



Guideline

BPEM Business Process Exception Management

SAP SE
Dietmar-Hopp-Allee 16
D-69190 Walldorf

Document Version 2.3 – July 2016, Public

Copyright

© 2016 SAP SE.

All rights reserved. SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP Business ByDesign, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

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

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

Disclaimer

This document is not part of the official SAP documentation and is subject to change without notice. It provides examples on how to use, configure and implement certain functions. Any use of these examples as a basis for your own implementation is at your own risk and we strongly advise you to make backup copies prior to any implementation and to carefully check whether and to what extent the examples given can be applied to your system landscape and system configuration.

SAP AG and its affiliated companies ("SAP Group") provide this document for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the information contained therein. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

Document History

Version	Date	Comment
1.9	April 2013	1 st document version published
2.0	May 2013	Minor correction
2.3	July 2016	New chapter about BPEM in the Utilities Industry New chapter: Analysis of Job Logs in Background New chapter <code>EMMACAP</code> transaction New chapter New Functions in ERP 6.0 EhP7 New BADIs and enhancements to existing BADIs during clarification case creation and maintenance List of important SAP Notes

Typographic Conventions

Type Style	Description
Example Text	Words or characters quoted from the screen. These include field names, screen titles, pushbutton labels, menu names, menu paths, and menu options. Cross-references to other documentation
<i>Example text</i>	Emphasized words or phrases in body text, graphic titles, and table titles
EXAMPLE TEXT	Technical names of system objects. These include report names, program names, transaction codes, table names, and key concepts of a programming language when they are surrounded by body text, for example, <code>SELECT</code> and <code>INCLUDE</code> .
Example text	Output on the screen. This includes file and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
Example text	Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.
<Example text>	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.

Symbols

Symbols	Meaning
	Caution
	Example
	Note

Motivation

The guideline **Business Process Exception Management BPEM** is aimed at customers, SAP consultants and partners who are responsible for configuring and executing the **Business Process Exception Management** framework. This document supplements the available setup guides, **SAP Online Library Documentation** and release notes.

Applications create error messages when a problem occurs in the processing of dialog transactions and massive batch processing and store them in application logs. In particular during the nightly batch processing huge application logs are created and require often manual analysis.

This effort is no longer required since **Business Process Exception Management** transforms all error messages belonging to one business object into an own clarification case. Clarification cases provide agents not only with the error messages but also with standard business processes for solving the problem root cause quickly and efficiently. Clarification case worklists are used to forward clarification cases to business departments and agents with the right skills and responsibility to resolve and complete their clarification cases.

Business Process Exception Management enables you to centrally analyze and monitor mass activities and dialog transactions. For this, you can use either the integrated monitor on your SAP ERP system. In addition, you can extract information, such as key performance indicators, to a connected SAP BW system to effectively monitor and control not only the current processing status of the created clarification cases but also the overall stability of your business processes. The collected data on the SAP BW system supports you in identifying and, subsequently, resolving the root causes to prevent the same problems or business exceptions to recur in the future.

The purpose of the document is to give guidance and to share experience for an efficient implementation of the **Business Process Exception Management** framework to allow a quick ROI, such as:

- Customizing of the **Business Process Exception Management** framework
- Working with clarification cases and clarification worklists
- Implementing Enhancements

This document has the following structure:

Chapter 1 introduces BPEM and gives an overview of the main functions

Chapter 2 describes the business architecture of BPEM and an implementation approach

Chapter 3 describes how to setup and customize BPEM

Chapter 4 describes all BPEM transactions in details

Chapter 5 provides technical insight into BPEM

Chapter 6 describes use cases and their implementation

Chapter 7 describes the BPEM related extraction to SAP BW

Chapter 8 describes the archiving of BPEM related transaction data

Chapter 9 describes the utility specific enhancements of BPEM

Chapter 10 gives an overview of the new functions available starting with ERP 6.0 EhP 7

Appendix A provides sample ABAP code for use cases

The **Online Documentation of BP EM** can be found at the SAP Help Portal <http://help.sap.com/> → *SAP for Industries* → *Select SAP Contract Accounts Receivable and Payable* → *Product Version Enhancement Package x for SAP ERP 6.0* → *Application Help SAP Library available in English or German* → *Enhanced Message Management*.

The **Online Documentation regarding the integration of BP EM with SAP BW** can be found at the SAP Help Portal <http://help.sap.com/> → *SAP NetWeaver* → *SAP NetWeaver Business Information Warehouse* → *Select BI Content* (select your SAP NetWeaver BI Content version) → *Application Help available in English or German* → *Select BI Content* → *Industry Solutions* → *Utilities* → *Process Management*.

This revised version of this document completes the previously published document with a description of the latest development and enhancements in BP EM as well as further use cases and sample coding examples for an easy implementation of BP EM.

Author

Friedrich Keller, SAP AG

Feedback

We would very much appreciate your feedback on this document. Please send your comments to <mailto:friedrich.keller@sap.com>.

Table of Contents

1	Introduction	10
1.1	Overview	10
1.2	History and Naming Convention of BPEM	11
1.3	Main Functions of BPEM	12
1.3.1	Monitoring of Business Processes	12
1.3.2	Management of Business Exceptions	13
1.3.3	Main Benefits of using BPEM	13
2	Business Architecture and Data Flow	14
2.1	Glossary of BPEM Terms	14
2.1.1	Business Process Area	14
2.1.2	Business Process Code	14
2.1.3	Primary Object Type	14
2.1.4	Clarification Case	14
2.1.5	Clarification Case Category	14
2.1.6	Clarification Case Type	14
2.1.7	Message Pool of a Clarification Case Category	15
2.1.8	Business Object Repository BOR and BAPIs	15
2.1.9	Container and Data Flow in a Clarification Case Category	15
2.1.10	Binding Editor	15
2.1.11	Condition Editor	15
2.1.12	BPEM Job	16
2.1.13	Mass Activity	16
2.1.14	Time Stamp	16
2.1.15	SAP Enhancement Framework and BAdI	16
2.1.16	SPA/GPA Parameters	17
2.1.17	ABAP List Viewer ALV	17
2.2	Overview BPEM Processes and Transactions	18
2.2.1	Step 1: BPEM Job Preparation	19
2.2.2	Step 2: Clarification Case Generation	19
2.2.3	Step 3: Distribution of Clarification Cases	19
2.2.4	Step 4: Resolution and Completion of Clarification Cases	20
2.2.5	Step 5: Transfer of KPIs to Business Information Warehouse	20
2.3	Implementation Approach	20
2.3.1	Getting Started with an Enhanced Message List	21
2.3.2	Getting Started with Clarification Cases	21
3	Customizing	25
3.1	Implementation Guide – General	25
3.2	Authorization	25
3.2.1	Authorization for BPEM Customizing	25
3.2.2	Authorization for BPEM Job Processing	26
3.2.3	Authorization for Clarification Case Processing	26
3.3	Specifications for Logging	27
3.3.1	Maintain Internal Number Range Intervals for Jobs	27
3.3.2	Activate Business Process Areas for Message Management	28
3.3.3	Activate Business Processes for Message Management	29
3.3.4	Suppression of Messages for Log Preparation	30
3.4	Specifications for Generating Clarification Cases	32

3.4.1	Maintain Number Range Intervals for Clarification Cases.....	32
3.4.2	Define Clarification Case Types	32
3.4.3	Create and Maintain Clarification Case Categories	33
3.4.4	Define Method for Forwarding Clarification Cases.....	49
3.5	Specifications for Processing Clarification Cases	49
3.5.1	Generate Selection Screen for Clarification List	49
3.5.2	Define Reasons for Forwarding Clarification Cases.....	49
3.5.3	Define Reasons for Reversing Clarification Cases	50
3.5.4	Maintain Shortcut Keys for Clarification Case Processing	50
3.6	Specifications for Business Processes and Messages.....	52
3.6.1	Define Business Process Areas for Message Management	53
3.6.2	Define Business Processes for Message Management.....	53
3.6.3	Define Business Processes for Processes from CIC	56
3.6.4	Define Transactions for Message Management	56
3.6.5	Define Business Objects for Customer-Defined Messages.....	57
3.7	Process Statistics (Utilities Industry)	58
3.7.1	IMG – Integration with BW (Utilities Industry)	58
3.7.2	Activate Business Process Areas for Message Management	59
3.7.3	Define Suppression of Messages for Log Preparation.....	59
3.7.4	Define Success Messages for the Extraction (SAP)	59
3.7.5	Define Success Messages for the Extraction (Customer).....	59
3.7.6	Define Business Process Areas for Message Management	59
3.7.7	Define Business Processes for Message Management.....	59
3.7.8	Define Business Processes for Processes from CIC	59
4	Dialog and Batch Transactions.....	60
4.1	Overview.....	60
4.2	Job Preparation and Automatic Clarification Case Processing	60
4.2.1	Transaction FPEMMAMA - <i>BPEM Mass Activity</i>	61
4.2.2	Transaction EMMAJP - <i>Process Jobs</i>	64
4.2.3	Transaction EMMACAP – <i>Execute Automatic Processes</i>	66
4.2.4	Transaction EMMAJL – <i>Display Jobs</i>	68
4.2.5	Transaction EMMA – <i>Classic EMMA</i>	74
4.3	Working with Clarification Cases.....	75
4.3.1	Transactions EMMACL/EMMACLS – <i>Clarification Case List</i>	75
4.3.2	Transaction EMMAC2 - <i>Clarification Case Maintenance</i>	80
4.3.3	Transaction EMMAC1 – <i>Create Manual Clarification Case</i>	87
5	Technical Implementation Details.....	89
5.1	General Information	89
5.1.1	Table EMMA_HDR – <i>BPEM Header Table</i>	89
5.1.2	Table EMMA_JOB RUNIDMSG – <i>Application Messages</i>	89
5.1.3	Table EMMA_INT – <i>Interval Statistics</i>	89
5.1.4	Table EMMA_CASE – <i>Clarification Case Header Table</i>	90
5.1.5	Table EMMA_CACTOR – <i>Clarification Case Processors</i>	90
5.1.6	Table EMMA_COBJECT – <i>Objects and Field Values for Case</i>	90
5.1.7	Table EMMA_CMSG_LINK – <i>Triggering Messages</i>	90
5.1.8	Table EMMA_CSOLP – <i>Solution Processes</i>	90
5.2	BPEM Job preparation.....	91
5.2.1	Analysis and Grouping of Messages	91

5.2.2	Transaction EMMALOG - <i>Display Messages</i>	92
5.2.3	Reversal of Job Preparation	93
5.2.4	Technical Implementation Job Preparation <i>New EMMA</i>	93
5.2.5	Technical Implementation Job Preparation <i>Classic EMMA</i>	94
5.3	Clarification Case Creation	94
5.3.1	Automatic Creation of Clarification Cases	95
5.3.2	Manual Creation of Clarification Cases.....	97
5.4	Automatic Solution Processes.....	97
5.5	Clarification Case Application Log	98
5.6	Mass Activity FPEMMAMA	98
5.6.1	General Information.....	98
5.6.2	Interval Determination	98
5.6.3	Application Log.....	99
5.7	Business Application Programming Interface	99
5.7.1	BAPI <code>EMMAJob</code>	100
5.7.2	BAPI <code>EMMACase</code>	100
5.8	Customer Enhancements	103
5.8.1	Enhancement Spot <code>EMMA_JOB</code>	103
5.8.2	Enhancement Spot <code>EMMA_CASE</code>	104
5.8.3	Classic BAdI <code>EMMA_CASE</code>	108
5.8.4	ABAP OO Interfaces	114
5.9	SPA/GPA Parameter	116
5.10	Lock Objects	118
6	Use Cases and Sample Implementations.....	119
6.1	Installation-Specific Fields	119
6.2	Installation-Specific Functions	121
6.2.1	Enhancement of the Application Toolbar	121
6.2.2	Control of User-Triggered Functions	123
6.3	Error Message without Clarification Case.....	123
6.3.1	Analysis of application log of <code>FPEMMAMA</code> transaction	124
6.3.2	Process Code Specific Sweeper Clarification Case Categories.....	126
6.3.3	General Sweeper Clarification Case Categories.....	128
6.4	Agent Determination.....	131
6.4.1	Agent Determination Rule	131
6.4.2	Forwarding Rule.....	134
6.4.3	Workflow and Workflow Events	139
6.5	Connect Reports and Interfaces to BPEM.....	146
6.5.1	Connection to BPEM via Application Log	146
6.5.2	Connection to BPEM via BAPI	152
6.6	Connect Mass Activities to BPEM	153
6.6.1	General Implementation	153
6.6.2	Multiple Parallel Processing Objects	155
6.7	Evaluation of Job Logs.....	157
6.7.1	Dialog Transaction	157
6.7.2	Background Processing	160
6.8	Special Solution Processes	161
6.8.1	Execute Reports and Transactions.....	161
6.8.2	Execute Workflow Tasks	166

6.9	Add Business Objects to a Clarification Case dynamically	171
6.10	Dynamic Clarification Case Text.....	171
6.11	Usage of Action Box of Customer Interaction Center CIC	174
7	Business Information Warehouse BW	177
7.1	Introduction	177
7.2	Process Statistics for Clarification Cases.....	177
7.3	Utilities Industries - Process Statistics	178
7.3.1	Process Statistics <i>Non-Clarification</i> and <i>Clarification</i>	178
7.3.2	Special Customizing.....	178
7.3.3	Technical Details.....	181
7.4	Enhancement of the Extraction Structure	182
8	Data Archiving	186
8.1	EMMA_JOB Archiving Object	186
8.2	EMMA_CASE Archiving Object	187
9	BPEM in the Utilities Industry	188
9.1	Clarification Cases in the SAP CRM IC WebClient (Utilities)	188
9.1.1	Customizing	188
9.1.2	Views in IC WebClient.....	190
9.2	Integration of ECRMREPL in BPEM for Utilities.....	194
9.2.1	Customizing	194
9.2.2	Create Clarification Cases.....	194
10	New Functions in ERP 6.0 EhP7	195
10.1	Solution Process.....	195
10.1.1	Solution Process Control (Customizing)	195
10.1.2	Solution Process Control (BAI).....	198
10.1.3	Status Display of Solution Processes	200
10.2	Hide Objects of Clarification Case Container from Agent.....	201
10.3	Evaluation of Runtime Errors.....	201
	Appendix A - Reports with BPEM Specific Coding.....	208
	Appendix A1 - Creation of Clarification Case via Application Log	208
	Appendix A2 - Creation of Clarification Case by BAPI.....	211
	Appendix A3 – Sample Mass Activity Interval Processing Module	213
	Appendix B – List of Important SAP Notes	215

1 Introduction

1.1 Overview

Business processes become more and more complex in our modern world. Legal and business requirements have led to business processes where process steps may cross system borders multiple times. Moreover, many business processes are not even initiated in the own corporate IT systems anymore and lead to lots of interaction with IT systems of other companies. An excellent example for such business processes are the processes related to the deregulated utilities market where, based on the role of a market participant, information is routed through various systems until a business process completes.

Since we do not live in a flawless world, many problems and exceptions occur and need to be remediated manually or automatically to allow failed business processes to continue and eventually to finish. Remediation of problems requires an identification of a business process step that failed to complete. You do so in closely monitoring and managing your complex business processes by analyzing both successfully completed and failed business process steps.

SAP developed *Business Process Exception Management BP EM* to support you with this task. BP EM provides you with an overview of selected business processes in your SAP system mainly by automatically analyzing application log messages. It shows the total number of business objects involved in a certain business process broken down by the number of business objects processed successfully or with errors. In addition, you can also analyze the messages for general problems not assigned to a specific business object.

Clarification cases can be created for any identified problem and be routed to a specific agent or group of agents for further analysis and resolution of the problem. The analytics part of BP EM allows monitoring errors and success rates of your business processes and provides key performance indicators to measure the quality of your business process over time.

The main purpose of BP EM is to provide an overview of the system activities in a summarized form and to manage exceptions in an appropriate manner. Currently there is no other tool available in the SAP system that provides this functionality in the same comprehensive manner. BP EM enables you to view all the activities in analyzing application logs especially created by massively parallelized batch jobs (so-called mass activities) but also dialog processing. BP EM saves agents processing time and increases efficiency in processing the error logs by eliminating the need to retrieve the information from the system manually.

An automatic creation of clarification cases allows a structured predetermined resolution of a problem by means of solution paths and in forwarding these clarification cases to the right business departments or predetermined agents for a fast resolution. The agents process the clarification cases and, in keeping track of the completion of clarification cases, you can improve the business process quality and performance over the time. The result of the business process exception handling can be extracted as key performance indicators to the SAP BW system for further analysis.

Today, most problems or business exceptions are identified by the agents in studying daily hundreds and thousands of error messages in huge application logs and extracting the essential information of the logs into spreadsheets. The same spreadsheets are used to manage the resolution of the identified problems. In addition, due to the nature of application logs and spreadsheets, there is no way to directly navigate to the involved business objects

for further analysis. Transactions have to be called where you have to enter the key information by copy and paste from the application logs or spreadsheets.

Obviously, managing problems in this way is a very time consuming task. BPEM provides a major support by reducing the amount of time needed to analyze the application logs and allowing navigation from the error message to the business objects directly related to the reported problem. In addition, the extraction of statistics of successfully and unsuccessfully processed business objects became a similar time consuming exercise. Therefore, you need a tool that can give you an overview of the system's activities and where you can quickly review each executed business process.

For all this, BPEM is the tool to support you. It allows you to focus on improving your business processes instead of wasting time in creating and maintaining spreadsheets.

1.2 History and Naming Convention of BPEM

BPEM is the acronym of Business Process Exception Management and is based on the former product called EMMA (Enhanced Message Management Aalysis). The increasing number of available functions within BPEM and the support of many business processes made it necessary to use a more fitting name for this more generic and powerful tool. Still, the name EMMA can be found in many transaction codes, table names and enhancements.

The previous version of BPEM, EMMA, was implemented in a mixture of conventional ABAP and ABAP OO. When more and more customers used EMMA to support their business, the demand of further functions grew tremendously. As it became increasingly difficult to enhance the initial implementation, SAP decided to redesign the EMMA implementation and rename the product to BPEM (project name is *New EMMA*). In contrary, the previous version of BPEM is still known as *Classic EMMA*. BPEM was mostly re-implemented in ABAP OO with all advantages an object oriented implementation offers not only for new business processes but it leaves also space for installation-specific enhancements. BPEM is fully downward compatible to EMMA. Only some very few application processes are still not yet updated to allow EMMA to take full advantage of all new features of BPEM. The further chapters describe in details all possible enhancements.

The most obvious difference between the *Classic EMMA* and the *New EMMA* respectively BPEM is that the well-known EMMA transaction was replaced by the EMMAJL and EMMAJP transactions. The functions of the EMMA transaction were split into the two EMMAJL and EMMAJP transactions for an easier handling of the BPEM related tasks. In addition, the FPEMMAMA transaction was implemented as a so-called mass activity allowing the BPEM related tasks to be processed with concurrently running batch jobs. SAP recommends using only the newly developed transactions. In the following chapters, this document introduce and explain all mentioned transactions in details.

BPEM is delivered as part of the Contract Accounts Receivable and Payable solution (also known as FI-CA) and can be used by any industry solution based on FI-CA. BPEM is not bound to a particular industry solution thus can easily be customized to analyze also the result of any installation-specific application.

1.3 Main Functions of BP EM

BP EM offers the two main functions **Monitoring of Business Processes** and the **Management of Business Exceptions**. They are introduced in the following chapters.

1.3.1 Monitoring of Business Processes

Almost all business processes document their progress, success and failure in writing messages into application logs. Some few business processes use the job log or own database tables to store status information. The detail level of the logged messages varies in dependency of the Customizing and the severity of the identified issues or exceptions.

An application log is nothing more than a series of messages describing the reasons or circumstances why a process failed and, usually at the end of a message block, to which business object the messages are related. In addition, many business processes, in particular the so-called mass activities that commonly process overnight huge volume of data, create a huge number of messages created, due to technical reasons, in more than one application log.

It is part of the daily work of many clerks or agents to check and analyze the logged information to determine the reason of failed business processes and, more importantly, to take appropriate actions to allow eventually the business process to continue or a successful repetition of the failed process step. For this, agents use commonly the `SLG1` and the SAP for Utilities solution specific `SLG_ISU` transactions to display and scroll through application logs or other application specific lists of messages. Besides these transactions, there are many more application specific transactions available for an exception and error monitoring.

The huge amount of information to be analyzed and the different ways messages are presented to the agent makes it difficult to focus on the main purpose why the messages must be analyzed in the first place: The identification and separation of successful and failed business processes and related business objects. Further, application logs do not provide a forward navigation in double-clicking on a message to immediately display the data of the related business object as referred to in the message.

BP EM presents the results of a business process sorted by business objects regardless how the messages have been stored by the application. It displays first the total number of processed business objects per business process broken down by the number of business objects processed successfully or with warnings or errors. From this overview you can drill down to the individual message per business object. In addition, you can also analyze messages for general problems not assigned to a particular business object.

Job	External Job Number	Mod	Inter	IncomO	Succ. Ob	Proc. O	Err	Inform	Total	Date	Time	Sta	Stat	Cases
100020251	BILL 19.12.2012 FK0041 Completed		4	2	0	2	0	35	44	19.12.2012	15:21:36			2
100020248	BILL 19.12.2012 FK0040 Completed		4	2	0	2	0	35	44	19.12.2012	15:16:51			0
100020198	BILL 18.12.2012 FK0039 Completed		4	2	0	2	0	35	44	18.12.2012	15:06:46			2
100020171	BILL 18.12.2012 FK0038 Completed		4	2	0	2	0	35	44	18.12.2012	10:39:14			2
100020055	BILL 01.12.2012 FK0037 Completed		4	2	0	2	0	35	44	01.12.2012	15:49:30			2
100020054	BILL 01.12.2012 FK0036 Completed		4	2	0	2	0	35	44	01.12.2012	14:02:16			2
100019778	BILL 06.11.2012 FK0035 Completed		4	2	0	2	0	35	44	06.11.2012	09:44:26			2
100019774	BILL 02.11.2012 FK0034 Completed		4	2	0	2	0	35	44	02.11.2012	17:54:23			2

Figure 1-1 Job Overview

1.3.2 Management of Business Exceptions

After an analysis of the application log by an agent the challenge remains how to resolve the newly created backlog of exceptions by the responsible clerks or agents in an organized way. Without BPEM, the agents commonly download the application logs to spreadsheets, sort it and organize the work from here in forwarding the exceptions to the respective teams and team members. The same agent must request and collect feedback on the progress, such as exceptions that are resolved, and feed it back into the spreadsheets. The endeavor is to ensure that by the end of the business day all exceptions are resolved hence the backlog is cleared and the teams are ready to work on the next wave of exceptions to come on the day after.

It is easy to imagine that, if this work is not done meticulously, exceptions may slip through and, as a consequence, business processes get stuck – until, for example, you need to manage an increasing number of churned customers who call and ask why they are not served well because their requests were not answered yet.

BPEM organizes the exception resolution by means of so-called clarification cases. Clarification cases can be created automatically, for example upon failed execution of an automatic business process, or it can be created manually. A clarification case aggregates the logged error messages, the failing business object itself and a list of transactions and business processes the agent can execute by a push of a button to resolve the reported business. The commonly maintained list of specific transaction codes in handbooks or even just on a piece of paper used to resolve a particular exception becomes obsolete since the same solution relevant transactions can be called directly from the clarification case.

Clarification cases can be automatically forwarded to a specific agent or agent group by means of determination rules specific to your organization or forwarded manually to a co-worker. In addition or alternatively, work lists can be created displaying clarification cases specific to a team or a team member sorted by priority and due date.

Clarification case KPIs can be analyzed on a connected BW system to support an identification of potential space for improvement in a continuous process innovation cycle.

1.3.3 Main Benefits of using BPEM

The main benefits of using BPEM are:

- Centralized monitoring of conducted business processes
- Statistical analysis of messages and processed business objects
- Centralized monitoring of exception handling
- Reduction of administration overhead
- Ability to distribute clarification cases to specific departments and/or representatives
- Prioritizations of clarification cases based on customizing and business object attributes
- Prioritized work lists of clarification cases
- Resolution of exceptions by manual, semi-automatic and automatic solution processes
- Business Process Analytics
- Reduction of exception rate leading to improved and smoother business processes
- Reduction of operating cost due to efficient exception handling

2 Business Architecture and Data Flow

2.1 Glossary of BPEM Terms

This chapter provides an overview of the BPEM specific terminology used in this document. It enables you to get quick and easy access into the topic and improves the understanding of the further explanations.

2.1.1 Business Process Area

A business process area describes a set of similar business processes that belong to the same functional application area. Examples for business process areas are *EMR Meter Reading*, *EBI Billing*, *EIN Invoicing*, *DUNN Dunning* and *PAYP Payment*.

2.1.2 Business Process Code

A business process code describes a particular business process performing a unique operation within a business process area. Examples for business codes in the billing area are *EBI00001 Automatic Billing*, *EBI00002 Manual Billing*, *EBI00003 Billing Reversal* and *EBI00005 Bill Correction*.

2.1.3 Primary Object Type

To each business process code a so-called primary object type is attached and represents a business object of the Business Object Repository (BOR). For example, the *INSTLN* BOR object, representing a utility installation, is attached to the *EBI00001* business process code *Automatic Billing*.

2.1.4 Clarification Case

A clarification case describes an encountered exception of a business process. It is created automatically or manually and is based on a clarification case category.

2.1.5 Clarification Case Category

A clarification case category describes a template for the creation of clarification cases. A clarification case category includes conditions when it should be used to create a clarification case, which business objects and solution processes are to be assigned to the clarification case and the distribution of the clarification case to agents.

2.1.6 Clarification Case Type

A clarification case type categorizes a clarification case and is for information and administration purposes only. It can be used to group clarification cases in work lists. A clarification case type does not have any further functionality. All manual cases created from the interaction call center could, for example, share a common clarification case type for quick identification.

2.1.7 Message Pool of a Clarification Case Category

On the *Messages* screen of a clarification case category you enter the messages that describe a particular business exception as it can be found, for example, in the application log of a business process. Usually one message or a message combination leads to the creation of a clarification case based on the given clarification case category. The list of entered messages is also called *message pool* of the clarification case category.

2.1.8 Business Object Repository BOR and BAPIs

The SAP Business object repository BOR contains a set of entities with common characteristics and common behavior representing well-defined business semantics. The set of entities is generally accepted in the business world (for example in an international standard or industry best practices) as business objects, such as *Business Partner* or *Service Order*.

Business objects of the BOR describe complete business processes. This type of encapsulation reduces complexity because the inner structure of the business object remains concealed. By calling Business Application Interfaces (BAPIs), external and internal applications can access and manipulate the business objects using the Internet or an interface. A BAPI can be, for example, a method, a function module or a web service.

2.1.9 Container and Data Flow in a Clarification Case Category

All business objects and message variables as defined in a clarification case category form the clarification case container. Later, during runtime, the objects and related data is replaced by the data of the currently processed business objects. The information in the clarification case feeds the solution processes of a clarification case and allows a replacement of used variables in the clarification case description. The links between the data in the clarification case container and the consumers of the data, the so-called bindings, are created with the binding editor of the SAP Business Workflow Builder.

2.1.10 Binding Editor

You use the binding editor to define a uni- or bidirectional data flow between one more elements of two containers. BPBM uses the binding editor of the SAP Business Workflow Builder. You find detailed information on the binding editor at the **SAP Online Library Documentation** <http://help.sap.com/>. To access the documentation, start the SAP Workflow Builder (transaction SWDD) and choose menu *Help* → *Application Help*.

You start the binding editor by pushing the displayed  button or  button. In most cases, the binding editor can generate the required binding automatically by simply pushing the  button.



BPBM supports as a binding only the simple value assignment. The additional binding options as known by the SAP workflow are not implemented yet, such as the further options that are available on the *Binding Instruction Change* window.

2.1.11 Condition Editor

You use the condition editor to formulate a condition, for example, to define a complex condition to determine if a clarification case should be created based on the current clarification case category. BPBM uses the condition editor of the SAP Business Workflow Builder. You find detailed information on the condition editor at the **SAP Online Library Documentation** (<http://sap.help.com/>). To access the documentation, start the SAP Workflow Builder (transaction SWDD) and choose menu *Help* → *Application Help*.



If you use the binding editor the first time, push the  button in the binding editor and maintain the user settings, for example, to enable drag and drop function or to switch to the full screen binding editor. You may also push the  button to access a comprehensive documentation about the binding editor.

2.1.12 BPEM Job

Business processes store information about encountered problems mostly in form of messages in application logs. When a business process is activated for BPEM, it creates an entry in the BPEM process list with an own unique identification, business process code of the executed process and further administration data, such as creation date. In the BPEM terminology, the entry in the BPEM process list creates a so-called BPEM Job and can be uniquely identified by its job number (EMMA_HDR-RUNID field).

2.1.13 Mass Activity

A mass activity is a program that realizes business processes where large data volumes are processed, such as the payment run or dunning run as available in Contract Account Receivable and Payable application. A mass activity is based on the Framework for Parallel Processing (FPP) and enables business processes to be processed in parallel.

Mass activities automatically split the dataset to be processed, such as a quantity of business partners or contract accounts, into intervals according to the definition of a parallelization object and related variant. Then the mass activity distributes the intervals to several technical jobs on multiple application servers and processes them concurrently.

2.1.14 Time Stamp

A time stamp represents the local date and time to determine unique points in time. BPEM uses a form of a time stamp that contains seven decimal places for fractions of seconds, which allow for an accuracy of up to 100 ns.

The accuracy of the decimal places of the long form depends on the hardware (processor) of the application server. The maximum resolution of 100 ns is not always reached. On some platforms, only a resolution of milliseconds is available. Even though, BPEM ensures unique time stamps to allow a correct sorting of messages by their creation time (for required corrections, check SAP Note [1730316](#) and related SAP Notes).

2.1.15 SAP Enhancement Framework and BAdI

The SAP Enhancement Framework enables you to add functionality to the standard SAP software without actually changing the original repository objects and to organize these enhancements as effectively as possible. With this technology you can enhance global classes, function modules, Web Dynpro ABAP components, and all source code units using implicit enhancement options provided by the system. An application developer can also define additional explicit enhancement options for source code plug-ins and new kernel-based BAdIs, which are also integrated in this new framework.

The previous concepts include two different methods for adapting SAP software that go beyond the scope of customization:

- Modifications – changes to development objects delivered by SAP with or without the Modification Assistant
- Enhancements – inserting user developments into SAP development objects at predefined positions using customer exits, appends, includes, enhancement spots and classic Business Add-Ins (BAdIs).

The new Enhancement Framework is intended to integrate the existing enhancement and modification concepts, to replace and improve some of them, and to add some new ones to address recent developments.

BAdI is an object-oriented enhancement option, which makes it the most sophisticated enhancement type. The main characteristic of a BAdI is that it provides a mechanism to change the functionality of a well-defined business function (e.g. a BAPI) without making changes to the delivered source code. Future upgrades of the original business function can be applied without losing the installation-specific enhancements or the need to merge the changes. The two code lines (the original one and the installation-specific coding) are strictly separated but still integrated.

Enhancement spots are containers for explicit enhancement options and carry information about the positions at which enhancement options were created. One enhancement spot can manage several enhancement options of a repository object and vice versa; several enhancement spots can be assigned to one enhancement option.

The new BAdI concept takes advantage of SAP's extensive experience in offering customers different ways to enhance their SAP system. The new BAdIs add some major improvements to the classic BAdIs such as better performance. The new BAdIs are integrated in the kernel and are switchable. Within the enhancement framework, a new BAdI is always meant when the term BAdI is used. If there is an explicit reference to the previous BAdI concept, such BAdIs are referred to as classic BAdIs.

You find detailed information on the SAP Enhancement Framework at the **SAP Online Library Documentation** (<http://sap.help.com/>). To access the documentation, start the SAP BAdI builder (transaction `SE18`) and choose menu *Help* → *Application Help*.

2.1.16 SPA/GPA Parameters

You can prefill fields on screens with default values from SAP memory by means of SPA/GPA parameter. SPA/GPA parameters are a technique to pass data between different sessions of a user. You maintain your personal SPA/GPA parameter with the `SU2` transaction. Fields on screens can only be filled automatically with the value saved under the parameter ID if the Set Parameter/Get Parameter attributes for the corresponding fields have been explicitly set in the Screen Painter.

2.1.17 ABAP List Viewer ALV

The ABAP List Viewer (ALV) is a tool for displaying and formatting lists in the SAP system based on the SAP GUI user-interface technology. The ABAP List Viewer standardizes and simplifies the use of lists and reports in the SAP system. The ABAP List Viewer provides you with standardized interfaces and list formats for all lists and reports. Some of the key functions of the ABAP List Viewer are:

- **Sort lines:** You can sort the lines in ascending  or descending  order according to column value.
- **Filter lines by values:** You can choose to have the system display only the lines that meet certain criteria in setting  filter values.
- **Detailed information:** You can display detailed information  for the individual lines of the list.
- **Search for values:** You can search  for lines containing specific information.
- **Managing layouts:** Layouts  enable you to change  the format of lists. You can specify which fields are to be displayed, change the sequence in which these fields are displayed, and how the lines of the list are sorted before the list is displayed. In addition to this, you can adjust the width of the individual columns to meet your specific requirements.
- **Export list:** You can print  list and call up a print review  before printing. In addition, you can export  the list, for example, into a spreadsheet or save it as a local file.



Not all of the above-mentioned functions are available in every list.

You find detailed information on the ABAP List Viewer at the **SAP Online Library Documentation** (<http://sap.help.com/>). To access the documentation, push the  button in the toolbar of an ALV list.

2.2 Overview BPEM Processes and Transactions

Business processes store their error messages and additional information in application logs or in application specific locations, such as database tables. In addition, the business processes create an entry in the BPEM job list. Starting from this entry, BPEM can pick up the stored information from any place and make it available to the agents.

The following figure shows the five main process steps BPEM supports once a business process finishes. These five process steps are explained in details in the following chapters.

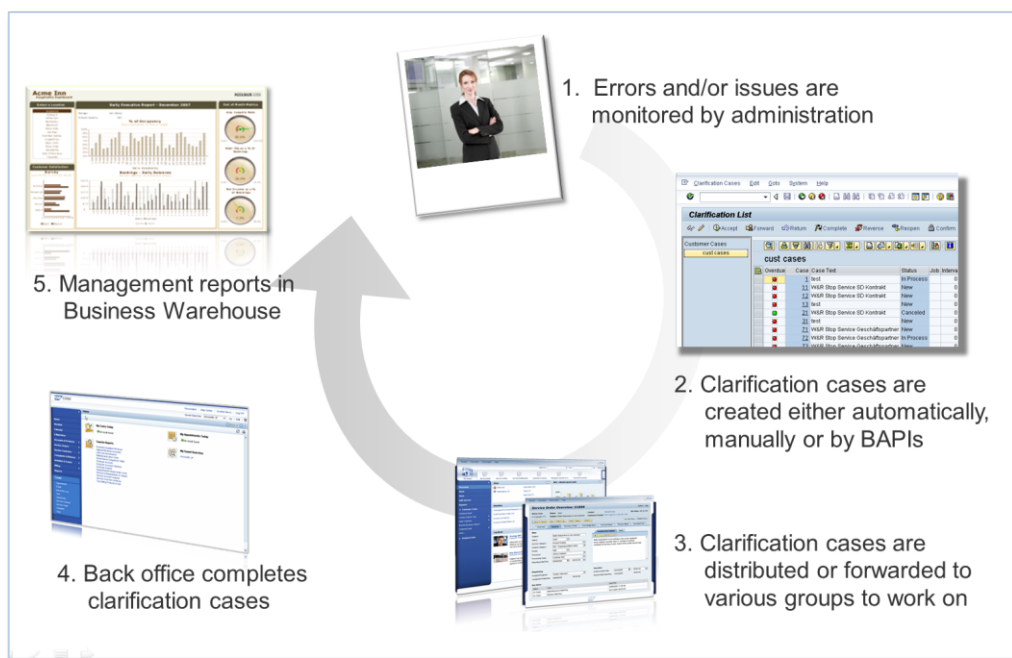


Figure 2-1 Overview BPEM Process Steps

2.2.1 Step 1: BPEM Job Preparation

During this process step, BPEM collects application relevant information from all places where the application stored it, analyzes their relevance and sorts it by the system time stamp and related business object, identifies failed business objects by their error and warning messages, consolidates the collected information and saves it in BPEM specific database tables for a later efficient access. The preparation step can be reversed and re-executed. You can reverse the preparation step if the subsequent process step 2 *Clarification Case Generation* was either not performed yet or, if it was already performed, previously reversed. However, a reversal of the clarification case generation step is only possible as long as the status of any related clarification cases was not changed since the generation of the clarification cases.

The following transactions are available to perform the preparation step and are explained in details in later chapters:

Transaction <code>FPEMMAMA</code>	<i>BPEM: Mass Activity</i>
Transaction <code>EMMAJP</code>	<i>Business Process Analysis: Process Jobs</i>
Transaction <code>EMMAJL</code>	<i>Business Process Analysis: Job List</i>

2.2.2 Step 2: Clarification Case Generation

During this process step, BPEM compares the collected error messages and the related business objects with conditions described in the defined clarification case categories. If the conditions match, a clarification case is created according to the definition of the matching clarification case category. If the conditions of more than one clarification case category match, multiple clarification cases can be the result.

The following transactions are available to perform the clarification case generation step and are explained in details in later chapters.

Transaction <code>FPEMMAMA</code>	<i>BPEM: Mass Activity</i>
Transaction <code>EMMAJP</code>	<i>Business Process Analysis: Process Jobs</i>
Transaction <code>EMMAJL</code>	<i>Business Process Analysis: Job Lists</i>

In addition, you can create clarification cases manually:

Transaction <code>EMMAC1</code>	<i>Create Clarification Cases</i>
BAPI <code>BAPI_EMMA_CASE_CREATE</code>	<i>Create Clarification Case Manually</i>
BOR Method <code>EMMACase.Create</code>	<i>Create Clarification Case Manually</i>

2.2.3 Step 3: Distribution of Clarification Cases

During this process step, a clarification case can be distributed and forwarded to a specific agent or agent group according to the processor determination rule as defined in the clarification case category. No particular transaction is available for the automatic distribution of clarification cases since the processor determination rule is performed immediately after the clarification case was created during process step 2.

The following transactions are available to display lists of created clarification cases and assign agents to clarification cases manually. The transactions are explained in details in later chapters:

Transaction <code>EMMACL</code>	<i>Clarification Case List</i>
Transaction <code>EMMACLS</code>	<i>Clarification Case List with Shortcut Keys</i>

2.2.4 Step 4: Resolution and Completion of Clarification Cases

During this process step, agents analyze and resolve the clarification case supported by the information and solution processes as presented in the clarification case. The system documents the result of the analysis of the business exceptions and executed solution processes in an application log.

You maintain clarification cases with the following transactions:

Transaction `EMMAC2` *Change a Clarification Case*

Transaction `EMMAC3` *Display a Clarification Case*

As explained before, an agent may pick a clarification case from a list of clarification cases to become a processor of the clarification case. You can work with clarification case lists with the following transactions. All transactions are explained in details in later chapters:

Transaction `EMMACL` *Clarification Case List*

Transaction `EMMACLS` *Clarification Case List with Shortcut Keys*

The system can execute solution processes of clarification cases if you marked these solution processes as automatic solution processes. The following transactions are available to start the execution of automatic solution processes. All transactions are explained in details in later chapters:

Transaction `FPEMMAMA` *BPEM: Mass Activity*

Transaction `EMMAJP` *Business Process Analysis: Process Jobs*

Transaction `EMMACAP` *Execute Automatic Solution Processes*

2.2.5 Step 5: Transfer of KPIs to Business Information Warehouse

During this process step, you extract key information from clarification cases to a business information warehouse to measure the performance of business processes. This is available for all applications. In addition, special extractors are available for the measurements of the SAP for Utilities business processes to extract further information, such as statistics of successful and failed SAP for Utilities specific business processes. For detailed information regarding the SAP standard content, see chapter 7 *Business Information Warehouse*.

2.3 Implementation Approach

There are a several options available to implement BPEM and take quickly full advantage of the available BPEM functions. Lessons learned from many projects proved that it is extremely beneficial to collect first some experience in one or two business areas or business processes prior to a broader rollout on all the business processes you run on your system. There are some quick wins, such as taking advantage of the BPEM enhanced message list instead of scrolling through application logs as presented by the `SLG1` transaction. Chapter 2.3.1 *Getting Started with an Enhanced Message List* describes the required Customizing steps to put it in place.

Some more preparation time and planning is required if you decide to take advantage of clarification cases and clarification case lists to eventually significantly improving the business processes in your company. See chapter 2.3.2 *Getting Started with Clarification Cases* for some guidance how to implement successfully the usage of clarification cases.

2.3.1 Getting Started with an Enhanced Message List

You can quickly get started with BPEM. In a first step, you just need to activate the BPEM message management for your SAP standard business processes to take advantage of the enhanced functions of structured lists of error messages. You do so in performing the two Customizing activities as described in the chapters 3.3.1 *Maintain Internal Number Range Intervals for Jobs* and 3.3.2 *Activate Business Process Areas for Message Management* that are relevant for your business processes.

In addition, you should include the `FPEMMAMA` transaction in your nightly batch chain to execute the BPEM job preparation for the activated business process areas (for more details, see chapter 4.2.1 *Transaction FPEMMAMA - BPEM Mass Activity*). With these, you avoid waiting time the first time you want to display the enhanced message lists, particularly for messages created by mass activities, since the BPEM job preparation is a mandatory activity before the system can displaying the enhanced message list and would be executed online if it was not performed before.

2.3.2 Getting Started with Clarification Cases

The introduction of clarification cases requires preparation to design and build a consistent approach for all business areas for which you intend to roll out BPEM. Many customers implement a phased roll-out starting with one or two business areas first, to gaining experience how to best setup clarification cases while receiving feedback from the agents to improve the implemented clarification case categories which, as said above, are used as templates to build clarification cases. You can increase the acceptance level significantly in standardizing the resolution of the most common exceptions you certainly will have to identify in the course of the BPEM implementation project.

2.3.2.1 Assessment of Today's Business Exceptions and Resolutions

You may use the following questions as a starting point to assess the as-is situation and create a blueprint for an implementation of BPEM:

- What kind of business exceptions do you experience day-to-day?
- How do you monitor business processes and their exceptions monitored today?
- What are the most recurring and most time-consuming business exceptions to work on?
- How do you notice the right agent or business department to resolve a business exception?
- What is the current process to resolve a specific business exception?
- How do you ensure that all business exceptions are resolved in a timely manner?

Some of the most common answers to these questions may provide some further assistance for your assessment.

What kind of business exceptions do you experience?

Mostly, business exceptions fall into one of the following categories:

- Missing or wrong Customizing
- Tolerances used in validation checks are too strict leading to, for example, too many invalid thus outsourced documents
- Missing or incorrect master data that exists in your system or is continuously created with your system by interfaces
- Required information is not available at the time a business process is performed
- Program errors

What are the processes for which you frequently experience business exceptions? Are there further causes for business exceptions? What is the root cause of the most re-occurring business exceptions? Has this root cause been addressed already?

How are business processes and their exceptions monitored today?

SAP offers a broad variety of specialized monitors for different kind of system activities and encountered exceptions. Which of them are used by whom on a daily, weekly or monthly basis? Often the customers complete the delivered SAP monitors by own installation-specific reports. How are the results of these monitors reported and how accurate are these results?

What are the most recurring and most time-consuming business exceptions?

Do the business departments maintain statistics of the encountered business exceptions? How do they do this? Are the statistics created either on the SAP ERP or SAP BW system and regularly analyzed? It is important to understand what the most re-occurring business exceptions are to pay special attention to treat them in the most efficient way.

How are business exceptions allocated to the right agent or business department?

What processes are in place to ensure that a business exception finds the right agent? How the agents are made aware of business exceptions she or he must resolve? Is this process implemented based on lists, emails or by workflows? Is an approval process in place to confirm the resolution of a business exception?

What is the process to resolve a specific business exception?

How does an agent approach the resolution of a business exception today? Is there a standardized process implemented? Which the transactions and reports the agents use? What kind of authorization the agents require for these transactions and reports?

How is it ensured that all business exceptions are resolved in a timely manner?

What kind of reporting is in place to monitor the not yet completed clarification case lists?

With answers to the questions above you have everything at hand to standardize the problem resolution process and an automatic creation of clarification cases.

2.3.2.2 Planning the Resolution of Business Exceptions with BP EM

The next paragraphs provide some answers to the questions of the previous chapter from a BP EM prospective. However, it is important to recap how BP EM works:

BP EM analyzes the reported problems of the business processes and transforms each problem into a problem specific clarification case. Clarification cases consist not only of the specific error messages that triggered the creation of the clarification case but also, in dependency of the individual Customizing, of a link to the involved business objects you can navigate to as well as of a list of remediation actions you can perform by a push of a button. In addition, the system maintains a log to document the progress until the resolution of the business exception. The system can automatically allocate to agents or business departments to clarification cases by means of BP EM forwarding methods.

You should consider the following questions and answers when planning your BP EM implementation:

How can the business processes and their exceptions be monitored by BP EM?

BP EM offers statistics for each business process with basic information how many business objects have been processed and how many of the related business processes failed to complete. After a complete implementation of BP EM, all encountered business exceptions

will be transformed into clarification cases. With this, further process statistics are available in SAP ERP as well as in SAP BW.

In addition, a processing status is allocated to each clarification case. This status allows to monitoring the completion of clarification cases and the required time until a clarification case is closed. These statistics are available in SAP BW.

What are the most recurring and most time-consuming business exceptions?

You can get an answer to this question in analyzing statistics. With the additional support of a connected SAP BW system, you can easily identify the business processes that create most business exceptions or the business exceptions that require the most attention of the agents. With this information, you can focus on exactly these business processes and their exceptions first, trying to prevent the issues to recur or at least providing a remediation that allows a fast problem resolution.

How are business exceptions allocated to the right agent or business department?

Clarification cases can be allocated and forwarded by means of agent determination rules or simply by clarification case lists where the selection of clarification cases can be individualized for each agent or business department. In addition, you can forward a clarification case manually to another agent or return the clarification case simply to the backlog queue without a specific processor.

What is the process to resolve a specific business exception?

A clarification case may include a list of solution processes that can be performed by the push of a button. Identifications of the involved business objects, such as the business partner number, are case specific and are transferred to the called SAP standard or installation-specific transactions and reports.

The automatic creation of clarification cases is based on clarification case categories the system uses as templates to transform similar business exceptions into clarification cases but each with their individual business objects. In order to create a clarification case category you need to analyze how an agent remediate the same issue today and what the related business rules are. Concisely, you will have to answer the following questions:

- How do you identify a particular business exception?
- What steps are required to resolve a business exception?
- Is it required to perform the steps in a certain order?
- Can steps be performed automatically without an agent intervention?
- Who should work on this business exception to resolve it?
- How important is this business exception and how fast does it need to be resolved?
- Which further actions are necessary to prevent the same business exception to recur?

With answers to these questions you have everything at your hand to standardize your problem resolution process and create clarification case categories. The automatic creation of clarification cases and the tracking of the clarification case completion help you to resolve similar problem much more efficiently.



Keep in mind that it is not possible to replace all system and application monitors by BPME. Even though, BPME offers you the option to make any business exception visible by means of clarification cases.

To take advantage of clarification cases you need to perform the activities as pointed out in chapter 2.3.1 *Getting Started with an Enhanced Message List*. In addition, you need to perform the additional Customizing activities as described in chapters 3.4 *Specifications for Generating Clarification Cases* and 3.5 *Specifications for Processing Clarification Cases*.

Furthermore, you should include the `FPEMMAMA` transaction in your nightly batch chain to execute not only the BPEM job preparation for the activated business process areas but also the automatic clarification case generation process.

2.3.2.3 Design of Clarification Case Categories

As said before, the system may generate clarification cases automatically if there is a match of the application error messages to the triggering messages as defined in a clarification case category.

When you start designing and creating clarification case categories you will be asking yourself how to ensure that for all possible error messages hence business exceptions a clarification case category is available since it is impossible to anticipate any possible exception situation particularly if a business exception occurs very rarely. In short: what to do if all defined business process specific clarification case categories and their triggering messages have not led to the required creation of a clarification case?

The good news is that there is a special business process code `EMMACAS2` available to handle exactly such a situation. You may define clarification case categories that are, by nature, generic but ensure that you definitely create a clarification case if none of the process specific clarification case clarification categories has led to a clarification case. You can monitor the creation of such clarification cases and create step-by-step more sophisticated clarification case categories hence clarification cases for these so far unknown business exceptions if required. You may even create a second clarification case with the purpose to indicate the creation of a new clarification case category. For more details, see chapter 6.3.3 *General Sweeper Clarification Case Categories*.

This means that you may start with only very few developed clarification case categories plus some generic sweeper clarification categories and put them into production. Subsequently, you may complete the specific clarification case categories step-by-step without being afraid that a business exception may slip through until you created all required clarification case categories.

3 Customizing

3.1 Implementation Guide – General

You find the BP EM related Customizing activities in the SAP Implementation Guide IMG at *Financial Accounting (New) → Contract Accounts Receivable and Payable → Basic Functions → Enhanced Message Management*. The figure below shows all available BP EM specific Customizing activities.

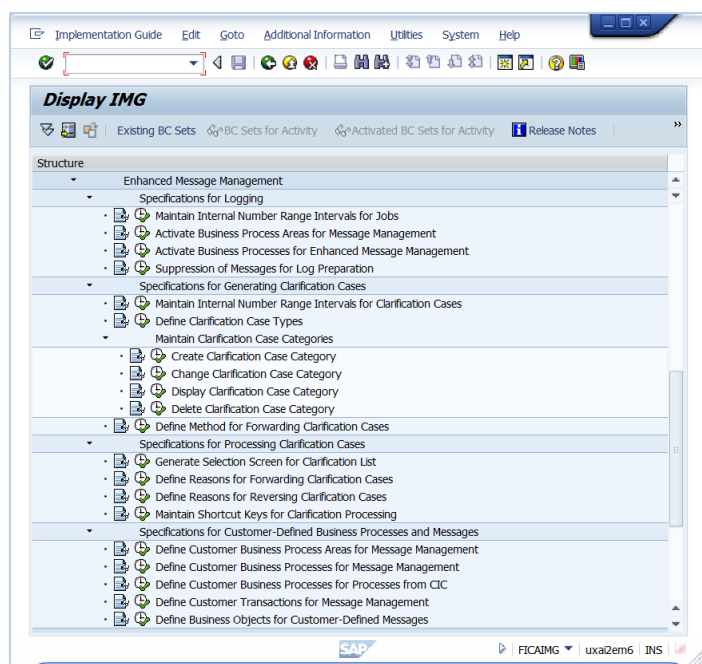


Figure 3-1 Overview BP EM Customizing Activities

The following chapters describe all Customizing activities in details with the objective to permit an initial simple Customizing of BP EM without having to know and thus being distracted by lots of additional and more sophisticated options of the BP EM Customizing. This approach allows to quickly taking advantage of BP EM. As you learn more and more about the advantages of BP EM as more you want to detail your implementation and implement a more sophisticated solution. For this, there are further chapters that explain how to take advantage of the available specialized and sophisticated functions and settings of BP EM.

3.2 Authorization

3.2.1 Authorization for BP EM Customizing

The **S_TABU_NAM** authorization object protects the update and/ or display access to the Customizing views when accessing these views via the SAP Implementation Guide. However, when you access the same table views via the **SM30** transaction, your authorization profile must include the **S_TABU_NAM** authorization object.

In addition, the activities to access the clarification case categories, the transactions **EMMACCAT1** (create clarification case category), **EMMACCAT2** (change clarification case category) and **EMMACCAT3** (display clarification case category), are protected by a check against the **S_TCODE** authorization object. In addition, the **S_TRANSPRT** authorization object prevents unauthorized agents to add clarification case categories to a transport request.

3.2.2 Authorization for *BPEM Job Processing*

The **B_EMMA_LOG** authorization object protects the functions for processing BPEM jobs within the framework of BPEM. The **B_EMMA_LOG** authorization object contains the two **ACTVT** (Activity) and **BRGRU** (Authorization Group) fields. The **ACTVT** field describes the allowed activity as stored in the **TACTZ** table, such as create, display, prepare BPEM jobs and delete the data created during BPEM job preparation. The **BRGRU** field describes authorization groups that allow extended authorization protection for particular objects. The authorization groups assigned to an authorization object are stored in **V_TBRG** table view and are freely definable via the **SM31** transaction.

The following activities on BPEM jobs are available:

Activity	Description
01	Create BPEM Job
03	Display BPEM job
71	BPEM job preparation
06	Reverse BPEM Job preparation
40	Generate clarification cases for a BPEM job
41	Delete generated clarification cases
A8	Execute automatic solution processes of a BPEM job
24	Archive BPEM job

Figure 3-2 Activity Codes of Authorization Object **B_EMMA_LOG**

3.2.3 Authorization for *Clarification Case Processing*

The **B_EMMA_CAS** authorization object protects the functions for processing clarification cases within the framework of BPEM. The **B_EMMA_CAS** authorization object contains the two **ACTVT** (Activity) and **BRGRU** (Authorization Group) fields. The **ACTVT** field describes the allowed activity as stored in the **TACTZ** table, such as create or generate, change, display, reopen, delete, archive and further clarification case specific activities. The **BRGRU** field describes authorization groups that allow extended authorization protection for particular objects. The authorization groups assigned to an authorization object are stored in **V_TBRG** table view and are freely definable via the **SM31** transaction.



The provided authorization affects the availability of buttons in the toolbar of the **EMMACL** and **EMMACLS** transactions. The system automatically removes the buttons in the toolbar without the required authorization for an execution of the related function. For more information, see chapter 4.3.1 *Transactions EMMACL/EMMACLS – Clarification Case List*.

The following activities on clarification cases are available:

Activity	Description
01	Create or generate clarification case
02	Change clarification case
03	Display clarification case
06	Delete clarification case
31	Confirm clarification case

37	Accept clarification case and assign it to current agent
78	Assign (forward) clarification case to a different agent
83	Complete / acknowledge confirmation clarification case
85	Reverse clarification case
87	Return clarification case to worklist
C5	Reopen clarification case
24	Archive clarification case

Figure 3-3 Activity Codes of Authorization Object B_EMMA_CAS

3.3 Specifications for Logging

The following Customizing affects the applications connected to BPDM to store BPDM specific data in addition to the application specific messages.

3.3.1 Maintain Internal Number Range Intervals for Jobs

The system uses internally assigned unique BPDM job numbers. In this activity, you define the number range intervals of the EMMA_RUNID number range object (*BPDM Job*) that the system accesses when creating new BPDM jobs. You will have to create at least one internal number range interval. BPDM does not support an external number assignment.

If you want to allocate numbers from a specific number range interval to business process areas, you need to setup several number range intervals. SAP delivers the two predefined number range interval 00 from 0100000000 – 0199999999 and 01 from 0000000001 – 0099999999.

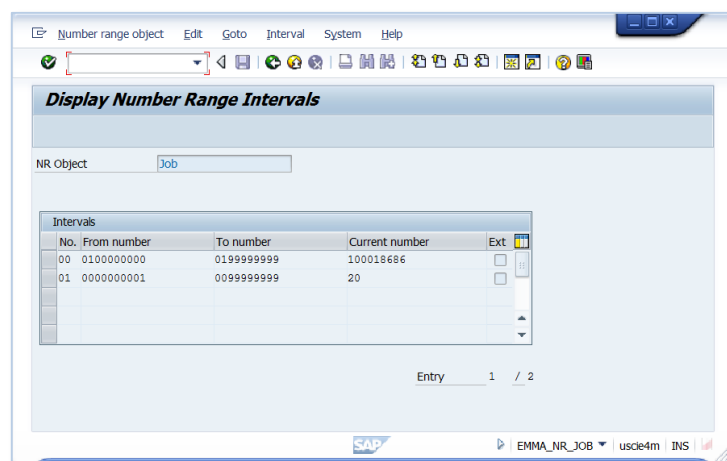


Figure 3-4 Intervals of EMMA_RUNID Number Range Object



Prior to adding and adjusting intervals, check for the highest number used in the RUNID field of the EMMA_HDR table and adjust the current numbers respectively.

3.3.2 Activate Business Process Areas for Message Management

In this activity, you define which business process areas shall be monitored by BP EM.

You have to enable the business process areas for monitoring in selecting one of the available entries for the **Record Process** field. By setting the **Record Process**, you include or exclude individual business process areas from the evaluation. You can choose one of the following options:

- **Deactivated:** You can evaluate the application logs of the jobs concerned; however, BP EM does not recognize the business processes and related business objects.
- **Message management activated:** Business process-specific and business object-specific messages are stored in the application log for a later evaluation by BP EM.
- **Message and BW process management activated:** In addition to the second setting, the information relevant for process management is transferred to the Business Information Warehouse. However, this function is currently only supported for certain business process areas in the SAP for Utilities industry solution (*BW relevant* indicator is marked).

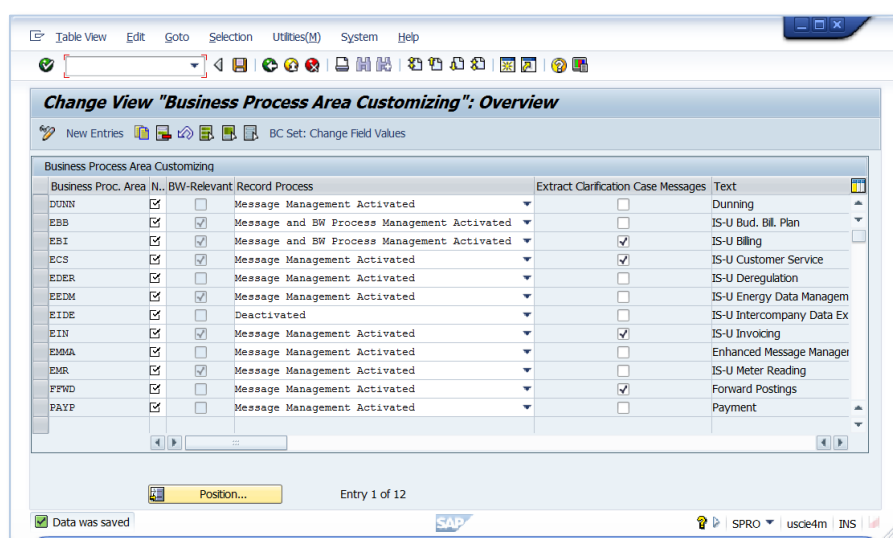


Figure 3-5 Activation of Business Process Area

If you want to allocate numbers from a specific number range interval to business process areas, you enter the related interval number of the `EMMA_RUNID` number range object in the **No.** field. However, this function is rarely used and should only be activated if an analysis of BP EM jobs is required based on number range level instead of business process level.

The **Extract Clarification Case Messages** indicator controls to what extent messages are extracted by the clarification case DataSource. You can choose one of the following options:

- Mark this indicator if you want to extract also the messages that have led to a clarification case. For each message, one data record is transferred to the Business Information Warehouse BW. Due to performance reasons, this setting should be used with caution for business process areas that mainly create their application logs in mass activities.
- Do not mark the indicator if you do not want to extract also the messages that have led to a clarification case. Only the BP EM start message is extracted for each business process code and business object completed by the general clarification case information.

For more details concerning the extraction to BW, see chapter 7 *Business Information Warehouse*.

An authorization group is part of the authority check for any BPEM related function. In the context of the business process area, you can restrict the creation of BPEM jobs for particular business process areas to certain agents. Enter an authorization group in the **Authorization** field if you want to define a default authorization group per business process area. The system uses the entered authorization group when creating a new BPEM job unless the default authorization group is explicitly overruled by the application during the BPEM job creation. For further information about authorization and authorization groups, see chapter 3.2 *Authorization*.

For more information how to create installation-specific business process areas, see chapter 3.6.1 *Define Business Process Areas for Message Management*.

3.3.3 Activate Business Processes for Message Management

In this activity, you override the Customizing for business process areas for specific business process codes to activate or deactivate the monitoring of the business process by BPEM.

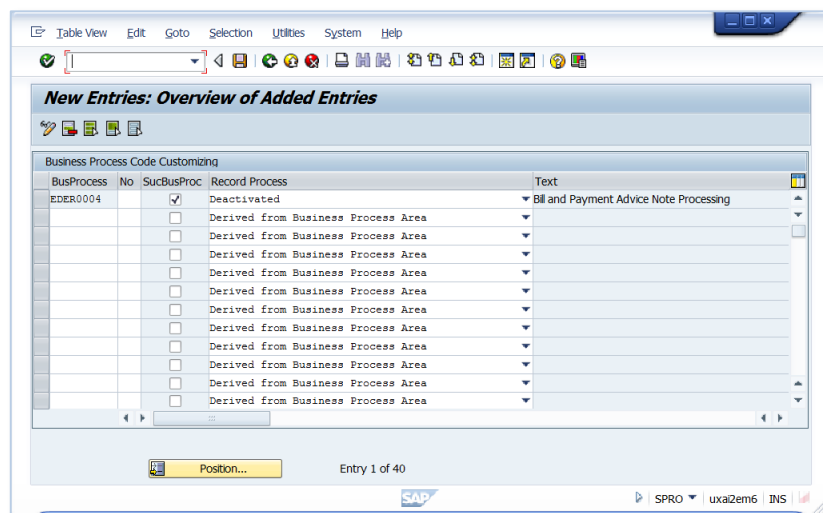


Figure 3-6 Override Customizing for a Business Process Code

You select the business processes using the input help. By selecting one of the available parameters in the *Record Process* field, you specify the type of monitoring for the selected business processes. You do this specification by either adopting or overriding the settings from the related business process area. This also applies for business processes defined in the area *Specifications for Customer-Defined Business Processes and Messages*.

You can choose one of the following options:

- **Derived from Business Process Area:** Specifies that the same procedure is followed for the business process that is specified for the related business process area.
- **Message management activated:** Business process-specific and business object-specific messages are written to the application log for a later evaluation by BPEM. This activation has an effect only if *Message Management Activated* or *Message and BW Process Management Activated* is specified for the related business process area.
- **Message and BW process management activated:** In addition to the second setting, the information relevant for process management is transferred to the Business Information Warehouse here. However, this function is currently only supported for certain business process areas in the SAP for Utilities industry solution (*BW relevant* indicator is marked) and if *Message Management Activated* or *Message and BW Process Management Activated* is specified for the related business process area.

- **Deactivated:** You can evaluate the application logs of the jobs concerned; however, BP EM does not recognize the business processes and business objects processed.

You may control with the **SucBusProc** indicator if you want to create clarification cases also for successfully processed business objects. Mark this flag if you want to activate this function.

SAP does not deliver any entries for this activity. This means that automatically monitoring takes place for all business processes as specified for the corresponding business process areas in the activity *Activate Business Process Areas for Message Management* unless these settings are overruled by the settings in this activity.

 **Example 1:** For the business process area **EDER IS-U Deregulation**, it has been specified that message management is activated. However, you do **not** want to have the business process **EDER0004 Invoice/Payment Advice Processing** monitored by BP EM. To deactivate monitoring for this specific business process, enter business process **EDER0004** in the Customizing activity *Activate Business Processes*, and in the *Record Process* column, choose the value **Deactivated**.

 **Example 2:** For the business process area **EBI - IS-U Billing**, it has been specified that message management is deactivated. However, you want to have the business process **EBI00001 - Automatic Billing** monitored by BP EM. To activate monitoring for this specific business process, enter business process **EBI00001** in the Customizing activity *Activate Business Processes*, and in the *Record Process* column, choose the value **Message Management Activated**.

3.3.4 Suppression of Messages for Log Preparation

In this activity, you define how the system processes messages during the BP EM preparation and clarification case generation steps.

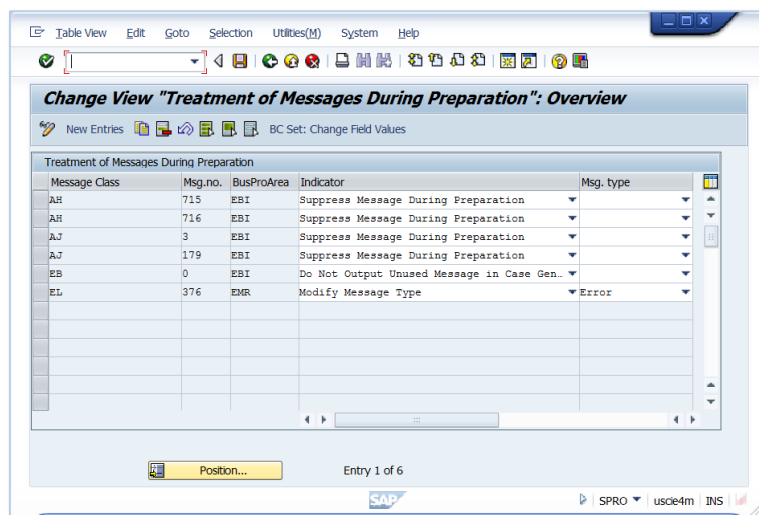


Figure 3-7 Treatment of Messages

The following options are available and are explained in more details in the following chapters:

- **Suppress message during preparation:** BP EM simply ignores the message during the BP EM preparation step.
- **Modify message type:** BP EM changes the severity level of a message (message type) during the BP EM preparation step.

- **Do not output unused message in case generation:** BPEM does not report this message as being unused during the BPEM clarification case generation step if it does not match with any of the triggering messages of a clarification case category.



BPEM never changes the source messages during its analysis. The settings in this activity affect only the BPEM preparation step. If you recognize that a Customizing entry has not lead to the expected result after carrying out the BPEM preparation step you can simply change or remove the entry, reverse the BPEM preparation step and carry it out again.

3.3.4.1 Suppress Messages during Preparation

This option allows you to specify messages BPEM shall not use in an error analysis and are not considered further during the BPEM preparation step. This can be applicable if a program writes messages that are not very important hence do not require a further analysis by BPEM. Even though it is not possible to prevent business processes to store such messages into their application logs, you can exclude these messages from being processed by BPEM.

By default, BPEM suppresses all message if only special characters can be found in either message text or message variables (for more details, see chapter 5.8.1.1 *BAdI Definition BADI_EMMA_JOB_ANALYSIS*).

Repeat the following steps for each message you want to suppress or simply have it ignored by the BPEM preparation step:

- Specify the message class and message number of the message.
- Enter a business process area if you want to suppress a message only within this specific business process.
- Choose the **Suppress message during preparation** indicator.

3.3.4.2 Modify Message Type

This option allows you to change the severity level of a message (message type) during the BPEM preparation step minding that the BPEM clarification case generation process ignores success and information messages.



You may want to change the severity level of a message during the BPEM preparation step to create a clarification case based on a message that was initially created by the application as a success or information message. After changing the severity level from success or information to warning or error, the BPEM clarification case generation step considers this message for the check if it matches with any of the triggering messages of a related clarification case category.

Repeat the following steps for each message for which you want to modify the severity level during the BPEM preparation step:

- Specify the message class and message number of the message.
- Enter a business process area if you want to suppress a message only within this specific business process.
- Choose the **Modify Message Type** indicator

3.3.4.3 Do not Log unused Messages in Case Generation

This option allows you to control the logging of a message if it is not being used to create a clarification case during the BPEM clarification case generation step. If the specified message does not match with any of the triggering messages of the related clarification case categories but further messages of the same business object matched, a message is logged as not being used in the application log of the clarification case creation process. If you set this indicator, a related message is not stored in the application log of the clarification case generation step.

 The logging of such a message can be useful if you want to control the clarification case generation process and this particular message could help to refine clarification cases. On the other hand, you may want to exclude the same message from being logged as not used, if it does not help to refine clarification cases.

Repeat the following steps for each message you do not want to be logged as unused during the BP EM clarification case generation step:

- Specify the message class and message number of the message.
- Enter a business process area if you want to suppress a message only within this specific business process.
- Choose the **Do not Output unused Messages in Case Generation** indicator.

3.4 Specifications for Generating Clarification Cases

3.4.1 Maintain Number Range Intervals for Clarification Cases

The system uses internally assigned unique clarification case numbers. In this activity, you define the number range intervals of the EMMA_CASE number range object that the system accesses when creating new clarification cases. You have to create at least one internal number range interval. BP EM does not support an external number assignment.

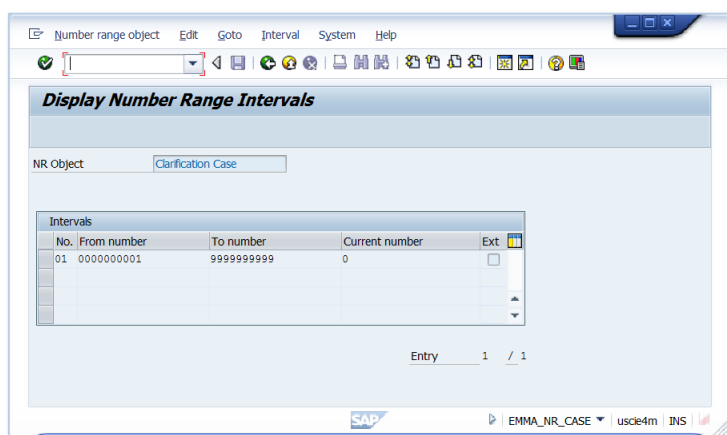


Figure 3-8 Intervals of EMMA_CASE Number Range Object

3.4.2 Define Clarification Case Types

In this activity, you define the clarification case types with their corresponding descriptive texts you can use to categorize clarification cases. Clarification case types are used for information and administration purposes only, for example, to select created clarification cases by its type rather than by the allocated business process area and business process code.

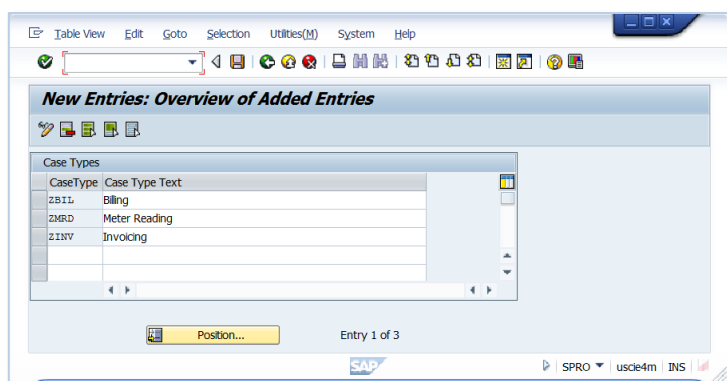


Figure 3-9 Examples of Clarification Case Types



All manual clarification cases created from the interaction center could share a common clarification case type for a quick identification.

3.4.3 Create and Maintain Clarification Case Categories

A clarification case category is always bound to a business process and is used as a template for the creation of clarification cases. You can maintain one or multiple clarification case categories per business process code.

For each clarification case category, you define conditions for the system to decide when a particular clarification case category shall be used as a template. A clarification case category considers different kind of exceptions of a business process where each exception manifests in one or more particular error or warning messages. These messages, the so-called triggering messages of a clarification case category, build the basis for the creation of a clarification case. In addition, you define solution processes to be attached to the created clarification case to support the agent to identify the reason for the exception and resolve it.

