Guideline

Performance of the Initial Download from SAP for Utilities into SAP CRM

Document Version 1.01 – February 1, 2010

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

Copyright

Document History

Version	Date	Comment
1.01	01/02/2010	1st version of document published

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
Example text	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, SELECT and INCLUDE.
`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.

Icons

Icon	Meaning
	Caution
	Example
	Note

Motivation

The guideline ***Performance of the Initial Download from SAP for Utilities into SAP CRM*** is aimed at customers, SAP consultants and partners who are responsible for configuring and executing the initial download from an SAP for Utilities system to the SAP CRM system. This document supplements the

- SAP courses concerning SAP CRM and CRM Middleware (e.g. CR010 CRM Overview, CRM100 CRM basics)
- Various Setup & Load guides available in the SAP for Utilities space of the ***SAP Service Marketplace*** (http://service.sap.com/)
- Further documentation is available in the ***SAP Online Library Documentation*** (http://sap.help.com/) and release notes.

The purpose of the document is to give guidelines as well as to share experience for an efficient and high performance execution of an initial download from an SAP for Utilities system to an SAP CRM system, such as:

- Configuration of the initial download
- Ensure optimum performance for the initial download of the relevant business objects
- Avoid unnecessary business data being downloaded
- Monitoring of the initial download and taking corrective actions in case of problems
- Handle errors occurring during the initial download of business objects
- Minimize performance bottlenecks

The document explains the setup and configuration of the SAP for Utilities and SAP CRM systems only if necessary for the understanding of the solution of a described performance problem. This document does not cover the performance optimization of the delta down- and upload and the synchronization download.

The document has the following structure:

Chapter 1 describes the basics of an initial download

Chapter 2 describes how to plan and execute tests to achieve an acceptable performance

Chapter 3 describes the most important technical activities to prepare the involved systems

Chapter 4 describes the most important functional activities to prepare the involved system

Chapter 5 describes how to perform an initial download

Chapter 6 describes measures to improve the performance of an initial download

Chapter 7 provides a list of important repository tables

Chapter 8 contains a list of important SAP Notes related to initial download

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 Initial Download

1.1 SAP CRM Middleware

There are two options to integrate a SAP CRM system into a system landscape: SAP CRM Middleware and the SAP Exchange Infrastructure. The SAP CRM Middleware is a lightweight, build-in integration technology, which is installed together with the CRM enterprise applications on the SAP CRM system. The SAP CRM middleware is optimized for CRM processes and may be used for a tight integration to SAP OLTP systems. The SAP Exchange Infrastructure is SAP's solution for large integration projects.

The SAP CRM Middleware (further referred to as CRM Middleware) connects individual applications and is an integrative part of the CRM solution. It replicates, synchronizes and distributes data between different components and various types of data producers, such as, mobile clients, ERP back end, SAP NetWeaver Business Intelligence, SAP APO, and the CRM system applications. The CRM Middleware provides all participants with the information they require in messages in various types of BDoc's (business documents). You can imagine a BDoc as a container to transport information on changes made to business objects. In turn, messages are created based on these BDoc types.

RFC (remote function call) connections are used for connecting systems with each other. BDoc's are send and received through outbound and inbound queues of the CRM Middleware thus building the connection points of the participating systems.

The message flow in the CRM system is controlled and monitored by the message flow control facility, which is a key component of the CRM Middleware. The messages flow control facility provides a control for incoming and outgoing messages.

The communication between the CRM and the ERP based OLTP system (online transaction processing system) supports three different types of data exchanges (replication): the Initial load to be performed once as part of the cutover and before the go-live, the delta down and upload to be performed after go-live and the synchronization download. In this context, the ***SAP for Utilities system*** is further referred to as the ***ERP system*** and the ***SAP CRM*** system as the ***CRM system***.

Initial download: The initial load from a SAP ERP based ERP system is used to download all business, customizing and condition data and insert it into the CRM system. The initial download supplies the CRM system with the initial dataset.

Delta down and upload: The delta down and upload is used to keep the business and master data in CRM and the ERP system in sync during normal operation. Therefore, all new and changed data relevant for the connected systems have to be exchanged. ***Download*** means that data is transferred from the ERP system into the CRM system; ***upload*** means transferring data from the CRM into the ERP system.

You can perform an initial download with either an Initial Download (transaction R3AS) or a Request Download (transaction R3AR4). Basically, the way the request download works is similar to the initial download. However, it is possible to perform a request download at any time and you can start a request download for individual business objects that already exist in the CRM system. However, to avoid errors it is always better to have the business objects deleted before re-replication (for example, you can use the CRM_ORDER_DELETE program to delete contracts in the CRM system).

Based on our experience and assuming a high volume of business objects to be replicated from the ERP system into the CRM system, we recommend to use the Request Download for the execution of an initial download. This is, because the data extraction of the relevant business objects of the SAP for Utilities relevant business objects is much faster than the

processing of the data in the CRM system and there are more options for a parallelization of the initial download.

Synchronization download: This type of download is used for customizing data and is only possible between a CRM system and an ERP based OLTP system. In contrast to the delta download for business objects, customizing data is not downloaded automatically. Instead, you must initiate the synchronization download manually or it should be scheduled for fixed time intervals (for example, weekly or monthly) in a productive system. Inconsistent datasets between the OLTP systems and the CRM system are cleaned up in this process.

1.2 Data Flow during Request Download

Figure 1-1 shows a brief overview of the data flow during a request download.

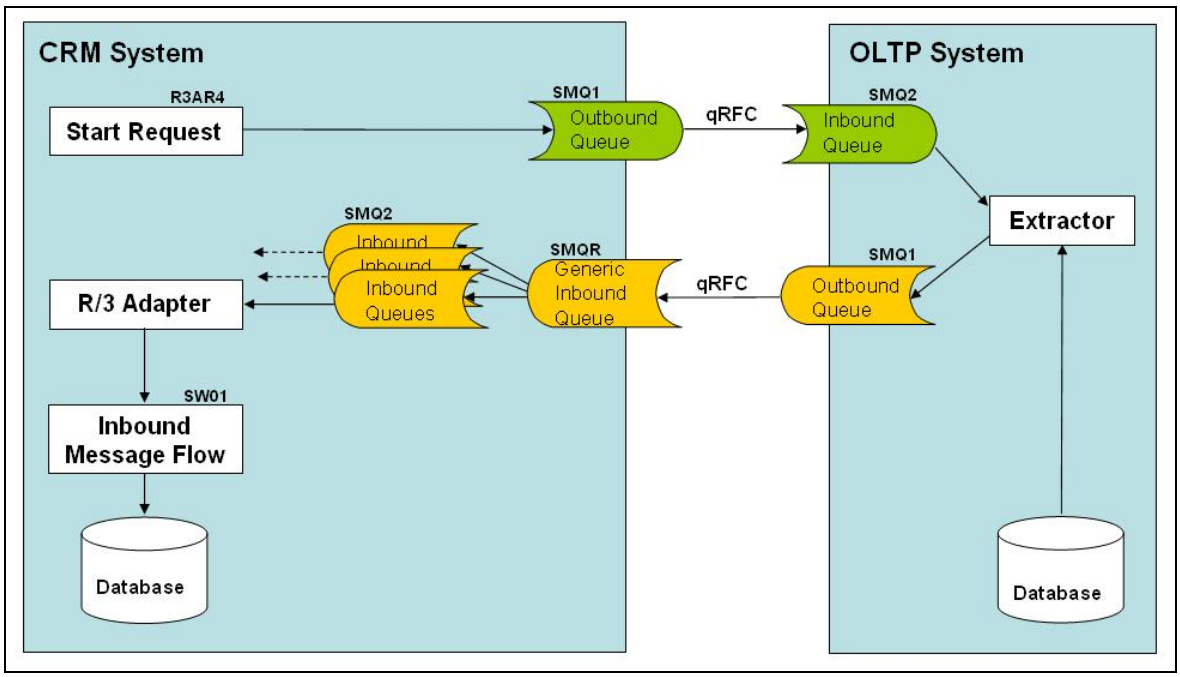


Figure 1-1: Data Flow Diagramm Request Download

The data flow consists out of the following steps (simplified):

- With the transaction R3AR4 (CRM Request Start) on the CRM system, you initiate a request download for a specified business object (adapter object) and send through the CRM outbound queue to the ERP inbound queue. The request remains in both queues until it finishes.
- In the ERP system, the Queue-In scheduler starts the configured extractor function module. The extractor module reads the requested business data and maps it into a BDoc. The filters you have defined in the request with transaction R3AR2 (Define CRM Request) are merged with the general adapter object filters that you can define with transaction R3AC1 (Adapter Object Overview).

  General adapter object filters for the table fields are ignored during a request if the filters of that request are defined for the same table fields. The reason for this is that the extractors combine filters for the same table fields with an OR statement. If the filter of the request is also combined with an OR statement, the result would be the entire data volume.
- The filled BDoc is sent in a LUW (logical unit of work) through the ERP outbound queue to the inbound queues of the CRM system using qRFCs.
- The Queue-In scheduler of the CRM system receives the LUW in a generic inbound queue and forwards it to one of the available inbound queues. You can configure the

maximum number of inbound queues the QIN scheduler may use (CRM_MAX_NO_QUEUES_PER_REQUEST parameter in the ERP system).

- The inbound message flow triggers the responsible CRM application in dependency of the adapter object. The application validates the data in the sent BDoc and updates the relevant database tables. After a successful validation, the outbound message flow is triggered to update further connected systems (sites).

1.3 Monitoring Tools

You can monitor the data flow with the following transactions:

- BDoc flow and CRM Middleware performance
  - Transaction R3AM1 (Monitor Load Objects)
  - Transaction R3AR3 (Monitor Requests)
  - Transaction SMW01 (Display BDoc messages)
  - Transaction SMQ1 (qRFC Monitor Outbound Queue)
  - Transaction SMQ2 (qRFC Monitor Inbound Queue)
  - Transaction SMWT (CRM Middleware Trace Monitor)
  - Transaction SMWP (CRM Middleware Portal)
- System performance and resource consumption, for example
  - Transaction ST06N (Operating System Monitor)
  - Transactions ST03 and ST03N (Workload Monitor)
  - Transaction ST04 (Database Monitor)
  - Transaction ST05 and ST01 (Traces)
  - Transaction SM50 and SM66 (Process Overview)
  - Transaction SMQR (qRFC Monitor QIN Scheduler)
  - Transaction SMQS (qRFC Monitor QOUT Scheduler)

The following chapters explain some of these transactions in more details.

1.3.1 Monitor Initial Download

With the initial download status monitor, you can monitor the status of an initial download from the ERP into the CRM system. You access the initial load monitor with transaction R3AM1 (Monitor Load Objects).

1.3.2 Monitor Request Download

With the request status monitor, you can monitor the status of a request download from the ERP into the CRM system. You access the request status monitor with transaction R3AR3 (Monitor Requests).

1.3.3 Display BDoc messages

You can monitor the processed BDocs (Business Document) with transaction SMW01. The column *State* shows the processing status of a BDoc and allows you to detect and analyze application or customizing errors during the data exchange. A status *green* means that the BDoc has been processed completely without an error. A status *yellow* indicates that the BDoc is currently being processed. A status *red* shows that the processing of the BDoc finished with an error. You can analyze the error messages and reprocess the BDoc if appropriate after resolving the reason of the error. There are further transactions available, to monitor BDocs:

- Transaction SMW01: Displays the BDoc message content and possible validation or receiver errors.
- Transaction SMW02: Displays a BDoc message summary in dependency on the sender site ID.

- Transaction SMW02A: Groups BDoc messages according to validation or receiver errors.
- Transaction SMW03: Display a client-independent BDoc message summary

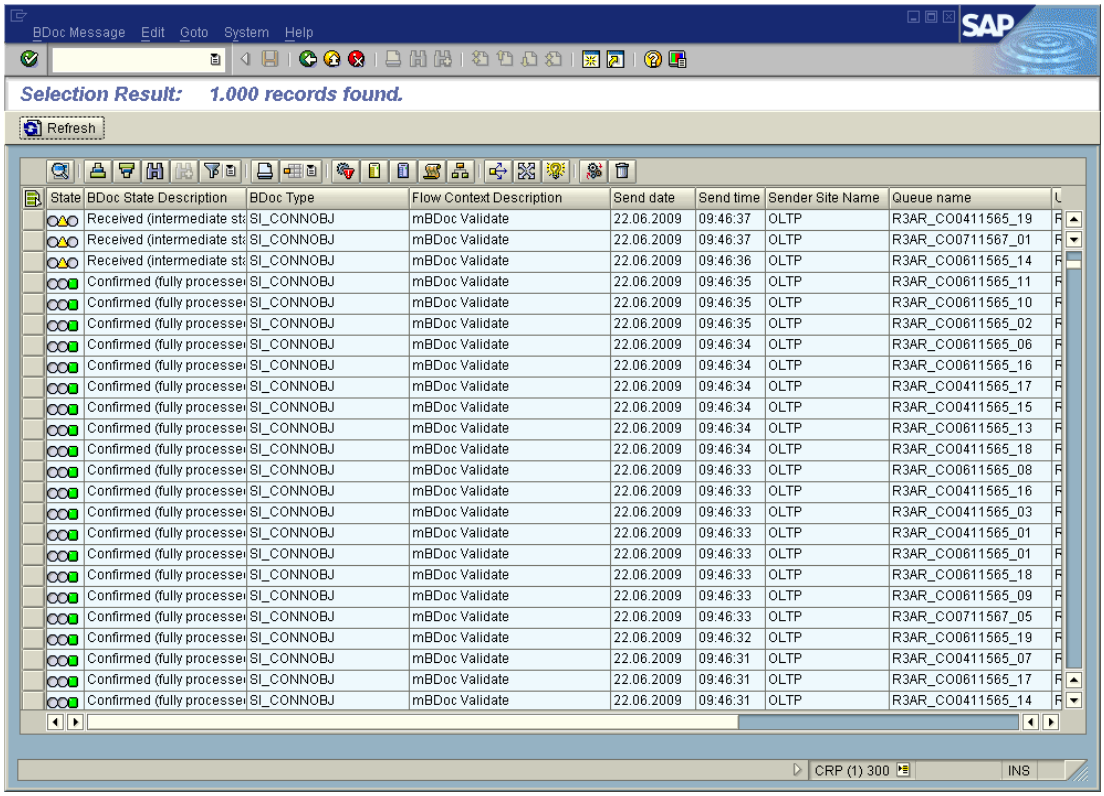


Figure 1-2: BDoc Message Monitor

1.3.4 Message Flow Statistics

The message flow statistics monitor collects statistical data about the workload on the CRM system caused by BDoc messages. You can use this monitor as a starting point for analyzing performance problems. You can monitor the message flow with transaction SMWMFLOW. You must make sure that the CRM Middleware message flow statistics are activated.

1.3.5 Outbound Queue Monitor

With the outbound queue monitor, you monitor the data transfer between the ERP and CRM system. You can find the LUW (Logical Unit of Work) that are sent to the corresponding system in the outbound queue of your system. You access the outbound queue monitor with transaction SMQ1 (qRFC Monitor Outbound Queue). Depending on the way a LUW is processed, an outbound queue can have different status. Figure 1-5 gives an overview of the possible status.

Queue Status	Description
READY	The queue is ready for transmission. This status should only be a temporary status. However, in the following case this status can also be a permanent status: A queue was locked manually with transaction SMQ1 or via a program and then unlocked without being activated at the same time. This queue must be activated explicitly.
RUNNING	The first LUW of this queue is currently being processed. If a queue in this status does not change the status for more than 30 minutes, this might mean that the work process responsible for sending this LUW has terminated. In this

	case, you can reactivate the queue. We recommend to wait for at least 30 minutes before reactivating the queue because an activation of a queue in status RUNNING may cause a LUW to be executed several times if this LUW is processed in the target system at the same time.
EXECUTED	The first LUW of this queue is processed. The system waits for an qRFC-internal confirmation from the target system before further LUWs are processed. If a queue in this STATUS hangs for more than 30 minutes, this might mean that the work process responsible for sending this LUW has terminated. In contrast to status RUNNING, this current LUW has been executed successfully. You can activate this queue again without any problems. The qRFC Manager will automatically delete the LUW already executed and send the next LUW.
SYSLOAD	At the time of the qRFC call, no DIALOG work processes were free in the sending system for sending the LUW asynchronously. A batch job for subsequent sending has already been scheduled (for more details, see SAP Note 319860).
SYSFAIL	A serious error occurred in the target system while the first LUW of this queue was executed. This error led to an interruption of the execution of the LUW. When you double-click on this status, the system displays an error text. You can find additional information on this error in the corresponding short dump in the target system (transaction ST22). No batch job is scheduled for a repetition and the queue is no longer processed. Information from the affected application is required to solve the problem (for more details, see SAP Note 335162).
CPICERR	During transmission or processing of the first LUW in the target system, a network or communication error occurred. When you double-click on this status, the system displays an error text. You can find additional information on this error in the system log (transaction SM21), the trace files dev_rd or dev_rfc*. Depending on the definition in transaction SM59 for the destination used, a batch job is scheduled for subsequent sending. The status CPICERR may also be shown although no communication error occurred. If an qRFC application detects an error and that a LUW cannot be processed any further, it calls the RESTART_OF_BACKGROUNDTASK function module in order to prompt the qRFC Manager to cancel the execution of this LUW and to repeat this LUW later in accordance with the specification in transaction SM59. In this case, qRFC simulates a communication error with the text *Command to tRFC/qRFC: Execute LUW once again.* If this error occurs very often, you must contact the corresponding application.
STOP	On this queue or a generic queue (for example, BASIS_*) a lock was set explicitly (with transaction SMQ1 or via a program). Note that the qRFC never locks a queue in its processing. After having informed the relevant application, you can unlock and activate this queue using transaction SMQ1.
WAITSTOP	The first LUW of this queue has dependencies to other queues, and at least one of these queues is currently still locked.
WAITING	The first LUW of this queue has dependencies to other queues, and at least one of these queues contains other LUWs with higher priorities.
NOSEND	LUWs of this queue are never sent but retrieved by a special application. These queues are only used internally at SAP (BW or CRM during communication with Mobile Clients). Even if a LUW was read by the corresponding application (BW, CRM), this status does not change. This LUW is only deleted from the queue if this application confirms collection (collective confirmation possible). Under no circumstances should this status be reset using transaction SMQ1 and the queue

	activated.
NOSENDS	During the qRFC call, the application determines at the same time that the current LUW is not sent immediately. This is used to debug the execution of an LUW with transaction SMQ1. Contact the corresponding qRFC application to clarify this status since this is either a programming or configuration problem.
WAITUPDA	This status is set if a qRFC is called within a transaction that also contains one or more update functions. This status blocks the LUW and and therefore the queue until the update has successfully terminated. If this status remains active longer than a few minutes, check the status of the update or the update requests using transaction SM13 (Monitor Update Requests). After a successful update, the blocked LUW is sent automatically. You can also restart the LUWs manually in the WAITUPDA status without a successful retroactive update (with transaction SMQ1, Reset status, Activate queue). However, to avoid possible inconsistencies, you may only carry out this action following consultation with the qRFC application (such as APO, BW, CRM). This WAITUPDA problem can be avoided as follows: If both qRFC calls and update calls occur within a transaction, qRFC must be executed exclusively within the update. In this case, the qRFC LUW is only created after the update has been completed successfully.
VBERROR	This status is set if a qRFC is called within a transaction that also contains one or more update functions. This status is set if the update not could be executed due to an error. Within the inbound queue monitor, you can branch directly to transaction SM13 (Monitor Update Requests) on the LUW level by double-clicking on the error message, to determine the cause of the update termination.
RETRY	During LUW execution, the application has diagnosed a temporary problem and has used a specific qRFC call to prompt the qRFC manager in the sending system to schedule a batch job. This batch job schedules a repetition after two minutes. You can use the definition in transaction SM59 (TRFC options) to suppress this batch job.
ARETRY	During LUW execution the application has diagnosed a temporary problem and has used a specific qRFC call to prompt the qRFC Manager in the sending system to schedule a batch job for a repetition on the basis of the definition in transaction SM59 (Configuration of RFC Connections).
ANORETRY	During the LUW execution, the application has diagnosed a serious error and prompted the qRFC Manager via a specific qRFC call to cancel processing of this LUW. Information from the affected application is required to solve the problem.
MODIFY	Processing of this queue is locked temporarily because the LUW data is being modified.

Figure 1-3: Status of an Outbound Queue

1.3.6 Inbound Queue Monitor

With the outbound queue monitor, you can monitor the data transfer between the CRM and the ERP systems. You can find the LUW (Logical Unit of Work) sent from the corresponding system in the inbound queue of your system. You access the inbound queue monitor with transaction SMQ2 (qRFC Monitor Inbound Queue).

Figure 1-4 shows the inbound queue monitor with 24 active inbound queues. The prefix R3AR of the queue names indicates that they have been created by requests. In fact, there

are two separate requests running: the CO04 request and the CO06 request. The queues are processing 579 LUWs in total.

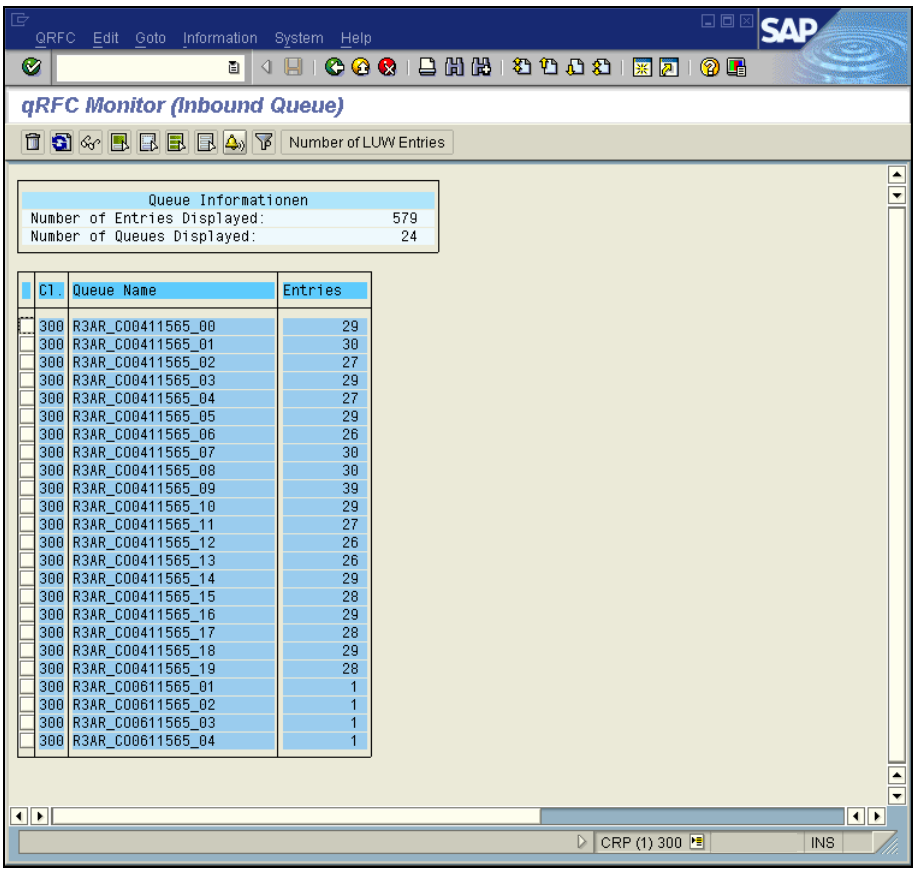


Figure 1-4: Inbound Queue Monitor

Depending on the way a LUW is processed, an inbound queue can have different status. Figure 1-5 gives an overview of the possible status of an inbound queue.

Queue Status	Description
READY	The queue is ready for processing. This status should only be a temporary status. However, in the following case this status can also be a permanent status: A queue was locked manually with transaction SMQ2 or via a program and then unlocked without being activated at the same time. This queue must be activated explicitly.
RUNNING	The first LUW of this queue is currently being processed. If a queue in this status hangs for more than 30 minutes, this may mean that the work process responsible for sending this LUW has terminated. In this case, you can activate this queue again. Note that activating a queue in status RUNNING may cause a LUW to be executed several times if this LUW is processed in the target system at that time. We therefore recommend a waiting time of at least 30 minutes before you activate the queue again.
SYSFAIL	A serious error occurred in the target system while the first LUW of this queue was executed. The execution was interrupted. When you double-click on this status, the system displays an error text. You can find additional information on this error in the corresponding short dump in the target system (transaction ST22). No batch job is scheduled for a repetition and the queue is no longer processed. Information from the affected application is required to solve the

	problem. Refer to SAP 335162 for the special error text " connection closed".
CPICERR	During transmission or processing of the first LUW in the target system, a network or communication error occurred. When you double-click on this status, the system displays an error text. For additional information on this error, refer to the dev_rd or dev_rfc* trace files in the syslog (transaction SM21). being referred for to dev_rfc*. Depending on the registration of this queue (transaction SMQR), a batch job is scheduled for repetition. Refer to SAP Note 369524 for the error text "R/3 logon failed". Status CPICERR may also occur if a qRFC application finds out that a LUW cannot be processed any further due to a temporary error in the application. Therefore, it calls the RESTART_OF_BACKGROUNDTASK function module in order to prompt the qRFC manager to cancel the execution of this LUW and to repeat this LUW later in accordance with the specification in transaction SM59. In this case, qRFC simulates a communication error with the text "Command to tRFC/qRFC: Execute LUW once again." If this error occurs very often, you must contact the corresponding application.
STOP	On this queue or a generic queue (for example, BASIS_*) a lock was set explicitly (transaction SMQ2 or programs). Note that the qRFC never locks a queue in its processing. After having informed the corresponding application, you can unlock and activate this queue using transaction SMQ2.
WAITSTOP	The first LUW of this queue has dependencies to other queues, and at least one of these queues is currently still locked.
WAITING	The first LUW of this queue has dependencies to other queues, and at least one of these queues contains other LUWs with higher priorities.
NOEXEC	During the qRFC call, the application simultaneously determines that the current LUW is not processed automatically even if the queue to the QIN Scheduler (transaction SMQR) is registered. This is used to debug the execution of an LUW with transaction SMQ2. Contact the corresponding qRFC application to clarify this status since this is either a programming or configuration problem.
RETRY	During LUW execution, the application has diagnosed a temporary problem and has used a specific qRFC call to prompt the qRFC manager in the sending system to schedule a batch job. This batch job schedules a repetition after two minutes.
ARETRY	During LUW execution the application has diagnosed a temporary problem and has prompted the qRFC Manager in the sending system via a specific qRFC call to schedule a batch job for a repetition on the basis of the registration in transaction SMQR.
ANORETRY	During the LUW execution, the application has diagnosed a serious error and has used a specific qRFC call to prompt the qRFC Manager to cancel processing of this LUW. Information from the affected application is required to solve the problem.
MODIFY	Processing of this queue is locked temporarily because the LUW data is being modified.

Figure 1-5: Status of an Inbound Queue

Figure 1-6 shows the inbound queue monitor with more details per queue (button ). All the queues are in status RUNNING.

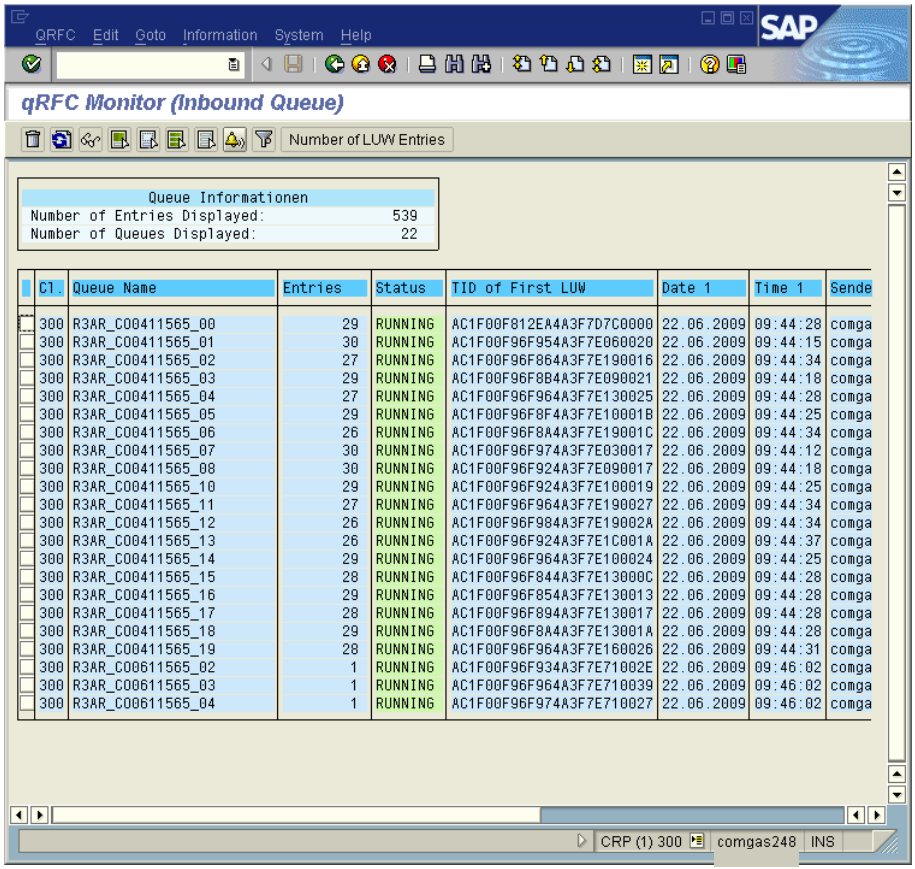


Figure 1-6: Inbound Queue Monitor (detailed view)

1.3.7 CRM Middleware Monitoring Cockpit

The CRM Middleware Monitoring Cockpit is the CRM Middleware centralized monitoring tool. It is based on the CCMS qRFC monitoring. This tool provides an overview over all the CRM Middleware specific tools. In addition, it monitors the status of background jobs and important system settings associated with the data exchange between the CRM and ERP systems. You access the monitor cockpit with the transaction SMWP (CRM Middleware Monitoring Cockpit).

