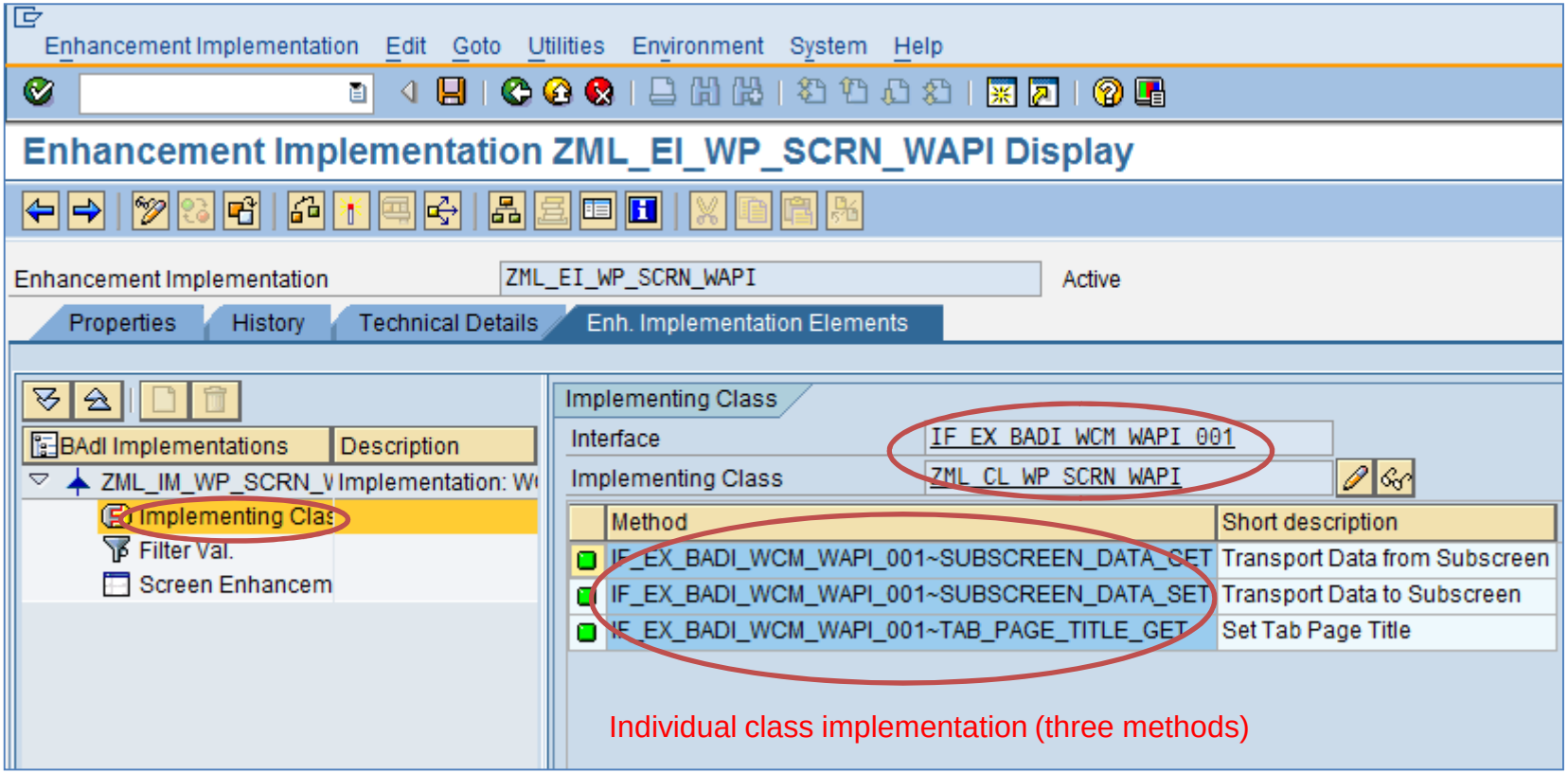
5.3) BAdIs for Screen Enhancements per WCM Object