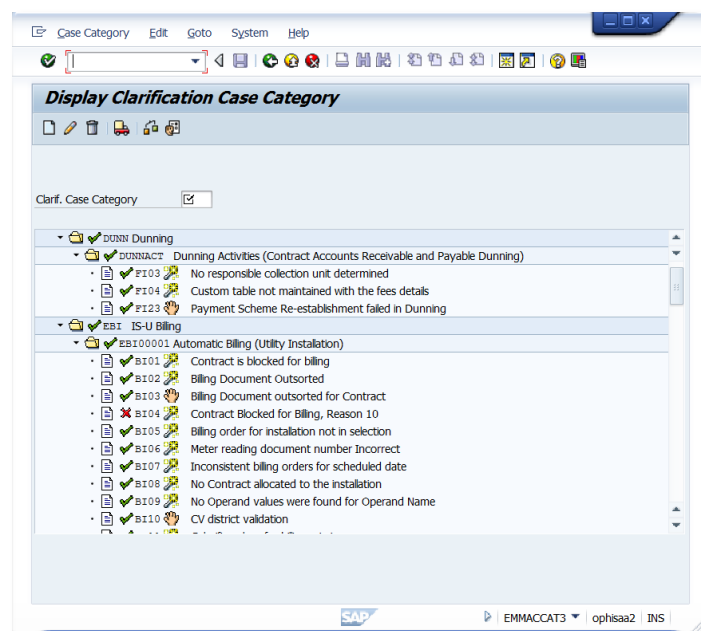


Figure 3-10 Overview Clarification Case Categories

The system displays the list of the defined clarification case categories sorted by business area and business process code. Push the  button (*Selection Tree for View*) to toggle between a list of business areas for which at least one clarification case category is defined and a list that shows all business areas regardless if a clarification case category is already allocated. You identify a clarification case category by its 4-character identification and a descriptive text. The displayed symbols have the following meaning:

Symbol	Meaning
	Business Area: business area is activated Clarification Case Category: clarification case category is activated
	Business Area: business area is deactivated Clarification Case Category: clarification case category is deactivated
	Automatic Clarification Case Category
	Manual Clarification Case Category

Figure 3-11 Status of Clarification Case Categories and corresponding Symbols

You can display and maintain an existing clarification case category after a double-click on the clarification case category. In addition, you can add a clarification case category to a transport request at any time in pushing the  button on the application toolbar.

 You cannot save a clarification case category if it contains errors unless you marked the clarification case category as *Deactivated*. You may push the  button at any time to check if the clarification case category is complete, consistent and free of configuration errors. You find the *Deactivated* flag on the top of the maintenance screen (see red box in the figure below).

The table below shows an overview of transactions related to the clarification case category maintenance that you may use as shortcuts for a clarification case category maintenance:

Transaction	Description
EMMACCAT1	Create Clarification Case Category
EMMACCAT1M	Create Clarification Case Category from Message
EMMACCAT2	Change Clarification Case Category
EMMACCAT3	Display Clarification Case Category
EMMACCAT4	Delete Clarification Case Category
EMMACCAT5	Add Clarification Case Category to Transport Request

Figure 3-12 Transaction Codes to maintain Clarification Case Categories

You use the EMMACCAT1M transaction to create a clarification case category based on a message. For this, you enter the business process code for which you want to create a clarification case category as well as a message class and message number on the selection screen. After pushing the  button, the system generates a clarification case category based on the entered business process code and message information. The system enters the given message as the triggering message, copies the message short text to the description field of the newly created clarification case category. In addition, the system copies the message short and long text to the fields describing the later created clarification case and help text for the agent.

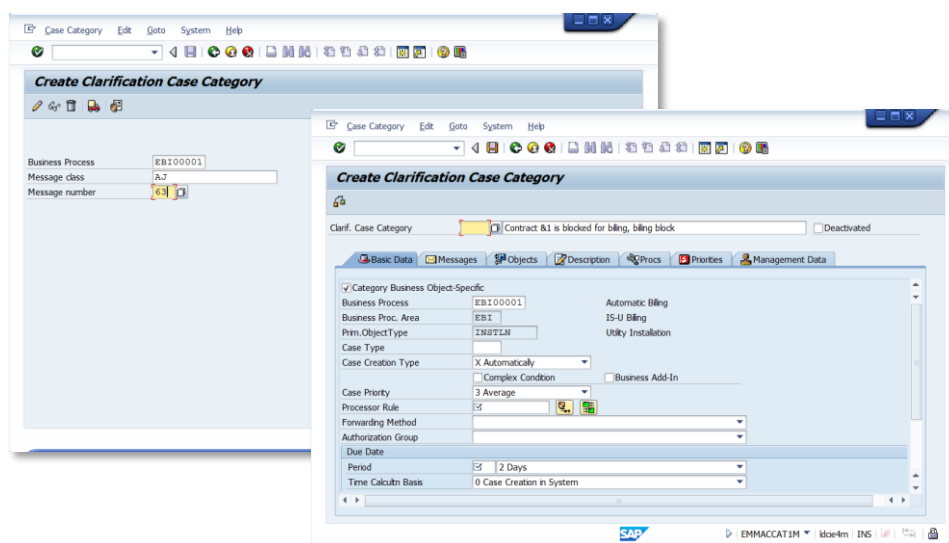


Figure 3-13 EMMACCAT1M Transaction

3.4.3.1 Screen *Basic Data*

On the screen *Basic Data* you enter general data regarding the new clarification case category. The following fields are available:

Figure 3-14 Maintenance Clarification Case Category: Screen *Basic Data*

Category Business Object-Specific

You commonly define clarification case categories specifically as a business exception of a particular business object; however, you can define clarification case categories also as being independent of the context of a business object.

If you do not mark this field - hence identify the clarification case category as a business object independent clarification case category - and the messages of the business exception match with the conditions of the clarification case category, the system creates only one clarification case using the data of the first occurrence of the error messages. In addition, it suppresses the further creation of clarification cases for the same triggering of a condition. This means that once a clarification case was created for a particular clarification case category, the same clarification case category is removed from the list of possibly eligible clarification cases thus preventing a second creation of clarification cases for the same clarification case category. In this case, only one single clarification case is created due to the same error message instead of creating repeatedly similar clarification cases.

The usage of business object independent clarification case category is advisable if, for example, the exception is caused by a customizing problem affecting many business objects in the same way and the exception can be resolved centrally, for example, in correcting Customizing, instead of remediating each reported business exceptions with an own business object specific process.

Business Process

Enter the business process code for which the new clarification case category should be considered as a template to create a clarification case. The system derives the business process area and the primary object type from the Customizing of the entered business process code.

Business Process Area

The system derives the business process area from the entered business process code and is displayed for information purposes only. The field is not available for input.

Primary Object Type

The system derives the primary object type from the entered business process code and is displayed for information purposes only. The field is not available for input and only visible if the *Category Business Object-Specific* checkbox is marked.

Case Type

Enter a clarification case type to improve the later selection of clarification cases. This field is not a mandatory field. For further information on clarification types see chapter 3.4.2 *Define Clarification Case Types*.

Case Creation Type

You can make a clarification case category available only for an *automatic* creation of clarification cases by the system or restrict the usage of a clarification case category for a *manual* creation by an agent or by a BAPI. If the clarification case category shall be used for an automatic creation of clarification cases, you can support the decision for the usage of a particular clarification case category by either a complex condition or an installation-specific business add-in (BAdI). Commonly, you allow an automatic creation of clarification cases simply based on triggering messages without a further validation or restriction of a complex condition or with a BAdI implementation (for further details on triggering messages, see chapter 3.4.3.2 *Screen Messages*).

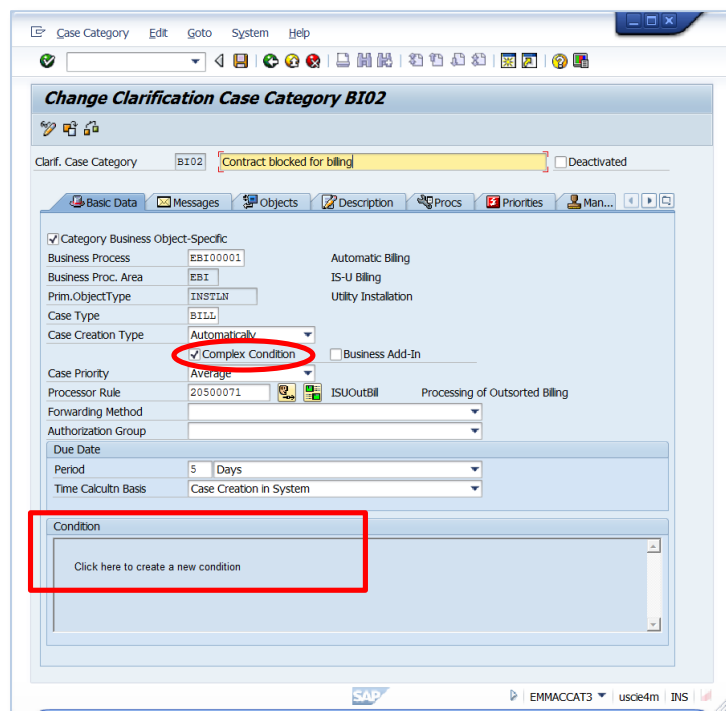


Figure 3-15 Define a Complex Condition as Selection Criteria

Complex Condition

You mark the *Complex Condition* field if you want to specify further conditions for the automatic creation of a clarification case. The system exits the processing of the clarification case category if the entered expression is false. In addition to the condition, you can still define triggering messages. If there are such supplemental triggering messages, the system checks first if these triggering messages match with the messages of the business

exception. Only if there is a match, the system processes the entered condition subsequently. If there are no triggering messages defined, the system just processes the entered condition.

If you chose the option *Complex Condition*, the *Condition* window becomes available. After you click on the *Condition* subscreen you can enter the condition in the condition editor.

 You can differentiate clarification case categories by the company code. The company code field itself is available in the current contract of the primary business object *Utility Installation*.

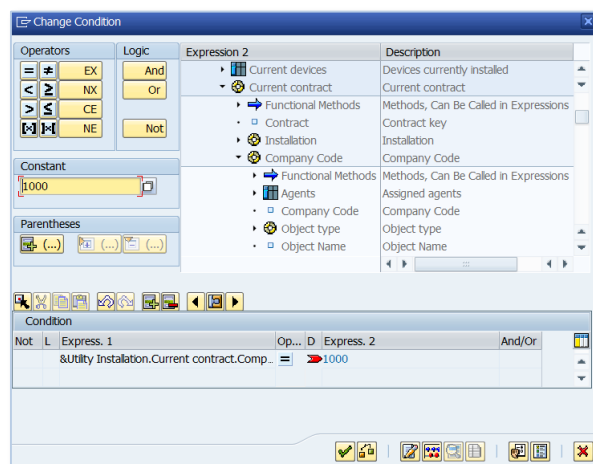


Figure 3-16 Condition Editor to enter a Complex Condition

Upon completion of your input, the system displays the condition in the *Condition* window.

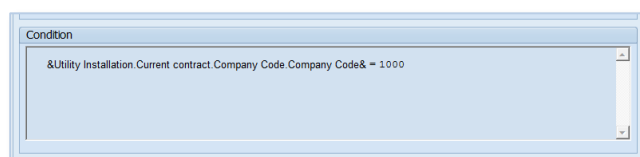


Figure 3-17 Example of a Complex Condition

 Operators **EX** and **NX** check for the existence (or non-existence) of a business object reference for which a valid object reference must be accessible as an expression solely of the `_EMMA_MainObj` object type (all other possibly defined business objects in the clarification case container are not determined at the time the condition is processed). The system highlights object references of the business object repository (BOR) by a  symbol in the expression tree. *Expression 2* remains unused when either **EX** or **NX** operator is entered.

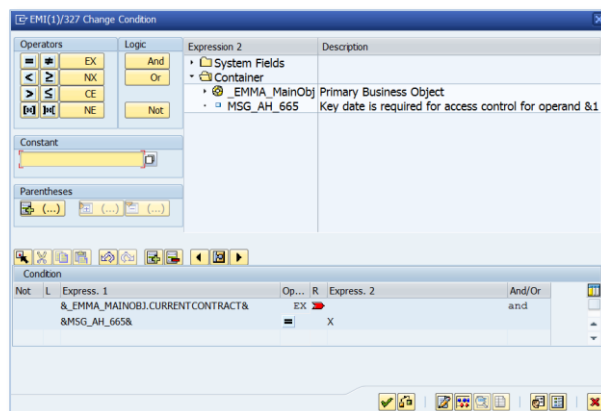


Figure 3-18 Condition to check for an Existence of Objects



If you want to check for the (non-) existence of a non-triggering message use the `=` operator and `X` or `SPACE` as *Expression 2* instead of the `EX` or `NX` operators.



The usage of a complex condition may result in system performance issues particularly if you define especially complex conditions. Consider performance tests if you need to implement complex conditions.

Business Add-In (BAdI)

You mark the *Business Add-In* field if you want to implement the `DETERMINE_CASE` method of the `EMMA_CASE` BAdI to implement an installation-specific condition in ABAP for a creation of a clarification case. The system exits the processing of the clarification case category if the BAdI method raises the `NO_CASE` exception (for more details about the `DETERMINE_CASE` method, see chapter 5.8.3.5 *Installation-specific Determination of Clarification Case Category*). In addition to the implementation of the BAdI, you can still define triggering messages. If there are such supplemental triggering messages, the system checks first if these triggering messages match with the messages of the business exception. Only if there is a match, the system processes the BAdI subsequently. If there are no triggering messages defined, the system just processes the BAdI.

Case Priority

The priority of a clarification case denotes the urgency until a later clarification case should be resolved. Possible values are *Very High*, *High*, *Average*, *Low* and *Very Low*.

Due Date

The due date defines the period during which the created clarification case should be resolved. In the *Deadline* field, enter a value between 0 and 999, and in the field next to that, enter the required unit of time (seconds, minutes, hours, days, months, years). In the *Calculation Basis* field, specify how the due date is to be calculated. The system calculates the due date of a clarification case on either the time stamp when the clarification case was created (*Case Creation in System*) or the time stamp the application identified the business exception and logged it (*Appearance of Case*).

Processor Rule (Agent Determination Rule)

Enter a predefined rule for determining the agents to work on clarification cases that are based on the given clarification case category. If the entered processor rule requires input data from the clarification case container push the  button to define the data flow. For more details about agent determination, see chapter 6.4 *Agent Determination*.

Alternatively, you simply leave this field empty and use clarification case work lists with a selection by, for example, clarification case category or clarification case type to create specific clarification case worklists for your various business departments.

Forwarding Method

The system processes the forwarding method always after the creation or change of a clarification case. Enter a forwarding method if you want to take special action, for example, forwarding the clarification case to a supervisor for confirmation. For more details about forwarding methods and use cases, see chapter 6.4 *Agent Determination*.

Authorization Group

Enter an authorization group if you want to restrict the possible actions of agents on clarification cases based on the given particular clarification case category. This is not a mandatory field. For more information about the required authorization when accessing a clarification case, see chapter 3.2.3 *Authorization for Clarification Case Processing*.

3.4.3.2 Screen Messages

On the *Messages* screen, you define the message pool of the clarification case category. You must enter at least one message as triggering message if you define the clarification case category with creation type *Automatically* except you use either option *Complex Condition* or *Business Add-In*. Commonly, you enter not only a single message but multiple messages, a so-called message group, to make your clarification case category more specific to a particular business exception.

You add new messages to the message pool by pushing the  button. The order of messages in the list does not matter. However, if you want to insert a message above an existing one, mark an existing message before you push the  button. Otherwise, the system appends a new message line. Push the  button to remove a message from the list. Enter the message class and message number of the message in the respective fields of the new line.

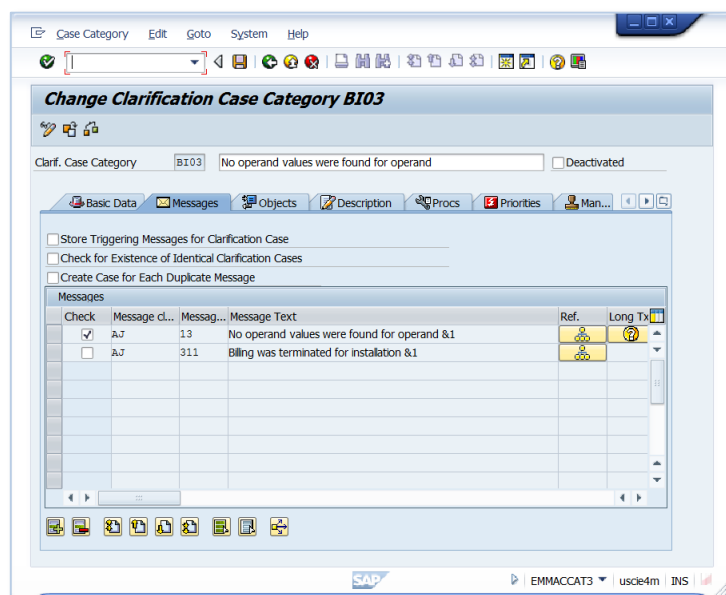


Figure 3-19 Maintenance Clarification Case Category: Screen Messages

The  symbol next to the message indicates that message variables of the message reference to business objects or table fields. The system displays the references of the related message when you push the  button. If the  symbol is not displayed, the message either does not contain any message variable or none of the message variables refers to a business object or table field. For more detail how to define references between message variables and business objects or table fields, see chapter 3.6.5 *Define Business Objects for Customer-Defined Messages*.

You mark the *Check* checkbox of each message that must match with one of the error or warning messages created by a business exception during the later clarification case creation process. These messages are called triggering messages because only these messages of the message list are checked against the messages of the business exception and decide the further processing of a clarification case category. Only if all triggering messages match with messages of the business exception, the system continues the processing of the actual clarification case category.

 During the clarification case creation process, the system compares the defined triggering messages only with messages of the business exception as of message type *Error* and *Warning*. The system ignores messages as of message type *Success* and *Information*.

Even if a message is not marked as triggering message you can still use the information of its related message variables during the further processing of the clarification case category, for example, as input to a data flow for a creation of business objects in the clarification case.



The system creates a clarification case regardless if all entered non-triggering messages match with messages of the business exception. In consequence, the business objects associated with message variables of a missing message will not always be available, for example, as input for a solution process.

Store Triggering Messages for Clarification Case

You mark this checkbox if you want to store the triggering messages in the message list of the clarification case.



This checkbox is marked automatically when the *Check for Existence of Identical Clarification Cases* checkbox is marked to be able to identify duplicate cases by their triggering messages what is only possible if the triggering messages are stored.

Check for Existence of Identical Clarification Cases

You must mark this checkbox if you want to prevent the creation of a further clarification case if a previously created and matching clarification case is not yet in status *Completed* or *Confirmed*. In this context, clarification cases match if they hold the same information for the clarification case category, main business object, triggering messages and an identical content of their related message variables.



Use this option with care since this check slows down the creation of clarification cases and therefore has a negative impact on the system performance. You should decide on a case-by-case basis using this option rather than using this option for all scenarios.

Create Clarification Case for each Duplicate Message

One business exception may occur with multiple groups including the same triggering messages. While checking for matching triggering messages the system can identify the groups and allow the creation of duplicate clarification cases if requested.



The figure below shows three **AJ 98** warning messages each reporting an outsourced document after processing the **48562** utility installation. If you entered the **AJ 98** message as triggering message and marked the *Create Clarification Case for each Duplicate Message* flag, the system creates three clarification cases with one of the **AJ 98** messages each. If you do not mark *Create Clarification Case for each Duplicate Message* flag, the system creates only one single clarification case including all three messages.

BusProcess	Obj. Type	Utility Installation	MC ID	No.	Message Text	LTxt BOBj.
EBI00001	INSTLN	0000048562	AJ	98	Billing document 30628 for contract 52426 has been outsourced	?
	INSTLN		AJ	98	Billing document 30629 for contract 52426 has been outsourced	?
	INSTLN		AJ	98	Billing document 30630 for contract 52426 has been outsourced	?
	INSTLN		AH	371	Bill. order for inst. 48562 sch.bill.date 30.04.2016 bill.proc. 01 not in selection	?

Figure 3-20 Message List with multiple Messages for a single Business Exception



Use the *Create Clarification Case for each Duplicate Message* option with care since the system holds the messages of all processed business exceptions in memory to allow an efficient processing of the clarification case creation process. You should decide on a case-by-case basis rather than using it for all implemented scenarios. Otherwise, you take the risk of a program cancellation due to a high memory consumption of the BPME application when generating clarification cases.



The system marks this option by default for case categories with a complex condition to force the processing of the complex condition during the duplicate check instead of just comparing the triggering messages with the message group.

3.4.3.3 Screen Objects

On the *Objects* screen, you create the elements representing objects and related data as input for solution processes, special rule definitions or simply to be displayed in a clarification case. The elements are stored in the so-called clarification case container.

Usually it is sufficient to push the *Generate from Message Pool* button  to complete the list of business object and message variables from the information already allocated to the clarification case category.

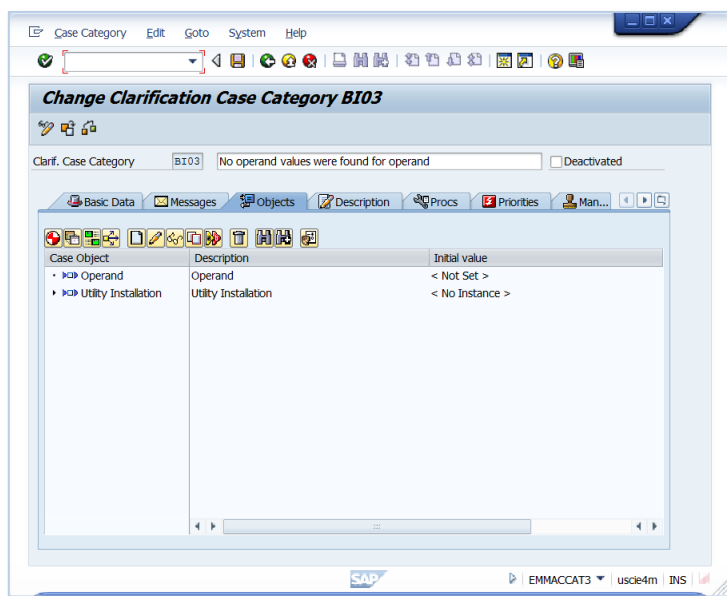


Figure 3-21 Maintenance Clarification Case Category: Screen *Objects*

Push the  button to create further elements of the clarification case container, enter the element name and either the name of a BOR object type or a reference to an ABAP dictionary field and eventually the  button to add the element to the clarification case container.

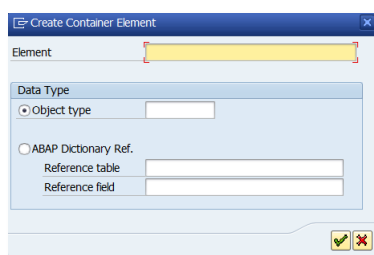


Figure 3-22 Create an Element of the Clarification Case Container



For technical reasons, there are some restrictions for container elements with a data type reference to a structure or a table:

- The total length of the field must not exceed 70 characters.
- Only alphanumeric data types are permitted

You complete the definition of the container elements by pushing the  button and define the mandatory data flow for all elements in the clarification case container. Figure 3-23 shows a clarification case container and a valid data flow definition.



An object of the clarification case container is only added automatically to a clarification case only if there is a data flow associated with the object. However the data flow for these objects does not necessarily require a source (see the data flow for the `&TEXT&` element in the example below).

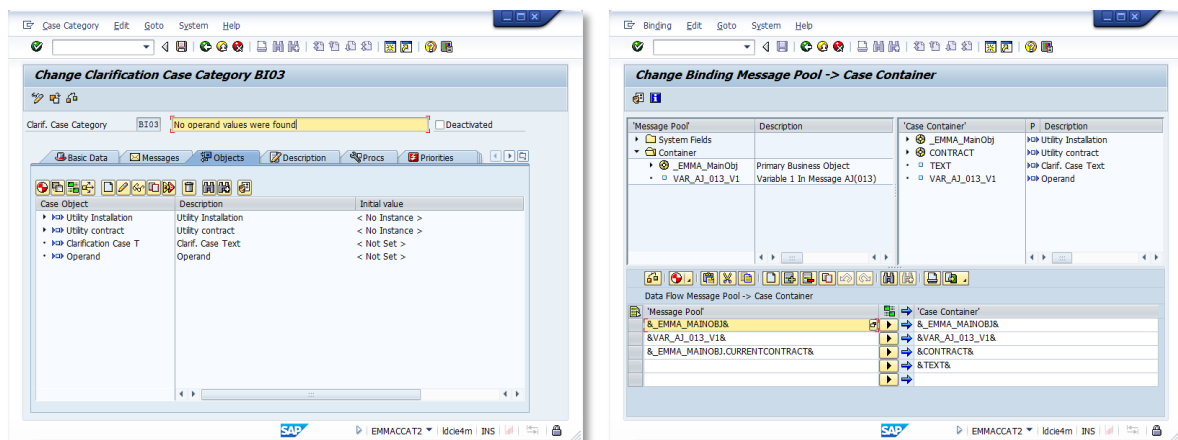


Figure 3-23 Data flow from the Message Pool to the Element of the Clarification Case Container

Push the button to display a where-use-list of the selected container element. The where-use-list supports you in analyzing data flow issues.

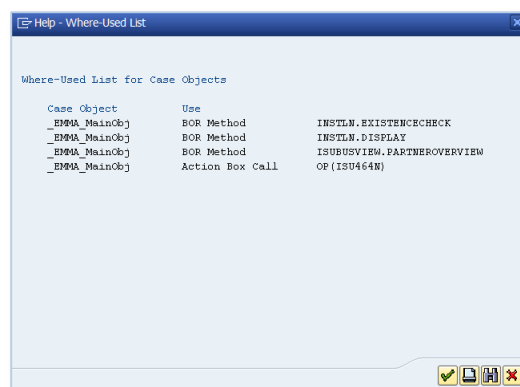


Figure 3-24 Where-Used-List of an Element of the Clarification Case Container

The following two examples describe when you should consider creating additional objects in the clarification case container to be used in the creation or processing of a clarification case:



The required attribute or information for a solution process is not available in the message pool. In such a scenario, you create an additional object in the clarification case container. In addition, you need to implement the `COMPLETE_CASE` method of the Classic Business Add-In `EMMA_CASE` to complete the added object with the missing object key or object information that could not be derived dynamically from the further objects in the clarification case container (see also chapter 6.9 *Add Business Objects to a Clarification Case dynamically*).



You want to store additional information in a clarification case to transfer data between automatic solution processes.



You can change the predefined order of the business objects the system displays in a clarification case or even hide business objects in implementing the `ADJUST_ALVOBJ_OUTPUT` method of the BADI `EMMA_CASE_TRANSACTION` BAdI.

3.4.3.4 Screen Description

On the *Descriptions* screen, you enter a detailed documentation to support the agent resolving the clarification case efficiently. The description should include a problem description and the steps to resolve the issue.

The brief clarification case description in the *Case Text* field and the long text may include variables as references to business objects, for example, derived from the message variables of the message pool. The system replaces the entered references by the key of the referenced business object during the clarification case creation. Please mind, that you must enclose variable names by & characters.

 The variable name of the message variable `V1` of the message with message class `AJ` and message number `013` is `&VAR_AJ_013_V1&`. During runtime, the system replaces this variable by the reported operand name.

 The variable name of the main or primary business object (as derived from the business process code) is `&_EMMA_MAINOBJ.NUMBER&`. During runtime, the system replaces this variable by the current utility installation number.

 The description field of a clarification case is only a 60-character field. However, the additional space of the description field in the clarification case category is available to add variable names to be replaced by the system by actual values, such as the business partner ID where the resulting required space is much less than the used variables names.

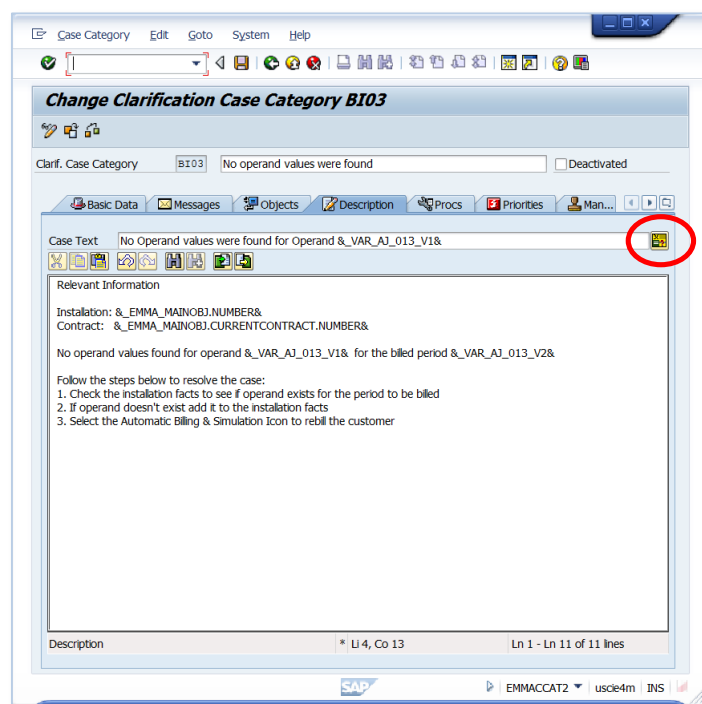


Figure 3-25 Maintenance Clarification Case Category: Screen Description

You can determine the available variable names easily in pushing the  button. Push the  button to toggle between the display of business object names and their technical representations. Once you identified the business object or one of its parameters, double-click on it to store the selected item in the copy & paste buffer. From there you insert the variable name of the selected items anywhere in the text by pushing the  button.

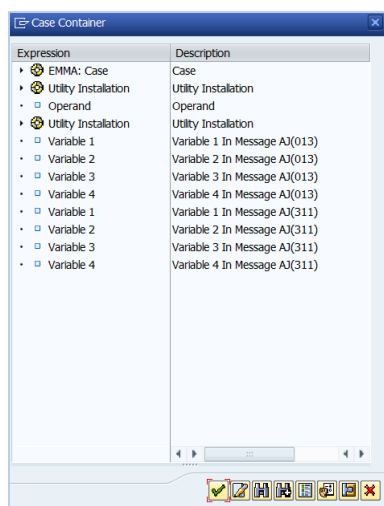


Figure 3-26 Example for a List of available Business Objects

The figure above shows a list of available business objects that are available in a sample clarification case container. On top of the list, you can see the business object for the later clarification case followed by the name of main business object that is attached to the entered business process code. Subsequently, there is a list of message variables belonging to the two messages of the message pool.

If required, you can take advantage of the BPEM enhancement design to make a more detailed documentation available to the agent in adding an installation specific button to the toolbar of the clarification case maintenance transaction. With a related additional implementation, you can display documents or call the web browser with a link to a company portal web page to display a sophisticated case specific documentation and a detailed step-by-step description how to resolve it. For more details about this enhancement, see chapter 5.8.2.3 *BAdI Definition BADI_EMMA_CASE_TRANSACTION*.

3.4.3.5 Screen Processes

On the *Processes* screen you define the list of solution processes that shall help the agents to resolve the clarification case. A solution process is represented in a clarification case as a button labeled with a symbol and a descriptive text. Solution processes can be executed either automatically or manually.

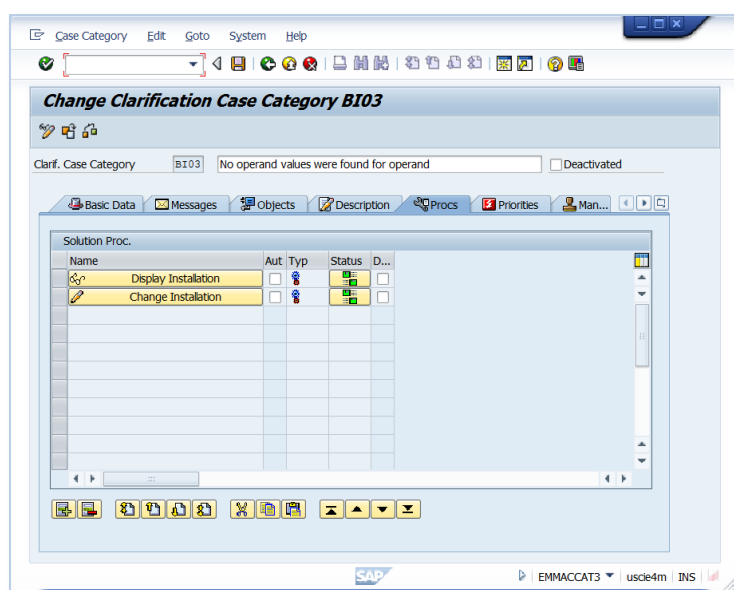


Figure 3-27 Maintenance Clarification Case Category: Screen Processes

You add new solution processes by pushing the  button. You can define a solution process either, first, as a call of a Business Object Repository (BOR) method, second, as an execution of a workflow task or, third, as a launch of a front office process (FOP) by means of a transaction of an action box configuration as defined in the Customer Interaction Center (CIC) framework. You should define the same activities as solution processes the agents already use in their daily work. The following figure shows the screen with, for example, a Customizing to call the `DISPLAY` method of the `INSTLN` BOR object (*Utility Installation*).

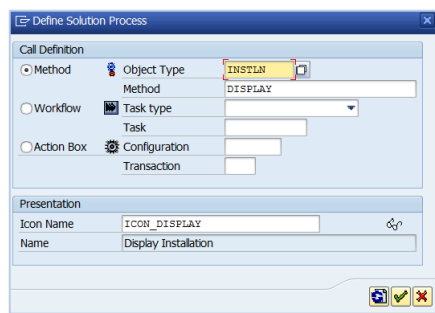


Figure 3-28 Example for a Definition to call a Method of BOR Object

In the same order in which you define the solution processes, the solution processes are presented to the agent in the action box of a clarification case. You can change the order of the offered solution processes with the *Entry to the Top*  button, *One Line Up*  button, *One Line down*  button and *Entry to the Bottom*  button. You must mark the solution process before shifting the solution process to another position in the process list.



You can change the predefined order of the solution processes the system displays in a clarification case or even hide solution processes in implementing the `ADJUST_ALVPROC_OUTPUT` method of the `BADI_EMMA_CASE_TRANSACTION` BAdI (see chapter 5.8.2.3 *BAdI Definition BADI_EMMA_CASE_TRANSACTION*).

The following fields are available on the *Processes* screen:

Automatic Solution Process

If you mark the *Automatic Solution Process* field, the system may carry out this solution process as long as the clarification case is either in status *New* or *In Process*. Of course, no other agent may lock the clarification case when the system attempts to execute its automatic solution processes. Regardless if a solution process is marked as *Automatic Solution Process*, the agent can always manually trigger the execution of an automatic solution process while maintain the clarification case. For more details about automatic solution processes, see chapter 5.4 *Automatic Solution Processes*.

Type

This field displays the call type of the defined solution process as a call of a method of the BOR , a workflow task  or FOP of an action box transaction  of the Customer Interaction Center (CIC) framework.

Data Flow Maintenance Status

Commonly, a solution process requires importing data, for example, the number of the utilities installation that failed during the billing process. By means of a data flow definition, you logically bind a business object or a particular field to the importing and exporting parameter of the solution process for a transfer of data from the clarification case container to the called solution process and from the process back to the clarification container.

There is no need to define an explicit data flow for a solution process defined as a front office process (FOP). The system uses the available business objects of the clarification case container to automatically resolve the importing parameter of the FOP. For restrictions and further information about the usage of FOPs as solution processes, see chapter 6.11 *Usage of Action Box of Customer Interaction Center CIC*.

To define the data flow of the solution process, push the  button or  button to start the binding editor. In most cases, you can allow the system to generate the required binding automatically in simply pushing the  button. The system analyzes the importing parameter of the defined solution process and tries to find the correct corresponding items in the clarification case container. Of course, you need to check the binding proposal and correct it if required.

If you use the binding editor for the first time, push the  button to maintain the agent settings to switch, for example, to the full screen editor. Push the  button for a comprehensive documentation about the binding editor.

The data flow maintenance status of the solution process switches from  (data flow not yet defined or inactive) to  (data flow defined and active) upon successful completion of the data flow configuration.

The following figure shows the completed binding to call the `DISPLAY` method of the `INSTLN` BOR object (*Utility Installation*) that requires the utility installation business object as importing parameter.

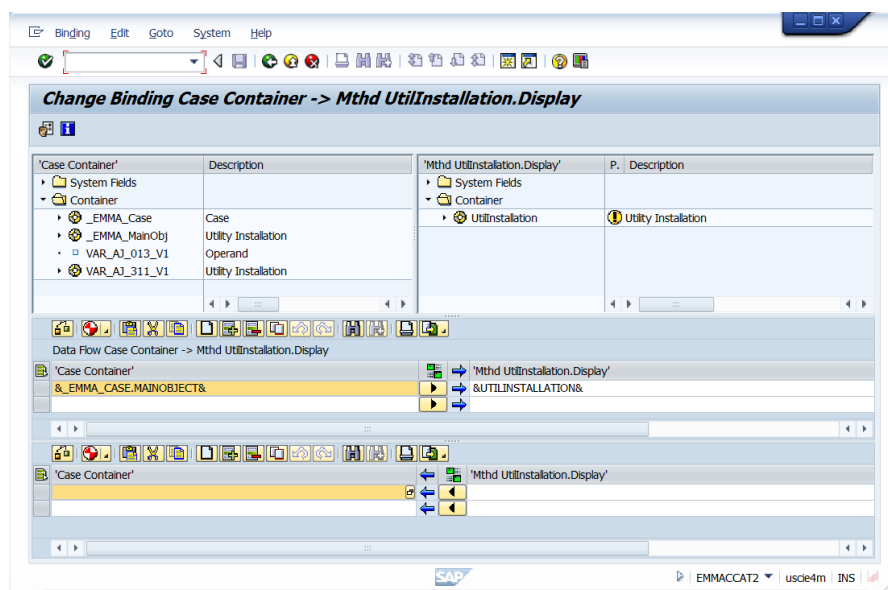


Figure 3-29 Binding Editor to enter Bindings between Business Objects

3.4.3.6 Screen Priorities

On the screen *Basic Data* (see chapter 3.4.3.1 *Screen Basic Data*) you define default values for the clarification case priority, forwarding method, authorization group and a due date. However, for clarification case categories of the case creation type *Automatically* you can redefine these settings in dependency of the later clarification case data and any data as available in the clarification case container while generating the clarification case.

 You want to allocate a different priority to the clarification case if the clarification case refers to customers with a particular account class (available in the contract account), for example, the account class of industry customers as shown in the figure below.

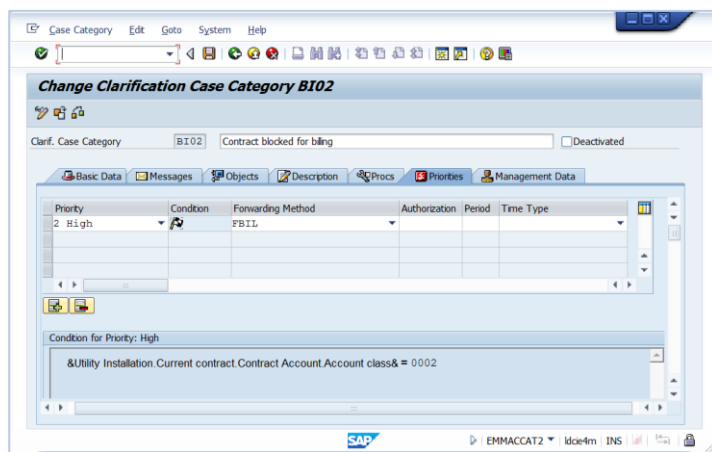


Figure 3-30 Maintenance Clarification Case Category: Screen *Priorities*

You may enter one condition for each available priority from *Very Low* to *Very High* to describe the circumstances when the system should deviate from the default setting. For this, you push the  button to create a new line on the screen. Then enter the priority to be assigned to the clarification case if the condition matches. When pushing the  button, the system requests you to click on the condition subscreen for the creation of a new condition.

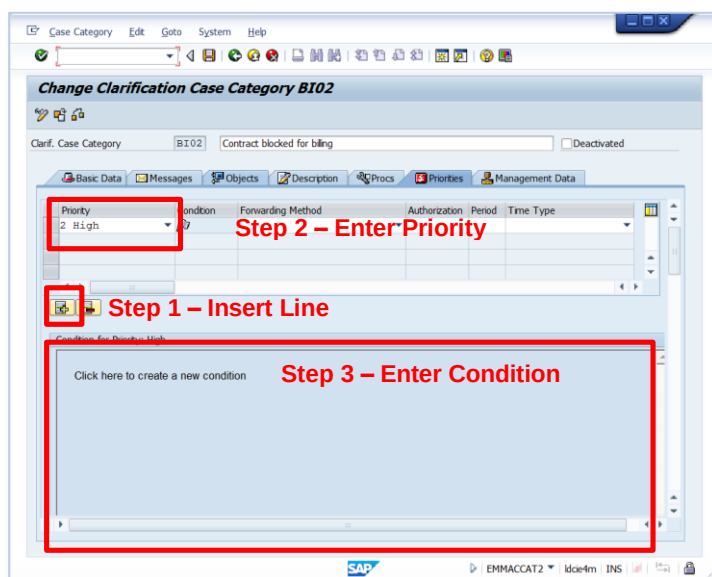


Figure 3-31 Maintenance Clarification Case Category: Enter a Condition

After you click on the *Condition* subscreen you can enter the condition in the condition editor (see figure below). Complete the condition by pushing the  button. In addition to the priority, you can also redefine the forwarding method and the period to the calculated due date.

During the automatic clarification case creation process, the system checks the entered conditions starting with the entry with the highest priority until one of the conditions matches. The system uses the first found parameter set to redefine the settings as entered on the *Basic Data* screen. If you do not enter a value for the *Forwarding Method* or *Time Type* field, the default settings from the *Basic Data* screen remain unchanged.

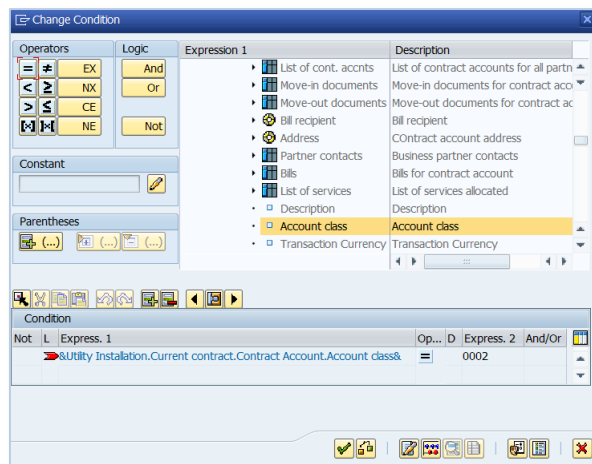


Figure 3-32 Condition Editor to enter a Complex Condition

3.4.3.7 Screen Management Data

You cannot make any changes on the *Screen Management Data* screen since only administrative data is displayed on this screen, such as creation date and time of the clarification case category and agent who created it as well as change information.

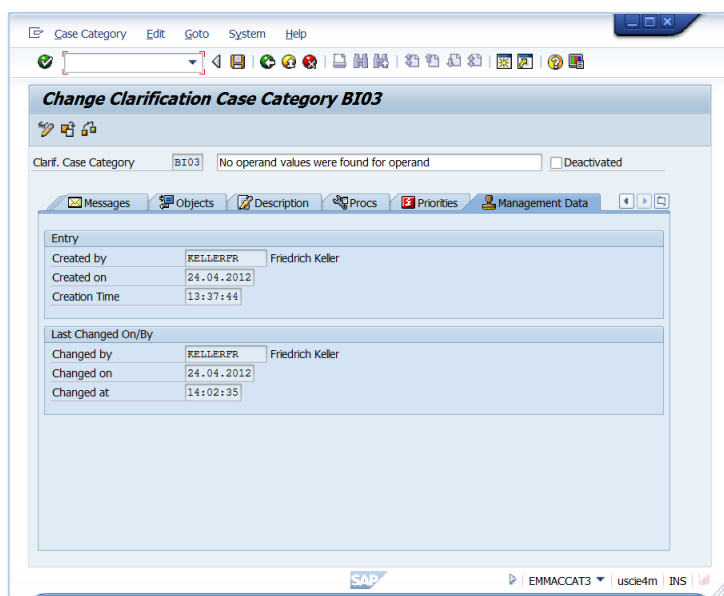


Figure 3-33 Maintenance Clarification Case Category: Management Data

3.4.4 Define Method for Forwarding Clarification Cases

In this activity, you define the methods and classes for forwarding clarification cases.

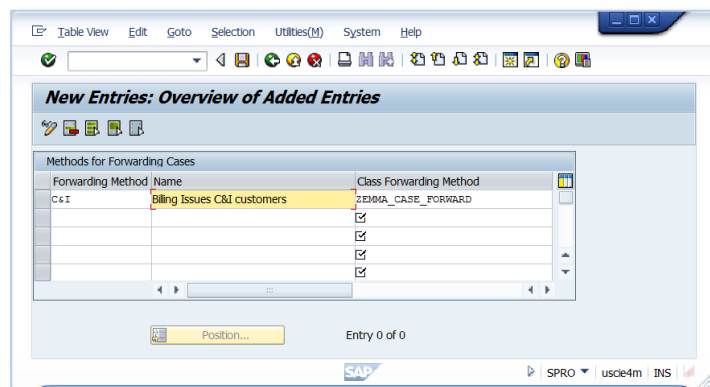


Figure 3-34 Maintain Methods for Forwarding Clarification Cases

You can forward a clarification case to the responsible business department or agent or take special action after a clarification case was created or changed. SAP provides the `IF_EMMA_CASE_FORWARD` ABAP OO interface with the `FORWARD` method for the implementation of your installation specific forwarding methods. For implementation examples, see chapter 6.4.2.

3.5 Specifications for Processing Clarification Cases

3.5.1 Generate Selection Screen for Clarification List

You can add additional installation-specific fields to the `CI_EMMA_CASE` customer include. This activity regenerates the selection screen and database selection routines of the clarification case lists to make the newly added fields available on the selection screen of the clarification list transaction `EMMACL`.

This activity, representing the `EMMACLGEN` transaction, regenerates and activates the two includes files `IEMMACL_CUSTSELEFIELDS` and `IEMMACL_DBSELECTIONCUST`. If requested, the system records the changes in a workbench transport request.

 You require the following additional authorization to regenerate the selection screen: authorization object `S_DEVELOP`, development class `DUMMY`, object type `PROG`, object name `REMMACL_SELSCRN_GENERATE`, authorization group `ABAP/4` program `DUMMY` and activity `02` (change).

3.5.2 Define Reasons for Forwarding Clarification Cases

The agent has to enter a reason when forwarding a clarification case to somebody else. In this activity, you define the reasons an agent may enter why forwarding a clarification case.

 By default, the system requests the agent to enter a valid forwarding reason when entering a new processor hence forwarding the clarification case to another agent. However, the system raises the error message `EMMA 261 Specify a valid forwarding reason` if an agent tries to save this clarification case without a forwarding reason. If you do not want to force the agents to enter a forwarding reason, create an additional entry with a blank forwarding reason. If there are scenarios where a forwarding reason is still required, you must implement the `CHECK_BEFFORE_SAVE` method of the `EMMA_CASE` BAdI to check for an entered forwarding reason.

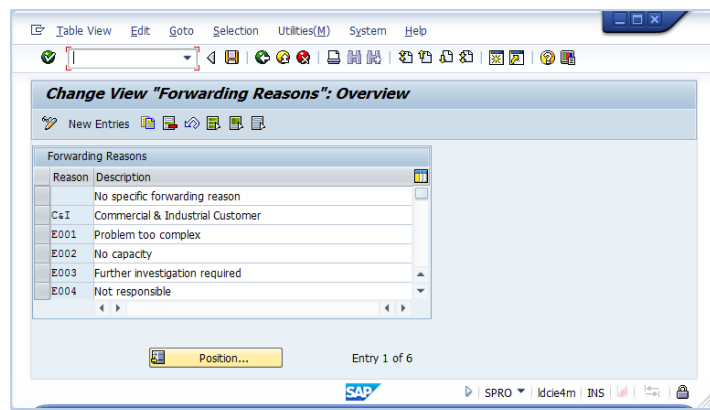


Figure 3-35 Example of Forwarding Reasons

3.5.3 Define Reasons for Reversing Clarification Cases

In this activity, you define the reasons an agent may specify for reversing a clarification case (a reversal of a clarification case is only permitted if the selected clarification case is either in status *New* or *In Process*).



By default, the system requests the agent to enter a valid forwarding reason when entering a new processor hence forwarding the clarification case to another agent. However, the system raises the error message `EMMA 371 Enter a valid reversal reason for status 'Reversed'` if an agent tries to save this clarification case without a reversal reason. If you do not want to force the agents to enter a reversal reason, create an additional entry with a blank reversal reason. If there are scenarios where a reversal reason is still required, you must implement the `CHECK_BEFFORE_SAVE` method of the `EMMA_CASE` BAdI to check for an entered reversal reason.

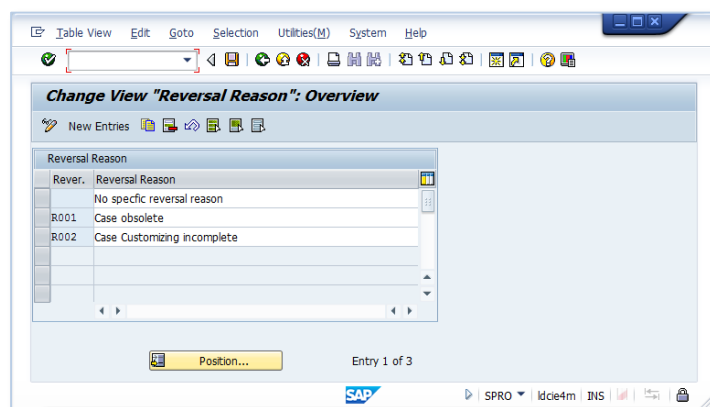


Figure 3-36 Example of Reversal Reasons

3.5.4 Maintain Shortcut Keys for Clarification Case Processing

In this activity, you define the shortcut keys to be displayed in the action box on the left side of the clarification case worklist screen (for more details, see chapter 4.3.1 *Transactions EMMACL/EMMACLS – Clarification Case List*). The agents use these shortcut keys for a quick access to clarification cases that are relevant for their work. Shortcut keys are grouped in layouts, so-called action boxes. You can define and allocate a fixed layout to an agent and allow the agents to switch between the available layouts.

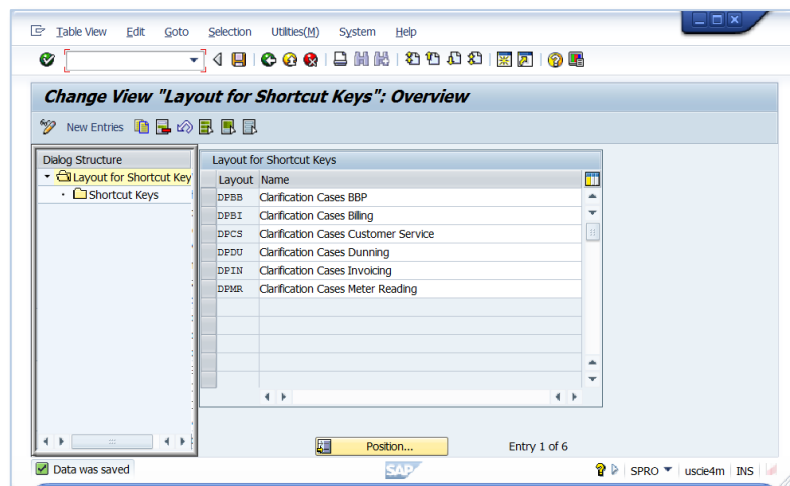


Figure 3-37 Example of Layouts

Push the *New Entries* button and enter the name of the new layout and a descriptive text if you want to create a new layout, or mark the layout you want to maintain. Push the *Shortcut Keys* activity to maintain the shortcut keys of the selected layout. For each shortcut key, you have to enter a number defining implicitly the order of the shortcut keys in the action box on the clarification worklist screen and a descriptive text to be displayed on the shortcut keys.

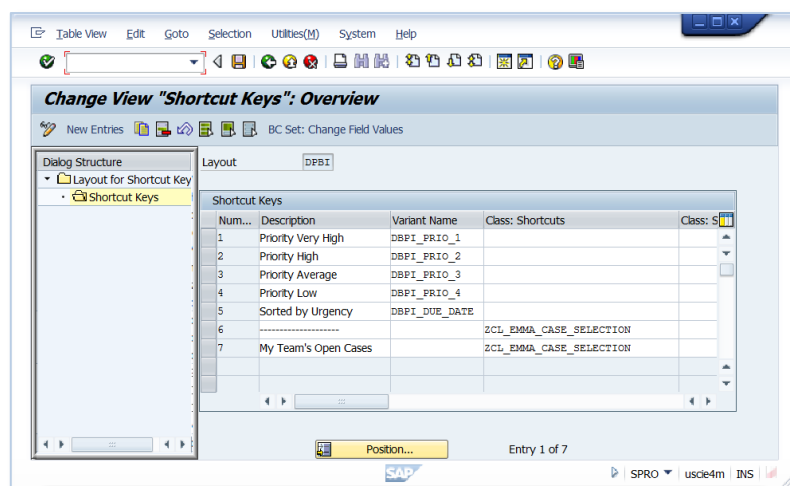


Figure 3-38 Example of Shortcut Keys

Subsequently, you define for every shortcut key how the selection of clarification cases should be performed. The following three options are available from which you must select one since the options are mutually exclusive.

- **Variant Name:** Enter the name of a previously created variant of the REMMACASELIST report. When the agent pushes the respective shortcut key, the system executes the REMMACASELIST report for the clarification cases to be selected and displayed according to the stored variant values.



You create a system variant if the name of your variant starts with the prefix CUS&. A variant with such a prefix is transported automatically. The system prompts you to enter the number of a workbench transport request when you save the variant.

- **Class for Selection Criteria:** Enter the name of an ABAP OO class that provides the selection criteria for the REMMACASELIST report by an installation-specific implementation of the DETERMINE_SELCRITERIA method of the IF_EMMA_CWL_LAYOUT interface. This enables you to circumvent the restrictions of the selection screen of the REMMACASELIST report and permits the dynamic creation of selection criteria.
- **Class for Case Selection:** Enter the name of an ABAP OO class that carries out the clarification case selection by an installation-specific implementation of the SELECT_CASES_FROM_DB method of the IF_EMMA_CWL_LAYOUT interface. This enables you to adjust the selection to your requirements in addition to the standard functions of the REMMACASELIST report.



You may allocate the same variant or ABAP OO class to more than one shortcut key.

The Customizing settings above leads to the action box as shown in the figure below where an agent of the billing department can select clarification cases according to defined or implemented selection criteria.

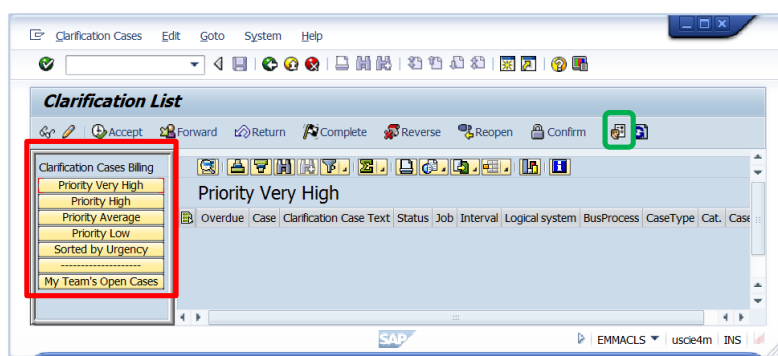


Figure 3-39 Example of an Action Box

You assign a specific action box to an agent by means of the EMMA_CLWS SPA/GPA parameter ID in the user profile of each agent. The related parameter value indicates the default layout the system uses to display the agent's clarification case worklists. The system administrator can edit the concerned user profiles with the SU01 transaction. Alternatively, users can define their default layout via the SU2 transaction or choosing *System* → *User profile* → *Own data* or changing the layout in simply pushing the  button.

3.6 Specifications for Business Processes and Messages

The texts of the Customizing activities in this chapter are slightly different in the IMG than mentioned in this document. All the activity texts refer to customer business processes, customer transactions and customer messages. This is not entirely correct. SAP delivers a configuration that is very complete but requires still installation-specific modifications or amendments by the customer. Because of this and, in addition, to improve the readability of this document, the term *Customer* has been removed in the titles of the Customizing activities in this chapter.

3.6.1 Define Business Process Areas for Message Management

In this activity, you amend your installation-specific business process areas you want to monitor with BPEM. The business process areas, for which SAP provides monitoring by BPEM, are contained in the delivery.

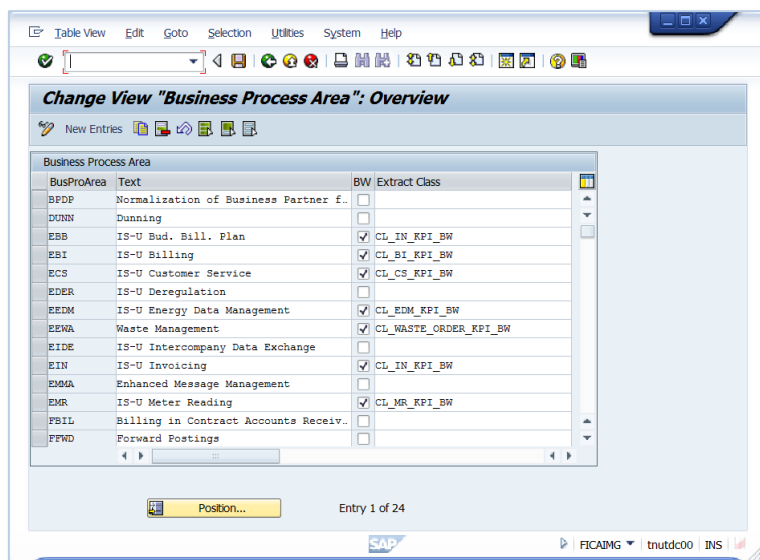


Figure 3-40 Maintenance of Business Areas

The **BW** field indicates whether process management relevant information of a business process area can be transferred to the Business Information Warehouse (BW). SAP delivers extraction classes and BW content only for certain business process areas in the industry solution SAP for Utilities. If you mark the **BW** field, you must also enter an ABAP OO class with the ABAP OO interface `IF_EMMA_KPI_BW` implemented. SAP provides this ABAP OO interface `IF_EMMA_KPI_BW` with the methods `WRITE_MSG_TO_BW_DELTA` and `LAST_CALL_TO_BW_DELTA`. For more details, see chapter 7.3 *Utilities Industries - Process Statistics*.

After you created your own business process areas, you must enable the related BPEM monitoring (for more details, see chapter 3.6.2 *Activate Business Process Areas for Message Management*).

3.6.2 Define Business Processes for Message Management

In this activity, you enter the installation-specific business process codes for business processes you want to monitor with BPEM. Push the *New Entries* button and enter the name of the business process code if you want to create a new business process code, or double-click on one of the entries to maintain the technical details of a business process code. Due to the evolution of BPEM, the Customizing offers fields for input that are not only relevant for BPEM but also for the previous version of BPEM, the *Classic EMMA* (for more details, see chapter 1.2 *History and Naming Convention of BPEM*). Figure 3-42 shows the maintenance screen to enter the technical details of a business process code.

General Data

In the general data section, you enter the new business process code, a descriptive text, the related business process area and the primary business object you will be processing during the related business process. This information is required regardless of the BPEM version and BPEM transactions you use.

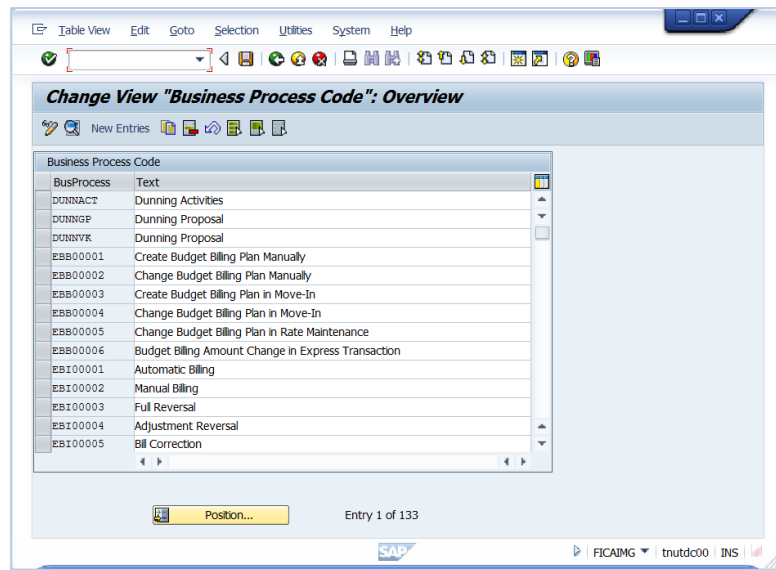


Figure 3-41 Maintenance of Business Process Codes

BP EM (New EMMA)

The *Job Analysis Class* field is the only additional relevant field for the BP EM transactions beside the business process code itself. In most cases, you simply enter `CL_EMMA_JOB_SINGLE` as the relevant ABAP OO class to carry out the BP EM job preparation step. The implemented methods read and analyze the application log containing both the application and BP EM related control messages. The remaining fields can remain empty.



For technical details about the BP EM job analysis class, see chapter 5.2.4 *Technical Implementation Job Preparation New EMMA*

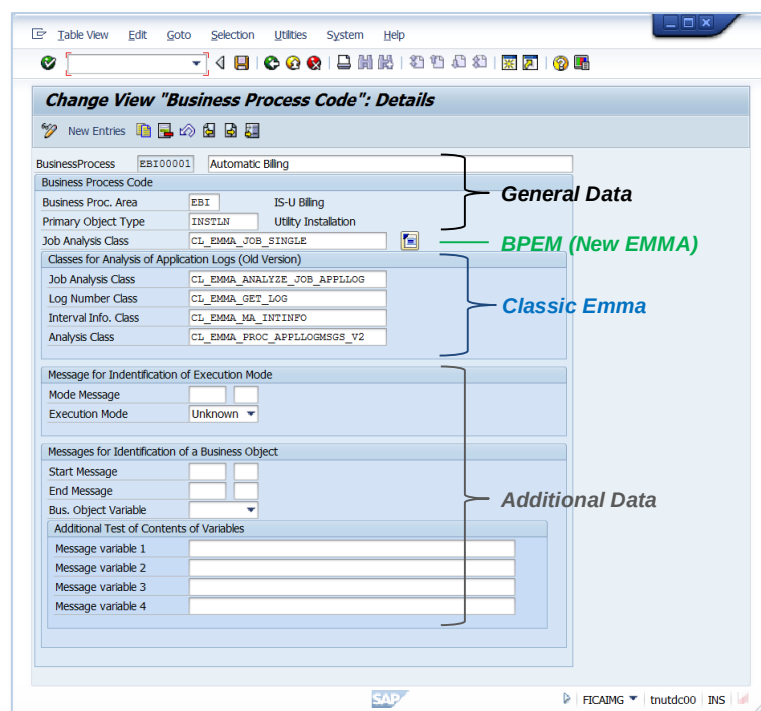


Figure 3-42 Maintenance of Technical Details of a Business Process Code

Classic EMMA

The Customizing is more complex if you still use the transactions of the *Classic EMMA*. This Customizing requires input at least for the fields as summarized in the section *Classes for Analysis of Application Logs (Old Version)*.



SAP highly recommends implementing new applications based on BPEM (New EMMA).

Instead of one single job analysis class as for BPEM, you must enter an own ABAP OO class per internal process step of the BPEM job preparation. Commonly, the following Customizing should be sufficient:

Field	Input
<i>Job Analysis Class</i>	CL_EMMA_ANALYZE_JOB_APPLLOG
<i>Log Number Class</i>	CL_EMMA_GETLOG
<i>Interval Info Class</i>	
<i>Analysis Class</i>	CL_EMMA_PROC_APPLLOGMSGSGS_V2

Figure 3-43 Standard Customizing for applicable for most Application

If you implemented your business process as a mass activity, enter the following data:

Field	Input
<i>Job Analysis Class</i>	CL_EMMA_ANALYZE_JOB_APPLLOG
<i>Log Number Class</i>	CL_EMMA_MALOG
<i>Interval Info Class</i>	CL_EMMA_MA_INTINFO
<i>Analysis Class</i>	CL_EMMA_PROC_APPLLOGMSGSGS_V2

Figure 3-44 Standard Customizing for applicable for most Mass Activities



A business process that is implemented as a mass activity requires additional mandatory Customizing. For more details see 3.6.4 *Define Transactions for Message Management*.

Additional Data

The Customizing in the additional data section is rarely used, very specific and relevant only if you want to use the *Classic EMMA*.

You can define the message, the **Mode Message**, that indicates if a process is executed in either test run mode (no database updates) or update run (database updates occur).

You can define a business process code that does not expect the EMMA 011 and EMMA 012 messages as the **Start Message** and **End Message** (for more information about these messages, see chapter 5.2.1 *Analysis and Grouping of Messages*). Instead, you need to enter the messages that indicate the start and the end of a message group related to a business object. In addition, you need to enter in which message variable, the **Business Object Variable**, the system shall find the key of the business object. For example, the business object variable of the EMMA 011 and EMMA 012 messages is the message variable v2. You find an example of such a Customizing in chapter 6.7 *Evaluation of Job Logs*.

The check for the correct start message can be further enhanced in having the system to additionally check for an expected content in the message variables of the start message above as entered in the **Additional Test of Contents of Variables** fields.

3.6.3 Define Business Processes for Processes from CIC

In this activity, you associate a business process code to a process that is executed with the action box from the Customer Interaction Center (CIC). You can only measure front office processes and methods of the BOR.

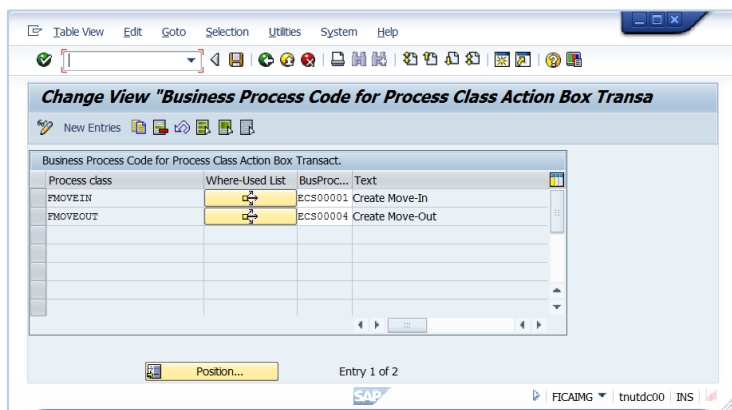


Figure 3-45 Business process code for processes from CIC

To activate BPME for either a specific front office process or a method of the BOR method that is executed via action box, select a process ID from the possible entries. You can specify either the action box configuration or the action box transaction.



When you activate BPME for a front office process, you activate it regardless in which action box configuration it is included.

3.6.4 Define Transactions for Message Management

This Customizing step is only necessary if a single business process is assigned to multiple transactions. However, it is obligatory to register the transaction via this activity if the transaction represents a mass activity.

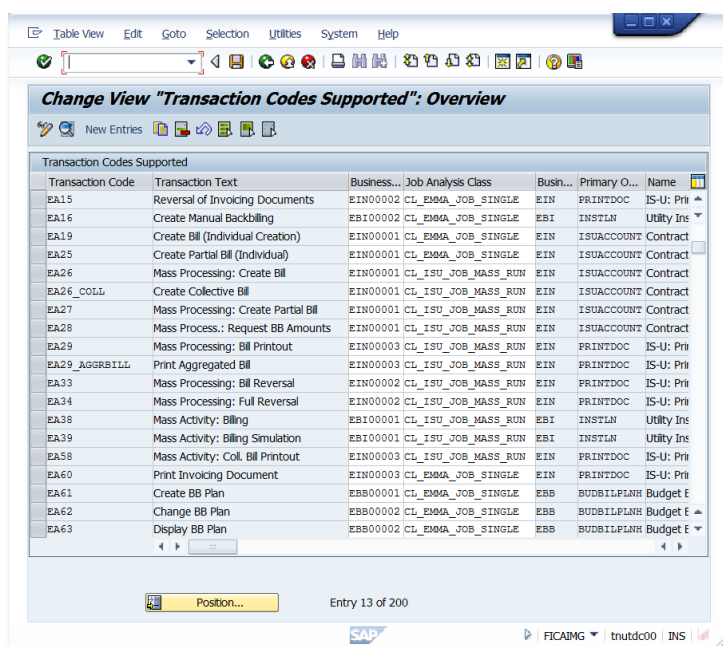


Figure 3-46 Overview BPME enabled Transactions

The required field entries are the same as for the IMG activity 3.6.2 *Define Business Processes for Message Management* except you define them for a specific transaction code.

New Entries: Details of Added Entries

Transaction Code: EA38 Mass Activity: Billing

Transaction Codes Supported

Business Process: EBI00001 Automatic Billing

Business Proc. Area: EBI IS-U Billing

Primary Object Type: INSTLN Utility Installation

Job Analysis Class: CL_ISU_JOB_MASS_RUN

Classes for Analysis of Application Logs (Old Version)

Job Analysis Class	CL_EMOA_ANALYZE_JOB_APPLOG
Log Number Class	CL_EMOA_ISU_MALOG
Interval Info. Class	CL_EMOA_MA_INTINFO
Analysis Class	CL_EMOA_PROC_APPLOGMSGGS_V2

Figure 3-47 Customizing to enable a Transaction for BPEM



The **Business Process Code** field is not a mandatory field. However, if you want the system to automatically generate clarification cases you need to register your transaction **with** an existing business process code. This is because a clarification case category as a template for clarification cases is always bound to a business process code hence without a business process code the system cannot identify eligible clarification case categories and generate a clarification case for the identified business exceptions.

3.6.5 Define Business Objects for Customer-Defined Messages

In this activity, you define the references between the message variables of application messages and corresponding business objects. This definition enables BPEM to recognize the information in a message variable as a business object or information related to a specific table field of the data dictionary (DDIC). BPEM uses this information to offer the agent the option to navigate from a message list to the related business object if the stored reference is a link to a specific business object. For the navigation, the system calls the default method of the related BOR object (commonly, the method to display the business object is the default method). If the stored reference is a link to a DDIC field, the system displays the field value together with some further field attributes.

The BPEM transactions display the  symbol next to a message in the message list indicating that for at least one message variable of this message there is reference available to a business object or a table field. The agent can click on this button to navigate to the related business object. If more than one link is available for a message, the system offers first a list of the available references for the agent to select. In addition, BPEM uses the defined relationships between messages and business objects to populate the clarification case container of a clarification case category.