The monitoring cockpit displays the status of the different CRM Middleware processes associated with the components involved (CRM Server, R/3 backend, mobile clients). To ensure that the business processes and the technical data exchange are working correctly, you must ensure that the traffic light is green for the different parts of the monitoring cockpit. In some cases, the traffic light is yellow that indicates a warning or information.

You can monitor runtime information as

- Message processing active
- Data Exchange using qRFC Queues
- Adapter Status Information
- Initial Load Status
- Request Status
- Parameters in R/3 Backend(s)
- CRM Middleware BDoc Messages in the Flow
- CRM Middleware Queues
- CRM Background Jobs
  - CRM Middleware Reorganization
  - Collector for Monitoring Cockpit
  - Collector for BDoc Messages/Site Statistics

- Check Generation Status of Objects
- Periodical Background Generation
- Administration Console Subscription Agent
- Administration Console Site Scheduling

You must schedule some jobs that collect the status information displayed in the monitoring cockpit. To schedule the jobs *daily*, call the transaction `SMWP` and open the ***Background jobs*** node. There you find a list of the recommended jobs and their scheduling status. To schedule a job, Use the secondary mouse button to click on a node and choose ***Show Suggested Action***. Then push the ***Action*** button.

You can find further information about the Monitoring Cockpit in the ***CRM Operations Guide*** in the SAP CRM space of the ***SAP Service Marketplace*** (http://service.sap.com/).

2 Test Planning and Execution

2.1 Motivation

The initial download is the most critical and challenging data transfer procedure between the ERP and the CRM system during the go-live of the CRM system. This is, because a large amount of business objects has to be transferred (replicated) using the CRM Middleware from the ERP system into the CRM system in a restricted period. The objective of the test cycles is to minimize this period as much as possible in order to reduce to a minimum the total cutover period during a go-live.

2.2 Planning of Test Cycles

The improvement of the performance requires some test cycles for performance analysis, implementation of changes to improve the performance followed by a verification whether the change resolved the identified bottleneck and actually improved the performance. The project must plan for test cycles, which can from last from several days up to even weeks in the beginning of the tests. You should consider multiple test phases up to several weeks in dependency of the data volume and the relevant business objects to achieve the desired objective.

Ideally, you use the production system environment for your test activities or at least a production-like environment. Your system environment should not share system resources (for example, database, CPU, storage, etc.) with other systems (for example, development system, quality system, etc.) where concurrent activities take place during your tests. Migration activities in your ERP system affect also directly the replication performance.

When you plan for multiple replication cycles, you should plan to refresh the CRM system between the cycles. It is best practices to maintain a golden backup of a fully setup ERP and CRM system. The golden backup reduces the preparation of a test cycle to a minimum. Also the test results become comparable also from a database prospective because the starting database statistics remain the same when you start your test cycle. Specifically the usage of multiple clients in your CRM system affects the results because certain tuning activities (for example, the buffering of database tables in memory, database statistics) are client-independent.

The following chapters give you an overview how to plan and execute the required test phases in order to achieve an adequate replication performance. The used transactions are explained in more details in own chapters.


It is important to create and maintain an activity log from the beginning to protocol all observations and occurrences and to keep track of all changes and their respective results. As a result, you create a checklist that includes all activities to turn the systems into a system that is ready for a high performance initial download and to turn it back into normal production operation on completion of the initial download.

2.3 Execution of Unit Test

The objective of the unit test is to test the system setup, the Customizing of the download objects and the successful replication of some few business objects. You should execute a unit test for the initial download in the following steps. All transactions are to be executed in the CRM system unless mentioned otherwise.

1. Setup and replicate the Customizing data with the transaction R3AS (Start Initial Load).
2. Setup and replicate one business object with transaction R3AS (Start Initial Load)

   You need to define a filter with transaction R3AC1 (Adapter Object Overview) for the business object, you want to replicate.

The initial download updates some system database tables (for example, the business transactions events in database table TBE31) in order to trigger the online download of delta information to synchronize the information in the ERP and the CRM system.

If you are performing test runs using an already productive ERP system, please ensure that the event entries are deleted after the test initial download is finished. Therefore, use transaction R3AC4 (Object Class Activation) and set the flag "inactive" for the corresponding download object. Do not forget to remove the inactive flag for the corresponding objects immediately before you start the productive initial download. Otherwise, the delta information will not be downloaded from the ERP system into CRM that leads to inconsistency between the two systems.

3. Monitor the replication process in the CRM system with transaction R3AM1: Monitor Load Objects
4. Setup and replicate very few business objects with the transactions R3AR2 (Define CRM Request) and R3AR4 (CRM Request Start).

   Prior to replicating business objects, you may need either to delete a previously defined general filter or define a general filter in the CRM system with transaction R3AC1 (Adapter Object Overview) for the overall range of the business object, you want to replicate.
5. Monitor the replication process in the CRM system
   - Transaction SMW01: Display BDoc messages
   - Transaction SMQ1: qRFC Monitor (Outbound Queue)
   - Transaction SMQ2: qRFC Monitor (Inbound Queue)
6. Verify the replicated business object in the CRM system with the respective online transaction.

   Check for errors that occurred during the download. Errors may occur if the data downloaded into the CRM system is not formatted in the way expected by the CRM system. For example, when downloading business partners, the phone number must be in a specific format. If the phone number has a different format in the ERP system than in the CRM system, an error will occur during the initial download of this business partner, as the data cannot be entered into the CRM system. Correct the errors that occurred during the replication and repeat the download to check whether the error correction worked. Repeat the tests until errors no longer occur.
7. Setup and replicate business objects with high volumes with the transactions R3AR2 (Define CRM Request) and R3AR4 (CRM Request Start).
8. Monitor the replication process
   - Transaction SMW01 (Display BDoc messages)
   - Transaction SMQ1 (qRFC Monitor Outbound Queue)
   - Transaction SMQ2 (qRFC Monitor Inbound Queue)

   Check for errors that occurred during the download. Correct the errors that occurred during the replication and repeat the download to check whether the error correction worked. Repeat the tests until errors no longer occur.
9. Verify the replicated business object in the CRM system with the respective online transaction.

Repeat steps 2 to 9 for all download objects.

2.4 Execution of Volume Test

The objective of the volume test is to test and improve the replication performance of an initial download of all relevant business objects. You should only start volume tests after you succeeded in replicating successfully the relevant business objects during the unit test. You need to plan for several test cycles for each download object.

You should execute the volume test in three steps per business object: In phase 1, you prepare the system for the volume test and determine and set the initial parameter values. In the second phase, you process a single request at a time, analyze the execution, and improve the performance of the request. After the completion of this phase, the request performs with a good performance but still not all available CRM system resources are used. In the final phase, you process requests concurrently in order to max out the available CRM system resources.

The objective of the monitoring of the system in phase two and three is the identification of potential system bottlenecks, for example, unnecessary creation of change documents, incorrect usage of the database indexes, and avoidance of exhaustion of system resources like dialog work processes and CPU processing capacity. You monitoring activities should focus on two main areas (1) BDoc flow and CRM Middleware performance and (2) system performance and resource consumption.

Phase 1: System preparation

1. Determination of the number range of the business objects you want to replicate. For this analyze the identifier of the business objects in the ERP system and determine the lower and upper identification limit (replication interval).
2. Either delete a previously defined general filter or define a general filter in the CRM system with transaction R3AC1 (Adapter Object Overview) for the replication interval of the business object, you want to replicate
3. Define the CRM Middleware parameter CRM_MAX_NO_QUEUES_PER_REQUEST (CRMPAROLTP database table in the ERP system) for the business object you want to replicate and set the parameter to 10. This will allow up to 10 concurrent processes per request (inbound queues) in the CRM system. This number will be optimized in the further steps.
4. Define the CRM Middleware parameter MAX_PARALLEL_PROCESSES (SMOFPARSFA database table in the CRM system) with transaction R3AC6 with the R3A_COMMON key and set the parameter to 99. This will allow you to process up to 99 requests concurrently.

Phase 2: Analysis of the performance of a single request

1. Define a request with transaction R3AR2 (Define CRM Request) using an interval that includes about 2.000 business objects. The idea is to process the request with 2.000 business objects within approximately 10 minutes. If this number is not accurate in your system, use a higher or lower number of business objects to be replicated.
2. Start the defined request with transaction R3AR4 (CRM Request Start)
3. Monitor the replication process in both ERP and CRM systems.
4. Implement changes to resolve the identified bottlenecks and repeat step 1 to 4 until the request is processed with an acceptable performance.
5. Determine the maximum number of concurrent active inbound queues in the CRM system with transaction SMQ2 (qRFC Monitor Inbound Queue).

Phase 3: Analysis of the performance of multiple concurrent requests

1. Adjust the CRM Middleware parameter CRM_MAX_NO_QUEUES_PER_REQUEST (CRMPAROLTP database table in the ERP system) to the maximum number of monitored concurrent inbound queues in the CRM system.
2. Adjust the operation mode of the CRM system to make a high number of dialog work processes available if necessary. During the initial download, there should be no need for neither background nor update work processes. The CRM Middleware parameter CRM_MAX_NO_QUEUES_PER_REQUEST defines also the minimum number of required dialog work processes per request in the CRM system.
3. Define an RFC logon group to control the distribution of the RFC on the CRM system while processing the request download and assign this RFC logon group to the QIN scheduler (for more details, see chapter 6.5.3 *Distribution of RFCs).* Initially, exclude the central instance from the RFC logon group.
4. Define more requests each of them with an interval size that will allow a processing time per request of about 10 minutes with transaction R3AR2 (Define CRM Request).
5. Start more than one (for example, initially three) of the defined requests at the same time and monitor the performance and the resource consumption, especially in the CRM system.
6. Implement appropriate changes to resolve the identified bottlenecks. Repeat the initial download with a higher number of concurrent requests until either all application servers are busy or the database server becomes close to be overloaded. If the application servers are busy but there are still resources available on the central instance, include the central instance in the RFC logon group of the QIN scheduler.
7. Define requests with larger intervals and start them. The objective is to have one request finished within about 30 minutes in order to optimize the overall runtime of a request.

You can measure the processing time of a BDoc easily with transaction SMW01 (Display BDoc messages). Choose the ***5 Minutes*** as the ***Send Date and Time*** interval and ***No Limitation*** as number of hits. Multiply the number of displayed BDocs by the block size defined with transaction R3AC1 (Define Business Object) and then by 12 (5 minutes = 1/12 hour). The result is the number of processed business objects per hour.

Measure the runtime for the initial download for all download objects with a high volume. This is essential for estimating the time necessary to download all business objects and to optimize the setup for parallel processing. Please keep in mind that the extrapolation is only a rule of thumb because the performance may change with an increasing database.

On the CRM system, monitor closely the maximum number of active inbound queues (transaction SMQ2). If you can see a queuing of the LUWs for the inbound queues and there are still system resources available, increase the CRM_MAX_NO_QUEUES_PER_REQUEST parameter to a higher and more appropriate number of inbound queues.

2.5 Test Mode

Be aware that you cannot run the initial download in test mode. This means that during each execution of an initial download, the data is actually inserted into the CRM database. In addition, tables are updated in the ERP system. There is no standard tool available to delete the created data in the CRM system and to reverse the updated data in the ERP system after a test run. This means that you have either to perform test runs with different filter settings for the same business object or to delete the data manually before performing another test run. An easier solution is to backup the systems before the initial download test and restore the backup before starting the next test cycle.

2.6 Reconciliation and Data Validation

2.6.1 Reconciliation

It is compulsory to compare the number of business objects to be replicated to the number of actually replicated objects. If the determined volumes do not match, you need to analyze what might have caused the mismatch. Beside an erroneous definition of the request filters, such as gaps between the intervals or overlap of the interval boundaries, functional or even technical problems could have prevented a complete data replication.

You must include in your analysis not only the error messages of the related BDocs (transaction `SW01`) but also the replication data flow, such as the status of all inbound and outbound queues of both ERP and CRM system and also BDocs in status *pending* in one of the queues.

The simplest way is to compare at least the number of rows in the header table of each kind of business object in both ERP and CRM system with transaction `SE16`. Figure 2-1 shows the header tables of the most important business objects in both ERP and CRM system.

Adapter Object	Description	ERP System	CRM System
BUPA_MAIN	Business partner	BUT000	BUT000
BUPA_REL	BP relationship	BUT050	BUT050
BUAG_MAIN	Contract account	FKKVKP	CRMM_BUAG
SI_CONNOBJ	Connection object	EHAUISU	ISU_CONNOBJ
SI_POD	Point of delivery	EUIHEAD	ISU_POD
SI_CONTRACT	Contract	EVERH	CRMD_ORDERADM_I

Figure 2-1 Header Tables of Business Objects

2.6.2 Data Validation

Equally important to a reconciliation of the replicated data is the validation of the created or changed business objects on the CRM system. Not necessarily, the application could identify a problem with the data sent from the ERP system and processed on the CRM system. Therefore, you must visualize the replicated business objects on both ERP and CRM system and compare the displayed field contents manually. Because you cannot perform a manual comparison of all replicated objects due to the high volume, you should validate a manageable number of representative business objects.

2.6.3 Data Integrity Manager (DIMa)

The Data Integrity Manager (DIMa) in the CRM system can help you to detect and repair inconsistencies between objects of the ERP and CRM system. The DIMa compares data in different components and displays inconsistencies. There are two compare types available in the DIMa:

- **Header Compare**: A header compare checks, if an object instance exists in both systems. Some objects may not allow a header compare. The detail compare is then performed instead.
- **Detail Compare**: A detail compare compares all data of an object instance found in both databases.

You can start the DIMa with the transaction `SDIMA` on the CRM system.

The comparison with the DIMa is very time consuming and may lead to an extended validation period. Instead of performing a data validation with the DIMa, you may choose to implement a manual data validation as described in the previous chapter.

3 Technical System Preparation

3.1 Preliminary Tasks

Before analyzing the performance of a replication, ensure that you perform at least the following preliminary tasks or checks in your system:

- o Complete all installation and post-installation actions and procedures including customizing
- o Apply all SAP recommendations from SAP service sessions and recommendations resulting from customer problem messages.
- o Implement all current SAP Support Packages upon availability.
- o Implement the highest available Plug-In and patches on the ERP and CRM system

You can use the following chapters to build your own checklist to setup technically the ERP system and CRM system for the replication. They provide some basic steps to prepare the system for a replication. The screen shots are taking during the preparation of a system after a client copy. The new client in the CRM system QCS is 202, the new client in the ERP system QES is 200.

3.1.1 Logical System

You must pay special attention to the definitions of the logical systems. A logical system is a system in which applications run integrated on a shared data basis. The distribution of data between systems requires that each system in the network has an unique identification. The logical system is used for this purpose. In the SAP system, the client corresponds to a logical system. To assign a logical system to a client, call transaction SCC4 (Client Maintenance) and choose *Goto* → *Detail*.

The logical system must be unique company-wide. It must not be used by any other system in an ALE (Application Link Enabling) integrated group, for example an integration by RFCs. You must not change the logical system in a production system. If the logical system of a document reference is not set to initial and does not match your own, the system assumes the document is located in a different system.

When you setup a system from scratch, the definition of the logical system is part of the setup process. Especially after a client copy in your test system it is important to verify, whether the correct logical system name for your client is defined.

3.1.2 RFC Connections

A remote function call (RFC) is the call of a function module that runs in a different system to the calling program. In an SAP system, the RFC interface system provides this function. The RFC interface system enables function calls between two SAP systems, or between an SAP system and an external system.

When calling a function module using the RFC interface, the calling program must specify the parameters of the connection in the form of a destination. This destination defines the type of connection, the partner program, and the target system. You can manage destinations with transaction SM59.

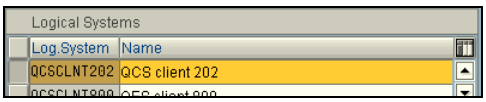
There are two types of RFCs available: tRFC (transaction RFC and qRFC (queued RFC). The main difference is that qRFCs allow a serialization of tRFC using wait queues. The CRM Middleware uses qRFC (parallel RFCs) for the data exchange between the ERP and CRM systems. As described in the chapters 1.3.5 *Outbound Queue Monitor* and 1.3.6 *Inbound Queue Monitor*, the outbound and inbound scheduler executes the RFC LUWs (Logical Unit of Work). The scheduler checks the resources and executes parallel RFCs, if resources are available. If no resources are available, the scheduler waits until resources (mainly dialog work processes) become available.

3.2 CRM System

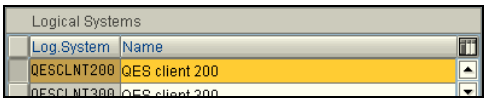
3.2.1 Check/Create Logical Systems

Figure 3-1 shows the procedure to define the logical systems for a data exchange for both CRM and ERP systems on the CRM system.

1. Call transaction BD54 (Change View Logical Systems)
2. Verify whether a logical system for your CRM system is already defined. Push the ***New Entries*** button if the logical system does not exist. Enter the name of the logical system, for example QCSCLNT202 (<CRM system ID>CLNT<client>) and save the definition.



3. Verify whether a logical system for your ERP system is already defined. Push the ***New Entries*** button if the logical system does not exist. Enter the name of the logical system, for example QESCLNT200 (<ERP system ID>CLNT<client>) and save the definition.



4. Verify whether the defined logical system is maintained correctly in the client setting of the CRM system. Call transaction SSC4 (Client Maintenance) and select the client. Then choose *Goto* → *Detail*. Maintain the logical system if necessary.

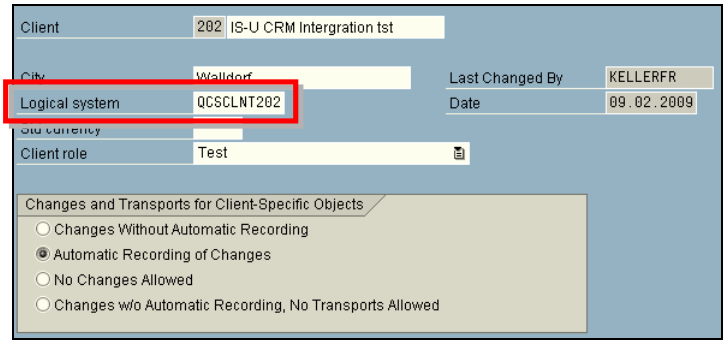
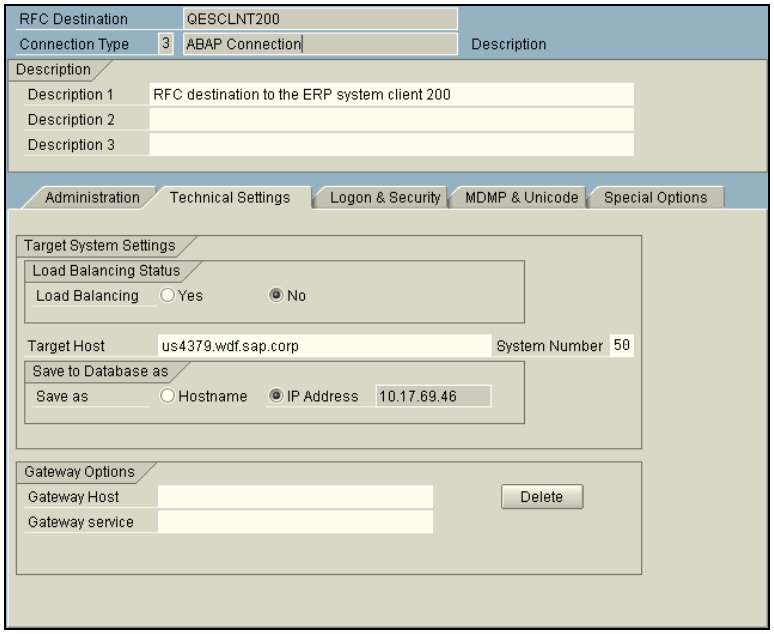


Figure 3-1: Definition of the Logical System of CRM Client 202

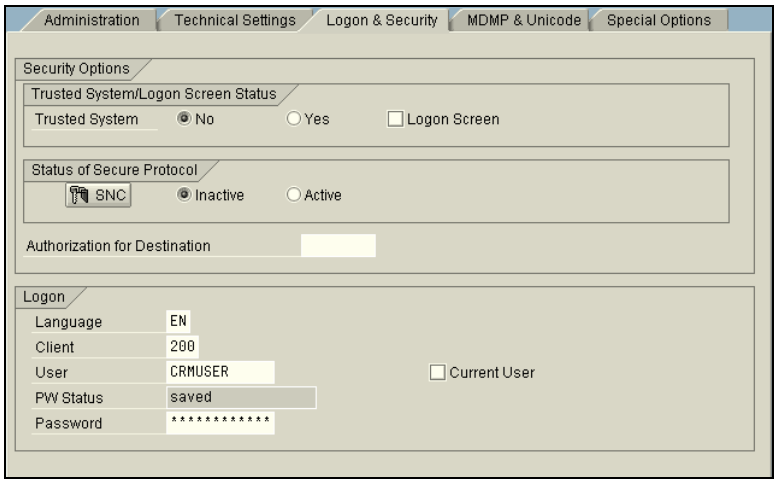
3.2.2 Check/Create RFC destination to ERP system

Figure 3-2 shows the procedure to define the RFC destination to the ERP system QES.

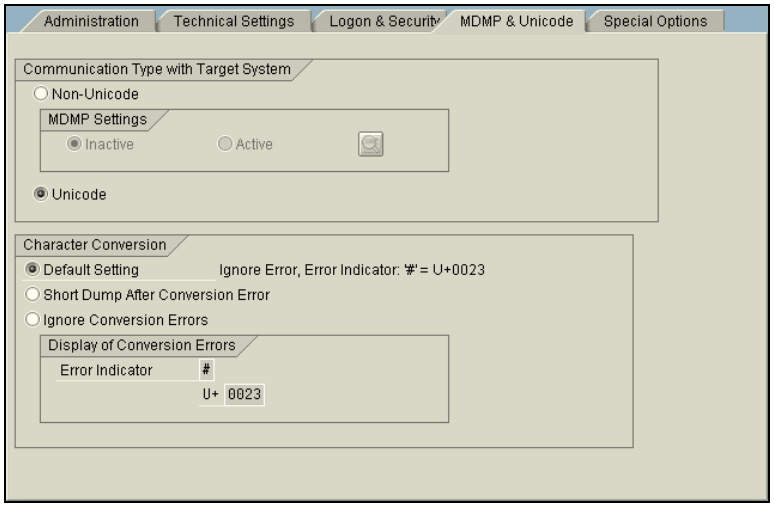
1. Call transaction SM59 (Configuration of RFC connections) and choose *Edit* → *Create* or push the ***Create*** button.
2. Enter the name of the RFC destination (for example QESCLNT200), the ***connection type*** *3* (ABAP connection) and a ***description***. The target system settings are ***Load Balancing*** *No*, the name of the ***target host*** and the ***system number*** of the ERP system.



3. Switch to sub screen ***Logon & Security*** and enter ***Trusted System*** *No*, ***secure protocol*** *Inactive* and the logon data as defined in the ERP system.



4. Switch to sub screen ***MDMP & Unicode*** and select the ***Unicode*** indicator. You can only change the ***Unicode*** indicator if you have a 2-byte unicode codepage installed.



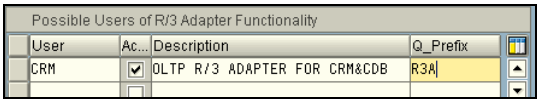
5. Save the new RFC destination.

Figure 3-2: Definition of the RFC destination to the ERP system client 200

3.2.3 Create CRM user of R/3 Adapter Functionality

Figure 3-3 shows the procedure to define the CRM user of R/3 adapter functionality.

1. Call transaction SM30 (Maintain Table Views) for CRMCONSUM table.
2. Verify whether a parameter set for your user CRM system is already defined. Push the ***New Entries*** button if the user CRM does not exist yet.
3. Enter the following data: ***User***: *CRM*, ***Indicator: Application Active*** *X*, ***Description***: a descriptive text, for example OLTP R/3 ADAPTER FOR CRM&CDB, ***Q_Prefix***: *R3A*.



4. Save the parameter

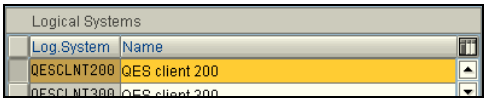
Figure 3-3: Definition of the CRM user of R/3 Adapter Functionality

3.3 ERP System

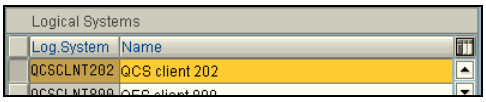
3.3.1 Check/Create Logical Systems

Figure 3-4 shows the procedure to define the logical systems for a data exchange for both CRM and ERP systems on the ERP system.

1. Call transaction BD54 (Change View Logical Systems)
2. Verify whether a logical system for your ERP system is already defined. Push the ***New Entries*** button if the logical system does not exist. Enter the name of the logical system, for example QESCLNT200 (<ERP system ID>CLNT<client>) and save the definition.



3. Verify whether a logical system for your CRM system is already defined. Push the ***New Entries*** button if the logical system does not exist. Enter the name of the logical system, for example QCSCLNT202 (<CRM system ID>CLNT<client>) and save the definition.



4. Verify whether the defined logical system is maintained correctly in the client setting of the ERP system. Call transaction SSC4 (Client Maintenance) and select the client. Then choose *Goto* → *Detail*. Maintain the logical system if necessary.

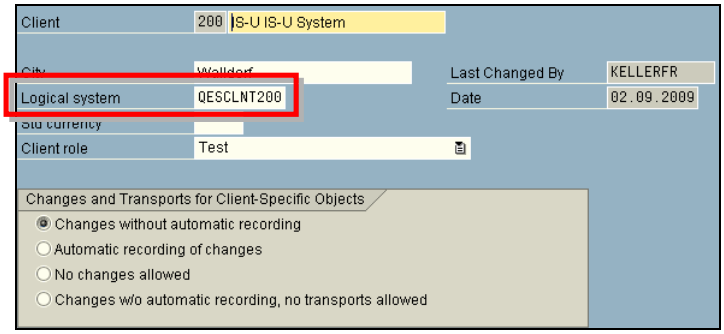
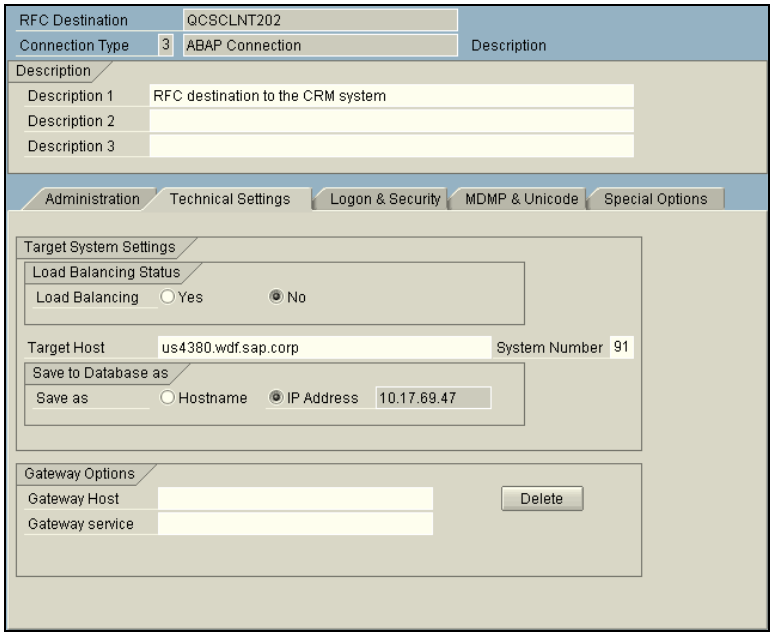


Figure 3-4: Definition of the Logical Systems in the SAP ERP system

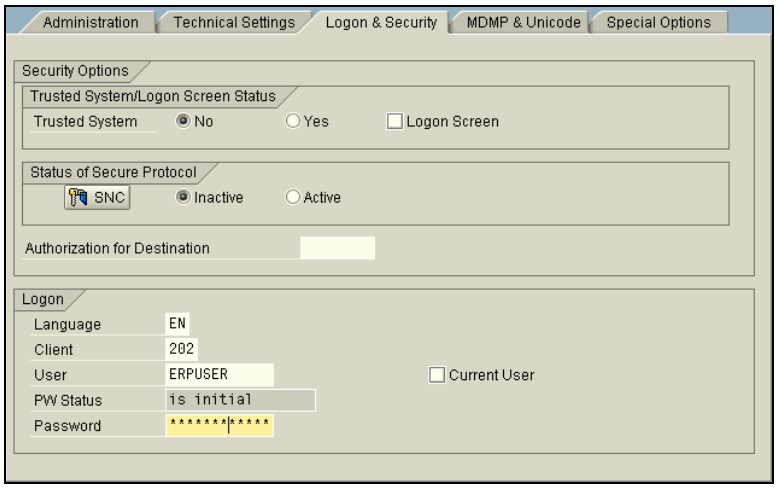
3.3.2 Check/Create RFC connection to CRM system

Figure 3-5 shows the procedure to define the RFC destination to the CRM system QCS.