- Screen BAdIs are built identically for all WCM objects. They consist of two basic parts:
  - The screen area provided for the screen enhancement of a WCM object.
    - In EhP3 + EhP4, SAP provides one screen enhancement area, integrated as 4th tab (next to Responsibilities, Location Data, Planning Data) on the header screen of a WCM object. On this tab, one customer-specific include subscreen is embedded.
    - As of EhP5, view profiles for WCM objects will support a flexible arrangement of all header subscreens, including two screen enhancement areas, each of them containing a customer-specific include subscreen.
  - Furthermore, screen enhancement of a WCM object requires implementing the following interface methods of the underlying BAdI:
    - SUBSCREEN_DATA_GET: Data communication from the customer-specific include screen to the outside (→ PAI)
    - SUBSCREEN_DATA_SET: Data communication from outside to the customer-specific include screen (→ PBO)
    - TAB_PAGE_TITLE_GET: Set title for the 4th tab on the header screen

5.3) Screen Enhancement Implementation: Screen Area



5.3) Screen Enhancement Implementation: Class Interface

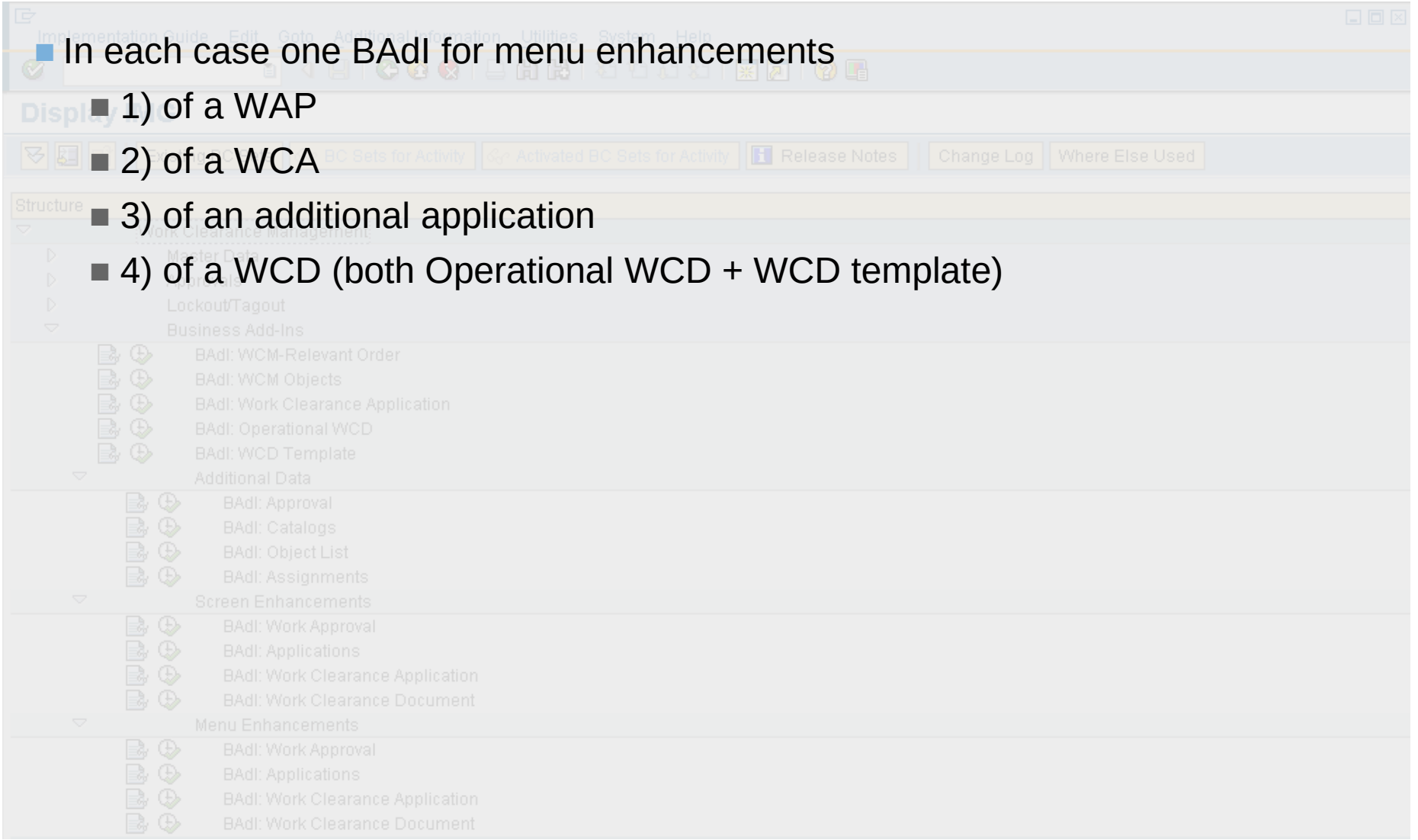


5) Appendix: Detailed View on the different WCM BAdIs

- 5.1. Process-controlling BAdIs for Order and single WCM Objects (EhP3)
- 5.2. BAdIs for Additional Data of single WCM Objects (EhP3)
- 5.3. BAdIs for Screen Enhancements of single WCM Objects (EhP3)
- **5.4. BAdIs for Menu Enhancements of single WCM Objects (EhP3)**
- 5.5. BAdIs for WCM List Selection Output (EhP5)
- 5.6. BAdIs for Menu Enhancements of WCM List Processing (EhP5)

5.4) BAdIs for Menu Enhancements of single WCM Objects

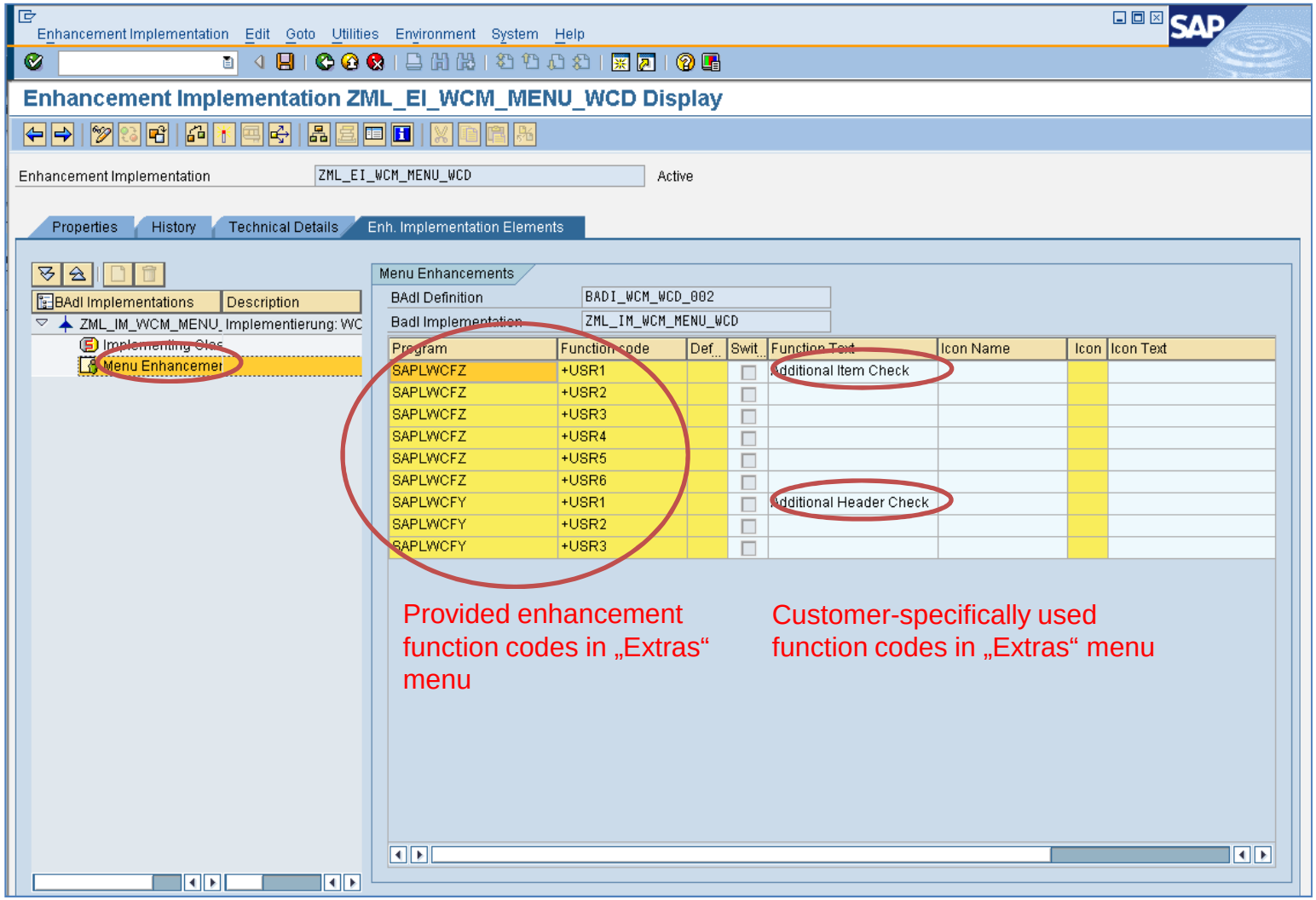
- In each case one BAdI for menu enhancements
  - 1) of a WAP
  - 2) of a WCA
  - 3) of an additional application
  - 4) of a WCD (both Operational WCD + WCD template)