You must enter a message class, message number, one of the message variable names V1, V2, V3 or V4, and either the object type of a business object or the table and field name of the referenced table field. You must register individual entries for each message variable of the same message containing a reference. Instead of an object type, you may also enter &V1&, &V2&, &V3& or &V4& for the system to dynamically determine the object type in the respective message variable V1, V2, V3 or V4.

SAP delivers the most frequently used messages and business objects. You may want to amend an additional message if this message contains always an identification of the same type of business object or a reference to one specific table field in at least one of its message variables.

Change View "Message Objects": Overview

Message Class	Nu...	VN	Object type	Reference ...	Reference ...	Message Text
AH	385	V3		EVER	VERTRAG	Installation &1: contract &2 must be billed first
AH	420	V1	SCHEMA			Schema &1 contains no schema steps
AH	421	V1	SCHEMA			Sub-transactions in schema &1 step &2 inconsistent
AH	432	V1	BILLINGDOC			Internal: billing document &1 invoicing document &2: i
AH	447	V1	INSTLN			In installation &1, there is a register without a rate type
AH	478	V2	INSTLN			Inconsist. installation structure for device/register &1 fr
AH	519	V1	SCHEMA			Schema &1: no steps in period &2 to &3
AH	609	V2	DEVICE			(Device &1) equip. no. &2 reg. &3 key date &4: no ent
AH	654	V1		EABL	GERNR	No meter reading result exists on &3 for device &1 regi
AH	654	V2		EABL	ZWNUMMER	No meter reading result exists on &3 for device &1 regi
AH	654	V3		EABL	ADAT	No meter reading result exists on &3 for device &1 regi
AH	654	V4	DEVICE			No meter reading result exists on &3 for device &1 regi
AH	691	V1		EABL	GERNR	Device &1 register &2 date &3: (periodic) meter readin
AH	691	V2		EABL	ZWNUMMER	Device &1 register &2 date &3: (periodic) meter readin
AH	691	V3		EABL	ADAT	Device &1 register &2 date &3: (periodic) meter readin
AH	714	V1	DEVICE			Dev. &1 reg. &2 date &3: move-in meter read. result n
AH	719	V1	ISUCONTRACT			Final billing has already been performed for contract &1

Position... Entry 133 of 490

Figure 3-48 Overview registered Messages

SAP delivers two entries for the message with message class AH and message number 385 *Installation &1: contract &2 must be billed first*. In this message the message variable V1 refers to the INSTLN BOR object (*Utility Installation*) and message variable V2 to the ISUCONTRACT BOR object (*Utility Contract*).

3.7 Process Statistics (Utilities Industry)

3.7.1 IMG – Integration with BW (Utilities Industry)

This Customizing is specific to the SAP for Utilities solution. You find the related activities below the SAP for Utilities section of the IMG at *Integration with other mySAP.com Components* → *Data Transfer to the SAP Business Information Warehouse* → *Settings for Application-Specific DataSources (PI)* → *Utilities Industries* → *Process Statistics*.

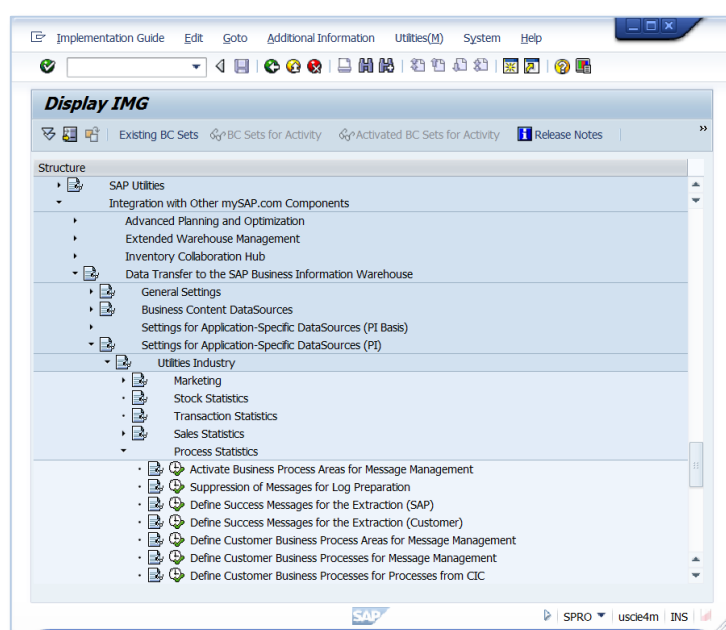


Figure 3-49 Overview SAP for Utilities specific Customizing Activities

The figure above shows all Customizing activities related to BPEM. For more information about the extraction of BPEM related data, see chapter 7 *Business Information Warehouse*.

3.7.2 Activate Business Process Areas for Message Management

In this activity, you maintain the same Customizing as in activity 3.3.2 *Activate Business Process Areas for Message Management*.

3.7.3 Define Suppression of Messages for Log Preparation

In this activity, you maintain the same Customizing as in activity 3.3.4 *Suppression of Messages for Log Preparation*.

3.7.4 Define Success Messages for the Extraction (SAP)

In this activity, SAP defines the success messages, information messages and warning messages the system shall incorporate into the process statistics. Messages that are not included in the success message list are not included in the process statistics.

The system incorporates messages into the process statistics only if they have not led to a BPEM clarification case, and, in addition, the settings in the business process area are such that messages leading to a clarification case are not to be extracted. You can change these settings when activating business process areas for message management. For more details, see chapter 3.3.2 *Activate Business Process Areas for Message Management*.

3.7.5 Define Success Messages for the Extraction (Customer)

In this activity, you define additional success messages, information messages and warning messages the system shall incorporate into the process statistics.

The system incorporates messages into the process statistics only if they have not led to a BPEM clarification case, and, in addition, the settings in the business process area are such that messages leading to a clarification case are not to be extracted. You can change these settings when activating business process areas for message management. For more details, see chapter 3.3.2 *Activate Business Process Areas for Message Management*.

3.7.6 Define Business Process Areas for Message Management

In this activity, you maintain the same Customizing as in activity 3.6.1 *Define Business Process Areas for Message Management*.

3.7.7 Define Business Processes for Message Management

In this activity, you maintain the same Customizing as in activity 3.6.2 *Define Business Processes for Message Management*.

3.7.8 Define Business Processes for Processes from CIC

In this activity, you maintain the same Customizing as in activity 3.6.3 *Define Business Processes for Processes from CIC*.

4 Dialog and Batch Transactions

4.1 Overview

Chapter 2.2 *Overview BPEM Processes and Transactions* already explained the following five main BPEM process steps:

- Step 1: BPEM Job Preparation
- Step 2: Clarification Case Generation
- Step 3: Distribution of Clarification Cases
- Step 4: Resolution and Completion of Clarification Cases
- Transaction EMMACAP Execute Automatic Solution Processes
- Step 5: Transfer of KPIs to Business Information Warehouse

The following transactions are available to perform the process steps. The following chapters describe them in more details except for the process step 5. For more details on step 5, see chapter 7 *Business Information Warehouse*.

Automatically perform process steps 1, 2 and 3:

Transaction FPEMMAMA	<i>BPEM: Mass Activity</i>
Transaction EMMAJP	<i>Business Process Analysis: Process Jobs</i>
Transaction EMMAJL	<i>Business Process Analysis: Job Lists</i>
(Transaction EMMA	<i>Business Process Analysis and Case Generation)</i>

Automatically perform process step 4 (automatic solution processes):

Transaction FPEMMAMA	<i>BPEM: Mass Activity</i>
Transaction EMMAJP	<i>Business Process Analysis: Process Jobs</i>
Transaction EMMACAP	<i>Execute Automatic Solution Processes</i>

Manually perform process steps 3 and 4:

Transaction EMMACL	<i>Clarification Case List</i>
Transaction EMMACLS	<i>Clarification Case List with Shortcut Keys</i>
Transaction EMMAC1	<i>Create Manual Clarification Case</i>
Transaction EMMAC2	<i>Change Clarification Case</i>
Transaction EMMAC3	<i>Display Clarification Case</i>

In addition, SAP delivers a programming interface that is explained in chapter 5.7 *Business Application Programming Interface*. The following chapters describe the transactions in details.

4.2 Job Preparation and Automatic Clarification Case Processing

During the execution of the *BPEM Job Preparation* process step all messages of the application process are read from the application logs or other sources, merged, sorted by their timestamps and eventually saved in BPEM specific database tables for a later fast access. In particular, the Customizing of the 3.3.4 *Suppression of Messages for Log Preparation* activity affects widely this process step since it decides if (error) messages are included or excluded from the further processing.

The subsequent process steps 2 and 3, *Clarification Case Generation*, are the most complex BPEM process steps. Chapter 5.3 provides additional technical information for a better understanding of the various parameters of a clarification case category that affect the generation of clarification cases.

Process step 4, *(Automatic) Resolution of Clarification Cases*, performs the automatic solution processes of clarification cases in status *New* or *In Process*.

4.2.1 Transaction FPEMMAMA - BPEM Mass Activity

This chapter describes the BPEM mass activity, the FPEMMAMA transaction, which is the preferred tool to carry out the BPEM process steps 1 - 3 and, as much automatic solution processes are concerned, also process step 4.

You can execute the BPEM process steps right upon completion of a BPEM relevant business process or a related mass activity in setting up multiple mass activity runs and the business process area or business process code as selection criteria. Alternatively, you start the FPEMMAMA mass activity at the end of the nightly batch chain without entering any selection criteria to capture all processed business process for which you enabled BPEM.

At a time before BPEM or even Classic EMMA, all mass activities called the SLG1 transaction (*Display Application Log*) to display the application logs. In the meantime this was changed and most mass activities use the BPEM framework to display the application logs in a more structured way. If the agent requests to display the application log the system performs automatically the BPEM process step 1 *Prepare Jobs* for the selected mass activity if it was not executed before, for example, as part of the nightly batch chain. The execution of this step may take some time if the application stored a high volume of messages in the application logs or if many intervals have been processed. In the worst case, the process dumps if a timeout occurs when execution it in dialog mode.



SAP recommends including the FPEMMAMA mass activity in the nightly batch chain and processing the mass activity at least in operation mode *Prepare Jobs*. This will significantly reduce the response time for the agents to display the application logs of all mass activities.

The FPEMMAMA mass activity creates own BPEM application logs using the following business process codes:

EMMAJOB0	Operating mode <i>Prepare Jobs</i>
EMMAJOB1	Operating Mode <i>Create Clarification Cases</i>
EMMAJOB2	Operating Mode <i>Execute Solution Processes</i>



You can analyze the application logs of the FPEMMAMA mass activity and create clarification cases as if the FPEMMAMA transaction were just any other mass activity.

On the upper part of the screen of the FPEMMAMA transaction, the standard fields for a mass activity are available, such as the identification of the mass activity and runtime information. The following chapters describe the screens specific to the FPEMMAMA transaction.

4.2.1.1 Screen General Selection

On the *General Selection* screen, you enter the operation mode of the FPEMMAMA transaction and the criteria to select the BPEM jobs to be processed.

The red box in the figure below shows the three possible operating modes (mass activity types) of the FPEMMAMA transaction:

Prepare Jobs	Carry out processing step 1
Create Clarification Cases	Carry out processing step 2 and 3
Execute Solution Processes	Carry out processing step 4



The processing mode *Execute Solution Process* is only relevant if there are automatic solution processes ready to be executed in one of the available clarification case categories (for more details about automatic solution processes, see chapter 3.4.3.5 *Screen Processes*).

Figure 4-1 Main Screen of Transaction `FPEMMAMA`

You do not have to enter any selection criteria for BPEM jobs because the `FPEMMAMA` transaction selects only the BPEM jobs that are relevant for the selected operating mode. You enter selection criteria only if you want to execute the BPEM process steps for selected BPEM jobs only.



The `FPEMMAMA` transaction does not offer an operating mode to reverse a processing step. If required, you must use the `EMMAJL` or `EMMAJP` transaction for a reversal.



During the execution of automatic solution processes of clarification cases, the system automatically changes the status of the selected clarification cases from *New* to *In Process*.

4.2.1.2 Screen *Technical Settings*

On the *Technical Settings* screen, you detail how the `FPEMMAMA` transaction is to be executed.

Parallel Processing Object

The system enters the `EMMA_JOB` parallelization object automatically since it is the only parallelization object used by the `FPEMMAMA` mass activity. The intervals of the mass activity to be processed are determined dynamically at the start of the mass activity (for more details about the interval determination, see chapter 5.6.2 *Interval Determination*)

Load Balancing

You define how many concurrent batch jobs are to be scheduled to process the job intervals and leave it to the system on which application server the batch job should perform (automatic load balancing). Alternatively, you specify how many batch jobs the system shall schedule per application server. Contact your SAP system administrator to discuss the best option for your implementation.

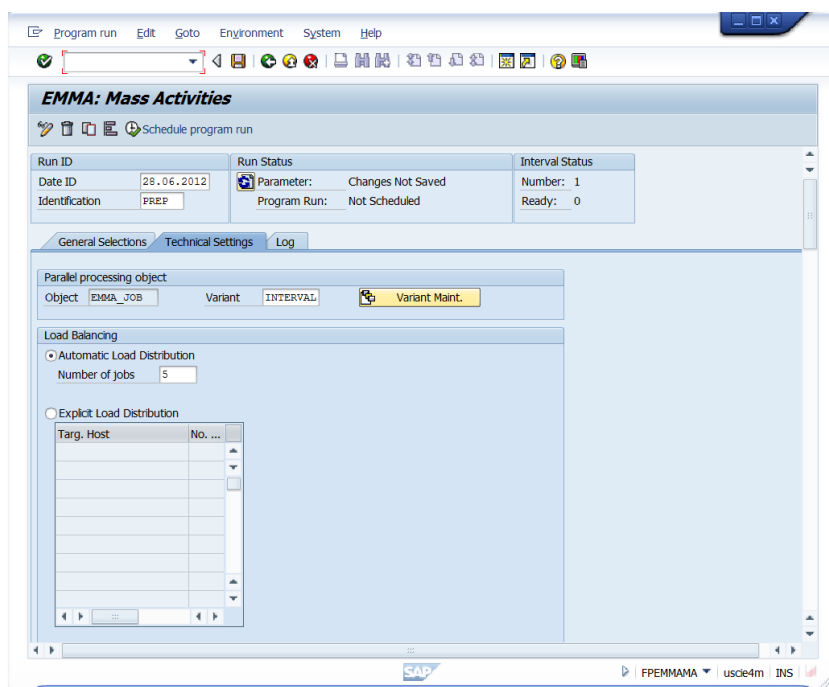


Figure 4-2 Technical Settings Screen of Transaction FPEMMAMA

4.2.1.3 Screen Log

On the *Log* screen, you define the application log parameters.

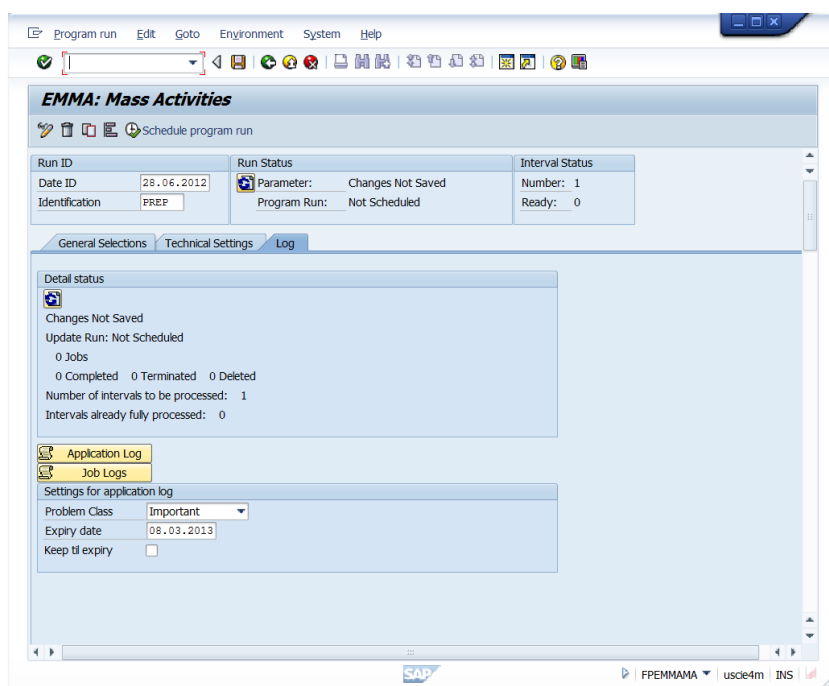
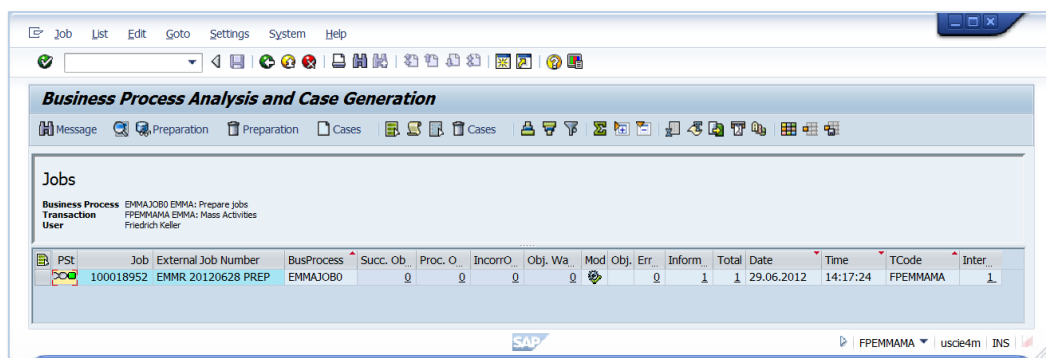


Figure 4-3 Log Screen of FPEMMAMA Transaction

You select the severity level, the *Problem Class*, of messages to be stored in the application log and the expiration date of the application log to indicate when it can be archived.

Upon completion of the mass activity, you can display a consolidated summary of the messages of the selected BPEM process in pushing the *Application Log* button. The overview is the same as the EMMAJL transaction display and is explained in chapter 4.2.4 *Transaction EMMAJL – Display Jobs*.



The screenshot shows the SAP Business Process Analysis and Case Generation interface. The top menu bar includes Job, List, Edit, Goto, Settings, System, and Help. Below the menu is a toolbar with various icons. The main area is titled 'Business Process Analysis and Case Generation' and contains a 'Jobs' section. A table displays job details for a specific transaction.

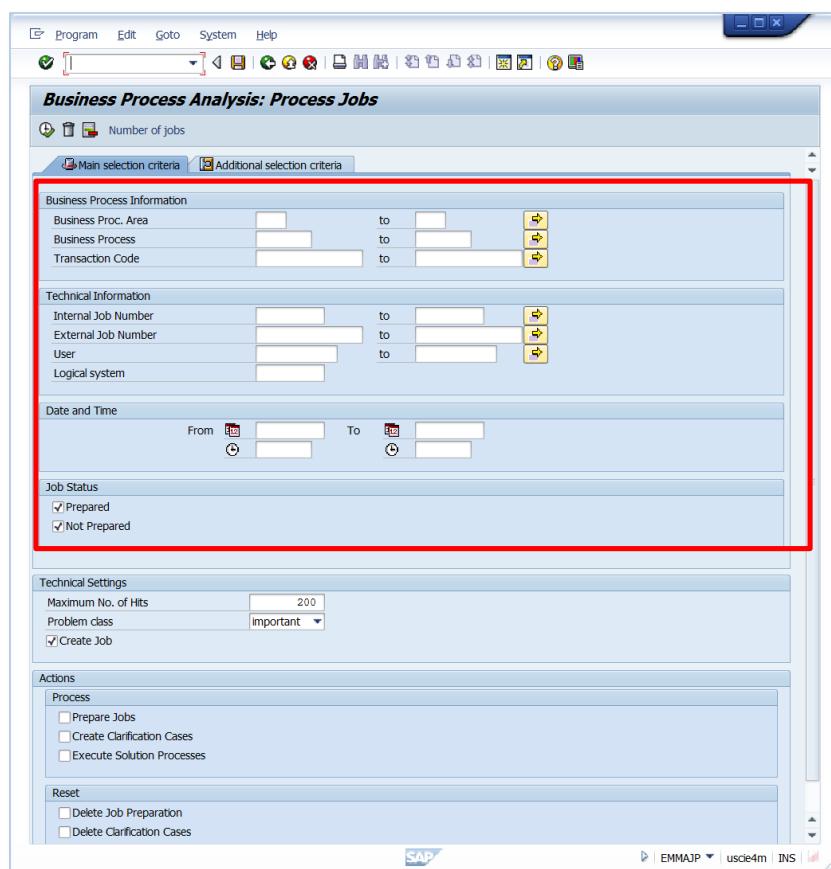
PSt	Job	External Job Number	BusProcess	Succ. Ob	Proc. O	Incorro	Obj. Wa	Mod	Obj. Err	Inform	Total	Date	Time	TCode	Inter
1	100018952	EMMR 20120628 PREP	EMMAJOB0	0	0	0	0	0	0	0	1	29.06.2012	14:17:24	FPEMMAMA	1

Figure 4-4 BP EM Display of Application Messages

4.2.2 Transaction EMMAJP - Process Jobs

The EMMAJP transaction carries out, as the FPEMMAMA transaction, the process steps 1 - 3 *Prepare Jobs* and *Create Clarification Cases* and, as much automatic solution steps are concerned, also process step 4 *Execute automatic solution processes*. In addition to the available functions of the FPEMMAMA transaction, the EMMAJP transaction offers functions to reverse the same process steps 1 – 3.

You use the EMMAJP transaction if you want to process a rather small volume of BP EM jobs since the EMMAJP transaction carries out the selected functions in one single job either online or as background job hence without parallelization.



The screenshot shows the SAP Business Process Analysis: Process Jobs selection screen. The interface includes a menu bar (Program, Edit, Goto, System, Help) and a toolbar. The main area is titled 'Business Process Analysis: Process Jobs' and contains several sections for selecting criteria and actions.

Main selection criteria

- Business Process Information:
 - Business Proc. Area: [] to []
 - Business Process: [] to []
 - Transaction Code: [] to []
- Technical Information:
 - Internal Job Number: [] to []
 - External Job Number: [] to []
 - User: [] to []
 - Logical system: []
- Date and Time:
 - From: [] To: []
 - [] []
- Job Status:
 - ☒ Prepared
 - ☒ Not Prepared
- Technical Settings:
 - Maximum No. of Hits: 200
 - Problem class: Important
 - ☒ Create Job
- Actions:
 - Process:
 - ☐ Prepare Jobs
 - ☐ Create Clarification Cases
 - ☐ Execute Solution Processes
 - Reset:
 - ☐ Delete Job Preparation
 - ☐ Delete Clarification Cases

Figure 4-5 Selection Screen of the EMMAJP Transaction

The `EMMAJP` transaction offers on the *Main Selection Criteria* screen the same criteria as the `FPEMMAMA` transaction. However, the `EMMAJP` transaction offers additional technical selection criteria on the *Additional Selection Criteria* screen.

The screenshot shows the 'Additional selection criteria' tab of the `EMMAJP` transaction. It contains four sections with selection criteria:

- Interval Counter:** Interval Number (to) to (to) [to button]
- Business Object Counter:**
 - Incorrect Objects (to) to (to) [to button]
 - Objects w/ Warnings (to) to (to) [to button]
 - Successful Objects (to) to (to) [to button]
 - Total No. of Objects (to) to (to) [to button]
- Message Counter:**
 - Error Messages (to) to (to) [to button]
 - Warning Messages (to) to (to) [to button]
 - Information Messages (to) to (to) [to button]
 - Total Number of Messages (to) to (to) [to button]
- Clarification Case Counter:** No. Clarif. Cases (to) to (to) [to button]

Figure 4-6 Additional Selection Screen of the `EMMAJP` Transaction

You can select all actions *Prepare Jobs*, *Create Clarification Cases*, *Execute Solution Process*, *Delete Job Preparation* and *Delete Clarification Cases* at the same time. The `EMMAJP` transaction always carries out the reversal function first (obviously only if there is anything to be reversed) before carrying out the process steps 1 – 4.

You choose the action to reverse a process step if you want to carry out the same process step anew. This option could be helpful if the selected BPBM jobs are in a different state but you do not want to enter a very complex selection criteria if, for example, you already carried out the job preparation step for some of the of BPBM jobs. A reversal could also be reasonable after you changed the Customizing and you want to test the new Customizing.

 A reversal of clarification cases is only possible if all clarification cases related to a BPBM jobs are still in status *New*. This means that no actions may have performed yet on the clarification cases after their creation. A reversal of the job preparation is only possible if no clarification cases are allocated to the BPBM job whose job preparation step you want to reverse.

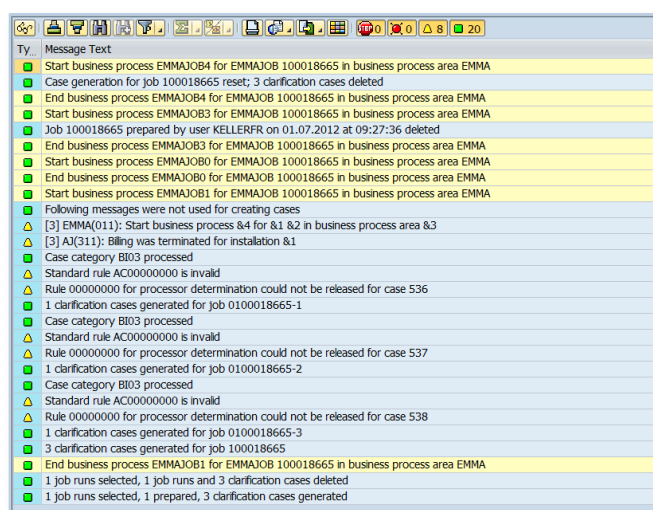


Figure 4-7 Example of an Application Log of the `EMMAJP` transaction

 During the execution of automatic solution processes of clarification cases the system changes automatically changes the status of the related clarification cases from *New* to *In Process*.

You can restrict the number of BPEM jobs to be processed in entering a limit in the *Maximum No. of Hits* field in the *Technical Settings* frame of the selection screen. In addition, you can limit the number of messages to be displayed in choosing an appropriate severity level of messages, the *Problem Class*, during the execution of the transaction. The figure above shows an exemplary application log of the `EMMAJJP` transaction.

Upon completion of the selected actions, the collected messages are either displayed on the screen as a temporary application log (see figure above) or stored in the job log if you executed the transaction in background as a batch job.

If you mark the **Create Job** field on the selection screen a BPEM job is created with own BPEM application logs with the following business process codes:

EMMAJOB0	Operating mode <i>Prepare Jobs</i>
EMMAJOB1	Operating Mode <i>Create Clarification Cases</i>
EMMAJOB2	Operating Mode <i>Execute Solution Processes</i>
EMMAJOB3	Operating Mode <i>Delete Job Preparation</i>
EMMAJOB4	Operating Mode <i>Delete Clarification Cases</i>

With the `EMMAJL` transaction, you can review the collected process messages anytime later.

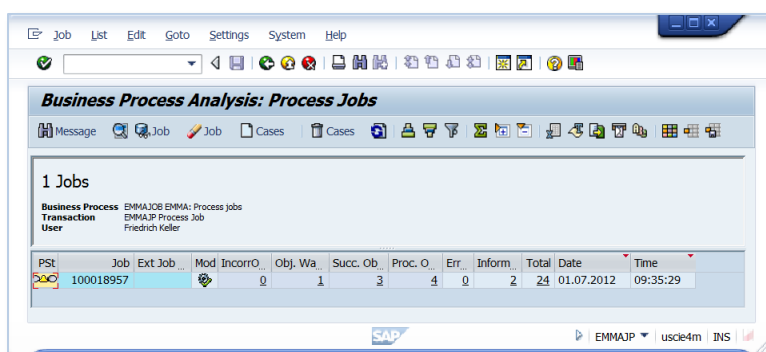


Figure 4-8 BPEM Display of Application Messages after Execution of `EMMAJJP` Transaction

4.2.3 Transaction `EMMACAP` – Execute Automatic Processes

The `EMMACAP` transaction carries out, as the `FPEMMAMA` transaction, the process step 4 *Execute automatic solution processes*. However, the `EMMACAP` transaction is meant to process the automatic solution processes for specifically selected clarification cases in a single process hence non-parallelized. The following paragraphs describe the transaction in more details.

The selection screen of the `EMMACAP` transaction is the same as of the `EMMACL` transaction except that there are two further checkboxes *Manual Case Selection* and *Execute Solution Process again*. The option *Execute Solution Process again* became obsolete with ERP 6.0 EhP7 due to additional options to configure a clarification case category that was implemented to overcome the restriction that the execution of an automatic solution process in background was permitted only once.

You can execute the `EMMACAP` transaction in either dialog or background mode. If you execute the `EMMACAP` transaction in background mode, the automatic solution processes for all selected clarification cases are processed.

Figure 4-9 Selection Screen of the EMMACAP Transaction

If you mark the *Manual Case Selection* checkbox and execute the EMMACAP transaction in dialog mode, the system displays a list showing the selected clarification cases for which automatic solution processes exist that are ready to be executed. You can display a clarification case in marking an individual row and pushing the button, or by clicking on the clarification case number.

Overdue	Case	Clarification Case Text	Status	Job	Interval	Logical system	BusProcess	Case Type	Cat.	Case Prio.	ObjDescrip
	1984	No Operand values were found for Operand EFACOR_2	New	100022584	2	E4MCLNT327	EBI00001	BILL	B103	High	Utility Installa
	1986	No Operand values were found for Operand EFACOR_2	New	100022584	4	E4MCLNT327	EBI00001	BILL	B103	High	Utility Installa
	1985	Billing Block Reason 04 for Contract 4235	New	100022584	3	E4MCLNT327	EBI00001	BILL	B105	Average	Utility Installa

Figure 4-10 Selection Screen of the EMMACAP Transaction

You can now select all or individual clarification cases for an execution of their automatic solution processes. Subsequently, push the *Run Processes* button to execute the automatic solution processes.

Upon completion, the system displays an application log with a list of all executed automatic solution processes.



During the execution of automatic solution processes of clarification cases the EMMACAP transaction automatically changes the status of the related clarification cases from *New* to *In Process*.

Date/Time/User	Nu...	External ID	Object text	Sub-object text	Transac...	Program	Mode	Log number
16.07.2015 17:48:02 KELLERFR	5	0100022584-0000001984	Extended Mess... Case Determination		EMMACAP	REMA...	Dialog pro...	000000000000000171746
16.07.2015 17:48:03 KELLERFR	7	0100022584-0000001985	Extended Mess... Case Determination		EMMACAP	REMA...	Dialog pro...	000000000000000171747
16.07.2015 17:48:03 KELLERFR	5	0100022584-0000001986	Extended Mess... Case Determination		EMMACAP	REMA...	Dialog pro...	000000000000000171748

Type	BusProcess	Obj. Type	Object Key	Cat.	Case	Message Text	LTxt	Det.
INSTLN	4325	B103	1984		16.07.2015 17:40:11 user KELLERFR,Friedrich Keller, PS IMS Services AG			
INSTLN	4325	B103	1984		Process 'Check if case can be closed' automatically executed by user KELLERFR on 16.07.2015 at 17:40:11			
INSTLN	4325	B103	1984		Clarification case 1984 has been returned to the worklist			
INSTLN	4326	B105	1985		16.07.2015 17:40:12 user KELLERFR,Friedrich Keller, PS IMS Services AG			
INSTLN	4326	B105	1985		Process 'Check if case can be closed' automatically executed by user KELLERFR on 16.07.2015 at 17:40:12			
INSTLN	4326	B105	1985		Process 'Check if contract is still blocked' automatically executed by user KELLERFR on 16.07.2015 at 17:40:12			
INSTLN	4326	B105	1985		Clarification case 1985 has been returned to the worklist			
INSTLN	4327	B103	1986		16.07.2015 17:40:21 user KELLERFR,Friedrich Keller, PS IMS Services AG			
INSTLN	4327	B103	1986		Process 'Check if case can be closed' automatically executed by user KELLERFR on 16.07.2015 at 17:40:21			

Figure 4-11 Selection Screen of the EMMACAP Transaction

4.2.4 Transaction EMMAJL – Display Jobs

You use the EMMAJL transaction mainly to review the results of the BP EM process steps *Job Preparation* and *Clarification Case Creation*. However, there are the same functions available to carry out the same BP EM process steps as the EMMAJP transaction but you can initiate these BP EM process steps only for a single BP EM job instead for a selection of BP EM jobs as the EMMAJP transaction.

Business Process Analysis: Job List

Number of jobs

Main selection criteria Additional selection criteria

Business Process Information

Business Proc. Area to

Business Process to

Transaction Code to

Technical Information

Internal Job Number to

External Job Number to

User to

Logical system

Date and Time

From To

Job Status

☒ Prepared

☒ Not Prepared

Technical Settings

Maximum No. of Hits 200

Problem class important

Figure 4-12 Selection Screen of the EMMAJL Transaction

4.2.4.1 Selection Screen

The selection screen of the EMMAJL transaction is the same as of the EMMAJP transaction except that the *Action* checkboxes are missing to request the execution of the BP EM process steps such as *Job Preparation* and *Clarification Case Creation*.

4.2.4.2 BPEM Job List

The following figure shows a sample list of selected BPEM jobs the system created during the execution of the EA38 *Billing* mass activity. From this initial screen, you may drill down to explore the details of each job.

PST	Job	External Job Number	Inter.	Incom.O.	Succ. Ob.	Proc. O.	Err.	Inform.	Total	Date	Time	Sta	Stat	Cases
●●●	100018958	BILL 04.07.2012 FK0001 Completed	2	2	0	2	0	33	42	04.07.2012	16:59:48	●	●	2
●●●	100018907	BILL 12.05.2012 FK0028 Completed	0	0	0	0	0	0	0	08.06.2012	11:06:45	●	●	0
●●●	100018665	BILL 20.04.2012 FK0008 Completed	2	2	0	2	0	30	39	20.04.2012	17:41:27	●	●	0
●●●	100018633	BILL 02.03.2012 FK0005 Completed	2	2	0	2	0	30	39	05.04.2012	11:15:08	●	●	2

Figure 4-13 Overview selected BPEM Jobs

The screen may vary if the resulting list contains the same value in either of the first two columns *BusProcess* (business process code) and *Transaction Code*. Instead of additional columns, the system displays identical values in the header instead (as in our example the business process code EBI00001 and the EA38 transaction code). You can select and arrange the columns of the list according to your business requirements (for more details see chapter 2.1.17 ABAP List Viewer ALV).

Column Description

Column Name	Description	Hotspot
General BPEM Job Information		
Business Process	Business process code of the executed business process	✗
Transaction Code	Transaction of the executed process	✗
Preparation Status PST	Overall result (BPEM job must be in status <i>Prepared</i>): ●●● at least one business object failed with errors ●●● at least one business object shows warnings ●●● all business processes with success messages ●●● no business objects found If there are both error and warning messages for a business object, the system categorizes the object as if containing errors and displays it in the list of objects with errors. Similarly, it classifies a business object for which there are warning, information, and success messages as an object with warnings.	✗
Internal Job Number	Unique identification of a BPEM job	✗
External Job Number	External identification of the BPEM job as defined by the executed transaction.	✓
Execution Mode	Execution mode of the business process: ● Update run ● at least one business object shows warnings - no symbol all business processes with success messages	✗
Interval Number	Number of processed intervals of a mass activity	✓

Business Object Statistics		
Incorrect Objects	Number of failed business objects indicated by error or warning messages	✓
Successful Objects	Number of business objects with success messages	✓
Processed Objects	Total number of processed business objects	✓
Message Statistics		
Error Messages	Total number of error messages without an allocation to a particular processed business object (hence without enclosing EMMA011 and EMMA 012 messages)	✓
Information Messages	Total number of success and information messages without an allocation to a particular processed business object (hence without enclosing EMMA011 and EMMA 012 messages)	✓
Total No. of Messages	Total number of messages excluding suppressed messages (see chapter 3.3.4 <i>Suppression of Messages for Log Preparation</i>) and messages without useful information (see chapter 5.2.1 <i>Analysis and Grouping of Messages</i>)	✓
Administration and Status Information		
Date/Time	User ID who carried out the process	✗
Job Status	Current BPEM job preparation status:  Job completely prepared  Job still active  Job terminated no symbol Job preparation not carried out yet	✓
Generation Case Status	Current clarification case creation status  Clarification case creation completed  Clarification case creation reversed  Clarification case creation partially completed (carried out only for individual intervals) no symbol Clarification case creation not carried out yet	✗
Cases	Total number of created clarification cases	✓

Figure 4-14 Description of the Columns of the BPEM Job Overview Screen

Available Functions

Select an entry of the BPEM job list and choose one of the following functions of the application toolbar.

Function	Use
 Message <i>Find Messages</i>	Enables you to search for messages using the following criteria: Message Search for particular messages in specifying message class and message number Business Object Search for message groups allocated to particular business objects in specifying a primary object type and object key The system displays a list of messages that match the entered criteria.
 Details / Show Details	Displays the details of the selected BPEM job
 Job / Prepare Jobs	Carry out the process step <i>BPEM Job Preparation</i>

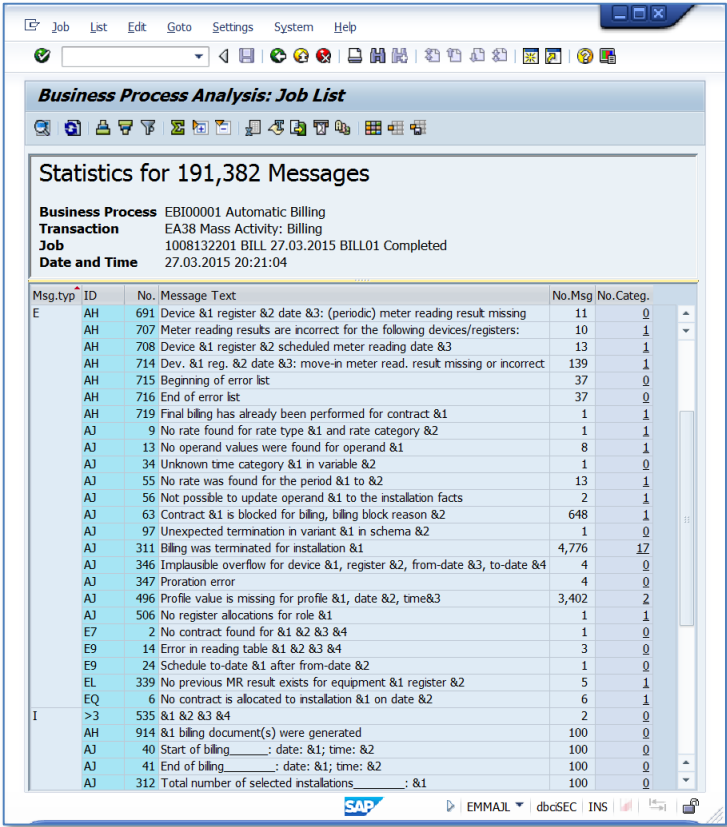
 Job Delete Job Preparation	Reverse the process step <i>BPEM Job Preparation</i> and delete the previous results. You can reverse a BPEM job preparation only if no clarification cases are attached to this BPEM job.
 Cases / Create Cases	Carry out the process step <i>Clarification Case Creation</i>
 Cases / Delete Cases	Reverse the process step <i>Clarification Case Creation</i> in deleting the created clarification cases. You can only delete the clarification cases if all clarification cases of the selected BPEM job have the status <i>New</i> or <i>Reversed</i> .
Job → Message Use	Create a statistic of all messages of a selected BPEM job or BPEM job interval:  The last two columns <i>No. of Cumulated Messages</i> and <i>No. of Clarification Case Categories</i> display per message the occurrence of each message and as well as the number of clarification case categories where you defined this message as a triggering message. The <i>Message Use</i> function is available on interval level, too.
Job → Execute Solution Processes	Execute automatic solution processes

Figure 4-15 Functions available on the BPEM Job Overview Screen

In the description of the columns above, some columns are marked with the  sign to indicate available hotspots. If you click on the information in this column, the system takes you the next level of detail of the selected information.



You can enhance the toolbar of the BPEM job list by own buttons. For more details see chapter 5.8.1.2 *BAdI Definition BADI_EMMA_JOB_WORK_LIST*.

4.2.4.3 Detail Information on Level *Business Object*

If you click on the displayed number of *Incorrect Objects*, *Successful Objects* or *Processed Objects*, the system shows a list of these business objects completed by the messages attached to them:

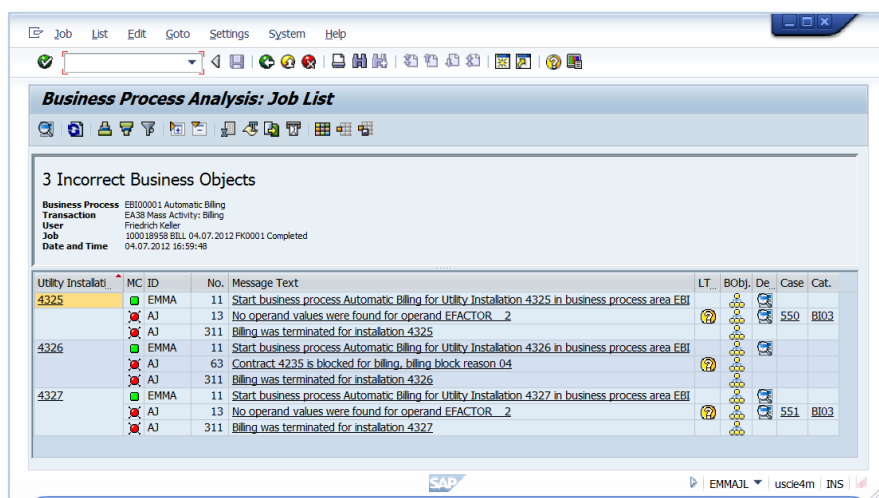


Figure 4-16 List of Business Objects with Error and Warning Messages

The displayed symbols have the following meaning:

Symbol	Meaning
	Error message
	Warning message
	Success message
	Long Text of the message available
	Business Objects available
	Additional message parameters available

Figure 4-17 Overview of used Symbols

The message list does not include the EMMA 012 control messages since they are automatically removed from the list during the previous BPBM job preparation step.

If there is already a clarification case for the displayed business exception, the system adds two additional columns to display the identification of the created clarification cases (Case column) and the related clarification case category (Cat. column). You can navigate to the clarification case in clicking on the clarification case number. To navigate to the displayed clarification case category click on the name of the clarification case category.

If more than one clarification case was created for a business object, only one of the created clarification case numbers is displayed.

You can create a new clarification case category at any time by double-clicking on either the message class or message number regardless if a clarification case has already been created based on the marked message. The system takes you then to the maintenance screen of the clarification case category to complete the template based on the data of the selected message.

4.2.4.4 Detail Information on Interval Level

A mass activity carries out a business process while processing intervals based on a parallelization object (for more information on mass activities, see chapter 2.1.13 *Mass Activity*). When you click on the number in the *Interval* column, the system displays the details on interval level. The figure below shows an example of the detailed information on interval level.

PSt	Interval	Lower limit	UpperLimit	External Interval Number	Proc. O.	IncorrO.	Obj. Wa.	Succ. Ob.	Err.	Inform.	Total	Start Time	End Time	Cases	Stat
1				32767- Opening processing of total run	0	0	0	0	0	2	2	00:00:00	00:00:00	0	
2				32766- Opening processing of job 001	0	0	0	0	0	0	0	00:00:00	00:00:00	0	
3				32765- Opening processing of job 002	0	0	0	0	0	0	0	00:00:00	00:00:00	0	
4				32764- Opening processing of job 003	0	0	0	0	0	0	0	00:00:00	00:00:00	0	
5				32763- Opening processing of job 004	0	0	0	0	0	0	0	00:00:00	00:00:00	0	
6				32762- Opening processing of job 005	0	0	0	0	0	0	0	00:00:00	00:00:00	0	
7	000000101	0000001187	1		0	0	0	0	0	Z	Z	19:20:49	19:20:49	0	
8	0000001188	0000002114	2		0	0	0	0	0	Z	Z	19:20:49	19:20:49	0	
9	0000002115	0000003024	3		0	0	0	0	0	Z	Z	19:20:49	19:20:49	0	
10	0000003025	0000004005	4		0	0	0	0	0	Z	Z	19:20:49	19:20:49	0	
11	0000004006	ZZZZZZZZZZ	5		4	0	0	4	0	Z	23	19:20:49	19:20:50	0	
12				32762 Closing processing of job 005	0	0	0	0	0	0	0	00:00:00	00:00:00	0	
13				32763 Closing processing of job 004	0	0	0	0	0	0	0	00:00:00	00:00:00	0	
14				32764 Closing processing of job 003	0	0	0	0	0	0	0	00:00:00	00:00:00	0	
15				32765 Closing processing of job 002	0	0	0	0	0	0	0	00:00:00	00:00:00	0	
16				32766 Closing processing of job 001	0	0	0	0	0	0	0	00:00:00	00:00:00	0	
17				32767 Closing processing of total run	0	0	0	0	0	0	0	00:00:00	00:00:00	0	

Figure 4-18 Overview Intervals of a Mass Activity

The displayed symbols and available functions are the same as explained in the previous chapter.

BP EM displays not only the messages per interval but additionally - if available - pseudo intervals for the messages the application and its jobs stored ...

- before processing any interval (*Opening processing of total run*)
- after all intervals have been processed (*Closing processing of total run*)
- before a job starts the processing of its first interval (*Opening processing of job xxx*)
- after a job finished the processing of its last interval (*Closing processing of job xxx*)

The respective application logs do not contain any business object related information but application specific information concerning the overall progress of the mass activity.

4.2.4.5 Detail Information Total Number of Messages

When you click on the number in the *Total Number of Messages* column, the system displays all messages as stored in the application of the application including the messages that are not attached to a particular business object.

The message list contains all messages and further information, such as if a clarification case was created. The figure below shows an example of such a full message list.

Business Process Analysis: Job List

All 42 Messages

Business Process: EBI00001 Automatic Billing
 Transaction: EA38 Mass Activity: Billing
 User: Friedrich Keller
 Job: 100018958 BILL 04.07.2012 FK0001 Completed
 Date and Time: 04.07.2012 16:59:48

Utility Instalati...	MC ID	No.	Message Text	LT	Bobj.	De	Case	Cat.
	AJ	40	Start of billing ; date: 04.07.2012; time: 16:59:50					
	EF_BI_KPI	2	Required KPI/ EMMA message					
	AJ	312	Total number of selected installations ; 0000000001					
	AJ	313	Total number of processed installations ; 0000000001					
	AJ	317	Number of installations billed successfully ; 0000000000					
	AH	914	0 billing document(s) were generated					
	AJ	314	Number of incorrect installations ; 0000000001					
	AJ	318	Number of incorrect installations per division: 0000000001, 01 division					
	AJ	333	First installation processed: 4325					
	AJ	316	Installation processed last ; 4325					
	AJ	41	End of billing ; date: 04.07.2012; time: 16:59:59					
4325	EMMA	11	Start business process Automatic Billing for Utility Installation 4325 in business process area EBI					
	AJ	13	No operand values were found for operand EFACTOR_ 2				550	BI03
	AJ	311	Billing was terminated for installation 4325					
	AJ	40	Start of billing ; date: 04.07.2012; time: 17:00:00					
	EF_BI_KPI	2	Required KPI/ EMMA message					
	AJ	312	Total number of selected installations ; 0000000001					
	AJ	313	Total number of processed installations ; 0000000001					
	AJ	317	Number of installations billed successfu ; 0000000000					

Figure 4-19 Overview of All Messages of a Mass Activity

4.2.5 Transaction EMMA – Classic EMMA

As previously explained, SAP replaced the EMMA transaction by the EMMAJP transaction (for more details, see chapter 4.2.2) and EMMAJL transaction (for more details, see chapter 4.2.4). The EMMA transaction carries out the BP EM process steps *Job Preparation* and *Clarification Case Creation* for all business process codes with a completed Customizing for Classic EMMA (for more details, see chapter 3.6.2 *Define Business Processes for Message Management*).

Business Process Analysis and Case Generation

Business Process Information

Business Proc. Area: [] to []

Business Process: [] to []

Technical Information

Internal Job Number: [] to []

Transaction Code: [] to []

External Job Number: [] to []

Date: [] to []

Time: 00:00:00 to 00:00:00

User: [] to []

Only for Case Generation

Interval Number: [] to []

Prepare

☐ Prepare Jobs

☐ Create Clarification Cases

Delete

☐ Delete Job Preparation

☐ Delete Clarification Cases

Job Types to Be Processed

☒ Prepared Jobs

☐ Jobs Not Prepared Yet

Figure 4-20 Selection Screen of the EMMA Transaction

You use the `EMMA` transaction to process a rather small volume of data since the `EMMA` transaction carries out the selected functions single threaded either online or as a background job. SAP recommends using the `FPEMMA` transaction to process larger volumes of BPEM jobs and clarification cases. Figure 4-20 shows the selection screen of the `EMMA` transaction. Be aware that you have to select the correct actions that match with the selected job types (*Prepared Jobs*, *Jobs not prepared yet*).

If you run the `EMMA` transaction in background, it creates an own BPEM job with business process codes `EMMAJOB Prepare Jobs`.



Do not use the `EMMA` transaction to work on BPEM jobs for which the job preparation was already performed by the *New EMMA* transactions because the `EMMA` transaction cannot access the message data storage used by the *New EMMA* transactions. However, the *New EMMA* transactions can deal with the message data storage as created and used by the `EMMA` transaction.

4.3 Working with Clarification Cases

The core of the BPEM application are the clarification cases and their processing. A clarification case represents an individual business exception, options to analyze the business exception and possible solution processes to support its resolution. In addition, there is a status allocated to the clarification case representing the progress in resolving the related business exception from the clarification case creation until the clarification case is closed.

The following chapters describe the usage of personalized clarification case lists as an inbox for clarification cases. The subsequent chapter describes how to work with clarification cases. Eventually, the manual creation of clarification cases is described.

4.3.1 Transactions EMMACL/EMMACLS – Clarification Case List

4.3.1.1 Description relevant for both EMMACL and EMMACLS transaction

Commonly, not the agents but the system automatically creates the majority of clarification cases mainly as a follow-up activity of mass activities as part of the nightly batch. This means that the day after a high volume of new clarification cases is waiting for a resolution. In order to get a high volume of clarification cases managed, they need to be prioritized by various criteria such as creation and due date and the assigned importance hence priority for its resolution. The `EMMACL` transaction allows the selection of clarification cases with a broad variety of options to select clarification cases and return them sorted in accordance to the priority they need to be resolved. Figure 4-21 shows the selection screen of the `EMMACL` transaction.

Due to the possible high volume of existing clarification cases, you may want to restrict the number of entries in the clarification case list by specifying an appropriate number of hits. However, the options *Oldest Cases First*, *Newest Cases First* and *Next Due Cases First* ensure that always the top list of clarification cases are displayed that match with the selection criteria regardless of the entered maximum number of hits.

In previous releases, the system did not offer these sort options by default on the selection screen due to compatibility reasons. However, if you want to define a sort order, define the `EMMA_CWLS_EXTSEL SPA/GPA` parameter ID in your user profile with the parameter `X`. For more information, see SAP Note [1618687](#).

You can control the automatic refresh of displayed clarification case list after upon completion of the maintenance of a clarification case you picked from the list. The system updates the clarification case list is automatically with the data of your last processed clarification case if the number of clarification cases of the list does not exceed the entered value in the field *Automatic Update to Max. Hits*.

The fields of the *Organizational Structure* section allow you to use organizational data of your company to select clarification cases. During the clarification case creation process the system resolves a possibly assigned processor rule (for more details how to assign such a rule, see chapter 3.4.3.1 Screen *Basic Data*). As a result, the system attaches the determined users or organizations to the created clarification case. This does not mean that only related agents may work on that particular clarification case but you can select clarification cases based on this allocation list as allocated to clarification cases. If you enter organizational structure criteria, the system selects only clarification cases where at least one of the previously allocated agents matches with the entered organizational data.

Figure 4-21 Selection Screen of the EMMACL Transaction

You can enhance the selection screen by installation specific fields. For more details, see chapter 3.5.1 *Generate Selection Screen for Clarification List*.

Figure 4-22 shows a clarification case list with some selected clarification cases.

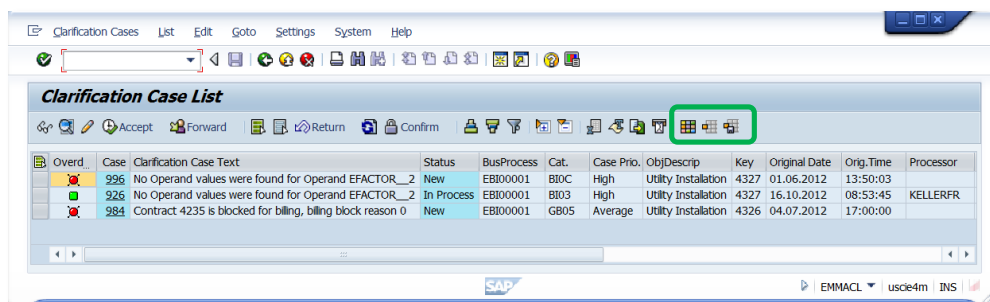


Figure 4-22 Clarification Case List

You can access the maintenance of a clarification case in either double-clicking on any of the fields of the clarification case, marking one line and push either the  or  button to display, or maintain the marked clarification case simply by clicking on the hotspot of the clarification case number. The light on the left hand side indicates if the clarification case is within the defined processing time  or if it is already overdue . See below a description of the further fields.

As for any ALV based list, you can personalize the layout of the list in pushing the  button. On the popup you can add and remove columns to be displayed, change their display order, and change many more settings (for more details, see chapter 2.1.17 *ABAP List Viewer ALV*).

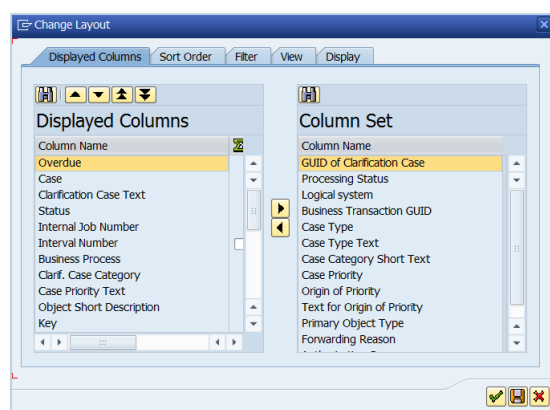


Figure 4-23 Personalization of the Display Variant

You can save the changes as a new display variant and mark it as your default variant after pushing the  button. You can maintain multiple display variants for the clarification case list. Push the  button to toggle between your saved display variants.

The following table describes the columns of a clarification case list in details (fields of the `EMMA_CASE` table). As described above, agents can personalize their lists in rearranging the fields and drop fields from the list if they do not require the information of a particular field for their daily work.

Column Description:

Column Name	Description
Overdue	Resolution status of clarification case regarding processing time  resolution is still within predefined processing time  resolution is overdue
Case	Clarification case number
Clarification Case Text	Descriptive text of clarification case

Status	Processing status of clarification case
Job	Related BPEM job number
Interval Number	Related interval number of a related mass activity
Logical System	Logical system name where the clarification case was created
Business Process	Business process code of the related business process
Case Type	Clarification case type
Clarification Case Cat.	Related clarification case category
Case Priority	Allocated priority of the clarification case
Original Date and Time	Creation date of clarification case
Due date and Time	Expected maximum processing time
Processor	Current processor
Previous Processor	Previous processor
<i>Additional Fields</i>	All fields of the <code>EMMA_CASE</code> table are available including the installation specific fields of the <code>CI_EMMA_CASE</code> include

Figure 4-24 Description of the Columns of the Clarification Case List Overview Screen

Available Functions

Select one or more entries in the clarification case list and choose one of the following functions of the application toolbar. The available buttons in the toolbar depends on the used transaction: the `EMMACL` transaction offers some few buttons less than the toolbar of the `EMMACLS` transaction.

Function	Use
 <i>Display</i>	Calls the clarification case maintenance transaction in <i>Display Mode</i>
 <i>Change</i>	Calls the clarification case maintenance transaction in <i>Change Mode</i> . This function can only be processed if the selected clarification case is in status <i>New</i> or <i>In Process</i> .
 <i>Accept</i>	Allocate you as the current processor to the clarification case. This function can only be processed if the selected clarification case is in status <i>New</i> or <i>In Process</i> .
 <i>Forward</i>	Allocate a processor to the clarification case. Beside the new processor a forwarding reason must be entered (see chapter 3.5.2 <i>Define Reasons for Forwarding Clarification Cases</i> how to define forwarding reasons). The status of the clarification case is changed to <i>In Process</i> . This function can only be processed if the selected clarification case is in status <i>New</i> or <i>In Process</i> .
 <i>Return</i>	Return the clarification case to the clarification case list in removing the current processor. A forwarding reason must be entered (see chapter 3.5.2 <i>Define Reasons for Forwarding Clarification Cases</i> how to define forwarding reasons) This function can only be processed if the selected clarification case is in status <i>New</i> or <i>In Process</i> .
 <i>Complete</i>	Change the status of clarification case to <i>Complete</i> . This function can only be processed if the selected clarification case is in status <i>New</i> or <i>In Process</i> .

 Reverse	Reverse the clarification case. A reversal reason must be entered (see chapter 3.5.3 <i>Define Reasons for Reversing Clarification Cases</i> how to define reversal reasons). This function can only be processed if the selected clarification case is in status <i>New</i> or <i>In Process</i> .
 Reopen	Reopen the clarification case. The status of the clarification case is changed to <i>In Process</i> . This function can only be processed if the selected clarification case is in status <i>Complete</i> or <i>Canceled</i> .
 Confirm	Confirm the clarification case. This function can only be processed if the selected clarification case is in status <i>Complete</i> .
 Refresh	Manual refresh of the clarification case list
 Select All	Select all displayed clarification cases (EMMACL transaction only)
 Deselect All	Deselect all selected clarification cases (EMMACL transaction only)
 Layout	Change the layout of the action box (EMMACLS transaction only)

Figure 4-25 Functions available on the Clarification Case List

The system removes buttons from the toolbar automatically if the agent does not have the authorization for the activity attached to the related button. You may use this feature to personalize the toolbar for agents. For more information about the `B_EMMA_CAS` authorization object, see chapter 3.2.3 *Authorization for Clarification Case Processing*.

 Remove for all agents the authorization for the activity `31 Confirm clarification case` of the `B_EMMA_CAS` authorization object if a confirmation step is not required as a final step to close a clarification case. Due to the lack of authorization, the system removes the *Confirm* button from the toolbar for all agents.

 You want to force an agent to forward a clarification case to another agent instead of simply returning it to the clarification case worklist. The system removes the *Return* button from the toolbar if the agent is not authorized for the activity `87 Return clarification case to worklist` of the `B_EMMA_CAS` authorization object.

 You can enhance the toolbar of the clarification case list by own buttons. For more details see chapter 5.8.2.1 *BAdI Definition BADI_EMMA_CASE_WORK_LIST*.

4.3.1.2 Description relevant only for the EMMACLS transaction

Due to its complexity, only experts should use the `EMMACL` transaction. In contrast, the `EMMACLS` transaction has the advantage of multiple personalized views tailored to meet the requirements of agents in their respective departments. Within the transaction, agents can switch easily between clarification lists by a push of a button in the action box as shown in the figure below where each button represents own specific selection criteria for clarification cases. Chapter 3.5.4 *Maintain Shortcut Keys for Clarification Case Processing* describes the Customizing of these personalized views. In addition, the agents can also personalize the layout of their clarification case lists in pushing the  button (see previous chapter). If needed, the agent can even switch to another defined action box layout to another set of buttons with their attached selection criteria in simply pushing the  button.



While the `EMMACL` transaction allows the processing of multiple clarification cases, the `EMMACLS` transaction supports the processing of only one single clarification case at a time.

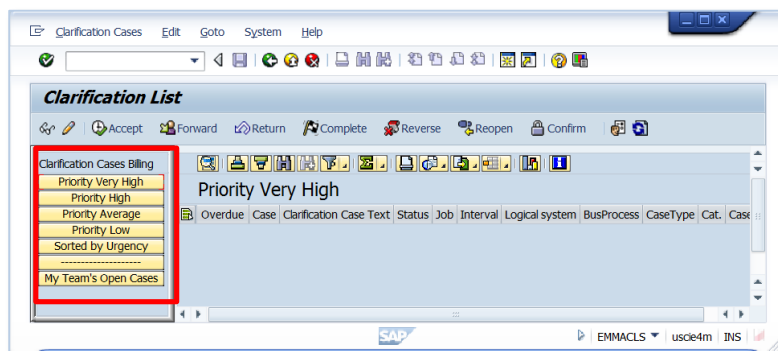


Figure 4-26 EMMCLS Transaction: Example of an Action Box with Shortcut Keys

4.3.2 Transaction EMMAC2 - Clarification Case Maintenance

The clarification case maintenance transaction is the core process of the BPBM clarification case processing. An agent can start the clarification case maintenance either from a clarification case list (as described in the previous chapter) or individually per clarification case via the `EMMAC2` transaction.

Figure 4-27 shows the clarification case maintenance screen. The upper part of the screen displays general information about the selected clarification case while the lower part of the screen consists out of various subscreens with the details of the clarification case. It is important to recognize that a clarification case always shows actual information of the related business exception. This is, because the system replaces the formal parameter as defined in the clarification case category by the actual and business exception specific information during the clarification case generation process.

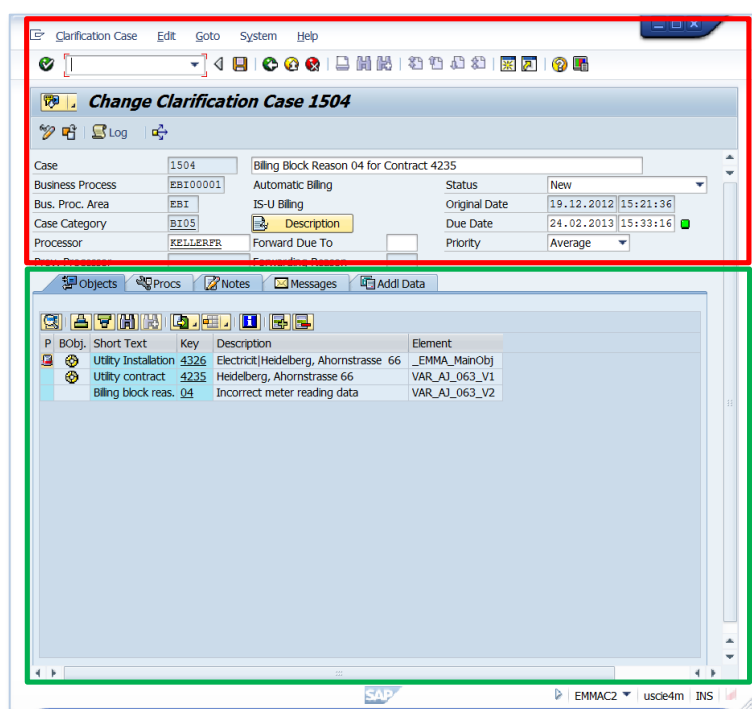


Figure 4-27 Clarification Case Maintenance Transaction

 You can define per agent which of the subscreens *Objects*, *Processes*, *Notes*, *Messages* or *Additional Data* the system shall displayed first when entering the clarification case maintenance. For this, create the `EMMA_START_SCREEN` SPA/GPA parameter in your user profile for a static definition of the start screen. Alternatively, you may define the start screen dynamically in setting the `EMMA_START_SCREEN` SPA/GPA parameter in the `CHECK_BEFORE_DISPLAY` and `CHECK_BEFORE_CHANGE` BAdIs by using the ABAP statement `SET PARAMETER`. For more details, see chapter 2.1.16 *SPA/GPA Parameters* and chapter 5.8.3.3 *Installation-specific Validations during Clarification Case Maintenance*.

The following chapters describe in details all screens and actions an agent can perform during the clarification case maintenance.

4.3.2.1 Screen *General Information*

The figure below shows the upper part of the clarification case maintenance screen with general information about the selected clarification case.

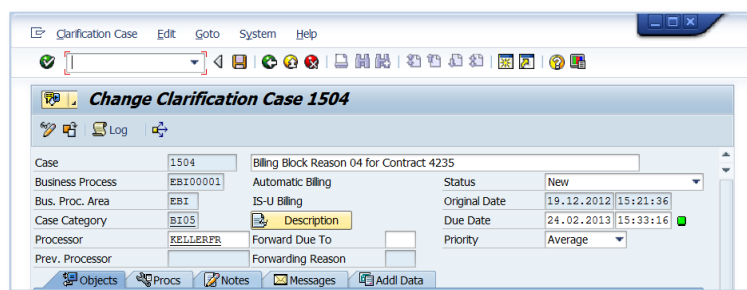


Figure 4-28 General Information about the Clarification Case

The *General Information* screen includes the following fields:

Field	Description
Case	Clarification case number
Case Text	Description of the business exception
Business Process	Business process that caused the business exception
Business Process Area	Business process area of the business process code
Case Category	Clarification case category that has been used as a template for the clarification case creation
 Description	Button to display clarification case specific work instructions how to analyze and resolve the business exception.
Processor	User ID of the current or new processor of the clarification case
Forward Due to	Reason why the clarification case is forwarded to another agent
Prev. Processor	User ID of the previous processor of the clarification case
Forwarding Reason	Reason why the previous processor forwarded the clarification case to another agent
Status	Status of clarification case, for example, <i>In Process</i>
Original Date	Date when the business exception occurred

Due Date	Due date of the clarification case as determined during the clarification case creation. It is dependent on the Customizing settings of the used clarification case category and can be based on the date of creation of the clarification case or on the date and time when the exception occurred. ☒ Indicates that the clarification case is not overdue yet ☐ Indicates that clarification case is already overdue and should be resolved as quickly as possible.
Priority	Priority of the clarification case

Figure 4-29 Description of the General Information Screen

Commonly, an agent starts working on a particular clarification case in pushing the  **Description** button to display a detailed description of the case and the detailed work instructions. Figure 4-30 shows an example of such a work instruction.

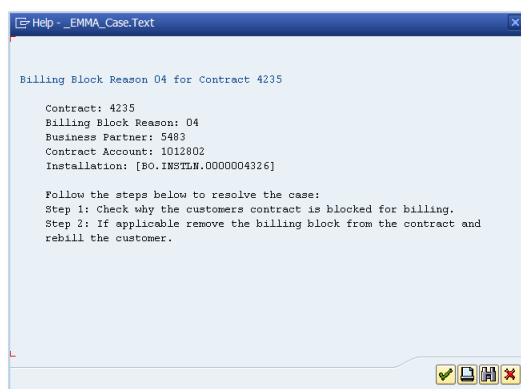


Figure 4-30 Maintenance Clarification Case: Work Instructions

The following standard functions are available in the toolbar:

Function	Use
	Switch between <i>Display</i> and <i>Change</i> mode
	Switch to another clarification case
	Display action log
	Display a list with clarification cases related to the main business object of the current clarification case
Processor Assignment	Display a list of processors allocated to the clarification case. The function is available in the menu <i>Goto</i> → <i>Processor Assignment</i>
	Reopen a clarification case that is in status <i>Completed</i> or <i>Canceled</i> . Note: This additional button is only offered when switching from display into change mode.
	Confirms a clarification case that is already in status <i>Completed</i> . Note: This additional button is offered only when switching from display into change mode.
	Generic object services with many additional functions, such as creating attachments, displaying workflows with related to the clarification case or even starting a workflow to initiate a special process for the current clarification case. Select <i>Help for Object Services</i> for a detailed description of the available generic functions.

 Description	Display clarification case specific work instructions how to analyze and resolve the business exception.
	Save changes of the clarification case to the database

Figure 4-31 Functions available during Clarification Case Maintenance



You can enhance the toolbar by installation specific buttons and functions. For more information, see chapter 6.2.1 *Enhancement of the Application Toolbar*.

The agent can perform the following additional general actions:

- **Status change of clarification case:** The agent can change the status of a clarification case in choosing a new status according to the following matrix:

Status Old/New	New	In Process	Completed	Confirmed	Canceled
New	--	X	X	--	X
In Process	--	--	X	--	X
Completed	--	X	--	X	--
Confirmed	--	--	--	--	--
Canceled	--	X	--	--	--

Figure 4-32 Matrix with possible Status Changes

If the agent decides to cancel the clarification case hence to change the status if the clarification case to *Canceled*, the agent has to enter a cancellation reason:

Figure 4-33 Maintenance Clarification Case: Status Change to *Canceled*

- **Change of processor:** The agent can change the processor of the clarification case in forwarding the clarification case to another agent or simply removing his user name from the *Processor* field. In either case the agent must enter a forwarding reason.
- **Change of due date:** The change of the due date has no immediate functional effect in the standard implementation. However, a changed due date will be reflected in reporting and in clarification case lists.
- **Change of priority:** The change of the priority has no immediate functional effect in the standard implementation. However, a change of the priority will be reflected in reporting and in clarification case lists.

4.3.2.2 Screen Objects

The figure below shows the *Object* screen with the clarification case specific objects.

Objects

Procs

Notes

Messages

Add Data

P	BObj	Short Text	Key	Description	Element
		Utility Installation	4326	Electricit[Heidelberg, Ahornstrasse 66	_EMMA_MainObj
		Utility contract	4235	Heidelberg, Ahornstrasse 66	VAR_A1_063_V1
		Billing block reas.	04	Incorrect meter reading data	VAR_A1_063_V2

EMMAC2

uscie4m

INS

Figure 4-34 Maintenance Clarification Case: Screen *Objects*

In addition to the standard ALV functions (for more details about the standard functions, see chapter 2.1.17 *ABAP List Viewer ALV*), the following application specific functions are available in the toolbar of the object list:

- **Add a business object:** The agent can add  additional objects to the clarification case. Commonly, this should only be necessary when the agents create a clarification case manually. In case of an automatically created clarification case, the system respectively the installation specific implementation should add all required objects automatically to the clarification case. For more details how to manually add an object to a clarification case, see chapter case (see chapter 4.3.3 *Transaction EMMAC1 – Create Manual Clarification Case*).
- **Remove a business object:** The agent can remove  objects from the clarification case. However, the agent should be careful in removing business objects if they are a required input to solution processes of the clarification case.



Objects with an initial key are hidden on the screen by default even they are available in the clarification case. Commonly, the keys of business objects are determined at a later stage by an automatic or manual solution process.



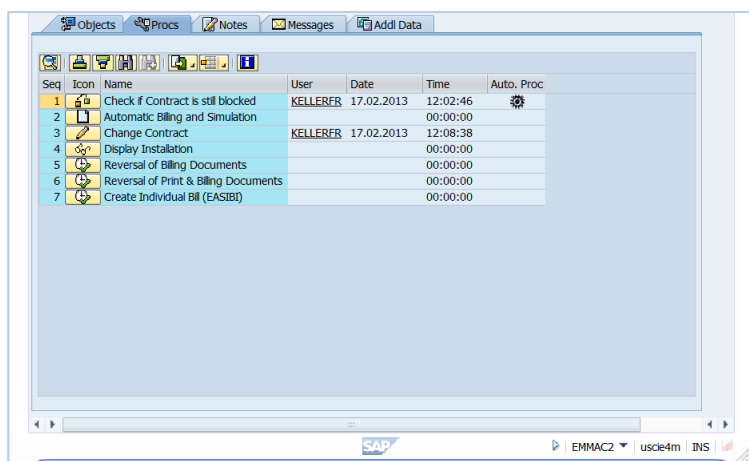
You can adapt the object list to the current clarification case data in implementing the `ADJUST_ALV OBJ_OUTPUT` method of the `BADI_EMMA_CASE_TRANSACTION` BAdI. For more information, see chapter 5.8.2.3 *BAdI Definition BADI_EMMA_CASE_TRANSACTION*.