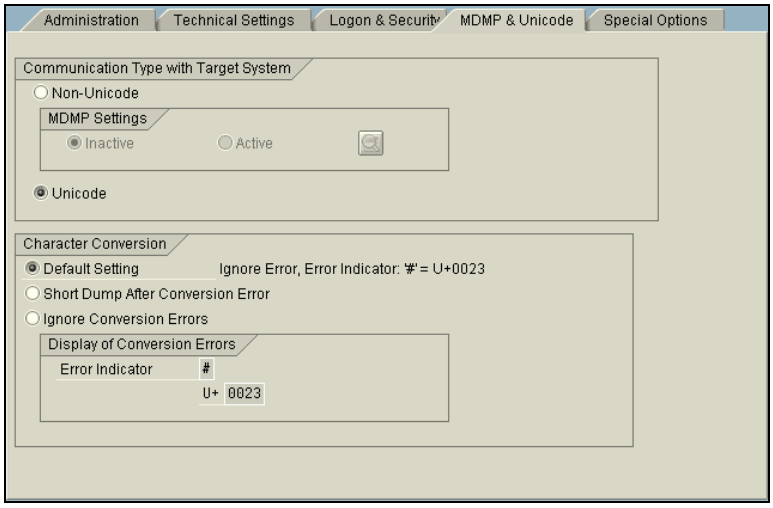
1. Call transaction SM59 (Configuration of RFC connections) and choose *Edit* → *Create* or push the ***Create*** button.
2. Enter the name of the RFC destination (for example QCSCLNT202), the ***connection type*** *3* (ABAP connection) and a ***description***. The target system settings are ***Load Balancing*** *No*, the name of the ***target host*** and the ***system number*** of the CRM system.



3. Switch to sub screen ***Logon & Security*** and enter ***Trusted System*** *No*, ***secure protocol*** *Inactive* and the logon data as defined in the CRM system.



4. Switch to sub screen ***MDMP & Unicode*** and select the ***Unicode*** indicator. You can only change the ***Unicode*** indicator if you have a 2-byte unicode codepage installed.



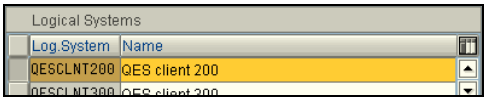
5. Save the new RFC destination.

Figure 3-5: Definition of the RFC destination to the CRM system (client 202)

3.3.3 Check/Create logical CRM Connection

Figure 3-6 shows the procedure to define the logical systems for both CRM and ERP systems in the ERP system.

1. Call transaction BD54 (Change View Logical Systems)
2. Verify whether a logical system for your ERP system is already defined. Push the ***New Entries*** button if the logical system does not exist,. Enter the name of the logical system, for example QESCLNT200 (<ERP system ID>CLNT<client>) and save the definition.



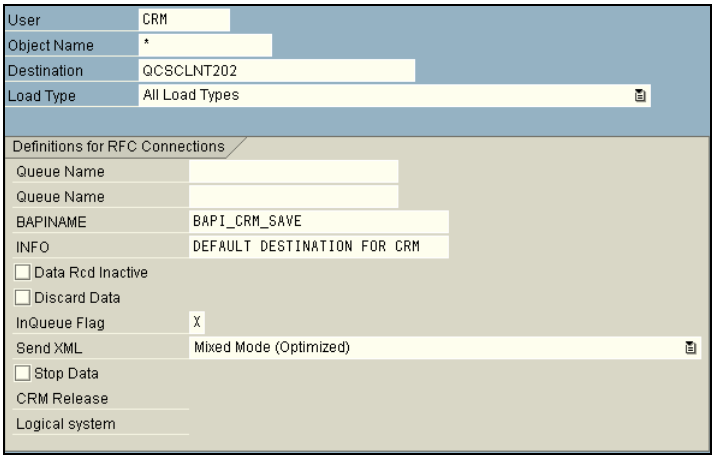
3. Verify whether a logical system for your CRM system is already defined. Push the ***New Entries*** button if the logical system does not exist,. Enter the name of the logical system, for example QCSCLNT202 (<CRM system ID>CLNT<client>) and save the definition.

Figure 3-6: Definition of the Logical Systems in the SAP ERP system

3.3.4 Create data exchange parameter between ERP and CRM systems

Figure 3-7 shows the procedure to define the CRM Middleware data exchange parameters in the ERP system. See chapter 7.1.2 *Important Parameters in CRMRFCPAR Table* for more details.

1. Call transaction SM30 (Maintain Table Views) for CRMRFCPAR table.
2. Verify whether a parameter set for your CRM system is already defined. Push the ***New Entries*** button if the user CRM does not exist yet.
3. Enter the following data: ***User***: *CRM,* ***Objectname*** **,* ***Destination***: the name of the logical system, for example *QCSCLNT202* (<CRM system ID>CLNT<client>), ***BAPINAME***: *BAPI_CRM_SAVE,* ***INFO***: a descriptive text, ***InQueueFlag***: *X,* ***Send XML***: *Mixed Mode (Optimized).*



4. Save the parameter

Figure 3-7: Definition of the exchange parameter for the CRM system

4 Functional System Preparation

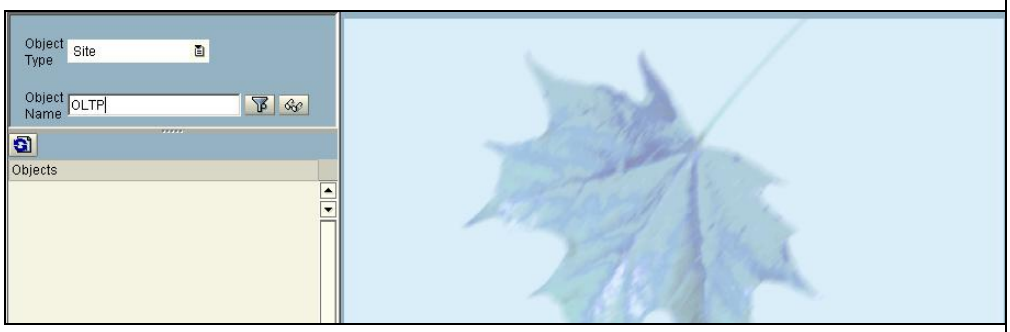
The following chapters show how to create a basic configuration in the CRM and SAP ERP system. Of course, the production Customizing is much more complex, for example, the number of required services and products.

4.1 CRM System

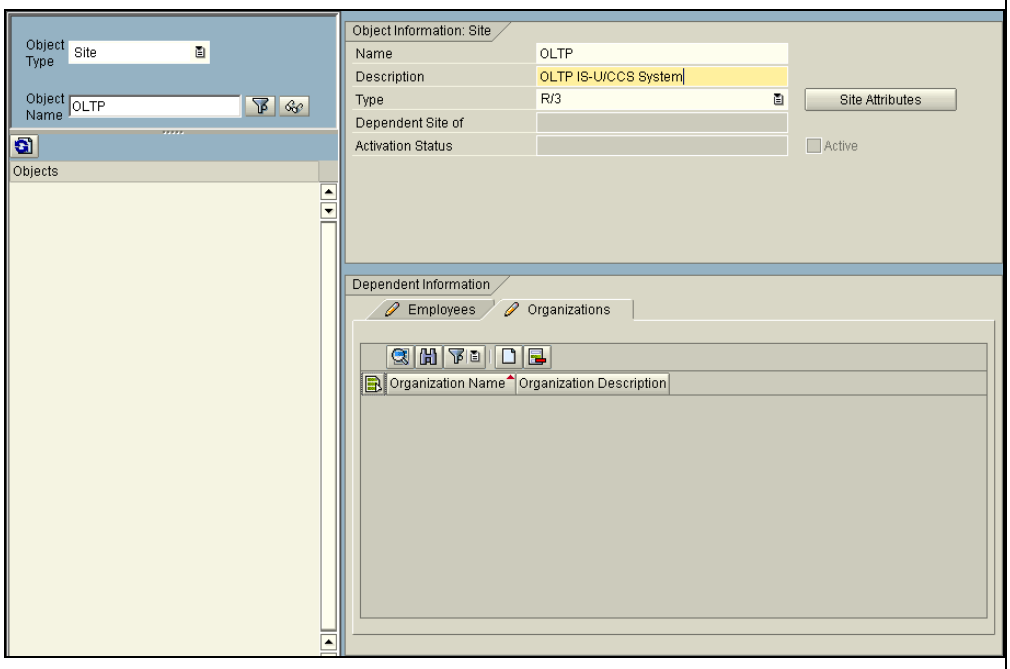
4.1.1 Check/Create OLTP Site

An OLTP site is a physical dataset and is defined on the CRM server. It acts as receiver or supplier of data in the context of replication. Figure 4-1 shows the procedure to define the OLTP site in the CRM system.

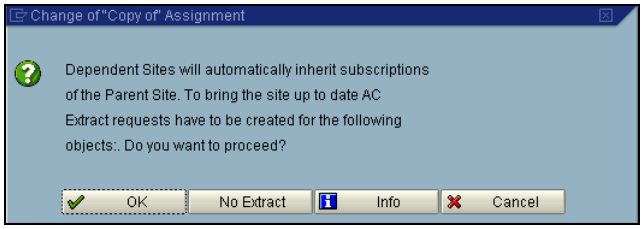
1. Call transaction SMOEAC (Administration Console). Choose ***Object Type*** *Site* and enter the ***Object Name*** *OLTP*. Then push the ***Create*** button.



2. Enter the ***Site Name*** *OLTP*, in the field ***Description*** a descriptive text, for example, *OLTP IS-U/CCS System*, and choose the ***Site Type*** *R/3*.

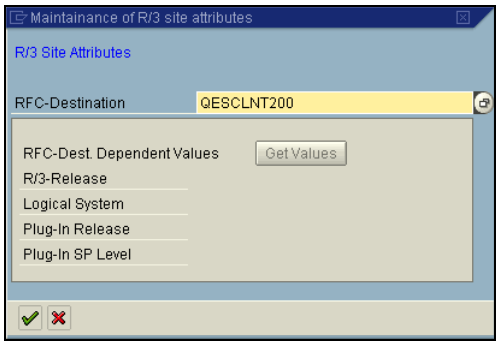


3. Push the ***Site Attributes*** button.

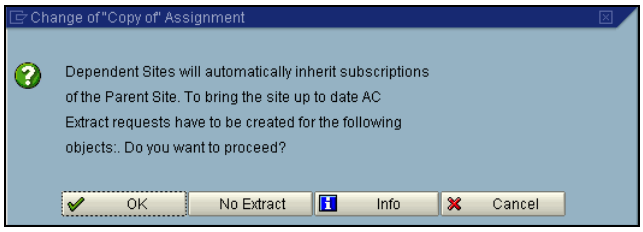


4. Push the ***No Extract*** button.

5. Enter in the field RFC-Destination the name of the RFC destination you created to communicate with the ERP system, for example QESCLNT200.



6. Push the ***Enter*** button and then the ***Save*** button.



7. Push the ***No Extract*** button.

8. The created site is displayed.

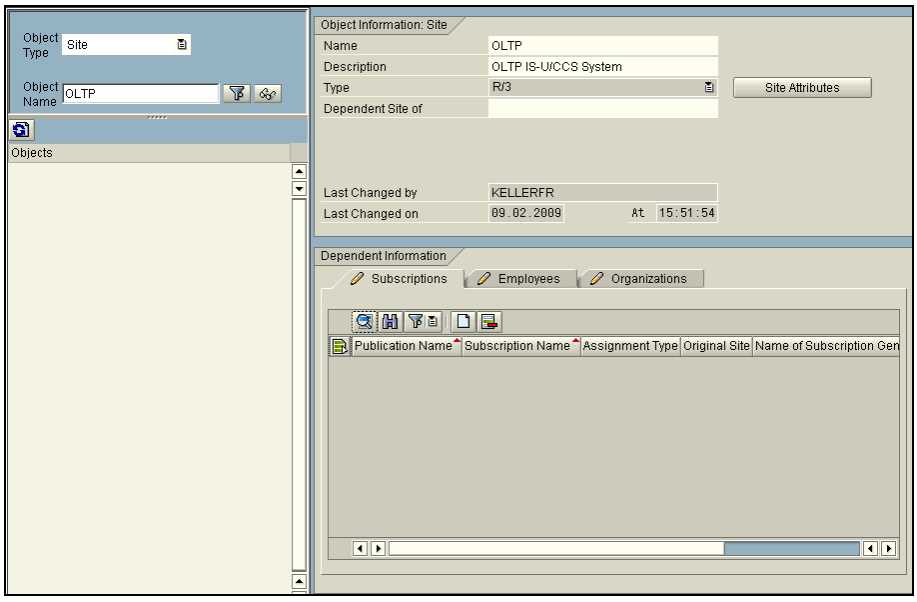
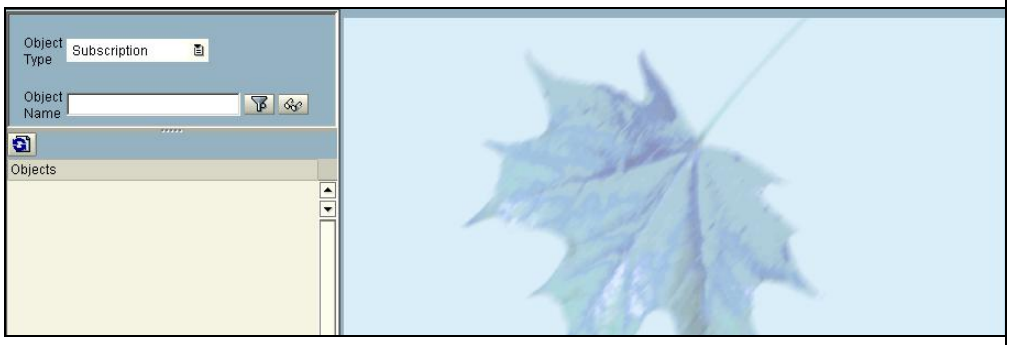


Figure 4-1: Definition of the OLTP site in the CRM system

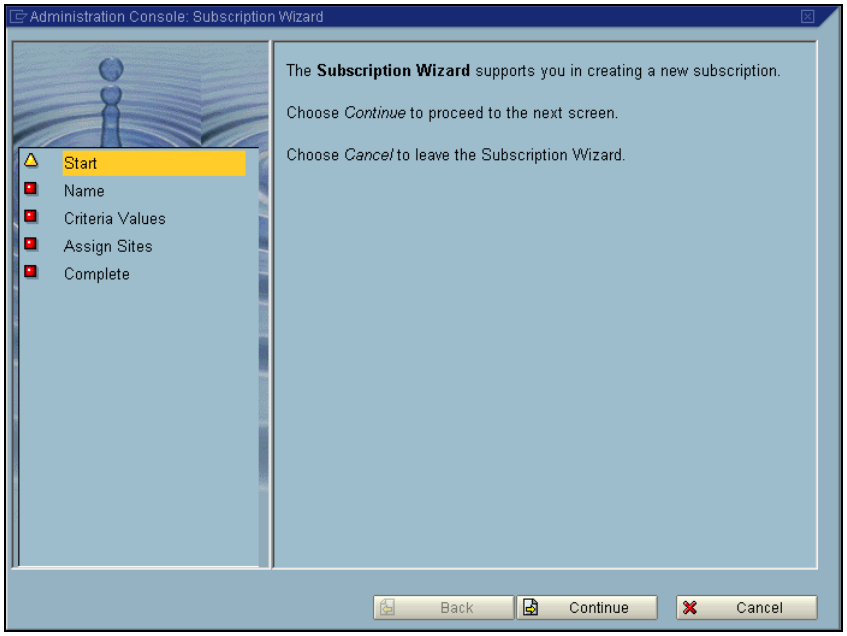
4.1.2 Check/Create Subscriptions

By creating subscriptions, you can determine the distribution of data to sites by assigning these sites to previously defined publications. Only when subscriptions have been made for a site, data is made available to this site. By choosing the corresponding function, you can create additional subscriptions, display, change or delete subscriptions that already exist. A Subscription Wizard is available to help you to create new subscriptions.

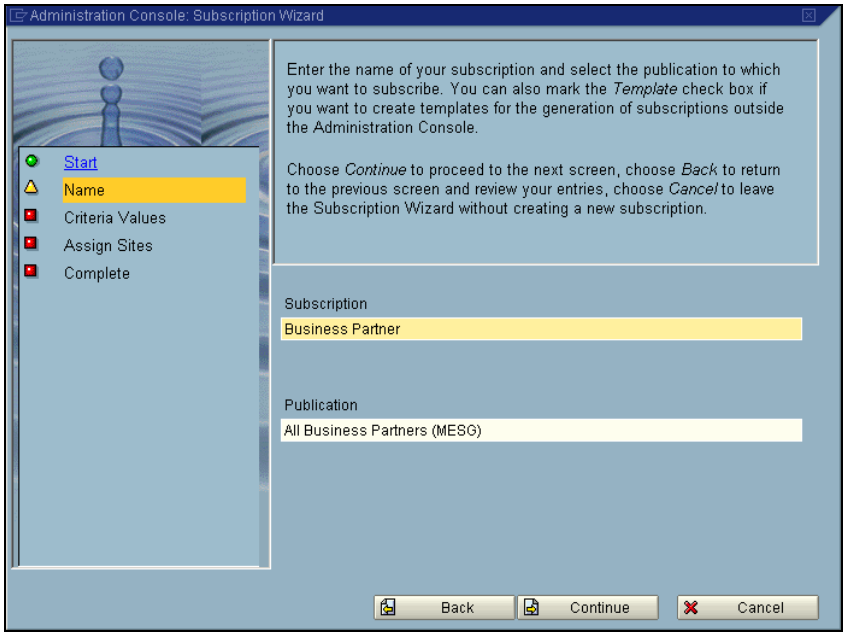
1. Call transaction SMOEAC (Administration Console). Choose ***Object Type*** *Subscription*.



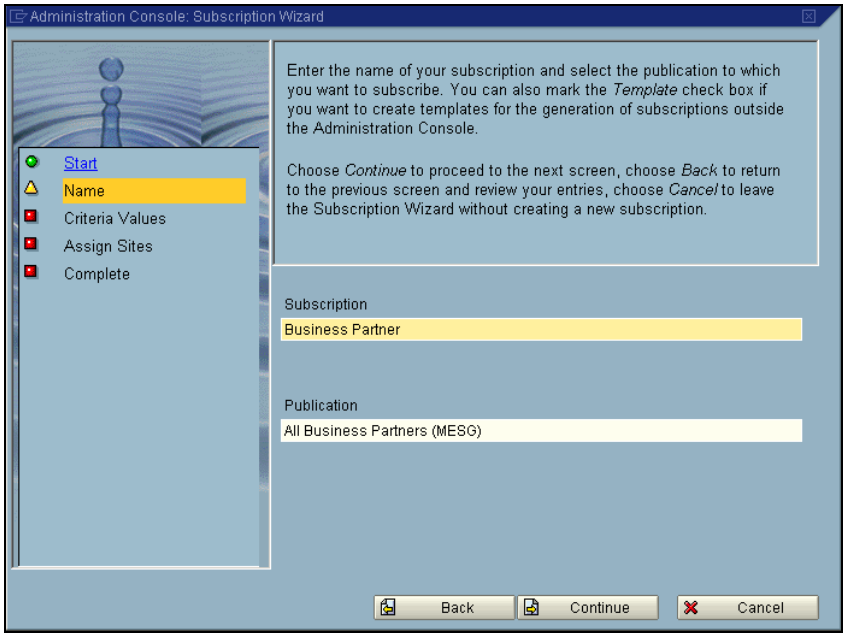
2. Push the ***Create*** button in order to start the Subscription Wizard



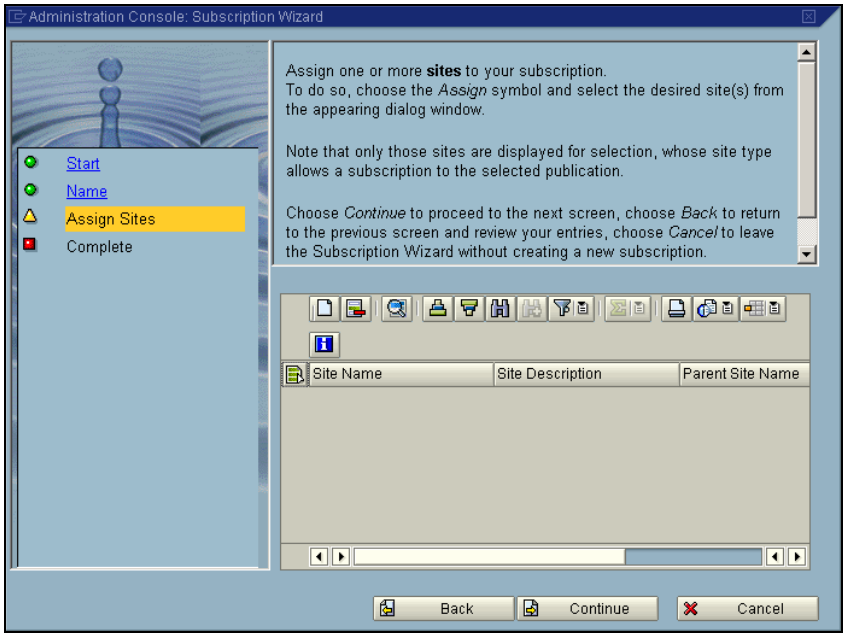
3. Push the ***Continue*** button



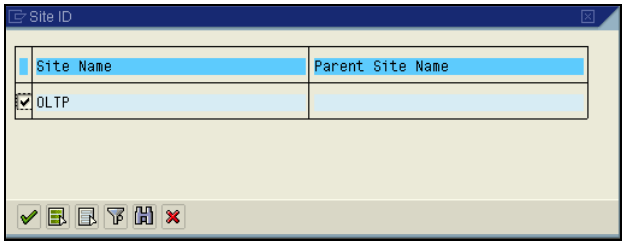
4. Enter in the name of your subscription in the ***Subscription*** field, for example *Business Partner* and choose *All Business Partners (MESG)* as the ***publication***.



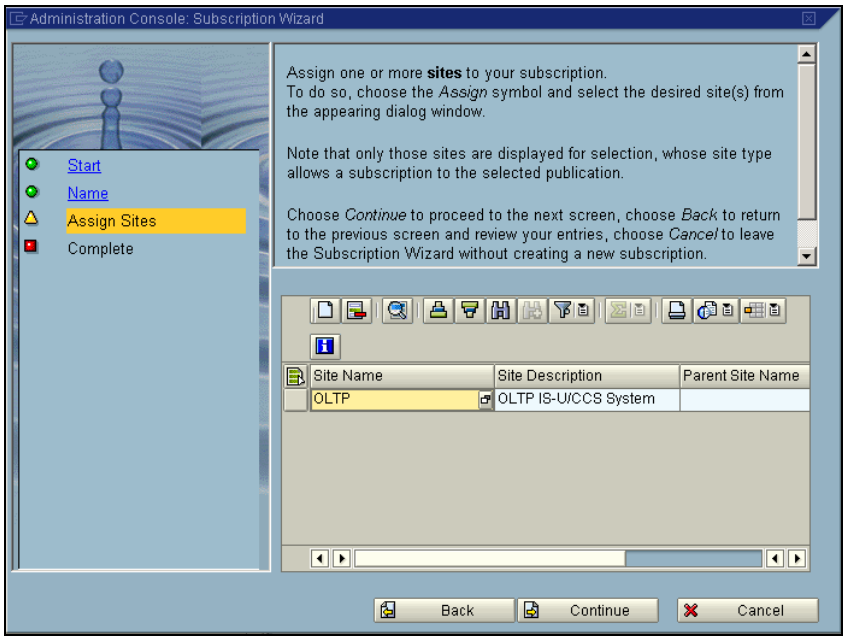
5. Push the ***Continue*** button.



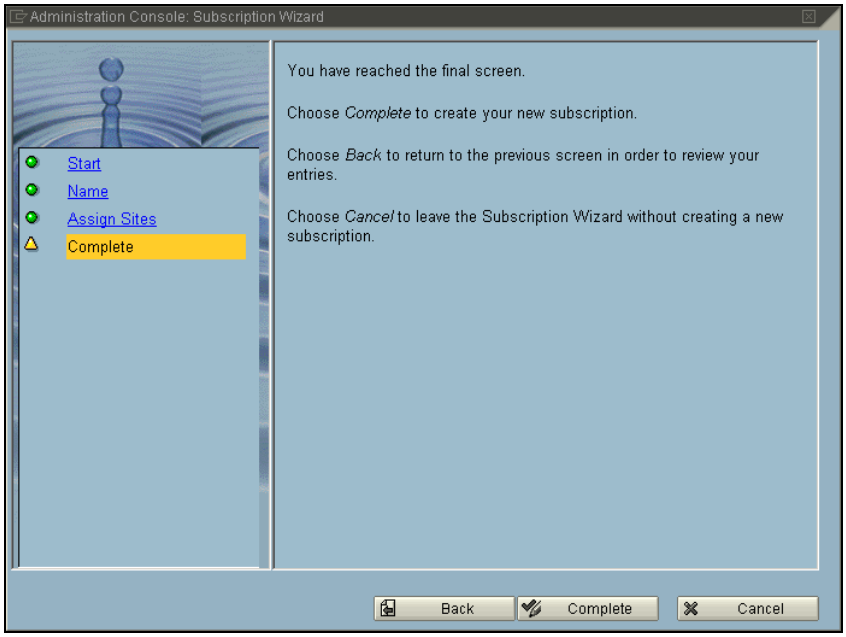
6. Push the ***Create*** button.



7. Select the OLTP site and push the ***Enter*** button. The OLTP site has been assigned to your subscription.



8. Push the ***Continue*** button.



9. Push the ***Complete*** button.

Figure 4-2: Definition of business partner subscription

Repeat the steps above to create the following subscriptions in the same way:

Subscription	Publication
Business Agreement	Business Agreements (MESG)
IS-U Connection Object	IS-U Connection Objects (MESG)
IS-U Point of Delivery	IS-U Points of Delivery (MESG)
Business Transactions	All Business Transactions (MESG)

Figure 4-3 Subscriptions for SAP for Utilities

Figure 4-4 shows the created OLTP site with all subscriptions for SAP for Utilities.

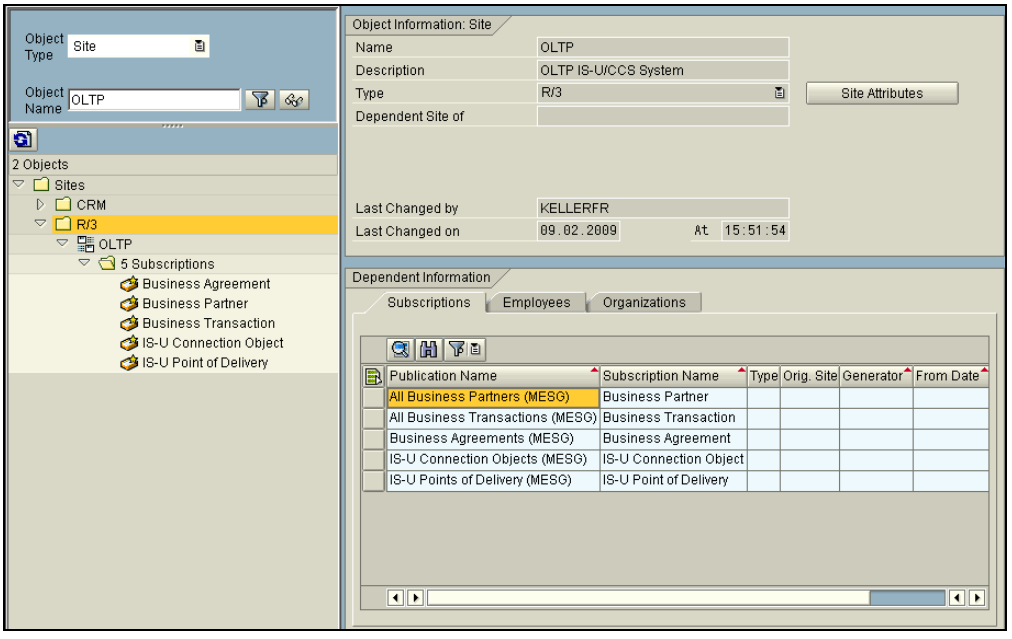
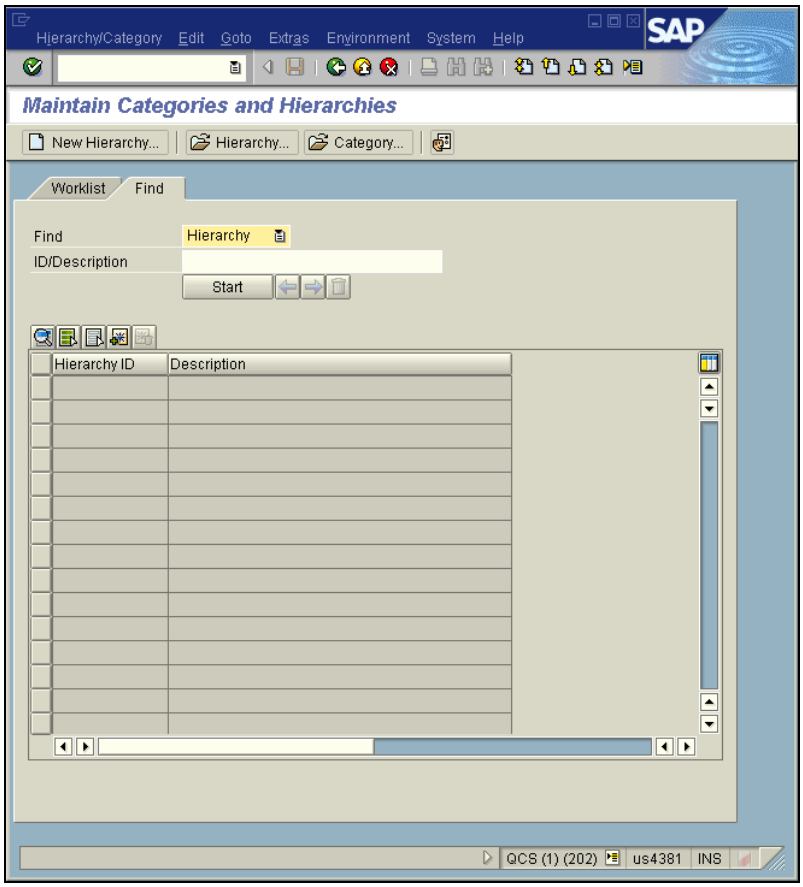


Figure 4-4 Created Site with all necessary Subscriptions

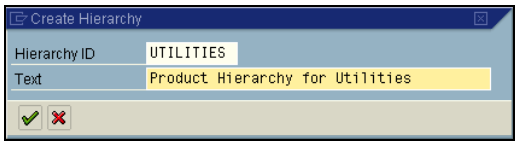
4.1.3 Check/Create Product Hierarchy

In a basic Customizing a basic hierarchy must exists with a main category and two subcategories for connection object and point of delivery.