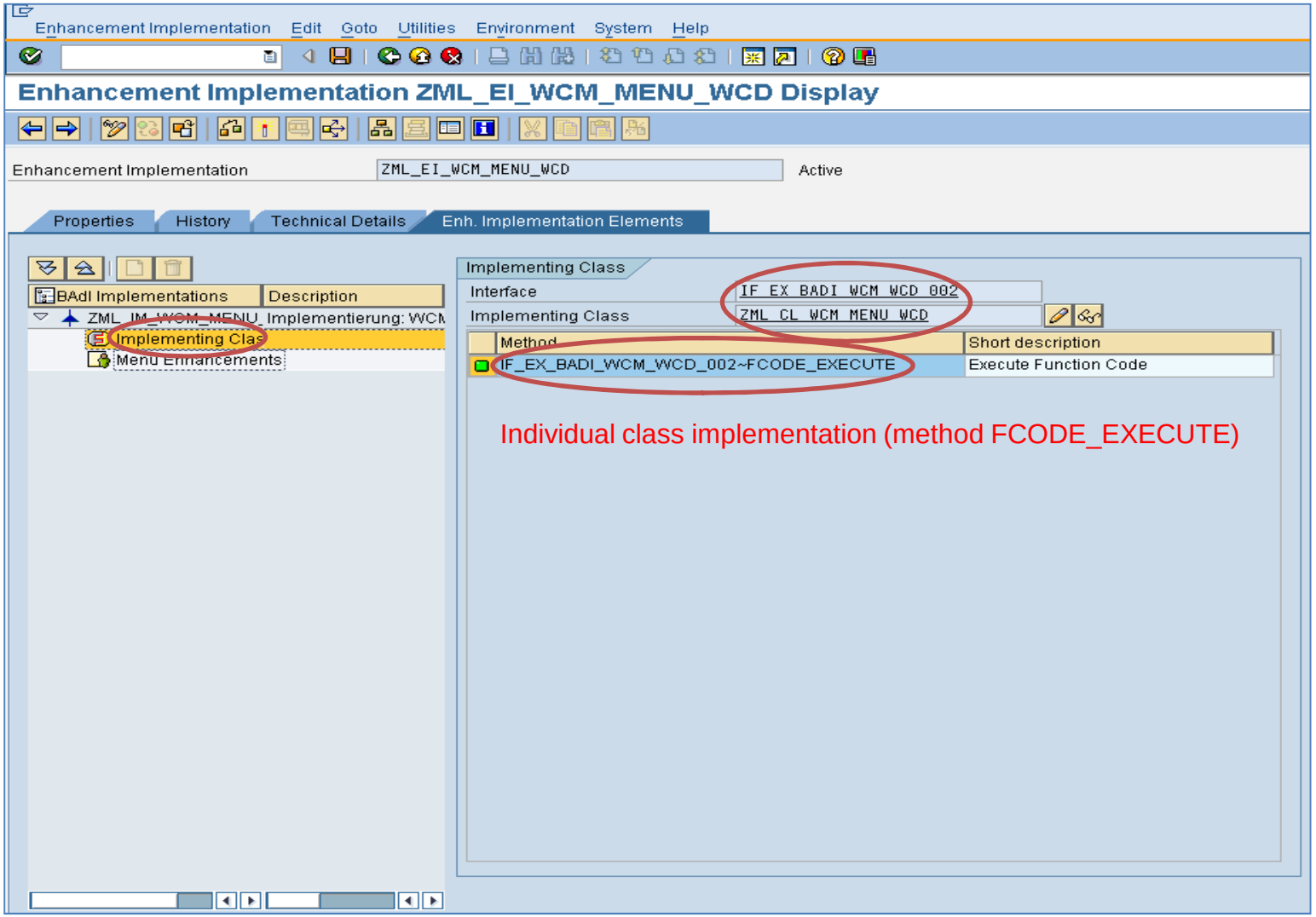
5.4) BAdIs for Menu Enhancements per WCM Object