4.3.2.3 Screen Processes

The screen *Processes* offers the functions to support the resolution of a clarification case hence business exception by means of solution processes that represent, for example, transactions and reports. The agent starts a solution process in pushing the respective button in the *Icon* column. All functions are based on the Customizing of the underlying clarification case category.



Only in change mode it is possible to start a solution process even the solution process opens a business object just in display mode.

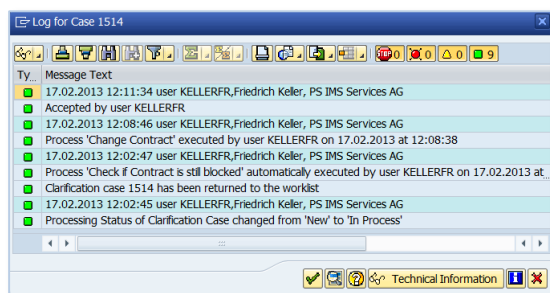


Seq	Icon	Name	User	Date	Time	Auto. Proc
1	⚙️	Check if Contract is still blocked	KELLERFR	17.02.2013	12:02:46	⚙️
2	📄	Automatic Billing and Simulation			00:00:00	
3	📄	Change Contract	KELLERFR	17.02.2013	12:08:38	
4	📄	Display Installation			00:00:00	
5	📄	Reversal of Billing Documents			00:00:00	
6	📄	Reversal of Print & Billing Documents			00:00:00	
7	📄	Create Individual Bill (EASIB)			00:00:00	

Figure 4-35 Maintenance Clarification Case: Screen Processes

The screen shows for each solution process additional information, such as when it was executed last and who started the solution process that time. The ⚙️ symbol indicates automatic solution processes that are commonly processed as part of the nightly batch chain by a specific batch user. However, the agent can execute also an automatic solution process anytime regardless if it was already processed.

The system documents the execution of solution processes in the application log 📖 of the clarification case. The figure below shows a sample application log:



Ty	Message Text
17.02.2013 12:11:34	user KELLERFR,Friedrich Keller, PS IMS Services AG
Accepted by user KELLERFR	
17.02.2013 12:08:46	user KELLERFR,Friedrich Keller, PS IMS Services AG
Process 'Change Contract' executed by user KELLERFR on 17.02.2013 at 12:08:38	
17.02.2013 12:02:47	user KELLERFR,Friedrich Keller, PS IMS Services AG
Process 'Check if Contract is still blocked' automatically executed by user KELLERFR on 17.02.2013 at	
Clarification case 1514 has been returned to the worklist	
17.02.2013 12:02:45	user KELLERFR,Friedrich Keller, PS IMS Services AG
Processing Status of Clarification Case changed from 'New' to 'In Process'	

Figure 4-36 Application Log of Clarification Case

💡 Arrange the solution processes of a clarification case carefully to support a thorough analysis and efficient resolution of the related business exception. You should instruct your agents to report missing solution processes rather than starting transactions outside of a clarification case.

💡 You can adapt the process list to the current clarification case data in implementing the `ADJUST_ALVPROC_OUTPUT` method of the `BADI_EMMA_CASE_TRANSACTION` BAdI. For more information, see chapter 5.8.2.3 *BAdI Definition BADI_EMMA_CASE_TRANSACTION*.

4.3.2.4 Screen Notes

The agent may leave comments in the available note editor to document the progress in resolving the business exception. The system adds automatically the write-protected line with the time stamp and user identification. Moreover, all previous comments including the own comments cannot be changed after you saved the clarification case.

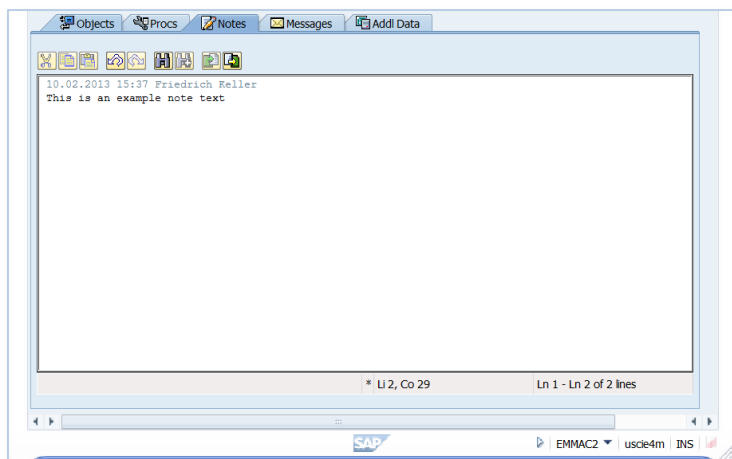


Figure 4-37 Maintenance Clarification Case: Screen Notes

4.3.2.5 Screen Messages

The screen *Messages* displays all messages from the application log of the business process that describes the related business exception completed by the message text, message class and message number.

The system displays the longtext of the message if the agent either pushes the  button or simply clicks on the message text. The *Triggering* flag identifies the messages that matched with the triggering messages of the clarification case category the system used as a template when creating the clarification case.

By default, the system displays only the messages that triggered the automatic creation of the clarification case (triggering messages). Push the  button to display all messages associated with the business exception and the  button to display only the triggering messages.

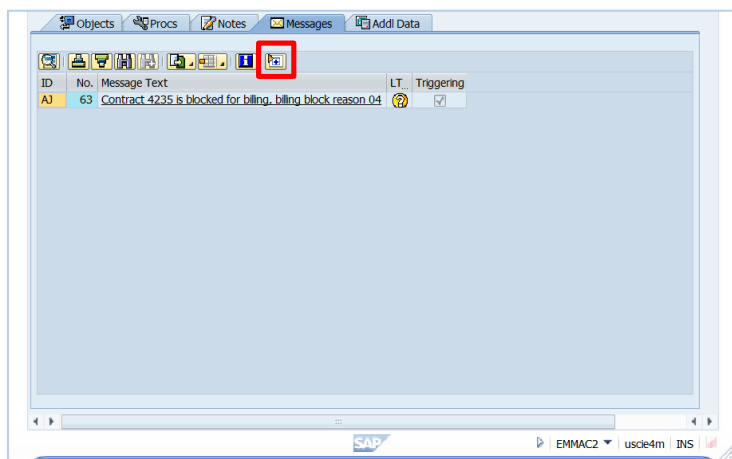


Figure 4-38 Maintenance Clarification Case: Screen Messages

 Triggering messages are only displayed and marked as such if the *Store Triggering Messages for Clarification Case* checkbox was marked in the related clarification case category (for more details, see chapter 3.4.3.2 *Screen Messages*).

 You can adapt the object list to the current clarification case data in implementing the `ADJUST_ALVMSG_OUTPUT` method of the `BADI_EMMA_CASE_TRANSACTION` BAdI. For more information, see chapter 5.8.2.3 *BAdI Definition BADI_EMMA_CASE_TRANSACTION*.

4.3.2.6 Screen *Additional Data*

The *Additional Data* screen contains two further subscreens. The *Management Data* screen displays technical details about the clarification case. In addition, the system displays the clarification case type and the authorization group. You can implement the available *Customer Data* screen as an installation specific enhancement of the clarification case (for more details, see chapter 6.1 *Installation-Specific Fields*).

The agent can change the clarification case type and authorization group on *Management Data* screen at any time. However, if the agent does not have authorization for the newly entered authorization group, the system reject the entered authorization group in raising the EMMA 241 *No authorization for authorization group* error message.

Figure 4-39 Maintenance Clarification Case: Screen *Addl. Data*

4.3.3 Transaction EMMAC1 – Create Manual Clarification Case

Commonly, a clarification case is created automatically by the respective BP EM clarification case generation transactions, such as the `FPEMMAMA` transaction (see chapter 4.2.1 *Transaction FPEMMAMA - BP EM Mass Activity*) or by a BAPI (see chapter 5.7.2.1 *Create Clarification Case*). Only in very rare situations, an agent creates clarification cases manually using the `EMMAC1` transaction. The `EMMAC1` transaction supports the creation of manual clarification cases based on clarification case categories with a clarification case creation type *Manually* (for more details about the clarification case type, see chapter 3.4.3.1 *Screen Basic Data*). The screen below shows the selection screen of the `EMMAC1` transaction. The agent must enter either the name of the clarification case category to be used as the clarification case template or the number of an existing clarification case to be copied.

Figure 4-40 Selection Screen of the `EMMAC1` Transaction

If the related clarification case category is marked as *Business Object specific*, the agent cannot save the newly created clarification case without the creation of at least the main business object on the *Object* screen. For this purpose, the agent changes to the *Object* screen and to pushes the  button. Of course, the agent may create further objects to describe the business exception. If the agent has not defined a required primary business object yet and the agent pushes the  button on the *Object* screen, the system displays a popup and requests the agent to enter the key of a business object that relates to the business object type as defined in the clarification case category.

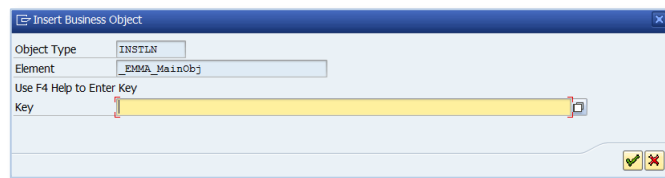


Figure 4-41 Popup to enter Business Object Key



You must enter the key of a business object with leading zeros. Alternatively, push the search help button to use the search help to identify a business object.

If the primary business object is already defined or not mandatory and the agent wants to create an additional business object by pushing the  button, the system shows the following popup. After selecting either of the *BusObject* or *Field Value* buttons, the system displays a second popup with the required fields to create a proper object for the clarification case.

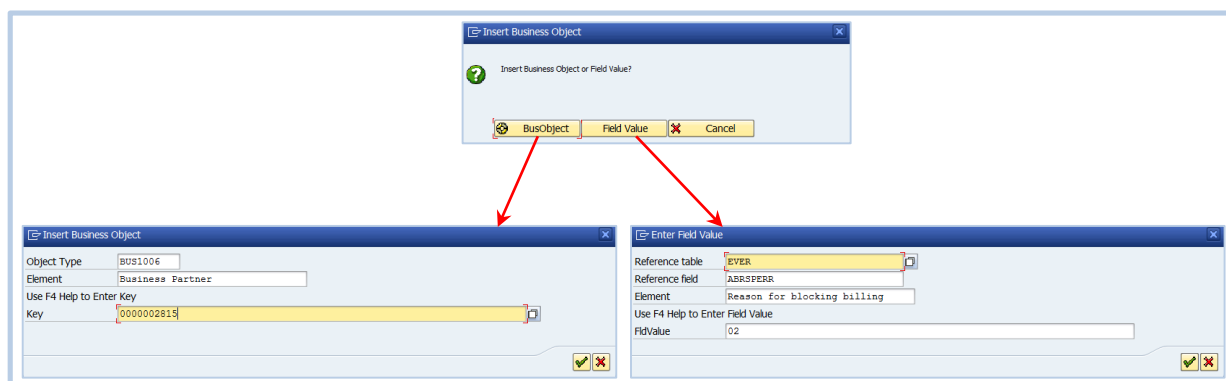


Figure 4-42 Maintenance Clarification Case: Screen Objects

When creating a business object, the agent must enter the business object related BOR object type and the key of this business object. In case of a field value, the agent must enter the technical DDIC reference and a field value. In both scenario, the agent must enter a descriptive object name of the business object, the *Element* name, that is later displayed in the clarification case.

5 Technical Implementation Details

This chapter describes many technical BPEM implementations details. The provided information supports you to enhance the BPEM transactions and to connect own installation-specific business processes to BPEM. However, all described technical details are subject to change without notice.

5.1 General Information

This chapter describes the most important BPEM specific database tables.



SAP does not recommend to access or manipulate directly the BPEM database tables. Always use the delivered and released business application interfaces and BAPIs instead.

5.1.1 Table EMMA_HDR – BPEM Header Table

The starting point of any BPEM processing step is the BPEM job. The EMMA_HDR table stores general information about a BPEM job including:

- Unique BPEM job identification (RUNID field)
- Application specific external job number (EXTRUNID field)
- Business process information (business process code, application specific information to uniquely identify the specific business process and its application logs, transaction code of the executed business process)
- Status information which BPEM process steps have already been executed
- Statistics of processed business objects (total number of processed business objects, number of successfully processed business objects, number of business objects that failed with severity levels *Error* and *Warning* and total number of messages)
- Number of generated clarification cases.

5.1.2 Table EMMA_JOBRUNIDMSG – Application Messages

The EMMA_JOBRUNIDMSG table is a cluster table BPEM uses to store high volumes of data in various structures with the advantage of a fast block access to the data. In this table, BPEM stores mainly the collected and prepared application messages.

The system uses the following so-called cluster areas when accessing the EMMA_JOBRUNIDMSG cluster table:

RELID	ID (SRTFD field)	Usage
KK	BPEM job identification	Application message
LT	BPEM job identification	Interval information accessed by child jobs of the FPEMMAMA mass activity
JI	Mass activity run key	No of intervals per BPEM job to be processed by FPEMMAMA mass activity

Figure 5-1 Overview Areas of EMMA_JOBRUNIDMSG cluster table

5.1.3 Table EMMA_INT – Interval Statistics

The EMMA_INT table stores detailed process statistics on interval level if the related business process was executed via a mass activity. BPEM stores statistics on an interval level regarding the processed business object statistics and status information.

5.1.4 Table EMMA_CASE – Clarification Case Header Table

BPEM stores general information of a clarification case in the EMMA_CASE table including:

- Unique identification of a clarification case (CASENR field)
- Identification of the related BPEM job (RUNID field)
- Business process information (business process code, main object type and identification of the business object)
- Clarification case category used to create the clarification case
- Current priority of the clarification case
- Further administrative data, such as creation date, due date, current and previous processor
- The customer include CI_EMMA_CASE with installation-specific fields of a clarification case

5.1.5 Table EMMA_CACTOR – Clarification Case Processors

The EMMA_CACTOR table stores the result of the agent determination rule at the time the system created the clarification case (of course only if such a rule was assigned to the related clarification case category). The result includes the determined entity of your organizational structure, such as work centers, positions or simply user IDs. The system accesses the EMMA_CACTOR table when selecting clarification cases based on organizational structure criteria. Commonly, the EMMA_CACTOR table is updated via an installation-specific forward method (for more details, see chapter 6.4.2 *Forwarding Rule*).

5.1.6 Table EMMA_COBJECT – Objects and Field Values for Case

The EMMA_COBJECT table stores the clarification case specific business objects and field values of the clarification case container and their keys. The CELEMNAME field uniquely identifies each of these elements. The further fields REFOBJECTTYPE and ID describe an existing business object by its BOR object type and key, while the REFSTRUCT, REFFIELD and ID fields describe the value with reference or type to REFSTRUCT and REFFIELD.

5.1.7 Table EMMA_CMSG_LINK – Triggering Messages

The EMMA_CMSG_LINK table stores the triggering messages of a clarification case.

5.1.8 Table EMMA_CSOLP – Solution Processes

The EMMA_CSOLP table stores the information concerning the attached solution processes and the status of their last execution. During the clarification case creation, the system copies only the links of the solution processes of the related clarification case category and resets the status of each copied solution process. However, the system does not copy the data flow definition of the solution processes hence they remain with the clarification case category and the system retrieves them from there whenever the agent or the system triggers the execution of a solution process.



Be careful in changing the solution process related Customizing of a clarification case category since any change affects all open clarification cases respectively their attached solution processes.

5.2 BPEM Job preparation

The system performs the following steps during a BPEM job analysis. The term *application* is used to describe the report or transaction that created the BPEM job including the application specific information, such as error messages, that is processed by BPEM during the job preparation task:

1. Determine the processing status of the application (completed, still active, etc.)
2. Read interval information if the application is implemented as a mass activity
3. Determine the log numbers of the relevant application logs if the application stored its messages in application logs
4. Read the application messages either from application logs or from a different source. If the message information is retrieved from a source other than an application log, the information must be converted to messages as used and stored in application logs by application specific implementation of the related business process code.
5. Sort, analyze and group the messages
6. Save the resulting message list in own BPEM database tables for a later rapid access

These steps above are implemented differently for BPEM and the previous versions of the *Classic* EMMA and the respective transactions. The following chapters explain the implementation in more details.

5.2.1 Analysis and Grouping of Messages

After you activated BPEM for a particular business process, the related application stores additional BPEM specific control messages to enclose the application messages (describing the business exception) with special control messages. By default, BPEM indicates the business object related application messages by an `EMMA 011` message and concludes the application messages by an `EMMA 012` message. The content of the message variables of these BPEM control messages describe the business exception specific business process code, the primary object type and the ID of the processed business object.

Many applications store BPEM control messages even after a successful execution of a business process. This is a desired behavior since such an implementation allows a later creation of process statistics showing the totals not only of failed but also of successfully processed business objects.



During the BPEM preparation step, the application messages are collected and processed either all together (no mass activity) or per interval (mass activity). For more information about the interval determination of the `FPEMMA` mass activity, see chapter 5.6.2 *Interval Determination*.

Possible sources of application messages can be one or multiple application logs or an application specific storage such as database tables. In the latter case, you must implement a specific ABAP OO method to transform application specific data into application log messages (for more implementation details, see the next chapters). In addition, you need to adjust the Customizing for your transaction or business process accordingly (for more details, see chapter 3.6.2 *Define Business Processes for Message Management*).

After the successful collection of all messages, BPEM sorts the messages by their time stamps. Subsequently, BPEM analyses the messages technically and applies the Customizing rules (for more details, see chapter 3.3.4 *Suppression of Messages for Log Preparation* and chapter 5.8.1.1 *BAdI Definition BADI_EMMA_JOB_ANALYSIS*.)

5.2.3 Reversal of Job Preparation

The application performs a reversal of a job preparation in deleting all job related entries in the `EMMA_JOBRUNIDMSG` and `EMMA_INT` tables, followed by a change of the status of the BPEM job header to reflect the reversal. Of course, a reversal of a job preparation is only possible if the process step *Job Preparation* was performed previously, and the process step *Clarification Case Creation* was either not yet carried out or has been reversed.

5.2.4 Technical Implementation Job Preparation New EMMA

SAP delivers the `CL_EMMA_JOB_ANALYSIS` ABAP OO superclass with the following methods to process the BPEM job analysis step.

Internal Step	Method
1 – Determine Processing Status	<code>GET_JOB_PROCESSING_STATUS</code>
2 – Get Interval Information	<code>GET_INTERVALS</code>
3 – Get Application Log Numbers	<code>GET_LOG_NUMBER</code>
4 – Read Messages	<code>GET_MESSAGES</code>
5 – Analyze Messages	<code>ANALYZE_MESSAGES</code>

Figure 5-3 Methods of the `CL_EMMA_JOB_ANALYSIS` ABAP OO Superclass

In addition, SAP delivers the following subclasses of the `CL_EMMA_JOB_ANALYSIS` superclass for the most common uses cases.

Subclass	Use Case
<code>CL_EMMA_JOB_SINGLE</code>	Single process with one application log containing both application and BPEM related control messages
<code>CL_FKK_JOB_MASS_RUN</code>	Mass activity with application logs per processed interval. The external IDs of the application logs follow the design rules of a mass activity.
<code>CL_ISU_JOB_MASS_RUN</code>	Mass activity with application logs per processed interval. The external IDs of the application logs follow the design rules of IS-U specific mass activity
<code>CL_EMMA_ANALYZE_JOB_LOG</code>	Single background job with a job log containing both application and BPEM related control messages

Figure 5-4 Implemented Subclasses of the `CL_EMMA_JOB` ABAP OO Superclass

SAP implemented further subclasses to deal with the special requirements of business processes that do not use application logs to store information and error messages.

Technically, a subclass inherits all the attributes and methods of the referenced superclass. If your business process does not follow one of the mentioned designs you need to implement an own ABAP OO class for a process specific job analysis. Use the `SE24` transaction to create a new ABAP OO class and enter the `CL_EMMA_JOB_ANALYSIS` class as the superclass. You may redefine one or more of the methods that have been inherited from the superclass. You may use the implementation of the `CL_EMMA_JOB_ANALYSIS` class as a basis for an own development.

5.2.5 Technical Implementation Job Preparation *Classic EMMA*

SAP delivers several ABAP OO interfaces with the following methods to process the 5 BPEM job analysis steps.

Internal Step	Interface~Method
1 – Determine Processing Status	(IF_EMMA_JOB_ANALYZE~ANALYZE_JOB)
2 – Get Interval Information	IF_EMMA_INTINFO_GET~GET_MARUN_STATUS IF_EMMA_INTINFO_GET~MASSACT_INTV_INFO IF_EMMA_INTINFO_GET~GET_MASSACT_INTERVAL
3 – Get Application Log Numbers	IF_EMMA_LOGNUMBER_GET~GET_LOG_NUMBER
4 – Read Messages	(IF_EMMA_JOB_ANALYZE~ANALYZE_JOB)
5 – Analyze Messages	IF_EMMA_LOGMSG_PROCESS~ANALYZE_MSGS IF_EMMA_LOGMSG_PROCESS~VALIDATE_MSG IF_EMMA_LOGMSG_PROCESS~SUPPRESS_MSG

Figure 5-5 ABAP OO Interfaces for BPEM job Analysis

The IF_EMMA_JOB_ANALYZE~ANALYZE_JOB ABAP OO method carries out the first BPEM job analysis step. For the internal step 2 *Get Application Log Numbers* the IF_EMMA_LOGNUMBER_GET~GET_LOGNUMBER ABAP OO method is processed and for step 5 *Analyze Messages* the IF_EMMA_LOGMSG_PROCESS~ANALYZE_MSGS ABAP OO method.

SAP implemented the following ABAP OO classes for the most common uses cases.

Field	Non-Mass Activity	Mass Activity
<i>Job Analysis Class</i>	CL_EMMA_ANALYZE_JOB_APPLLOG	CL_EMMA_ANALYZE_JOB_APPLLOG
<i>Log Number Class</i>	CL_EMMA_GET_LOG	CL_EMMA_MALOG
<i>Interval Info Class</i>		CL_EMMA_MA_INTINFO
<i>Analysis Class</i>	CL_EMMA_PROC_APPLLOGMSGSGS_V2	CL_EMMA_PROC_APPLLOGMSGSGS_V2

Figure 5-6 Implemented ABAP OO Classes for BPEM job Analysis

5.3 Clarification Case Creation

Clarification cases can be created automatically or manually. In either case, the creation of a clarification case is based on a clarification case category and the respective case creation type *Automatically* or *Manually*. The **automatic creation of clarification cases** is a simple process. However, the variety of options to define a clarification case category, performance considerations and the innumerable options for installations specific enhancements makes the process more complex. In contrary, the **manual creation of clarification cases** is much simpler and can be done via the EMMAC1 transaction or with one of the available BAPIs.

The following chapters describe the internal processes of creating clarification cases in more details.

5.3.1 Automatic Creation of Clarification Cases

5.3.1.1 General Process

The process of creating clarification cases automatically uses the results of the previous process step *BPEM Job Preparation* and its message grouping. As said above, BPEM groups the messages by the related business object and its identification as processed by the application.

During the automatic clarification case creation process, BPEM processes all message groups of an application one by one within one processing unit. If the process is a mass activity, the processing unit includes all message groups of an interval. BPEM persists the created clarification cases on the database after processing all message groups of the processing unit.

The automatic creation of clarification cases consists out of the following steps:

1. Fetch the next message group
2. Create a list of clarification case categories to be processed: (a) Get all clarification case categories with creation type *Automatically* of the relevant business process code and sort them by name in ascending order. (b) Add special clarification case categories defined for the special business process codes `EMMACAS1` and `EMMACAS2`.
3. Process the list of selected clarification case categories as follows:
 - If there are triggering messages defined, compare the messages in message group with the triggering messages of the clarification case category in processing the `DETERMINE_CASE` method of the `CL_EMMA_CASE` ABAP OO class. Exit the further processing of the current clarification case category if there are no matching triggering messages.
 - If the *Business Add-In* field of the clarification case category is marked, process the `DETERMINE_CASE` method of the `EMMA_CASE` BAdI. Exit the further processing of the current clarification case category if the BAdI raises the exception `NO_CASE`.
 - If the *Complex Condition* field of the clarification case category is marked, process the entered condition; exit the further processing of the current clarification case category if the clarification case data or data environment does not meet the condition.
 - If the *Check for Existence of Identical Clarification Cases* field of the clarification case category is marked, check for existing duplicate clarification cases, and process subsequently the `CHECK_IDENTICAL_CASE` method of the `EMMA_CASE` BAdI to overwrite the result of the previous standard check if applicable.
 - Create the clarification case in memory
 - Process the `COMPLETE_CASE` method of the `EMMA_CASE` BAdI. If the BAdI raises the `NO_CASE` exception, remove the created clarification case from the memory and exit the further processing of the current clarification case category
4. Persist clarification case from memory on the database.
5. If there is a forwarding method attached to the clarification case category (possibly changed by the `COMPLETE_CASE` method of the `EMMA_CASE` BAdI, process the forwarding method.



The system creates a clarification case for any matching clarification case category thus the system may create multiple clarification cases for the same message group.

You should only take advantage of the option to check for existing identical clarification cases in scenarios where there is a real risk of creating duplicated clarification cases. Be aware that this check could become a database related performance issue because the BPEM standard check tests the already available clarification cases if they meet the following conditions:

- Same clarification case category
- Same IDs of the main business object
- Same triggering messages and identical content of the related message variables

It may occur that the system creates duplicate clarification cases despite previous checks for existing duplicates. This happens commonly when concurrent BP EM jobs of the `FP EM MAMA` mass activity process similar (identical) message groups of separate intervals and application logs, and test the available clarification cases at the same time.

5.3.1.2 Clarification Case Categories not specific to a Business Object

You commonly define clarification case categories specific to a business exception and associated with a particular business object. However, you can also define clarification case categories being independent of the context of a business object (see figure below for the *Category Business Object Specific* indicator).

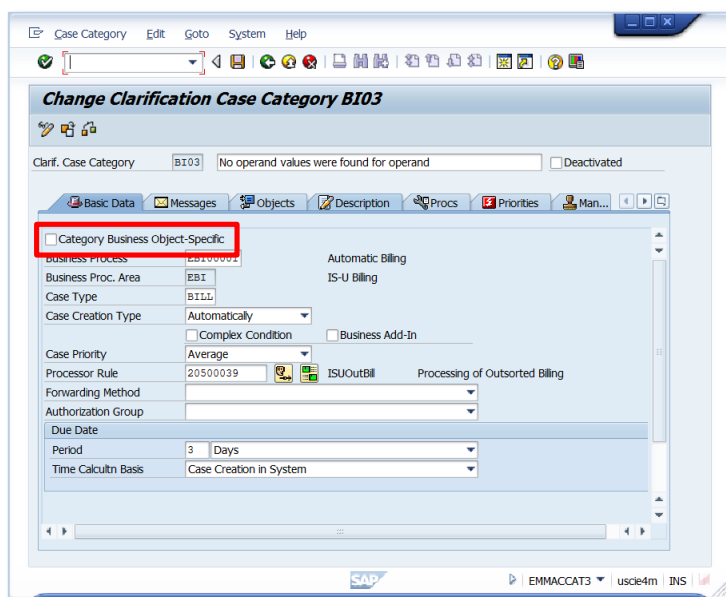


Figure 5-7 *Category Business Object-Specific* field

BP EM processes such as a clarification case category for all available message groups and regardless if a business object is associated with the message group. However, BP EM excludes this clarification case category from further processing once a clarification case was created based on the same clarification case category. This design prevents a repeatedly creation of clarification cases for a recurring message for the same clarification case category. BP EM processes all clarification case category again when commencing with the next processing unit of message groups.

5.3.1.3 Special Business Process Codes `EMMACAS1` and `EMMACAS2`

There are the two special `EMMACAS1` and `EMMACAS2` business process codes that no standard business process use to report business exceptions. Clarification case categories assigned to the `EMMACAS1` and `EMMACAS2` business process codes are always processed regardless of the business process code allocated to the processed message group. This allows you to define cross-application clarification case categories only once instead of one per business process code.

In contrast to clarification case categories assigned to the `EMMACAS1` business process code, the system processes clarification case categories assigned to the `EMMACAS2` business process code only if previously no clarification case was created for the currently processed

message group. This allows you to create clarification cases for messages for which you have not defined yet a special clarification case category. For more details and uses cases, see chapter 6.3.2 *Process Code Specific Sweeper Clarification Case Categories*.

5.3.2 Manual Creation of Clarification Cases

You create clarification cases manually either with the `EMMAC1` transaction (for a detailed description, see chapter 4.3.3 *Transaction EMMAC1 – Create Manual Clarification Case*) or with one of the available BAPIs (for a list of available BAPIs, see chapter 5.7.2.1 *Create Clarification Case*). For a manual clarification case creation only clarification case categories with creation type *Manually* can be used.

During the creation of manual clarification cases, the following peculiarities apply compared to the automatic creation of clarification cases:

- The system does not check for existing duplicate clarification cases
- The priority of the clarification case is either defined by the caller of the BAPI, or taken from the default priority of the provided clarification case category. The system does not automatically redefine the priority of a manual clarification case based on the conditions as defined on the Priorities screen of the underlying clarification case category (for more information, see chapter 3.4.3.6 *Screen Priorities*).
- The forwarding method to allocate agents to the clarification case is performed only on request (`FORWARD` parameter)
- You must have the authorization to create clarification cases for the entered authorization group.



The system creates an own BPEM job for every manual clarification case to allow a selection of also these clarification cases for a possible execution of automatic solution processes.

5.4 Automatic Solution Processes

You can define solution processes the system shall execute automatically during background processing hence without being explicitly triggered by an agent. Commonly, you process them via the `FPEMMAMA` transaction. In addition, you can manually trigger the execution of automatic solution processes with the `EMMAJL` and `EMMAJP` transactions.

The system processes automatic solution processes of a clarification case only if the status of the clarification is either *New* or *In Processing*. The system submits the `REMA_CLIST_AUTPROC_EXEC_INT` report to execute the solution processes of a clarification case sequentially in an own internal session. This implementation prevents side effects when executing solution processes for multiple successive clarification cases.

Because you can define the data flow of a solution process such that returns values for one or more objects of the clarification case container, the system updates the objects of the clarification case with returned data (commonly object numbers) in accordance to the defined data flow of the respective solution process after the execution of the solution process.

The system blocks a clarification case against concurrent access while processing the attached automatic solution processes. If the system fails to block the clarification case, the system skips the execution of all solution processes of the clarification case.

5.5 Clarification Case Application Log

The system maintains an application log per clarification case for a later review of the case related actions. Application logs for clarification cases are created with the `EMMA` application code (log object) and the `CASE` sub-object. You can uniquely identify the application log of a clarification case by its external ID that is built as a concatenation of the related BPDM job number (`EMMA_HDR-RUNID` field) and the case number (`EMMA_CASE-CASENR` field) separated by a hyphen.

 The application log of the clarification case 0000000876 and the related BPDM job 0100019345 can be found with the search criteria *Log Object* `EMMA`, *Subobject* `CASE` and external ID 0100019345-0000000876.

5.6 Mass Activity FPEMMAMA

5.6.1 General Information

To the `FPEMMAMA` mass activity the `EMMR` mass activity type is assigned. You find more details about the detailed configuration of the `FPEMMAMA` mass activity in the IMG at *Financial Accounting (New) → Contract Accounts Receivable and Payable → Technical Settings → Prepare Mass Activities*.

 The `FPEMMA` mass activity is prepared to process not only business process codes with *New EMMA Customizing* but also business process codes that are still configured with only the *Classic EMMA Customizing* (for more information, see chapter 3.6.2 *Define Business Processes for Message Management*).

5.6.2 Interval Determination

The `FPEMMAMA` mass activity uses always the `EMMA_JOB` parallelization object. This parallelization object is based on the BPDM job number (for more details about BPDM jobs, see chapter 2.1.12 *BPDM Job*).

When you start the `FPEMMAMA` mass activity, the mass activity framework processes the `FKK_EMMA_JOB_VARIANT_ADJUST` function module for a dynamic interval determination of the interval ranges to be processed (see `V_TFKDI` table view for more details about the Customizing of the `EMMA_JOB` parallelization object). This function module build the interval ranges of the BPDM slave jobs based on either the number of objects per interval or the total number of intervals to be processed in accordance the given variant for. If you do not provide a variant for parallel processing, `FPEMMAMA` mass activity checks all possible BPDM job numbers in the range of `SPACE - 99999999999999999999`.

The `FKK_EMMA_JOB_VARIANT_ADJUST` function checks all BPDM jobs matching the entered conditions for their number of intervals if related to a mass activity. Per interval, a new number is composed using the BPDM job number (10 characters) followed by an interval number (10 characters). If the BPDM job does not relate to a mass activity, the dummy interval number 0000000000 is used).

 If there is a BPDM job with the number 0000018298 related to a mass activity with 100 processed intervals, the system creates a list of 100 internal identifications from 00000182980000000001, 00001829800000000002 to 000001829800000000100.

The system eventually determines the final interval ranges based on all newly composed identifications and according to the setting of the defined variant for parallel processing.

5.6.3 Application Log

The `FPEMMAMA` transaction creates an own BP EM job and related application logs. The application logs are created with the `FICA` application code (log object) and the `MAEMMR` sub-object. The external ID of the application log is built using the prefix `EMMR`, the data ID in the `yyyymmdd` representation and the identification of the mass run. In addition, further interval information is added to the external ID.

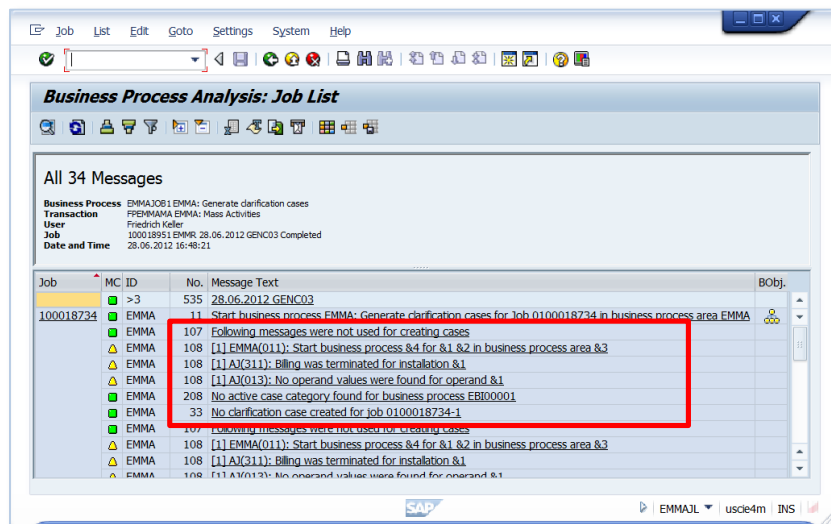


Figure 5-8 Application Log of the FPEMMAMA Mass Activity

You must take special care on the `FPEMMAMA` application log that is created in operating mode *Generate Clarification Cases*. In this application log, you can find alerts, for example, indicating that there are business exceptions without a matching clarification case category to create a clarification case (for more details how you can deal with business exceptions without a matching clarification case category, see chapter 6.3 *Error Message without Clarification Case*).

5.7 Business Application Programming Interface

SAP provides a comprehensive Business Application Programming Interface (BAPI) for BP EM functions. BAPIs enable access to SAP functions via formal, stable and dialog-free interfaces. These interfaces can be used by external applications developed by customers and complementary software partners as well as by other SAP applications. Defined as API methods of SAP business object types, these object types are used within the Business Framework to enable object-based communication between components. The provided BAPIs are based on the technologies function modules, objects of the BOR (and their methods) and web services

SAP delivers two sets of BAPIs:

- **EMMAJob** BP EM process list
- **EMMACase** Clarification Case

Use the following transactions to display details of the BAPIs. Both transactions allow searching for BP EM related BAPIs and services with the search string `EMMA*`.

- Transaction `BAPI` (*BAPI Explorer*) to explore BP EM BAPIs
- Transaction `SOAMANAGER` (*SOA Manager*) to explore BP EM services
- Transaction `SWO1` (*Business Object Builder*) to explore BOR objects

5.7.1 BAPI `EMMAJob`

5.7.1.1 Create BPEM Job

BOR Method `EMMAJob.Create`

BAPI `BAPI_EMMA_JOB_CREATE`

Web Service `EMMA_JOB_CREATE`

Use this function to register your process in BPEM as a BPEM (relevant) job.

5.7.1.2 Execute BPEM Job Preparation

BOR Method `EMMAJob.Prepare`

BAPI `BAPI_EMMA_JOB_PREPARE`

Web Service `EMMA_JOB_PREPARE`

Use this function to execute the BPEM process step *BPEM Job Preparation* of a BPEM job.

5.7.1.3 Generate Clarification Cases

BOR Method `EMMAJob.GenerateCases`

BAPI `BAPI_EMMA_JOB_GENERATE_CASES`

Web Service `EMMA_JOB_GENERATE_CASES`

Use this function to execute the BPEM process step *Create Clarification Cases* to create clarification cases of a BPEM job.

5.7.1.4 Execute Automatic Solution Processes

BOR Method `EMMAJob.ExecuteProcesses`

BAPI `BAPI_EMMA_JOB_EXECUTE_PROCESS`

Web Service `EMMA_JOB_EXECUTE_PROCESS`

Use this function to execute the BPEM process step *Execute Automatic Solution Process* of clarification cases related to a BPEM job.

5.7.1.5 Delete Clarification Cases

BOR Method `EMMAJob.DeleteCases`

BAPI `BAPI_EMMA_JOB_DELETE_CASES`

Web Service `EMMA_JOB_DELETE_CASES`

Use this function to execute the BPEM process step *Delete Clarification Cases* of clarification cases related to one BPEM job.

5.7.1.6 Delete BPEM Job preparation

BOR Method `EMMAJob.DeletePreparation`

BAPI `BAPI_EMMA_JOB_DEL_PREPARATION`

Web Service `EMMA_JOB_DEL_PREPARATION`

Use this function to reverse the BPEM process step *BPEM Job Preparation* of a BPEM job.

5.7.2 BAPI `EMMACase`

The `EMMACase` BAPI provides all functions to work with clarification cases. You find the related BAPI function modules in the `EMMA_API` function group.

5.7.2.1 Create Clarification Case

BOR Method `EMMACase.Create`

BAPI `BAPI_EMMA_CASE_CREATE`

Web Service `EMMA_CASE_CREATE`

Use this function module to create a manual clarification case. You have to provide all the information required to generate a clarification case.

5.7.2.2 Change Clarification Case

BOR Method `EMMACase.Change`

BAPI `BAPI_EMMA_CASE_CHANGE`

Web Service `EMMA_CASE_CHANGE`

Use this function to change an existing clarification case. You can change any information related to the clarification case. Changes to clarification case fields are subject to authorization checks.

5.7.2.3 Get Details of Clarification Case

BOR Method `EMMACase.Get_Detail`

BAPI `BAPI_EMMA_CASE_GET_DETAIL`

Web Service `EMMA_CASE_GET_DETAIL`

Use this function to get detail data of a clarification case.

5.7.2.4 Assign Processor to Clarification Case

BOR Method `EMMACase.Accept`

BAPI `BAPI_EMMA_CASE_ACCEPT`

Web Service `EMMA_CASE_ACCEPT`

Use this function to assign a new processor to a clarification case. This function can only be performed if no processor is assigned yet and the status of the clarification case is either *New* or *In Process*.

5.7.2.5 Forward Clarification Case

BOR Method `EMMACase.Forward`

BAPI `BAPI_EMMA_CASE_FORWARD`

Web Service `EMMA_CASE_FORWARD`

Use this function to forward a clarification case to a new processor. After the new processor was assigned the status of the clarification case is changed to *In Process*. This function can only be performed if the status of the clarification case is either *New* or *In Process*.

5.7.2.6 Return Clarification Case to Queue

BOR Method `EMMACase.BackToQueue`

BAPI `BAPI_EMMA_CASE_BACK_TO_QUEUE`

Web Service `EMMA_CASE_BACK_TO_QUEUE`

Use this function module to re-assign the previous processor to a clarification case. This function can only be performed if the status of the clarification case is either *New* or *In Process*.

5.7.2.7 Reopen Clarification Case

BOR Method `EMMACase.Reopen`

BAPI `BAPI_EMMA_CASE_REOPEN`

Web Service `EMMA_CASE_REOPEN`

Use this function module to change the status of a clarification case from either *Completed* or *Canceled* to the status *In Process*.

5.7.2.8 Cancel Clarification Case

BOR Method `EMMACase.Cancel`

BAPI `BAPI_EMMA_CASE_CANCEL`

Web Service `EMMA_CASE_CANCEL`

Use this function module to change the status of a clarification case from either *New* or *In Process* to the status *Canceled*.

5.7.2.9 Complete Clarification Case

BOR Method `EMMACase.Complete`

BAPI `BAPI_EMMA_CASE_COMPLETE`

Web Service `EMMA_CASE_COMPLETE`

Use this function module to change the status of a clarification case from either *New* or *In Process* to the status *Completed*.

5.7.2.10 Confirm Clarification Case

BOR Method `EMMACase.Confirm`

BAPI `BAPI_EMMA_CASE_CONFIRM`

Web Service `EMMA_CASE_CONFIRM`

Use this function module to change the status of a clarification case from *Completed* to the status *Confirmed*.

5.7.2.11 Event Clarification Case Created

Event `EMMACASE.Created`

This event is created when a clarification case is created. You use this event to start, terminate or continue own tasks and workflows after a clarification case was created.

5.7.2.12 Event Clarification Case Changed

Event `EMMACASE.Changed`

This event is created when a clarification case is changed. You use this event to start, terminate or continue own tasks and workflows after a clarification case was changed.

5.8 Customer Enhancements

SAP provides Business Add-Ins (BAdIs) to implement BPEM specific enhancements with the SAP Enhancement Framework (for more details about BAdIs and the SAP Enhancement Framework see chapter 2.1.15 *SAP Enhancement Framework*).

The following three BAdIs enable you to add functions to the BPEM standard implementation to accommodate own requirements too specific to be included in the SAP standard delivery:

- **Enhancement spot `EMMA_JOB`:** Enhancements of the BPEM job processing:
 - `BADI_EMMA_JOB_ANALYSIS` Enhancement of the BPEM job preparation step
 - `BADI_EMMA_JOB_WORK_LIST` Menu extensions of BPEM job lists
- **Enhancement spot `EMMA_CASE`:** Enhancements of the clarification case processing:
 - `BADI_EMMA_CASE` Enhancement of the clarification case transaction
 - `BADI_EMMA_CASE_TRANSACTION` Menu Enhancements of clarification case transaction
 - `BADI_EMMA_CASE_WORK_LIST` Menu extensions of clarification case lists
- **Classic BAdI `EMMA_CASE`:** General Business Add-In for clarification case processing

All BAdIs are explained in details in the following chapters.

5.8.1 Enhancement Spot `EMMA_JOB`

5.8.1.1 BAdI Definition `BADI_EMMA_JOB_ANALYSIS`

You implement the `BADI_EMMA_JOB_ANALYSIS` BAdI definition to enhance the default message processing during the BPEM job preparation step. The `BADI_EMMA_JOB_ANALYSIS` BAdI definition includes the `IF_BADI_EMMA_JOB_ANALYSIS` ABAP OO interface with the following methods:

Method	Description
<code>CHECK_FORMATTING_MESSAGE</code>	The standard implementation checks the message text and the message variables for any useful information. A message is suppressed hence removed from the message list if only one or more of the following characters can be found in either message text or message variable @#%\$^&*() ?/\ -_+={} [] : ; . " ' ' < > ,
<code>MODIFY_MESSAGE_TYPE</code>	The standard implementation modifies the message type of a message according to the Customizing.
<code>CHECK_SUPPRESSED_MESSAGE</code>	The standard implementation removes a message from the message list according to the Customizing.

Figure 5-9 Methods of the `IF_BADI_EMMA_JOB_ANALYSIS` ABAP Interface



Your implementation replaces the standard implementation of the respective interface methods of the `CL_EMMA_JOB` ABAP OO class that implements the standard Customizing. You may use the implementation of these interface methods as a basis for your own implementation.

5.8.1.2 BAdI Definition `BADI_EMMA_JOB_WORK_LIST`

You implement the `BADI_EMMA_JOB_WORK_LIST` BAdI definition to enhance the processing of BPEM job lists when using the `EMMAJL` transaction (`REMAJOBLIST` report). The BAdI definition includes a menu enhancement and an ABAP OO interface.

With the menu enhancement, you add installation-specific buttons to the application toolbar of the `EMMAJL` transaction. The `BADI_EMMA_JOB_WORK_LIST` BAdI definition includes the `IF_BADI_EMMA_JOB_WORK_LIST` ABAP OO interface with the following method:

Method	Description
CALL_ENHANCEMENT_FUNCTION	The system processes this method if the agent pushes one of the buttons you implemented via the menu enhancement. The respected function codes <code>+EF1</code> , <code>+EF2</code> and <code>+EF3</code> and are transferred to the method with the <code>IV_UCOMM</code> parameter.

Figure 5-10 Method of the `IF_BADI_EMMA_JOB_WORK_LIST` ABAP OO Interface

5.8.2 Enhancement Spot `EMMA_CASE`

5.8.2.1 BAdI Definition `BADI_EMMA_CASE_WORK_LIST`

You implement the `BADI_EMMA_CASE_WORK_LIST` BAdI definition to enhance the processing of clarification case lists when using the `EMMACL` and `EMMACLS` transactions. The `EMMACL` transaction executes the `REMMACASELIST` report and the `EMMACLS` transaction the `REMMACASELIST_SHL` report respectively. The BAdI definition includes a menu enhancement and an ABAP OO interface.

With the menu enhancement, you add installation-specific functions to the menu of both `EMMACL` and `EMMACLS` transactions and, in addition, respective buttons to the application toolbar of the `EMMACLS` transaction. The `BADI_EMMA_CASE_WORK_LIST` BAdI definition includes the `IF_BADI_EMMA_CASE_WORK_LIST` ABAP OO interface with the following methods:

Handling of Function Codes

Method	Description
EXCLUDE_FCODES	The system processes this method once when entering the agent enters a clarification case list. You may deactivate any (including the installation-specific) function of the GUI in adding the respective function code to the <code>ET_FCODE</code> table. Refer to the GUI Status <code>STATUS</code> of the <code>REMMACASELIST_SHL</code> report for the available function codes.

Figure 5-11 Methods of the `IF_BADI_EMMA_CASE_WORK_LIST` ABAP OO Interface related to Function Codes

Enhancement Functions

Method	Description
CALL_ENHANCEMENT_FUNCTION	The system process this method if the agent pushes one of the buttons you implemented with the menu enhancement. The function codes are <code>+EF1</code> , <code>+EF2</code> and <code>+EF3</code> and are transferred to the method with the <code>IV_UCOMM</code> parameter. You may use the <code>SY-TCODE</code> system field if you want to execute functions specific to the <code>EMMACL</code> or <code>EMMACLS</code> transaction.

Figure 5-12 Methods of the `IF_BADI_EMMA_CASE_WORK_LIST` ABAP OO Interface related to Enhancement Functions

5.8.2.2 BAdI Definition BADI_EMMA_CASE

The system processes the implementation of the `BADI_EMMA_CASE` BAdI definition after the agent selected a clarification case from the clarification case list to be displayed or changed, or when executing the `EMMACL` or `EMMACLS` transaction. You implement `BADI_EMMA_CASE` BAdI definition if you want, for example, to maintain the selected clarification case with an own implementation, or change standard parameters to affect the processing of the clarification case (for example, not to permit the agent to switch between display and change mode).

The `BADI_EMMA_CASE` BAdI definition includes the `IF_BADI_EMMA_CASE` ABAP OO interface with the following method:

Method	Description
<code>TRANSACTION_START</code>	The default implementation processes the <code>EMMA_CASE_TRANSACTION_START</code> function module.

Figure 5-13 Method of the `IF_BADI_EMMA_CASE` ABAP OO Interface



Your implementation replaces the standard implementation of the respective interface method of the `CL_EMMA_CASE` ABAP OO class. You may use the implementation of this interface method as a basis for your own implementation.

5.8.2.3 BAdI Definition BADI_EMMA_CASE_TRANSACTION

You implement the `BADI_EMMA_CASE_TRANSACTION` BAdI definition to enhance the maintenance of clarification cases.

The BAdI definition includes a menu enhancement and an ABAP OO interface. With the menu enhancement, you add installation-specific buttons to the application toolbar of the clarification case maintenance transactions. You can also control the visibility of the standard buttons and related functions of the application toolbar as well as the visibility of the buttons and related functions in the toolbars of the various ALV lists. In addition, you can change the descriptive text of the clarification case. The `BADI_EMMA_CASE_TRANSACTION` BAdI definition includes the `IF_BADI_EMMA_CASE_TRANSACTION` ABAP OO interface with the following methods.

Enhancement Functions

Method	Description
<code>CALL_ENHANCEMENT_FUNCTION</code>	The system processes this method if the agent pushes one of the buttons you implemented with the menu enhancement. The function codes are <code>+EF1</code> , <code>+EF2</code> and <code>+EF3</code> . The active function code is transferred to the method via the <code>IV_UCOMM</code> parameter.

Figure 5-14 Methods of the `IF_BADI_EMMA_CASE_TRANSACTION` ABAP OO Interface related to Enhancement Functions

Handling of Function Codes

Method	Description
EXCLUDE_FCODES	The system processes this method once when the agent enters the clarification case maintenance. You may deactivate any (including the installation-specific) function of the GUI in adding the respective function code to the <code>ET_FCODE</code> table. Refer to the <i>GUI Status 0010 Case Transaction</i> of the <code>EMMA_CASE_TRANSACTION</code> function group for the available function codes.
EXCLUDE_ALV_BUTTONS	The system processes this method once when the agent enters the clarification case maintenance. You may deactivate any function of the ALV toolbars in adding the respective function code to either screen-specific table <code>CT_EXCLOBJ</code> (<i>Objects</i> screen), <code>CT_EXCLMSG</code> (<i>Messages</i> screen) and <code>CT_EXCLPROC</code> (<i>Processes</i> screen). Refer for the relevant function codes to the attributes section of the <code>CL_GUI_ALV_GRID</code> class and the constant attributes with the prefix <code>MC_FC_</code> , for example, the function code <code>CL_GUI_ALV_GRID=>MC_FC_FIND</code> representing the search function available via the  symbol in the ALV toolbar.

Figure 5-15 Methods of the `IF_BADI_EMMA_CASE_TRANSACTION` ABAP OO Interface related to Function Codes

Display of Objects, Messages and Solution Processes

Method	Description
ADJUST_ALVOBJ_OUTPUT	The system processes this method when the agent displays the <i>Objects</i> screen or returns to this screen after processing a screen function. The <code>IT_OBJ</code> importing table contains the sequence number and the description of all available objects of the clarification case container. You can change the display order of the objects on the screen as well as the descriptive texts per object in rearranging and changing the entries of the <code>CT_OBJ_CHANGE</code> table or in even removing entries from this table to hide objects from the agent.
ADJUST_ALVMSG_OUTPUT	The system processes this method when the agent displays to the <i>Messages</i> screen or returns to this screen after processing a screen function. The <code>IT_MSG</code> importing table contains the sequence number of all messages related to the clarification case. You can change the display order of the messages on the screen in rearranging the entries of the <code>CT_MSG_CHANGE</code> table or in even removing entries from this table to hide messages from the agent.

ADJUST_ALVPROC_OUTPUT	The system processes this method when the agent displays the <i>Processes</i> screen or returns to this screen after processing a screen function. The <code>IT_PROC</code> importing table contains the sequence number of all available solution processes. You can change the display order of the solution processes on the screen as well as the symbol and description per solution process in rearranging the entries of the <code>CT_PROC_CHANGE</code> table or in even removing entries from this table to hide solution processes from the agent.
------------------------------	--

Figure 5-16 Methods of the `IF_BADI_EMMA_CASE_TRANSACTION` ABAP OO Interface related to the Display of Objects, Messages and Solution Processes



An implementation of the above mentioned BAdI definition does not affect the available (additional) layout features of the ABAP list viewer (ALV), such as the sort order and filter settings of the list output on the respective object, message and solution process screens.

Clarification Case Description

Method	Description
CHANGE_DESCRIPTION	The system processes this method if the agent pushes the  Description button. You may amend the clarification case description as defined in the clarification case category or even compose a new descriptive text including variables as references to business objects the system replaces by its values in a subsequent step prior to displaying the text.
CHANGE_DESCRIPTION_OUTPUT	The system processes this method just before the system displays the clarification case description and after entered text variables have been replaced. You may finalize the text to be displayed, for example, in changing the formatting of the text, or even compose a new descriptive text.

Figure 5-17 Methods of the `IF_BADI_EMMA_CASE_TRANSACTION` ABAP OO Interface related to the Clarification Case Descriptions

5.8.3 Classic BAdI EMMA_CASE

The EMMA_CASE BAdI definition includes the IF_EX_EMMA_CASE ABAP OO interface. Most of the methods of this interface support mainly the enhancement of the clarification case maintenance dialog except of the COMPLETE_CASE, DETERMINE_CASE and UPDATE_ACTIONLOG_SOLPTRACK methods that serve different purposes. The following chapters describe the available methods in details while chapter 6 Use Cases provides some implementation examples you may use as a basis for an own implementation.

5.8.3.1 Installation-specific Subscreen for Clarification Case Maintenance

You can enhance the clarification case maintenance transaction with installation-specific data. For this, a subscreen is available on the *Additional Data* screen (see figure below).

First, you must create your own subscreen and implement the flow logic of the screen in an installation-specific report, module pool or function group because it is not possible to create a screen in an ABAP OO class implementation. You may structure the layout of the subscreen such that it suits best the user requirements. However, the size should not exceed 20 lines and 90 rows.

Before creating the subscreen, you commonly enhance clarification cases with installation-specific data in adding new fields to the CI_EMMA_CASE include structure that is included in the EMMA_CASE clarification case header table (for more details about the EMMA_CASE table, see chapter 5.1.4 Table EMMA_CASE – Clarification Case Header Table).

 You may add new fields to the CI_EMMA_CASE include structure at any time during the implementation. However, note that the activation of your enhancement may take a long time if there are already a large number of clarification cases in your system.

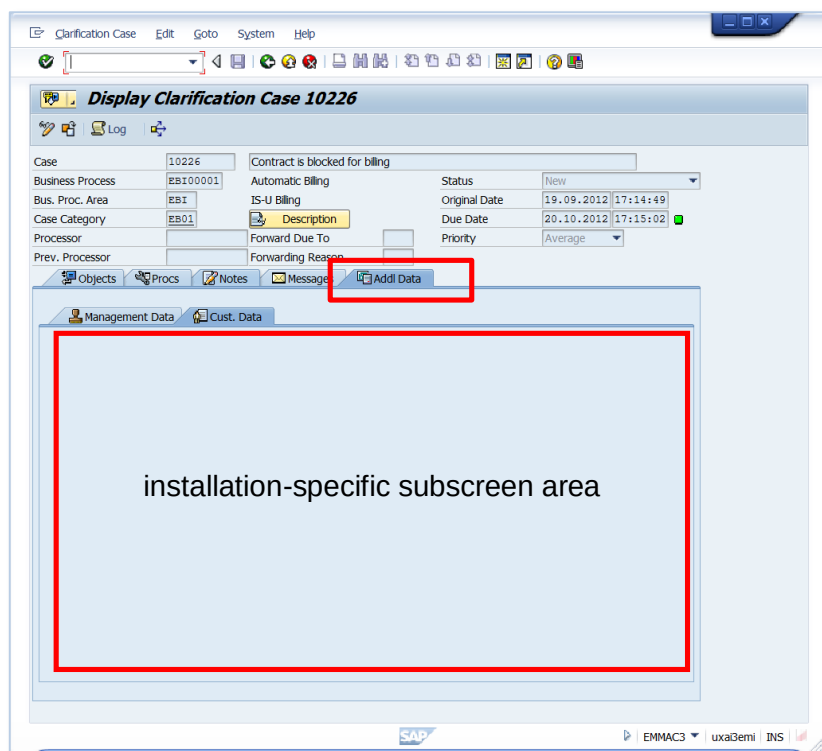


Figure 5-18 Space for installation-specific Subscreen

The `EMMA_CASE` BAdI definition provides the following methods to accommodate the transfer of data between the subscreen and your program. Mind that there is no automatic data transfer to and from your installation-specific screen if you did not implement these methods properly.

Method	Description
DETERMINE_CUSTOMER_SUBSCREEN	The system processes this method whenever the agent switches to <i>Customer Data</i> subscreen during the clarification case maintenance. You must return the screen number and the program name of your installation-specific screen. You can use the transferred clarification case data to determine the subscreen you want to display. You may use, for example, a specific installation-specific screen in dependency of the clarification case category, or even switch between subscreens during the maintenance of the same clarification case.
TRANSFER_DATA_TO_SUBSCREEN	With this method, you transfer data from your program to the fields of your installation-specific screen based on the transferred clarification case-specific data.
TRANSFER_DATA_FROM_SUBSCREEN	The clarification case maintenance transaction transfers the content of the fields of the <code>CI_EMMA_CASE</code> include structure to your program for further processing.

Figure 5-19 Methods of the `EMMA_CASE` BAdI for Installation Specific Subscreen

5.8.3.2 Handle Function Codes during Clarification Case Maintenance

You can control the execution of any function triggered by the agent during the dialog processing of a clarification case. If the agent triggers a function, for example, in pushing a button on the screen or selecting a menu entry, the system processes the `PROCESS_CUSTSUB_OKCODE` method prior to initiating any action as defined for the function code. This includes the function codes of the buttons you might have added to the toolbar with an implementation of the `BADI_EMMA_CASE_TRANSACTION` menu enhancement.

If you do not want the system to process the triggered function, return an 'X' in the `EV_CANCEL` parameter of the `PROCESS_CUSTSUB_OKCODE` method. Alternatively (or in addition due to downward compatibility reasons), you may provide a message in the `ES_RETURN` parameter for the system to display. If the method returns a message, the system displays this message and cancels further processing of the function code.

Method	Description
PROCESS_CUSTSUB_OKCODE	The system processes this method for any function that is triggered by the agent. The related function code is transferred to the method by the <code>IV_OKCODE</code> parameter. If the agent selected one of the solution processes to execute, the system transfers additional information concerning the selected solution process via the <code>IS_SOLP</code> parameter. Return <code>EV_CANCEL = 'X'</code> and/or add a message to the <code>ES_RETURN</code> parameter, if you do not want the function to be processed.

Figure 5-20 Method of the `EMMA_CASE` BAdI to control OKCODE

5.8.3.3 Installation-specific Validations during Clarification Case Maintenance

You can enhance the clarification case maintenance transaction with installation-specific checks. The `EMMA_CASE` BAdI definition provides the following methods, for example, to control if an agent may open a clarification case and, in addition, a data validation method.

Method	Description
CHECK_BEFORE_DISPLAY	The system processes this method before the selected clarification case is displayed in display mode. If you raise the <code>DISPLAY_NOT_PERMITTED</code> exception, the system displays the message you specified in the <code>ES_RETURN</code> parameter and stops further processing of the clarification case.
CHECK_BEFORE_CHANGE	The system processes this method before the selected clarification case is displayed in change mode. If you raise the <code>CHANGE_NOT_PERMITTED</code> exception, the system displays the message you specified in the <code>ES_RETURN</code> parameter and stops further processing of the clarification case.
CHECK_BEFORE_SAVE	The system processes this method prior saving a changed clarification. If you return a message in the <code>ES_RETURN</code> parameter, the system displays the message and returns to the clarification case maintenance screen without persisting the changed data on the database. In addition, you may update the installation-specific fields you may have added to the <code>CI_EMMA_CASE</code> include structure.

Figure 5-21 Further Methods of the `EMMA_CASE` BAdI to control User Actions

 The `CHECK_BEFORE_SAVE` method is not only processed when saving a clarification case during the manual clarification case maintenance but also when a report changes the processor of a clarification case by means of a BAPI or changes the status of a clarification case.

 You can use the `CHECK_BEFORE_SAVE` method to verify if a status change of the clarification case, for example to *Completed*, is valid or the root of the case has not been resolved yet indeed. In the latter case, you may return a message in the `ES_RETURN` parameter to make the agent aware of the respective reasons.

 You can statically define the default start screen when entering the clarification case maintenance by means of the `EMMA_CASE_START` SPA/GPA parameter (for more details, see chapter 5.9 SPA/GPA Parameter). However, you may want dynamically overwrite the default screen setting and define the *Notes* screen as the start screen when entering the clarification case maintenance in display mode if a clarification case is associated with a particular clarification case category. You may use the following code snippet as a basis for your own implementation of the `CHECK_BEFORE_DISPLAY` method. You have to implement the same code in the `CHECK_BEFORE_CHANGE` method if you want to overwrite the default screen also when entering the clarification case maintenance in change mode.

```

METHOD if_ex_emma_case~check_before_display.

* determine start screen
IF is_case-ccat = 'BI01'.
    " User should start with the Messages start screen
    SET PARAMETER ID 'EMMA_START_SCREEN' FIELD 'NOTES'.
ENDCASE.

ENDMETHOD.

```

5.8.3.4 Amend and complete Clarification Case with installation-specific Data

You can amend a clarification case with installation-specific data (EV_CUSTFIELDS parameter) prior to saving a clarification case on the database. In addition, you may add additional business objects to the clarification case (CT_OBJECTS parameter) and adjust the priority (EV_PRIO parameter), the authorization group (EV_BRGRU parameter), the due date (EV_DUEDATE and EV_DUETIME parameters) and the forwarding method (EV_FWM parameter) of the clarification case. Otherwise, the system determines the default values of these parameters based on the underlying clarification case category.

The EMMA_CASE BAdI definition provides the COMPLETE_CASE method that is only processed when creating a clarification case.

Method	Description
COMPLETE_CASE	The system processes this method at the end of the clarification case creation process. If you raise the NO_CASE exception the clarification case is not created.

Figure 5-22 COMPLETE_CASE Method of the EMMA_CASE BAdI



During the processing of the COMPLETE_CASE method, you may take advantage of the importing IV_CASE_WITH_PREV_CCAT_CREATED parameter. This parameter indicates if for the same message group a clarification case was already created based on another clarification case category. You can check this parameter in the COMPLETE_CASE method and raise the NO_CASE exception if you want to prevent the creation of a further clarification case due to the same business exception.

5.8.3.5 Installation-specific Determination of Clarification Case Category

You implement the DETERMINE_CASE method of the EMMA_CASE BAdI definition if you want to perform additional ABAP based checks whether a clarification case category should be used for the automatic creating of a clarification case besides the system check for matching triggering messages.

Method	Description
DETERMINE_CASE	The system processes this method only if the <i>Business Add-In</i> flag is marked for the clarification case category and, as already explained in chapter 5.3.1 <i>Automatic Creation of Clarification Cases</i> , the triggering messages of this clarification case category match the messages of the business exception. If you raise the NO_CASE exception, the system stops the processing of the currently processed clarification case category.

Figure 5-23 DETERMINE_CASE Method of the EMMA_CASE BAdI



With the `DETERMINE_CASE` method, you may also amend the triggering messages of the new clarification case by additional messages in replacing or adding messages to the `ET_MSG_LINK` parameter of the `DETERMINE_CASE` method. For an example, see chapter 6.3.3 *General Sweeper Clarification Case Categories*.

5.8.3.6 Installation-specific Check for identical Clarification Cases

You implement this check if you want to amend the existing standard checks for identical clarification cases. The `EMMA_CASE` BAdI definition provides the `CHECK_IDENTICAL_CASE` method that is processed only when a clarification case is to be created.

Method	Description
CHECK_IDENTICAL_CASE	The system processes this method only if the <i>Check for Existence of Identical Clarification Case</i> flag is marked for the clarification case category (see chapter 3.4.3.2 <i>Screen Messages</i>). The <code>IT_MSG_LINK</code> parameter holds the actual list of triggering messages the system is currently analyzing for a possible clarification case creation. The <code>IT_CASE_MSG_LINK</code> parameter holds a list of the triggering messages of all still open clarification cases with matching clarification case category, main object type and object key as retrieved from the database, whereas the content of message variables was not part of the query. The <code>CV_NEW_CASE</code> changing parameter holds the result of the previously performed standard check for identical clarification cases (for further details, see chapter 5.3.1 <i>Automatic Creation of Clarification Cases</i>) basically checking if one of the message lists of the <code>IT_CASE_MSG_LINK</code> parameter matches the message list of the <code>IT_MSG_LINK</code> parameter. You may overwrite the value of the <code>CV_NEW_CASE</code> parameter to reflect the result of your check of identical clarification cases.

Figure 5-24 `CHECK_IDENTICAL_CASE` Method of the `EMMA_CASE` BAdI



You can use the `CHECK_IDENTICAL_CASE` method to check for identical clarification cases in excluding a message variable. In the example below the content of message variable 4 of the `EMMA 195` message is blanked for the subsequent check.

```
METHOD if_ex_emma_case~check_identical_case.

* Local data definition
DATA: lt_msg_link      TYPE emma_msg_link_tab,
      lt_case_msg_link TYPE emma_case_msg_link_tab.

* exclude message variable 4 of EMMA(195) message from the comparison
CASE is_ccat-ccat.
  WHEN 'JL01'.

*   copy actual messages
  lt_msg_link = it_msg_link.
  lt_case_msg_link = it_case_msg_link.

*   clear message variable 4 in messages of new clarification case
  LOOP AT lt_msg_link ASSIGNING FIELD-SYMBOL(<fs_msg_link>)
```

```

                                WHERE msgid = 'EMMA'
                                AND msgno = '195'.

    CLEAR <fs_msg_link>-msgv4.
    ENDLOOP.

*   check all identified messages
    LOOP AT lt_case_msg_link
                                ASSIGNING FIELD-SYMBOL(<fs_case_msg_link>).
*   clear message variable 4 in existing and still open cases
    LOOP AT <fs_case_msg_link>-msg ASSIGNING FIELD-SYMBOL(<fs_msg>)
                                WHERE msgid = 'EMMA'
                                AND msgno = '195'.

    CLEAR <fs_msg>-msgv4.
    ENDLOOP.
    IF lt_msg_link[] = <fs_case_msg_link>-msg[].
*   one case has matching messages hence there is a duplicate case
        cv_new_case = cl_emma_case=>co_false.
        EXIT.
    ENDIF.
    ENDLOOP.
    WHEN OTHERS.
    ENDCASE.

ENDMETHOD.

```

5.8.3.7 Update Application Log during Clarification Case Maintenance

You can add additional messages to the application log of the clarification case during the processing of automatic solution processes of a clarification case.

The `EMMA_CASE` BAdI definition provides the `UPDATE_CASE_ACTION_LOG` method to specify the messages you want to add to the application log in addition to the system messages.

Method	Description
UPDATE_CASE_ACTION_LOG	The system processes this method after the automatic execution of an automatic solution process of a clarification case. With this method, you specify messages in the <code>ET_MSGS</code> parameter the system shall add to the application log of the clarification case in addition to the solution process related system messages.

Figure 5-25 `UPDATE_CASE_ACTION_LOG` Method of the `EMMA_CASE` BAdI

5.8.4 ABAP OO Interfaces

5.8.4.1 ABAP OO Interface IF_EMMA_CASE_FORWARD

You can enhance the process of automatically determining and changing the current processor of a clarification case after creation or change of a clarification case. SAP provides the `IF_EMMA_CASE_FORWARD` ABAP OO interface and its `FORWARD` method for an installation-specific implementation.

There is a Customizing activity available to allocate a name to an implemented forwarding method. You can implement various installation-specific ABAP OO classes and register them as available forwarding methods when defining a clarification case categories (for more details, see chapters 3.4.3.1 *Screen Basic Data* and 3.4.4 *Define Method for Forwarding Clarification Cases*). During runtime, the system transfers the currently processed clarification case to the `FORWARD` method by means of an object reference.

Method	Description
FORWARD	The system processes this method after a clarification case was created or changed. Commonly, you use this method, for example, to allocate a specific organization or user to a clarification case or to inform users of a clarification case by a workflow item or by email.

Figure 5-26 `FORWARD` Method of the `IF_EMMA_CASE_FORWARD` ABAP OO Interface

For use cases and sample implementation of this method, see chapter 6.4 *Agent Determination*.

5.8.4.2 ABAP OO Interface IF_EMMA_CWL_LAYOUT

The `EMMACLS` transaction offers shortcut keys for a selection of clarification cases based on different selection criteria by the push of a button. As described in chapter 3.5.4 *Maintain Shortcut Keys for Clarification Case Processing*, you can enhance the selection of clarification cases with more sophisticated or dynamic selection criteria that are difficult to implement in just a static report variant. Alternatively, you can implement the selection of clarification cases and just return a list of clarification cases for a further processing by the agent.

SAP provides the `IF_EMMA_CWL_LAYOUT` ABAP OO interface with the two methods to enhance the dynamic selection of clarification cases. The system transfers the name of the active layout and the shortcut key number that was pushed by the agent:

Method	Description
DETERMINE_SELCRITERIA	The system processes this method to determine dynamically the selection criteria to be used for a selection of clarification cases instead of an existing variant of the <code>REMMACASELIST</code> report. You return the selection criteria in the <code>ET_SELCRIT</code> parameter as basic range table of type <code>RSTI_T_PAR</code> that includes not only the range information but also the related parameter name of the selection screen of the <code>REMMACASELIST</code> report. Technically, the system submits the <code>REMMACASELIST</code> report via the ABAP statement <code>SUBMIT REMMACASELIST WITH SELECTION-TABLE ET_SELCRIT.</code>

SELECT_CASES_FROM_DB	The system processes this method to determine the relevant clarification cases by own database accesses. The selected clarification cases are returned for further processing in the <code>ET_CASES</code> parameter of type <code>REMA_CASE_TAB</code> that represents rows of the <code>EMMA_CASE</code> table (clarification case header table). You can transfer a threshold value in the <code>EV_AUTO_LIST_REFRESH</code> return parameter to control the automatic refresh of the displayed clarification case list (for more details, see chapter 4.3.1 <i>Transactions EMMACL/EMMACLS – Clarification Case List</i>)
-----------------------------	--

Figure 5-27 Methods of the `IF_EMMA_CWL_LAYOUT` ABAP OO Interface

 SAP delivers the `CL_EMMA_CWLSHL_EXAMPLE` ABAP OO class as a sample implementation. You may use this implementation as a basis for your own implementation.