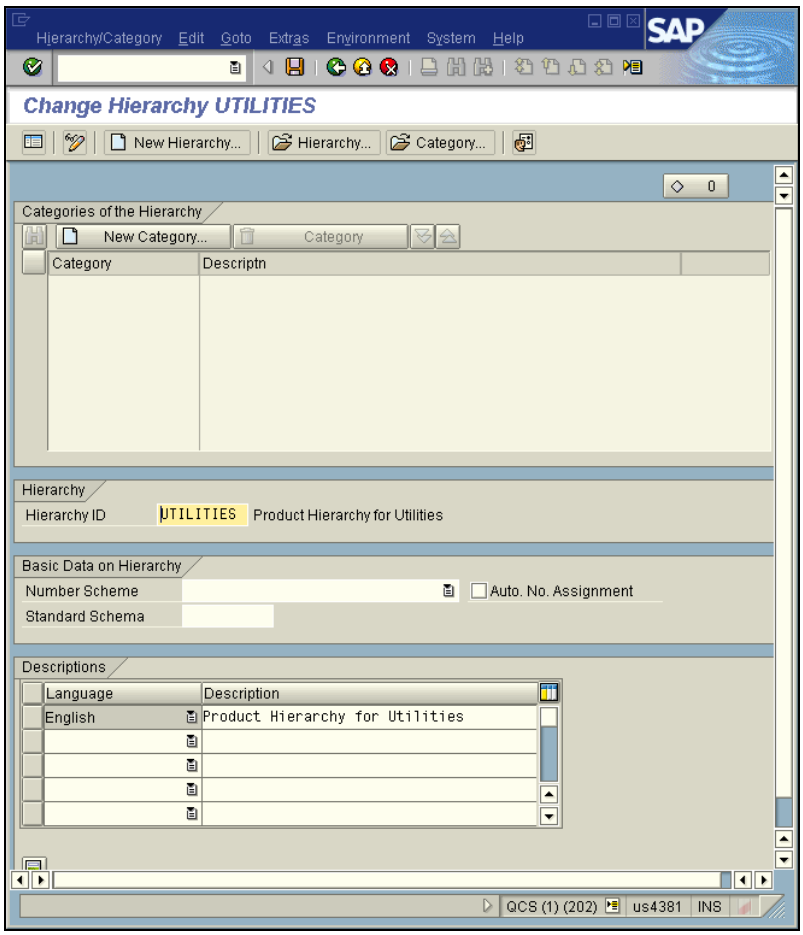
1. Call transaction `COMM_HIERARCHY` (Maintain Categories and Hierarchies)



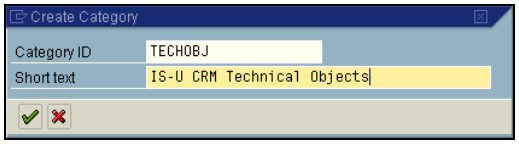
2. Push the ***New Hierarchy*** button. Enter in the ***Hierarchy ID*** field *UTILITIES* and in the ***Text*** field a descriptive text of the hierarchy.



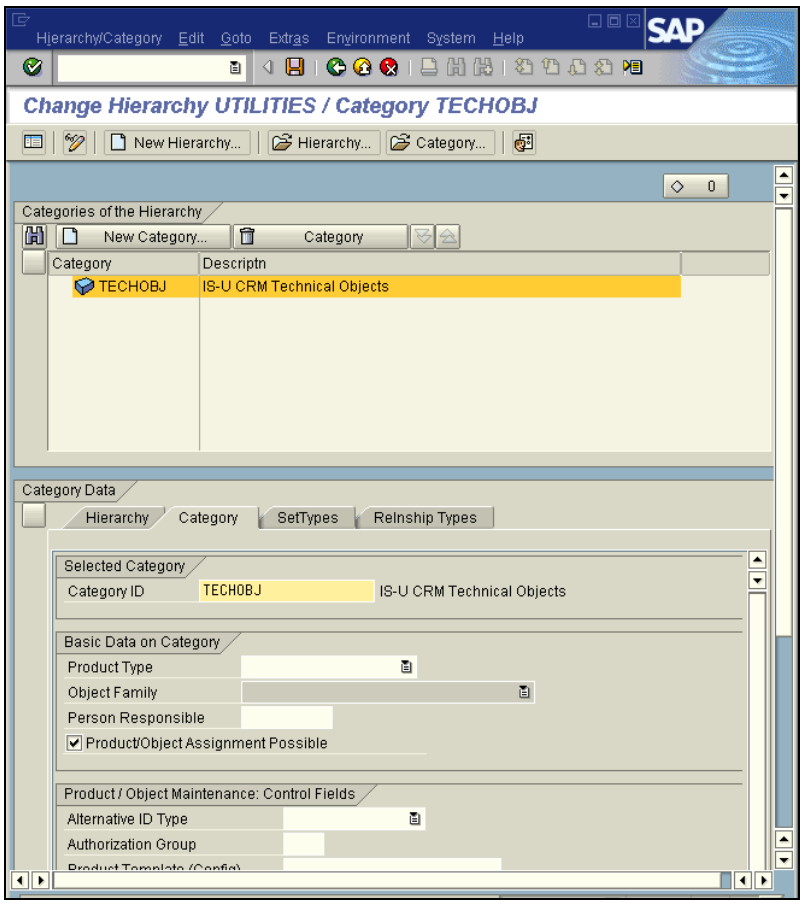
3. Push the ***Continue*** button.



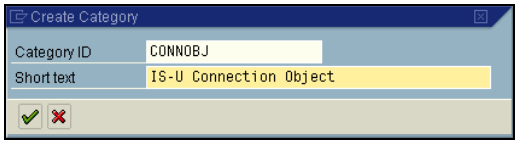
4. Push the ***New Category*** button. Enter in the ***Category ID*** field *TECHOBJ* and in the ***Short Text*** field a descriptive text of the category.



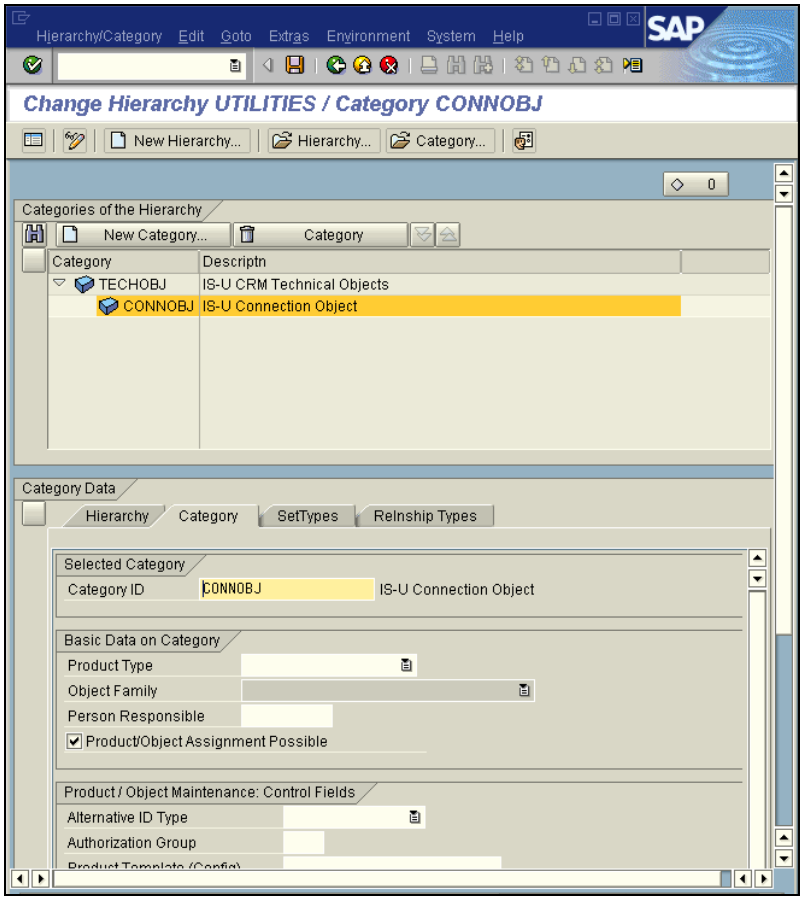
5. Push the ***Continue*** button.



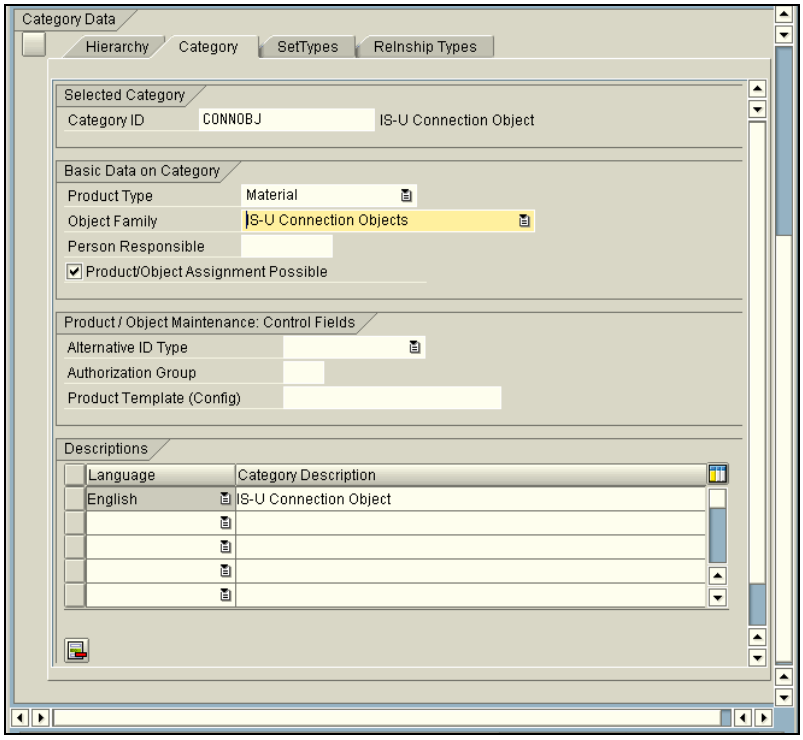
6. Select the defined *TECHOBJ* category and push the ***New Category*** button. Enter in the ***Category ID*** field *CONNOBJ* and in the ***Short Text*** field a descriptive text of the category.



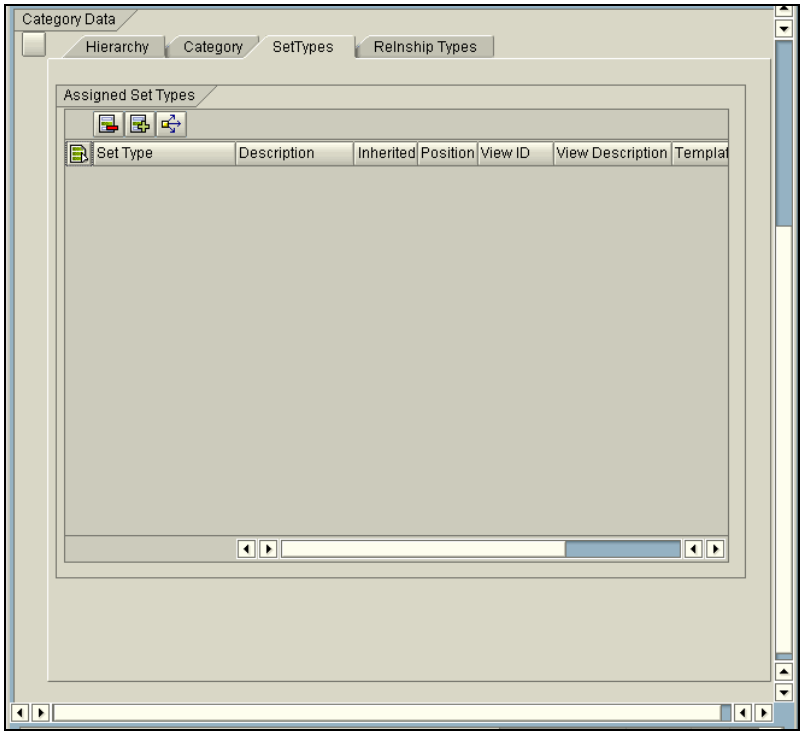
7. Push the ***Continue*** button.



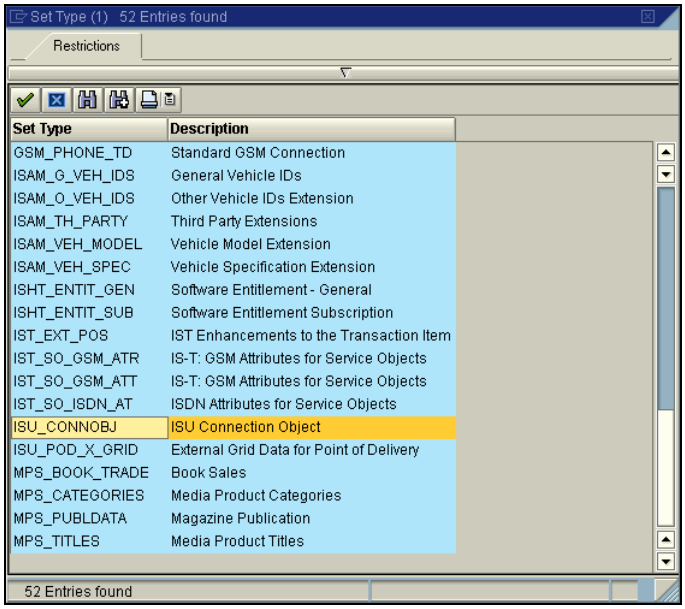
8. On the ***Category*** sub screen choose *Material* as the ***Product Type*** and *IS-U Connection Objects* as ***Object Family***.



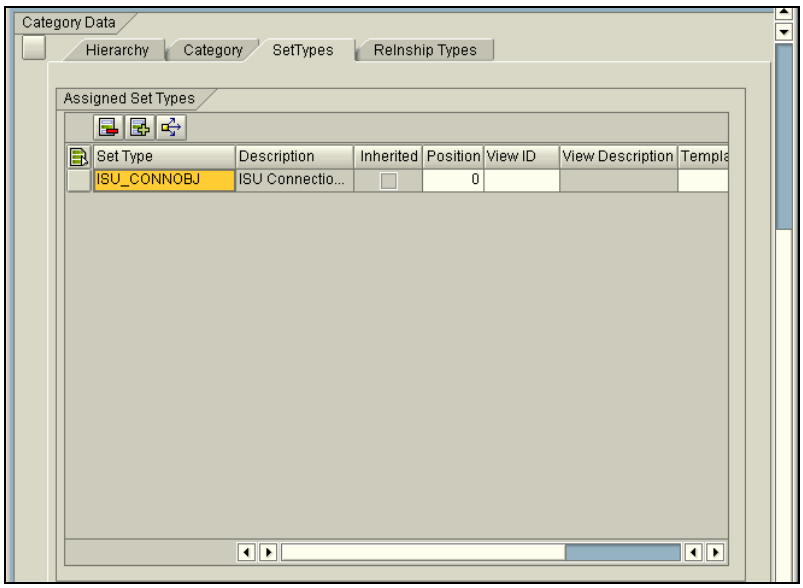
9. Choose the ***SetTypes*** sub screen.



10. Push the ***Add Set Types*** button.



11. Choose *ISU_CONNOBJ*. Then push the ***Return*** key. The category ISU_CONNOB has been added to the hierarchy.



12. Push the **Save** button to save the hierarchy

Figure 4-5: Definition of the Product Hierarchy in the CRM system

Repeat the steps 6 – 12 above to create the following categories in the same way:

Category ID	Product Type	Object Family	Set Type
POD	Material	IS-U Points of Delivery	ISU_POD
SAP_ENERGY	Service		INDUSTRYSECTOR

Figure 4-6 Further Categories for the Hierarchy TECHOBJ

Figure 4-7 shows the created hierarchy:

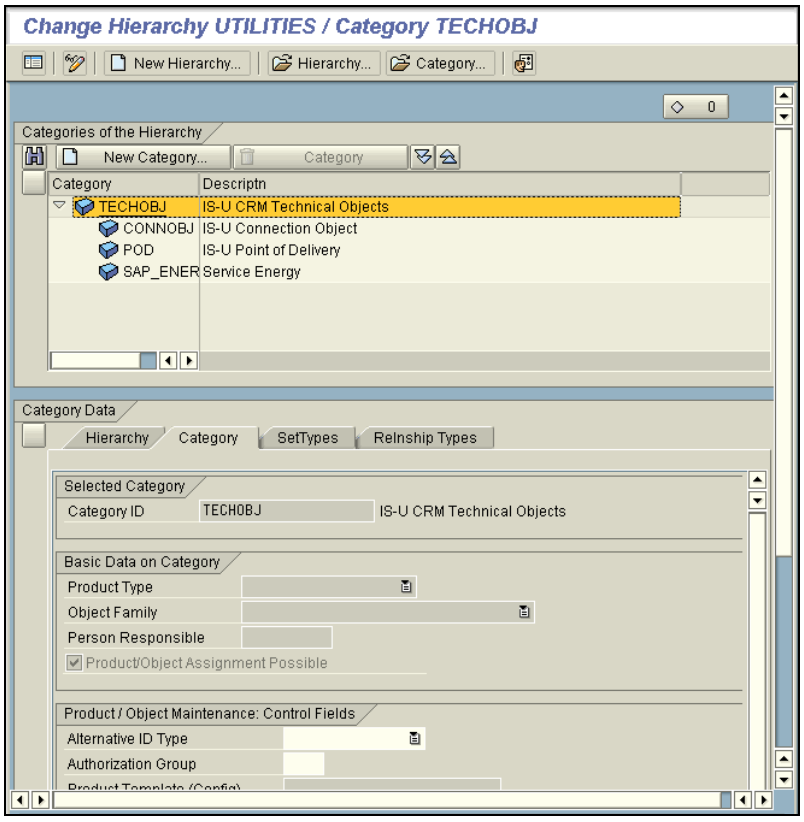
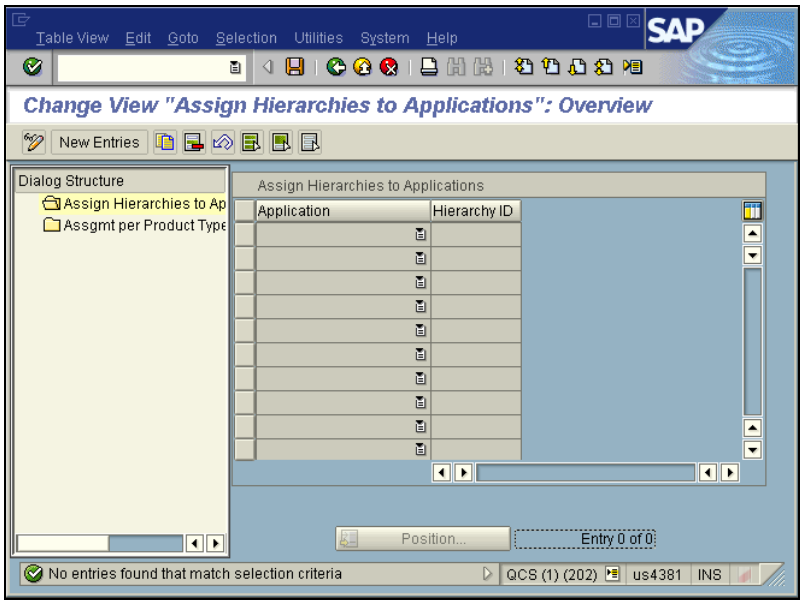


Figure 4-7 Newly Created Hierarchy TECHOBJ

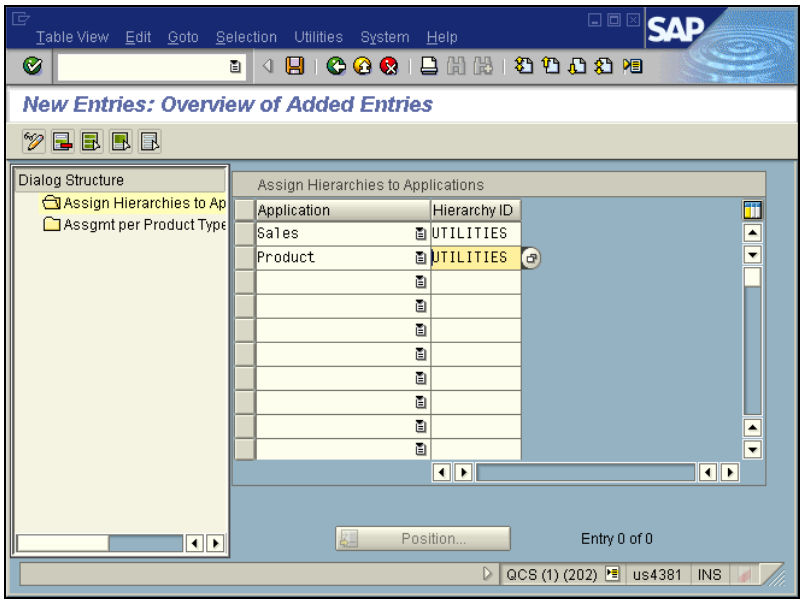
4.1.4 Check/Assign Hierarchies to Applications

You must assign the defined hierarchy at least to the *Product* application. Figure 4-8 shows how to assign the hierarchy.

1. Call transaction `COMM_PRAPPLCAT` (Assign Hierarchies to Applications).



2. Push the ***New Entries*** button. Then choose the ***applications*** *Sales* and *Product* and assign it to the ***hierarchy ID*** *UTILITIES*.



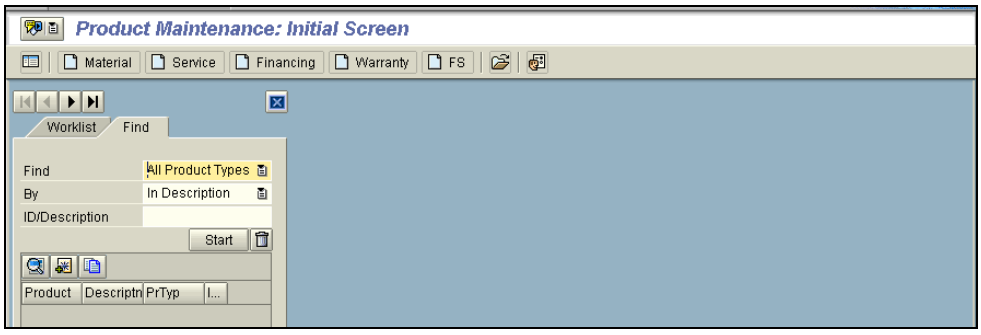
3. Push the ***Save*** button

Figure 4-8: Assign hierarchies to applications

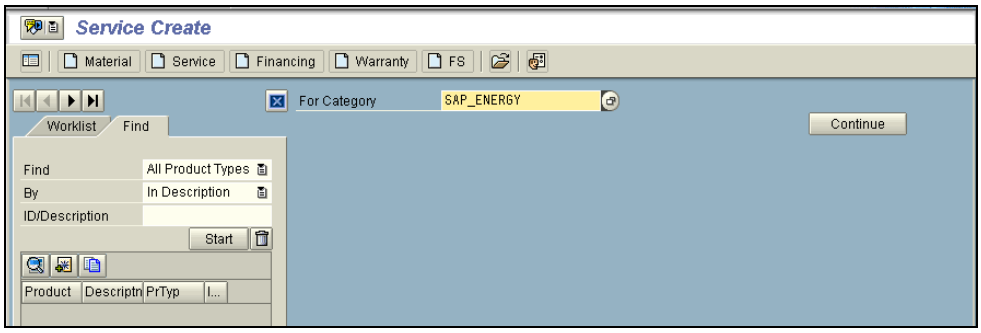
4.1.5 Check/Assign Definition of Product

You must define products. A Product corresponds to rates in SAP for Utilities. Figure 4-9 shows how to a basic product.

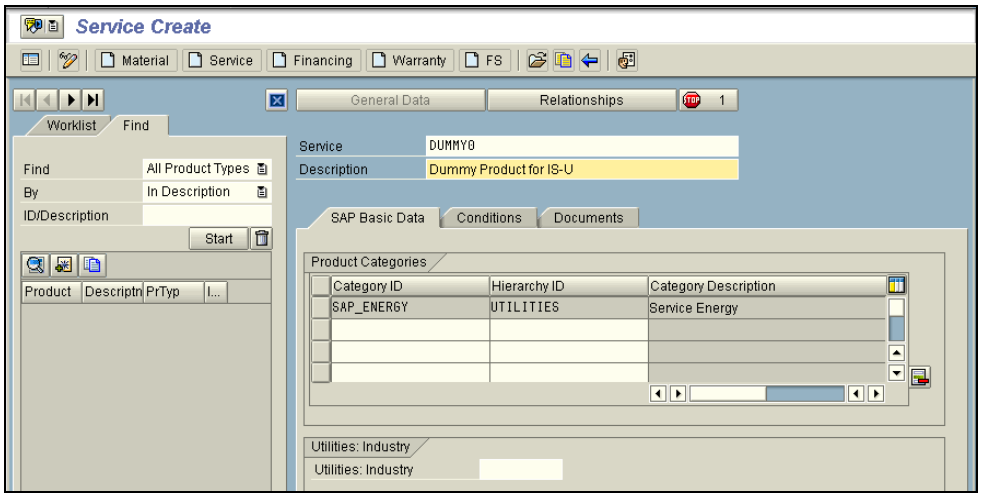
1. Call transaction `COMMPR01` (Product Workbench).



2. Push the ***Create Service*** button. Then enter the name of the Category you defined in chapter 4.1.3 *Check/Create Product Hierarchy*.



3. Push the ***Continue*** button. Then enter the name of ***Service*** name that is the product name and a ***Description*** of the new product.

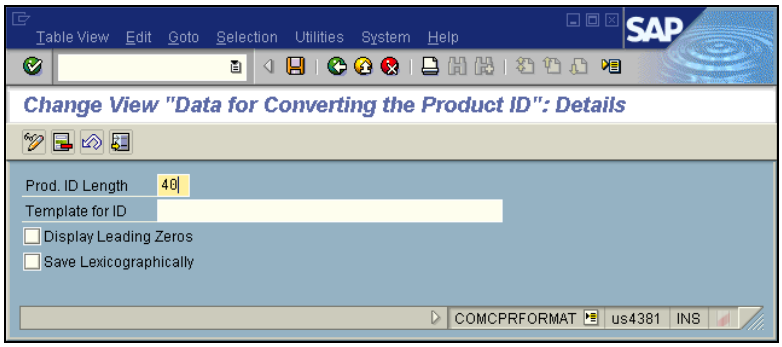


4. Push the ***Save*** button.

Figure 4-9: Definition a Product

In addition, you must define the maximum input/output length of a product ID. Figure 4-10 shows how change the length of the product ID.

1. Call transaction COMCPRFORMAT (Data for Converting the Product ID).

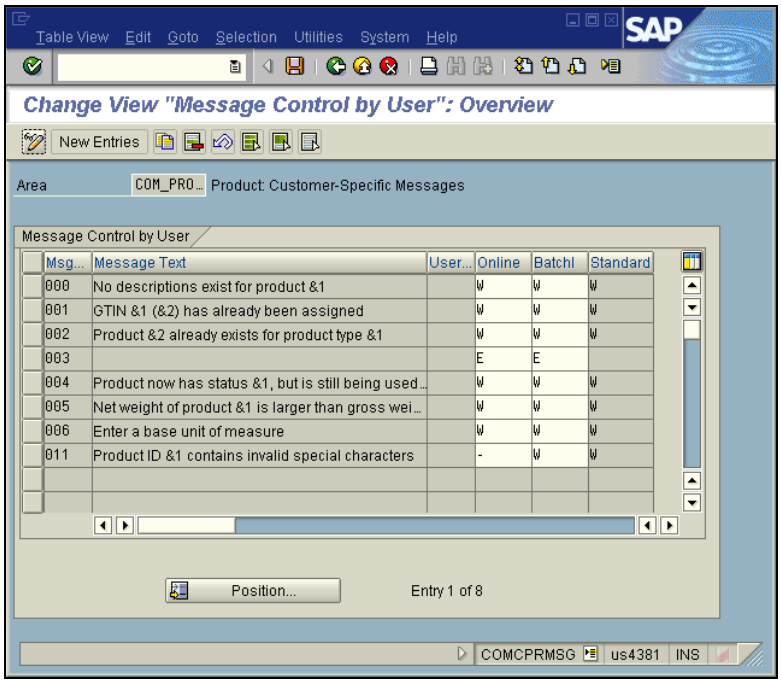


2. Change the ***Product ID Length*** to 40 and push the ***Save*** button.

Figure 4-10: Definition the correct Length of a Product ID

Switch off certain message processing during the request download. Figure 4-10 shows how change the length of the product ID.