- Menu BAdIs are built identically for all WCM objects. They consist of two basic parts:
  - The function codes provided for the menu enhancement of a WCM object are integrated in the “Extras” menu on the header screen of a WCM object as well as on its item screens (→ maintenance screen, switching screen).
    - On the header screen it is possible to define up to three customer-specific function codes.
    - On the item screens it is possible to define up to six customer-specific function codes.
  - Furthermore, menu enhancement of a WCM object requires implementing the following interface method of the underlying BAdI:
    - FCODE_EXECUTE: Execute a customer-specific function code

5.4) Menu Enhancement Implementation: Function Codes



5.4) Menu Enhancement Implementation: Class Interface



5) Appendix: Detailed View on the different WCM BAdIs

- 5.1. Process-controlling BAdIs for Order and single WCM Objects (EhP3)
- 5.2. BAdIs for Additional Data of single WCM Objects (EhP3)
- 5.3. BAdIs for Screen Enhancements of single WCM Objects (EhP3)
- 5.4. BAdIs for Menu Enhancements of single WCM Objects (EhP3)
- **5.5. BAdIs for WCM List Selection Output (EhP5)**
- 5.6. BAdIs for Menu Enhancements of WCM List Processing (EhP5)

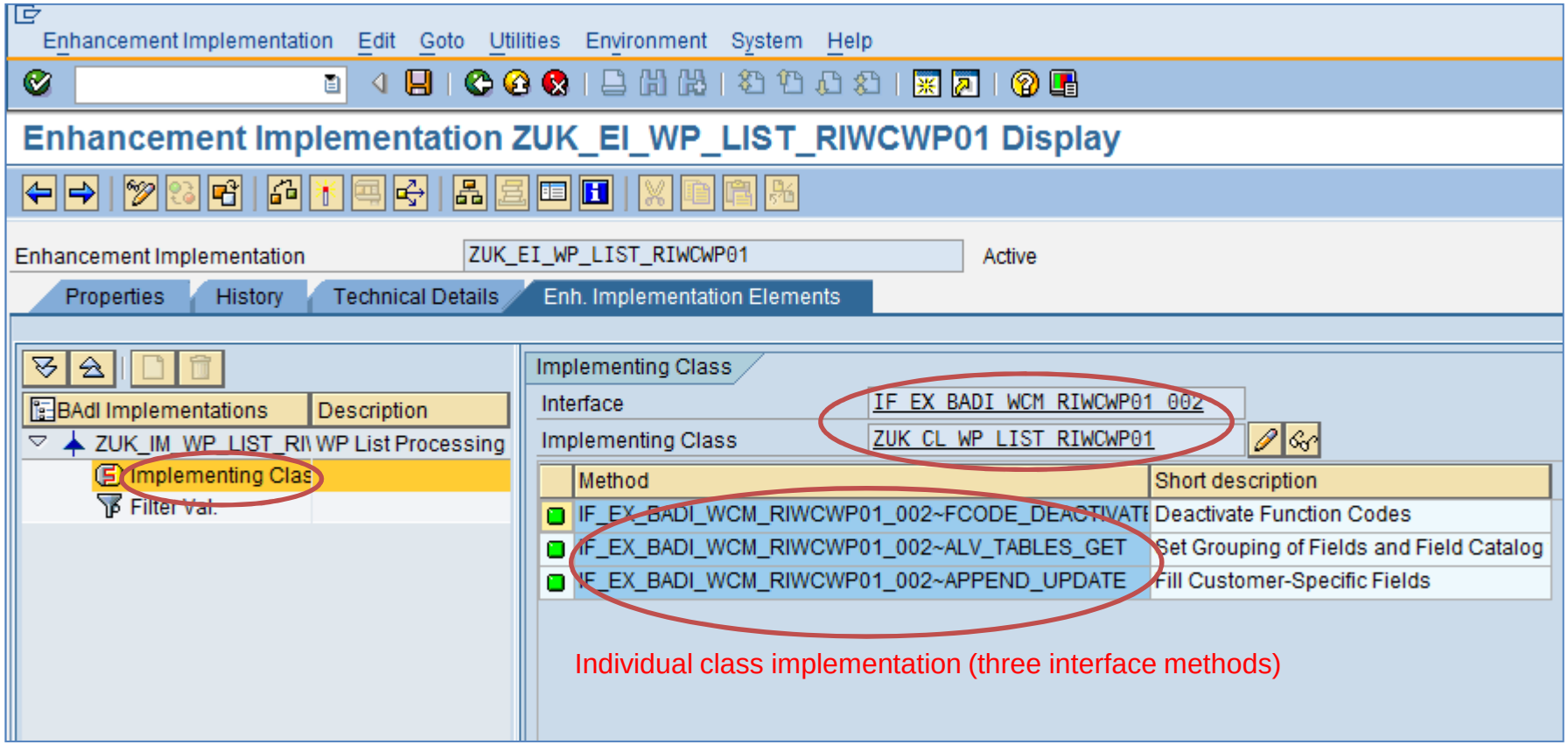
5.5) BAdIs for WCM List Selection Output