5.8.4.3 Interfaces of *Classic EMMA*

Some few ABAP OO interfaces belong to the *Classic EMMA* implementation and should not be used anymore for a new installation-specific implementation. SAP recommends using the `CL_EMMA_JOB_ANALYSIS` class as an ABAP OO superclass instead. For more implementation details, see chapter 5.2.4 *Technical Implementation Job Preparation New EMMA*.

The following ABAP OO interfaces have been implemented as part of the *Classic EMMA*:

Interface	Description
IF_EMMA_JOB_ANALYZE	Determine processing status of BPEM job and read messages of process
IF_EMMA_INTINFO_GET	Get interval information of process if the process was executed as a mass activity.
IF_EMMA_LOGNUMBER_GET	Retrieve numbers of the application logs of the process
IF_EMMA_LOGMSG_PROCESS	Analyze and validate messages and assignment of messages to business objects

Figure 5-28 Obsolete *Classic EMMA* ABAP OO Interfaces



SAP delivered previously classes using the *Classic EMMA* interfaces. In the meantime, most of the interfaces and classes for the delivered business processes have been migrated to support the *New EMMA* implementation.

5.8.4.4 Interface `IF_EMMA_KPI_*`

SAP delivers BW extractors and related Business Add-Ins (BADIs) for the following selected business process areas (BPA) of the SAP for Utilities industry solution. All these delivered ABAP OO classes have in common that they include the `IF_EMMA_KPI_BW` ABAP OO interface. In addition, most classes have an additional interface included whose prefix `IF_EX_` leads to the related available Business Add-In (BAI). You can implement the BAI to amend the data to be transferred to BW. The list below shows the names of the available BADIs.

BPA	Description	Class	Business Add-In
EMR	Meter Reading	CL_MR_KPI_BW	ISU_MR_KPI_PROCESS
EBI	Billing	CL_BI_KPI_BW	ISU_BI_KPI_PROCESS
EIN	Invoicing	CL_IN_KPI_BW	ISU_INV_KPI_PROCESS
ECS	Customer Service	CL_CS_KPI_BW	- / -
EBB	Budget billing plan	CL_IN_KPI_BW	ISU_INV_KPI_PROCESS
EEDM	Energy Data Management	CL_EDM_KPI_BW	ISU_EDM_KPI_PROCESS
EEWA	Waste Management	CL_WASTE_ORDER_KPI_BW	ISU_WA_ORDER_KPI_PROCESS

Figure 5-29 BW specific ABAP OO Interfaces



The ISU_EDM_KPI_PROCESS and ISU_WA_ORDER_KPI_PROCESS BAdIs are not implemented as classic BAdIs but as enhancement spots with the given name.

For more information about the extraction to BW, see chapter 7 *Business Information Warehouse*.

5.9 SPA/GPA Parameter

Input fields on screens can be associated with SPA/GPA parameters hence be populated with pre-assigned values when the screen is called (for more details, see chapter 2.1.16 *SPA/GPA Parameters*). You cannot only pre-assign values to screens but also control BPEM related reports and transactions.

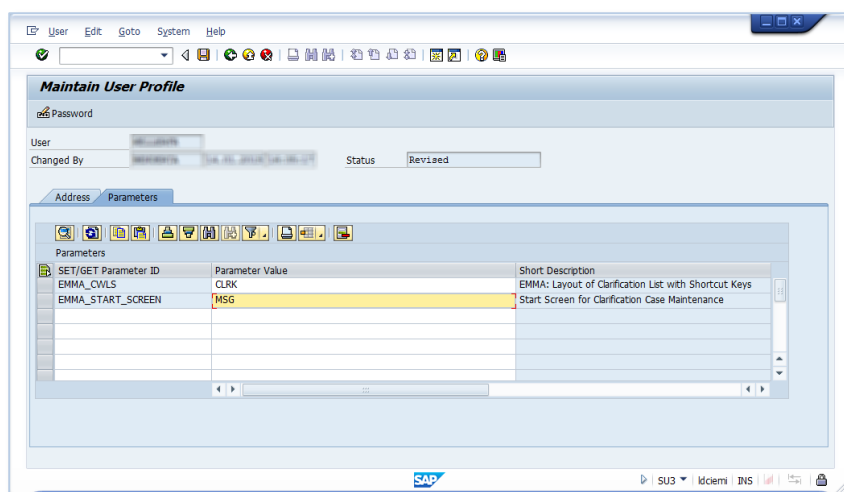


Figure 5-30 Editing parameter of a user profile

You define SPA/GPA parameters via the menu *Tools* → *Administration*, then *User maintenance* → *Users* and enter the chosen parameter in the *Parameter* column. Alternatively, you edit maintain your own user data by choosing *System* → *User profile* → *Own data* or calling the SU52 transaction (see figure above).

BPEM uses the following SPA/GPA parameters:

Parameter	Description														
EMMA_TRX	Layout of BPEM job lists and clarification case lists <table border="1"> <thead> <tr> <th>Output</th><th>Parameter Value</th></tr> </thead> <tbody> <tr> <td>Display as classic list</td><td>X</td></tr> <tr> <td>Display with ABAP list viewer ALV</td><td>space</td></tr> </tbody> </table>	Output	Parameter Value	Display as classic list	X	Display with ABAP list viewer ALV	space								
Output	Parameter Value														
Display as classic list	X														
Display with ABAP list viewer ALV	space														
EMMA_CCAT_TREE	Display selection tree of clarification case categories showing business process areas and business process codes: <table border="1"> <thead> <tr> <th>Output</th><th>Parameter Value</th></tr> </thead> <tbody> <tr> <td>Full selection tree</td><td>X</td></tr> <tr> <td>Only already defined categories</td><td>space</td></tr> </tbody> </table>	Output	Parameter Value	Full selection tree	X	Only already defined categories	space								
Output	Parameter Value														
Full selection tree	X														
Only already defined categories	space														
EMMA_CWLS	Predefined layout for clarification list with shortcut keys (see <code>EMMACLS</code> transaction)														
EMMA_CWLS_EXTSEL	Display radio buttons for a selection (see selection screen of <code>EMMACL</code> transaction)														
EMMA_JOB_MAX_SEL	Predefined maximum number of selected jobs (see selection screen of <code>EMMACL</code> transaction)														
EMMA_RUNID	Last used BPEM job number														
EMMA_BPAREA	Last used business process area														
EMMA_BPCODE	Last used business process code														
EMMA_CCAT	Last used clarification case category														
EMMA_CNR	Last used clarification case number														
MAG	Message class (see selection screen of <code>EMMAC1M</code> transaction)														
EMMA_MSGNO	Message number (see selection screen of <code>EMMAC1M</code> transaction)														
EMMA_START_SCREEN	Start screen of the clarification case maintenance transaction: <table border="1"> <thead> <tr> <th>Start Screen</th><th>Parameter Value</th></tr> </thead> <tbody> <tr> <td>Object</td><td>OBJ</td></tr> <tr> <td>Solution Processes</td><td>PROC</td></tr> <tr> <td>Notes</td><td>NOTES</td></tr> <tr> <td>Messages</td><td>MSG</td></tr> <tr> <td>Management Data</td><td>ADM</td></tr> <tr> <td>Customer Data</td><td>CUST</td></tr> </tbody> </table>	Start Screen	Parameter Value	Object	OBJ	Solution Processes	PROC	Notes	NOTES	Messages	MSG	Management Data	ADM	Customer Data	CUST
Start Screen	Parameter Value														
Object	OBJ														
Solution Processes	PROC														
Notes	NOTES														
Messages	MSG														
Management Data	ADM														
Customer Data	CUST														
EMMA_APPL_LOG_ISU_MA	Use BPEM to display application log of IS-U mass activities														
EMMA	<i>Obsolete</i>														

Figure 5-31 SPA/GPA Parameter

 You want to define the *Messages* screen instead of the default *Objects* screen as the start screen when entering the clarification case maintenance (for more details, see chapter 4.3.2 *Transaction EMMAC2 - Clarification Case Maintenance*). For this, enter the `EMMA_START_SCREEN` parameter ID and complete the definition in entering the `MSG` parameter value.

5.10 Lock Objects

The system uses lock objects to synchronize access to the same data by more than one program. You can synchronize access by several programs to the same data with a logical lock mechanism. This lock mechanism fulfills two main functions:

- A program can tell other programs which data records it is just reading or changing.
- A program can prevent itself from reading data that is just being changed by another program.

The data records of a table to be locked are defined by a logical condition. When a lock is set, this logical condition is entered in a lock table. This entry is retained until it is removed by the program or the program comes to an end. All the locks set by a program are thus removed at the end of the program.

When accessing data records, the records just being edited by other programs can be identified by the entry in the lock table. Such an entry for the lock must define a number of fully specified key fields that is either a value is passed for the key field or this field is locked generically. You can monitor the current locks via the `SM12` transaction.

BP EM uses the following lock objects to synchronize access to clarification cases and BP EM jobs:

Lock Object	Description
EMMA_CASE	Lock Object for Clarification Cases
EMMA_INT	Lock Object for BP EM jobs
EMMA_HDR	<i>Lock Object for BP EM job (obsolete)</i>

Figure 5-32 Lock Objects

6 Use Cases and Sample Implementations

This chapter provides use cases, implementation examples and step-by step instructions how you can enhance the BPEM application according to your business requirements. Of course, you may use the provided code snippets as a basis for your own implementation.

6.1 Installation-Specific Fields

You want to add installation-specific clarification case attributes to a clarification case and display these attributes during the clarification case maintenance.

Step 1: Create the `CI_EMMA_CASE` include structure via the `SE11` transaction, add the required fields and activate the newly created structure.

Step 2: Create an own function group or module pool to implement your subscreen to be displayed on the clarification case maintenance screen.

```
*&-----*
*
*& Modulpool          ZBPEM_SCREEN
*&-----*
PROGRAM    zbpem_screen.

TABLES:   emma_case.
DATA:     gv_workmode TYPE emma_ctxn_wmode.
```

Step 3: Create an own subscreen and implement the screen flow logic. The screen size of your subscreen may have up to 20 lines and 90 columns.

```
PROCESS BEFORE OUTPUT.

    MODULE modify_0100.

PROCESS AFTER INPUT.
```

Step 4: Implement the required modules of the screen flow logic but do not handle function codes in a PAI module of your subscreen. Instead, implement the `PROCESS_CUSTSUB_OKCODE` method of the `EMMA_CASE` BAdI.

```
*&-----*
*&      Module  MODIFY_0100  OUTPUT
*&-----*
MODULE modify_0100 OUTPUT.

    LOOP AT SCREEN.

        CASE gv_workmode.
            WHEN cl_emma_case_txn=>co_wmode_display.
                screen-input    = '0'.
                MODIFY SCREEN.
            WHEN OTHERS.
                screen-input    = '1'.
                MODIFY SCREEN.
            ENDCASE.

        ENDLOOP.

    ENDMODULE.                " MODIFY_0100  OUTPUT
```

Step 5: Implement the required subroutine to handle the data flow logic from the transaction to your subscreen.

```
*&-----*
*&      Form  TRANSFER_DATA_TO_CUSTSUB
*&-----*
FORM transfer_data_to_custsub USING pv_emma_case TYPE emma_case
                                pt_objects TYPE emma_cobj_t
                                pt_msg_link TYPE emma_msg_link_tab
                                pt_actors TYPE tswfactor
                                pt_solpath TYPE emma_csop_t
                                pv_workmode TYPE emma_ctxn_wmode.

* transfer case data to variables of subscreen
  emma_case = pv_emma_case.
  gv_workmode = pv_workmode.

ENDFORM.                "TRANSFER_DATA_TO_CUSTSUB
```

Step 6: Implement the required subroutine to handle the data flow logic from your subscreen to the transaction.

```
*&-----*
*&      Form  TRANSFER_DATA_FROM_CUSTSUB
*&-----*
FORM transfer_data_from_custsub USING pv_custfields TYPE emma_cci.

* transfer case data back from screen to clarification case transaction
  move-corresponding emma_case to pv_custfields

ENDFORM.                "TRANSFER_DATA_FROM_CUSTSUB
```

Step 7: Integrate your subscreen in the clarification case transaction. To do this, you must implement via transaction SE19 several methods of the EMMA_CASE classic BAdI. The figure below shows a sample BAdI implementation and available methods (for a detailed description of the method, see chapter 5.8.3 *Classic BAdI EMMA_CASE*).

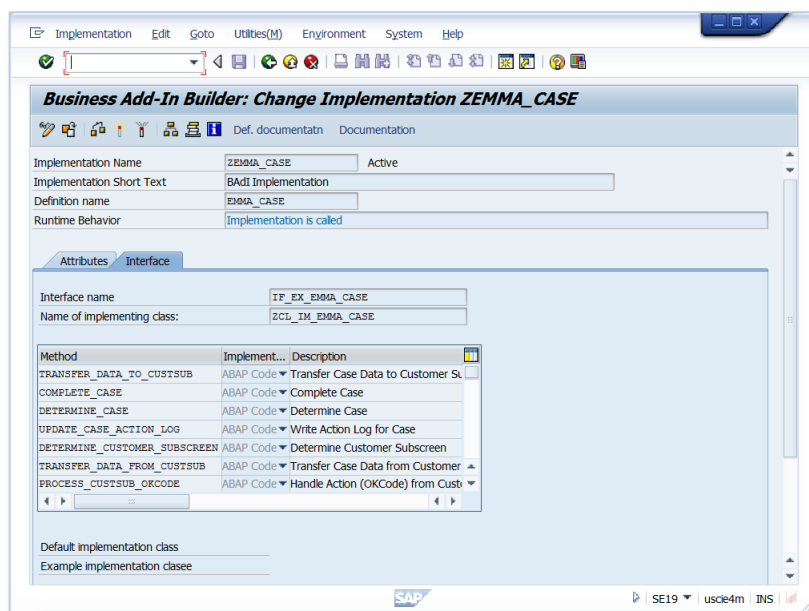


Figure 6-1 Example of a EMMA_CASE classic BAdI Implementation

You must implement at least the following methods to complete the integration of your subscreen in the clarification case maintenance transaction:

- **DETERMINE_CUSTOMER_SUBSCREEN:** Publish the program name and screen number of your subscreen. If required, you may even switch between your screens since the system calls this method always prior to displaying the installation-specific screen.

```
METHOD if_ex_emma_case~determine_customer_subscreen.

CASE is_case-ccat.
  WHEN 'C&I'.
    ev_repid = 'ZBPEM_SCREEN'.
    ev_dynnr = '0200'.
  WHEN OTHERS.
    ev_repid = 'ZBPEM_SCREEN'.
    ev_dynnr = '0100'.
ENDCASE.

ENDMETHOD.
```

- **TRANSFER_DATA_TO_CUSTSUB:** Transfer case data from the clarification case transaction to your subscreen.

```
METHOD if_ex_emma_case~transfer_data_to_custsub.

PERFORM transfer_data_to_custsub IN PROGRAM zbpem_screen IF FOUND
  USING is_case
        it_objects it_msg_link it_actors it_solpath
        iv_work_mode.

ENDMETHOD.
```

- **TRANSFER_DATA_FROM_CUSTSUB:** Transfer case data from your subscreen to the clarification case.

```
METHOD if_ex_emma_case~transfer_data_from_custsub.

PERFORM transfer_data_from_custsub IN PROGRAM zbpem_screen IF FOUND
  USING ev_custfields.

ENDMETHOD.
```

- **PROCESS_CUSTSUB_OKCODE:** You must implement this method only if an own processing is required for a function that appears on your subscreen.

6.2 Installation-Specific Functions

You can enhance the processing of clarification cases in controlling the execution of the agent-triggered functions and in adding additional buttons to the application toolbar. The following chapters describe the implementation of these options.

6.2.1 Enhancement of the Application Toolbar

You can add up to three additional installation-specific buttons in the application toolbar of the clarification case maintenance transaction in implementing the menu enhancement of the `BADI_EMMA_CASE_TRANSACTION` BAdI via transaction `SE19`.

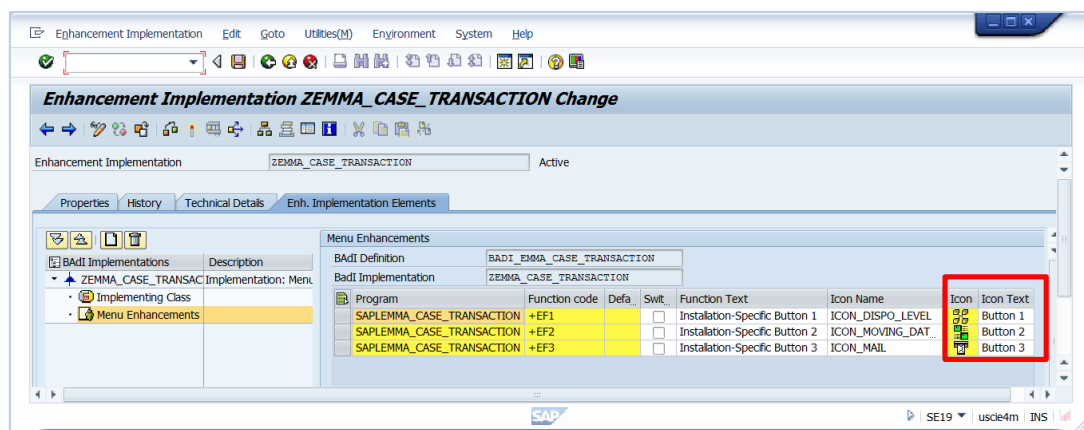


Figure 6-2 Example of a BADI_EMM_CASE_TRANSACTION BAdI Implementation

Figure 6-2 shows a sample implementation of the BAdI definition with three installation-specific buttons. Of course, you can use your own icons, functions and icon texts. The sample texts are only to showcase the implementation.

After activation, the same buttons are available in the application toolbar of the clarification case maintenance transaction as shown in the figure below.

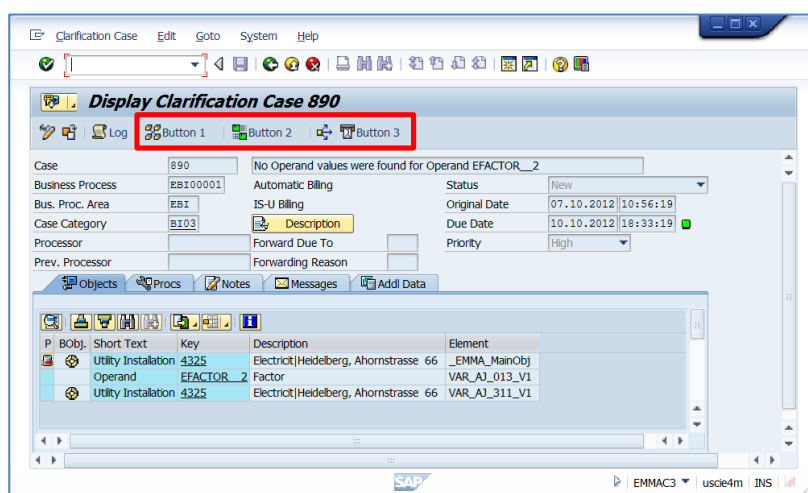


Figure 6-3 Clarification Case Transaction with Installation-Specific Buttons

At last, you need to implement the `CALL_ENHANCEMENT_FUNCTION` method of the `BADI_EMM_CASE_TRANSACTION` implementation to handle the processing of the related function codes once an agent pushed one of the implemented buttons.

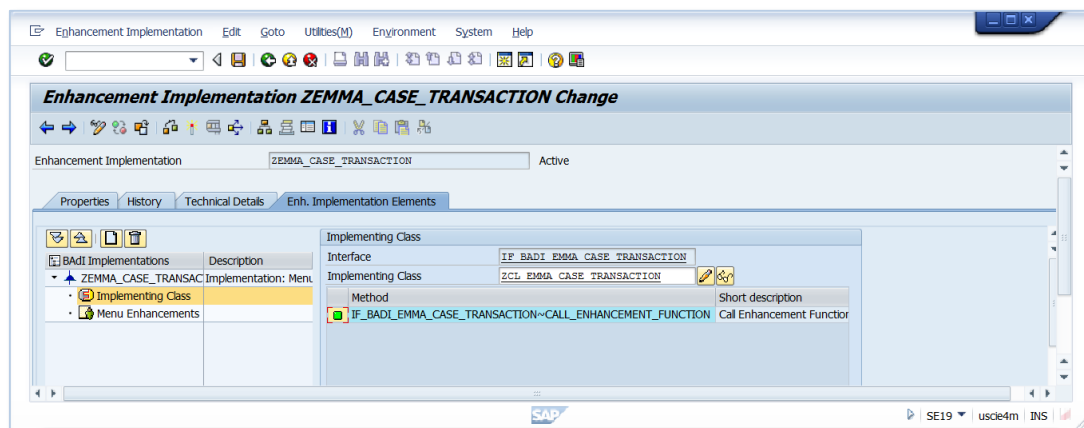


Figure 6-4 Method of a BADI_EMM_CASE_TRANSACTION Implementation

You may use the following code snippets as a basis for your own implementation of a `CALL_ENHANCEMENT_FUNCTION` method.

```
METHOD if_badi_emma_case_transaction~call_enhancement_function.

    DATA: lv_case TYPE emma_case.

    * get clarification case data
    CALL METHOD ir_case->get_data
        RECEIVING
            ev_case = lv_case.

    * process installation-specific function
    CASE iv_ucomm.
        WHEN '+EF1'.
            " call function specific to button 1

        WHEN '+EF1'.
            " call function specific to button 2

        WHEN '+EF1'.
            " call function specific to button 3

    ENDCASE.

ENDMETHOD.
```

6.2.2 Control of User-Triggered Functions

There is an additional option to implement installation specific processing of function codes in the clarification case maintenance transaction. For this, you must implement the `PROCESS_CUSTSUB_OKCODE` method of the `EMMA_CASE` Business Add-In as a classic BAdI implementation.



If you want to prevent an agent from closing a clarification case before a specific solution process was executed, return an appropriate message with the `ES_RETURN` parameter. The system displays this message and stops further processing until the next user action.

6.3 Error Message without Clarification Case

It is very important that all business exceptions lead to a clarification case for a later remediation. This is particularly true when you allow the system to create clarification cases automatically. You want to rely on the implementation and you do not want to check regularly the BPEM specific application logs for messages indicating that there have been unexpected so far unknown business exceptions for which the system did create a clarification case.

When starting to implement BPEM, one of the frequently asked question is: “What are the options if there is a business exception for which no matching clarification case category exists to be used as a template for the clarification case creation?” Even it seems virtually impossible to create clarification case categories for any possible upcoming business exception, BPEM offers three options how you can ensure that a business exception leads to a clarification case even there is no matching clarification case category available. The following three chapters explain the available options in details. SAP recommends implementing only the third option as explained in chapter 6.3.3 *General Sweeper Clarification Case Categories* due to its simplicity. However, you still may want to validate the implementation of the other two options for a better understanding of BPEM.

6.3.1 Analysis of application log of FPEMMAMA transaction

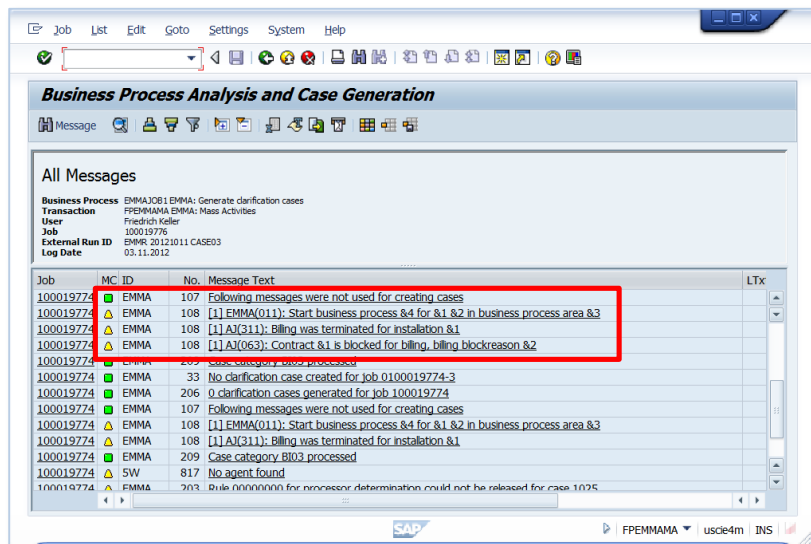
During the automatic creation of clarification cases, the FPEMMA transaction documents the messages in its application log that have not led to a created clarification cases. You can identify these messages in the application in looking for an EMMA 107 message *Following messages were not used for creating cases*. The subsequent EMMA 108 messages describe the messages that have not been used to create clarification cases.

The message text of an EMMA 108 message has the following structure:

[<occurrence>] <message class>(message number): <message text>

 The message *Contract x is blocked for billing, billing block reason y* with its message class AJ and message number 63 indicates a business exception. If this message of the underlying business process has not led to a clarification case, the following EMMA 108 message can be found in the application log of the FPEMMAMA transaction as [1] AJ(063): *Contract &1 is blocked for billing, billing blockreason &2*.

The figure below shows an excerpt of an application log of the FPEMMAMA mass activity with EMMA 107 and EMMA 108 message sequence.



Job	MC ID	No.	Message Text	LTx
100019774	EMMA	107	Following messages were not used for creating cases	
100019774	EMMA	108	[1] EMMA(011): Start business process &4 for &1 &2 in business process area &3	
100019774	EMMA	108	[1] AJ(311): Billing was terminated for installation &1	
100019774	EMMA	108	[1] AJ(063): Contract &1 is blocked for billing, billing blockreason &2	
100019774	EMMA	209	Case category B103 processed	
100019774	EMMA	33	No clarification case created for job 0100019774-3	
100019774	EMMA	206	0 clarification cases generated for job 100019774	
100019774	EMMA	107	Following messages were not used for creating cases	
100019774	EMMA	108	[1] EMMA(011): Start business process &4 for &1 &2 in business process area &3	
100019774	EMMA	108	[1] AJ(311): Billing was terminated for installation &1	
100019774	EMMA	209	Case category B103 processed	
100019774	5W	817	No agent found	
100019774	EMMA	203	Rule 00000000 for processor determination could not be released for case 1025	

Figure 6-5 Application log during automatic creation of clarification cases

You can pick up EMMA 108 messages in creating a specific clarification case category for the EMMAJOB1 business process code since this is the business process code of the *Generate Clarification Case* business process. Obviously, the clarification case category of the EMMAJOB1 business process code cannot be object specific because there is no business object allocated to the EMMA 108 message hence do not mark the *Category Business Object specific* field. For now, the solution is using the EMMA 108 message as the triggering message and the system creates one clarification case showing all EMMA 108 messages.

In addition, you may want to implement the `DETERMINE_CASE` method of the `EMMA_CASE` BAdI to remove unwanted or duplicated messages, for example, the `AJ 311` messages, if you have not suppressed it during the BPEM job preparation step (for more details how to suppress messages, see chapter 3.3.4.1 *Suppress Messages during Preparation*).

```

METHOD if_ex_emma_case~determine_case.

  DATA: ls_it_msg TYPE LINE OF emma_msg_tab,
        ls_et_msg_link TYPE LINE OF emma_msg_link_tab.

  CASE is_ccat-ccat.

  *   sweeper clarification case category
  WHEN 'MSG'.
    " eliminate possible duplicates messages
    SORT et_msg_link.
    DELETE ADJACENT DUPLICATES FROM et_msg_link COMPARING msgid msgno.

    " delete EMMA 011 message and further unwanted messages
    LOOP AT et_msg_link INTO ls_et_msg_link.
      IF ls_et_msg_link-msgid = 'AJ' AND
         ls_et_msg_link-msgno = '311'.
        DELETE et_msg_link INDEX sy-tabix.
      ENDIF.
    ENDLOOP.

  ENDCASE.

ENDMETHOD.

```

The figure below shows the clarification case that was created based on the given example application log and with the `DETERMINE_CASE` method of the `EMMA_CASE` BAdI implemented.

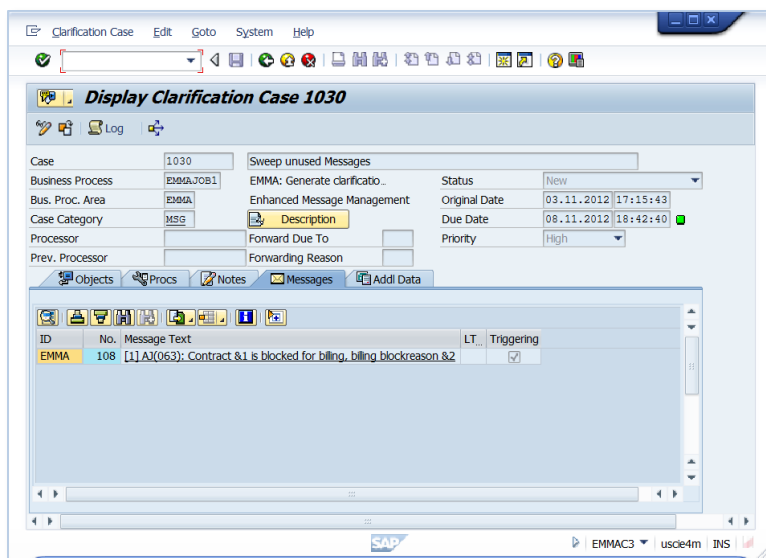


Figure 6-6 Messages of a Clarification Case for EMMA 108 Triggering Message

The advantage of the described approach is that clarification cases are created to identify all messages that have not led to any clarification case. However, there are some disadvantages coming with this approach: At the end of the clarification case creation process you must schedule the `FPEMMAMA` transaction once more to analyze the resulting application logs of the previous `FPEMMAMA` runs. In addition, the created clarification cases provide only an unsorted summary of error messages without a link to the actual business process and business objects that failed.

6.3.2 Process Code Specific Sweeper Clarification Case Categories

There is a simple way to overcome the disadvantage of re-executing the `FPEMMAMA` transaction for an analysis of the application log to identify unused messages or messages representing business exceptions without an attached clarification case. The solution is to create a clarification case category using the `EMMA 011` message as the triggering message and ensure that this clarification case category leads only to a clarification case if no previous clarification case category led to the desired clarification case. The figure below shows an example of such a clarification case category.

This new sweeper clarification case category must still be adjusted. Otherwise, it leads to the creation of a clarification case (assuming no identical clarification cases exist on the database) regardless if a previous clarification case already led to a clarification case. Two actions are required to ensure that this cannot happen:

The system must not process this sweeper clarification category prior to any other clarification case category of the same business process. You can ensure this easily in assigning a name to the new sweeper clarification case category that is last in the sorted list of the relevant clarification case categories to be processed (for more details, see chapter 5.3.1.1 *General Process*).

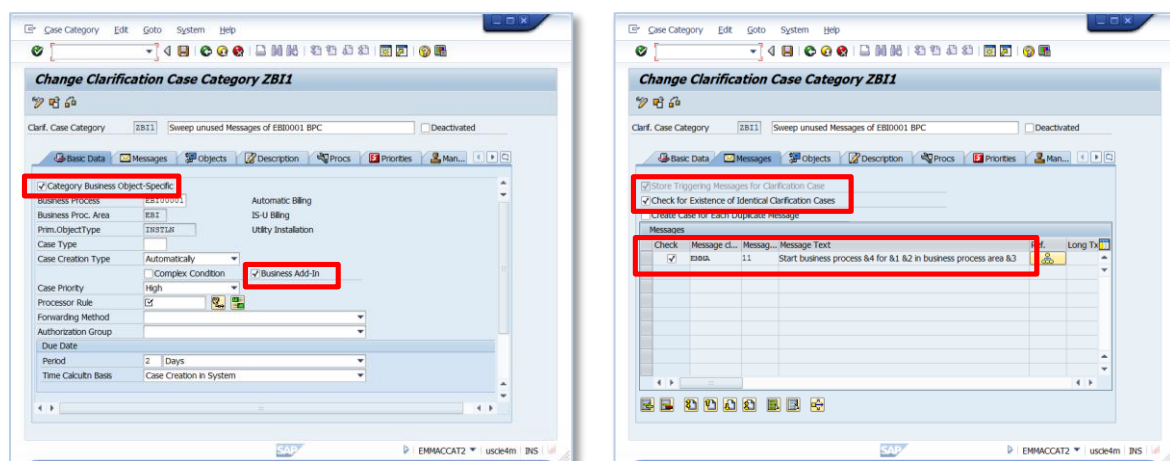


Figure 6-7 Definition of Business Process Code specific Sweeper Clarification Case Category

In addition, you need to implement the `COMPLETE_CASE` method of the `EMMA_CASE` BAdI and check the `IV_CASE_WITH_PREV_CCAT_CREATED` parameter if a previously processed clarification case category already led to a clarification case. If the system created a clarification case by any of the previously processed clarification case categories, you just raise the `NO_CASE` exception. You may use the following code snippet as a basis for your own implementation.

```
METHOD if_ex_emma_case~complete_case.

  CASE is_case-ccat.

  * sweeper clarification case category
  WHEN 'ZBI1'.
    " do not create case if a previously processed
    " clarification case category already created one
    IF iv_case_with_prev_ccat_created = abap_true.
      RAISE no_case.
    ENDIF.

  ENDCASE.
ENDMETHOD.
```

With the given implementation, the system creates a clarification case only if no clarification case was created by any of the previously processed clarification case categories. However, the resulting clarification case shows only the EMMA 011 message since it is the triggering message but not, as desired, the 'real' messages representing the business exception. To change this, you must implement the DETERMINE_CASE method of the EMMA_CASE BAdI to remove unwanted EMMA 011 message and other messages you may want to suppress.



Never suppress the EMMA 011 message in the Customizing (see chapter 3.3.4.1 *Suppress Messages during Preparation*). After a suppression of the EMMA 011 message, the system cannot link anymore the messages describing a business exception to a particular business object.

You may use the following code snippet as a basis for your own implementation.

```
METHOD if_ex_emma_case~determine_case.

  DATA: ls_it_msg TYPE LINE OF emma_msg_tab,
         ls_et_msg_link TYPE LINE OF emma_msg_link_tab.

  CASE is_ccat-ccat.

  *   sweeper clarification case category
      WHEN 'ZBI1'.

        " delete EMMA 011 triggering message
        REFRESH et_msg_link.

        " add messages (and eliminate possible duplicates)
        LOOP AT it_msg INTO ls_it_msg.
          IF ls_it_msg-msgid = 'EMMA' AND ls_it_msg-msgno = '011' OR
             ls_it_msg-msgid = 'AJ'   AND ls_it_msg-msgno = '311'.
            " ignore these undesired messages
          ELSE.
            MOVE-CORRESPONDING ls_it_msg TO ls_et_msg_link.
            READ TABLE et_msg_link FROM ls_et_msg_link
              TRANSPORTING NO FIELDS.

            IF sy-subrc <> 0.
              APPEND ls_et_msg_link TO et_msg_link.
            ENDIF.
          ENDIF.
        ENDLOOP.

      ENDCASE.
    ENDMETHOD.
```

Figure 6-8 shows a resulting clarification case now created with a proper message representing the business exception. Because this clarification case got its proper business process code and a business object attached to it, you may enhance such clarification cases easily in adding solution processes to the related clarification case category based on the primary object type of the attached business process code.

The disadvantage of this solution is that you need to create an own clarification case category per business process code. This is, because the clarification case generation process minds only clarification case categories with a business process code that matches with the business process code of the application as registered in the EMMA 011 message. However, the following chapter describes a solution that overcomes also this issue.

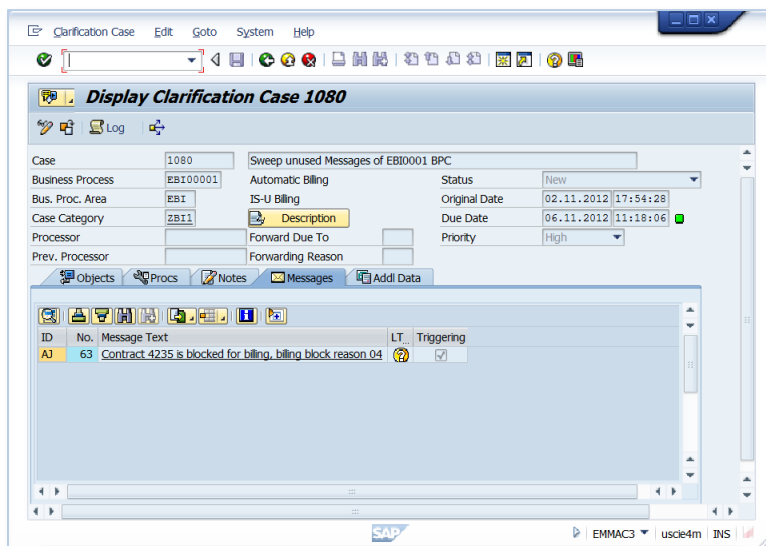


Figure 6-8 Resulting Clarification Case based on a Sweeper Clarification Category

6.3.3 General Sweeper Clarification Case Categories

As already explained in chapter 5.3.1.3 *Special Business Process Codes EMMACAS1 and EMMACAS2*, there is a special EMMACAS2 business process code that eventually resolves the disadvantages of the previously described approaches. The figure below shows the definition of a clarification case category based on the EMMACAS2 business process code and the EMMA 011 message as the triggering messages.

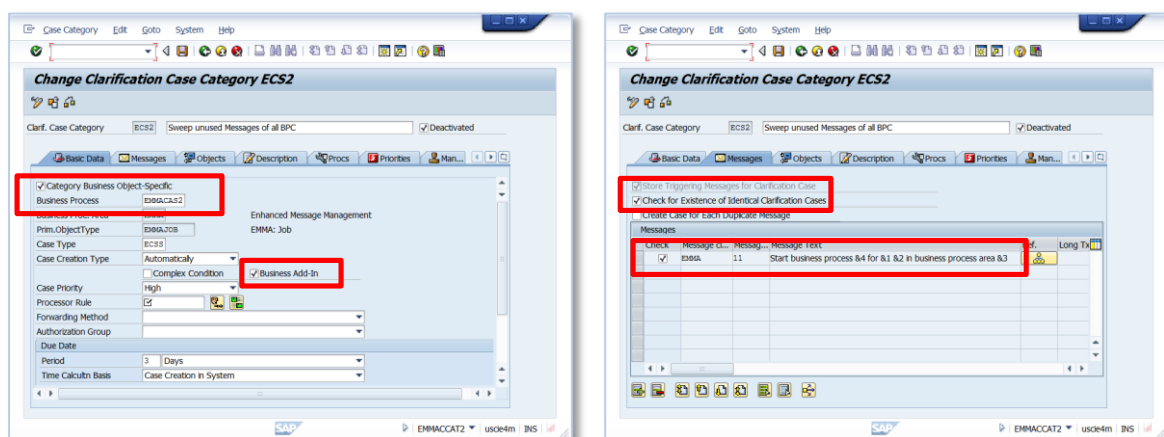


Figure 6-9 Definition of a General Sweeper Clarification Case Category

You implement the `DETERMINE_CASE` method of the `EMMA_CASE` BAdI to replace the undesired `EMMA 011` message and any further messages. You may use the following code snippet as a basis for your own implementation.

```
METHOD if_ex_emma_case~determine_case.

  DATA: ls_it_msg TYPE LINE OF emma_msg_tab,
        ls_et_msg_link TYPE LINE OF emma_msg_link_tab.

  CASE is_ccat-ccat.

    * sweeper clarification case category
    WHEN 'ECS2'.

      " delete EMMA 011 triggering message
      REFRESH et_msg_link.
```

```

" add messages (and eliminate possible duplicates)
LOOP AT it_msg INTO ls_it_msg.
  IF ls_it_msg-msgid = 'EMMA' AND ls_it_msg-msgno = '011' OR
     ls_it_msg-msgid = 'AJ'   AND ls_it_msg-msgno = '311'.
    " ignore these undesired messages
  ELSE.
    MOVE-CORRESPONDING ls_it_msg TO ls_et_msg_link.
    READ TABLE et_msg_link FROM ls_et_msg_link
                        TRANSPORTING NO FIELDS.

    IF sy-subrc <> 0.
      APPEND ls_et_msg_link TO et_msg_link.
    ENDIF.
  ENDIF.
ENDLOOP.

ENDCASE.

ENDMETHOD.

```

The system creates now a clarification case with a proper message representing the business exception and the correct business process code even just the `EMMACAS2` business process code is attached to the underlying clarification case category. Only the clarification case category indicates still that the system created a clarification case based on a generic sweeper clarification case category.



Via the `EMMACL` or `EMMACLS` transaction, you should periodically check for clarification cases that have been created based on such a generic clarification case category. If found, you should decide if a clarification case based on a specific clarification case category and specific solution processes supports the agents more efficiently in resolving the reported business exception than the more generic clarification case category with only rather generic solution processes.

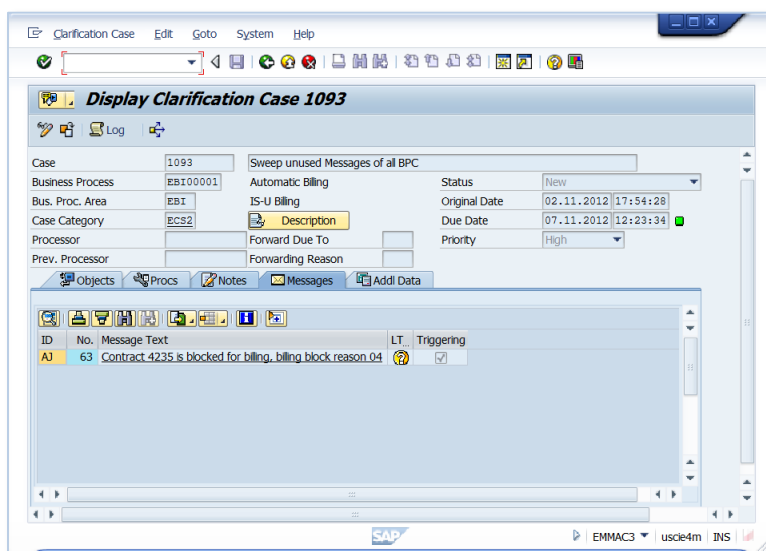


Figure 6-10 Resulting Clarification Case based on a generic Sweeper Clarification Category

The advantages of this solution is that, first, only one single generic clarification case category is required to create clarification cases for messages that lack of an appropriate clarification case category. Second, the `COMPLETE_CASE` method of the `EMMA_CASE` BAdI is not required to be implemented due to the nature of the `EMMACAS2` business process code (as described in chapter 5.3.1.3 *Special Business Process Codes* `EMMACAS1` and `EMMACAS2`).

With the described implementation approach, you offer a clarification case to the agent that shows the real error message and is bound to the correct business process code even it does not offer any specific solution process yet. However, you may consider creating clarification case category based on the main object type and define some few related solution processes to allow the agent at least to resolve the clarification case despite its general nature. Based on the implementation above, there are only three further steps required to make this happen:

- Step 1:** Determine the main object types for which you want to create specific clarification case categories. For example, you may consider the business object types `BUS1006` (business partner) and `ACCOUNT` (contract account), `CA_RECKEY` (reconciliation key) and, in addition for a SAP for Utilities implementation, the solution specific business object types `INSTLN` (utility installation), `ISUACCOUNT` (contract account) and `MTRREADDOC` (meter reading document).
- Step 2:** Create per business object type a clarification case category and add solution processes that are specific to the given business object type. Since most solution processes require business objects or their attributes as input you will have to add the required business objects to the clarification case container of the clarification case category prior to defining the data flow of the solution processes. The keys of the added business objects are determined at runtime during the generation of the clarification case as implemented in step 3.
- Step 3:** Enhance the `COMPLETE_CASE` method of the `EMMA_CASE` BAdI such that keys of the business objects of the clarification case container are determined based and allocated to the defined business objects. You may use the following code snippet as a basis for your own implementation. The code snippet uses the data finder of the SAP for Utilities industry solution.

```
METHOD if_ex_emma_case~complete_case.

    DATA: ls_object TYPE LINE OF emma_cobj_t,
           ls_search_para TYPE bapi_isufinder_efindpar,
           lt_search_result TYPE TABLE OF bapi_isufinder_result,
           ls_search_result TYPE bapi_isufinder_result,
           ls_return TYPE bapiret2,
           lv_incompl TYPE bapi_isufinder_e-incompl.

    CASE is_case-ccat.

    *   sweeper clarification case categories
        WHEN 'ECS2'.

    *   check if no clarification case was created yet and if the business
        object type matches the object type INSTLN for which the ECS2
        clarification case category was created with utility installation
        specific solution processes.
        IF iv_case_with_prev_ccat_created = abap_false AND
           is_case-mainobjtype = 'INSTLN'.

    *   determine current contract required for solution process
        IF NOT is_case-mainobjkey IS INITIAL.
            ls_search_para-installation = is_case-mainobjkey.
            ls_search_para-date_contract = sy-datum.
            CALL FUNCTION 'BAPI_ISUFINDER_FINDOBJOFDATA'
                EXPORTING
                    objecttype      = 'ISUCONTRACT'
                    searchparameters = ls_search_para
                IMPORTING
                    resultincomplete = lv_incompl
                    return           = ls_return
                TABLES
                    searchresult     = lt_search_result.
```

```

        READ TABLE lt_search_result INDEX 1 INTO ls_search_result.

*      update contract business object in clarification case container
      IF sy-subrc = 0.
        ls_object-celemname = 'ISUVertrag'.
        ls_object-refobjtype = 'ISUCONTRCT'.
        ls_object-id = ls_search_result-objkey.
        APPEND ls_object TO ct_objects.
      ENDIF.
    ENDIF.

    ELSE.
      RAISE no_case.
    ENDIF.

    WHEN 'ECZZ'.
*      create always a clarification case to indicate
*      requirement for a new clarification case category

    ENDCASE.

  ENDMETHOD.

```

The system creates the resulting clarification cases now not only with a proper message representing the business exception but also with appropriate solution processes supporting at least an initial analysis of the business exception.

The code snippet above demonstrates as well that it is possible to allow the creation of multiple clarification cases based on the same business exception. With the given implementation, the system creates a clarification case if the main business object type of the business exception matches the intended clarification case category. In addition, you may want to always create a clarification case of the `ECZZ` clarification case category (according to the code snippet above) intended to be sent to the attention of your corporate application support group to indicate the need for an addition clarification case.

6.4 Agent Determination

Most clarification cases require a manual intervention for a resolution. You can either create an agent specific clarification case worklist where the agent can pick the next clarification case to resolve (for details, see chapter 4.3.1 *Transactions EMMACL/EMMACLS – Clarification Case List*) or directly assign a clarification case to a specific agent or agent group by means of an agent determination rule. The following chapters describe the options for a direct assignment of clarification cases to agents in details.

6.4.1 Agent Determination Rule

You can assign an agent determination rule to a clarification case category. The system processes this rule only once during the creation of a clarification case. The agent determination rule returns organization structure entities, such as organization units or user IDs. The system stores these entities (so-called actors) with the clarification case as possible future processors. No further action takes place at this time.

You create and maintain agent determination rules with the `PFAC_INS` and `PFAC_CHG` transactions. The figures below shows the configuration steps of a simple rule to assign agents of an organizational unit to clarification cases, for example, based on business process codes. You may use the rule as a basis for your installation specific agent determination rules.

Step 1: Create the basic data of the new rule definition

The screenshot shows the 'Rule: Change' dialog box in SAP. The 'Basic data' tab is selected, displaying the following information:

- Rule: 56300001 BPEMCaseCirk
- Name: Clarific. Case Agents
- Pack.: \$TMP
- Applic. Component: (empty)

The 'Rule definition' tab is also visible, showing:

- Category: Agent Determination: Responsibilities
- ☐ Consider Secondary Priorities
- ☐ Personal Rule
- ☐ Terminate If Rule Resolution Without Result

Figure 6-11 Sample Agent Determination Rule

Step 2: Create a `BusinessProcessCode` container element on the *Container* screen with a reference to the `EMMA_BPCODE` ABAP Dictionary data type. The figure below shows the created container and the definition of the `BusinessProcessCode` container element.

The screenshot shows the 'Rule: Change' dialog box with the 'Container' tab active. A table lists the container elements:

Expression	M	Description	Init
BusinessProcessCode		Business Process Code of a Clarification Case	< N

The 'Change Container Element' dialog box is open, showing the definition for 'BusinessProcessCode':

- Element: BusinessProcessCode
- Name: BusinessProcessCode
- Short Descript.: BusinessProcessCode
- D. Type: Properties
- SelectionOfPredefinedTypes: ☐ Object Type
- ABAP Dict. Reference:
 - Structure: EMMA_CASE
 - Field: BPCODE
 - Clarification Case
 - Business Process Code
- ABAP Dict. Data Type: ☐ Type Name

Figure 6-12 Container of a Sample Agent Determination Rule

Step 3: Assign agents or organizations to the business process codes on the *Responsibility* screen. The figure below shows two so-called responsibilities with organizational units assigned.

For the sample agent determination rule two separate responsibilities for *Billing* and *Dunning* have been created and completed by the relevant organizational units. Of course, you can also define agent assignments for responsibilities based on any other entities of your organizational structure, such as work centers, positions or simply users instead of organizational units.



You should always try to take advantage of an already existing organizational structure and organizational units since then an agent assignment is bound to

users belonging to a specific organization and not necessarily bound to specific user IDs.

The following figure to the right shows that a responsibility rule based on the billing relevant business process codes. The second responsibility is similarly configured for dunning related business process codes.

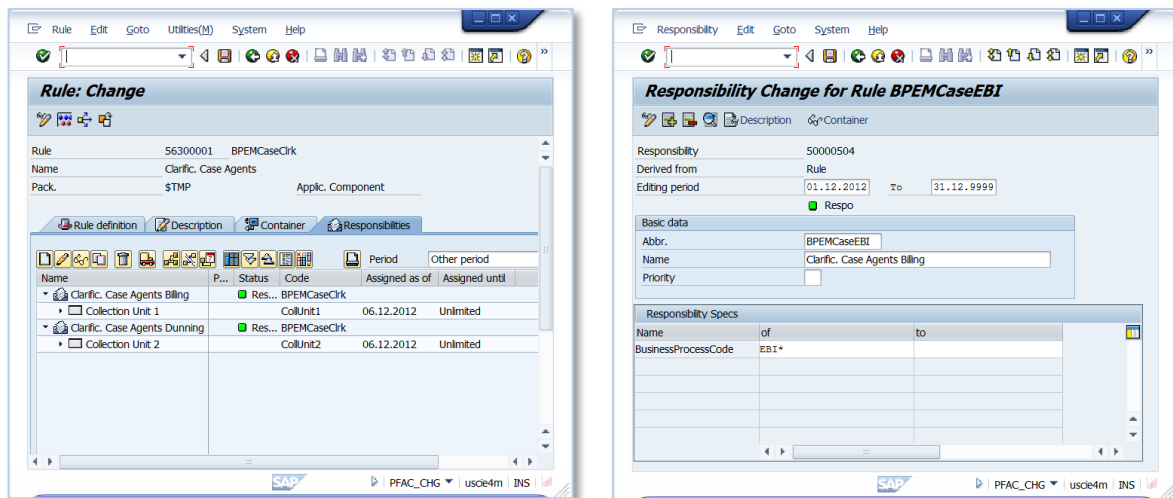


Figure 6-13 Assignment of Responsibilities

Step 4: Allocate the newly created agent determination rule to the clarification case category. Create subsequently the data flow as shown in the figure below.

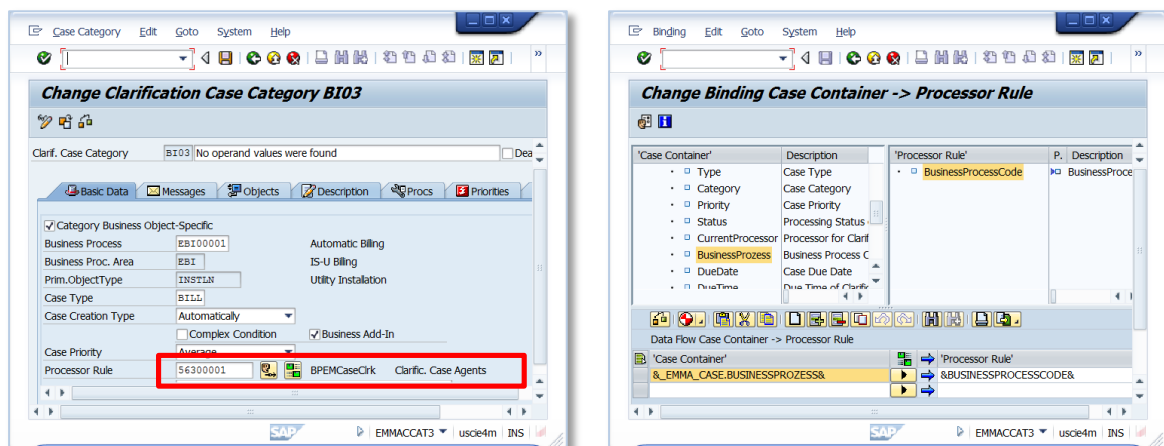


Figure 6-14 Assignment of an Agent Determination Rule to a Clarification Case

With the described configuration, the system processes the processor rule 56300001 after the creation of a new clarification case based on the BI03 clarification case category. The configuration returns the *Collection Unit 1* organizational unit for the BI03 clarification case category due to the attached EBI00001 business process code. The system stores the ID of the resulting *Collection Unit 1* organizational unit with the newly created clarification case. No further actions take place at this point of time.

As a result, agents can now select clarification cases for their personal clarification case worklists using organizational structure selection criteria. During the clarification cases selection, the system determines the clarification case worklist in comparing the entered selection criteria to the organizational structure information stored with the clarification cases.

The figure below shows the lower part of the selection screen of the `EMMACL` transaction where you can specify the details of the organizational structure (for more details about this transaction, see chapter 4.3.1 *Transactions EMMACL/EMMACLS – Clarification Case List*). In the example, the entered object type `O` (`O` = organization) and internal ID `50000077` describes the *Collection Unit 1* organizational unit.

Organizational Structure	
Plan version	01
Object type	O
Object	50000077
Structure Parameters	
Evaluation Path	
Start date	07.12.2012
End Date	07.12.2012
Technical depth	

Figure 6-15 Selection Screen of `EMMACLS` Transaction related to Organizational Structure



The agent cannot change the organizational unit the system assigned to the clarification case. If you need to change the assigned processor, you may implement, for example, a forwarding rule as described in the next chapter.



The system raises an error message if the assigned agent determination rule fails to return an organization structure entity. However, if this issue occurs during background processing the clarification case is created nonetheless.

6.4.2 Forwarding Rule

You can attach a forwarding rule to a clarification case. The system processes this forwarding rule once after the creation or a change of a clarification case (for more details about the related Customizing activity, see chapters 3.4.3.1 *Screen Basic Data* and 3.4.4 *Define Method for Forwarding Clarification Cases*).

During the clarification case creation, the system determines the forwarding rule by the processed clarification case category. However, it may be changed by the `COMPLETE_CASE` method of the `EMMA_CASE` Classic BAdI (for more details about the `COMPLETE_CASE` method, see chapter 5.8.3.4 *Amend and complete Clarification Case with installation-specific Data*). Note, that you cannot change the eventually determined forwarding rule anymore after the creation of the clarification case.

Technically, you implement a forwarding rule as an ABAP OO method. SAP provides the `IF_EMMA_CASE_FORWARD` ABAP OO interface with the `FORWARD` method for an implementation of your installation specific forwarding methods (for technical details, see chapter 5.8.4.1 *ABAP OO Interface IF_EMMA_CASE_FORWARD*). You may process a `COMMIT WORK` in your forwarding rule; however, a `COMMIT WORK` is not mandatory.

You can take advantage of a forwarding method to cope with various business requirements. The following list shows some common business requirements:

- Automatically forward a clarification case to specific specialized agents if requested by the agent who processed the clarification case initially.
- Automatically forward a clarification case to a different department on request of the agent hence changing the initial result of the agent determination rule
- Notify somebody by email if the due date of the clarification case is closing in and there was no action on the clarification case for a certain period.
- Check a clarification case and close it automatically if the related business exception was resolved but, for example, the agent missed to change the status of the clarification case to *Complete*.

For most of such requirements, there exist various possible implementation approaches. The following implementation examples have in common that, as a first action, the header data of the clarification case is retrieved in calling the public `GET_DATA` method of the `CL_EMMA_CASE` ABAP OO class. In addition, you may need to check if a forwarding method should still be processed in dependency of the clarification case status `LS_CASE-STATUS`.

The **first example** demonstrates the usage of the `BAPI_EMMA_CASE_FORWARD` BAPI to forward a clarification case to a second agent (for example a specialist) when an agent indicated such a request in entering the clarification case type to `SPEC` during the clarification case maintenance (for a Customizing of clarification case types, see chapter 3.4.2 *Define Clarification Case Types*). Instead of calling the `BAPI_EMMA_CASE_FORWARD` BAPI, you may as well call other BAPIs, such as the `BAPI_EMMA_CASE_COMPLETE` to complete a clarification case after having checked the current data environment of the business exception represented by the clarification case.

```
METHOD if_emma_case_forward~forward.

* local data definition
DATA: ls_case TYPE emma_case,
      lt_return TYPE bapiret2_t,
      ls_return TYPE LINE OF bapiret2_t.

* get clarification case data
CALL METHOD ir_case->get_data
  RECEIVING
    ev_case = ls_case.

* check if case requires attention of a specialist
IF ls_case-ctype = 'SPEC'.
  CASE ls_case-bpcode(3).

    WHEN 'EBI'.
      * billing specialist required
      IF ls_case-status = cl_emma_case=>co_status_inproc AND
         ls_case-currproc NE 'HUGO'.
      * forward the case to HUGO
      CALL FUNCTION 'BAPI_EMMA_CASE_FORWARD'
        EXPORTING
          case           = ls_case-casenr
          new_processor  = 'HUGO'
          reason_code    = 'SPEC'
        TABLES
          return         = lt_return.
      READ TABLE lt_return INTO ls_return WITH KEY type = 'E'.
      IF sy-subrc EQ 0.
        et_return = lt_return.
        RAISE error_forwarding_case.
      ENDIF.
    ENDIF.

    WHEN 'DUN'.
      * dunning specialist required
      [...]

  ENDCASE.
ENDIF.

ENDMETHOD.
```

The **second example** shows a code snippet that replaces the list of organizations and/or users that the system determines initially when processing the agent determination rule attached to the clarification case category. The implemented forwarding method checks if the agent specified a special forwarding reason while maintaining the clarification case. The business meaning of the entered forwarding reason is, for example, that the clarification case cannot be resolved within the current organization and must be further processed by another organization. Technically, the agent forwards the clarification case to another organization using a special forwarding reason (for a Customizing of forwarding reasons, see chapter 3.5.2 *Define Reasons for Forwarding Clarification Cases*). The technical answer to the entered forwarding reason is to replace the list of possible processors or organizational units of the clarification case with a new list or new organizational units as requested by the specific forwarding reason.

The code sample below shows the usage of the BAPI_EMMA_CASE_CHANGE BAPI to update the list of possible processors followed by a call of the BAPI_EMMA_CASE_BACK_TO_QUEUE to remove the current processor from the clarification case. In addition, the authorization group allocated to the clarification case is adjusted to cope with the new target processor group. Of course, you may enhance the example ABAP code below, for example, to allow clarification cases to be forwarded across your organization to find the required specialist outside of your organization.

```

METHOD if_emma_case_forward~forward.

* local data definition
CONSTANTS: co_fwd_reason_billing TYPE emma_case-reacode VALUE 'FBIL',
            co_otype_org TYPE bapi_swhactor-otype VALUE 'O',
            co_objid_org_bill TYPE string VALUE 'TaskForceBilling'.
DATA: ls_case TYPE emma_case,
      ls_case_change TYPE bapi_emma_case_change,
      lt_actors_del TYPE TABLE OF bapi_swhactor,
      lt_actors_add TYPE TABLE OF bapi_swhactor,
      ls_actor TYPE bapi_swhactor,
      ls_rh_object TYPE objec,
      ls_return TYPE LINE OF bapiret2_t.

* get clarification case data
CALL METHOD ir_case->get_data
  RECEIVING
    ev_case = ls_case.

* check if case is not yet closed and has no processor
CHECK ls_case-status < cl_emma_case=>co_status_compl AND
      NOT ls_case-reacode IS INITIAL.

* check if agent requested forward to a specific organization
CASE ls_case-reacode.
  WHEN co_fwd_reason_billing.

*   get current actors
  CALL METHOD ir_case->get_actors
    RECEIVING
      et_actors = lt_actors_del.

*   search for ID of requested organizational unit
  CALL FUNCTION 'REUSE_HR_OBJID_CHECK'
    EXPORTING
      otype      = co_otype_org
      seark      = co_objid_org_bill
      no_dialog  = 'X'
    IMPORTING
      sel_object = ls_rh_object
    EXCEPTIONS
      OTHERS     = 1.
  IF sy-subrc NE 0.

```

```

ls_return-type = sy-msgty.
ls_return-id = sy-msgid.
ls_return-number = sy-msgno.
ls_return-message_v1 = sy-msgv1.
ls_return-message_v2 = sy-msgv2.
ls_return-message_v3 = sy-msgv3.
ls_return-message_v4 = sy-msgv4.
APPEND ls_return TO et_return.
EXIT.
ENDIF.

* define new actors list to address Task Force Billing
ls_actor-otype = ls_rh_object-otype.
ls_actor-objid = ls_rh_object-objid.
APPEND ls_actor TO lt_actors_add.

* update the actors list if required
IF lt_actors_del[] <> lt_actors_add[].
  MOVE-CORRESPONDING ls_case TO ls_case_change.
  ls_case_change-brgru = 'TFBI'.
  CALL FUNCTION 'BAPI_EMMA_CASE_CHANGE'
    EXPORTING
      case           = ls_case-casenr
      case_change    = ls_case_change
      forward        = abap_false
    TABLES
      actors_delete  = lt_actors_del
      actors_add     = lt_actors_add
      return         = et_return.
ENDIF.
CALL FUNCTION 'BAPI_EMMA_CASE_BACK_TO_QUEUE'
  EXPORTING
    case           = ls_case-casenr
    reason_code    = ls_case-reacode.

ENDCASE.

* error handling
READ TABLE et_return WITH KEY type = 'E' TRANSPORTING NO FIELDS.
IF sy-subrc EQ 0. RAISE error_forwarding_case. ENDIF.

ENDMETHOD.

```

 You can implement an installation specific function to support the agent in selecting a new organizational unit in offering a list of possible and relevant organizations. During the processing of the forwarding rule the implemented ABAP OO method picks up the selected new organizational unit and replace the previous organization unit attached to the clarification case. For more information on installation specific functions, see chapter 6.2 *Installation-Specific Functions*.

The **third example** of a forwarding method shows a simple code snippet that sends a notification email to an agent once a clarification case was allocated to him or her.

```

METHOD if_emma_case_forward~forward.

* local data definition
DATA: ls_case TYPE emma_case.
DATA: lt_addsmtp TYPE STANDARD TABLE OF bapiadsmtp,
      lt_return TYPE STANDARD TABLE OF bapiRET2,
      lv_ex_bcs TYPE REF TO cx_bcs,
      lv_subject TYPE string,
      lv_body TYPE string,
      lr_msg_obj TYPE REF TO cl_bcs_message.

* determine recipient's email address
CALL FUNCTION 'BAPI_USER_GET_DETAIL'
  EXPORTING
    username      = sy-uname
    cache_results = 'X'

```

```

TABLES
    return          = lt_return
    addsmtp         = lt_addsmtp.
READ TABLE lt_addsmtp ASSIGNING FIELD-SYMBOL(<fs_addsmtp>)
                                WITH KEY std_no = abap_true.
IF sy-subrc = 0.

* get clarification case data
CALL METHOD ir_case->get_data
    RECEIVING
        ev_case = ls_case.

* send email only if there was a change of the processor
IF ls_case-currproc NE ls_case-prevproc AND
    NOT ls_case-currproc IS INITIAL.

* compose subject line
lv_subject = 'Clarification Case' && | | && ls_case-casenr.

* compose email body
lv_body = 'Dear colleague,' &&
    cl_abap_char_utilities=>cr_lf &&
    cl_abap_char_utilities=>cr_lf &&
    'The following clarification case was assigned to you:'
    && | | && ls_case-casenr &&
    cl_abap_char_utilities=>cr_lf &&
    cl_abap_char_utilities=>cr_lf &&
    |Thanks & best regards!! &&
    cl_abap_char_utilities=>cr_lf &&
    'Your Notification Service'.

TRY.
* send email
CREATE OBJECT lr_msg_obj.
lr_msg_obj->set_subject( lv_subject ).
lr_msg_obj->set_main_doc( lv_body ).
lr_msg_obj->add_recipient( CONV #( <fs_addsmtp>-e_mail ) ).
lr_msg_obj->set_requested_status( 'N' ).
lr_msg_obj->set_update_task( abap_false ).
lr_msg_obj->set_send_immediately( abap_true ).
lr_msg_obj->send( ).
CATCH cx_bcs INTO lv_ex_bcs.
    MESSAGE e315(fdt_expressions) RAISING error_forwarding_case.
ENDTRY.
ENDIF.
ENDIF.

ENDMETHOD.

```



If the clarification case is in status *New*, both `BAPI_EMMA_CASE_CHANGE` and `BAPI_EMMA_CASE_FORWARD` function modules change the status of the clarification case to *In Process*.

6.4.3 Workflow and Workflow Events

The system generates the `EMMACASE.Created` workflow event after creating a clarification case. Furthermore, the system creates the `EMMACASE.Change` workflow event after a changed clarification cases is saved on the database. You can use the workflow events to design a clarification case maintenance workflow. The following description does not want to replace a proper workflow training but shall help you to create a first workflow prototype for testing purposes. You find the required transactions in the *SAP Menu → Tools → Business Workflow → Development → Definition Tools → Tasks / Task Groups*. The basic steps to create a BPWM related workflow are:

Step 1: Create a workflow template with the `PFTC_INS`.

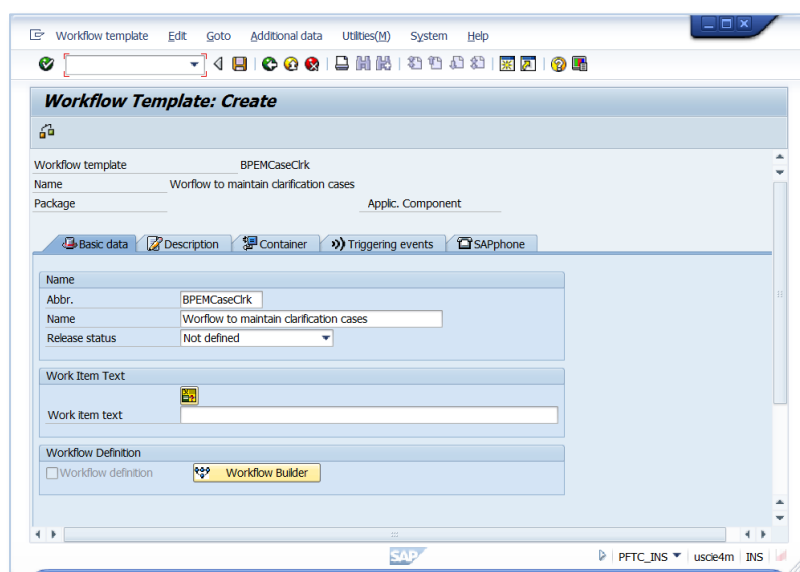


Figure 6-16 Basic Data Screen of the Workflow Template

Step 2: Create a container element to represent the clarification case and mark it as importing parameter on the *Property* screen.

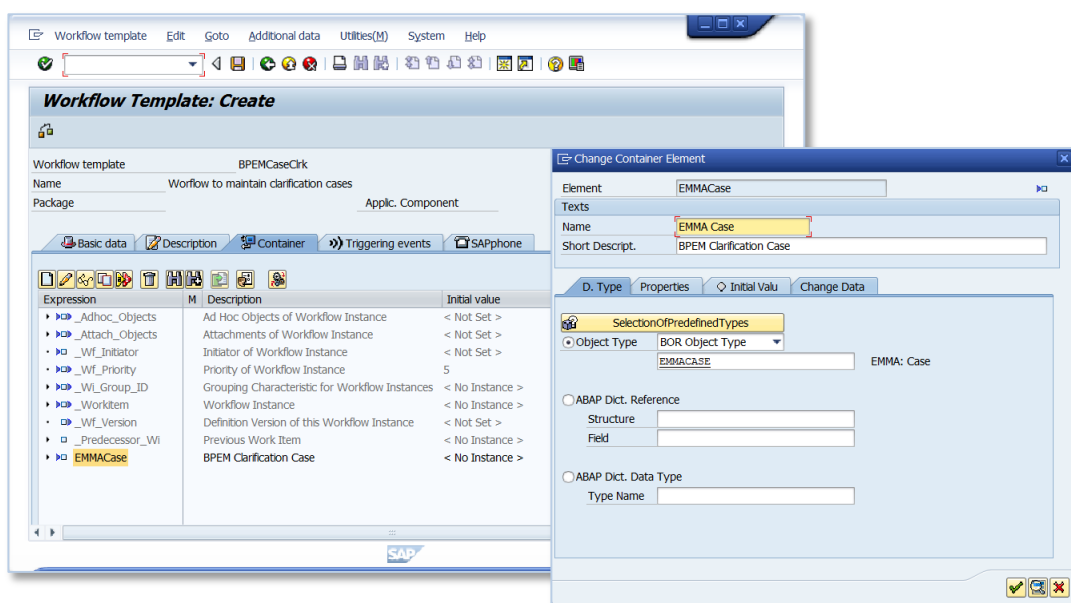


Figure 6-17 Container Screen of the Workflow Template

Step 3: Add the triggering event `EMMACASE.Created`.

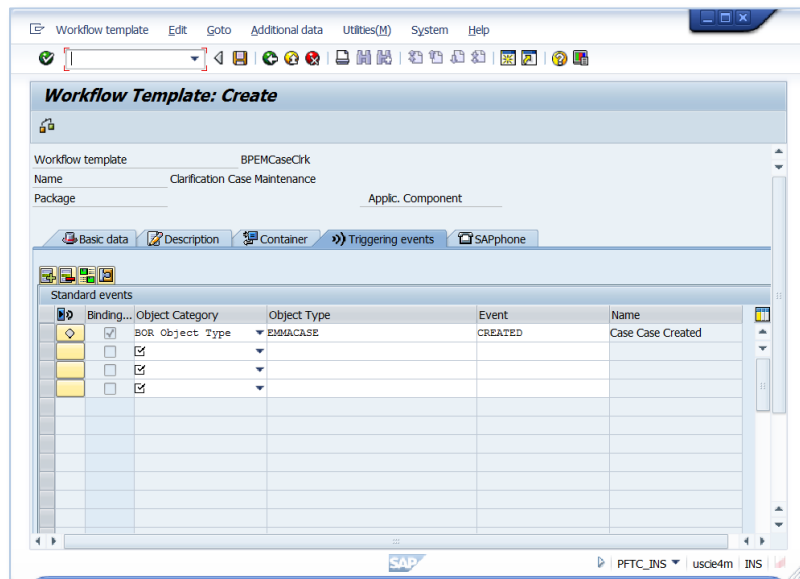


Figure 6-18 Triggering Events Screen of the Workflow Template

Step 4: Mark the line with the entered event and push the  button to create the binding between the generic event object and the previously created `EMMACase` container element of your workflow template.