1. Call transaction COMCPRMSG (Message Control by User).



2. Specify that the system does not issue a message 009 and 011 in the online mode and push the ***Save*** button.

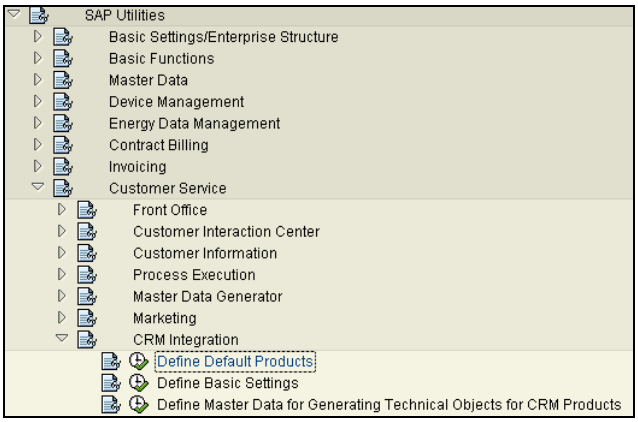
Figure 4-11: Deactivation of Message during Request Download

4.2 ERP System

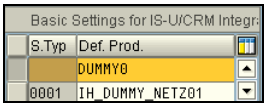
4.2.1 Check/Define the Default Product

You must define the default product based on the service type. The CRM system must contain a matching default product. Figure 4-12 shows how to define the default product.

1. Start activity in IMG ***Define Default Product***.



2. Enter the name of the default product.



3. Push the ***Save*** button.

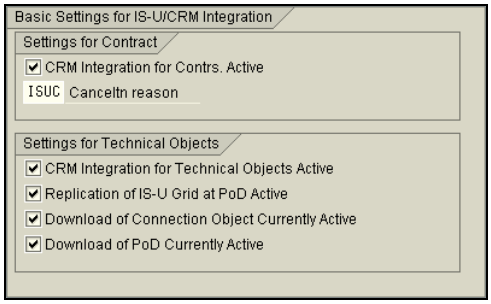
Figure 4-12: Definition a Default Product

4.2.2 Check/Create Basic Settings for IS-U/CRM Integration

You must define the basic settings to enable the replication between the SAP for Utilities and CRM systems. Figure 4-13 shows how to define the basic settings.

1. Start activity in IMG ***Define Basic Settings***.

2. Define the necessary Customizing.



3. Push the ***Save*** button.

Figure 4-13: Configuration of the Basic Settings

5 Execution of the Initial Download

5.1 Dependencies between business objects

When you map out a strategy for the initial download, you must take the implemented data model in the CRM system into account as well as the data migration strategy for a data load into the ERP system. Figure 5-1 shows the data model in SAP CRM and the corresponding business objects in SAP for Utilities.

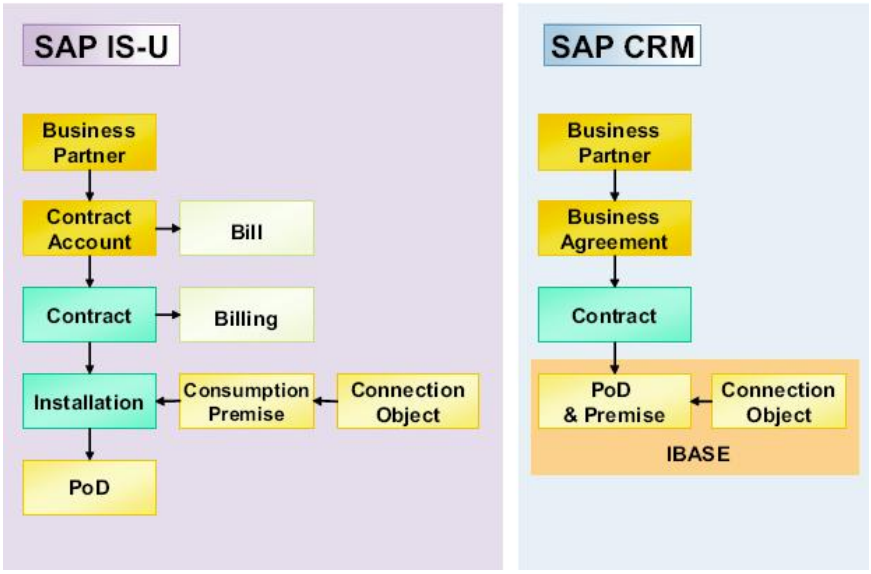


Figure 5-1 Data Model in an IS-U/CRM Integration Scenario

An elementary principle of the SAP for Utility / CRM integration is that an Installed Base (IBase) component (the point of delivery) in the object list is generally allocated to a CRM contract item (that is a contract in SAP for Utilities). In this way, the link between business and technical objects is created in CRM. Technical objects in CRM are stored in the IBases (transactions `IB51-IB53`).

An IBase usually consists of a connection object, to which any number of points of delivery can be allocated. Each connection object has its own IBase. The top component is the connection object itself. The points of delivery are directly subordinated in a hierarchy. This means that the IBase can have a maximum of two hierarchy levels.

A utility contract corresponds to a service contract item in CRM. You can only allocate IBase components that correspond to a point of delivery to a contract item. For SAP for Utilities, the IBase maps the structure of the buildings and flats. These are independent of the actual owner or user nor a contract must exist. This means that in the standard no business partners are defined at the connection object and the points of delivery. Only addresses are defined here. These addresses do not contain any personal data, such as a name. They only contain the address and further details, such as, apartment X or other data on the point of delivery item in the building.

The basic dependencies between business objects are:

- Before replicating any business object, you must replicate the relevant Customizing from the ERP into the CRM system.
- A business partner must exist in CRM (thus be replicated to CRM) before his contract accounts can be replicated.
- A connection object must exist in CRM before the PoDs (point of delivery) of the related installations can be replicated that are linked to the same connection object.

- The business partner and his contract account as well as the connection object and relevant PoDs must exist in CRM before a service contract can be replicated that is linked to the respective PoD.

As explained in chapter 1.1 *SAP CRM* Middleware, there are options available to conduct an initial download:

- Initial Download
- Request Download

Based on the experience in many projects the best practice for an initial replication of business objects from the SAP for Utilities ERP system into the CRM system is a combination of both: a replication of very few business object with an initial download followed by one or more request downloads. The initial download is required in order to update some system tables in both systems as well as to enable a delta replication after the initial download. Request downloads suits best to allow an initial download with the best flexibility and performance. In contrary, Customizing is always replicated by an initial download.

The following chapters show how to replicate some customizing objects followed by the replication of business objects from the ERP system into the CRM system.

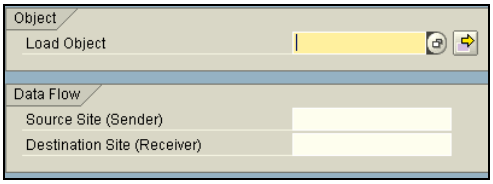
5.2 Initial Download of Customizing

Instead of re-defining the Customizing on the CRM system, you should replicate the Customizing data, such as, bank details, regional structure from the ERP system into the CRM system. The following chapters show how you perform an initial download of some of the necessary Customizing objects.

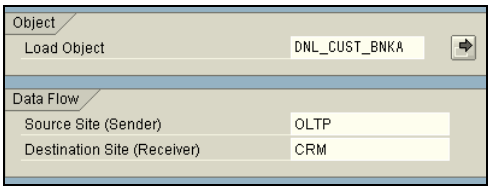
5.2.1 Initial Download of Bank Data

Figure 5-2 shows how to replicate bank details from the ERP system into the CRM system.

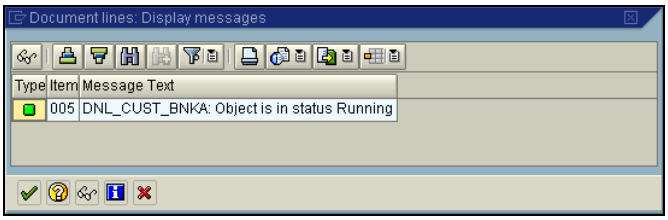
1. Call transaction R3AS (Start Initial Download)



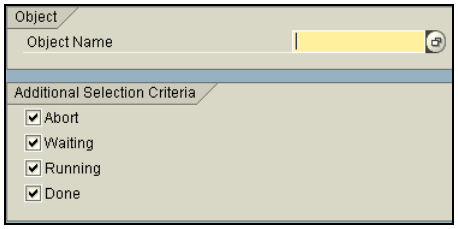
2. Enter *DNL_CUST_BNKA* as the ***load object***. The ***Source Site*** is *OLTP* and the ***Destination Site*** is *CRM*.



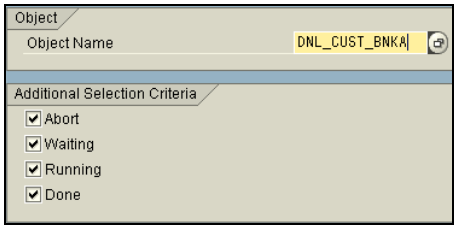
3. Push the ***Execute*** button to start the initial download.



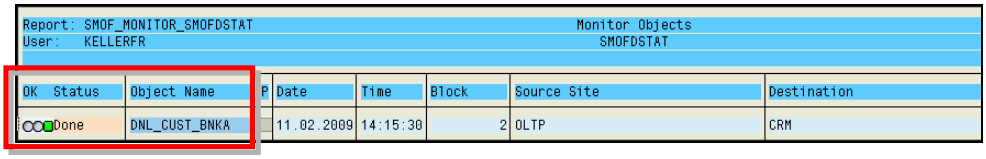
4. Push the ***Continue*** button to confirm the popup.
5. Call transaction R3AM1 (Monitor Initial Download)



6. Enter the ***Object Name*** *DNL_CUST_BNKA*.



7. Push the ***Execute*** button.



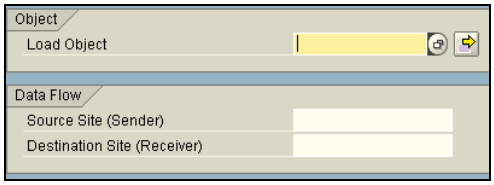
8. Check the status of the initial download.

Figure 5-2: Initial Download of Bank Data

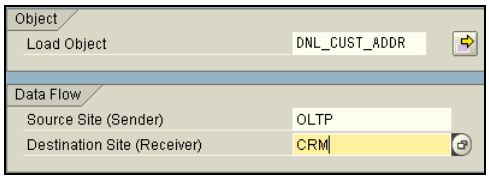
5.2.2 Initial Download of Regional Structure

Figure 5-3 shows how to replicate the regional structure from the ERP system into the CRM system.

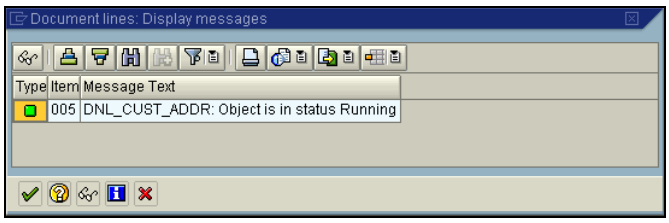
1. Call transaction R3AS (Start Initial Download)



2. Enter *DNL_CUST_ADDR* as the ***load object***. The ***Source Site*** is *OLTP* and the ***Destination Site*** is *CRM*.

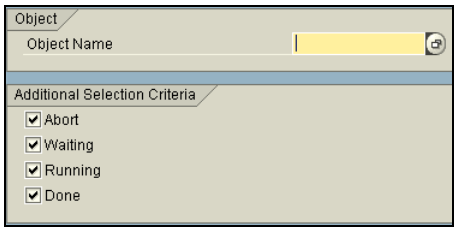


3. Push the ***Execute*** button to start the initial download.

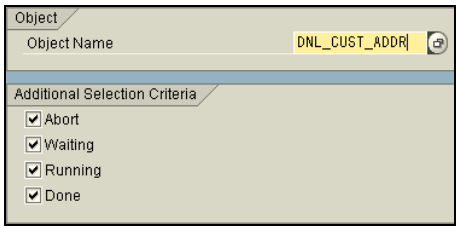


4. Push the ***Continue*** button to confirm the popup.

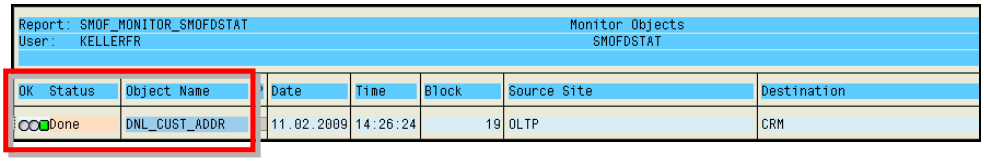
5. Call transaction R3AM1 (Monitor Initial Download)



6. Enter the ***Object Name*** *DNL_CUST_ADDR*.



7. Push the ***Execute*** button.



8. Check the status of the initial download.

Figure 5-3: Initial Download of the Regional Structure

5.3 Initial Download of Business Objects

The following chapters show how to perform the initial download as request downloads of the data environment as shown in Figure 5-4.

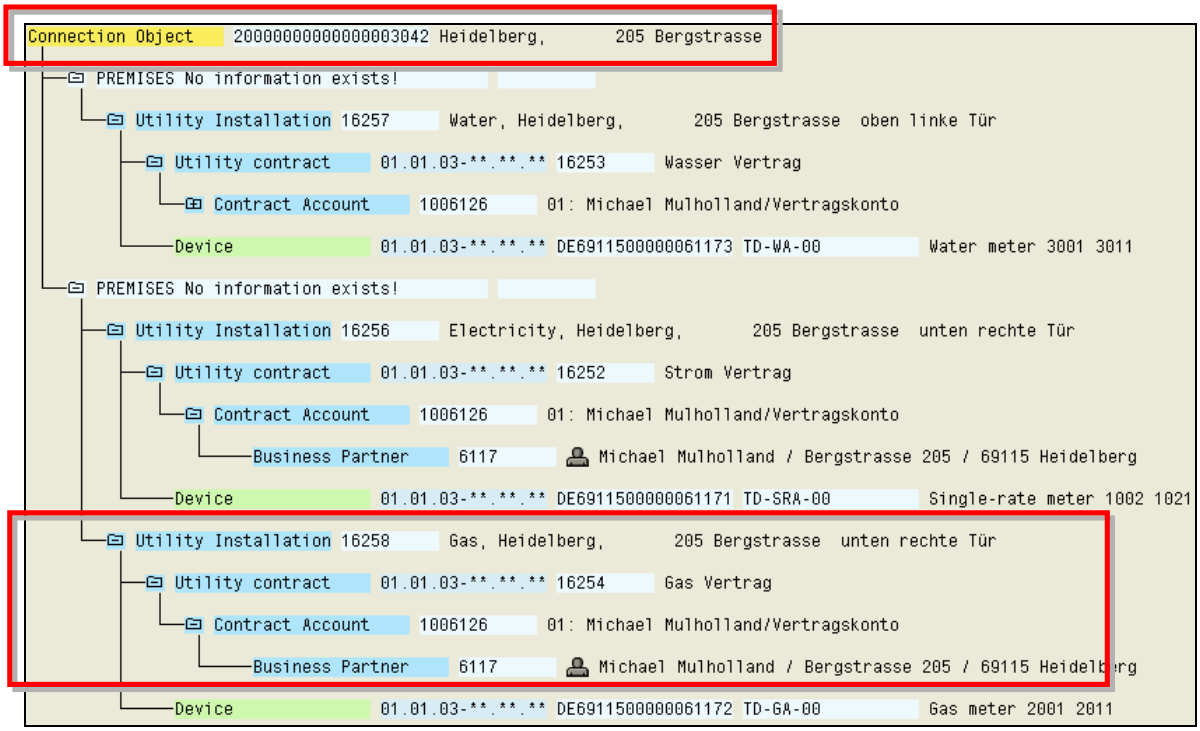


Figure 5-4 Data Environment for Business Partner 000006117 in SAP for Utilities

You must assign names to your requests that are unique in the first eight characters in the name. Naming in this way facilitates the assignment of an inbound queue for the corresponding request. This is particularly important, once you start more than one request concurrently. The CRM Middleware builds the name of the inbound queue of a related request as follows:

R3AR_	*R* for Request
<12345678>	Request name as defined in transaction R3AR2 (Define Request)
123456	PID (process identifier) of the used dialog work process
_xx	Inbound queue number for request (see chapter 7.1.1 *Important Parameters in CRMPAROLTP Table* for details of the relevant CRM_MAX_NO_QUEUES_PER_REQUEST parameter).

Figure 5-5 Components of the Name of an Inbound Queue

5.3.1 Preparation

If you want to use the request download to replicate high volumes business objects, you must execute a successful initial download for each adapter object that includes at least one business object. This is required to ensure that the CRM Middleware is functioning but also to enable the automatic delta replication on completion of the replication of all business objects. To achieve this you must execute an initial download for the related adapter object as follows:

- Change the global filter of the adapter object to select at least one business object. You can change the filter with the transaction R3AC1 (Adapter Object Overview). For more

information how to change the filter settings for an adapter object, see Figure 6-1 in chapter 6.3 *Reducing Data Volume during the Initial Download*.

- o Start an initial download with the transaction R3AS (Execute Initial Download). For more information, how to execute an initial download, see chapter 5.2 *Initial Download of Customizing*.
- o Reset the filter settings to the desired range.

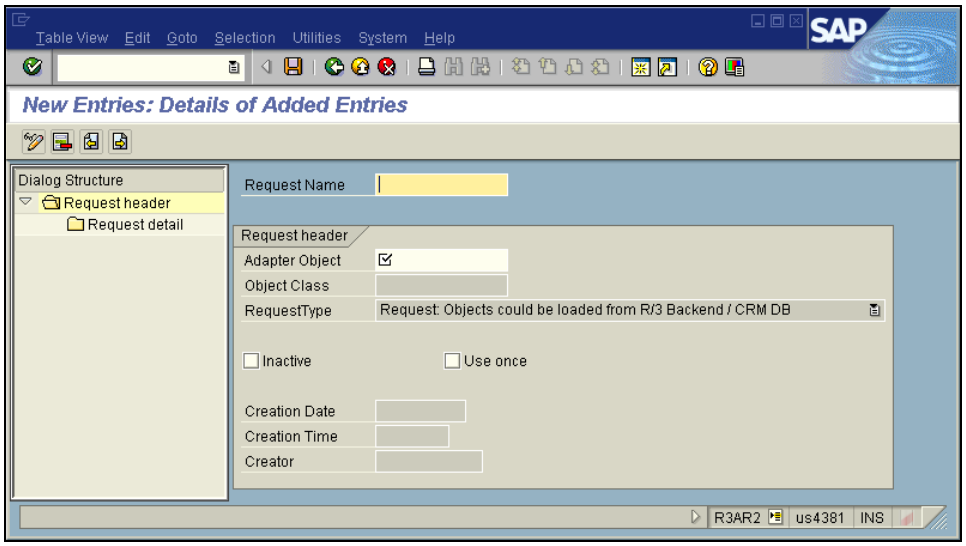
As a result, both the ERP and CRM system are ready for an automatic delta replication. The procedure has no affect on the execution of a request download.

5.3.2 Request Download of Business Partners

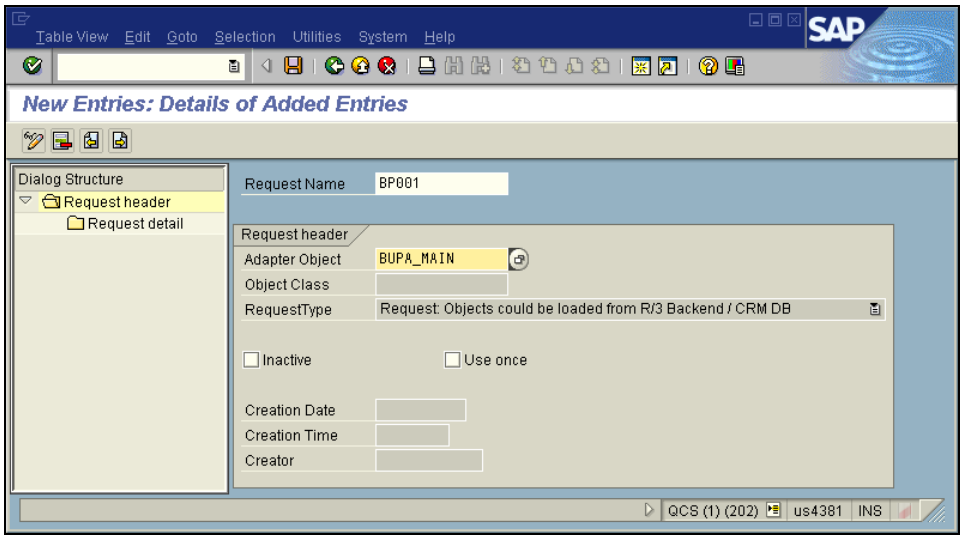
This chapter shows how to perform a request download of an existing business partner 0000006117 from the ERP system into the CRM system. You must define, start and monitor a request download into the CRM system. However, in order to monitor the data selection and retrieval in the ERP system, you must access the ERP system.

Definition of a request

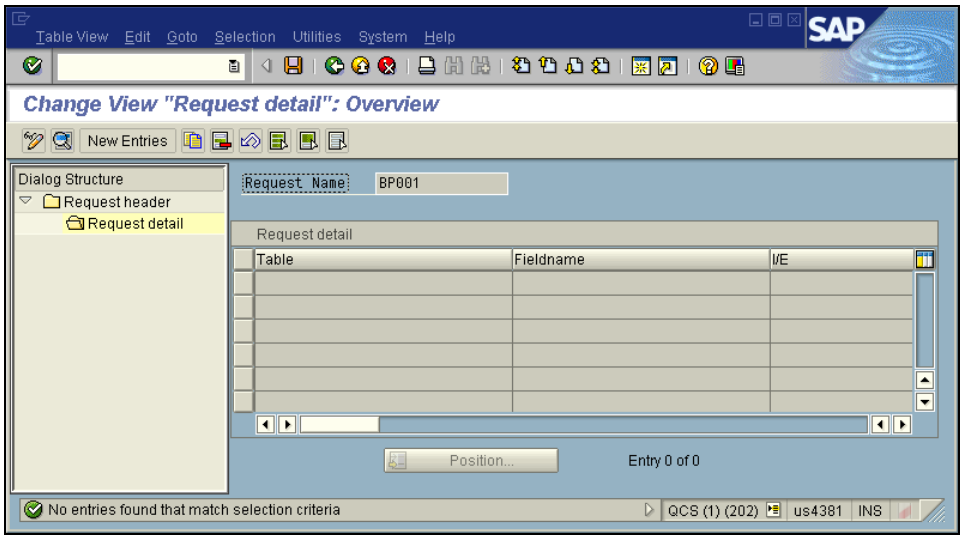
1. Call transaction R3AR2 (Define Request)



2. Push the ***Display-> Change*** button and then the ***New Entries*** button. Then, enter the request name, for example, *BP001*, and the name of the business partner ***adapter object*** *BUPA_MAIN*.

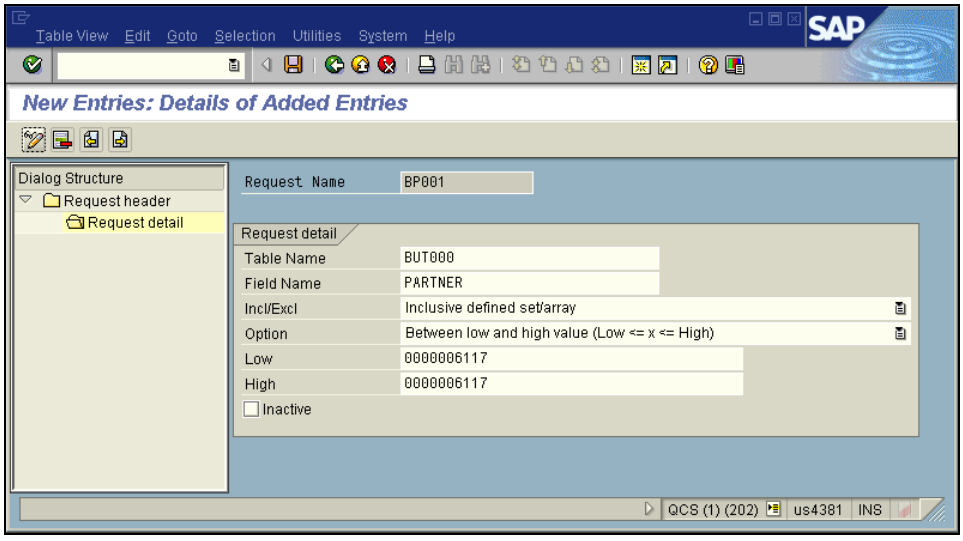


3. Change to the request details with a double-click on ***Request detail***.



4. Push the ***New Entries*** button. Then standard definition of the request details to replicate one single business partner is as below. The business partner number must be entered with leading zeros.

Field Name	**Entries**
Table Name	BUT000
Field Name	PARTNER
Incl/Excl	Inclusive defined set/array
Option	Between low and high value (Low <= x <= High)
Low	0000006117
High	0000006117

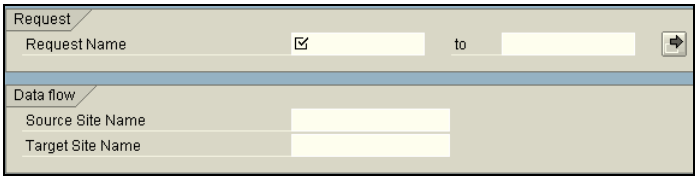


5. Push the ***Save*** button. Then push the ***Back*** button.

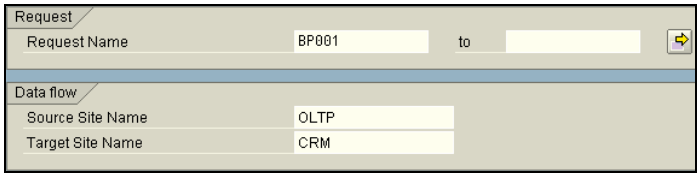
Figure 5-6: Definition of a Request

Start of a the defined request

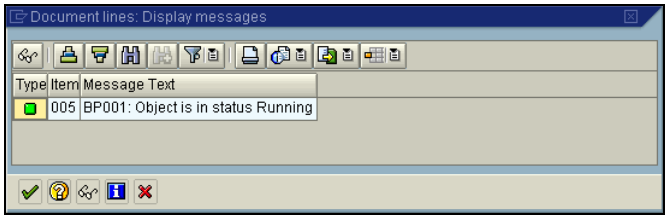
1. Call transaction R3AR4 (Start Request)



2. Enter the request name, for example, *BP001*, and the name of the ***Source Site Name*** *OLTP* and ***Target Site Name*** *CRM*.



3. Push the ***Execute*** button.

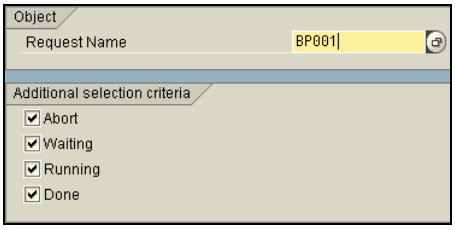


4. Push the ***Continue*** button.

Figure 5-7: Start of a Request

Monitor the started request

1. Call transaction R3AR3 (Monitor Request) and enter the ***request name***., for example, *BP001*.



2. Push the ***Execute*** button.

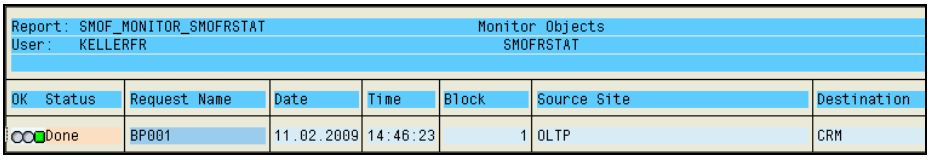


Figure 5-8: Monitor a Request

5.3.3 Request Download of further Business Objects

Repeat the steps above to replicate the dependant business objects of business partner 0000006117 in the same way. For each object ID, you need enter all leading zeros.