- In each case one BAdI for native list processing output
  - 1) of WAPs
  - 2) of applications
  - 3) of WCDs (both Operational WCD + WCD template)
  - 4) of operational lists
  - 5) of lineups
- In each case one BAdI for list selection output during assignment
  - 1) of orders
  - 2) of WAPs
  - 3) of applications
  - 4) of Operational WCDs
- One BAdI for WCD list selection output within clipboard
- One BAdI for the combined work permit/order list selection output
- One BAdI for multilevel list selection output

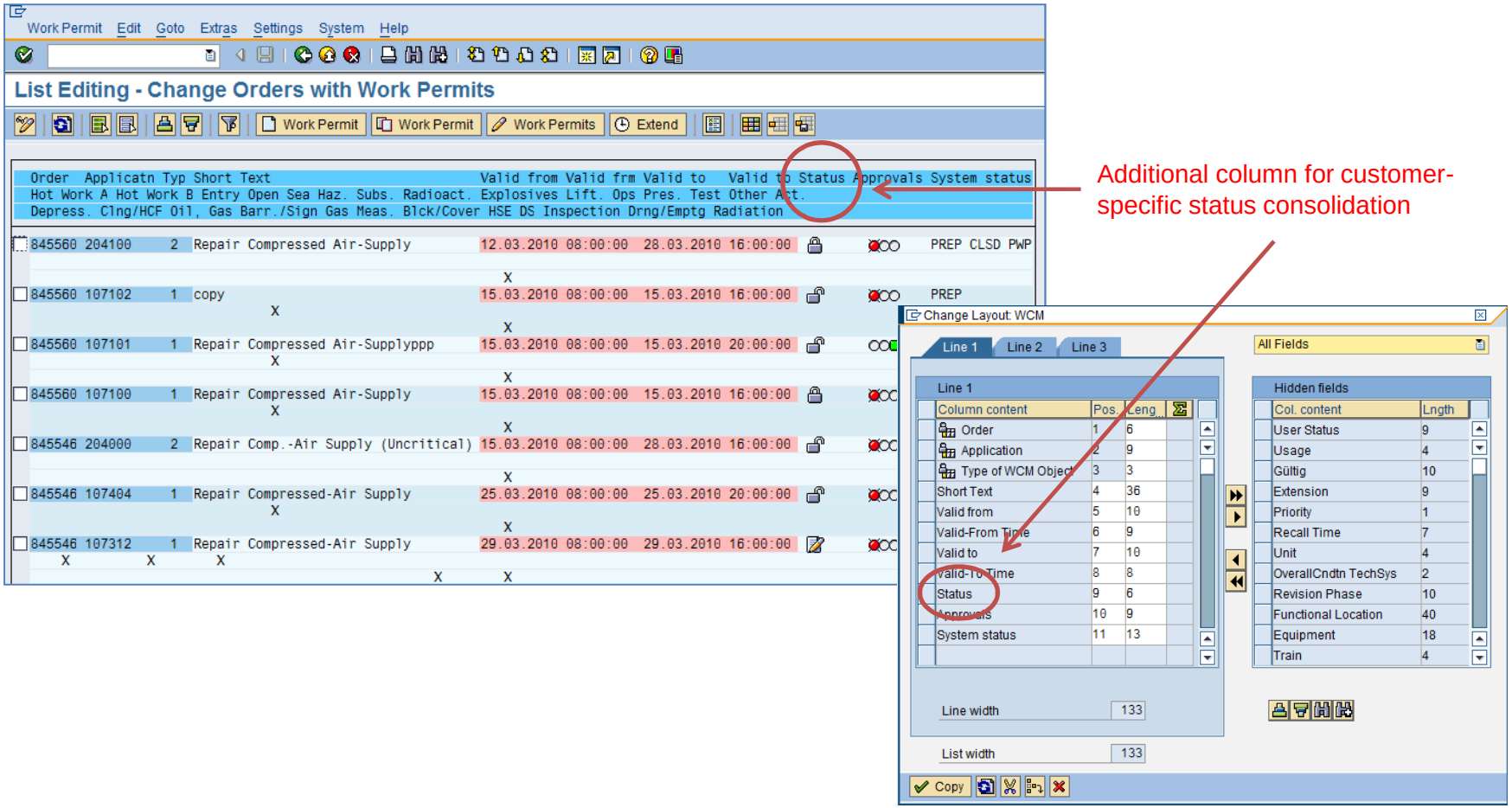
5.5) BAdIs for WCM List Selection Output