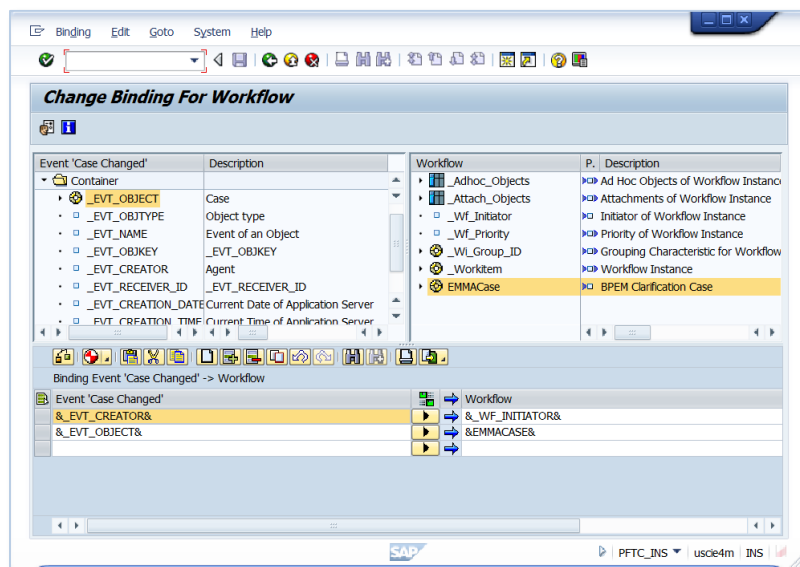


Figure 6-19 Binding of the Triggering Events of the Workflow Template

Step 5: Mark the triggering event as active.

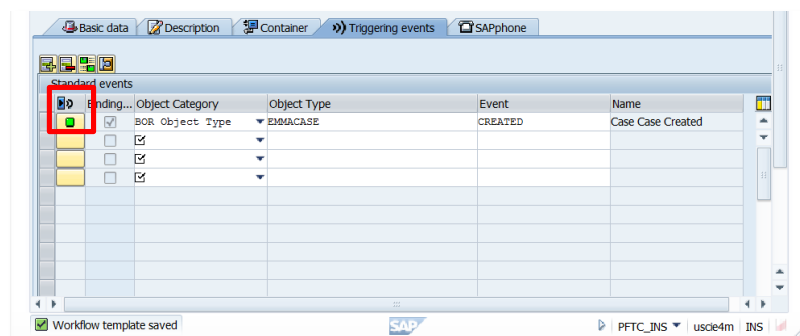


Figure 6-20 Activation of Triggering Events of the Workflow Template

Step 6: Push the  *Workflow Builder* button on the *Basic data* screen to start the workflow builder.

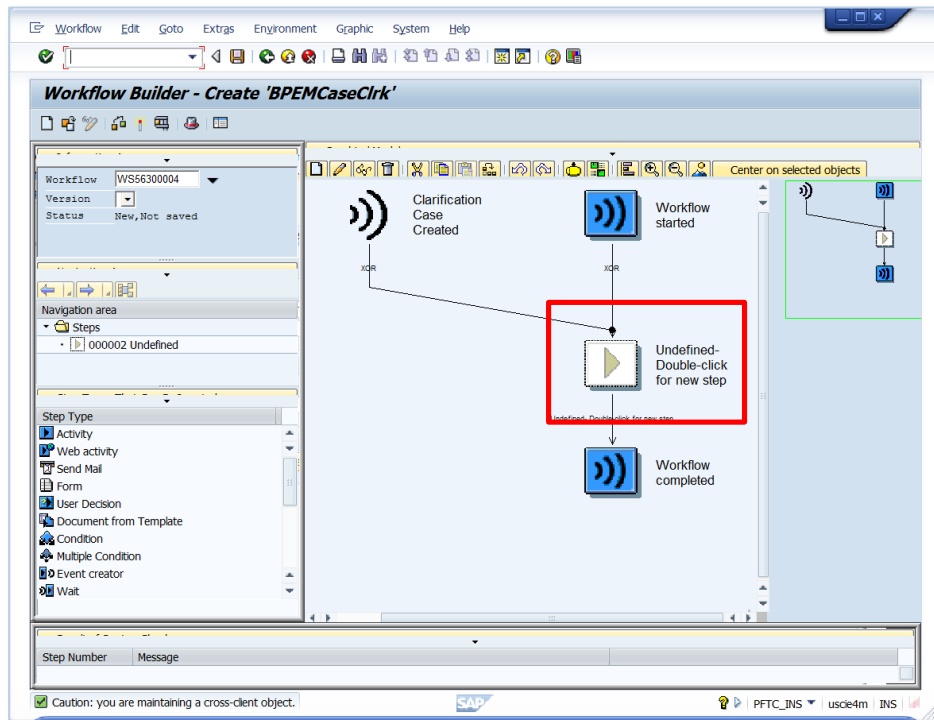


Figure 6-21 Initial Basic Workflow

Step 7: Double-Click on the undefined step symbol and choose *Activity* on the popup as the workflow step to be defined. On the workflow step maintenance screen choose *Create Task*.

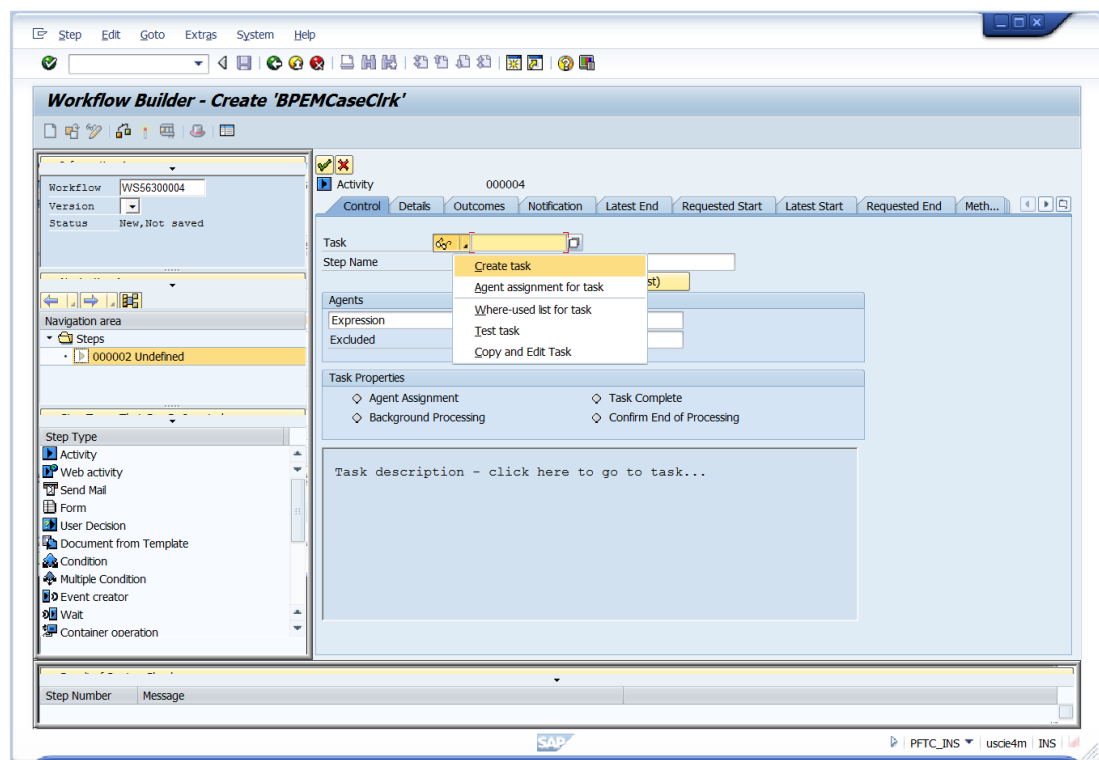


Figure 6-22 Creation of the new Workflow Task

Step 8: Maintain the workflow task. Prior to saving the task, change the classification of the workflow task to *General Task* (choose menu *Additional data* → *Classification* → *Create*).

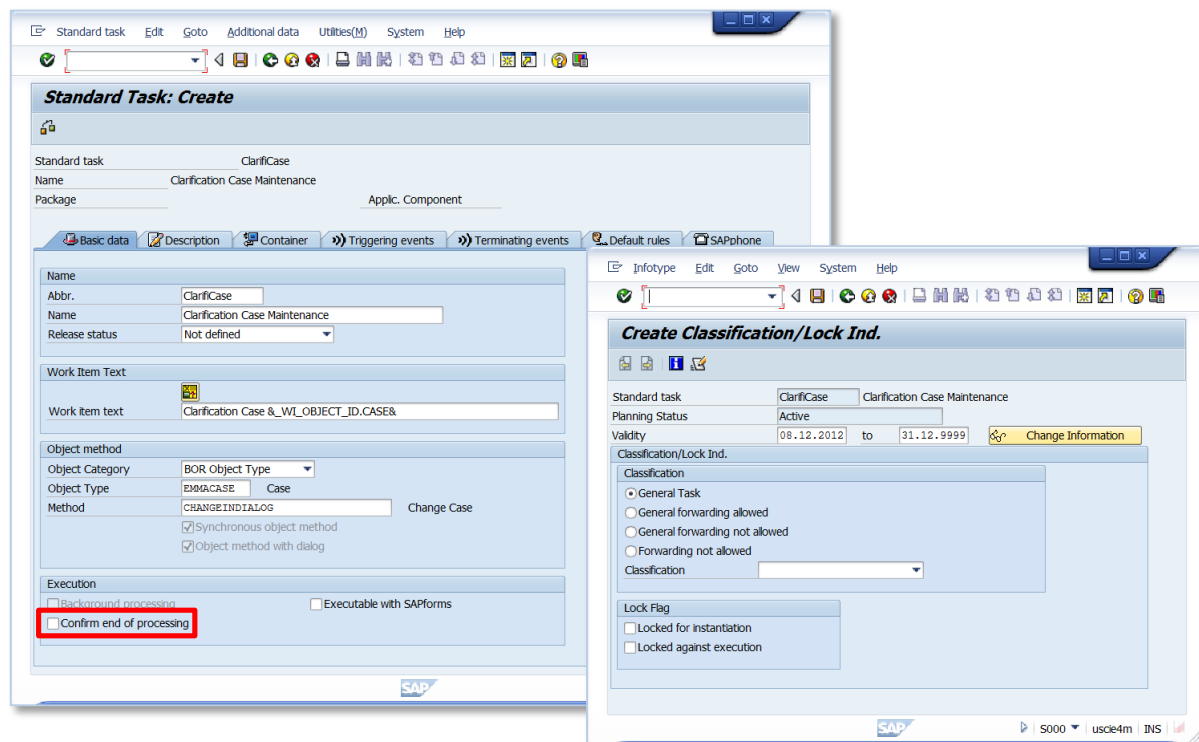


Figure 6-23 Workflow Step Activity



You have to mark the *Confirm end of processing* field if the agent should confirm if the work item related to the clarification case can be deleted from the inbox.

Step 9: Return to the workflow builder

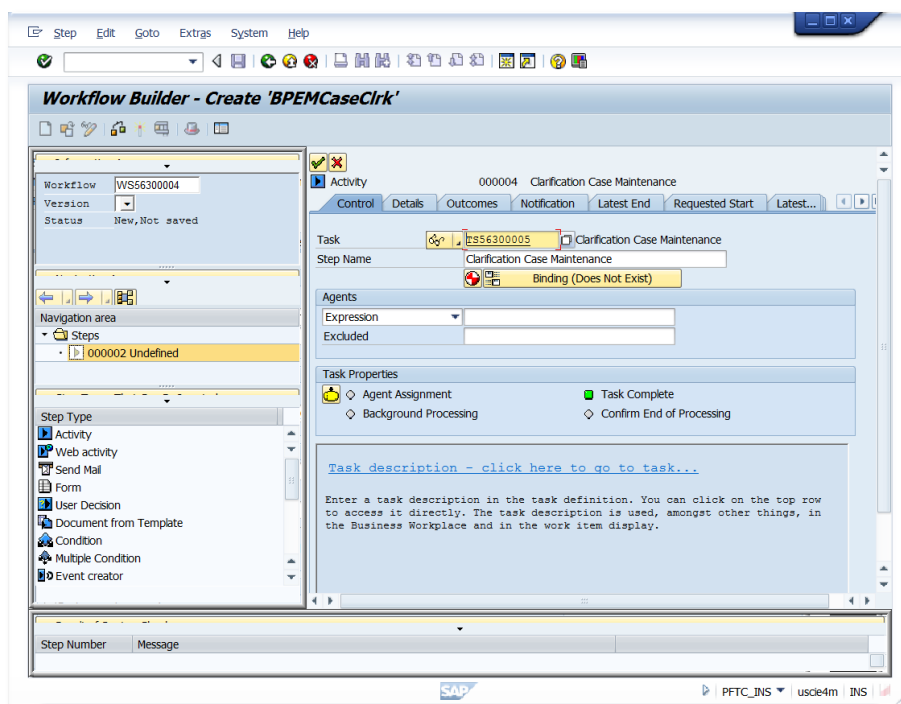


Figure 6-24 Control Screen of the Workflow Step Activity

Step 10: Push the  button to create the binding between the workflow and the created workflow task. In the binding editor simply push the  button to generate the binding. Then return to the workflow step maintenance screen.

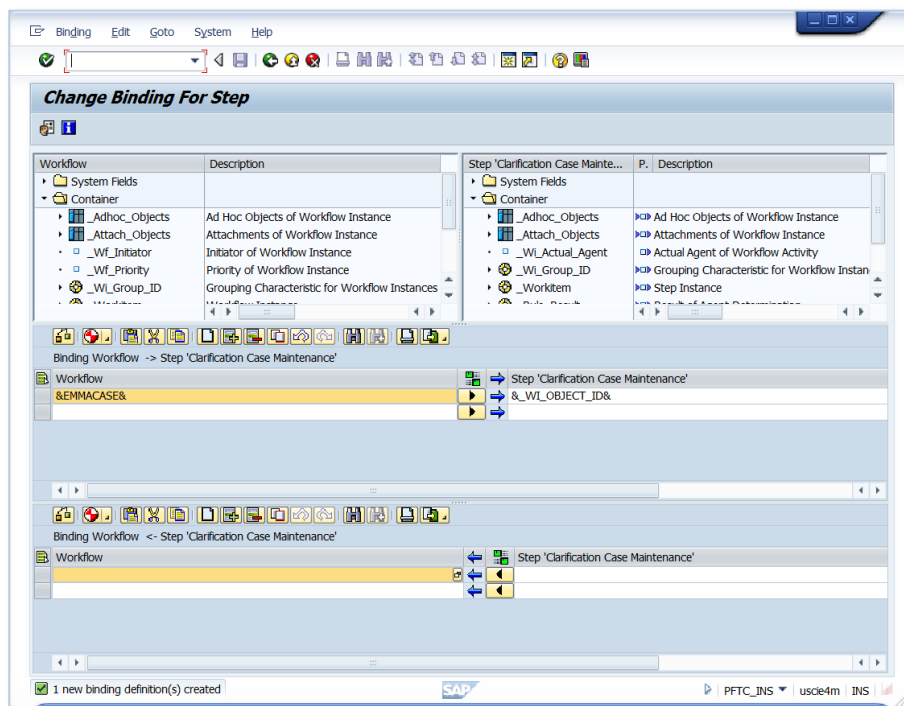


Figure 6-25 Binding of the Workflow Step of the Workflow

Step 11: Choose *Rule* and enter an agent assignment. Instead of allocating an agent assignment rule to the workflow step itself you may allocate the same or another rule to the underlying workflow task as a default agent rule assignment rule or even create a direct agent assignment on workflow task level.

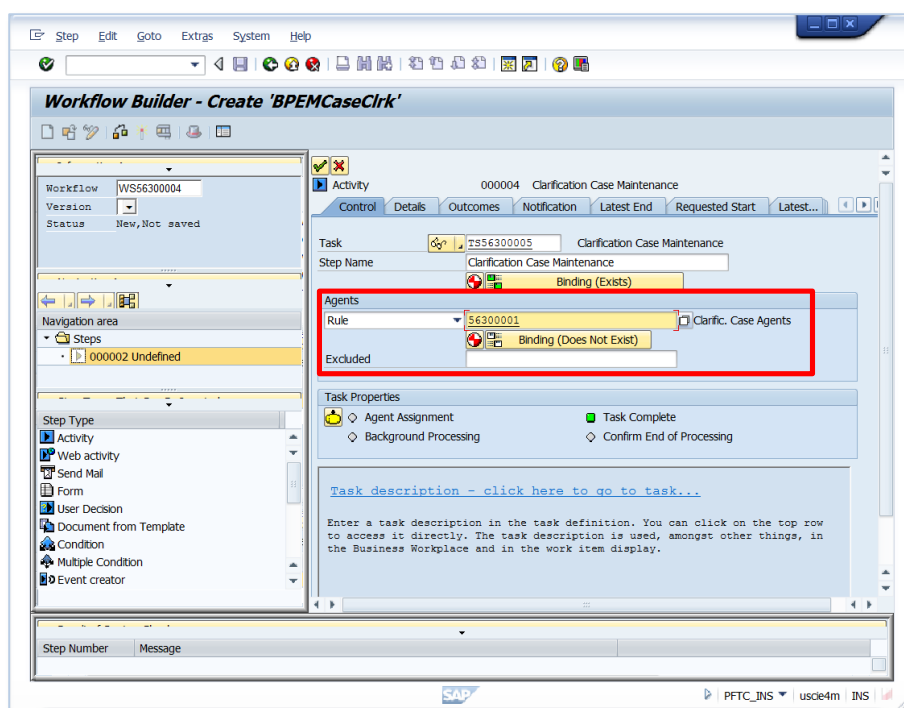


Figure 6-26 Allocation of Agent Assignment Rule to Workflow Step Activity

 You can attach either an entity of the organizational structure rules to the workflow step as a fixed agent determination rule or, if you require a dynamic determination of responsible agent, assign an agent assignment rule as defined in chapter 6.4.1 *Agent Determination Rule* to the workflow activity.

Step 12: Push the  button to generate the binding between the *BusinessProcessCode* clarification case attribute and the *BusinessProcessCode* importing container element of the agent assignment rule. The figure below shows the resulting binding.

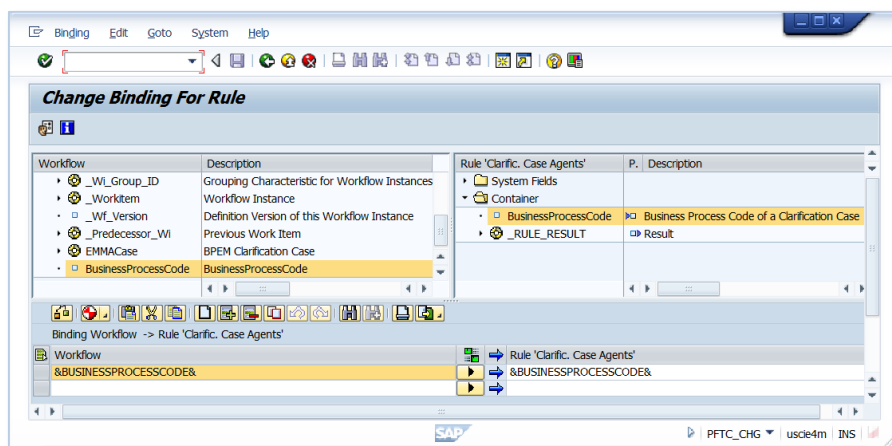


Figure 6-27 Binding of the Agent Assignment Rule of the Workflow Step

Step 13: Return to the workflow activity maintenance screen and from there to the workflow builder. Save and activate the newly created workflow.

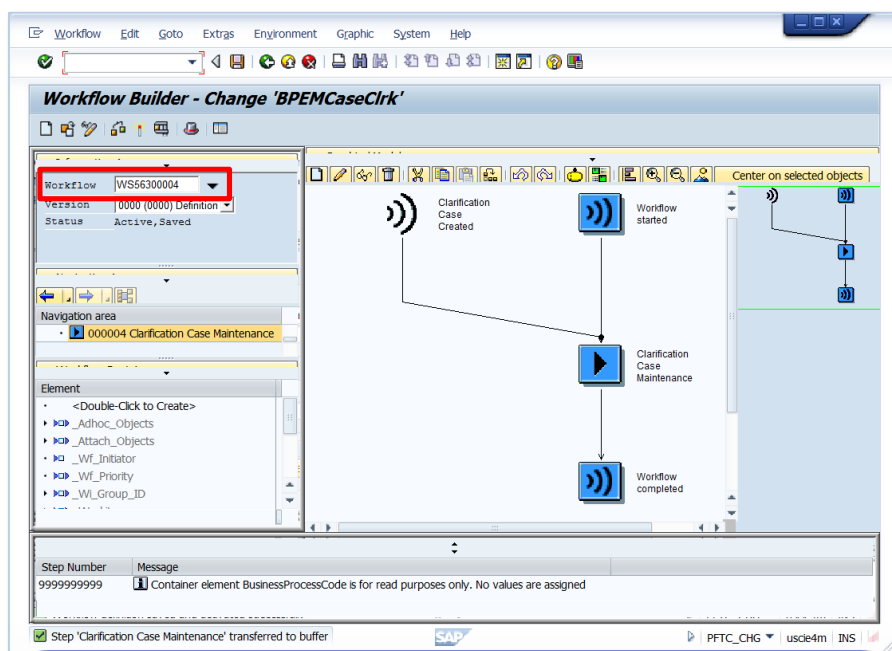


Figure 6-28 Workflow triggered by the *EMMACASE .Created Workflow Event*

The system creates automatically an entry in the so-called event type linkage table to link the *EMMACASE .Created* workflow event and the created workflow. The system checks for active event type linkages whenever an application creates a workflow event. The system starts for all found active linkages the related workflow or creates a work item for the entered workflow task. The figure below shows the created entry in the event type linkage table (*SWETYPV* transaction).

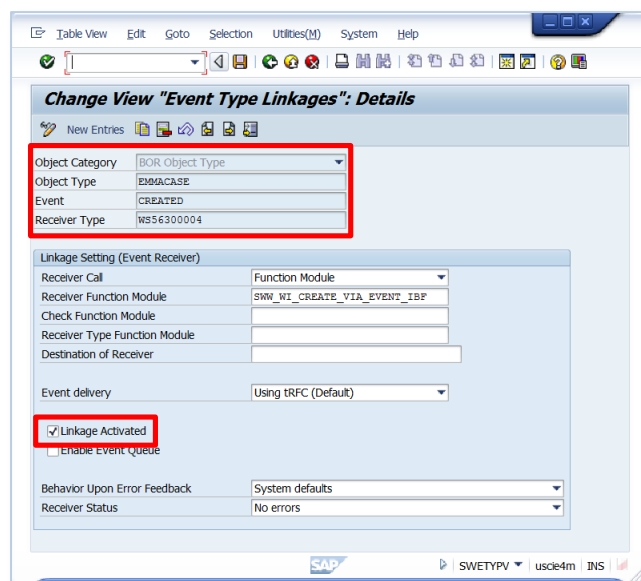


Figure 6-29 Event Type Linkage for EMMACASE.Created Workflow Event

Of course, the example workflow is very basic and does not cope with sophisticated business requirements hence further development and additional workflow development activities are required.

 SAP provides the WS20500233 workflow for utilities customer as an example workflow to derive the division from various primary object types of a clarification case.

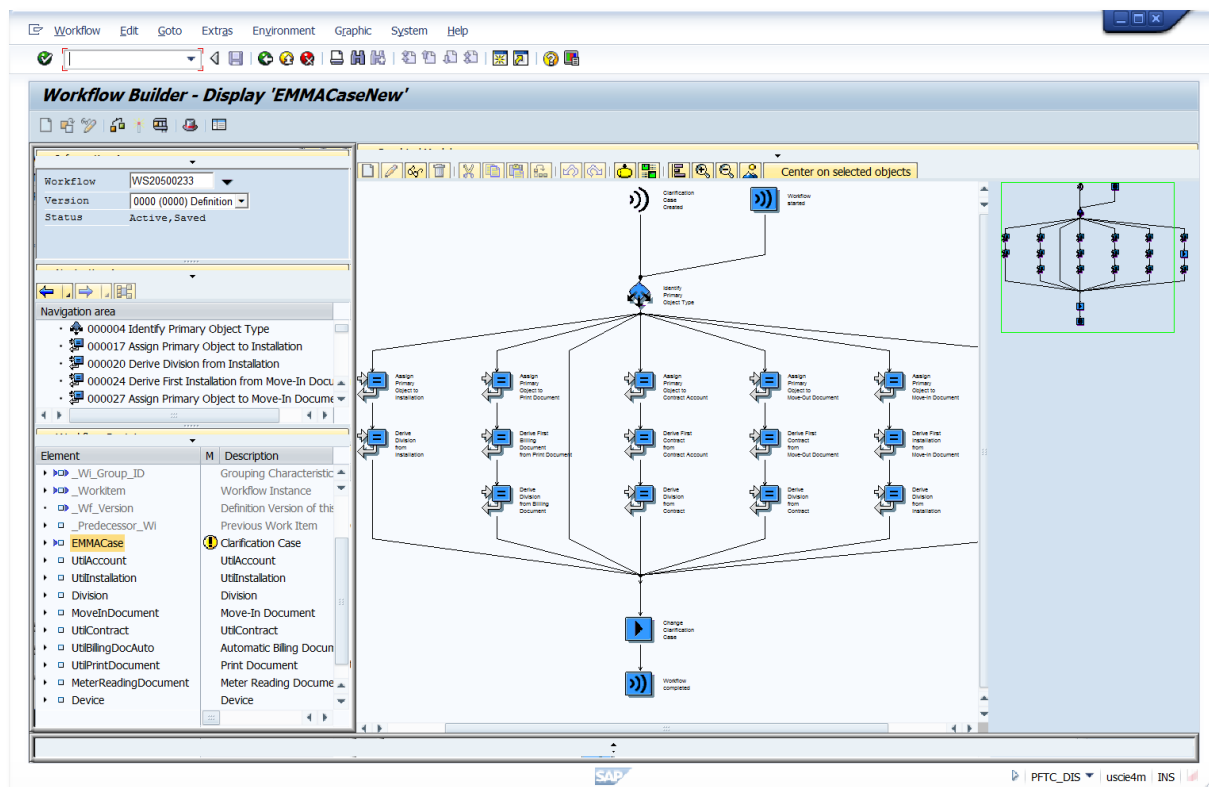


Figure 6-30 Template Workflow WS20500233

In addition, you must consider how to integrate the option to forward a clarification case to another agent group or organization if the initial processor cannot resolve the issue. Further considerations are required to design the workflow such that the root of a business exception has been resolved outside of the related clarification case and the clarification case was automatically closed by an installation specific application in processing, for example, the `BAPI_EMMA_CASE_COMPLETE` BAPI.

6.5 Connect Reports and Interfaces to BPEM

In most SAP implementations there are many installation specific reports and interfaces that write error and status information in own application logs. You must pay special attention also to those application logs and their respective analysis because commonly these application logs are either not monitored at all or an analysis is as much time consuming as an analysis of the huge application logs of many mass activities.

This chapter explains how you can easily connect your reports and interfaces to BPEM to take full advantage of all the BPEM functions, such as the automatic creation of clarification cases to follow-up the registered business exceptions.

There are two common implementation designs of reports and transactions:

- Many reports follow the design of, first, collecting data and, second, output the collected data to a database table or to the screen, for example, by means of an ABAP List Viewer ALV. From there agents eventually download the data to spreadsheets for a further analysis and processing.
- There are other scenarios where the business processes are executed periodically with changing business objects whereas occurring business exceptions are logged in an application log. System interfaces and mass processes are typical examples of such an implementation.

The general BPEM implementation approach is to write error messages and, if required, success messages to an application log enhanced by BPEM specific control messages that enclose the application messages and, at the same time, link them to the processed main business object and business object type by its message variables. BPEM transactions can process such an application log instantly to executing the BPEM specific processing steps as described in chapter 2.2 *Overview BPEM Processes and Transactions*.

Alternatively, you can create clarification cases directly in your application immediately after a business exception occurred in calling one of the available BAPIs, such as the `BAPI_EMMA_CASE_CREATE` function module. However, with either approach you must first setup a BPEM specific Customizing to connect your installation specific application to BPEM.

6.5.1 Connection to BPEM via Application Log

The following step-by-step description assumes that there is a single message you use to transfer all business process or business exception related information to the application log. The idea is to use the same message as a triggering message in clarification case categories.

Step 1: Create an installation-specific message with the `SE91` transaction that describes why a particular clarification was created, for example, *Installation &1 has not been billed since &2* and add a descriptive long text to the message.

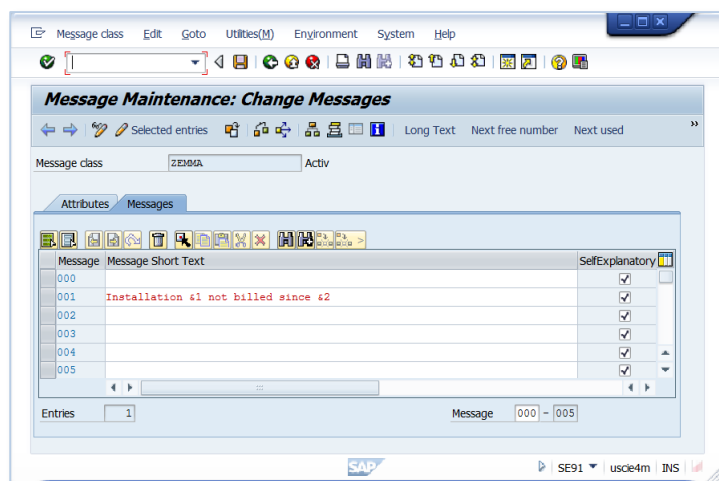


Figure 6-31 Installation specific Messages

 The figure above shows the usage of only two out of four standard message variables indicated by the placeholders &1 and &2. However, you can add further named message variables to a message during runtime thus you will have to add more message variables to the Customizing as described in the next step even they are not used in the message definition itself.

Step 2: Define for all your message variables the referenced business objects in the Customizing activity *Define Business Objects for Customer-Defined Messages* to allow navigation from the clarification case to these business objects (for more details, see chapter 3.6.5).

 It simplifies the creation of clarification case categories based on this message when you define all message variable in this Customizing because all referenced objects of the message will be added automatically to the clarification case container when creating the clarification case category (see step 4).

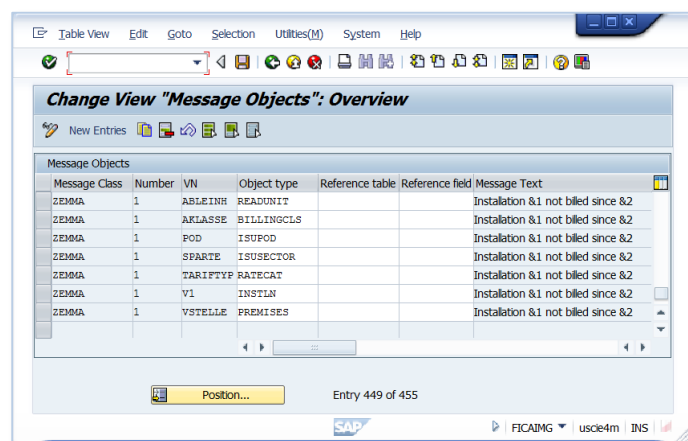


Figure 6-32 Definition of Business Object for Installation specific Messages

Step 3: Define a business process code:

- Check if a suitable business process area is available (see chapter 3.6.1 *Define Business Process Areas for Message Management*). Create a new business process area in the customer name space if required.
- Check if suitable business process code is available (see chapter 3.6.2 *Define Business Processes for Message Management*). Create a new business process code in the customer name space using an appropriate job analysis class to

support the BPEM (New EMMA) transactions. You may also consider creating a new business process area if there is no suitable standard business process area.

New Entries: Details of Added Entries

Business Process: ZBIL001 Identification of installations not billed for a long time

Business Process Code: EB1

Primary Object Type: inst1n

Job Analysis Class: CL_EMMA_JOB_SINGLE

Classes for Analysis of Application Logs (Old Version)

Job Analysis Class: ☒

Log Number Class:

Interval Info. Class:

Analysis Class:

Message for Identification of Execution Mode

Mode Message:

Execution Mode: Unknown

Messages for Identification of a Business Object

Start Message:

End Message:

Bus. Object Variable:

Additional Test of Contents of Variables

Message variable 1:

Message variable 2:

Message variable 3:

Message variable 4:

Figure 6-33 Installation specific Business Process Code

Step 4: Create a clarification case category in the Customizing activity *Create and Maintain Clarification Case Categories* based on your newly created business process code (for more details see chapter 3.4.3).



You can create a clarification case based on a specific message with the EMMACCAT1M transaction. This transaction copies all relevant information based on a message number (including the long text of the message) to the new clarification case category in addition to the entered business process code.

Create Clarification Case Category

Clarif. Case Category: B121 Installation &1 not billed since &2 Deactivated

Basic Data

☒ Category Business Object-Specific

Business Process: ZBIL001 Identification of installations not billed for a long time

Prim. Object Type: INST1N Utility Installation

Case Type: Automatic

Case Creation Type: Average

Case Priority: Average

Processor Rule: ☒ Business Add-In

Forwarding Method:

Authorization Group:

Due Date:

Period: 5 Days

Time Calculat. Basis: Case Creation in System

Messages

Check	Message ID	Message	Message Text	Ref.	Long Text
☒	1	Installation &1 not billed since &2			

Figure 6-34 Basic Data and Messages Screen of a simple Clarification Case Category

In the figure above, the most important entries are marked: the newly created ZBILL001 business process code and the ZEMMA(001) triggering message. You must mark also the *Category Business Object Specific* flag if the clarification case is related to a specific business object.

The following figure shows the clarification case container and the *Description* screen. The system adds automatically the relevant business objects and the long text of the message to the clarification case container if you created the clarification case category with the EMMACCAT1M transaction. Otherwise, you must add the business objects either manually or simply in pushing the  button to allow the system to check for relevant business objects you defined in the Customizing activity *Define Business Objects for Customer-Defined Messages* during step 2.

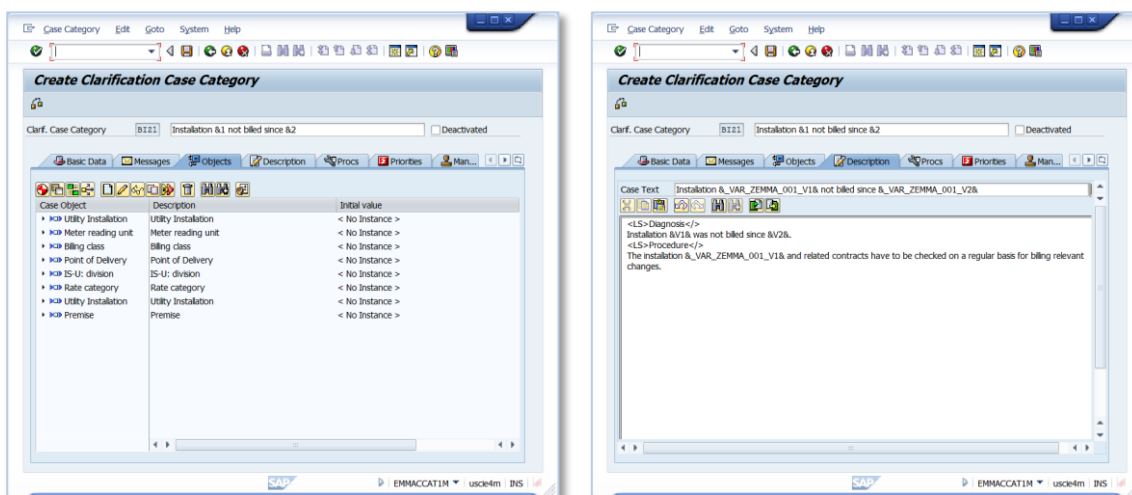


Figure 6-35 Objects and Description Screen of a simple Clarification Case Category

In subsequent steps, you complete the configuration of the clarification case category, for example, in adding appropriate solution processes and further configuration.

Step 5: Enhance your installation specific report in adding BPEM related information to the application log. You find the complete ABAP code in *Appendix A1 - Creation of Clarification Case via Application Log*. It includes also the macro definitions that simplifies the code to access to the application log.

The described enhancements create an application log for the messages, as well as the mandatory BPEM header.

- Check whether your business process code is marked as active using the EMMA_LOG_PROCESS_CHECK_ACTIVE function module.

```
* Step 1: Check if BPEM business process is active
CALL FUNCTION 'EMMA_LOG_PROCESS_CHECK_ACTIVE'
  EXPORTING
    iv_bpcode          = gc_bpcode
  IMPORTING
    ev_bparea          = gv_bparea
    ev_bpa_active      = gv_bpa_active
  EXCEPTIONS
    extract_class_not_found = 1
    bpcode_not_found       = 2
    bparea_not_found       = 3
    OTHERS                  = 4.
```

- Create an application log with an appropriate log object and a unique external application log ID, for example, in using the `BAL_LOG_CREATE` function module.

```
* Step 2: create application log
gs_log-object      = gc_log_object.
gs_log-subobject   = gc_log_subobject.
CONCATENATE 'ZREPORT_1' sy-datum sy-timlo INTO gs_log-extnumber.
gs_log-alprog      = sy-cprog.
gs_log-altcode     = sy-tcode.
CALL FUNCTION 'BAL_LOG_CREATE'
  EXPORTING
    i_s_log          = gs_log
  IMPORTING
    e_log_handle     = gv_log_handle.
```

- Register your business process. The system creates a new BPEM header entry for the later BPEM specific activities, such as the automatic generation of clarification cases.

```
* Step 3: register your business process
CALL METHOD cl_emma_job=>create_job
  EXPORTING
    iv_extnumber = gs_log-extnumber
    iv_bpcode    = gc_bpcode
    iv_tcode     = gc_tcode
    iv_silent    = 'X'
  EXCEPTIONS
    not_created = 1.
```

- Create an instance of the `CL_EMMA_BPC` ABAP OO class. This simplifies the creation of the BPEM specific messages you want to add to the application log.

```
* Step 4: Create instance for BPEM start/end object message
CREATE OBJECT gr_bpc
  EXPORTING
    iv_bpcode = gc_bpcode.
```

The previous development relate to the preparation for a creation of your installation-specific messages to the application log eventually completed by the BPEM specific control messages.

In the concluding implementation step 5 you add not only your installation-specific messages to the application log but also the mandatory preceding and successive BPEM control messages. Of course, you need to place the suggested code within a loop if your report or interface creates a list that includes all relevant information about all executed business processes. As previously said, the used ABAP macros are defined in the provided sample report in *Appendix A1 - Creation of Clarification Case via Application Log*.

- Create the BPEM specific start message based on the ID of the currently processed main business object using the previously created instance of the `CL_EMMA_BPC` ABAP OO class and write it to the application log with the `MACRO_ADD_MSG_TO_BAL_LOG` macro.

```
* Step 5a: Create BPEM start message and write it to application log
CALL METHOD gr_bpc->start
  EXPORTING
    iv_main_bus_objtype = gc_main_object_type
    iv_main_bus_objkey   = gs_data-anlage " Key of current object
  IMPORTING
    es_msg                = gs_msg.
macro_add_msg_to_bal_log gv_log_handle.
```

- Create your message with the `MACRO_CREATE_MSG` macro, add the message parameters with the `MACRO_ADD_MSG_PARAMETER` macro and post eventually the message to the application log with the `MACRO_ADD_MSG_TO_BAL_LOG` macro.

*** Step 5b: Write customer message based on received data**

```
IF 1 = 2. MESSAGE s001(zemma). ENDIF.
macro_create_msg 'E' 'ZEMMA' '001'
                gs_data-anlage sy-datum space space.
macro_add_msg_parameter 'ACCOUNT' gs_data-vkonto.
macro_add_msg_parameter 'VSTELLE' gs_data-vstelle.
macro_add_msg_parameter 'SPARTE' gs_data-sparte.
macro_add_msg_parameter 'ABLEINH' gs_data-ableinh.
macro_add_msg_parameter 'AKLASSE' gs_data-aklasse.
macro_add_msg_parameter 'TARIFTYP' gs_data-tariftyp.
macro_add_msg_to_bal_log gv_log_handle.
```

- Create the BPBM specific end message again based on the ID of the currently processed main business object and write it to the application log.

*** Step 5c: Create BPBM end message and write it to application log**

```
CALL METHOD gr_bpc->end
EXPORTING
    iv_main_bus_objtype = gc_main_object_type
    iv_main_bus_objkey  = gs_data-anlage " Key of current object
IMPORTING
    es_msg              = gs_msg.
macro_add_msg_to_bal_log gv_log_handle.
```

In a final step, you must persist the created application log on the database since for now the system handled all message operations only in memory. You can use the `BAL_DB_SAVE` function module to save the application log.

*** Step 6: Save application log on database**

```
CALL FUNCTION 'BAL_DB_SAVE'
EXPORTING
    i_t_log_handle = gt_log_handle.
```

The figure below shows the resulting application log and the list of clarification cases BPBM created based on your enhanced application log and the defined clarification case category as created in step 4.

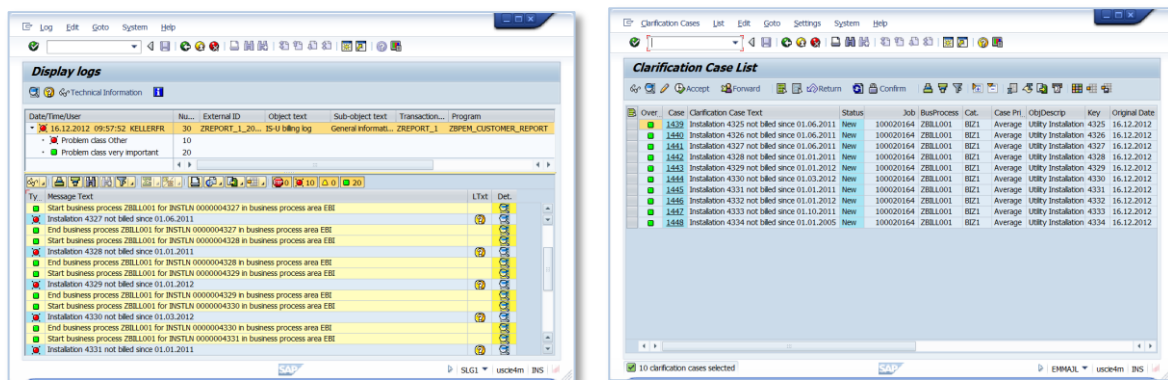


Figure 6-36 Resulting Application Log of Report

Due to the configuration, the agent can also navigate from each clarification case to the respective allocated business object that was added to the underlying message `ZEMMA (001)` you created during step 1.

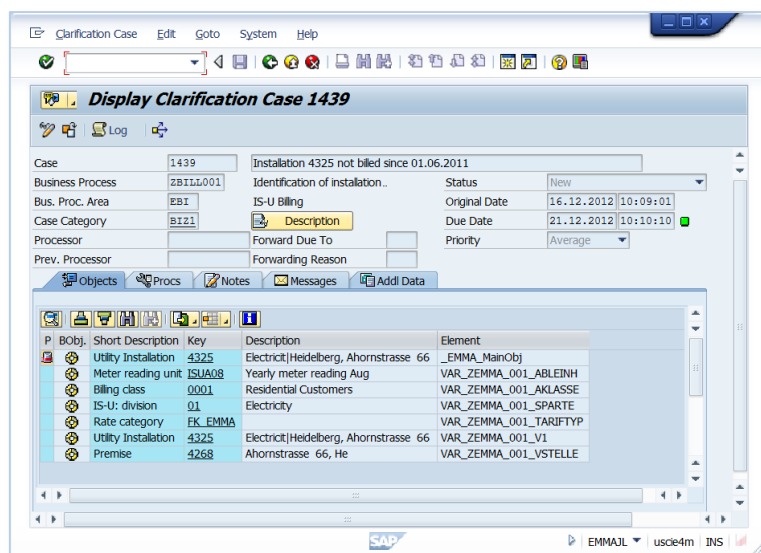


Figure 6-37 Example of a Resulting Clarification Case

6.5.2 Connection to BP EM via BAPI

As described in the introduction to this chapter, you can also create clarification cases directly in your application instead of storing messages in an application log and allow BP EM with subsequent steps to generate clarification cases. You achieve this in calling a related BAPI, such as the `BAPI_EMMA_CASE_CREATE` function module.

In a first step you need to ensure that the case creation type of your clarification case category is not set to *Automatically* but to *Manually*.

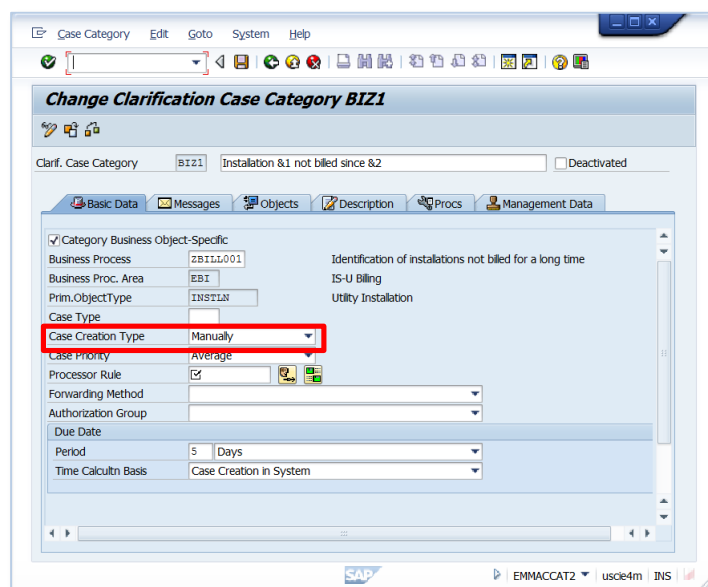


Figure 6-38 Clarification Case Category with Case Creation Type *Manually*

The following code snippet shows how to prepare the data to be transferred to the `BAPI_EMMA_CASE_CREATE` function module that eventually creates the clarification case. You find the complete ABAP code in *Appendix A2 - Creation of Clarification Case by BAPI*.

```

* Basic clarification case data
ls_case_create-ccat      = 'BIZ1'.
ls_case_create-prio      = gc_emma_case_prio_average.
ls_case_create-brgru     = space.
ls_case_create-mainobjtype = gc_main_object_type.
ls_case_create-mainobjkey = gs_data-anlage.
ls_case_create-orig_date  = sy-datum.
ls_case_create-orig_time  = sy-uzeit.

* Objects for the clarification case container
macro_add_case_object 'VAR_ZEMMA_001_VSTELLE' 'PREMISES' gs_data-vstelle.
macro_add_case_object 'VAR_ZEMMA_001_SPARTE' 'ISUSECTOR' gs_data-sparte.
macro_add_case_object 'VAR_ZEMMA_001_ABLEINH' 'READUNIT' gs_data-ableinh.
macro_add_case_object 'VAR_ZEMMA_001_AKLASSE' 'BILLINGCLS' gs_data-aklasse.
macro_add_case_object 'VAR_ZEMMA_001_TARIFTYP' 'RATECAT' gs_data-tariftyp.

* Create clarification case
CALL FUNCTION 'BAPI_EMMA_CASE_CREATE'
  EXPORTING
    case_create = ls_case_create
  IMPORTING
    case        = lv_case
  TABLES
    objects     = lt_objects
    return      = lt_return.

```

You must adhere your ABAP code to the technical definition of the clarification case category particularly with respect to the business objects to be included in the clarification case.

6.6 Connect Mass Activities to BPEM

6.6.1 General Implementation

SAP Note [144461](#) describes how to setup a mass activity (for a general description of mass activities, see chapter 2.1.13 *Mass Activity*). You find the central Customizing activity for mass activities at *Financial Accounting (New) → Prepare Mass Activities*. In addition, you must perform the following BPEM specific Customizing activities:

- Step 1:** Check if a suitable business process area is available (see chapter 3.6.1 *Define Business Process Areas for Message Management*). Create a new business process area in the customer name space if required.
- Step 2:** Check if a suitable business process code is available (see chapter 3.6.2 *Define Business Processes for Message Management*). Create a new business process code in the customer name space if required using an appropriate job analysis class to support the *New EMMA* transactions.
- Step 3:** Register the transaction code of your mass activity (see chapter 3.6.4 *Define Transactions for Message Management*). This is a mandatory Customizing activity for a new mass activity. Register the transaction code with the correct business process code and the appropriate job analysis class.
- Step 4:** Activate the related business process area if not activated yet (see chapter 3.3.2 *Activate Business Process Areas for Message Management*) or, if this Customizing activity is available on your system, activate just the business process (see chapter 3.3.3 *Activate Business Processes for Message Management*).

Commonly, SAP standard mass activities use in their interval processing function module the following standard function modules to write not only the application messages to the application log but also the BPEM specific control messages, such as the `EMMA_011` and `EMMA_012` messages per processed business object:

- FKK_AKTIV2_EMMA_REG_BUS_OBJ: Register a specific business object to link the subsequent application messages to this business object.
- FKK_AKTIV2_EMMA_UNREG_BUS_OBJ: Unregister a business object you previously registered with the FKK_AKTIV2_EMMA_REG_BUS_OBJ function module.
- FKK_AKTIV2_EMMA_LOG_MSG: Write application specific messages



The system writes the BP EM specific EMMA 011 and EMMA 012 messages to the application log only if you registered a business object with the FKK_AKTIV2_EMMA_REG_BUS_OBJ function module and you created subsequently at least one application message in the application log using the FKK_AKTIV2_EMMA_LOG_MSG function module. This means that the BP EM specific messages are only saved on the application log if at least one further message is written to the application log. As a result, the application log is kept small and clearly set out.

The code snippet below outlines the calling sequence of these function modules in an interval processing function module of a mass activity (for the complete ABAP code, see Appendix A3 – *Sample Mass Activity Interval Processing Module*). You may also check and compare the implementation of some standard interval processing function modules. You can identify these function modules via the Customizing activity *Prepare Mass Activities Customizing activity* → *General Control* → *Screen Event*. The *Activity Interval field* shows an FI-CA event number under which the interval processing function module of the selected mass activity is registered. Use the FQEVENTS transaction to search and identify the related function module by its event number.

```

FUNCTION zmass_activity_interval_sample.
*"-----

* preparations and validations before start of interval processing
[...]
IF lv_validation = co_failed.
*   write messages w/o a reference to a business object
  CALL FUNCTION 'FKK_AKTIV2_APPL_LOG_MSG'.
  EXIT.
ENDIF.

* initialize first object to be processed
lv_business_object = i_low.

* process object list
DO.

*   register business object
  CALL FUNCTION 'FKK_AKTIV2_EMMA_REG_BUS_OBJ'.

*   process selected business object
  [...]

*   check result of processing business object
  IF lv_object_status = co_failed.
*     add one or more error messages to the application log
    CALL FUNCTION 'FKK_AKTIV2_EMMA_LOG_MSG'.
  ENDIF.

*   unregister business object
  CALL FUNCTION 'FKK_AKTIV2_EMMA_UNREG_BUS_OBJ'.

*   get next business object
  lv_business_object = lv_business_object + 1.
  IF lv_business_object > i_high. EXIT. ENDIF.

ENDDO.

```

```

* do interval cleanup
* [...]
IF lv_cleanup = co_failed.
* write messages w/o a reference to a business object
  CALL FUNCTION 'FKK_AKTIV2_APPL_LOG_MSG'
ENDIF.

ENDFUNCTION.

```

The code snippet above shows that the usage of these function modules simplifies considerably the creation of application messages in application logs as created by the mass activity framework.



The SAP for Utilities specific implementation does not use the `FKK_AKTIV2_EMM*` function modules but the following auxiliary function modules:

Function Module	Description
EMMA_LOG_PROCESS_START	Start processing of a business object
EMMA_LOG_PROCESS_MSG_WRITE	Add message to the application log
EMMA_LOG_PROCESS_END	Stop processing of business object
EMMA_LOG_PROCESS_CHECK_ACTIVE	Check if a business process is activated

During the start of any mass activity (regardless if the business process area or business process code are activated), the system calls the `EMMA_MASS_ACTIVITY_RUNID` function module to process the following BP EM related initialization and preparation steps:

- Obtain the BP EM business process code related to the transaction of the mass activity as registered in the BP EM Customizing (see chapter 3.6.4 *Define Transactions for Message Management*).
- Create the BP EM job with the process code as attached to the transaction code. If the mass activity was previously executed in simulation mode, BP EM deletes the related records in the `EMMA_INT` and `EMMA_JOBRUNIDMSG` tables to wash up the BP EM related data of the simulation run. The external job number (`EMMA_HDR-EXTRUNID` field) is composed by a concatenation of the mass activity specific fields `<AKTYP>` (mass activity type that relates to the transaction), `<LAUFD>` (date on which the mass activity run is scheduled to start) and `<LAUFI>` (additional mass activity run identification) parameters.

6.6.2 Multiple Parallel Processing Objects

Mass activities use disjointed intervals of business master data identifications or transaction data keys to permit an execution of a business process by multiple concurrent jobs. Every mass activity uses own parallel processing objects to determine the ranges and size of the intervals to be processed. The parallel processing object determines eventually which database table is used for the initial interval creation. In the Customizing, you can allocate more than one parallelization object to your mass activity and the agent has to select later one of them when defining the run parameters of the mass activity.

The figure below shows the Customizing of the `FPVA` transaction (Dunning Proposal) with the two parallel processing objects `GPART` (Business Partner) and `VKONT` (Contract Account). In dependency of the chosen parallel processing object, the interval ranges are determined based on either the business partner or contract account numbers. Obviously, the interval processing function module of the mass activity must be prepared to process the resulting interval ranges that are based on either business partners or contract accounts numbers.

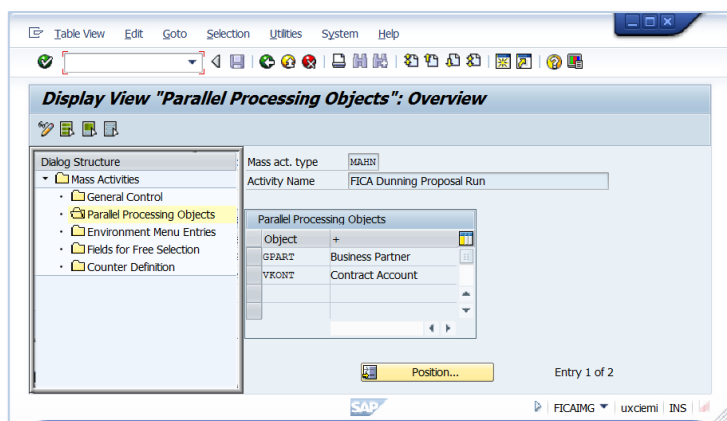


Figure 6-39 Parallel Processing Objects of the FPVA transaction (Dunning Proposal)

It is currently not possible to register a transaction code multiple times with different BPDM business process codes hence BPDM primary object types. Therefore, you must dynamically adjust the business process code in the BPDM job header that has been created automatically by the mass activity framework. SAP recommends the adjustment of the business process code in the BPDM job header if the following conditions are met:

- Your mass activity offers more than one parallel processing object to be used
- The different parallel processing objects lead to different primary object types hence business objects as indicated by the business process code you register, for example, with the EMMA 012 control message in the application log.
- The parallel processing object leads to a different primary object type (hence business objects) as it is defined in the BPDM Customizing in conjunction with the transaction code of your mass activity.

The FI-CA event 1796 allows you to change the business process code dynamically according to the selected parallel processing object and prior to the processing of any interval. You must develop an own function module and register it with the FQEVENTS transaction. The following code snippet shows a sample implementation of a function module that updates the business process code when executing a mass activity of the ZMAS mass activity type.

```

FUNCTION zfk_event_1796.
  *-----
  * "Lokale Schnittstelle:
  *   IMPORTING
  *     VALUE(I_FKKCOCC) LIKE FKKCOCC STRUCTURE FKKCOCC
  *     VALUE(I_FKK_PROT) LIKE FKKPROT STRUCTURE FKKPROT
  *     VALUE(I_BASICS) TYPE FKK_MAD_BASICS OPTIONAL
  *   TABLES
  *     T_FIMSG STRUCTURE FIMSG
  *     T_FKKCOINFO STRUCTURE FKKCOINFO
  *     T_PROT_GPART STRUCTURE FKRR_GPART OPTIONAL
  *     T_PROT_VKONT STRUCTURE FKRR_VKONT OPTIONAL
  *   EXCEPTIONS
  *     ERROR_MESSAGE
  *-----

  * local data definition
  DATA: lv_bpcode TYPE emma_bpcode.

  * Differentiate between mass activity types
  CASE i_basics-runkey-aktyp.
    WHEN 'ZMAS'.
      * determine correct BPDM business process code
      * in dependency of parallel processing object
      CASE i_basics-tech-object.
        WHEN 'GPART'.

```

```

        lv_bpcode = 'ZMASGP'.
        WHEN 'VKONT'.
        lv_bpcode = 'ZMASVK'.
    ENDCASE.
*   Update BPCODE in BPEM job header
    CALL FUNCTION 'FKK_EMMA_UPDATE_BPCODE'
        EXPORTING
            i_basics          = i_basics
            i_bpcode          = lv_bpcode
        EXCEPTIONS
            job_already_prepared = 1
            bpcode_not_found    = 2
            job_not_found       = 3
            OTHERS              = 4.
    IF sy-subrc <> 0.
*       stop mass activity
        MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
            WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4
                RAISING error_message.
    ENDIF.
    ENDCASE.
ENDFUNCTION.

```

6.7 Evaluation of Job Logs

The system creates a job log for every active background job to document the execution of the related report. In addition, the report may add additional messages to its job log. Commonly, you use the SM37 transaction to check job logs whether an irregular event occurred during job execution, for example, for a job cancellation due to a runtime error.

SAP delivers the JLOG0001 business process code to support the analysis of a job log by BPEM. The implementation of the related CL_EMMA_ANALYZE_JOB_LOG ABAP OO job analysis class identifies a job log by the information stored in the EXTRUNID field (application specific external job number) in the BPEM job header.

As previously explained, BPEM always requires a BPEM job as a starting point for the BPEM specific steps, such as BPEM job preparation and clarification case generation. However, jobs do not create BPEM jobs by themselves hence you must create them in a separate activity upon completion of the jobs if you want to use BPEM for an analysis of the related job logs. The next chapters describe the transactions to evaluate job logs in dialog (EMMAJOBLOG transaction) and in background processing (EMMAJOBLOG_DARK transaction) in more details.

6.7.1 Dialog Transaction

You can create the BPEM jobs for an analysis of job logs with the EMMAJOBLOG transaction. The figure below shows the selection screen of the EMMAJOBLOG transaction.

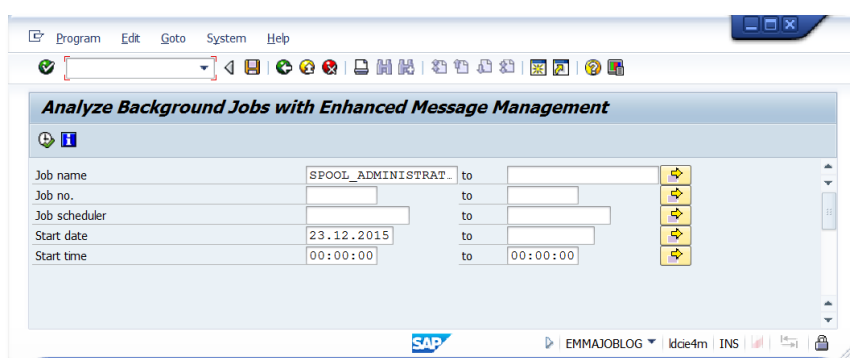


Figure 6-40 Selection Screen of EMMAJOBLOG Transaction

Push the  button to display the selected background jobs.

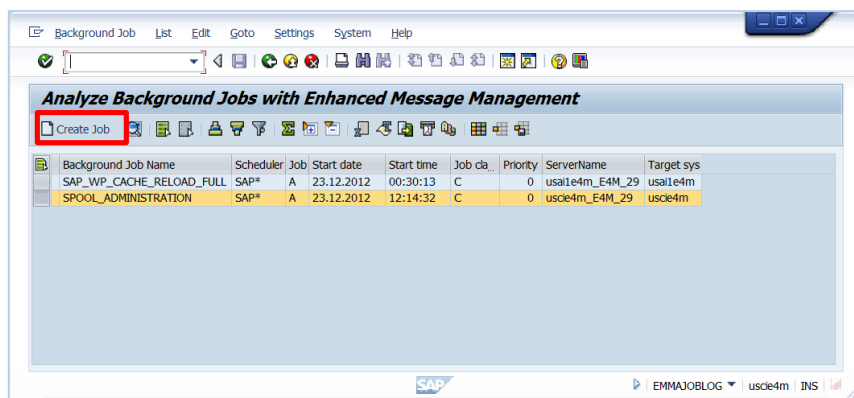


Figure 6-41 List with selected Background Job

Select one or multiple jobs and push the  button to create a BP EM job for each selected background job automatically followed by the BP EM preparation step.

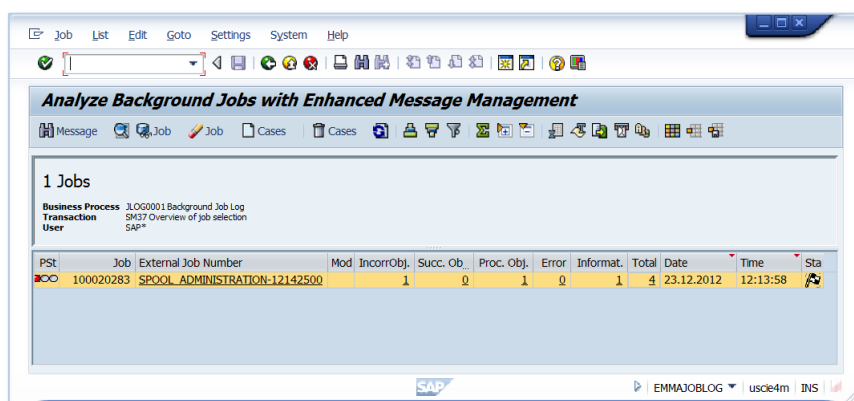


Figure 6-42 Example of a BP EM Job after Analysis of a Job Log

The BP EM job indicates with the  symbol and a non-zero number of incorrect objects that there must be at least one error message in the job log. You can display the job log after a mouse-click on the number in the *Incorrect Object* column. The system displays all messages of the job log including the error messages.

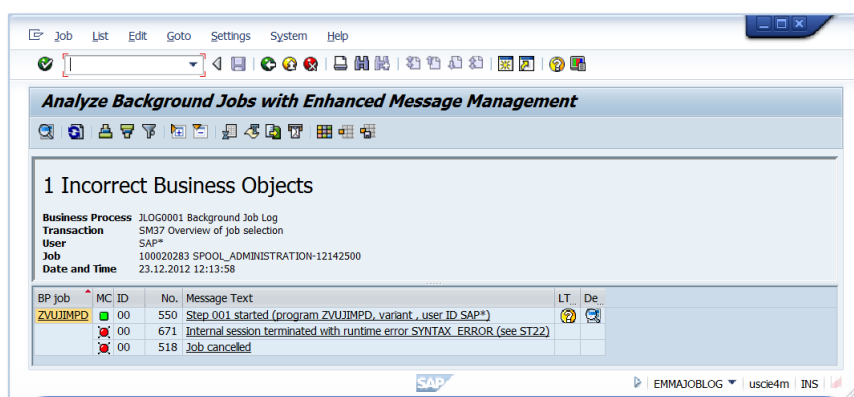


Figure 6-43 Example of a Job Log after the BP EM Preparation Step

Based on the messages in the job log, you may now generate clarification cases. Obviously, the underlying clarification case categories must be bound to the `JLOG0001` business process code and you need to allocate appropriate solution processes for a further analysis of the issue, for example, in calling the `ST22` transaction from the clarification case in case of a short dump if there is a message with message type A.

Let us have a look into the original job log and compare the entries with the job log as displayed by BPEM.

Date	Time	Message text	Message class	Message no.	Message type
23.12.2012	12:14:32	Job started	00	516	S
23.12.2012	12:14:32	Step 001 started (program ZVUJIMPD, variant , user ID SAP*)	00	550	S
23.12.2012	12:14:33	Internal session terminated with runtime error SYNTAX_ERROR (see ST22)	00	671	A
23.12.2012	12:14:33	Job canceled	00	518	A

Figure 6-44 Original Job Log

The job log starts with a `00 615` message followed by the `00 550` message. This raises the questions why the `00 615` message is not displayed in the BPEM view and why the messages are preceded by a column with the job name even the expected `EMMA 011` BPEM control message can be found in the job log that contains the business object type and business object key in its message variables. However, the first displayed column with the job name indicates that BPEM could identify a primary object, but how does this work?

You can display all messages of the job log with a simple mouse-click on the number as displayed in the *Total* column in the BPEM job list. The first `00 516` message is obviously not a so-called object-specific message because the field in the first column is empty. Only the messages starting with the second `00 550` message are related to the `ZVUJIMPD BP Job` business object as shown in the figure below.

BP Job	MC	ID	No.	Message Text	LT.	De.
	00	516		Job started		
ZVUJIMPD	00	550		Step 001 started (program ZVUJIMPD, variant , user ID SAP*)		
	00	671		Internal session terminated with runtime error SYNTAX_ERROR (see ST22)		
	00	518		Job canceled		

Figure 6-45 Job Log with all Messages after the BPEM Preparation Step

Again, the BPEM job view of the job log does not show any `EMMA 011` message that represents the standard BPEM starting control message. However, you can find the answer to all these questions in the Customizing of the `JLOG0001` business process code:

Change View "Business Process Code": Details

BusinessProcess: JLOG0001 Background Job Log

Business Process Code

Business Proc. Area: JLOG Background Job

Primary Object Type: BPJOB BP job

Job Analysis Class: CL_EMA_ANALYZE_JOB_LOG

Classes for Analysis of Application Logs (Old Version)

Job Analysis Class: ☒

Log Number Class:

Interval Info. Class:

Analysis Class:

Message for Identification of Execution Mode

Mode Message:

Execution Mode: Unknown

Messages for Identification of a Business Object

Start Message: 00 550 Step & started (program &, variant &, user ID &)

End Message:

Bus. Object Variable: Variable 2

Additional list of contents or variables

Message variable 1:

Message variable 2:

Message variable 3:

Message variable 4:

Figure 6-46 Customizing of the JLOG0001 Business Process Code

In the Customizing of the JLOG0001 business process code, the 00 550 message replaces the standard EMMA 011 message as the BPEM control message. Further, the key of the BPJOB business object, the job name, shall be found in the message variable 2. Since no further stop or other control message is defined, BPEM relates all messages after the 00 550 message to the same business object.

6.7.2 Background Processing

In addition to the EMMAJOBLOG dialog transaction there is the EMMAJOBLOG_DARK transaction you can schedule to analyze job logs in background. The figure below shows the selection screen of the EMMAJOBLOG_DARK transaction. This function is available with SAP ERP 6.0 EhP7. The EMMAJOBLOG_DARK dialog transaction allows not only the creation and preparation of BPEM jobs but also the creation of clarification cases in the same run. Figure 6-47 shows the selection screen of the SAP standard REMMALOADJOBLOG_DARK report.

Analysis of Job Log with BPEM (Background)

General Selection Criteria for Jobs

Job name: to

Job no.: to

Job scheduler: to

Start date: 01.01.2012 to

Start time: 00:00:00 to 00:00:00

Rept Name: to

Job Status

☐ Active ☒ Completed ☒ Canceled

BPEM Control Parameters

☐ Create New Job

Authorization Group:

☐ Prepare Jobs

☐ Create Clarification Cases

Figure 6-47 Selection Screen of REMMALOADJOBLOG_DARK Transaction

6.8 Special Solution Processes

A clarification case supports the resolution of a clarification case best if the clarification case offers very specific solution processes to be executed either automatically by the system or manually by an agent. Commonly, methods of the business object repository (BOR), a workflow task or a transaction of a front office process configuration, for example, a front office process FOP, are used when implementing solution processes of a clarification case category (for more details, see chapter 3.4.3.5 *Screen Processes*). You need to develop own solution processes if you cannot find an appropriate method in the BOR, workflow task or action box transaction that meets your business requirements of an appropriate solution process. This chapter provides assistance and examples of not commonly known options for a development of installation-specific solution processes.

6.8.1 Execute Reports and Transactions

You can easily implement an installation specific object type in the business object repository (BOR) that provides methods to either call a transaction or to submit a report as a solution process.

Step 1: Call the SWO1 transaction and push the  button to create a new object type.

Step 2: Enter a new object type in the customer name space and additional data. Push subsequently the  button to confirm the entered data.

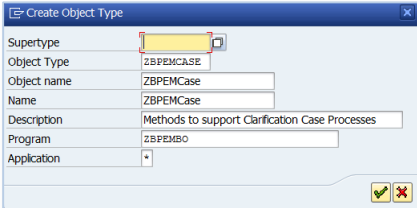


Figure 6-48 Screen to define an Object Type

Step 3: Enter Package name and subsequently a transport request

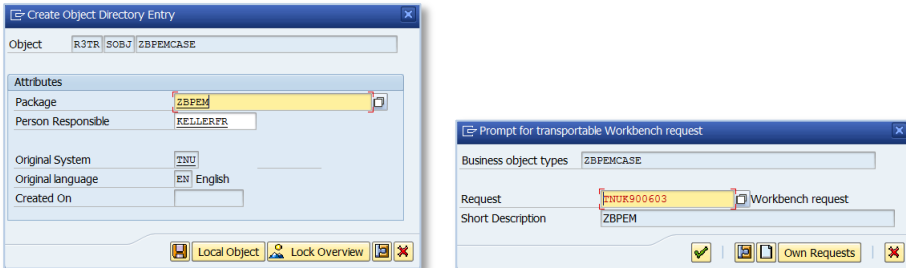


Figure 6-49 Screens to associate a Package and a Transport Request

Step 4: Place the cursor on *Methods*, push the  button and push subsequently the *No* when the system asks if you want the method to be created with a function module as template. Enter the required data to create the `CallReport` dialog method to submit a report synchronously.