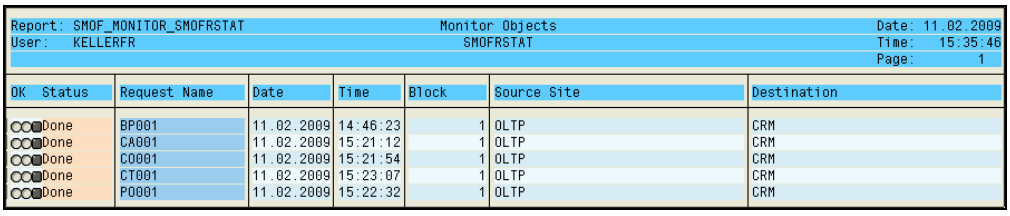
Request	Adapter Object	Table Name	Field Name	ID
CA001	BUAG_MAIN	FKKVKP	VKONT	000001006126
CO001	SI_CONNOBJ	EHAUISU	HAUS	20000000000000003402
PO001	SI_POD	EHAUISU	HAUS	20000000000000003402
CT001	SI_CONTRACT	EVERH	VERTRAG	0000016252

Figure 5-9 Filter Settings for required requests

Before you start the initial download of migrated contracts with the SI_CONTRACT adapter object, you must allocate a product to each contract. SAP delivers the ECRM_GENERATE_EVERH report to determine and allocate a suitable product to a utility contract and creates an entry in the EVERH table. For more information, see chapter 6.21 *Preparation of Contract Replication (SI_CONTRACT)*.

5.3.4 Display Replication Results

Call transaction R3AR3 (Monitor Request) to display the request status. Figure 5-10 shows that all request have been processed successfully.

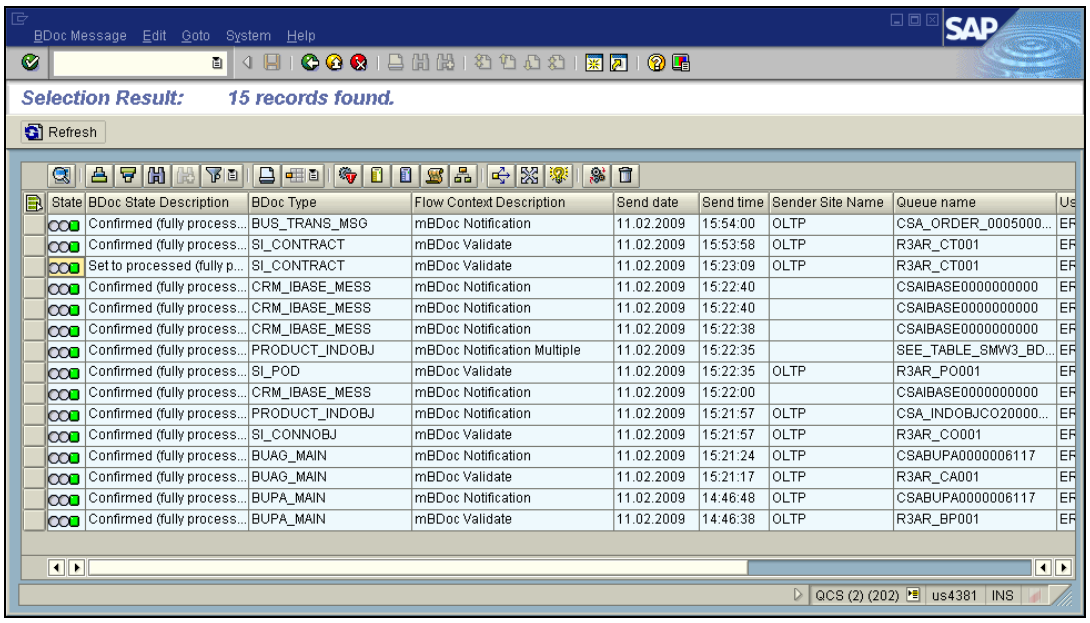


Report: SMOF_MONITOR_SMOFRSTAT — Monitor Objects — Date: 11.02.2009
User: KELLERFR — SMOFRSTAT — Time: 15:35:46
Page: 1

OK	Status	Request Name	Date	Time	Block	Source Site	Destination
	Done	BP001	11.02.2009	14:46:23	1	OLTP	CRM
	Done	CA001	11.02.2009	15:21:12	1	OLTP	CRM
	Done	CO001	11.02.2009	15:21:54	1	OLTP	CRM
	Done	CT001	11.02.2009	15:23:07	1	OLTP	CRM
	Done	PO001	11.02.2009	15:22:32	1	OLTP	CRM

Figure 5-10 Screen to display the Status of the Request

Call transaction SMW01 (Monitor BDocs). Figure 5-11 shows all BDocs that have been processed during the replication of the business objects together with the related requests.



BDoc Message Edit Goto System Help

Selection Result: *15 records found.*

Refresh

State	BDoc State Description	BDoc Type	Flow Context Description	Send date	Send time	Sender Site Name	Queue name	Us
	Confirmed (fully process...	BUS_TRANS_MSG	mBDoc Notification	11.02.2009	15:54:00	OLTP	CSA_ORDER_0005000...	ER
	Confirmed (fully process...	SI_CONTRACT	mBDoc Validate	11.02.2009	15:53:58	OLTP	R3AR_CT001	ER
	Set to processed (fully p...	SI_CONTRACT	mBDoc Validate	11.02.2009	15:23:09	OLTP	R3AR_CT001	ER
	Confirmed (fully process...	CRM_IBASE_MESS	mBDoc Notification	11.02.2009	15:22:40		CSAIBASE0000000000	ER
	Confirmed (fully process...	CRM_IBASE_MESS	mBDoc Notification	11.02.2009	15:22:40		CSAIBASE0000000000	ER
	Confirmed (fully process...	CRM_IBASE_MESS	mBDoc Notification	11.02.2009	15:22:38		CSAIBASE0000000000	ER
	Confirmed (fully process...	PRODUCT_INDOBJ	mBDoc Notification Multiple	11.02.2009	15:22:35		SEE_TABLE_SMW3_BD...	ER
	Confirmed (fully process...	SI_POD	mBDoc Validate	11.02.2009	15:22:35	OLTP	R3AR_PO001	ER
	Confirmed (fully process...	CRM_IBASE_MESS	mBDoc Notification	11.02.2009	15:22:00		CSAIBASE0000000000	ER
	Confirmed (fully process...	PRODUCT_INDOBJ	mBDoc Notification	11.02.2009	15:21:57	OLTP	CSA_INDOBJCO20000...	ER
	Confirmed (fully process...	SI_CONNOBJ	mBDoc Validate	11.02.2009	15:21:57	OLTP	R3AR_CO001	ER
	Confirmed (fully process...	BUAG_MAIN	mBDoc Notification	11.02.2009	15:21:24	OLTP	CSABUPA0000006117	ER
	Confirmed (fully process...	BUAG_MAIN	mBDoc Validate	11.02.2009	15:21:17	OLTP	R3AR_CA001	ER
	Confirmed (fully process...	BUPA_MAIN	mBDoc Notification	11.02.2009	14:46:48	OLTP	CSABUPA0000006117	ER
	Confirmed (fully process...	BUPA_MAIN	mBDoc Validate	11.02.2009	14:46:38	OLTP	R3AR_BP001	ER

QCS (2) (202) us4381 INS

Figure 5-11 Screen to display the Status of processed BDocs

A request of the adapter object SI_CONTRACT always shows a success status, even an error occurred during the creation of the contract on the CRM system. You can analyze errors in the following ways:

- Transaction SLG1 (Analyze Application Log): Analyze the application log of the ***Object*** *CRM_DOCUMENT* (CRM One Order Document) and the ***Subobject*** *SINGLE*. You may enter the GUID of a specific contract in the ***External ID*** field. Figure 5-12 shows the selection screen to analyze the error logs of a contract replication on a specific day.

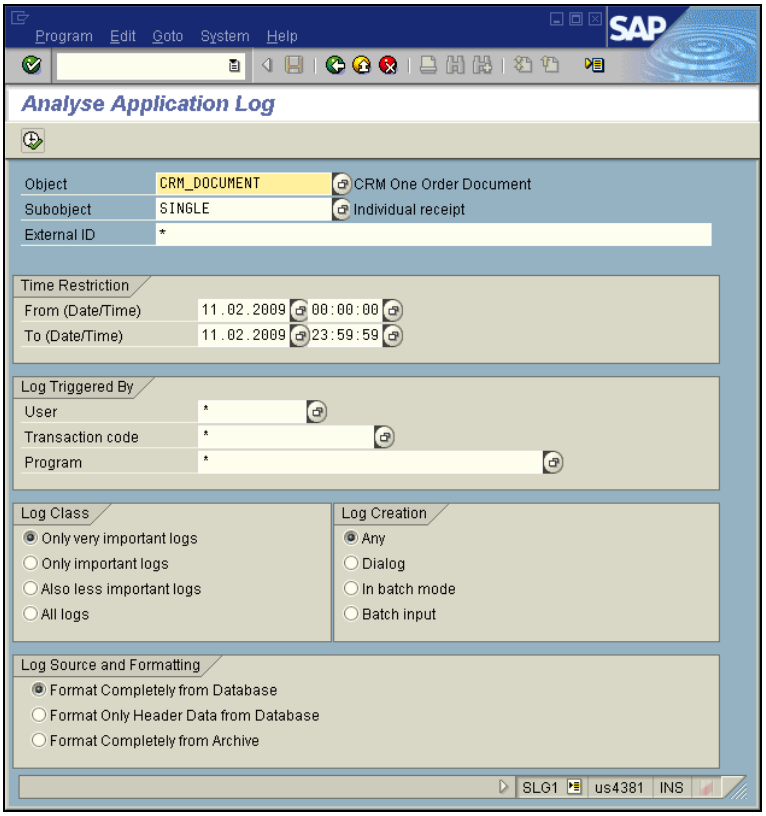


Figure 5-12 Selection of the Contract Application Log

- For a list of erroneous contracts create a query as a table join where `CRM_JEST-OBJNR` = `CRMV_ISU_INDEX_H-GUID` and check for datasets with `CRM_JEST-STAT = I1030` (status I1030 means *contains errors*) and `CRM_JEST-INACT = space`. The `CRMV_ISU_INDEX_H` table holds the IS-U contract number in the `ISUCONTRACT` field.
- Display the contract in the CRM interaction center. If there are errors, they are displayed in the message area of the screen.

5.3.5 Overview Business Objects and their Standard Filters

Figure 5-13 shows a list of all replication relevant business objects of SAP for Utilities together with their adapter objects and the predefined tables for request filters.

Business Object	Adapter Object	Table	Field Name	Field Description
Business Partner	BUPA_MAIN	BUT000	PARTNER BPEXT BU_GROUP BU_SORT1 BU_SORT2 TYPE	BP Number External BP number BP group Search term 1 Search term 2 BP category
Contract Account	BUAG_MAIN	FKKVKP BUT000	VKONT GPART PARTNER	CA Number Business Partner Business Partner
Business Partner Relationship	BUPA_REL	BUT050	PARTNER1 PARTNER2 RELTYP	Number of BP 1 Number of BP 2 Relationship Category
Business Partner Contact	SI_BCONTACT	BCONT BPCT_CRM_GUIDS	BPCONTACT PARTNER REPLCNT	BP contact number BP number Replication Control
Connection Object	SI_CONNOBJ	EHAUISU ECRM_TEMP_OBJ	HAUS REGIOGROUP XREPLCNT	CO number Regional group Replication control
Point of Delivery	SI_POD	EHAUISU ECRM_TEMP_OBJ	HAUS REGIOGROUP XREPLCNT	CO number Regional group Replication control
Service Contract	SI_CONTRACT	EVERH	VERTRAG BIS	Contract number Contract end date

Figure 5-13 Business Objects and their Standard Filters

6 Improving the Performance

The following chapters describe measures to improve the performance of the initial download. You need to verify which of the actions are of a temporary nature are hence active only during the initial download and you need to reverse on completion of the initial download. However, some of the settings may remain in the system because the improve the performance of the delta replication, such as, deactivation of message flow processing if there are no mobile business processes implemented.

The recommendations given below have to be understood as a collection of the possible measures that have been implemented in many projects in order to improve the performance of the initial download. Not all of them are applicable to any implementation and any business scenario.


Analyze each of the recommendations carefully prior to implementing them and, more importantly, you must test the results of the initial download meticulously prior to a go-live when implementing a change.

6.1 Support Package Level

You should make sure that you have implemented the latest support package on your ERP and CRM system. This provides the ideal basis for a high performance because with the implementation of the latest support package you implement implicitly also all recent performance relevant SAP Notes and further enhancements.


As of Basis Release 6.20, you import the current qRFC version only by Support Package. Previously, qRFC releases have been independent of the ERP and CRM release cycle. For more information, see SAP Note 438015.

6.2 General Recommendations

We do not recommend to permit production activities during the initial download in none of the involved systems. This is mainly due to the following reasons:

- The objective of the performance tuning activity is to maximize the usage of the system resources by the initial download. This affects the response time of any dialog user negatively.
- Some of the measures buffer information in memory that remains unchanged during an initial download. However, the same information can be subject to be changed as part of a business process. Buffering the information in memory might cause inconsistencies in the database once the system is in production.
- Production activities impedes a rollback of the production databases even if a reconciliation or a validation of the download data shows that only a rollback could save further business processes to be executed based on incorrect or inconsistent data.


Some of the recommended changes, such as, buffering of database tables in the main memory of the application server, affect not only the actual client in the SAP system but also any other client in the SAP system. Such changes are called cross-client changes. Therefore, access not only to a single client but moreover to the SAP system must remain limited even when interrupting the initial download activities only temporarily.

The general procedure to apply the changes to the systems is:

- Implement all changes in the development system and save the changes in a transport request. Release the transport request and write down the request number.
- Remove all implemented changes in the development system (return to the SAP standard) and save the changes in a second transport request. Release the transport request and write down the request number.

- Block any user activities in the SAP systems except for users who must have access to execute and to monitor the initial download.
- Re-import manually the first transport request to your test system.
- Execute the initial download
- Re-import manually the second transport request to remove the changes and return to the standard settings.

This procedure allows you to prepare a system for the initial download and set it back to its initial state once the initial download has finished. It is important to import the transport requests manually. Otherwise, the second transport request would remove implicitly all measures to improve the performance you have applied with the first transport request.

6.3 Reducing Data Volume during the Initial Download

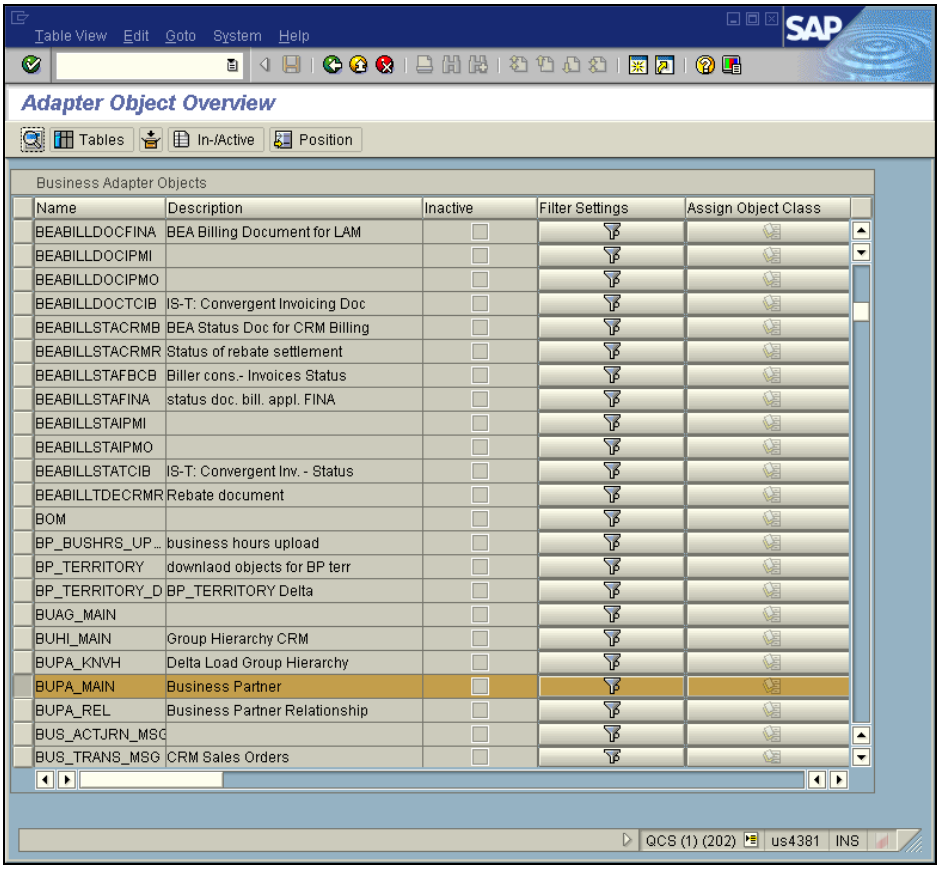
One major factor affecting the runtime of the initial download is the data volume you must download into the CRM system. If applicable, you can use general filter settings to reduce the downloaded data volume. Filter criteria are easy to implement, as no programming is required. The procedure for applying filters and some recommendations regarding filter settings are given in this chapter.

The transactions to specify the filter criteria for the various objects are:

- Business Objects: transaction R3AC1 (Adapter Object Overview Business Objects)
- Customizing Objects: transaction R3AC3 (Adapter Object Overview Customizing Objects)
- Condition Objects: transaction R3AC5 (Adapter Object Overview Condition Objects)

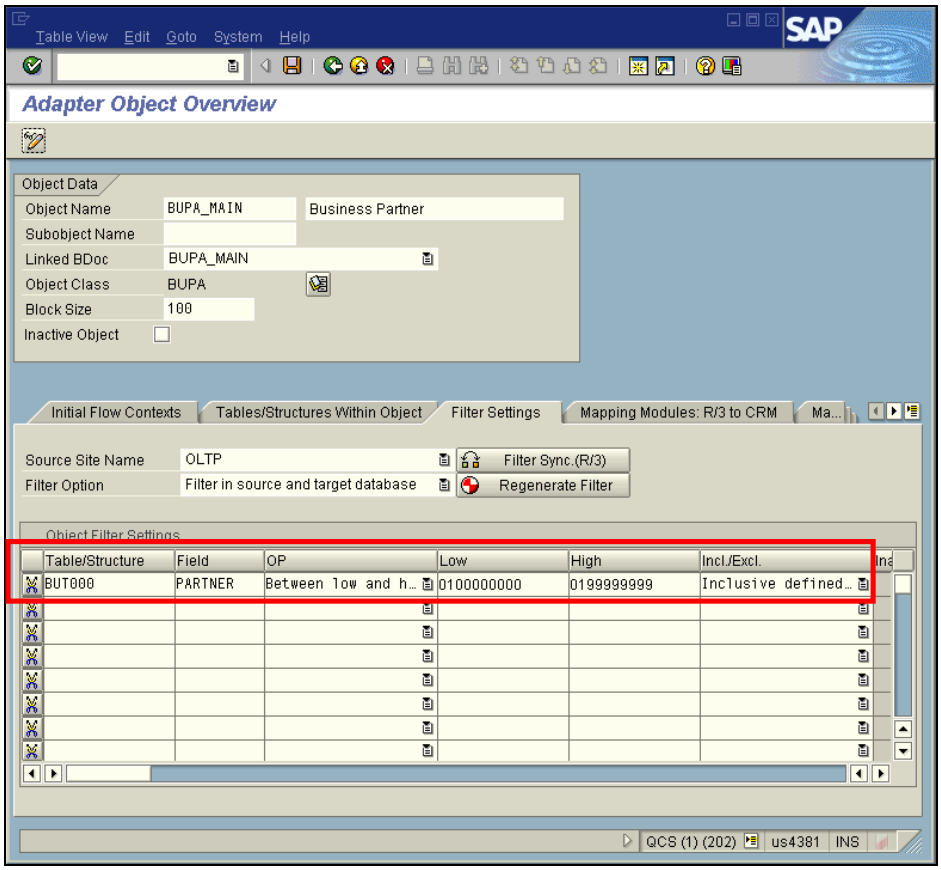
The filters are set in the CRM system. Figure 6-1 shows how to set the filters for a replication of business partners (BUPA_MAIN adapter object).

1. Call transaction R3AC1 (Adapter Object Overview). Then select the BUPA_MAIN adapter object.

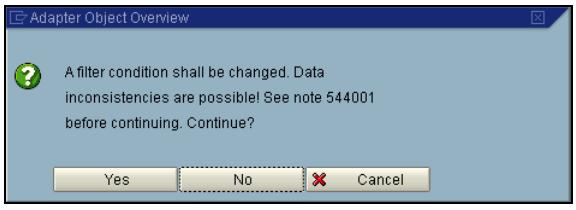


2. Push the ***Filter Settings*** button and then the ***Change*** button.

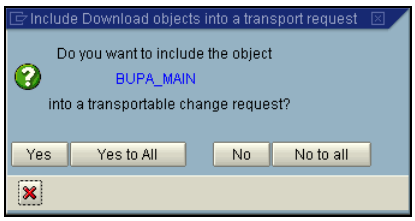
3. Change to the ***Filter Settings*** sub screen and enter the ***Source Site Name*** *OLTP* and the ***filter option*** *Filter in source or target database*. Then enter the lower and upper limit of the business partner IDs for the `BUT000-PARTNER` field.



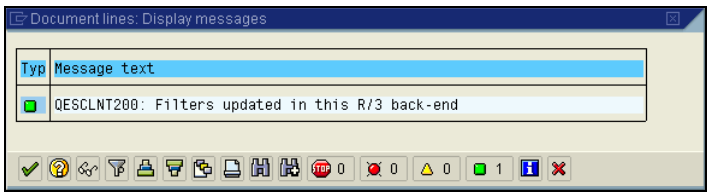
4. Push the ***Save*** button.



5. Push the ***Yes*** button.



6. Push the ***Yes to All*** button, if you define the filter settings in the development system and you want to transport them to the production system. Otherwise, push the ***No to All*** button.



7. Push the ***Continue*** button to confirm the popup.

Figure 6-1: Define Filter Settings

The filter settings are now the same in both CRM and ERP system.

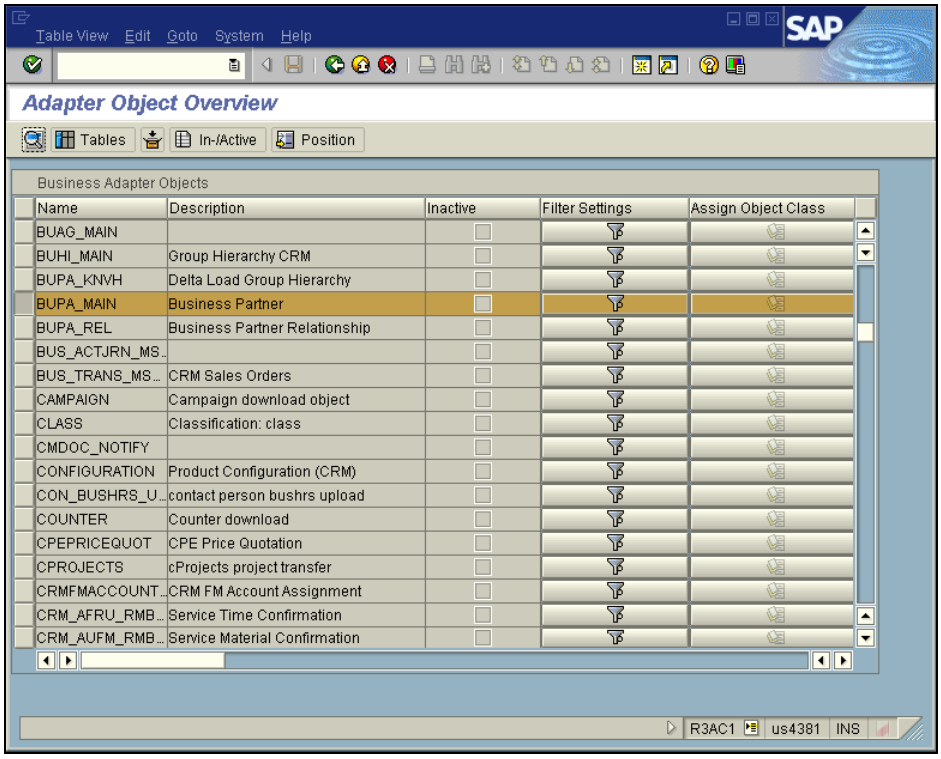
Some general remarks regarding filter settings:

- Filter options allow the filtering of business objects at the source, at the target or at both the source and the target for business objects. However, business data are usually filtered at the source. Customizing or condition objects can be filtered at the source only.
- Filter settings are stored in the table `SMOFFILTAB` on the CRM system and the `CRMFILTAB` table on ERP system and refer to table fields.
- Filters for the initial download are also used for the delta download
- If you use more than one filter entry per object, filters to the same table field are linked with an `OR`, e.g. the filter condition `VKORG = 0001`, `VKORG = 0002` results in a set that contains either the first or the second sales organization.
- Filters to different table fields are linked by `AND`, e.g. the filter condition `VKORG = 0001`, `VTWEG = 01` results in objects that fulfill both conditions at the same time.
- Filter conditions are only stored locally and are not contained in the transport of adapter objects. In this way, filters are not transported from the development system to the production system.
- Filter values must contain leading zeros for numeric identifier
- Filter values are case-sensitive. This means that data is filtered out if a filter condition is in lowercase but the data values are stored in uppercase.

6.4 Replication of multiple objects in one BDoc

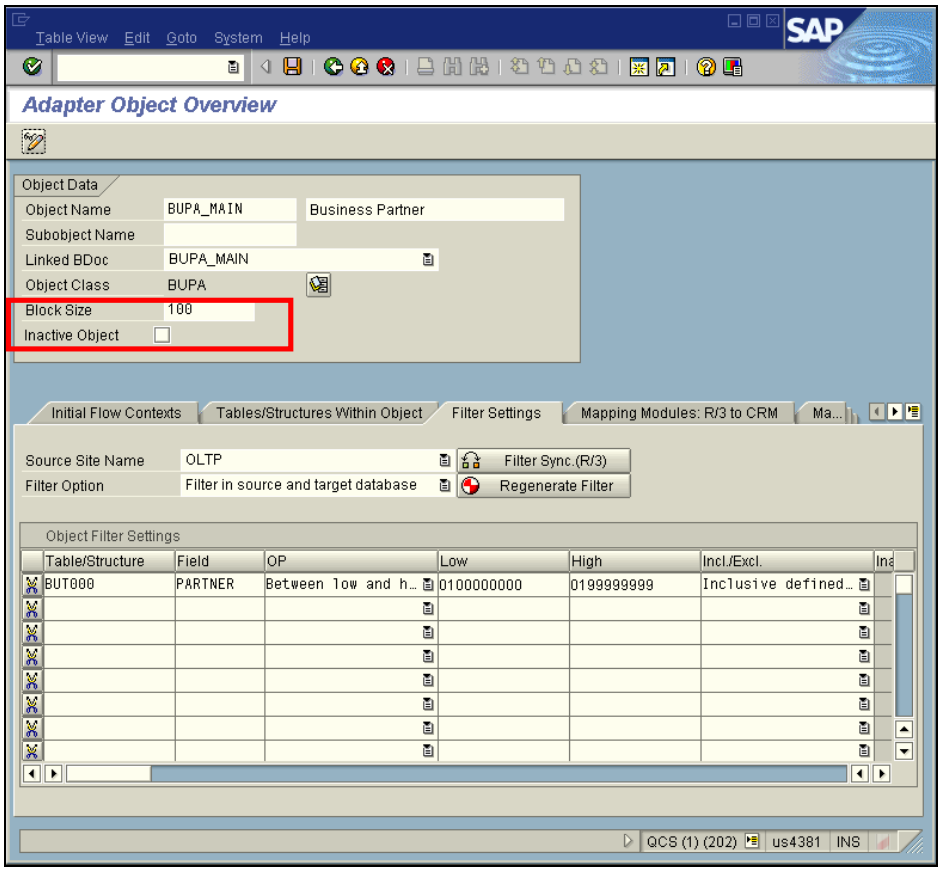
You can transfer more than one business object from the ERP into the CRM system in a single BDoc. This reduces particularly the administration activities of RFCs in both ERP and CRM system. As a result, the database load on the CRM and ERP systems is reduced significantly. Figure 6-2 shows how to define the block size on the CRM system.

1. Call transaction `R3AC1` (Adapter Object Overview). Then select the BUPA_MAIN adapter object.

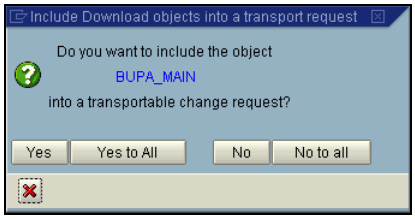


2. Push the ***Filter Settings*** button and then the ***Change*** button.

3. Enter the desired block size in the ***Block Size*** field.



4. Push the Save button to save the settings.



5. Push the ***Yes to All*** button, if you change the adapter object settings in the development system and you want to transport them to the production system. Otherwise, push the ***No to All*** button.

Figure 6-2: Definition of the Block Size

Repeat the steps above for all business objects to set the block sizes for the respective adapter objects. Of course, the best block size in your project has to be determined during volume and performance tests.

We do not recommend to spend too much time in determining an exact optimal number of the block size. It is much more important to replicate more than one business object in one BDoc. Figure 6-3 shows the commonly used parameter values.