- BAdIs for the list selection output are built (nearly) identical for all WCM objects:
  - All BAdIs (except the one for the multilevel list) provide the following interface methods:
    - ALV_TABLES_GET: Set ALV field catalog and grouping of fields for the result list
    - APPEND_UPDATE: Fill customer-specific fields in the result list
    - FCODE_DEACTIVATE: Deactivate function codes for the result list
  - The BAdI for the multilevel list provides the following interface method:
    - FCODE_DEACTIVATE: Deactivate function codes for the result list

5.5) BAdI Implementation for List Selection Output: Class Interface



5.5) BAdI Implementation for List Selection Output: Example



- Method ALV_TABLES_GET adds customer-specific field to ALV field catalog.
- Method APPEND_UPDATE calculates the field value (not stored on DB).

5) Appendix: Detailed View on the different WCM BAdIs

- 5.1. Process-controlling BAdIs for Order and single WCM Objects (EhP3)
- 5.2. BAdIs for Additional Data of single WCM Objects (EhP3)
- 5.3. BAdIs for Screen Enhancements of single WCM Objects (EhP3)
- 5.4. BAdIs for Menu Enhancements of single WCM Objects (EhP3)
- 5.5. BAdIs for WCM List Selection Output (EhP5)
- **5.6. BAdIs for Menu Enhancements of WCM List Processing (EhP5)**

5.6) BAdIs for Menu Enhancements of WCM List Processing

- In each case one BAdI for menu enhancements of native list processing output
  - 1) of WAPs
  - 2) of applications
  - 3) of WCDs (both Operational WCD + WCD template)
  - 4) of operational lists
  - 5) of lineups
- In each case one BAdI for menu enhancements of list processing output during assignment
  - 1) of orders
  - 2) of WAPs
  - 3) of applications
  - 4) of Operational WCDs
- One BAdI for menu enhancements of WCD list processing output within clipboard
- One BAdI for menu enhancements of the combined work permit/order list processing output
- One BAdI for menu enhancements of multilevel list processing output

5.6) BAdIs for Menu Enhancements of WCM List Processing

- BAdIs for menu enhancements of WCM list processing are built identically for all WCM objects. They consist of two basic parts:
  - The function codes provided for the menu enhancement of WCM list processing are integrated in the “Extras” menu on the result list. It is possible to define up to three customer-specific function codes.
  - Furthermore, menu enhancement of a WCM object requires implementing the following interface method of the underlying BAdI:
    - FCODE_EXECUTE: Execute a customer-specific function
- Regarding design and methods to be implemented, menu enhancements of WCM list processing is identical to menu enhancements of single WCM objects.

Copyright 2010 WCM GmbH