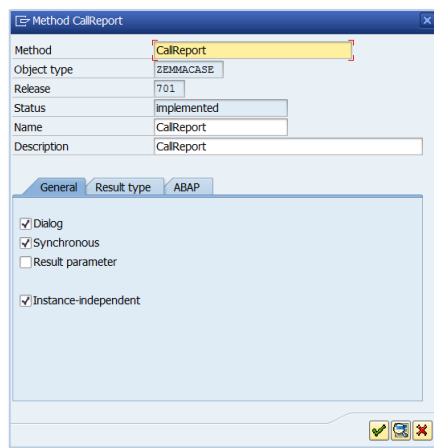


Figure 6-50 Definition of the CallReport Method

Step 5: The following parameters are required for the new CallReport Method:

Parameter	Reference Table	Reference Field	Multiline
Report	DISVARIANT	REPORT	
Parameter	RSPARAMS		X
SelectionScreen	SRTAPPL	STRT_SEL	

Figure 6-51 Required Parameters of the CallReport Method

Place the cursor on the newly created method and push the *Parameter* button to enter the parameter for your method. Push the  button to create a new parameter and enter the importing parameter details as follows (mark just the Report parameter as a mandatory parameter):

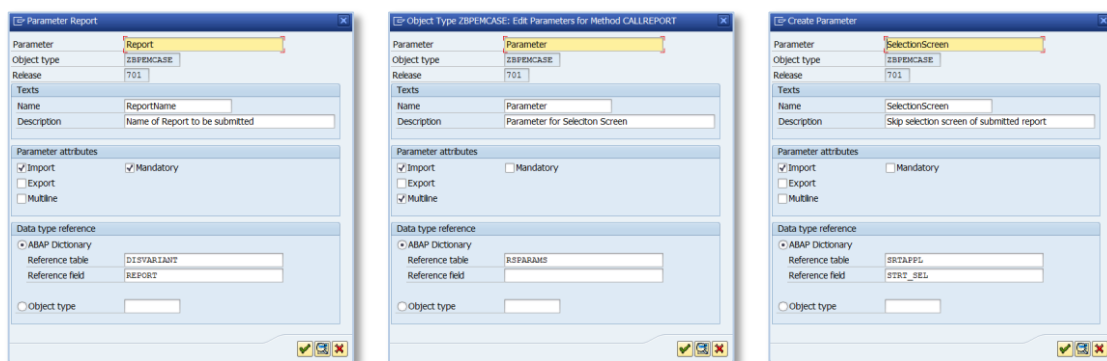


Figure 6-52 Definition of the Parameters of the CallReport Method

Step 6: Push the Program button and enter the following ABAP code, save the implementation and return to the overview screen of the object type.

```

BEGIN_METHOD callreport CHANGING container.
  DATA:
    lv_report      TYPE disvariant-report,
    lv_sel_screen  TYPE flag,
    lt_params      TYPE STANDARD TABLE OF rsparams.

  swc_get_element container 'Report' lv_report.
  swc_get_element container 'SelectionScreen' lv_sel_screen.
  swc_get_table container 'Parameter' lt_params.
  IF lv_sel_screen IS INITIAL.
    SUBMIT (lv_report)
      WITH SELECTION-TABLE lt_params
      AND RETURN.
  "#EC CI SUBMIT

```

```

ELSE.
  SUBMIT (lv_report)
    WITH SELECTION-TABLE lt_params
    VIA SELECTION-SCREEN
    AND RETURN.                                "#EC CI_SUBMIT
ENDIF.
END_METHOD.

```

Step 7: Place the cursor on *Methods*, push the  button and push subsequently the *No* button when the system asks if you want the method to be created with a function module as template. Enter the required data to create the *CallTransaction* dialog method that is executed synchronously.

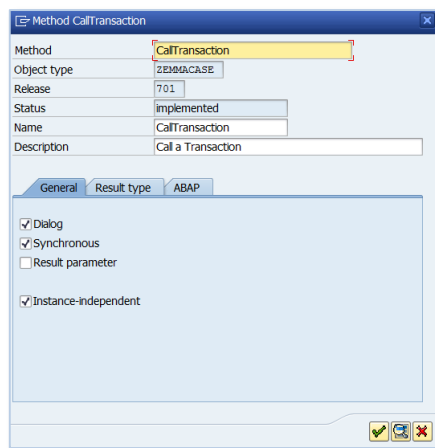
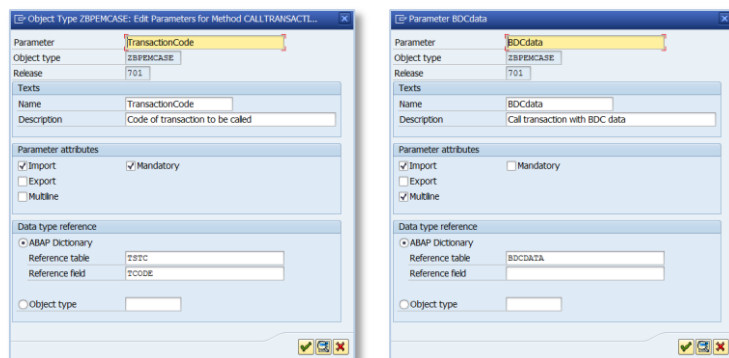


Figure 6-53 Definition of the *CallTransaction* Method

Step 8: The following parameter are required for the new *CallTransaction* Method:

Parameter	Reference Table	Reference Field	Multiline
TransactionCode	TSTC	TCODE	
BDCdata	BDCDATA		X
SetGetParameter	XU350		X
SelectionScreen	SRTAPPL	STRT_SEL	

Place the cursor on the newly created method and push the *Parameter* button to enter the parameter for your method. Push the  button to create a new parameter and enter the importing parameter details as follows (you must mark only the *TransactionCode* parameter as a mandatory parameter):



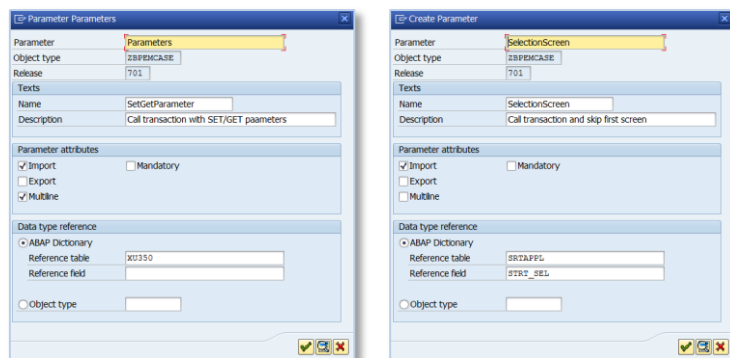


Figure 6-54 Definition of the Parameters of the CallTransaction Method

Step 9: Push the *Program* button and enter the following ABAP code. Save the implementation and return to the overview screen of the object type.

```

BEGIN_METHOD calltransaction CHANGING container.
DATA: lv_tcode      TYPE tstc-tcode,
      lt_bdcdata    LIKE bdcdata OCCURS 0,
      lt_parameter   LIKE xu350 OCCURS 0,
      ls_parameter   LIKE xu350,
      lv_selcreen   TYPE srtappl-strt_sel.

* check authorization
swc_get_element container 'TransactionCode' lv_tcode.
AUTHORITY-CHECK OBJECT 'S_TCODE'
                ID 'TCD' FIELD lv_tcode.
IF SY-SUBRC NE 0.
  EXIT_RETURN 1000 lv_selcreen space space space.
ENDIF.

* call transaction with bdcdata if provided
swc_get_table container 'BDCdata' lt_bdcdata.
IF NOT lt_bdcdata[] IS INITIAL.
  CALL TRANSACTION lv_tcode USING lt_bdcdata.
  EXIT.
ENDIF.

* call transaction with SPA/GPA parameter if provided
swc_get_table container 'Parameters' lt_parameter.
LOOP AT lt_parameter INTO ls_parameter.
  CHECK NOT ls_parameter-parid IS INITIAL.
  SET PARAMETER ID ls_parameter-parid FIELD ls_parameter-parva.
ENDLOOP.

swc_get_element container 'SelectionScreen' lv_selcreen.
IF lv_selcreen IS INITIAL.
  CALL TRANSACTION lv_tcode.
ELSE.
  CALL TRANSACTION lv_tcode AND SKIP FIRST SCREEN.
ENDIF.
END_METHOD.

```

Step 10: Push the  button to generate the object type. Then select the menu Edit -> Change Release Status first to *To Implemented* and then *To released* for both object type component and object type itself.

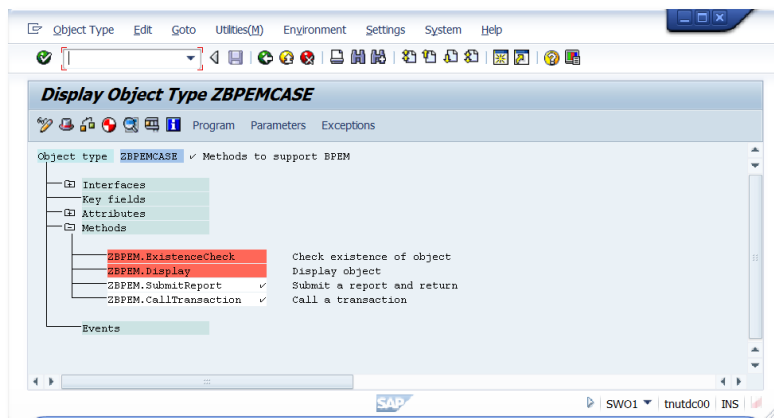


Figure 6-55 Overview of the ZBPEMCase Object Type

You can now use these methods to configure solution processes that call transactions and submit reports. The figure below shows an example how to create the binding from the clarification case container to the CallReport method.

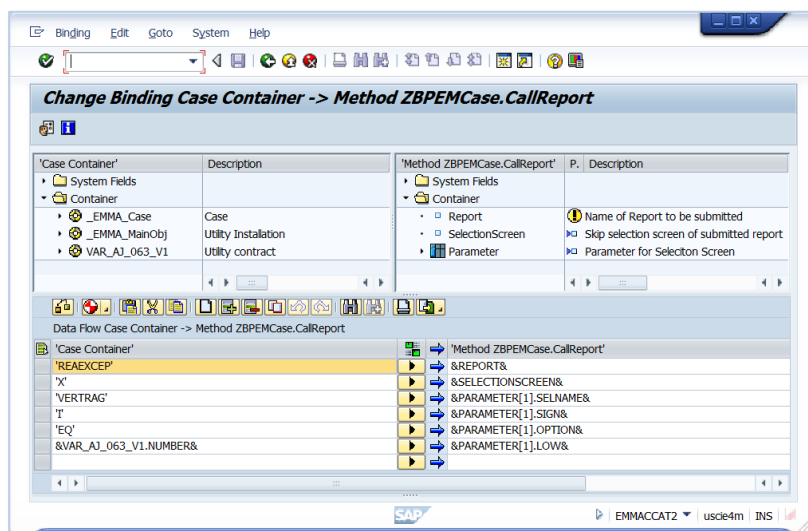


Figure 6-56 Data Flow to the CallReport Method

To identify the field names of a selection screen start the report to display the selection screen, place the cursor on a field on the selection screen and push the F1 key to display the application help on that particular field. Then push the *Technical Information* button to display some technical information about the marked field.

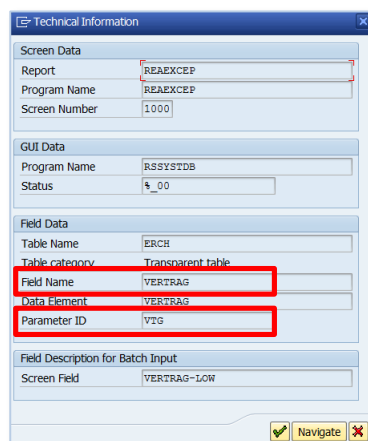


Figure 6-57 Technical Information of Selection Screen Field

You can identify the technical selection screen field name of the marked field as *Field Name* and use it to populate the `PARAMETER[index].SELNAME` field in the binding editor (in the example below the technical selection screen field name is `VERTRAG`).

Figure 6-58 shows a further example to process the `REAEXCEP` report. Here the report is called via its `EA05` transaction code and uses SPA/GPA parameters associated with the fields of the selection screen (for more details about these parameters, see chapter 2.1.16 *SPA/GPA Parameters*). You can identify the SPA/GPA parameter ID of a field on the *Technical Information* screen as *Parameter ID* and use it to populate the `PARAMETERS[index].PARID` field (in the example the used SPA/GPA parameter IDs are `KTO` for the contract account field and `VTG` for the contract field).

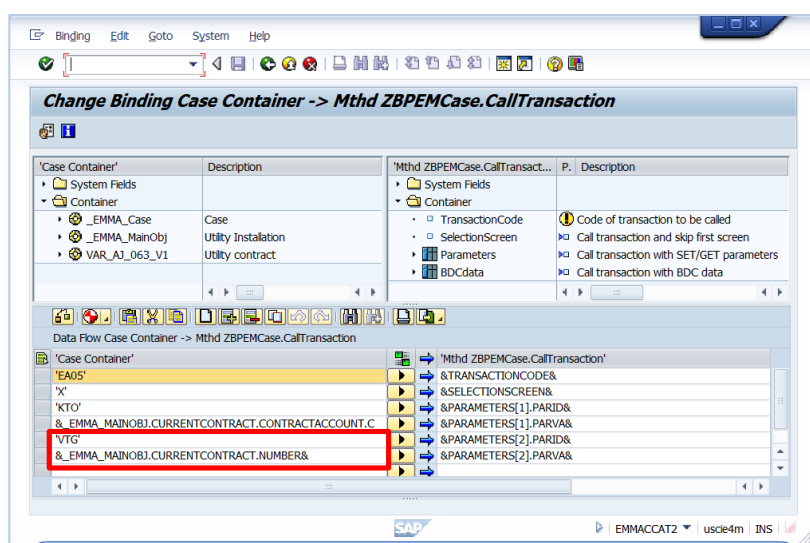


Figure 6-58 Data Flow to the `CallTransaction` Method

Commonly, you implement own methods for specific transactions and reports in BOR methods rather than using the above described generic `CallReport` and `CallTransaction` methods. The issue is that the system derives the text of a button of a solution process from associated method and you cannot (including release SAP ERP 6.0 EhP 6) overrule the automatically derived text with an own more specific text. However, you can circumvent this behavior in implementing an own method per transaction or report to be called.

Instead of implementing more and more methods you can call the generic `CallReport` and `CallTransaction` methods from a workflow task and define this workflow task as a solution process. If you configure a workflow task as a solution process, the system derives the button text from the description of the workflow task hence solving the issue with the button text without developing too many methods in the BOR but re-using the newly developed methods to submit a report or calling a transaction. The following chapter describes the approach using workflow tasks in more details.

6.8.2 Execute Workflow Tasks

You can connect workflow tasks to solution processes. For this you define and maintain a workflow task with the `PFTC_INS` and `PFTC_CHG` transactions (*SAP Menu* → *Tools* → *Business Workflow* → *Development* → *Definition Tools* → *Tasks / Task Groups* and enter the created workflow task on the call definition screen of a solution process. The following chapters describe how to develop and implement a workflow task as solution processes.

6.8.2.1 Create an executable Workflow Task based on BOR Object Type

You can create a workflow task and attach it to a solution process of a clarification case category to be executed immediately (instead of just creating a work item) when the agent pushes a button of the solution process. The following description describe how to create such a workflow task:

Step 1: Create a standard task with the `PFTC_INS`, chose `BOR Object Type` as the object category and enter the required data.



The system displays the name of the workflow task on the button of the solution process.

Figure 6-59 Basic Data Screen of a Standard Task

Step 2: Push the button to continue

Step 3: Confirm the display popup with Yes to transfer the missing elements from the object method (that are the parameters of the method) to the task container.

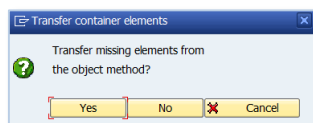


Figure 6-60 Popup to confirm Transfer of Elements to the Task Container

Step 4: Push the button to save the newly created standard task. Enter a package and transport request when requested by the system.

Figure 6-61 Screens to associate a Package and a Transport Request

Step 5: Push the  button to create the binding of the data flow to the method. You just need to push the  button to generate the binding automatically that includes all parameters of the entered method. Figure 6-62 shows the automatically generated bindings. Push the  button to return to the *Basic Data* screen of the workflow task.

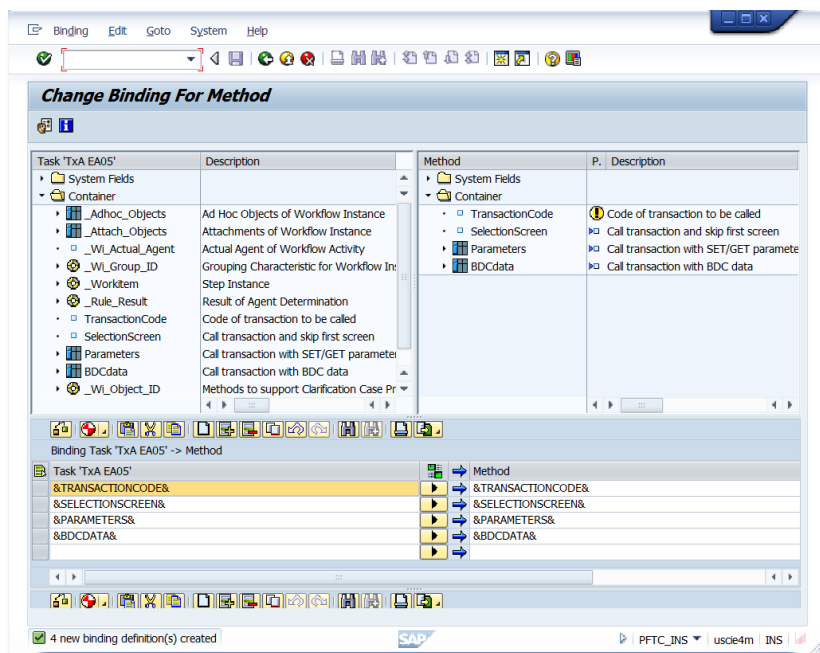


Figure 6-62 Binding Editor with automatically generated Bindings

Step 6: Change to the *Container* screen. Double-Click on the `TransactionCode` parameter. Remove the *Mandatory* Flag on the *Properties* screen and enter the default transaction code on the *Initial Value* screen. This setting suppresses the need to populate the `TransactionCode` parameter when defining the binding of a solution process of a clarification case category since a default value is provided by the workflow task itself (see step 10).

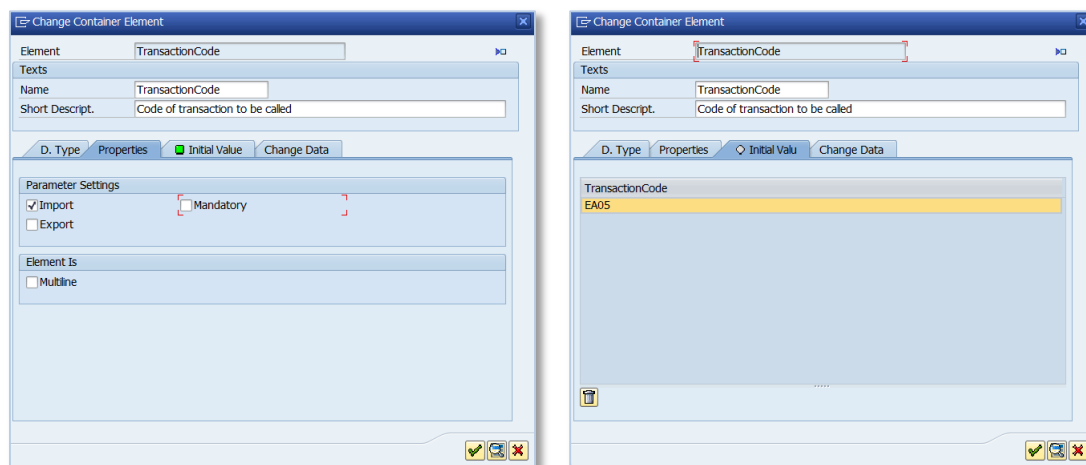


Figure 6-63 Default Value for the `TransactionCode` Parameter

Step 7: Choose *Additional Data* → *Classification* → *Create* and select *General Task* to allow a direct execution of the workflow task instead of creating a workflow item. This means also that all agents can execute the workflow task. Push the  button to save the classification data.

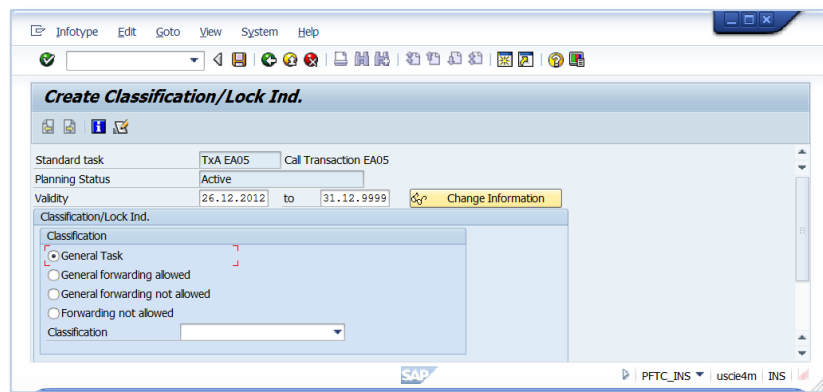


Figure 6-64 Classification of Workflow Task as *General Task*

Step 8: Push the  button to save the workflow task.

Step 9: Call the EMMACCAT2 transaction to maintain a clarification case category and create a solution process using the newly created workflow task.

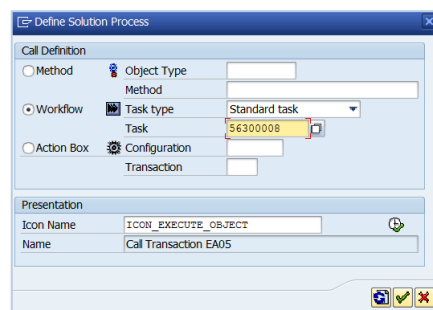


Figure 6-65 Definition of a Solution Process calling the newly created Workflow Task

Step 10: Enter the binding. You do not need to provide a value for the `TransactionCode` parameter due to, first, the previous removal of the *Mandatory* flag of the `TransactionCode` element of the container of the workflow task and, second, the provided default transaction code in the workflow task.

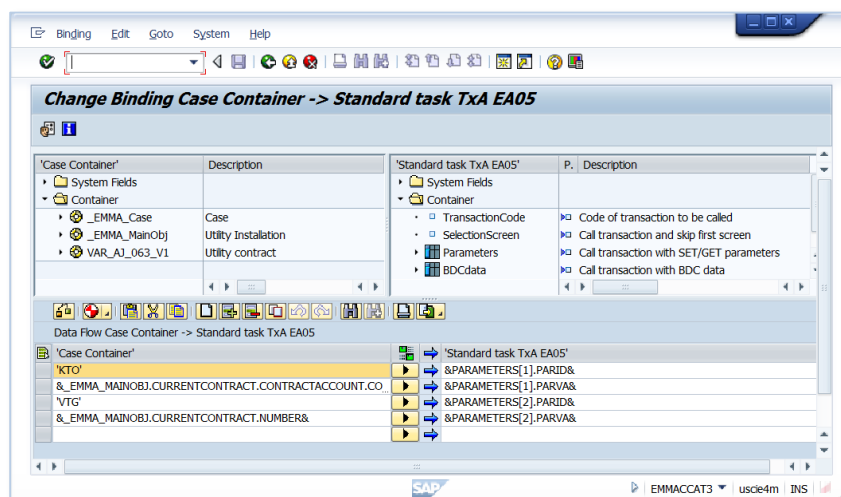


Figure 6-66 Binding of a Solution Process calling the newly created Workflow Task

Step 11: Push the  button to return to the clarification case category definition.

Step 12: Push the  button to save the clarification case category.

The clarification case category is now ready to be used.

6.8.2.2 Create an executable Workflow Task based on an ABAP OO Method

Figure 6-59 shows the *Basic Data* Screen of a standard task that processes the `CALLTRANSACTION` method of the `ZBPMECase` BOR object type. You can also implement an own ABAP OO method and allocate it to a workflow task.

For technical reasons, there are some restrictions in connection with BPME solution processes compared to the usage of a workflow task as part of a workflow:

 You can only create a binding to the clarification case container and save an exporting parameter of your ABAP OO method in the clarification case container if it is a discrete field (no structure or internal table) with a alphanumeric data type and the total length of the field does not exceed 70 characters (see also chapter 3.4.3.3 *Screen Objects*).

An ABAP OO class is only suitable for a workflow if it implements the `IF_WORKFLOW` ABAP OO interface. The steps to create such an ABAP OO class and an executable method to be used as a solution process are as follows:

Step 1: Call the `SE24` transaction, enter the name of the new ABAP OO class. Then push the  *Create* button and subsequently the  *Save* button.

Step 2: Change to the *Interfaces* screen of the class builder, enter the `IF_WORKFLOW` interface. Then push the *Enter* button to allow the system to add automatically the two `BI_OBJECT` and `BI_PERSISTENT` sub-interfaces and the related methods to your ABAP OO class.

Step 3: Change to the *Method* screen and define and implement your method(s) to be executed as solution processes as public static methods.

Step 4: Push the  *Activate* button to activate the ABAP OO class.

Step 5: Create a workflow standard task and enter the name of your ABAP OO method.

Figure 6-67 Basic Data Screen of a Standard Task to process a ABAP OO Method

Step 6: Push the *Enter* button and confirm subsequently the question if the missing elements should be transferred to the object method.

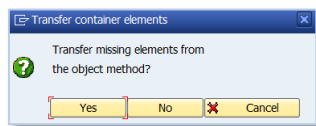


Figure 6-68 Popup to confirm Transfer of Elements to the Task Container

Step 7: Push the  button, allocate a transport request and save the workflow task.

Step 8: Mark the *Synchronous Object Method* flag and push the  button to create the binding of the data flow to the method. You just need to push the  button to generate the binding automatically that includes all parameters of the entered method. Push the  button to return to the *Basic Data* screen of the workflow task.

Step 9: Choose *Additional Data* → *Classification* → *Create* and select *General Task* to allow a direct execution of the workflow task without any specific agent assignment. This means that all agents can execute the workflow task. Push the  button to save the classification (see Figure 6-64).

Step 10: Push the  button to eventually save the finalized workflow task.

Step 11: Link the newly created workflow task to a solution process (see previous chapter).



You need to implement the methods of the `IF_WORKFLOW` interface to maintain the instantiation and the instances by yourself if you want to use the workflow task not only as a solution process of a clarification case but also when using an ABAP OO class within the SAP WebFlow Engine. The SAP WebFlow Engine handles all objects generically and objects have to be saved in the event of a context change. Therefore, it is required to convert object references such that they can be saved persistently. Conversely, you have to be able to generate the corresponding instance of an ABAP class from the persistently saved key. A detailed description of the implementation details is beyond the scope of this document thus refer to the workflow documentation related to the `IF_WORKFLOW` interface.

6.9 Add Business Objects to a Clarification Case dynamically

Commonly, the clarification case container contains only the business objects and additional data objects required to either execute the attached solution processes or display additional information of the business exception data environment.

The keys or values of these objects are usually determined during the clarification case creation process by means of the allocated data flow. However, you can determine and set the keys or values of these objects by yourself in implementing the `COMPLETE_CASE` method of the `EMMA_CASE` Classic BAdI and return the object information in the `CT_OBJECTS` parameter. In addition, with the same method you can add further business objects to the clarification case container if appropriate. For further information, see chapter 3.4.3.3 *Screen Objects*.

6.10 Dynamic Clarification Case Text

Commonly, the clarification text as displayed in any clarification case list is derived from the static text as defined in the affiliated clarification case category. However, you may want, for example, to allocate the respective triggering message as the clarification case text dynamically if the clarification case was created based on a sweeper clarification case category.

A possible solution is to define a variable with reference to a container element in the clarification case text field, and assign the desired text to the container element in the `COMPLETE_CASE` BAdI. The system will then replace the variable in the clarification case text field by its actual content while creating the clarification case.

The basis for the following example implementation is the implementation as described in chapter 6.3.3 *General Sweeper Clarification Case Categories*.

Step 1: Change to the *Object* screen of the clarification case category. Then push the  *Create* button and enter an element name and an ABAP reference to a text field. Subsequently push the  *Save* button to store the newly created container element.

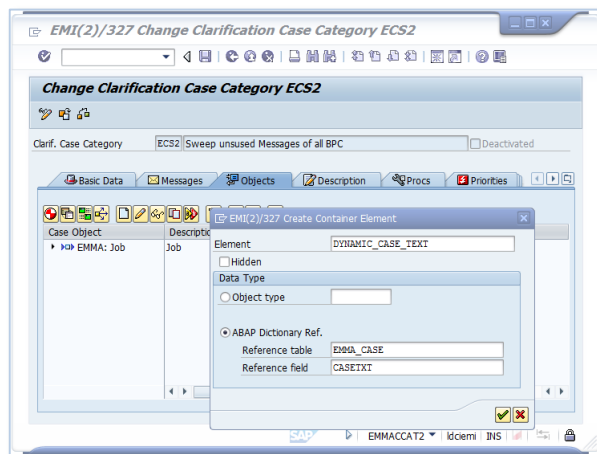


Figure 6-69 Creation of a Container Element with Reference to a Text Field

Step 2: Push the  button and create a simple data flow for the newly created container element.

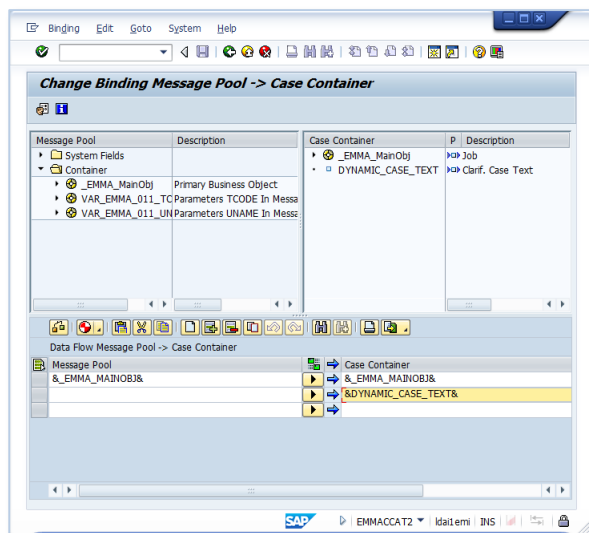


Figure 6-70 Creation of a simple Data Flow for the Container Element

Step 3: Change to the *Description* screen of the clarification case category and enter a text completed by the variable name as reference to the created container element.

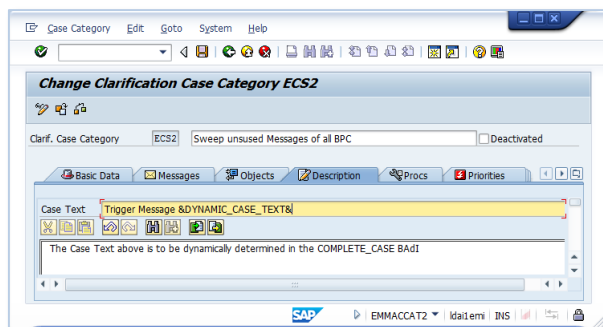


Figure 6-71 Clarification Case Text becomes dynamic due to the included Variable

Step 4: Create an implementation of the `COMPLETE_CASE` method of the `EMMA_CASE` BAdI. You may use the code snippet below as a basis for your own implementation.

```
METHOD if_ex_emma_case~complete_case.

  FIELD-SYMBOLS: <fs_obj> TYPE LINE OF emma_cobj_t,
                 <fs_msg> TYPE LINE OF emma_msg_link_tab.

  CASE is_case-ccat.

  *   sweeper clarification case categories
      WHEN 'ECS2'.

  *   Identify text element in clarification case container
      READ TABLE ct_objects ASSIGNING <fs_obj>
        WITH KEY celemname = 'DYNAMIC_CASE_TEXT'.

  IF sy-subrc = 0.
  *   Assign the first found error message id/number
  *   to text element in clarification case container
      LOOP AT it_msg_link ASSIGNING <fs_msg>.
        <fs_obj>-id = <fs_msg>-msgid && '(' && <fs_msg>-msgno && ')'.
        EXIT.
      ENDLOOP.
  ENDIF.

  ENDCASE.

END_METHOD.
```

The following figure shows the dynamically determined clarification case text *Trigger Message AJ(063)* based on the implementation above.

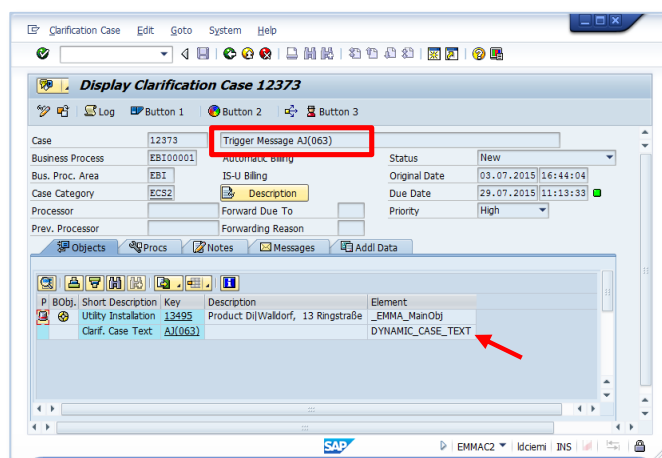


Figure 6-72 Dynamically changed Clarification Case Text

In a final step, you need to hide the text container element from the agent. You can do so in implementing the `ADJUST_ALVOBJ_OUTPUT` method (for more details, see chapter 5.8.2.3 *BAdI Definition BADI_EMMA_CASE_TRANSACTION*) or, if you have at least ERP 6.0 EhP7 installed, mark the *Hidden* flag on the container element definition screen (for more details, see chapter 10.2 *Hide Objects of Clarification Case Container from Agent*).

You may use the following sample implementation of the `ADJUST_ALVOBJ_OUTPUT` method of the `BADI_EMMA_CASE_TRANSACTION` BAdI as a basis for your own implementation. This code snippet removes the text container element dynamically from the output list.

```
METHOD if_badi_emma_case_transaction~adjust_alvobj_output.

    FIELD-SYMBOLS <fs_obj_list>    TYPE LINE OF emma_ctxn_alvobjs_t.
    FIELD-SYMBOLS <fs_obj>        TYPE LINE OF emma_ctxn_alvobjs_t.

    CASE is_case-ccat.

        *   sweeper clarification case categories
            WHEN 'ECS2'.

            *   Hide the text container element
                READ TABLE it_obj ASSIGNING <fs_obj_list>
                    WITH KEY celemname = 'DYNAMIC_CASE_TEXT'.

                IF sy-subrc = 0.
                    *   Remove object from output list
                        DELETE ct_obj_change WHERE consnr = <fs_obj_list>-consnr.
                    ENDIF.

            ENDCASE.

    END_METHOD.
```

6.11 Usage of Action Box of Customer Interaction Center CIC

When defining a solution process, you have the option to either, first, call a Business Object Repository BOR method, second, execute a workflow task or, third, launch a front office process FOP. The launch of the FOP is done by means of a transaction of an action box configuration as defined in the Customer Interaction Center CIC framework (more general information how to define solution processes, see chapter 3.4.3.5 *Screen Processes*).

Previously, there was only the option to process a front office process when defining a solution process. The options to process a solution process by means of a method of the BOR or a workflow task as well as the flexible definition of data flows have been added in later releases. The development of further improvements concerning the usage of FOPs ceased since the introduction of the two further options to process a solution process. Of course, you may still use FOPs and reuse your repository of FOPs. However, SAP encourages the usage of methods of the BOR and workflow tasks to take full advantage of the flexibility of these options.

The usage of FOPs is subject to the following restrictions:

- It is not possible to define a specific data flow from the clarification case container to the FOP; the importing parameter must be available in the clarification case container.
- You cannot call the same FOP with different input parameters
- The usage of search modules and related search object types is not supported.
- The primary object of the clarification case is automatically made available to the data flow of the FOP as the desk object.

If there is a problem with the data flow from the clarification case container to the selected FOP for a particular solution process the warning message **EMMA 163 Data flow Case Container->Action Box Call is incorrect** is displayed followed by specific messages describing the identified issues. The following list describes the most common error messages when using FOP as a solution process and possible solutions:

- **SWF_BND_001 073** *The object types '[BO.object type]' (source) and '[BO.object type]' (target) are not compatible.* As said above, the primary object of the clarification case is automatically made available to the data flow of the FOP as the desk object. The message indicates that the related primary object type, as defined in the business process code, is not compatible to the used desk object in the data flow of the FOP.

What to do: Check the data flow of the FOP where the desk object is displayed as `&<DESKOBJ>object-type&` (for example, the desk object contract account, object type ACCOUNT, is displayed as `&<DESKOBJ>ACCOUNT&`). Then compare the desk object to the primary object type attached to the clarification case category (see chapter 3.4.3.1 *Screen Basic Data*) and replace the data source in the data flow by an element that is defined in the clarification case container. If required, create a new object in the clarification case container of your clarification case completed by the respective data flow definition. Subsequently, enter the newly defined object of the clarification case container as the data source in the data flow of your FOP as `&object-type&` (for example, as `&ACCOUNT&`.) If you use the same FOP in your front office, you should consider creating a new FOP to serve the needs of a BPEM solution process.

- **SWF_BND_001 035** *Container element '<element name>' does not exist:* The system could not identify an object in the clarification case container with the same element name as used in the data flow of the FOP.

What to do: Check the data flow of the FOP and compare the entered data sources with the element names of the objects in the clarification case container. If required, create a new object in the clarification case container of your clarification case completed by the respective data flow definition. Be aware that the element name of the newly created object matches with the element name as defined in the data flow of the FOP. It is not possible to define an object in the clarification case container with an element that contains the characters `<` and `>`. If the element name of the data source in the data flow of the FOP relates to the search module of the FOP, for example, `<SEARCHSELECT>` or `<SELECT>`, remove the search object from the FOP definition.

- **EMMA 164** *Only primary object attributes can be used in Action Box calls.* You want to use an attribute of a business object in the data flow of the FOP other than the desk object. However, as data source you can only use the complete case objects or attributes of the primary business object (since the primary business object is transferred as the desk object).

What to do: Check the data flow of the FOP and compare the entered data sources with the data type definition of the objects in the clarification case container. If required, adjust the data flow of the FOP.

- **EMMA 165** *Use desktop object <DESKOBJ>attribute instead of object-type.attribute.* You want to use an attribute of the primary object as a data source in the data flow of the FOP but the used syntax is not recognized by the system during the clarification case maintenance.

What to do: Check the data flow of the FOP and use the syntax `<DESKOBJ>attribute` instead of `object-type.attribute` as the data source in the data flow of the FOP.

- **SWF_BND_001 092** *Conversion From '[BO.]' to '[BO.object-type]' Can Cause Data-Dependent Errors*. This message is issued as a warning message and indicates that the assignment between the two data types requires a conversion at runtime when calling the FOP. This conversion can – depending on the values involved - lead to conversion errors during runtime.

What to do: Check the data flow of the FOP and compare the entered data sources with the data type definition of the objects in the clarification case container. If required, adjust the data type of the object in the clarification case container.

- **EMMA 308** *Use other process categories instead of action box calls*. The system issues this message as a warning message to suggest the usage of methods of the BOR or workflow tasks instead of FOPs.

What to do: Check if you can replace the FOP by a method of the BOR or by a workflow task.

7 Business Information Warehouse BW

7.1 Introduction

There is a BPEM specific BI (Business Intelligence, refers to the front-end tools used to present information to the agent in terms of reports, analytics and dashboards) content available to extract and transfer BPEM related KPIs to a business information warehouse. An analysis of the statistics of reported business exceptions and related clarification cases as well as the processing time per clarification case supports your continuous attempts to improve the implemented business processes.

Obviously, all attempts for improvement should aim for a reduction of the volume of the most recurring business exceptions not only to reduce the attached costs for a resolution of the business exceptions but also to increase the customer satisfaction due to fewer interrupted or even broken business processes and a reduced overall processing time.

This chapter describes the available general BI content (process statistics for clarification cases) of BPEM as well as the Utilities Industry specific BI content focusing on the DataSources and extractors on the ERP system. Of course, you can create an own BI content for further business processes other than SAP delivers for the Utilities Industry.

There is an online documentation available regarding the integration of BPEM with SAP BW at the SAP Help Portal <http://help.sap.com/> → SAP NetWeaver → SAP NetWeaver Business Information Warehouse → Select BI Content (select your SAP NetWeaver BI Content version) → Application Help available in English or German → Select BI Content → Industry Solutions → Utilities → Process Management.

7.2 Process Statistics for Clarification Cases

This `0FCEMMA_CASE` InfoSource describes the interface for the BPEM specific data. The DataSource of the same name (`0FCEMMA_CASE`) provides the basic data of a clarification case to BW after a new clarification case was created or a clarification case was changed. This DataSource is available for all BPEM enabled business processes. You must activate the business process code, such as `EBI0001`, or the business process area, such as billing (`EBI`), for message and BW process management. For more details, see chapter 3.3.2 *Activate Business Process Areas for Message Management*.

During initialization of the delta update, the system transfers the status of the clarification case. The system does not write performed changes to a clarification case to a delta queue but to the `BIW_EMMA_CASE` table instead, which is also used as the extraction structure. The delta load extracts the data from the `BIW_EMMA_CASE` table.

The extractor supports the following update modes:

- **F:** Transfer of all requested data
- **D:** Transfer of delta since last request
- **C:** Initialization of delta transfer
- **S:** Initialization of delta transfer without data update

7.3 Utilities Industries - Process Statistics

7.3.1 Process Statistics *Non-Clarification* and *Clarification*

The utilities industry specific BI content includes the process statistics for the business processes billing, invoicing, move-in and move-out, budget billing plan and meter reading. There are two DataSources available for each business process:

- **Process statistics *Non-Clarification*:** Data for the related process specific DataSource is extracted when no clarification case was created based on the given message.
- **Process statistics *Clarification*:** Data for the related process specific DataSource is extracted when a clarification case was created based on the error message. The process specific extraction structure is the same as for the process statistics *Non-Clarification* but includes additionally details about the created clarification case.

 You must activate the business process area of the related business process code, such as billing (EBI), for message and BW process management. In addition, you must specify an extractor class the system shall use to determine the DataSource. For more details, see chapter 3.3.2 *Activate Business Process Areas for Message Management*.

BPEM extracts the process statistics upon completion of the following BPEM process steps:

- **BPEM job preparation step:** The system transfers all successfully processed business objects to BW as process statistics *Non-Clarification* if requested (for related Customizing, see next chapter).
- **Clarification case generation step:** The system updates the process statistics *Clarification* if a clarification case was created for a particular business object. In contrary, the process statistics *Non-Clarification* is updated if for the same business object no clarification case was created.

 The DataSources for process statistics *Non-Clarification* are commonly used to identify business exceptions for which no clarification case was created due to a lack of a matching clarification case category. Since the introduction of the `EMMACAS2` business process code, you can ensure that for every business exception a clarification case is created (for details about the `EMMACAS2` business process code, see chapter 5.3.1.3 *Special Business Process Codes* `EMMACAS1` and `EMMACAS2`).

7.3.2 Special Customizing

There are various extraction specific Customizing settings that affect the extraction to BW.

7.3.2.1 Extract Clarification Case Messages

The ***Extract Clarification Case Messages*** indicator (see chapter 3.3.2 *Activate Business Process Areas for Message Management*) controls to what extent messages are extracted by the clarification case DataSource. You can choose one of the following options:

- Mark this indicator if you want to extract also the messages that have led to a clarification case. For each message, one data record is transferred to the Business Information Warehouse (BW). This setting should be used with caution (for performance reasons) for business process areas that mainly create their application logs in mass activities.
- Do not mark the indicator if you do not want to extract also the messages that have led to a clarification case. Only the BPEM start message (`EMMA 011` message) is extracted for each business process code and business object completed by the general clarification case information.



The system never extracts EMMA 012 messages to BW. However, the process end timestamp is supplied with the data of the EMMA 011 message.



If the process end timestamp of a business process is missing, no process statistics is extracted to BW. This can, for example, happen if the agent cancels the process without the system being able to log the end of the process.

7.3.2.2 Scope of extracted Success Messages

SAP delivers a positive list of success messages, information messages and warning messages that restricts the scope of the messages to be extracted to BW. You can extend this positive list (for more information, see chapter 3.7.5 *Define Success Messages for the Extraction (Customer)*). The BPEM specific EMMA 011 message is always extracted for each processed business object. However, the message type of an EMMA 011 message is converted into an error message if no message of the positive list was extracted to BW. Otherwise, the message type of the EMMA 011 message remains a success message.

The clarification case information is supplied with the message that triggered the clarification case. If the extraction suppresses this message via Customizing, or the messages uses different clarification case categories to trigger different clarifications, this can result in records being extracted, which contain only clarification case information and no message information. The key is extracted to BW for each case, for which an object is triggered.



Error messages are always extracted to BW if they are not suppressed during the BPEM job preparation step (for more details how to suppress message, see chapter 3.3.4 *Suppression of Messages for Log Preparation*) or if they have been explicitly excluded from extraction (see previous chapter).

7.3.2.3 Clarification Cases for successfully processed Business Objects

The **SucBusProc** indicator (see chapter 3.3.3 *Activate Business Processes for Message Management*) controls if you want to create clarification cases also for successfully processed business objects. Mark this flag if you want to create clarification cases also for successfully processed business objects. Consequently, a significantly higher volume of clarification cases will have to be extracted to BW, In addition, process statistics *Clarification* is affected.

7.3.2.4 Additional Control for Message Management

There is a special Customizing for some selected utilities specific business processes to suppress the creation of BPEM specific messages, such as the EMMA 011 and EMMA 012 messages, for successfully processed business objects. The objective is to reduce the size of the related application logs and, subsequently, the performance of a later analysis of the application log. The activity is available at SAP Utilities → Tools → Monitoring of Mass Runs → Suppress EMMA Messages if No Error Occur.

In this activity, you can define for which business process codes success messages shall be suppressed hence not to be saved in the process application log.



You must activate the business process area of the related business process code, such as billing (EBI), for message management. For more details, see chapter 3.3.2 *Activate Business Process Areas for Message Management*.

Mark the *Mass Processes* flag if you want the business process code related mass activity to document in the application log only business objects that could not be processed due to a exception hence it failed during processing. Mark the *Single Process* flag if you want the same but a restriction related only to dialog transactions. These flags also suppress the creation of

BP EM specific messages, such as the EMMA 011 and EMMA 012 messages, for successfully processed business objects.

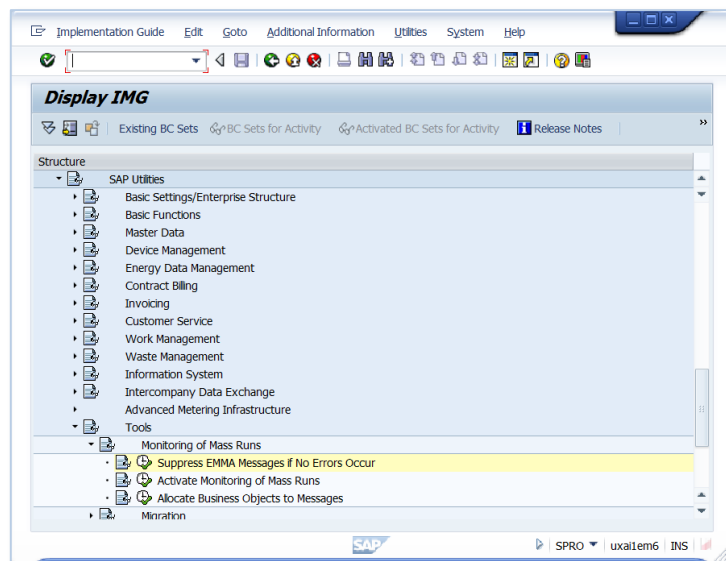


Figure 7-1 Suppress EMMA Messages if No Errors Occur Customizing Activity



You can also implement the R567 FI-CA event to overrule the Customizing settings depending on the transaction code.

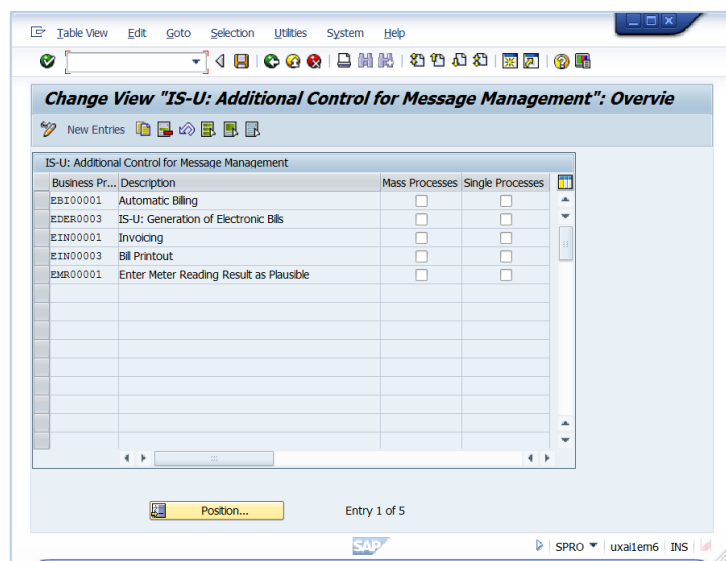


Figure 7-2 IS-U: Additional Control for Message Management



If you mark for the EIN00001 business process code (Invoicing) the *Mass Activities* flag but not the *Single Processes* flag, the application log remains unchanged for dialog transactions, such as the EA19 or EASIBI transactions. However, during the execution of mass activities, such as the EA10, EA26, or EA10_COLL transactions, not only the success message EB 020 *Invoicing unit completed successfully* but also the BP EM specific messages EMMA 011 and EMMA 012 are suppressed.



The option to suppress success messages is only available for the displayed business process codes. In addition, the option to suppress success messages affects the process statistics *Non-Clarification* since successfully processed business objects are not registered anymore in the application log.

7.3.3 Technical Details

This chapter describes some further technical details about the usage of the available DataSources including details how you can complete the data in the extraction structure before sending it to BW. The figure below shows the DataSources specific for the Utilities Industry (RSA2 transaction):

Business Process	Process statistics Non-Clarification	Process statistics Clarification
Meter Reading	0UC_CM_NCL_MR	0UC_CM_CL_MR
Billing	0UC_CM_NCL_BILL01	0UC_CM_CL_BILL01
Invoicing	0UC_CM_NCL_INV01	0UC_CM_CL_INV01
Budget Billing Plan	0UC_CM_NCL_INV02	0UC_CM_CL_INV02
Import Profile Values	0UC_CM_NCL_EDM01	0UC_CM_CL_EDM01
Move-In/Move-Out	0UC_CM_NCL_CS01	0UC_CM_CL_CS01
Rate Maintenance	0UC_CM_NCL_CS02	0UC_CM_CL_CS02
Waste Disposal Order	0UC_CM_NCL_WDO	0UC_CM_CL_WDO

Figure 7-3 Utilities Industries specific DataSources

The next figure shows the extract structures of the abovementioned DataSources.

Business Process	Extract Structure Non-Clarification	Extract Structure Clarification
Meter Reading	BIW_ISU_KPI_MR_NCL	BIW_ISU_KPI_MR_CL
Billing	BIW_ISU_KPI_BI_NCL	BIW_ISU_KPI_BI_CL
Invoicing	BIW_ISU_KPI_IN01_NCL	BIW_ISU_KPI_IN01_CL
Budget Billing Plan	BIW_ISU_KPI_IN02_NCL	BIW_ISU_KPI_IN02_CL
Import Profile Values	BIW_ISU_KPI_EDM01_NCL	BIW_ISU_KPI_EDM01_CL
Move-In/Move-Out	BIW_ISU_NCL_CS01	BIW_ISU_CL_CS01
Rate Maintenance	BIW_ISU_NCL_CS02	BIW_ISU_CL_CS02
Waste Disposal Order	BIW_ISUWA_KPI_WDO_NCL	BIW_ISUWA_KPI_WDO_CL

Figure 7-4 Utilities Industries specific Extract Structures

These DataSources are extracted by means of ABAP OO classes that are delivered by SAP. During the data collection of the related extract structure, the attached BADIs are processed.

Business Process	BPA	ABAP OO Extract Class	Business AddIn
Meter Reading	EMR	CL_MR_KPI_BW	ISU_MR_KPI_PROCESS
Billing	EBI	CL_BI_KPI_BW	ISU_BI_KPI_PROCESS
Invoicing	EEIN	CL_IN_KPI_BW	ISU_INV_KPI_PROCESS
Budget Billing Plan	EBB	CL_IN_KPI_BW	ISU_INV_KPI_PROCESS
Import Profile Values	EEDM	CL_EDM_KPI_BW	ISU_EDM_KPI_PROCESS
Move-In/Move-Out	ECS	CL_CS_KPI_BW	ISU_BW_KPI_CS
Rate Maintenance	ECS	CL_CS_KPI_BW	ISU_BW_KPI_CS
Waste Disposal Order	EEWA	CL_WASTE_ORDER_KPI_BW	ISU_WA_ORDER_KPI_PROCESS

Figure 7-5 Utilities Industries specific Extractors and BADIs

You can implement these BADIs to complete or change data in the extract structure instead of implementing the `RSAP0001 Customer function calls in the service API` enhancement as explained in the next chapter. The next chapter describes how to enhance the extract structure by installation-specific fields.

7.4 Enhancement of the Extraction Structure

You can enhance any extract structure by adding installation specific fields. You can execute all required steps with the `RSA6` transaction:

Step 1: Call `RSA6` transaction and select the DataSource you want to enhance.

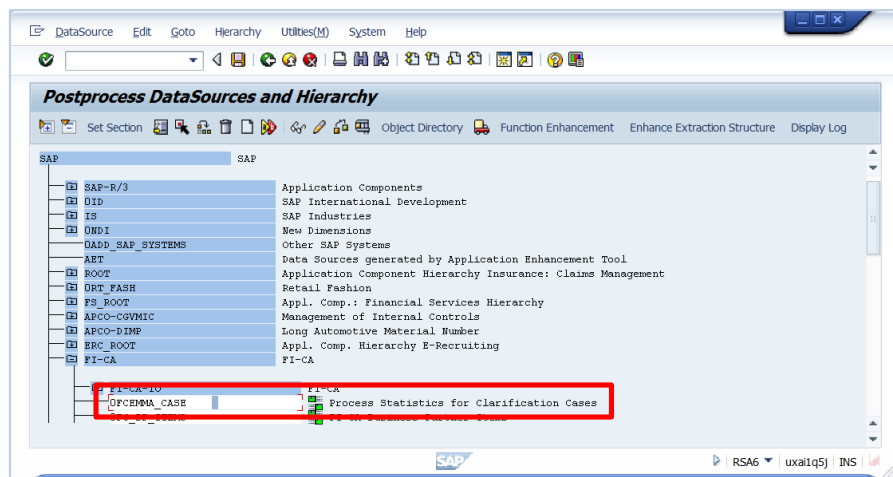


Figure 7-6 Postprocess DataSources Screen of `RSA6` transaction

Step 2: Push the **Enhance Extraction Structure** button and enter the name of the structure to be created as an append structure to the delivered extraction structure.

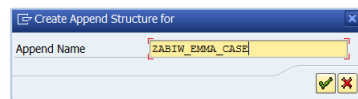


Figure 7-7 Screen to enter the Append Structure Name

Step 3: Enter the fields you want to add to the extract structure and eventually activate the newly created append structure. If requested, enter an installation specific package and a transport request.

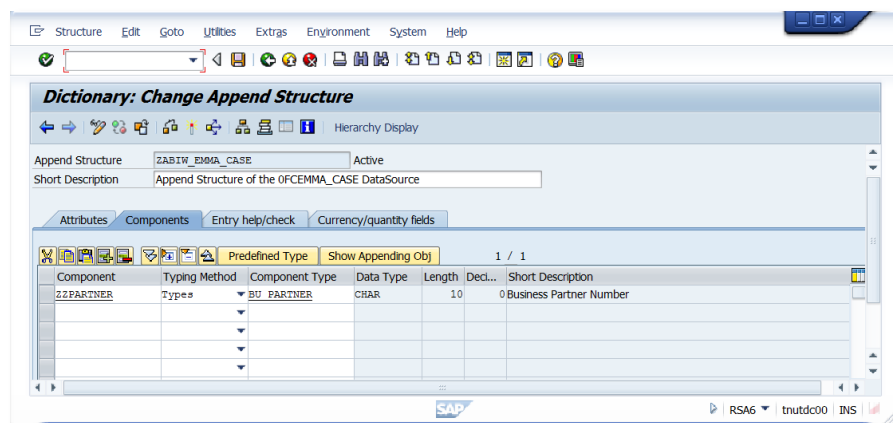


Figure 7-8 Screen to enter the Append Structure Name

Step 4: Return to the *Postprocess DataSources and Hierarchy* screen of the RSA6 transaction. Then push the *Function Enhancement* button.

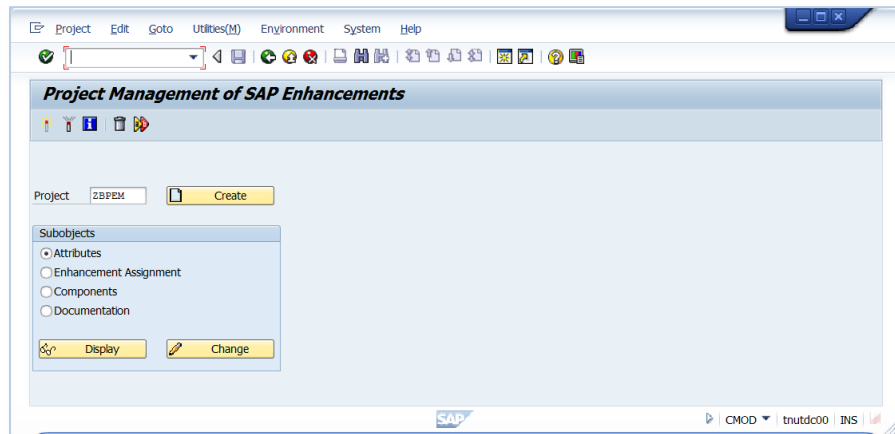


Figure 7-9 Selection Screen of the CMOD Transaction

Step 5: Push the  button to create a new customer enhancement project.

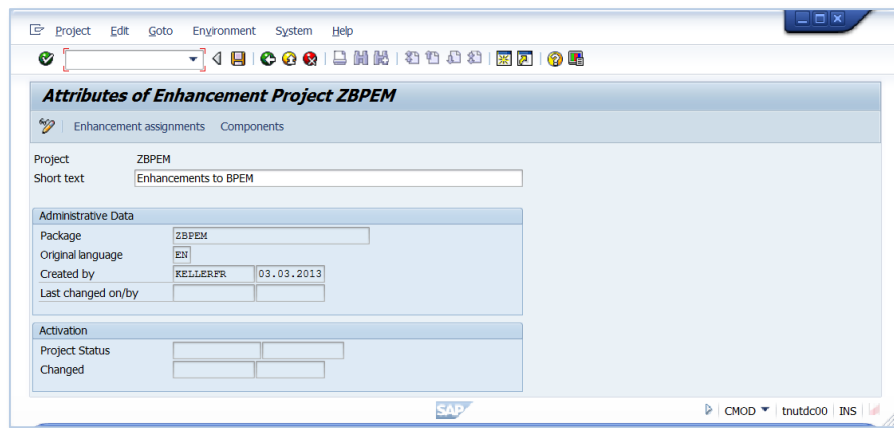


Figure 7-10 Attributes Screen of the CMOD Transaction

Step 6: Push the  button and enter a package and a transport request for the newly created customer enhancement project if requested.

Step 7: Push the *Enhancement Assignments* button and add the RSAP0001 *Customer function calls in the service API* enhancement.

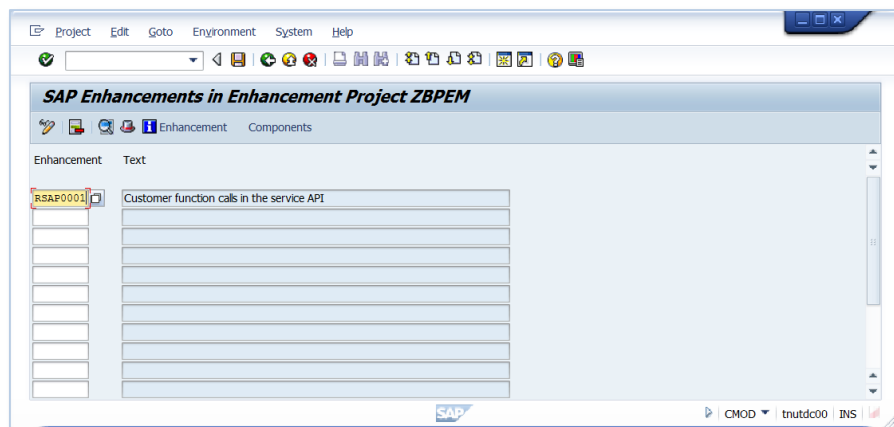


Figure 7-11 Enhancement Screen in Enhancement Project

Step 8: Push the *Components* button to display the components of the enhancements you assigned to your enhancement project, such as the RSAP0001 enhancement.

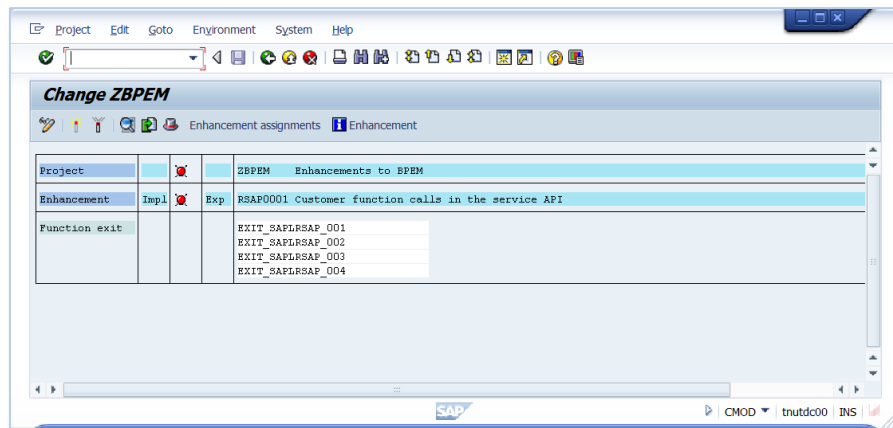


Figure 7-12 Overview Enhancements assigned to the Enhancement Project

Step 9: Double-click on the EXIT_SAPLRSAP_001 enhancement you need to implement because clarification case related data is considered as transaction data. The double-click takes you to the EXIT_SAPLRSAP_001 function module.

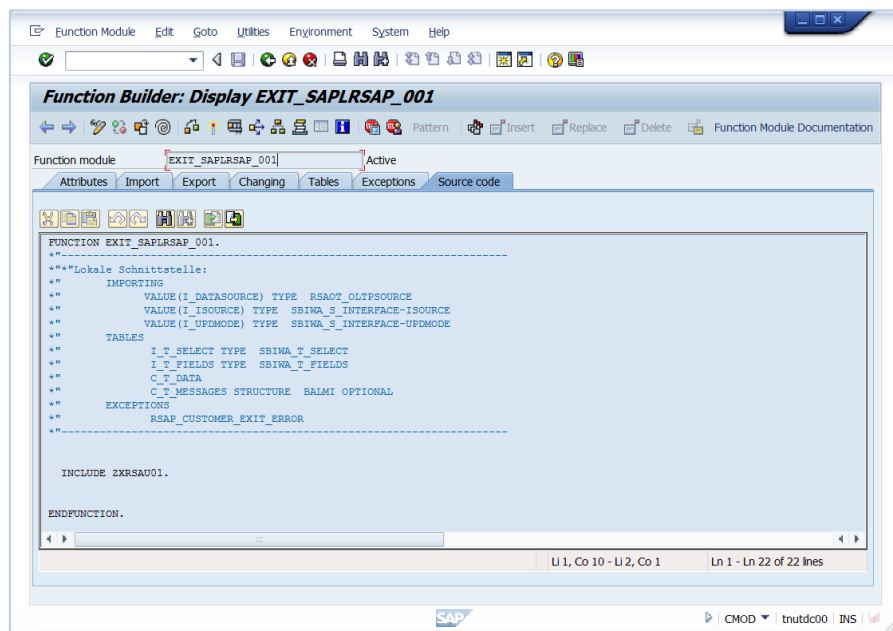


Figure 7-13 Function Module with ZXRSAU01 Include to implement the required Enhancement

Step 10: Double-Click on the ZXRSAU01 include names. This takes you through the process of creating, implementing and activating the include. Push the *Function Module Documentation* button to display the documentation and understand the implementation steps to correctly populate the fields you added to the extraction structure. The documentation of the function module includes an ABAP code example you may use as a basis for your own development.

Step 11: Return to the *Enhancement Overview* screen. Then push the  button to activate the enhancement.

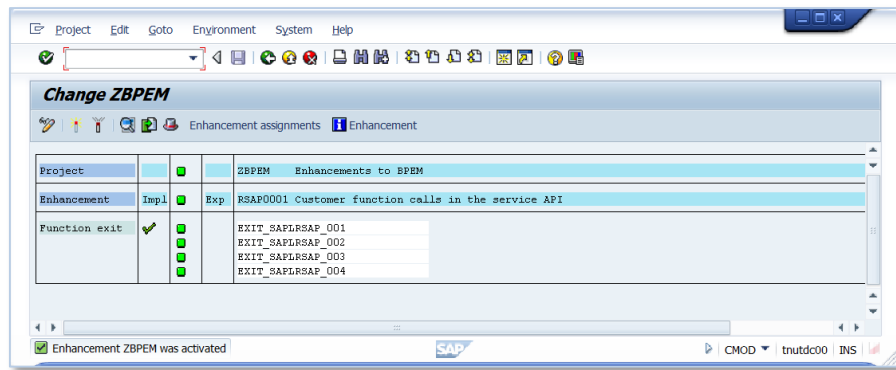


Figure 7-14 Overview of activated Enhancements assigned to the Enhancement Project

Step 12: Return to the *Postprocess DataSources and Hierarchy* screen of the RSA6 transaction and test the DataSource extraction in pushing the  button.

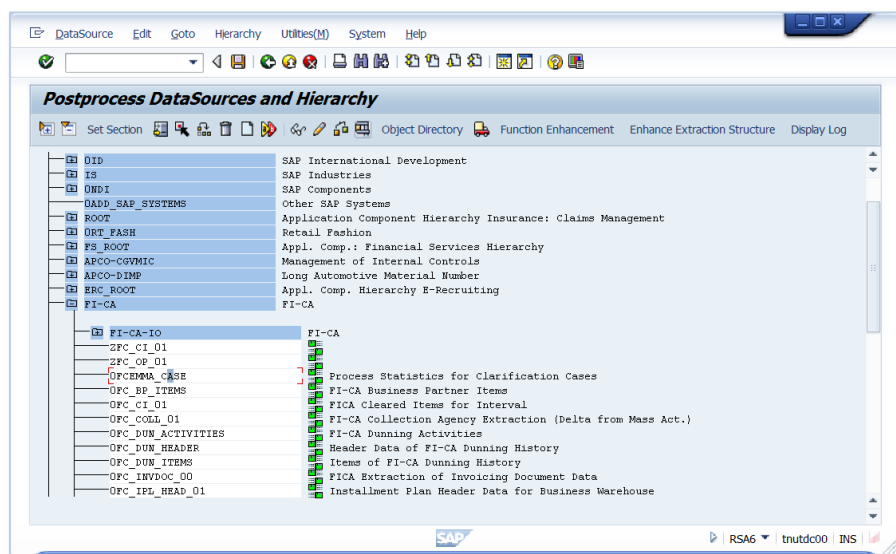


Figure 7-15 Postprocess DataSources Screen of RSA6 transaction

8 Data Archiving

Data archiving removes mass data that the system no longer needs online. Data in the database can only be archived by means of archiving objects, which describe the data structure and context. Archiving objects are used to write documents to archive files, which can be stored on other media. The SAP data archiving concept is based on the Archive Development Kit (ADK). ADK-based archiving is carried out with the help of the Archive Administration, transaction SARA.

The archiving procedure is separated into three main steps:

- **Creation of archive files:** In the write phase, the data to be archived is written sequentially into newly created archive files.
- **Delete from the database:** The delete program reads the data from the archive files and deletes it from the database subsequently.
- **Storage of archive files:** You can now move the newly created archive files to a storage system or copy it to a tape. You can trigger the file move to an external storage system both manually or automatically.

Commonly, you schedule archiving programs periodically for a background processing. However, they can also run in online mode. The following documentation describes the SAP delivered BPEM specific archiving objects `EMMA_JOB` (archiving of BPEM jobs) and `EMMA_CASE` (archiving of clarification cases).

8.1 EMMA_JOB Archiving Object

The `EMMA_JOB` archiving object archives all data related to a BPEM job:

Database Table	Description
EMMA_HDR	BPEM job header
EMMA_INT	Interval Statistics
EMMA_JOBRUNIDMSG	Application Messages

Figure 8-1 Tables from which BPEM jobs are archived

The figure below shows the selection screen to select the BPEM jobs to be archived:

Figure 8-2 Selection Screen of the `EMMA_JOB` Archiving Report



You can archive a BPEM job only after having archived all related clarification cases. The reason for this dependency is that BPEM selects clarification cases for a possible execution of automatic solution processes by selecting the respective BPEM jobs.

8.2 EMMA_CASE Archiving Object

The EMMA_CASE archiving object archives all data related to a clarification case:

Database Table	Description
BALINDEX	Application Log: INDX tables
BALDAT	Application Log: Log data
BALHDR	Application log: log header
CDCLS	Cluster structure for change documents
CDHDR	Change document header
CDPOS_STR	Additional change document table
CDPOS_UID	Additional change document table
EMMA_CACTOR	Clarification Case Processor
EMMA_CASE	Clarification Case
EMMA_CMSG_LINK	Case Messages
EMMA_COBJECT	Objects and Field Values for Case
EMMA_CSOLP	Case Solution Path

Figure 8-3 Tables from which clarification cases are archived

The figure below shows the selection screen to select the clarification cases to be archived:

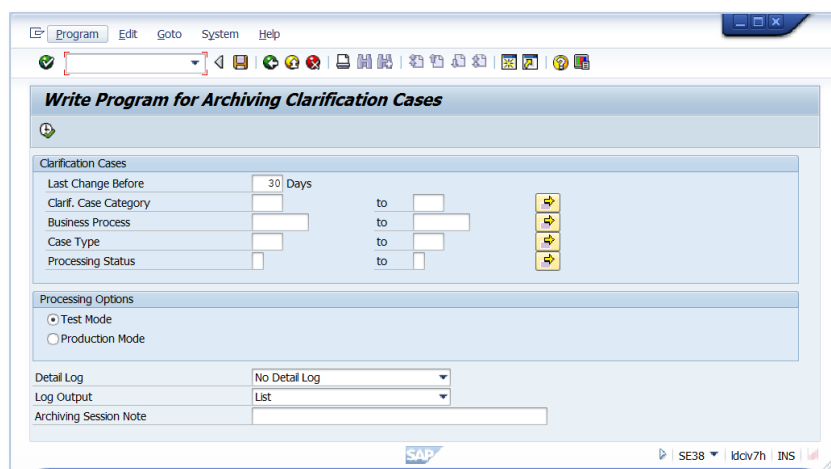


Figure 8-4 Selection Screen of the EMMA_CASE Archiving Report

The system selects by default only clarification cases for archiving that are in the status *Completed* or *Confirmed*. This presumes that you do not enter any clarification case processing status on the selection screen.