Adapter Object	Block Size
BUPA_MAIN	50 - 100
BUAG_MAIN	50 - 100
SI_CONNOBJ	50 - 100
SI_POD	< 50
SI_CONTRACT	30

Figure 6-3 Recommend Values for the Block Size Parameter

6.5 Parallel Processing

Parallel processing of the initial download is very effective when you have to replicate high volumes of business objects. Only parallel processing will normally allow to complete an initial download within a given period. There are two possibilities to parallelize the initial download with requests:

- Parallel processing of a request with multiple inbound queues in the in the CRM system, if the process to select data of a BDoc in the ERP system is faster than the processing of the BDoc in the CRM system
- Parallel processing of requests, if a single request with multiple inbound queues cannot utilize the system resources of the CRM system sufficiently

You must analyze the available CRM system resources carefully, when parallelizing the processing of BDocs. Especially a high number of available dialog work processes is essential for a high level of parallelization. This is because each inbound is processed by one dialog work process whereas the usage of the available dialog work processes is controlled by RFC logon group assigned to the QIN scheduler (for more information, see chapter 6.5.3 *Distribution of RFCs*). If due to the configuration more dialog work processes are to be used than available, RFCs are queued further in the CRM system increasing the data load in the CRM system due to additional administration activities as a result.

You can check the resource consumption in the CRM system. Call transaction SMQS (Inbound Queue Monitor), then choose *Goto → qRFC Resources*. Figure 6-4 shows the resource monitor of the QIN scheduler. Alternatively, you may use transaction SARFC (Server Resources).

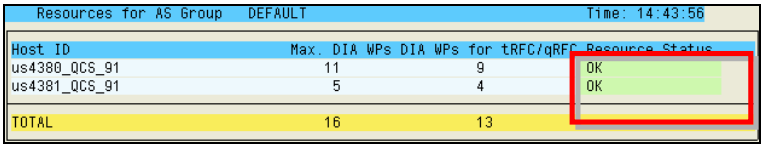


Figure 6-4 Resource Monitor in the QIN Scheduler

6.5.1 Request with multiple Inbound Queues

The CRM_MAX_NO_QUEUES_PER_REQUEST parameter in the CRMPAROLTP table controls the maximum number of inbound queues on the CRM system per request of a specific adapter object. For information how to define the CRM_MAX_NO_QUEUES_PER_REQUEST parameter, see chapter 7.1.1 *Important Parameters in CRMPAROLTP Table*.

In theory, you can define up to 99 inbound queues per request. In praxis, you will have no more than 20 to 30 inbound queues per request active at the same time. You use the CRM_MAX_NO_QUEUES_PER_REQUEST parameter to enable parallel processing with multiple inbound queues and, more importantly, to limit the usage of dialog work processes during the processing of a request. You can adjust the CRM_MAX_NO_QUEUES_PER_REQUEST parameter also online to control the data load on the CRM system.

6.5.2 Processing of multiple Requests in parallel

You can execute the data selection in the source system in parallel by defining and starting several requests. The MAX_PARALLEL_PROCESSES parameter controls how many requests may be active concurrently. However, if you start more requests than the defined limit, the requests that exceed the limit (defined with the MAX_PARALLEL_PROCESSES parameter) remain in their outbound queues in the status STOP until one of the active requests finishes. For information how to define the CRM_MAX_NO_QUEUES_PER_REQUEST parameter, see chapter 7.2.1 *Important* Parameters in SMOFPARSFA Table.


You can create requests from a text file that is located on your desktop. SAP Note 770849 describes a customer-specific report you can use as a basis for an own development.


You should start only requests of the same adapter object concurrently that do not have any overlapping intervals in order to prevent a concurrent processing of the same business object by two or more requests. Overlapping intervals may cause inconsistencies and usually lead to short dumps of processes in the CRM system (inbound queues) and the related inbound queues on the CRM system show the status SYSFAIL. As a result, the requests will have to be re-processed. Therefore, you must plan and define the intervals per request carefully in order to avoid overlapping intervals. A common mistake is, to define two requests whereas the upper interval limit of the first request corresponds to the lower limit of the second request.

6.5.3 Distribution of RFCs in the CRM System

By default, the QIN Scheduler processes all the registered queues using the dialog work processes of all instances[1] of the CRM system. An RFC server group controls the usage of dialog work processes on the available instances.

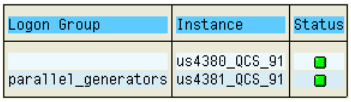
The name of the default RFC server group is DEFAULT. The server group DEFAULT does not need to be created and it is not visible in transaction RZ12. However, we recommend to create a RFC server group specifically for the initial download in order to release as many dialog work processes as possible and necessary for the processing of the BDocs.

[1] SAP instances are defined during the installation of the SAP system. The term “instance” is frequently used synonymously with “server” or “application server” in the SAP system. This depends on the selected installation environment. You can install multiple instances on one application server.

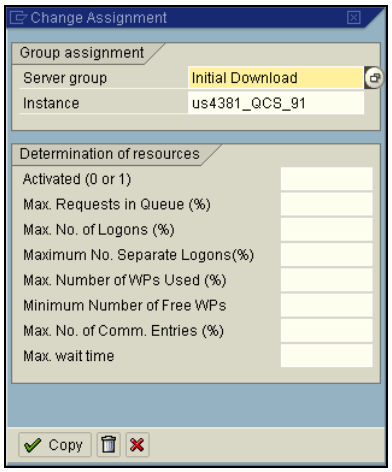
You must adjust the RFC server group after the initial download. This is because you maximize the availability of dialog work processes for an RFC processing. This reduces the availability of dialog work processes for dialog user after the initial download significantly affecting the response time negatively

Figure 6-5 shows how to define a RFC server group.

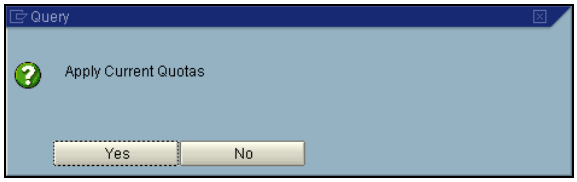
1. Call transaction R3AC1 (Adapter Object Overview). A table is displayed showing:
   - The names of already-defined RFC groups, or "logon groups" or "server groups"
   - A list of the servers, or "instances," in your SAP System
   - The current status (running/not-running) of each server



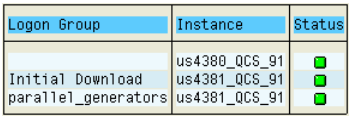
2. Choose *Edit* → *Create assignment* and in the ***Server group*** field, either choose a name from the list of already defined groups or type a new name, for example, *Initial Download*. In the ***Instance*** field, select one of the servers from the list. You may want to leave the quotas empty to copy them from the system defaults.



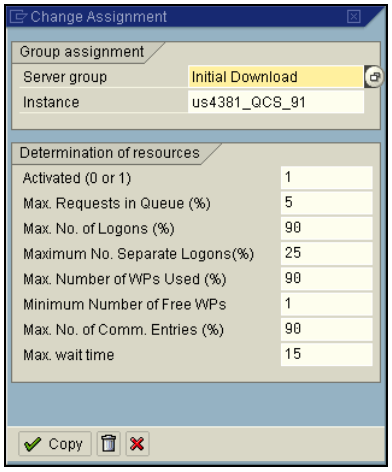
3. Push the ***Copy*** button.



4. Push the ***Yes*** button to confirm. A table with the newly defined assignment is displayed.



5. Repeat this step for each server that should belong to the group. When a server belongs to more than one group, processes that use the group will compete for free work processes on the shared server(s).
6. Position the cursor on a valid group assignment and choose *Edit* → *Change Assignment* to adjust the RFC quotas for the assignment.



7. Push the ***Save*** button to change the changed quotas.
8. Repeat this step for each server that belongs to the group

Figure 6-5: Definition of an RFC Server Group

The RFCs quotas depend on the available system resources. Figure 6-6 explains each of the quotas in details.

Parameter	Meaning	Profile Parameter
Activated (0 or 1)	0: Resource determination is deactivated 1: Resource determination is activated	`rdisp/rfc_use_quotas` Default: 1
Max. Requests in Queue	Quota for the maximum number of waiting requests in the dialog queue of the dispatcher	`rdisp/rfc_max_queue` Default: 5 % of total capacity (`rdisp/elem_per_queue`)
Max. No. of Logons	This value specifies the maximum percentage of logons to this instance (maximum total number `rdisp/tm_max_no`) due to asynchronous RFCs. The remaining percentage (default: 10%) is then reserved for dialog and HTTP users.	`rdisp/rfc_max_login` Default: 90 %
Max. No. Separate Logons	Quota for the number of own logons. This value specifies the maximum percentage of logons to this instance (maximum total number `rdisp/tm_max_no`) due asynchronous RFCs of one user (that is, of one application that is using parallel RFCs).	`rdisp/rfc_max_own_login` Default: 25
Max. Number of WPs Used	This value specifies the maximum percentage of the dialog work processes that can be occupied by one RFC user. This quota prevents all dialog work processes being occupied by RFCs of one user.	`rdisp/rfc_max_own_used_wp` Default: 75 %
Min. Number of Free	Minimum number of free dialog	`rdisp/rfc_min_wait_dia_wp`

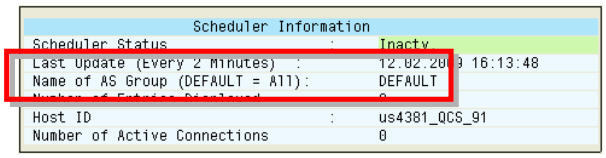
WPs	WPs. This parameter must have a value of least 1. Number of dialog WPs the initial download cannot occupy. At least this number of dialog work processes is therefore reserved for dialog users.	Default: 1 Note: You should have at least 3 dialog WPs per server available because the system often needs free dialog WPs for short periods to allow system monitoring, filling number range buffering and other system activities. You may have to setup additional dialog WPs in the operation mode that the number of WPs you want to use for the initial download and the free dialog work processes the system requires are both available.
Max. No. of Comm. Entries	This value specifies the maximum percentage of communication entries of an instance (`rdisp/max_comm_entries`) may be occupied by parallel RFCs.	`rdisp/rfc_max_comm_entries` Default: 90 %
Max. wait time	Quota for the maximum wait time after load check in the system	`rdisp/rfc_max_wait_time` Default 10 s

Figure 6-6 Quotas to control Usage of Dialog Work Processes

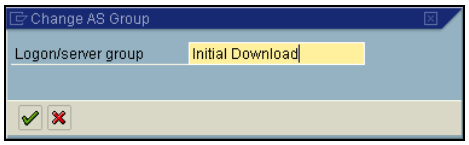
It might be necessary to increase certain the gateway parameter accordingly (for additional information how to setup an RFC environment, see SAP Note 74141).

You must assign the newly defined RFC server group to the QIN scheduler. Figure 6-7 shows, how to change the QIN scheduler relevant RFC server group.

1. Call transaction SMQR (qRFC QIN Scheduler).



2. Choose *Edit* → *Change AS Group*. Enter the name of the RFC server group you want to use during the initial download, for example, *Initial Download*.



3. Push the ***Continue*** button

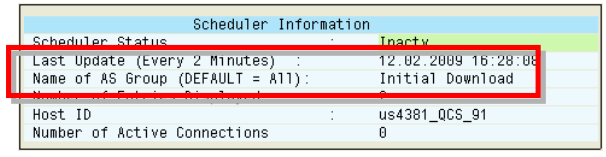


Figure 6-7: Assign new RFC Server Group to QIN Scheduler

If it appears that the QIN scheduler does not utilize all available resources, check whether SAP Note 1115861 is implemented on your system.

6.5.4 Operation Mode during Initial Download

An operation mode defines a resource configuration for the instances in your ERP and CRM system. It defines the number of work processes used for each service in an instance (application server) and the times that the services are available.

You can define separate operation modes to suit specific system requirements, for example, to provide additional dialog or background processing resources during a particular period without having to restart the SAP system. Usually, three operation modes are defined in a CRM system: operation mode *DAY*, operation mode *NIGHT*, and an operation mode *EXCEPTIONAL*. This allows guaranteeing response times for important data entry transactions during the day (dialog processing), and using more work processes at night for job processing (non-dialog processing with background jobs).

During an initial download, the inbound queues require a high number of dialog work processes to process the RFCs respectively the sent BDocs. We recommend to define a special operation mode *INITIAL DOWNLOAD* in order to maximize the availability of dialog work processes during the initial download. Before starting the initial download, you need to set the operation mode *INITIAL DOWNLOAD* as an exception operation mode in order to avoid an automatic operation mode switch based on the defined timetable. You create and maintain an operation mode with transaction `RZ04` (Maintain Operation Modes and Instances). Figure 6-8 shows the procedure to set the exception operation mode.

1. Call transaction `SM63` (Display/Maintain Operation Mode Set)
2. Choose ***Exception operation***. Then push the ***Change*** button.
3. You can now change the date on which the operation mode is to be activated. Choose *Day+1*, *Day-1*, *Specify a day* accordingly
4. Choose *Assign* to set the time at which the operation mode is to be activated. You do not need to define a full 24 hour cycle for exception operation. You must only define the period in which you want the operation mode to be active.
5. Push the ***Save*** button to save the changes.

Figure 6-8: Set the Exception operation Mode

6.5.5 Business Locks

During the creation or a change of business objects, the application protects business objects by means of locks to prevent a concurrent change access to the same business object by another process. The enqueue server (also known as the lock server) is the SAP system component that manages the lock table (enqueue table) this is a list of all active locks (enqueues). There is only one enqueue server in a distributed SAP system. There are various installation options for this server:

- The enqueue server is configured as an enqueue work process in an instance. This instance is known as the “central instance”. This installation option is used mainly in older ABAP-only systems and in systems that consist of one instance only.
- The enqueue server is installed as part of a separate instance. This is when the standalone enqueue server is used. Together with the message server, this server becomes the SCS instance (SAP, Central Services – AS Java) or the ASCS instance (ABAP Central Services – AS ABAP). This means that this instance is a single point of failure, which can be designed for high availability, together with the enqueue replication server.

Most commonly, projects choose to configure the enqueue server with enqueue work processes in the central instance. The enqueue server receives a lock request and checks the lock table to determine whether the lock request collides with an existing lock. If it does,

the enqueue server rejects it. If it does not collide, the enqueue server sets the lock and makes the necessary entry in the lock table.

The lock table is located in the main memory (shared memory) of the central instance. All work processes on the central instance have access to the lock table. External application servers execute their lock operations in the enqueue process on the central instance. The communication occurs through the relevant dispatcher and the message server.

There are three potential bottlenecks:

- **Size of the lock table:** Due to the high parallelism of the initial download on the CRM System, many locks are active at the same time. The standard size of the lock table might not be sufficient to cope with that many locks during an initial download and a lock table overflow occurs. The maximum number of locks in the lock table is subject to an upper limit that is determined by the `enque/table_size` profile parameter. The `enque/table_size` profile parameter determines the size of the lock table in Kbytes. You can monitor the usage of the lock table with transaction `SM12`. Choose *Extras* → *Statistics* to display the lock table statistics and to determine both the maximum number and current number of entries.
- **Response time of the enqueue work process:** A throughput bottleneck may occur if the enqueue work process has to process too many lock requests in a short period. A high CPU consumption of the enqueue work process (monitor CPU consumption in transaction SM51) and/or a high enqueue queue fill level (monitor queue information in transaction `SM51` or CCMS alerts) indicate this. You can improve the throughput and response time of lock requests by increasing the number of enqueue work processes. You configured too many enqueue work processes, if the CPU time of more than one enqueue work process remains almost at zero.
- **Throughput of the dispatcher**: With a high enqueue load, also the dispatcher of the enqueue server can cause a bottleneck. With a higher number of enqueue work processes, you can also improve the throughput of the dispatcher. This is because the dispatcher can then transfer several lock requests to the enqueue work processes during each round.

If necessary, you can configure several enqueue processes on an enqueue server. For very large installations, up to approximately four enqueue processes are useful.


If the number of enqueue work processes is too high, it is possible that a performance loss occurs again due to the increased CPU requirements for context switch, cache coherence and so on. For more information, see SAP Note 127773.

6.6 Message Error Handling

You can use the error handling function to process error situations of a message in the framework of message flow control. You can assign one or more error actions to a message. This is a very useful function in the operation of a production system. However, particularly during the first test cycles of an initial download you must expect a high volume of erroneous data in BDocs due to data migration problems on the ERP system. An automatic system response to report the error, such as sending an email notification, has a negative impact on the system performance. Call transaction `SMW00` to deactivate the automatic system response to errors during the initial download.

6.7 Deactivation of Message Flow Processing

By default, CRM system creates further BDoc messages during the processing of the BDoc messages sent from the ERP system in order to distribute created or changed data to further connected systems. These BDoc's are processed in inbound queues with the prefix *CSA*.

In a mobile scenario or groupware connection, a data transfer between the CRM system and the consolidated database (CDB)[2] on the CRM system is required. If a BW connection exists, data exchange between the CRM system and the BW system is also carried out. In the message flow, the created BDoc messages are written to *CSA* queues on the CRM system. Although it is an outbound flow, inbound queues are used for *CSA* queues due to performance reasons.

Only few SAP for Utilities installations use a SAP CRM Mobile Client Application. Therefore, you can improve the performance of an initial download in deactivating the Mobile Bridge and any further distribution of BDoc messages. Due to the deactivation, more system resources are available to process the BDocs sent by the ERP system.

We recommend to deactivate the delta load to BW on the CRM system when performing an initial download. On completion of the initial download, an initial load into the BW system of the newly created or changed data is required. If you need the delta load into BW or other systems you have to reactivate the *CSA* queues after the initial download.

You deactivate the message flow processing based on BDoc types. Figure 6-9 shows a list of processed BDocs during the initial download of a contract and its dependant business objects. The marked lines indicates the BDocs created and processed by *CSA* queues.

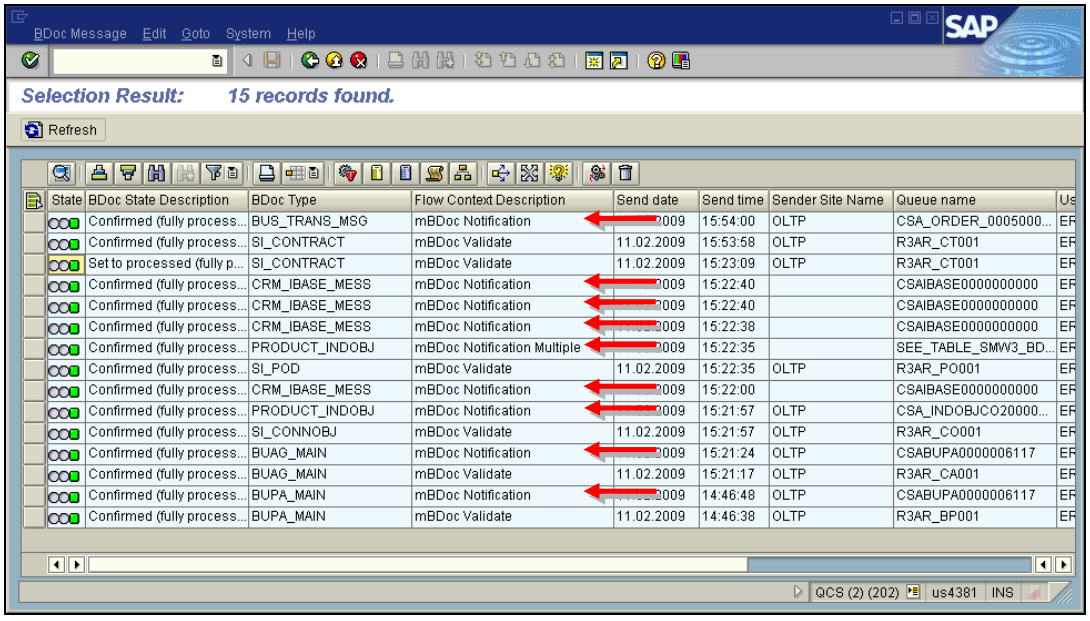
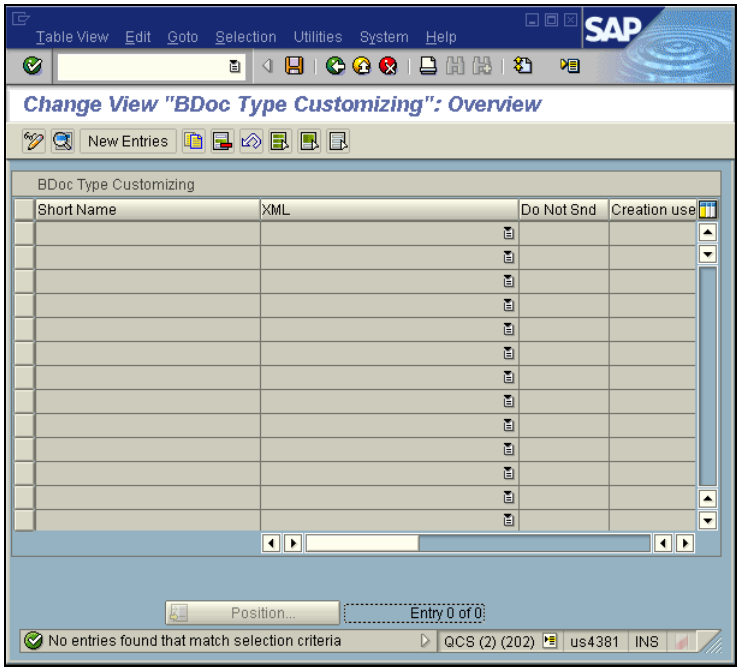


Figure 6-9 Screen to display the Status of processed BDocs

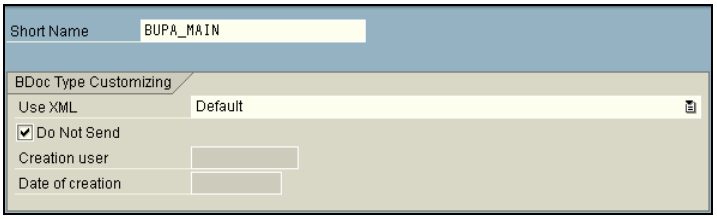
Figure 6-10 shows, how to deactivate of the message flow of BDoc type BUPA_MAIN.

1. Call transaction `SMW3_00` (BDoc Type Setting).

[2] A Consolidated Database contains data relevant to replication (CDB tables) for mobile scenarios. The CDB is a logical part of the entire CRM database, which also contains application data.



2. Push the ***New Entries*** button. Enter *BUPA_MAIN* in the ***Short Name*** field, *Default* in the ***Use XML*** field and mark the ***Do not send*** checkbox.



3. Push the ***Save*** button

Figure 6-10: Change BDoc Type Customizing for BDoc Type BUPA_MAIN

Repeat the steps above to deactivate the message flow of all relevant BDoc types and respective adapter objects.

Adapter Object	**BDoc Type not to be sent**
BUPA_MAIN	BUPA_MAIN
BUAG_MAIN	BUAG_MAIN
SI_CONNOBJ	CRM_IBASE_MESS PRODUCT_INDOBJ MSV_IBA_WRITE
SI_POD	CRM_IBASE_MESS PRODUCT_INDOBJ MSV_IBA_WRITE
SI_CONTRACT	BUS_TRANS_MESS

Figure 6-11 Subscriptions for SAP for Utilities

Figure 6-12 shows the Customizing with all relevant BDoc types.

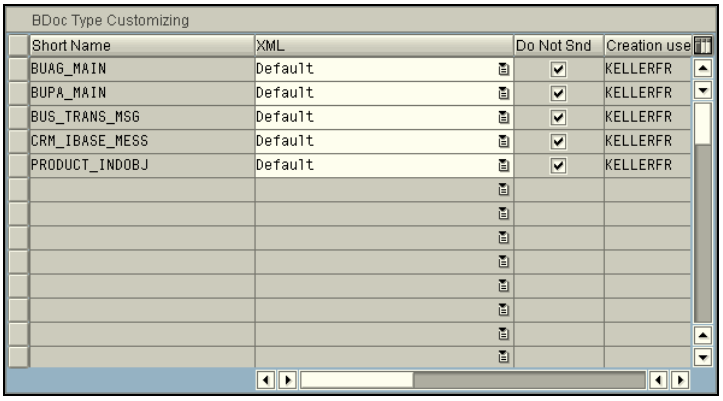


Figure 6-12 Overview Screen for BDoc Type Settings

The following figure shows the processed BDocs after the deactivation of the message flow processing. A comparison with Figure 6-9 shows that *CSA* queues are not processed further.

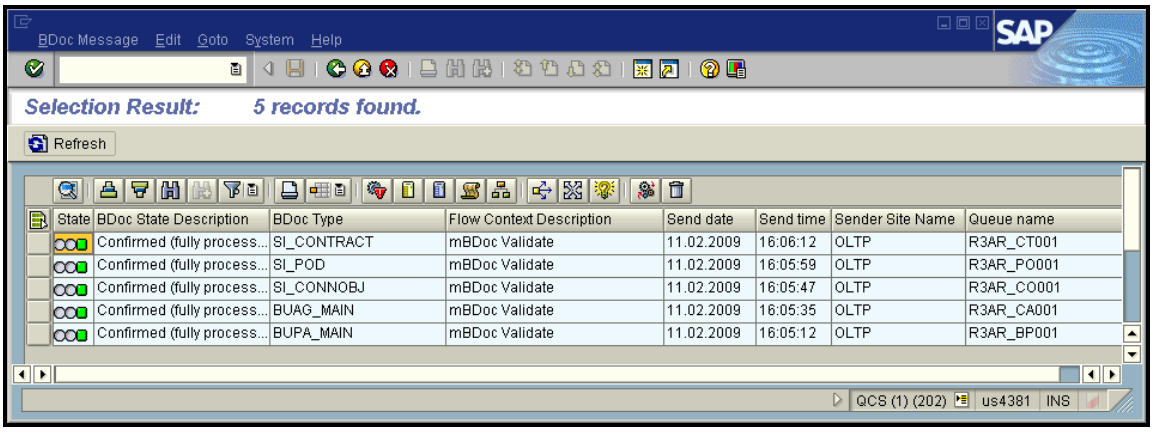


Figure 6-13 Screen to display the Status of processed BDocs

6.8 Data transfer using XML

The XML interface consists of an XML (eXtensible Markup Language) generating program on the ERP Plug-In and an XML parser on the CRM system. The generating program generates an XML document from the data stream that is sent to the CRM system, where the XML parser converts the data back to the original data stream. In a heterogeneous system landscape (for example, if servers with different CPU architecture are involved), data inconsistencies during data transfer can occur caused by the Endean problem. The data transfer in XML format solves both the Endean problem and the code page problem which may occur in a data exchange.

The XML format has the performance disadvantage that the XML parser needs CPU time. In some special cases, it may be possible to deactivate the XML conversion during the initial download to improve the performance. With CRM 4.0 and the applicable PI on the ERP system, the option `SEND_XML = M` (=mixed) is possible. The option `SEND_XML = M` in the `CRMRFCPAR` table specifies that only parts of the `BAPIMTCS` container used for the data exchange between the ERP System and the CRM system are sent using XML. Character-type fields are transferred with high performance. For non character-type fields, an XML conversion occurs in spite of the 'M' entry that has a negative impact on the performance.

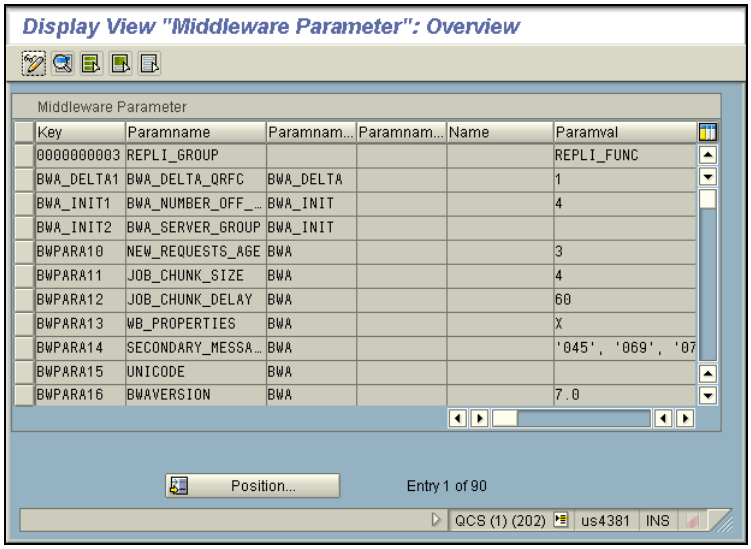
We recommend to use always the *Mixed Mode (Optimized)* mode. For more information, see SAP Note 510192. Figure 3-7 shows, how to define the data transfer mode.

6.9 Flow Control Trace (CRM System)

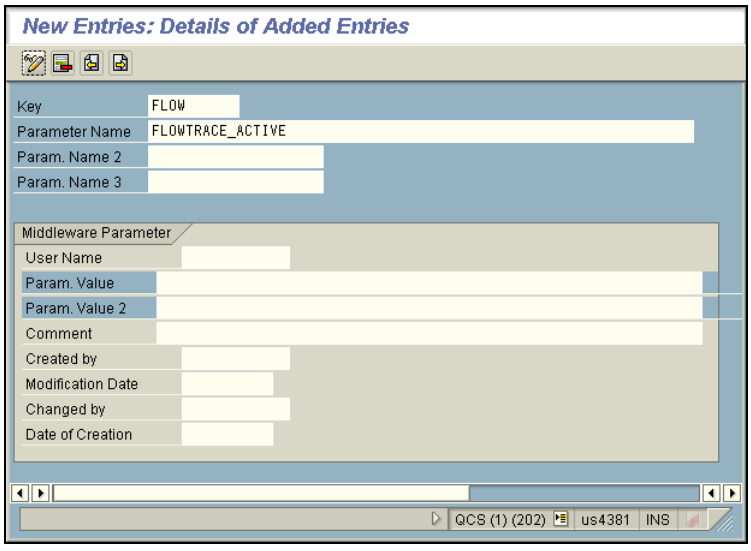
The log entries are saved in the database. Furthermore, all messages (also the processed ones) are stored in other tables. This is causing a high load on the CRM database server. Therefore, you could deactivate the flow control (see also SAP Note 206366 for further details). You can do this in either removing the FLOWTRACE_ACTIVE parameter or setting the parameter value ***PARAM. VALUE*** = space (see SAP Notes 204172 and 311946 for further details).