However, the system archives all clarification cases that match with an explicitly entered clarification case processing status. This means that you can archive clarification cases even if they have not yet reached the status *Completed* or *Confirmed*. This permits to archive clarification cases that are unlikely to be resolved after a long retention period.

9 BPEM in the Utilities Industry

9.1 Clarification Cases in the SAP CRM IC WebClient (Utilities)

You can view current and historical clarification cases in the SAP CRM Interaction Center (IC) WebClient related to the confirmed business partner or confirmed premise. You can also create clarification cases for a follow-up in the back-office. The following functions are available with SAP CRM 7.0 EhP3.

9.1.1 Customizing

9.1.1.1 Customizing CRM System

Your business role should use the `INR_BPEMCASE` function profile to activate the clarification case functionality in the *Interaction Record* view. By default, this function profile is assigned to the `UTIL_IC_REG` and `UTIL_IC` role.

You maintain the required Customizing for the creation of clarification cases at *Customer Relationship Management* → *Industry-Specific Solutions* → *Utilities Industry* → *Settings for User Interfaces* → *Clarification Cases* → *Maintain Parameters for Clarification Case Creation*.

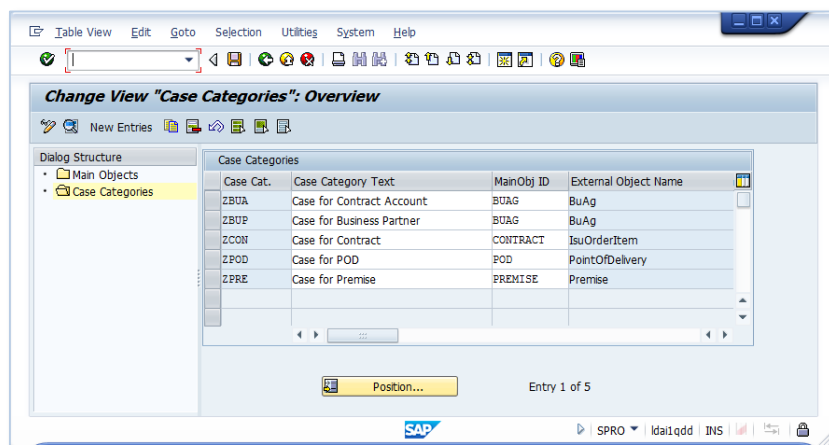


Figure 9-1 Maintain Parameters for Clarification Case Creation

In the **Main Objects** section, you define all possible main objects that can be used within a clarification case category. The SAP standard delivery includes the main objects *Business Partner*, *Business Agreement*, *Contract*, *Point of Delivery* and *Premise*. The following attributes are available:

- **Main Object ID:** Descriptive identifier of the main object.
- **Object Short Description:** Text which will be displayed in the Interaction Center to describe the main object
- **Component Name:** The name of the component in which the BOL Object exists.
- **External Object Name:** The name of the CRM BOL object representing the main object. The External BOL Object is required for the mapping of the BOL Object to a BOR Object in certain scenarios.
- **ERP BOR Object:** The name of the ERP BOR Object representing the main object. The BOR Object is required for the successful creation of a clarification case on ERP.

In the **Case Categories** section, you enter all possible case categories released for a creation of clarification cases in the IC WebClient on the CRM system. The following attributes are available:

- **Case Category:** The name of the case category. It is important to ensure that the case category exists in the ERP system; otherwise, the creation will fail.
- **Case Category Text:** Text which will be displayed in the Interaction Center to describe the case category
- **Main Object ID:** The identifier of a main object as defined under the Main Objects view in the Clarification Case customizing. This main object must be the same as the main object assigned in the ERP customizing for clarification cases.

The remaining fields are not editable; they are prefilled using the *Main Object ID* Customizing values from the *Main Objects* view in the Clarification Case customizing. You can re-use the same Main Object ID by multiple clarification case categories.



You must maintain the clarification case categories used for the case creation in the IC WebClient on both CRM and ERP systems. It is essential to keep the Customizing on both systems in sync since no cross-system validation exists.



SAP recommends the usage of a trusted RFC connection between the CRM and ERP systems to ensure that the user names are the same in both systems when creating and maintaining clarification cases. If a trusted RFC connection to the ERP system is not available, the system uses the standard RFC connection user and the fields such as *Created By* and the notes header will be populated using the respective user name.

9.1.1.2 Customizing ERP System

You can maintain the required Customizing on the ERP system at *Maintain Clarification Case Categories Customizing node* (for more details, see chapter 3.4.3 *Create and Maintain Clarification Case Categories*)



You must mark the clarification case categories to be used on the CRM system as of clarification case creation type *Manually*.

The screenshot shows the 'Change Clarification Case Category ZPOD' window in SAP. The 'Clarif. Case Category' is 'ZPOD' and the 'Clarification Case for POD' is 'Clarification Case for POD'. The 'Case Creation Type' is set to 'Manually' and is highlighted with a red box. Other fields include 'Business Process' (ZISUPOD), 'Business Proc. Area' (ECS), 'Prim.ObjectType' (ISUPOD), 'Case Priority' (High), 'Processor Rule' (checked), 'Forwarding Method', 'Authorization Group', 'Due Date' (10 Days), and 'Time Calculn Basis' (Appearance of Case).

Figure 9-2 Example of a Clarification Case Category as of Creation Type *Manually*

9.1.2 Views in IC WebClient

9.1.2.1 View Open Clarification Cases for a Confirmed Business Partner

The system displays open clarification cases for a confirmed business partner in the *Account Overview* tree whereas clarification cases in status *New* or *In Process* are considered as open. When you expand the *Clarification Cases* folder in the *Account Overview* tree, the system displays all open clarification cases for the confirmed business partner and associated business agreements, contracts, points of delivery or premises of the confirmed business partner. You can click on the clarification case description link to navigate to the clarification case detailed view.



Use the ISU_BP_BPEM_OBJ_QUERY BAdI definition of the ISU_BP_BPEM_OBJ_QUERY enhancement spot on the ERP system to influence the display of open clarification cases for the confirmed business partner in the *Account Overview* tree. Using this BAdI, you can influence the display of clarification cases based on the clarification case status and main object associated with the clarification case, specify the desired period for the display of cases and select the maximum number of clarification cases to be displayed. See the Customizing for SAP Utilities at *Customer Service* → *CRM Integration* → *BAdIs for Accessing Objects* → *Account Overview: BAdI for Clarification Cases*.

The screenshot shows the SAP IC WebClient interface for the account 'Mr. Philipp Schmidt / 1260509'. The 'Clarification Cases' tab is active, displaying a table of open clarification cases. The table has columns for Object, Additional Information, and ID. The highlighted entries are:

Object	Additional Information	ID
Mr. Philipp Schmidt Mr. Philipp Schmidt	Philipp Schmidt D-69190 WALLDORF	6010639
Related Accounts		
Vertragskonto	Vertragskonto	1260509
Premise		2000002505
POD 01 Electricity	DE12345_UWEDPP02	XD7KAw7KpYkzX2SaDm
DUMMY 01 Electricity	Electricity 29.04.2014 31.12.9999	28717
Clarification Cases	2 Open	
Case for POD	0000005032	In Processing Period: High
Case for POD	0000005031	Overdue: High

Figure 9-3 Clarification Cases in Account Overview in IC WebClient

9.1.2.2 View Open Clarification Cases for a Confirmed Premise

The system displays open clarification cases for a confirmed premise in the *Premise Overview* tree whereas clarification cases in status *New* or *In Process* are considered as open. When you expand the *Clarification Cases* folder in the *Premise Overview* tree, the system displays all open clarification cases for the current premise and associated connection object, business agreements, contracts, points of delivery or devices of the confirmed premise.

You can also select the *Clarification Cases* folder in the *Premise Overview* tree to display all open clarification cases for the current premise and associated objects in a table on the right-hand side. When you select a clarification case from within this table, the system displays the details of the case underneath. Alternatively, you can expand the folder in the tree and select a single clarification case; the details of the case will be displayed on the right-hand side.



Use the `ISU_PREMISE_OBJ_QUERY` BAdI definition of the `ISU_PREMISE_OBJ_QUERY` enhancement spot on the ERP system to influence how the system displays open clarification cases in the Premise Overview tree. Using this BAdI, you can influence the display of clarification cases based on its status and the associated main object, specify the desired period for the clarification cases selection and the maximum number of clarification cases to be displayed. For more details, see the Customizing for SAP Utilities at *Customer Service* → *CRM Integration* → *BAdIs for Accessing Objects* → *Premise Overview: BAdI for Premise Objects*.

9.1.2.3 Create Clarification Cases

Agents can create new clarification cases in the Interaction Records work center. There is no connection between the Interaction Record and the creation of clarification cases; this means that it is possible to create several clarification cases regardless of the Interaction Record process. At minimum, the agent must enter the Clarification Case Category, Main Object and (Main Object) Key attributes in order to create a clarification case successfully. Clarification cases are located on the ERP system so the information provided in the Interaction Center view is sent from the IC WebClient to the ERP backend system for the actual creation.

Action	Due	Status	Object Type	Key	Case ID	Description	Created On	Created By	Priority	Case Type	Case Category
			Point of Deliv...	[XQ]TKAwTk...	0000005032	Case for POD	03.07.2015	CRMOLTP	High		ZPOD - Case ...
			Point of Deliv...	[XQ]TKAwTk...	0000005031	Case for POD	03.07.2015	CRMOLTP	High		ZPOD - Case ...

Figure 9-4 Clarification Case Creation in IC WebClient



The *Main Object/Key* drop-down lists are populated using the logic specified within the `CRM_ISU_BPEN_MAINOBJKEY` BAdI definition of the `CRM_ISU_BPEN_ENHSPOT` enhancement spot in CRM. A value from the *Main Object* drop down list must be selected in order to trigger the logic, which pre-fills the *Main Object/Key* drop down lists. You can tailor this behavior using the provided BAdI. You can find additional information as well as a detailed explanation of the default behavior in the related BAdI documentation.

There are three options for a creation of clarification cases:

- **In the Interaction Record work center:** Click on the Clarification Cases link to create a clarification case. The Main Object/Key drop down list is prefilled once the Main Object value has been set using the logic available within the BAdI implementation. The Notes text area will automatically have a time stamp inserted along with the current user name once the clarification case is saved.
- **By using the one click menu actions on the Premise, Business Agreement, Point of Delivery, Contract and Business Partner object symbols:** Clicking on these symbols in the Premise Overview tree or the Business Data Environment tree (in the Account Overview) and choosing the Create Clarification Case option allows you to navigate to the Interaction Record work center. On this page, the Main Object/Key and Clarification Case Category fields are prefilled with values that correspond to the selected object.
- **By clicking the *New* pushbutton** on the Clarification Cases History table in the Interaction Record work center or the Account Overview page.

9.1.2.4 Edit Clarification Cases

You can edit open clarification cases by adding notes in the *Notes* section of the clarification case detail view. Click on the *Edit* symbol in the case detail view to make the case editable. You can also navigate to the case detail view in *Edit* mode, by clicking the *Edit* symbol in the *Action* column of the *Clarification Case* tables.



Only clarification cases in status *New* and *In Process* can be edited and only the clarification case notes are editable in the detail view.

9.1.2.5 Historical Clarification Cases

A table of historical clarification cases for the confirmed business partner is available in the Interaction Record work center and the Account Overview work center (see Figure 9-4). You can also see a table with historical clarification cases for a confirmed premise in the Premise Overview. From these tables, you can navigate to each clarification case in display mode. You can also click the *Edit* symbol in the *Action* column to enter notes of the clarification case.

Display Historical Clarification Cases for a confirmed Business Partner

There are two options to display historical clarification cases for a confirmed business partner in the Interaction Center:

- **In the Interaction Record work center,** click on the Clarification Cases History link to display all historical clarification cases related to the confirmed business partner.
- **In the Account Overview,** click on the Clarification Cases link to open a table of historical clarification cases related to the confirmed business partner.



The selection of historical clarification cases for a confirmed business partner will be determined based on the implementation of the `ISU_BP_BPEM_OBJ_QUERY` BAdI definition of the `ISU_BP_BPEM_OBJ_QUERY` enhancement spot on the ERP system, which you can use to influence the display of historical cases in the Account Overview and Interaction Records work centers. Using this BAdI, you can influence the display of clarification cases based on the clarification case status and main object associated with the case, specify the desired period for the display of cases and select the maximum number of clarification cases to be displayed. See the Customizing for SAP Utilities at *Customer Service* → *CRM Integration* → *BAdIs for Accessing Objects* → *Account Overview: BAdI for Clarification Cases*.

The sample implementation of the `ISU_BP_BPEM_OBJ_QUERY` BAdI definition returns a list of clarification cases based on the following criteria:

- The main object is the business partner
- The main object is one of the contract accounts of the confirmed business partner
- The main object is one of the contracts of the confirmed business partner
- The main object is one of the PoDs of the confirmed business partner
- The main object is one of the premises of the confirmed business partner

In the table of historical clarification cases, you can click on the *Case ID* hyperlinks to navigate to the clarification cases in display mode. You can also sort or filter the results.

Display Historical Clarification Cases for a confirmed Premise

In the *Premise Overview* work center, click on the *Clarification Cases* folder in the *Premise Overview* tree to display all open clarification cases based on the confirmed premise in the *Clarification Cases* table on the right hand side. Click the *View All* button on the top of the *Clarification Cases* table to display historical clarification cases. Select a clarification case in the table to see the case detail view below the table.

The selection of historical clarification cases for a confirmed premise is determined based on the implementation of the `ISU_PREMISE_OBJ` BAdI definition of the `ISU_PREMISE_OBJ_QUERY` enhancement spot on the ERP system, which can be used to influence the display of historical cases in the *Premise Overview*. Using this BAdI, you can influence the display of clarification cases based on the clarification case status and main object associated with the case, specify the desired period for the display of cases and select the maximum number of clarification cases to be displayed.

See the Customizing for SAP Utilities at *Customer Service* → *CRM Integration* → *BAdIs for Accessing Objects* → *Premise Overview: BAdI for Premise Objects*.

The sample implementation of the `ISU_PREMISE_OBJ` BAdI definition returns a list of clarification cases based on the following criteria:

- The main object is the confirmed premise
- The main object is the connection object of the premise
- The main object is one of the contract accounts associated with the premise
- The main object is one of the contracts associated with the premise
- The main object is one of the PODs in the premise

9.2 Integration of ECRMREPL in BPEM for Utilities

The SAP IS-U Contract Error Monitor does not provide any specific error processing of a failed replication of CRM utility contracts in an integrated CRM and ERP system landscape. The related `ECRMREPL` transaction merely displays the list of failed replication attempts of CRM transactions. The available integration between SAP IS-U Contract Error Monitor and BPEM supports the creation and processing of contracts failed during replication by means of BPEM clarification cases.

9.2.1 Customizing

9.2.1.1 Implementation of SAP Notes

Ensure that you implemented the following SAP Notes on your system:

- SAP Note [1875850](#) *Integration of ECRMREPL in BPEM for Utilities*
- SAP Note [1879135](#) *Integration of BPEM for ECRMREPL (Customizing)*

In addition, ensure that the following business process codes are available on your system and allocated to the `ECRM` business process area:

Business Process Code	Description
<code>ECRM0001</code>	ECRMREPL: Generation BPEM Job Header
<code>ECRM0002</code>	CRM: Replication of Contracts
<code>ECRM0003</code>	EC70: Create/Change BP and CA for Accelerated Entry
<code>ECRM0004</code>	CRM: Replication of Connection Object
<code>ECRM0005</code>	CRM: General Replication
<code>ECRM0006</code>	EC70: Create Move-Out for Accelerated Entry
<code>ECRM0007</code>	CRM: Replication of Point of Delivery
<code>ECRM0008</code>	CRM: Replication of Marked Contract Changes
<code>ECRM0009</code>	EC70: Create Move-Out for Accelerated Entry
<code>ECRM0010</code>	CRM: Mass Product Change

9.2.1.1 Clarification Case Categories

Define clarification case categories for the relevant business process codes as templates for the later creation of clarification cases (for details, see chapter 3.4.3 *Create and Maintain Clarification Case Categories*).



You can take advantage of the `ECRMREPL` BOR object type and its methods to navigate from a clarification case to the maintenance screen of the `ECRMREPL` transaction by means of a solution process.

9.2.2 Create Clarification Cases

Run the `ECRM_ECRMREPL_GEN_EMMA` report periodically on the ERP system to attach all contracts that failed to replicate to a BPEM job. Subsequently, prepare the created BPEM job followed by the creation of the respective clarification cases (for details, see chapter 5.2 *BPEM Job preparation* and 5.3 *Clarification Case Creation*).

The following options are available for an automatic execution of automatic solution processes as long the status of a clarification case is either in status *New* or *In Process*:

Option	Description
space	Solution process can be executed only once (default)
1	Solution process is executed only once; however, if the solution process finishes with an exception no automatic solution process of the same clarification case is further executed
2	Solution process is executed only once; however, if the solution process finishes without an exception the clarification case is automatically closed with a status change to <i>Completed</i>
3	Solution process can be executed periodically
4	Solution process is executed periodically until the solution process finishes without an exception
5	Solution process is executed periodically until the solution process finishes with an exception
6	Solution process is executed periodically; however, if the solution process finishes with an exception no automatic solution process of the same clarification case is further executed
7	Solution process is executed periodically; however, if the solution process finishes without an exception the clarification case is automatically closed with a status change to <i>Completed</i>

Figure 10-2 Options for Multiple Executions of Automatic Solution Processes



The system exits the processing of further automatic solution processes of a clarification case if the clarification case is requested to be closed (either condition 2 or 7 applies).

Update Application Log for Automatic Solution Processes

You can control the update of the application log of a clarification case after an execution of an automatic solution process. This option is particularly helpful if you marked an automatic solution process to be executed periodically to check for an automatic or manual resolution of the related business exception.

The following options are available for the logging of the execution of an automatic solution process in the application log of the clarification case:

Option	Description
Space	Always update the application log after the execution of a solution process
1	Never update the application log after the execution of a solution process
2	Update the application log only if the solution process finishes with an exception
3	Update the application log only if the solution process finishes without an exception

Figure 10-3 Options for Multiple Executions of Automatic Solution Processes

Execution Control of Solution Processes

You can make the execution of a solution process dependent on the execution of preceding solution processes in assigning a restriction to the solution process.

The following restrictions are available for both automatic and manual solution processes:

Option	Description
Space	No restriction: The solution process can be executed at any time
1	Execution only if at least one preceding solution process was executed
2	Execution only if at least one preceding manual solution process was executed
3	Execution only if the immediately preceding solution process was executed
Z	Execution of solution process is temporarily suspended

Figure 10-4 Options for Execution Control of Solution Processes

Alternative Description of Solution Processes

In previous releases, the displayed text of a solution process is derived from the descriptive text of its call definition, such as the method of the BOR, workflow task or action box. If there was the need for a different descriptive text, commonly an additional workflow task or a BOR method with the desired descriptive text for the solution process was implemented as a workaround. Within the newly created workflow task or BOR method, the initially intended method was processed. This workaround to change the descriptive text is no longer required.

Alternatively, you can create descriptive texts for solution processes in the Customizing and allocates them to any solution process thus overruling the default text of the entered call definition.

The figure below shows the Customizing of solution texts.

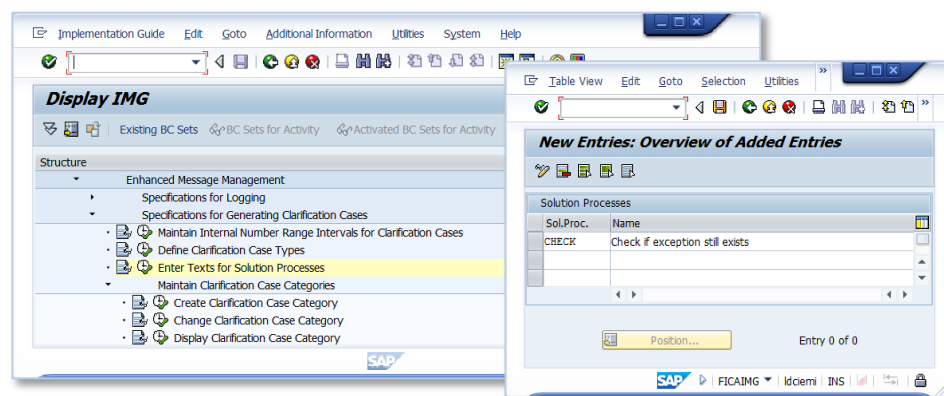


Figure 10-5 Customizing of Solution Texts

The following figure shows the Customizing of a solution process and the allocation of the previously defined solution text.

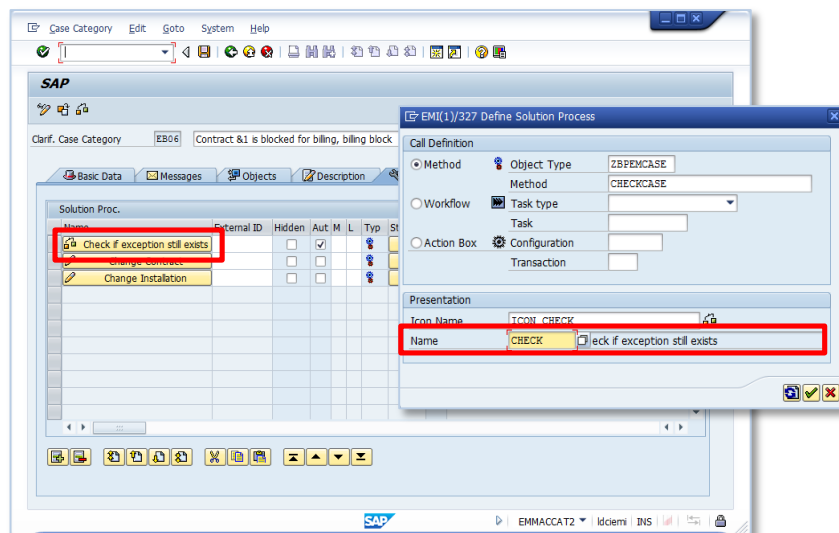


Figure 10-6 Customizing of a Sample Solution Process

10.1.2 Solution Process Control (BAdI)

In addition to the static Customizing of the available control settings for solution processes, you can dynamically control the execution of solution processes in implementing the `PREPARE_PROCESS_EXECUTION` method of the `EMMA_CASE` BAdI definition and changing various control parameters.

Method	Description
PREPARE_PROCESS_EXECUTION	The system processes this method before it checks the above described control settings for a restriction in executing any of the available solution processes. You can dynamically overwrite all control settings per solution process, for example, activating of temporarily deactivated solution processes or changing the order of the solution processes.

Figure 10-7 New Method of the `IF_EX_EMMA_CASE` ABAP OO Interface

Be very careful and plan for comprehensive testing when dynamically overwriting the control settings of the solution processes and, at the same time, change also the order of the solution processes. This could possibly lead to the situation that all available solution processes are blocked against execution due to the newly defined order. This can lead to a confusing result when implementing both `PREPARE_PROCESS_EXECUTION` and `ADJUST_ALVPROC_OUTPUT` methods of the `EMMA_CASE` BAdI definition since you can change in either method the visibility and the order of the solution processes. For more details about the `ADJUST_ALVPROC_OUTPUT` BAdI method, see chapter 5.8.2.3 *BAdI Definition BADI_EMMA_CASE_TRANSACTION*.

You can further control the execution of solution processes by the `CHECK_BEFORE_PROCESS_EXECUTION` and `CHECK_AFTER_PROCESS_EXEC_LIST` methods of the `EMMA_CASE` BAdI. With the `CHECK_BEFORE_PROCESS_EXECUTION` method, you can prevent the execution of a solution process exactly at the time when the execution of this solution process was triggered in either dialog or background mode.

The `CHECK_AFTER_PROCESS_EXEC_LIST` method completes not only the *Multiple Executions of Automatic Solution Processes* Customizing parameter of automatic solution processes but allows also an automatic closure of a clarification case in both dialog and background mode.

Method	Description
CHECK_BEFORE_PROCESS_EXECUTION	The system processes this method just before processing a solution process. This method allows you to dynamically and ultimately to decide if an execution of the selected solution process shall be permitted. If you raise the <code>SKIP_SOLP</code> exception, the system skips the execution of the solution process; if you raise the <code>SKIP_ALL</code> exception (valid only during the execution of automatic solution processes via the <code>EMMAJP</code>, <code>EMMACAP</code> or <code>FPEMMAMA</code> transactions), the systems exit the execution of the still available automatic solution processes of a clarification case. If you skip the execution of a solution process, the system either displays a respective message (dialog mode) or adds it to the log of the clarification case (background mode). You may specify an own message in the <code>ES_RETURN</code> parameter that replaces the standard message.
CHECK_AFTER_PROCESS_EXEC_LIST	The system processes this method after processing the last solution process of the initial list of solution processes to be processed. Return an <code>X</code> with the <code>CV_COMPLETED</code> parameter if you want to automatically complete the clarification case and, only in dialog mode, notify the agent that the system changed the status of the clarification case to <i>Completed</i>. You may specify an own text in the <code>ES_RETURN</code> parameter to replace the standard message text by an own message. The <code>CV_COMPLETED</code> parameter is pre-set to <code>X</code> if the previous automatic solution process was executed in background mode <u>and</u> the clarification case is requested to be completed according to the <i>Multiple Executions of Automatic Solution Processes</i> Customizing parameter of the solution process. The <code>CV_COMPLETED</code> parameter is always pre-set to <code>space</code> if an agent triggered the execution of the solution process during the clarification case maintenance because the <i>Multiple Executions of Automatic Solution Processes</i> parameter is ineffective at that time.

Figure 10-8 Further new Methods of the `IF_EX_EMMA_CASE` ABAP OO Interface

 If the `CHECK_AFTER_PROCESS_EXEC_LIST` method requests the closure of the clarification case (`CV_COMPLETED` parameter is set to X) in dialog mode, the system automatically changes the status of the clarification case to *Completed* prior to notifying the agent and requesting a confirmation of the automatic clarification case closure. This allows the agent to complete any activity related to the clarification case, for example, adding a conclusive note, or even change the status of the clarification case back to *In Process* if required.

10.1.3 Status Display of Solution Processes

The previously described possible restrictions on execution of a solution process are reflected on the *Process* screen during the clarification case maintenance. The *Status* column shows possible restrictions in dependency of the previously executed solution processes:

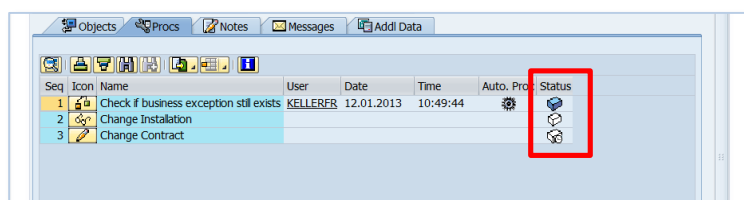


Figure 10-9 Processes Screen with Symbols to indicate Status of Execution Control

The symbols have the following meaning and are available as tooltips with details, for example, which solution process blocks the execution of another solution process. Note, that you must move the mouse to one of the symbols to display the respective tooltip.

Symbol	Meaning
	Solution process not executed yet but can be executed at any time
	Solution process was processed successfully
	Execution of solution process blocked because one of the previous solution processes has to be executed first (reason according to the tooltip).
	Execution of solution process canceled by an agent
	Execution of solution process canceled by the system
	Execution of solution process suspended

Figure 10-10 Description of Symbols on the *Processes* Screen

 You can overwrite the symbol in the status field with the `ADJUST_ALVPROC_OUTPUT` method of the `BADI_EMMA_CASE_TRANSACTION` BAdI. The same method allows you to change the symbol and the descriptive text of the solution process, too. For more detail about the `ADJUST_ALVPROC_OUTPUT` method, see chapter 5.8.2.3 *BAdI Definition BADI_EMMA_CASE_TRANSACTION*.

10.2 Hide Objects of Clarification Case Container from Agent

You can mark an object of the clarification case container as *Hidden* to prevent it from being visible to the agent. This option allows you to define objects to be used for internal clarification case processing but - as an object - without any significance concerning its visibility to the agent since the system minds the *Hidden* flag only during output on the *Object* screen of a clarification case.

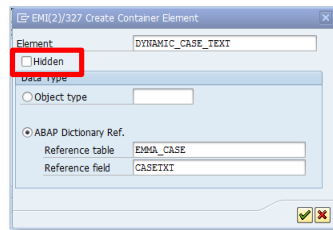


Figure 10-11 Checkbox to hide an Object from Agents

10.3 Evaluation of Runtime Errors

The ABAP runtime environment triggers an ABAP runtime error if an unhandled exception occurs while an ABAP program is being executed. In this case, the system terminates the execution of the program, creates a detailed error log (short dump) and eventually saves it in the affected ABAP system.

You can use the tools of the ABAP Dump Analysis (ST22 transaction) to monitor the ABAP runtime errors that have occurred in an ABAP system. The figure below shows the output of the ST22 transaction from where you can navigate to the detail screen of each runtime error, the so-called runtime error long text, for a further detailed analysis.

Current Date	Time	Application Server	User	Client	Runtime Error	Exception	Canceled Program
08.03.2016	19:22:42	ldciemw_EMW_17	UC4ADMIN	097	C CONV_T_CODEPAGE	CX_SY_CONVERSION_CODEPAGE	RS_QAP_COMPADR
08.03.2016	19:22:42	ldciemw_EMW_17	UC4ADMIN	170	C CONV_T_CODEPAGE	CX_SY_CONVERSION_CODEPAGE	RS_QAP_COMPADR
08.03.2016	19:22:42	ldciemw_EMW_17	UC4ADMIN	320	C CONV_T_CODEPAGE	CX_SY_CONVERSION_CODEPAGE	RS_QAP_COMPADR
08.03.2016	19:21:46	ldaiemw_EMW_17	UC4ADMIN	001	C CONV_T_CODEPAGE	CX_SY_CONVERSION_CODEPAGE	RS_QAP_COMPADR
08.03.2016	19:21:45	ldaiemw_EMW_17	UC4ADMIN	003	C CONV_T_CODEPAGE	CX_SY_CONVERSION_CODEPAGE	RS_QAP_COMPADR
08.03.2016	19:21:44	ldaiemw_EMW_17	UC4ADMIN	090	C CONV_T_CODEPAGE	CX_SY_CONVERSION_CODEPAGE	RS_QAP_COMPADR
08.03.2016	19:21:44	ldaiemw_EMW_17	UC4ADMIN	095	C CONV_T_CODEPAGE	CX_SY_CONVERSION_CODEPAGE	RS_QAP_COMPADR
08.03.2016	19:21:44	ldaiemw_EMW_17	UC4ADMIN	160	C CONV_T_CODEPAGE	CX_SY_CONVERSION_CODEPAGE	RS_QAP_COMPADR
08.03.2016	19:21:43	ldaiemw_EMW_17	UC4ADMIN	030	C CONV_T_CODEPAGE	CX_SY_CONVERSION_CODEPAGE	RS_QAP_COMPADR
08.03.2016	19:21:43	ldaiemw_EMW_17	UC4ADMIN	035	C CONV_T_CODEPAGE	CX_SY_CONVERSION_CODEPAGE	RS_QAP_COMPADR
08.03.2016	19:21:43	ldaiemw_EMW_17	UC4ADMIN	100	C CONV_T_CODEPAGE	CX_SY_CONVERSION_CODEPAGE	RS_QAP_COMPADR
08.03.2016	19:21:43	ldaiemw_EMW_17	UC4ADMIN	300	C CONV_T_CODEPAGE	CX_SY_CONVERSION_CODEPAGE	RS_QAP_COMPADR
08.03.2016	19:21:43	ldaiemw_EMW_17	UC4ADMIN	901	C CONV_T_CODEPAGE	CX_SY_CONVERSION_CODEPAGE	RS_QAP_COMPADR
08.03.2016	19:21:43	ldaiemw_EMW_17	UC4ADMIN	903	C CONV_T_CODEPAGE	CX_SY_CONVERSION_CODEPAGE	RS_QAP_COMPADR
08.03.2016	19:21:43	ldciemw_EMW_17	UC4ADMIN	075	C CONV_T_CODEPAGE	CX_SY_CONVERSION_CODEPAGE	RS_QAP_COMPADR
08.03.2016	19:21:43	ldciemw_EMW_17	UC4ADMIN	077	C CONV_T_CODEPAGE	CX_SY_CONVERSION_CODEPAGE	RS_QAP_COMPADR
08.03.2016	19:21:42	ldaiemw_EMW_17	UC4ADMIN	070	C CONV_T_CODEPAGE	CX_SY_CONVERSION_CODEPAGE	RS_QAP_COMPADR
08.03.2016	19:21:42	ldaiemw_EMW_17	UC4ADMIN	327	C CONV_T_CODEPAGE	CX_SY_CONVERSION_CODEPAGE	RS_QAP_COMPADR
08.03.2016	19:21:42	ldaiemw_EMW_17	UC4ADMIN	328	C CONV_T_CODEPAGE	CX_SY_CONVERSION_CODEPAGE	RS_QAP_COMPADR
08.03.2016	19:21:42	ldciemw_EMW_17	UC4ADMIN	000	C CONV_T_CODEPAGE	CX_SY_CONVERSION_CODEPAGE	RS_QAP_COMPADR
08.03.2016	19:21:42	ldciemw_EMW_17	UC4ADMIN	110	C CONV_T_CODEPAGE	CX_SY_CONVERSION_CODEPAGE	RS_QAP_COMPADR
08.03.2016	19:21:42	ldciemw_EMW_17	UC4ADMIN	770	C CONV_T_CODEPAGE	CX_SY_CONVERSION_CODEPAGE	RS_QAP_COMPADR
08.03.2016	15:47:52	ldaiemw_EMW_17	SAPSYS	300	C SYSTEM_CORE_DUMPED		SAPLOEA
08.03.2016	13:47:22	ldciemw_EMW_17	CS228259	000	C LOAD_PROGRAM_CLASS_MISMATCH		CL_COS_UTILITIES=====CP
08.03.2016	12:09:28	ldaiemw_EMW_17	SCHWARZSTE	095	C LOAD_PROGRAM_CLASS_MISMATCH		CL_RSO_TLOGO_BEFORE_EXPORT====CP
08.03.2016	12:03:50	ldciemw_EMW_17	DDIC	000	C DBSQL_SQL_ERROR	CX_SY_OPEN_SQL_DB	SAPLSTFI
08.03.2016	11:54:47	ldaiemw_EMW_17	SAPSYS	000	C SYNTAX_ERROR		CL_ENQ_CONTEXT_ADMIN=====CP
08.03.2016	11:53:17	ldciemw_EMW_17	MOURAVIN	300	C SYNTAX_ERROR		CL_SABP_RABAX_ADT_RES_DUMPS====CP
08.03.2016	11:53:16	ldciemw_EMW_17	MOURAVIN	300	C SYNTAX_ERROR		CL_SABP_RABAX_ADT_RES_DUMPS====CP

Figure 10-12 Transaction ST22

However, if the same background job cancel periodically due to the same root cause, or if the transactions cancel due to the same reason, the obligatory monitoring task become a challenge to sort out the runtime errors that occurred multiple times from the single occurrence of a particular runtime error. Of course, you can take advantage of the *Overview* option that is available in the toolbar of the ST22 transaction screen and navigate to the same type of runtime errors from there. Still the issue is to sort out the various roots of the same runtime error, for example, the various programs terminated due to a syntax error.

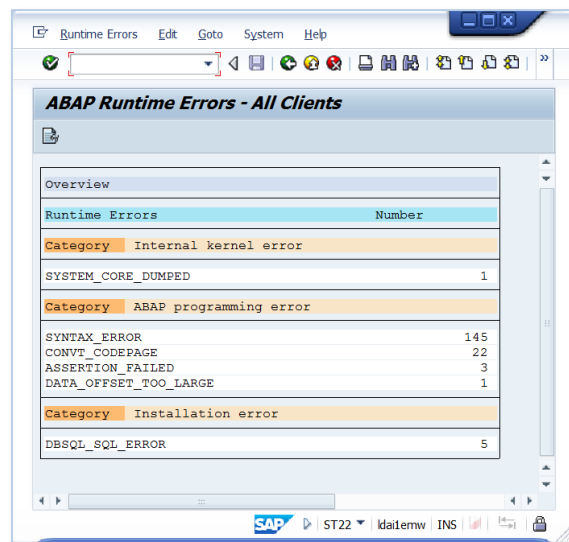


Figure 10-13 Runtime Error Statistics ST22 Transaction

SAP delivers the JLOG0002 business process code to support system operators in supporting the analysis of runtime errors by BPEM. The implementation of the related CL_EMMA_ANALYZE_RUNTIME_ERROR ABAP OO job analysis class retrieves information about runtime errors in application logs with an external ID composed as a concatenation of JLOG0002 and the number of the newly created BPEM job, for example, JLOG0002 (0100160264).

As previously explained, BPEM always requires a BPEM job as a starting point for the BPEM specific steps, such as BPEM job preparation and clarification case generation. However, the ABAP runtime environment does not create BPEM jobs by itself hence you must create them in a separate activity.

You create the required BPEM job with the EMMARTA transaction. The EMMARTA transaction creates one or more application logs in which the EMMARTA transaction records all selected ABAP runtime errors with the two EMMA 195 and EMMA 196 messages. These messages include all relevant information about the runtime errors in their message variables.

The figure on the next page shows the selection screen of the EMMARTA transaction with obvious similarities to the selection screen of the ST22 transaction.

Analysis of ABAP Runtime Errors - All Clients

General Selection Criteria for Runtime Errors

Current Date		to		
Time		to		
Application Server		to		
User Name		to		
Client ID		to		
ABAP Runtime Error		to		
Program Name		to		
Exception		to		
Transaction ID		to		

Grouping of Runtime Errors per BPEM Job

☒ No Group Creation
☐ ABAP Runtime Error
☐ Runtime Errors/Exception
☐ Runtime Err/Exception/Program
☐ All Runtime Errors

Summary of Runtime Error Trigger Messages in a BPEM Job

☐ Bus. Object per Runtime Error
☒ Only One Bus. Object per Job

BPEM Control Parameters

Authorization Group:
☐ Check for Open Cases
☐ Prepare Jobs
☐ Create Clarification Cases
☐ Display Jobs

SAP EMMARTA kda1emw INS

Figure 10-14 Selection Screen of the EMMARTA Transaction

The *General Selection Criteria for Runtime* includes all fields to identify the runtime errors to be processed by BPEM.

The *Grouping of Runtime Errors per BPEM Job* section defines how many BPEM jobs to create if there are similar runtime errors with concordant attributes, such as the same ABAP runtime error, exception or program name. In accordance with the specification, the EMMARTA transaction creates one or more application logs per BPEM Job. This grouping of runtime errors is especially helpful, if a number of similar ABAP runtime errors occur in the system likely to have the same root cause, so that they can be clarified together and their common cause removed.

Based on your specifications for *Summary of Trigger Messages of Runtime Errors in a BPEM Job*, you can specify if each ABAP runtime error is registered either with an own BPEM-specific message grouping sequence EMMA 011 and EMMA 012 (*Bus. Object per Runtime Error*), or if all ABAP runtime errors are grouped together within this sequence (*Only one Business Object per Job*). In the latter case, the system uses per group the identifier of the first found runtime error as the main object key.

In the *BPEM Control Parameters* section of the selection screen, you define the authorization group to be attached to the created BPEM job(s). In addition, you may suppress already at this stage the creation of duplicate clarification cases for the same ABAP runtime error. This is especially helpful when you group multiple ABAP runtime errors within the BPEM-specific message grouping sequence EMMA 011 and EMMA 012 since the standard duplicate check for already existing open clarification cases includes only the search for the same main object of clarification cases but not business objects attached to a clarification case. Mark the *Prepare Jobs* and *Create Clarification Settings* checkboxes if you want to perform the BPEM job preparation and the clarification case creation for the created BPEM jobs immediately after you created them via the EMMARTA transaction.

The following figures explain the result of the various grouping options using and focusing on the 145 occurred runtime errors due to a syntax error on the basis of the example above (see Figure 10-12), and using a simple clarification case category to create clarification cases. The sample clarification case category is defined as business object-specific for the JLOG0002 business process code, with the EMMA 195 message as triggering message and the concatenation of the three placeholders &_EMMA_MAINOBJ.ABAPRUNTIMEERROR&, &_EMMA_MAINOBJ.PROGRAMNAME& and &_EMMA_MAINOBJ.EXCEPTION& as case text.

When defining solution processes for clarification cases as of the JLOG0002 business process code, you can take advantage of the EMMART BOR object type and its `DISPLAY` and `DisplayRuntimeErrorsFromCase` methods to navigate from a clarification case to the analysis of either a single ABAP runtime analysis or all ABAP runtime errors of the respective clarification case via the ST22 transaction.

Two solution processes are attached to the sample clarification case category with the respective `DISPLAY` and `DISPLAYRUNTIMEERRORSFROMCASE` methods of the EMMART BOR object type. According to the definition of the EMMART~DISPLAY method, its &EMMARUNTIMEERROR& importing parameter requires the &_EMMA_MAINOBJ& object as input in the data flow definition of the solution process, while the &CASE& importing parameter of the EMMART~DISPLAYRUNTIMEERRORSFROMCASE method requires the &_EMMA_CASE.CASE& parameter as input in the data flow definition.

When you mark the *Runtime Error/Exception/Program* grouping parameter, the system analyses all runtime errors and creates one BPEM job per unique combination of the three attributes runtime error, exception and program name of all selected runtime errors. In combination with the *Only One Business Object per Job* parameter, the system creates one clarification case per BPEM job while, in the given example, all 17 created BPEM jobs relate to a different *Runtime Error/Exception/Program* grouping combination.

BusProce	TCode	PST	Job Ext.	Job Number	Mode	Interval	IncorrObj	Succ. Obj	Proc. Obj	Error	Informat	Total	Start Date	Start Time	Sta JP	Sta	Stat Cases
JLOG0002	ST22		ABAP Runtime Error	100160461		0	1	0	1	0	0	3	09.03.2016	14:50:30	X	1	1
			ABAP Runtime Error	100160462		0	1	0	1	0	0	3			X	1	1
			ABAP Runtime Error	100160463		0	1	0	1	0	0	127			X	1	1
			ABAP Runtime Error	100160464		0	1	0	1	0	0	5			X	1	1
			ABAP Runtime Error	100160465		0	1	0	1	0	0	21			X	1	1
			ABAP Runtime Error	100160466		0	1	0	1	0	0	3			X	1	1
			ABAP Runtime Error	100160467		0	1	0	1	0	0	5			X	1	1
			ABAP Runtime Error	100160468		0	1	0	1	0	0	2			X	1	1
			ABAP Runtime Error	100160469		0	1	0	1	0	0	3			X	1	1
			ABAP Runtime Error	100160470		0	1	0	1	0	0	3			X	1	1
			ABAP Runtime Error	100160471		0	1	0	1	0	0	5			X	1	1
			ABAP Runtime Error	100160472		0	1	0	1	0	0	3			X	1	1
			ABAP Runtime Error	100160473		0	1	0	1	0	0	107			X	1	1
			ABAP Runtime Error	100160474		0	1	0	1	0	0	3			X	1	1
			ABAP Runtime Error	100160475		0	1	0	1	0	0	3			X	1	1
			ABAP Runtime Error	100160476		0	1	0	1	0	0	3			X	1	1
			ABAP Runtime Error	100160460		0	1	0	1	0	0	3		14:50:28	X	1	1

Figure 10-15 List of created BPEM jobs by the EMMART Transaction

From the message list of each of these clarification cases, you can navigate to the respective runtime error. However, the EMMART~DISPLAYRUNTIMEERRORSFROMCASE method does not yield to a list of the desired runtime errors attached to the clarification cases but calls the ST22 transaction instead passing all runtime errors that can be found in the object list of a particular clarification case. The required object list needs to be created during the clarification case creation process, for example, in adding the code snippet below to the COMPLETE_CASE method of the EMMA_CASE BAdI.

```
METHOD if_ex_emma_case~complete_case.

    CASE is_case-ccat.

        WHEN 'JL01'.
            * add runtime errors to object list of clarification case
            DATA: lv_object TYPE LINE OF emma_cobj_t.
            CLEAR lv_object.
            LOOP AT it_msg_link ASSIGNING FIELD-SYMBOL(<fs_msg_link>)
                WHERE msgid = 'EMMA'
                    AND msgno = '195'
                    AND NOT msgv4 IS INITIAL
                    AND msgv4 <> is_case-mainobjkey.
                lv_object-celemname = '#' && sy-tabix.
                lv_object-refobjtype =
                    cl_emma_analyze_runtime_error=>co_mainobjtype.
                lv_object-id = <fs_msg_link>-msgv4.
                APPEND lv_object TO ct_objects.
            ENDLOOP.
        ENDCASE.

    ENDMETHOD.
```

Figure 10-16 shows the resulting object list the runtime errors after implementing the code snippet above and generating the related clarification cases:

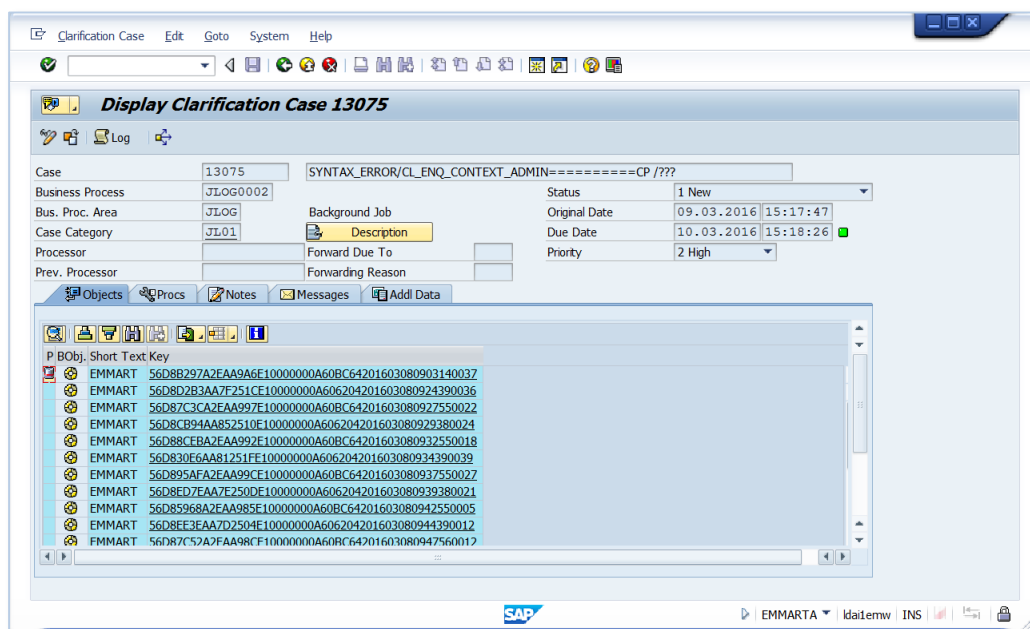


Figure 10-16 List of created BPEM jobs by the EMMARTA Transaction

Figure 10-17 shows the definition of a solution process and its related data flow of the clarification case category using the EMMART~DISPLAYRUNTIMEERRORSFROMCASE method:

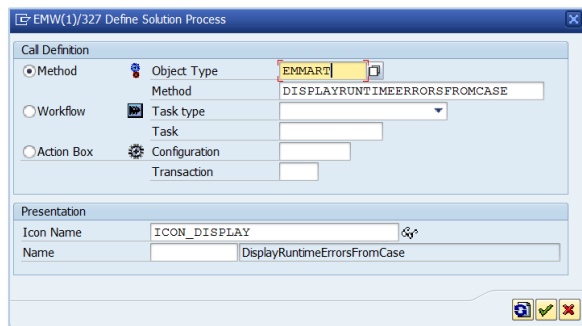


Figure 10-17 Solution Process to display all Runtime Errors allocated to a Clarification Case

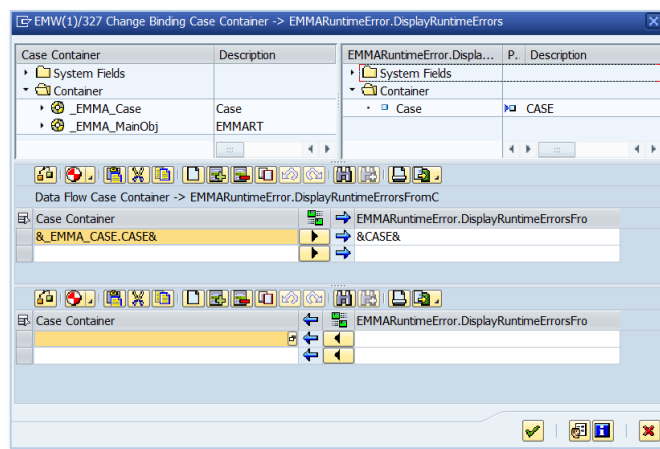


Figure 10-18 Data Flow of the Solution Process as of the figure above

Figure 10-19 shows the list of runtime errors as referenced in the clarification case when executing the solution process that, as said above, call eventually the ST22 transaction via the EMMART~DISPLAYRUNTIMEERRORSFROMCASE method.

List of Selected Runtime Errors							
Runtime Errors							
Current Date	Time	Application Server	User	Client	K	Runtime Error	Exception
08.03.2016	11:54:47	ldaiemwv_EMW_17	SAPSYS	000	C	SYNTAX_ERROR	CL_ENQ_CONTEXT_ADMIN=====CP
08.03.2016	11:34:38	ldaiemwv_EMW_17	SAPSYS	000	C	SYNTAX_ERROR	CL_ENQ_CONTEXT_ADMIN=====CP
08.03.2016	11:32:56	ldaiemwv_EMW_17	SAPSYS	000	C	SYNTAX_ERROR	CL_ENQ_CONTEXT_ADMIN=====CP
08.03.2016	11:29:39	ldaiemwv_EMW_17	SAPSYS	000	C	SYNTAX_ERROR	CL_ENQ_CONTEXT_ADMIN=====CP
08.03.2016	11:27:55	ldaiemwv_EMW_17	SAPSYS	000	C	SYNTAX_ERROR	CL_ENQ_CONTEXT_ADMIN=====CP
08.03.2016	11:24:38	ldaiemwv_EMW_17	SAPSYS	000	C	SYNTAX_ERROR	CL_ENQ_CONTEXT_ADMIN=====CP
08.03.2016	11:22:55	ldaiemwv_EMW_17	SAPSYS	000	C	SYNTAX_ERROR	CL_ENQ_CONTEXT_ADMIN=====CP
08.03.2016	11:19:38	ldaiemwv_EMW_17	SAPSYS	000	C	SYNTAX_ERROR	CL_ENQ_CONTEXT_ADMIN=====CP
08.03.2016	11:17:55	ldaiemwv_EMW_17	SAPSYS	000	C	SYNTAX_ERROR	CL_ENQ_CONTEXT_ADMIN=====CP
08.03.2016	11:14:38	ldaiemwv_EMW_17	SAPSYS	000	C	SYNTAX_ERROR	CL_ENQ_CONTEXT_ADMIN=====CP
08.03.2016	11:12:55	ldaiemwv_EMW_17	SAPSYS	000	C	SYNTAX_ERROR	CL_ENQ_CONTEXT_ADMIN=====CP
08.03.2016	11:09:38	ldaiemwv_EMW_17	SAPSYS	000	C	SYNTAX_ERROR	CL_ENQ_CONTEXT_ADMIN=====CP
08.03.2016	11:07:55	ldaiemwv_EMW_17	SAPSYS	000	C	SYNTAX_ERROR	CL_ENQ_CONTEXT_ADMIN=====CP
08.03.2016	11:04:38	ldaiemwv_EMW_17	SAPSYS	000	C	SYNTAX_ERROR	CL_ENQ_CONTEXT_ADMIN=====CP
08.03.2016	11:02:55	ldaiemwv_EMW_17	SAPSYS	000	C	SYNTAX_ERROR	CL_ENQ_CONTEXT_ADMIN=====CP
08.03.2016	10:59:38	ldaiemwv_EMW_17	SAPSYS	000	C	SYNTAX_ERROR	CL_ENQ_CONTEXT_ADMIN=====CP
08.03.2016	10:57:56	ldaiemwv_EMW_17	SAPSYS	000	C	SYNTAX_ERROR	CL_ENQ_CONTEXT_ADMIN=====CP
08.03.2016	10:54:39	ldaiemwv_EMW_17	SAPSYS	000	C	SYNTAX_ERROR	CL_ENQ_CONTEXT_ADMIN=====CP
08.03.2016	10:52:55	ldaiemwv_EMW_17	SAPSYS	000	C	SYNTAX_ERROR	CL_ENQ_CONTEXT_ADMIN=====CP

Figure 10-19 List of created BPBM jobs by the EMMART Transaction

If you mark on the selection screen of the EMMARTA transaction the *Business Object per Runtime Error* parameter instead of the *Only One Business Object per Job* parameter, the system creates still 17 BPEM jobs but only a single clarification case for each runtime error thus in total 145 clarification cases. This is because each individual runtime error is treated individually as an own business exception but still summarized in one single BPEM job.

BusProce	Obj. Type	EMMARTA	MC ID	No.	Message Text
JLOG0002	EMMART	56D8098DAA7C2503E10000000A606204201603081104380011	EMMA	11	Start business process JLOG0002 for EMMART 56D8098DAA7C2503E10000000A606204201603081104380011 in business process area JLOG
EMMART			EMMA	195	Runtime error SYNTAX ERROR in CL ENQ_CONTEXT ADMIN=====CP with the exception ??? (ID 56D8098DAA7C2503E10000000A6062042
EMMART			EMMA	196	Date 08.03.2016 / Time 11:04:38 / User SAPSYS / Transaction ID 56D8098DAA7C2503E10000000A606204
EMMART		56D809ADAA7C2503E10000000A606204201603081114380011	EMMA	11	Start business process JLOG0002 for EMMART 56D809ADAA7C2503E10000000A606204201603081114380011 in business process area JLOG
EMMART			EMMA	195	Runtime error SYNTAX ERROR in CL ENQ_CONTEXT ADMIN=====CP with the exception ??? (ID 56D809ADAA7C2503E10000000A606204
EMMART			EMMA	196	Date 08.03.2016 / Time 11:14:38 / User SAPSYS / Transaction ID 56D809ADAA7C2503E10000000A606204
EMMART		56D822CEAA7B2508E10000000A606204201603080919390016	EMMA	11	Start business process JLOG0002 for EMMART 56D822CEAA7B2508E10000000A606204201603080919390016 in business process area JLOG
EMMART			EMMA	195	Runtime error SYNTAX ERROR in CL ENQ_CONTEXT ADMIN=====CP with the exception ??? (ID 56D822CEAA7B2508E10000000A6062042
EMMART			EMMA	196	Date 08.03.2016 / Time 09:19:39 / User SAPSYS / Transaction ID 56D822CEAA7B2508E10000000A606204
EMMART		56D82391AA7B250EE10000000A606204201603081109380022	EMMA	11	Start business process JLOG0002 for EMMART 56D82391AA7B250EE10000000A606204201603081109380022 in business process area JLOG
EMMART			EMMA	195	Runtime error SYNTAX ERROR in CL ENQ_CONTEXT ADMIN=====CP with the exception ??? (ID 56D82391AA7B250EE10000000A6062042
EMMART			EMMA	196	Date 08.03.2016 / Time 11:09:38 / User SAPSYS / Transaction ID 56D82391AA7B250EE10000000A606204
EMMART		56D824FFAA7B2508E10000000A606204201603081019390016	EMMA	11	Start business process JLOG0002 for EMMART 56D824FFAA7B2508E10000000A606204201603081019390016 in business process area JLOG
EMMART			EMMA	195	Runtime error SYNTAX ERROR in CL ENQ_CONTEXT ADMIN=====CP with the exception ??? (ID 56D824FFAA7B2508E10000000A6062042
EMMART			EMMA	196	Date 08.03.2016 / Time 10:19:39 / User SAPSYS / Transaction ID 56D824FFAA7B2508E10000000A606204

Figure 10-20 Runtime Errors treated as individual business expectations

Appendix A - Reports with BPEM Specific Coding

Appendix A1 - Creation of Clarification Case via Application Log

The following report writes the result of a data processing subroutine to messages to an application log completed by the mandatory BPEM messages.

```

*&-----*
*& Report  ZBPEN_CUSTOMER_REPORT
*&
*&-----*
REPORT  zbpem_customer_report MESSAGE-ID zemma.

* constant values
CONSTANTS: gc_bpcode TYPE emma_bpcode VALUE ' ZBILL001',
            gc_tcode  TYPE tcode  VALUE 'ZREPORT_1',
            gc_main_object_type TYPE swo_objtyp VALUE 'INSTLN'.

* customer data definition
TYPES: BEGIN OF ty_data,
        vkonto      TYPE ever-vkonto,
        anlage      TYPE v_eanl-anlage,
        vstelle     TYPE v_eanl-vstelle,
        sparte      TYPE v_eanl-sparte,
        ableinh     TYPE v_eanl-ableinh,
        akklasse    TYPE v_eanl-aklasse,
        tariftyp    TYPE v_eanl-tariftyp,
        ab          TYPE v_eanl-ab,
      END OF ty_data.
DATA: gt_data TYPE TABLE OF ty_data,
      gs_data TYPE ty_data.

* data definition to handle BPEM
DATA: gr_bpc TYPE REF TO cl_emma_bpc,
      gv_bparea TYPE emma_bparea,
      gv_bpa_active TYPE emma_bpa_active.

INCLUDE ZLOGDEF.

START-OF-SELECTION.

* Step 1: Check if BPEM business process is active
CALL FUNCTION 'EMMA_LOG_PROCESS_CHECK_ACTIVE'
  EXPORTING
    iv_bpcode = gc_bpcode
  IMPORTING
    ev_bparea = gv_bparea
    ev_bpa_active = gv_bpa_active
  EXCEPTIONS
    extract_class_not_found = 1
    bpcode_not_found = 2
    bparea_not_found = 3
    OTHERS = 4.
IF sy-subrc <> 0 OR gv_bpa_active IS INITIAL.
  EXIT.
ENDIF.

* Step 2: create application log
gs_log-object = gc_log_object.
gs_log-subobject = gc_log_subobject.
CONCATENATE 'ZREPORT_1' sy-datum sy-timlo INTO gs_log-extnumber.
gs_log-alprog = sy-cprog.
gs_log-altcode = sy-tcode.
CALL FUNCTION 'BAL_LOG_CREATE'
  EXPORTING
    i_s_log = gs_log
  IMPORTING
    e_log_handle = gv_log_handle.

* Step 3: register your business process

```

```

CALL METHOD cl_emma_job=>create_job
EXPORTING
  iv_extnumber = gs_log-extnumber
  iv_bpcode    = gc_bpcode
  iv_tcode     = gc_tcode
  iv_silent    = 'X'
EXCEPTIONS
  not_created = 1.
IF sy-subrc NE 0.
  MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
    WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
ENDIF.

* Step 4: Create instance for BPBM start/end object message
CREATE OBJECT gr_bpc
EXPORTING
  iv_bpcode = gc_bpcode.

* process customer data
PERFORM get_data TABLES gt_data.

* write resulting messages to application log
* for a later evaluation by BPBM
LOOP AT gt_data INTO gs_data.

* Step 5a: Create BPBM start message and write it to application log
CALL METHOD gr_bpc->start
EXPORTING
  iv_main_bus_objtype = gc_main_object_type
  iv_main_bus_objkey  = gs_data-anlage " ID of current business object
IMPORTING
  es_msg              = gs_msg.
macro_add_msg_to_bal_log gv_log_handle.

* Step 5b: Write customer message based on received data
IF 1 = 2. MESSAGE s001(zemma). ENDIF.
macro_create_msg 'E' 'ZEMMA' '001' gs_data-anlage gs_data-ab space space.
macro_add_msg_parameter 'ACCOUNT' gs_data-vkonto.
macro_add_msg_parameter 'VSTELLE' gs_data-vstelle.
macro_add_msg_parameter 'SPARTE' gs_data-sparte.
macro_add_msg_parameter 'ABLEINH' gs_data-ableinh.
macro_add_msg_parameter 'AKLASSE' gs_data-aklasse.
macro_add_msg_parameter 'TARIFTYP' gs_data-tariftyp.
macro_add_msg_to_bal_log gv_log_handle.

* Step 5c: Create BPBM end message and write it to application log
CALL METHOD gr_bpc->end
EXPORTING
  iv_main_bus_objtype = gc_main_object_type
  iv_main_bus_objkey  = gs_data-anlage " ID of current business object
IMPORTING
  es_msg              = gs_msg.
macro_add_msg_to_bal_log gv_log_handle.

ENDLOOP.

* Step 6: Save application log on database
REFRESH gt_log_handle.
APPEND gv_log_handle TO gt_log_handle.
CALL FUNCTION 'BAL_DB_SAVE'
EXPORTING
  i_t_log_handle = gt_log_handle.

*&-----*&
*& Form get_data
*&-----*&
FORM get_data.
  [...]
ENDFORM.

```

The following ZLOGDEF include contains data definitions and macros to handle the accesses to the application log.

```
*&-----*
*& General definition to handle application logs
*&-----*

* constant values
CONSTANTS: gc_log_object TYPE bal_s_log-object VALUE 'IU00',
            gc_log_subobject TYPE bal_s_log-subobject VALUE 'INFO'.

* data definition to handle application log
DATA: gs_log          TYPE bal_s_log,
      gv_log_handle   TYPE balloghndl,
      gt_log_handle   TYPE bal_t_logh,
      gs_msg          TYPE bal_s_msg,
      gs_msg_par      TYPE bal_s_par.

* macros for application log
DEFINE macro_create_msg.
  clear gs_msg.
  gs_msg-probclass = space.    " problem class: other
  gs_msg-msgty     = &1.       " message type: error
  gs_msg-msgid     = &2.       " message id
  gs_msg-msgno     = &3.       " message number
  write &4          to gs_msg-msgv1 left-justified.
  write &5          to gs_msg-msgv2 left-justified.
  write &6          to gs_msg-msgv3 left-justified.
  write &7          to gs_msg-msgv4 left-justified.
END-OF-DEFINITION.
DEFINE macro_add_msg_parameter.
  gs_msg_par-parname = &1.
  write &2 to gs_msg_par-parvalue left-justified.
  append gs_msg_par to gs_msg-params-t_par.
  gs_msg-params-altext = sy-cprog.
END-OF-DEFINITION.
DEFINE macro_add_msg_to_bal_log.
  call function 'BAL_LOG_MSG_ADD'
    exporting
      i_log_handle = &1
      i_s_msg      = gs_msg.
END-OF-DEFINITION.
```

Appendix A2 - Creation of Clarification Case by BAPI

The following report creates clarification cases directly in the application by the BAPI_EMMA_CASE_CREATE function module.

```

*&-----*
*& Report  ZBPEM_CUSTOMER_REPORT
*&
*&-----*
REPORT  zbpem_customer_report MESSAGE-ID zemma.

* constant values
CONSTANTS: gc_bpcode TYPE emma_bpcode VALUE ' ZBILL001',
            gc_tcode  TYPE tcode      VALUE 'ZREPORT_1',
            gc_main_object_type TYPE swo_objtyp VALUE 'INSTLN',
            gc_emma_case_prio_average TYPE emma_cprio VALUE '3'.

* customer data definition
TYPES: BEGIN OF ty_data,
        vkonto TYPE ever-vkonto,
        anlage TYPE v_eanl-anlage,
        vstelle TYPE v_eanl-vstelle,
        sparte TYPE v_eanl-sparte,
        ableinh TYPE v_eanl-ableinh,
        aklasse TYPE v_eanl-aklasse,
        tariftyp TYPE v_eanl-tariftyp,
        ab TYPE v_eanl-ab,
      END OF ty_data.
DATA: gt_data TYPE TABLE OF ty_data,
      gs_data TYPE ty_data.

* data definition for clarification case creation
DATA: ls_case_create TYPE bapi_emma_case_create,
      lv_test_run TYPE bapi_emma_case-testrun,
      lv_case TYPE bapi_emma_case-casenr,
      lt_objects TYPE TABLE OF bapi_emma_case_object,
      ls_object TYPE bapi_emma_case_object,
      lt_return TYPE bapiret2 OCCURS 0.

* macro to collect business objects of clarification case
DEFINE macro_add_case_object.
  ls_object-celemname = &1.
  ls_object-refobjtype = &2.
  ls_object-id = &3.
  append ls_object to lt_objects.
END-OF-DEFINITION.

START-OF-SELECTION.

* process customer data
PERFORM get_data TABLES gt_data.

* use resulting messages to create clarification cases

LOOP AT gt_data INTO gs_data.

* Basic clarification case data
  ls_case_create-ccat = 'BIZ1'.
  ls_case_create-prio = co_emma_case_prio_average.
  ls_case_create-brgru = space.
  ls_case_create-mainobjtype = gc_main_object_type.
  ls_case_create-mainobjkey = gs_data-anlage.
  ls_case_create-orig_date = sy-datum.
  ls_case_create-orig_time = sy-uzeit.
  ls_case_create-logsys = space.

* Objects for the clarification case container
  REFRESH lt_objects.
  macro_add_case_object 'VAR_ZEMMA_001_VSTELLE'
                      'PREMISES' gs_data-vstelle.
  macro_add_case_object 'VAR_ZEMMA_001_SPARTE'
                      'ISUSECTOR' gs_data-sparte.

```

```
macro_add_case_object 'VAR_ZEMMA_001_ABLEINH'
                        'READUNIT'    gs_data-ableinh.
macro_add_case_object 'VAR_ZEMMA_001_AKLASSE'
                        'BILLINGCLS'  gs_data-aklasse.
macro_add_case_object 'VAR_ZEMMA_001_TARIFTYP'
                        'RATECAT'    gs_data-tariftyp.

* Call BAPI BPAI_EMMA_CASE to create clarification case
CALL FUNCTION 'BAPI_EMMA_CASE_CREATE'
  EXPORTING
    case_create = ls_case_create
  IMPORTING
    case        = lv_case
  TABLES
    OBJECTS     = lt_objects
    return      = lt_return.

ENDLOOP.

* Save created clarification cases on the database
CALL FUNCTION 'BAPI_TRANSACTION_COMMIT'.

*&-----*&
*&  Form  get_data
*&-----*&
FORM get_data.
  [...]
ENDFORM.
```

Appendix A3 – Sample Mass Activity Interval Processing Module

The following code snippet outlines the calling sequence in an interval processing function module in more details than as explained in chapter 6.6 *Connect Mass Activities to BP*EM. You may use the code snippet below as a basis for your own implementation.

```

FUNCTION zmass_activity_interval_sample.
*-----
*""Local Interface:
*  IMPORTING
*    VALUE(I_LOW) TYPE C
*    VALUE(I_HIGH) TYPE C
*    VALUE(I_BASICS) TYPE FKK_MAD_BASICS OPTIONAL
*  TABLES
*    T_COUNTER STRUCTURE FKK_MASS_ACT_COUNT OPTIONAL
*  CHANGING
*    REFERENCE(C_TEST_NUM) TYPE I
*    REFERENCE(C_SUCCESS_NUM) TYPE I
*    REFERENCE(C_ERROR_NUM) TYPE I
*-----

* constant values to be replaced by suitable values
CONSTANTS: co_probclass TYPE balprobcl VALUE '4',
            co_failed TYPE boole VALUE 'X',
            co_bparea TYPE emma_bparea VALUE 'Z*',
            co_bpcode TYPE emma_bpcode VALUE 'Z*',
            co_main_bus_objtype TYPE swo_objtyp VALUE 'MY_BUSOBJ'.

* local data definition
DATA: lv_validation      TYPE boole,
      lv_cleanup         TYPE boole,
      lv_object_status   TYPE boole,
      lv_business_object TYPE string.

* preparations and validations before start of interval processing
[...].
IF lv_validation = co_failed.
*   write messages w/o a reference to a business object
CALL FUNCTION 'FKK_AKTIV2_APPL_LOG_MSG'
  EXPORTING
    i_msgty      = sy-msgty
    i_msgid      = sy-msgid
    i_msgno      = sy-msgno
    i_msgv1      = sy-msgv1
    i_msgv2      = sy-msgv2
    i_msgv3      = sy-msgv3
    i_msgv4      = sy-msgv4
    i_probclass_this_message = co_probclass.
  EXIT.
ENDIF.

* initialize first object to be processed
lv_business_object = i_low.

* process object list
DO.

*   register business object
CALL FUNCTION 'FKK_AKTIV2_EMMA_REG_BUS_OBJ'
  EXPORTING
    i_bparea      = co_bparea
    i_bpcode      = co_bpcode
    i_main_bus_objtype = co_main_bus_objtype
    i_main_bus_objkey  = lv_business_object.

*   process selected business object
[...].

*   check result of processing business object
IF lv_object_status = co_failed.
*     add one or more error messages to the application log
CALL FUNCTION 'FKK_AKTIV2_EMMA_LOG_MSG'

```

```
EXPORTING
    i_msgty                = sy-msgty
    i_msgid                = sy-msgid
    i_msgno                = sy-msgno
    i_msgv1                = sy-msgv1
    i_msgv2                = sy-msgv2
    i_msgv3                = sy-msgv3
    i_msgv4                = sy-msgv4
    i_probclass_this_message = co_probclass.
ENDIF.

* unregister business object
CALL FUNCTION 'FKK_AKTIV2_EMMA_UNREG_BUS_OBJ'.

* get next business object
lv_business_object = lv_business_object + 1.
IF lv_business_object > i_high. EXIT. ENDIF.

ENDDO.

* do interval cleanup
* [...]
IF lv_cleanup = co_failed.
* write messages w/o a reference to a business object
CALL FUNCTION 'FKK_AKTIV2_APPL_LOG_MSG'
    EXPORTING
        i_msgty                = 'I'
        i_msgid                = '>3'
        i_msgno                = '211'
        i_msgv1                = space
        i_msgv2                = space
        i_msgv3                = space
        i_msgv4                = space
        i_probclass_this_message = co_probclass.
ENDIF.

ENDFUNCTION.
```

Appendix B – List of Important SAP Notes

SAP Note	Description
1704335	New EMMA: New job analysis classes
1704940	EMMA - table parameters are not processed
2203014	BAdI interface for change of authorization group
2195757	Enhancement of BAdI interfaces for maintenance of clarification case
2117693	Downport BPEM connection ECRMREPL
1927697	EMMAC2 - improved error handling for forwarding method
2099146	Clarification case category changeable despite client setting
2048504	FI-CA: BPEM - Opening the application log in FI-CA mass transaction takes long
1966799	BPEM: Intervals of mass activities incomplete
1970378	BPEM: EMMAC2 toolbar does not correspond to the authorization
1916039	BPEM: Unique identification of solution processes
1906876	BPEM: EMMACAP repeatedly executes automatic solution processes
1908200	BPEM: Standard forwarding reason or reversal reason
1776229	Determining the forwarding method
1796153	Employee determination with attributes of clarification case
1746279	BPEM: Business process codes for clarification case generation
Performance related SAP Notes	
2109793	Improved Database Access in class CL_EMMA_JOB
1952655	BPEM: DBTABLOG growing caused by wrong technical settings
2019068	FPEMMAMA performance during job preparation