Figure 6-15 shows, how to deactivate the flow control trace.

1. Call transaction R3AC6 (Define CRM Middleware Parameters).



2. Check whether the FLOW FLOWTRACE_ACTIVE parameter already exists.
3. If the parameter already exists, select the parameter and either push the ***DELETE*** button to delete the entry or push the ***Change*** button to change the parameter value.
4. If the parameter does not exist, push the ***New Entries*** button. Enter *FLOW* in the ***Key*** field, *FLOWTRACE_ACTIVE* in the ***Parameter Name*** field and clear the ***Parameter Value*** field empty.



5. Push the ***Save*** button.

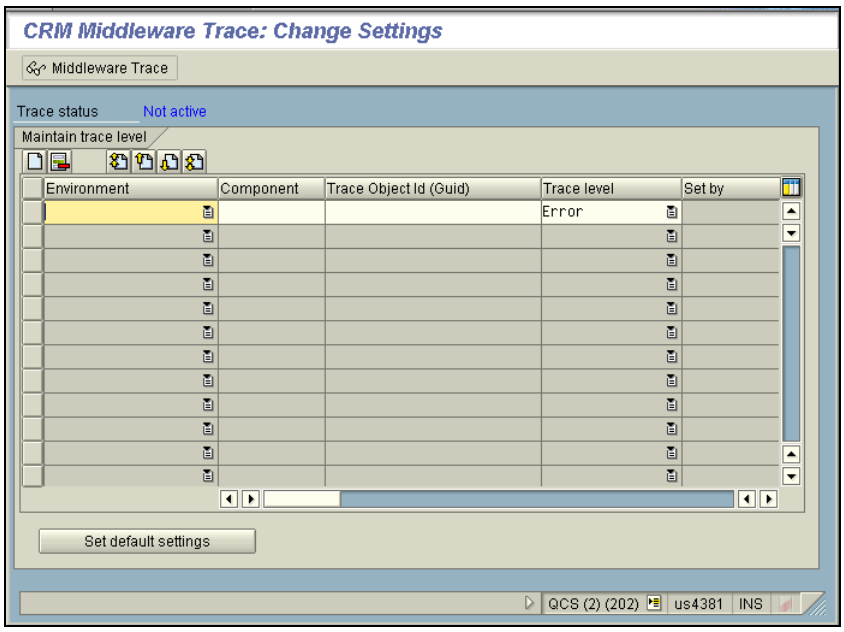
Figure 6-14: Deactivation of the Flow Control Trace

6.10 CRM Middleware Trace (CRM System)

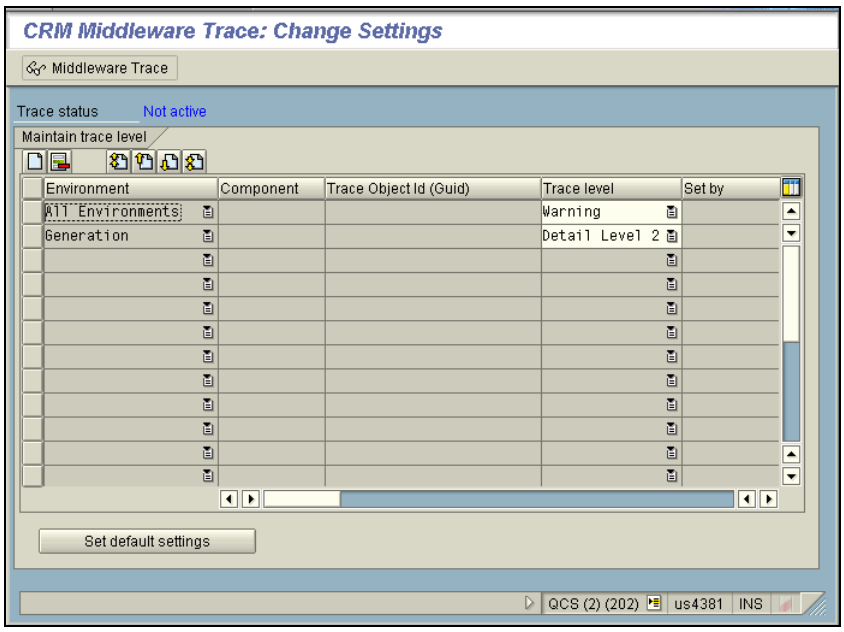
The CRM Middleware can create traces in order to monitor the BDoc processing. You can analyze the trace with transaction SMWT (CRM Middleware Trace). During normal operation and particularly during the initial download, you should set the trace level to ***Error***. Otherwise, a large amount of trace data is saved in the database (SMWT_TRC table). This has a negative impact on the system performance. You should increase the trace level only during active error monitoring. For more information, see SAP Note 206439.

Figure 6-15 shows, how to change the trace level.

1. Call transaction SMWTAD (Change CRM Middleware Trace).



2. Push the ***Set default settings*** button.



3. Change the ***Trace level*** for *All Environments* to *0 Error*
4. Push the ***Save*** button.

Figure 6-15: Change the CRM Middleware Trace Level

6.11 Object Links in BDocs (CRM System)

Object links are created for effective identification and analysis of the objects and the related BDoc messages that are formed in a single transaction. With object links, the CRM Middleware provides the end user with data to analyze the application data flow starting from the inbound flow, the business object creation to the outbound flow. You can do the analysis from the BDoc store (transaction SMW01) by selecting neighbors or linked BDocs for a particular BDoc.

The information about the relation is stored in the SMW0REL and SRRELROLES tables. A periodic job reorganizes these tables. If there are many BDocs within a short period, the number of entries in the relevant tables grows and causes a huge performance impact. By disabling the object links, the performance of the system will improve. However, the information that relates to the different objects created during the transaction will not be stored.

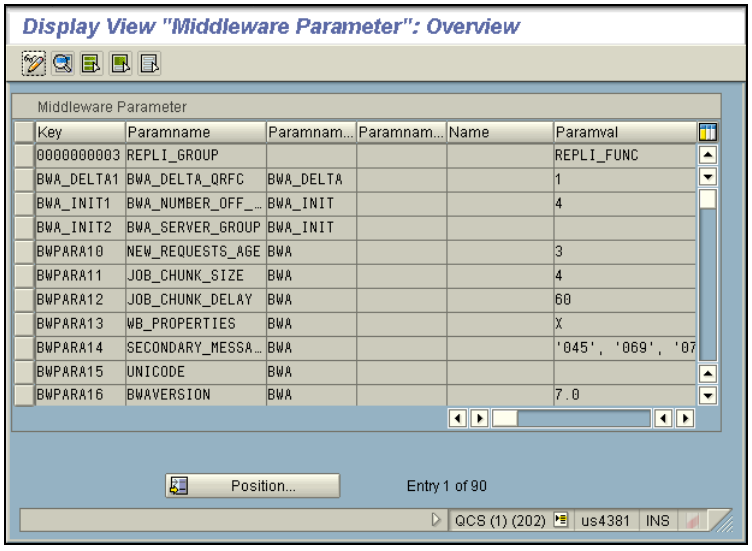
Figure 6-16 shows the relevant parameter names to deactivate the creation of links between objects. For more information, see SAP Notes 792954 and 1254264.

Parameter Name	Description
MW_NO_INMBDOC_OBJ_LINKS MW_NO_OUTMBDOC_OBJ_LINKS	Prevents the creation of a link between an mBDoc and the related object
MOBBRIDGE_NO_LINKS	Prevents the creation of a link between an outbound mBDoc and the outbound sBDoc, if there is an active mobile bridge
MW_NO_MBDOC_LINKS	Prevents the creation of a link between an inbound mBDoc and an inbound sBDoc, if there is an active mobile bridge

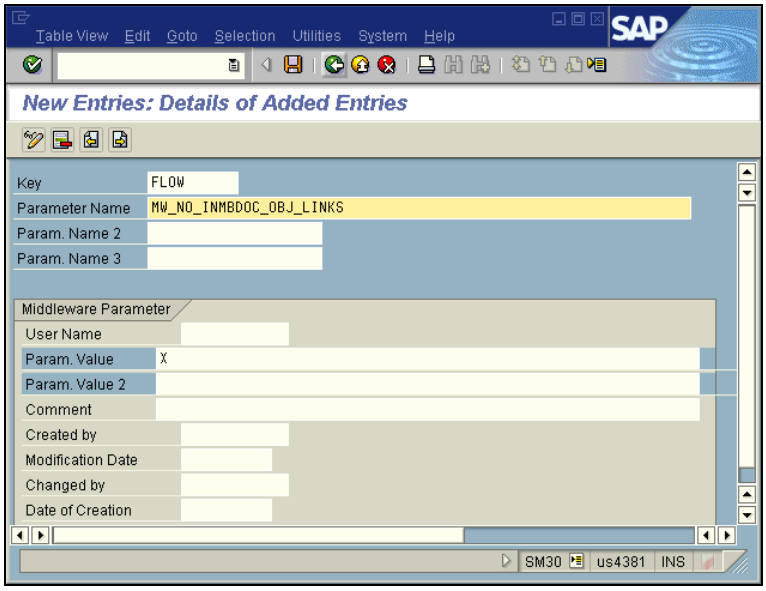
Figure 6-16 Parameters to Deactivate the Creation of Object Links

The following figure shows, how to configure the parameter to deactivate the object links.

1. Call transaction R3AC6 (Define CRM Middleware Parameters).



2. If the parameter does not exist, push the ***New Entries*** button. Enter *FLOW* in the ***Key*** field, *MW_NO_INMBDOC_OBJ_LINKS* in the ***Parameter Name*** field and X in the ***Parameter Value*** field.



3. Push the ***Save*** button.

Figure 6-17: Deactivation of the Object Links

6.12 Business Warehouse

Generally, you initiate the extraction of business partners from the CRM system to the BW system on completion of the initial download. That is, in most cases no further initial download occurs after a delta initialization. Only if you create or change individual business partners, an update to the active BW delta queues occurs. The changed data is loaded to the BW system using Delta Updates.

If you have already started a delta initialization for business partners from the Business Information Warehouse there is an active delta queue for at least one connected BW system. After the delta initialization, the system carries out an update to the active delta queues of the corresponding Data Source during an initial download of business partners from the ERP system into the CRM system.

If you execute the initial download of business partners from the ERP system into the CRM system after you have started a delta initialization for the relevant business partner from a connected BW system, the CRM system treats the initial download as a mass update. That means that the system writes many data records to the corresponding BW delta queues in a very short period.

To prevent this, the `CRM_SUPPRESS_BW_UPD` implementation of the `BUPA_BW` BAdI is available as of CRM Release 5.0. SAP delivers this BAdI deactivated. By activating the BAdI implementation, you can suppress the update to the BW delta queues of the business partner during an initial download of business partners from the ERP system into the CRM system.

If you have activated the BAdI implementation, you must carry out a full update of the business partner from the CRM system to the connected BW system after the initial download in order to ensure data consistency in the BW system. For more information, see SAP Note 825232.

⚠ Deactivate the BAdI implementation on completion of the initial download.

6.13 Buffering Database Tables in Memory

Buffering a table improves the performance when accessing the data records contained in the table. The table buffers reside locally on each application server in the system. The data of buffered tables can be accessed directly from the buffer of the application server. This avoids the time-consuming process of accessing the database.

If a program accesses the data of a buffered table, the database interface determines whether this data is in the buffer of the application server. If this is the case, the data is read directly from the buffer. If the data is not in the buffer of the application server, it is read from the database and loaded into the buffer. The next access to this data can then use the buffer.

It is noted in the log table by the database interface if a program changes the data contained in the table on an application server, this. The buffers still have the old status on all the other application servers, so that the programs might read obsolete data. A synchronization mechanism runs at a fixed time interval, usually every 1-2 minutes. The log table is read and the buffer contents that were changed by other servers are invalidated. In the next access, the data of invalidated tables is read directly from the database and updated in the buffer.

During the initial download, you can buffer temporarily additional tables in memory, even SAP delivers them with the ***Buffering note allowed*** indicator. You may buffer only tables that are accessed during the initial download by a read access only and contain only very few rows. You must not buffer a table if the table data is frequently modified. In this case the cost of synchronization could be greater than the gain in performance resulting from buffering. Roughly speaking, it does not make sense to buffer a table if more than one percent of the accesses to the table are modifying accesses.

Prior to buffering a table in memory, you should run an initial download without additional table buffering. After the initial download you must check, which tables have been populated during the initial download. Only tables with no additional rows should be buffered in memory. Specifically in a multi-client system, for example, your QA system (quality assurance system), you must expect that during tests in a second client tables have been filled by online transactions which are not filled by an initial download excluding such tables from being buffered in memory. It is good practice to ask the database administrator to check the number of rows on database level instead of just using transaction SE16 (Data browser).

SAP delivers also tables that are already buffered in memory. The assumption is that applications change such tables only rarely. This situation might change when a high volume of business objects are created in the SAP system. In such situations, you must change the buffering setting of such a table to ***Buffering allowed but switched off*** in the technical DDIC settings of that table.

On completion of the initial download, you must return to the standard setting in order to avoid inconsistencies in the database.

Prerequisite for table buffering is a table buffer that is large enough to hold all buffered tables. The size of the generic key table buffer *TABL* is determined by the following SAP system profile parameter. The efficiency of the resident-table buffer depends not only upon its size but also upon the number of directory entries in the buffer (the maximum number of tables that can be buffered).

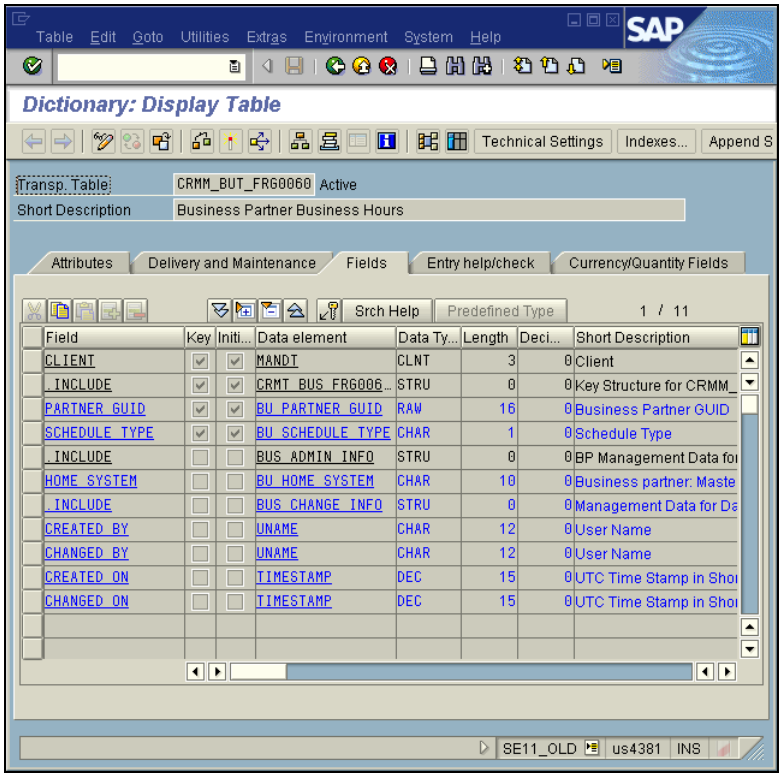
- `zcsa/table_buffer_area`: Specifies the size of the resident-table (100%) buffer. This buffer holds entries from tables which have been defined in the Data Dictionary with buffer type X (buffer all entries that belong to active clients) and G (buffer entries according to a generic key specification).
- `zcsa/db_max_buftab`: Specifies the number of directory entries in the resident-table buffer. This is the maximum number of tables that can be buffered. Set the parameter to a large enough value to ensure that there are always free directory entries.

You can monitor the buffer efficiency with transaction ST02 (setup and tune buffers). Also, a SQL trace created with transaction ST05 (SQL Performance Analysis) must not show any access to a buffered table. If you find an access to a table you have marked as buffered, you must analyze, why the table is not loaded into the buffer. It is important to know that the following accesses always bypass the buffer and proceed directly to the database:

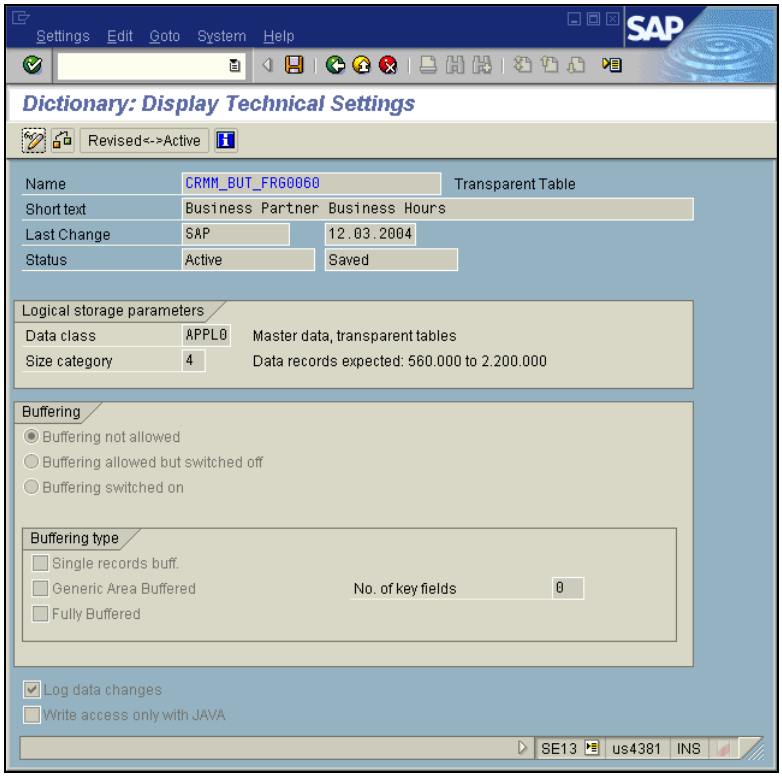
- SELECT... BYPASSING BUFFER
- SELECT FOR UPDATE
- SELECT with aggregate function, for example COUNT, MIN, MAX, SUM, AVG
- SELECT DISTINCT
- SELECT... WHERE… IS NULL
- ORDER BY (with the exception of PRIMARY KEY)

Figure 6-18 shows, how to buffer a table fully in memory.

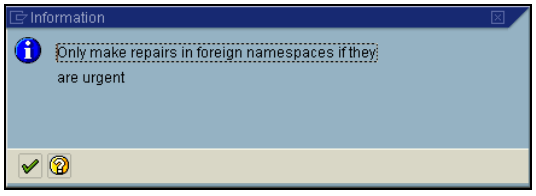
1. Call transaction SE11 (ABAP Dictionary). Choose ***Database table*** and enter the name of the table.



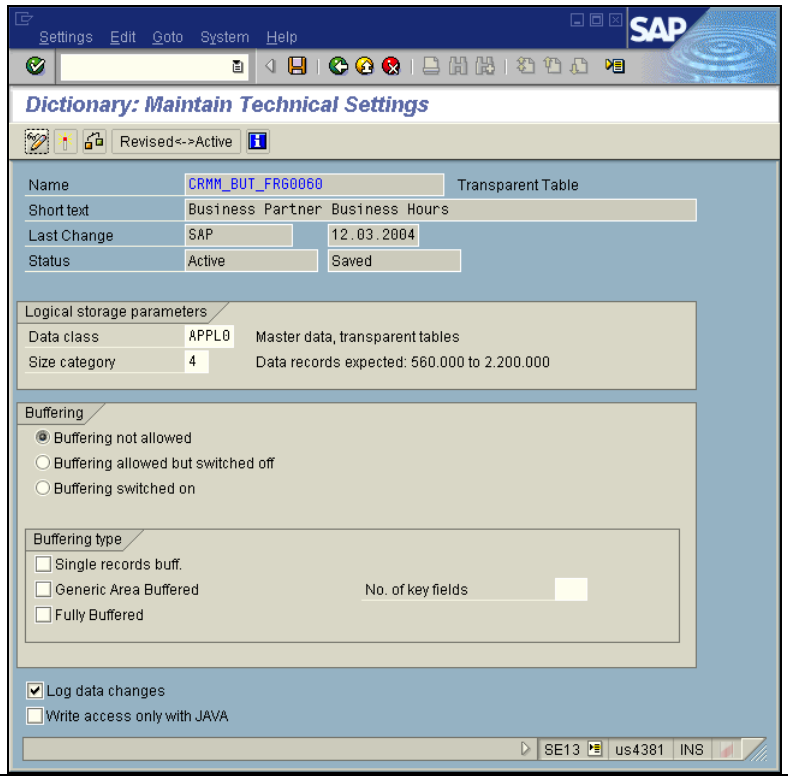
2. Push the ***Technical Settings*** button.



3. Push the ***Change*** button.



4. Push the ***Continue*** button to confirm.



5. Choose ***Buffering switched*** on and ***Fully Buffered***. The push the ***Activate*** button.

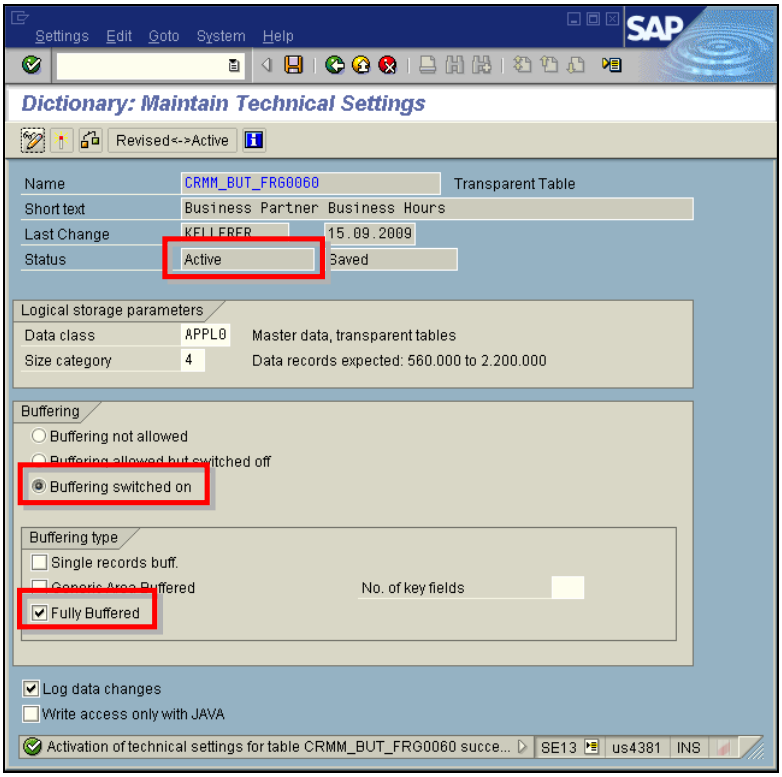


Figure 6-18: Buffering a Table in Memory

6.13.1 Business Partner (BUPA_MAIN)

During the initial download of business partners, you can consider to buffer the following tables fully in memory.

Table Name	Description
ADRCITY	Postal cities
ADRCITYT	Postal districts (long texts)
ADRSTREET	Streets
ADRSTREETT	Streets (long texts)
ADRSTRPCD	Street (section)/postal code: Assignment
BNKA	Bank master record
BNKAIN	Bank Master (Additional Fields for BNKA)
TIBAN	IBAN
BKK21	Business Partner Aliases
BP001	FS-Specific Attributes; Organization
BP1010	BP: Credit standing data
BP1030	BP: Business partner - reporting data
BUT_FRG0010	Business Partner: Tax Classification
CRMM_BUT_FRG0060	Business Partner Business Hours
CRMM_BUT_FRG0061	Business Partner Business Hours Interval
CRMM_BUT_FRG0070	Business Partner Excluded Partner Functions List
CRMM_BUT_FRG0071	Business Partner Excluded Partner Functions
CRMM_BUT_LNK0011	Business Partner - Sales Rule
CRMM_BUT_LNK0021	Business Partner - Shipping Rule
CRMM_BUT_LNK0031	Business Partner - Billing Rule
CRMM_BUT_CLASS	CRM Business Partner Classification
TAX_TXJCD_FLAG	Logic Control for Jurisdiction Code Determination
HRP1000	HRP1000 DB Table
HRP1001	Infotype 1001 DB Table
ROOSPRMSC	Control Parameter Per DataSource Channel
ROOSPRMSF	Control Parameters Per DataSource
ROOSOURCE	Table Header for SAP BW OLTP Sources

Figure 6-19 Tables that can be buffered in Memory (BUPA_MAIN)

You should switch off buffering for the following tables due to possible change accesses:

Table Name	Description
SPERS_OBJ	Personalization object repository

Figure 6-20 Tables that must not be buffered in Memory (BUPA_MAIN)

6.13.2 Connection Object (SI_CONNOBJ)

During the initial download of connection objects, you can consider to buffer the following tables fully in memory.

Table Name	Description
ADRCITY	Postal cities
ADRCITYT	Postal districts (long texts)
ADRSTREET	Streets
ADRSTREETT	Streets (long texts)
ADRSTRPCD	Street (section)/postal code: Assignment
IBIBT	IB: Installed base short texts
IBINOWN	IB: Owner of an IBase Component
IBSTREF	IB: Installed Base - Reference to Master Data
COMM_CATEGORY	Category
COMM_CATEGORYT	Category Description
COMM_CATFRGR	Category Set-Type Assignment
COMM_CLEAR_TARG	Data Cleansing: Target Object List
COMM_HIERARCHY	Hierarchy
COMM_HIERARCHYT	Hierarchy Description
COMM_PCAT_ITM	Product Catalog Item
COMM_PME_CAT_MAP	Mapping of a Category or Product to PME Classes
COMM_PRCAT	Product Category
COMM_PRCATFRGR	Assignment Product Category - Set Type
COMM_PRSHTEXT	Product Description
COMM_PR_UNIT	Product Unit of Measurement
TAX_TXJCD_FLAG	Logic Control for Jurisdiction Code Determination

Figure 6-21 Tables that can be buffered in Memory (SI_CONNOBJ)

You should switch off buffering for the following tables due to possible change accesses:

Table Name	Description
IBINTX	IBase: Text for the Instance

Figure 6-22 Tables that must not be buffered in Memory (BUPA_MAIN)

You may choose to buffer also the COMC_ATTRIBUTE in memory. In order to take effect, you will also need to modify the COM_ATTRIBUTE_TAB_READ_SINGLE function module.

```
FUNCTION com_attribute_tab_read_single.
[…]
    ELSE.

      SELECT SINGLE * FROM comc_attribute INTO wa_attribute
*<==== start of deletion
*           BYPASSING BUFFER
*<==== end of deletion
          WHERE attr_id = iv_attribute_id.                "#EC *
```



```
      IF sy-subrc <> 0.
[…]

ENDFUNCTION.
```

Figure 6-23 Modification to take advantage of Buffering of COMC_ATTRIBUTE Table

6.13.3 Point of Delivery (SI_POD)

During the initial download of connection objects, you can consider to buffer the same tables as suggested for connection objects (SI_CONNOBJ).

6.13.4 Contract (SI_CONTRACT)

During the initial download of contracts, you can consider to buffer the following tables fully in memory.

Table Name	Description
SCAPPT	User-Specific Appointments
SCRULES	Periodic appointments: Rule
SCGENAPPT	generated appointments
TIMECUEV	Symbolic Name for Activity ('Event')

Figure 6-24 Tables that can be buffered in Memory (SI_CONTRACT)

6.14 Buffering Number Range Objects in Memory

A number range object contains all the definitions required for assigning unique numbers to a business object. One number range object corresponds to one business object, but can include one or more number ranges. A number range contains a number range interval with a defined character set. The number range interval consists of numeric or alphanumeric characters (only for external number ranges) and is delimited by the fields ***From number*** and ***To number***.

During the initial download, the processes in the CRM system request new numbers from various number range objects if not the identification number of the respective business object in the ERP system can be used. During the determination of a new number a business lock on the relevant number range object respectively on the relevant number range interval is created thus preventing other processes to access and update the same number range object at the same time. The application releases the lock on the number range object with a `COMMIT WORK` or a `ROLLBACK WORK`. Until them, the processes have to wait for the release of the lock leading to a serialized instead of concurrent creation of business objects in the CRM system.

A solution to this problem is to buffer free numbers in the memory of the application servers and serving the requests for new numbers from memory. When the buffer is exhausted, the application server loads a new set of numbers into its memory. This process reduces the probability of blocking situations significantly because the number range interval is blocked once per block access instead of once for each new number. The used mechanism to fetch new numbers used is called ***number range server***. For technical reasons a second work process (`WP2`) is required for this. This usually runs on the same server as the process that wants to assign a number (`WP1`). Figure 6-25 shows the process flow and the components involved.

Buffered numbered are lost after a shutdown of an instance or of an application server. This may lead to large gaps in the sequence of assigned numbers to business objects. You can find more information about the various buffering options in SAP Note 504875, whereas SAP Note 920234 describes the mechanism to fill the number range buffer and possible problems particularly due to a lack of dialog work process to load new buffers into memory.

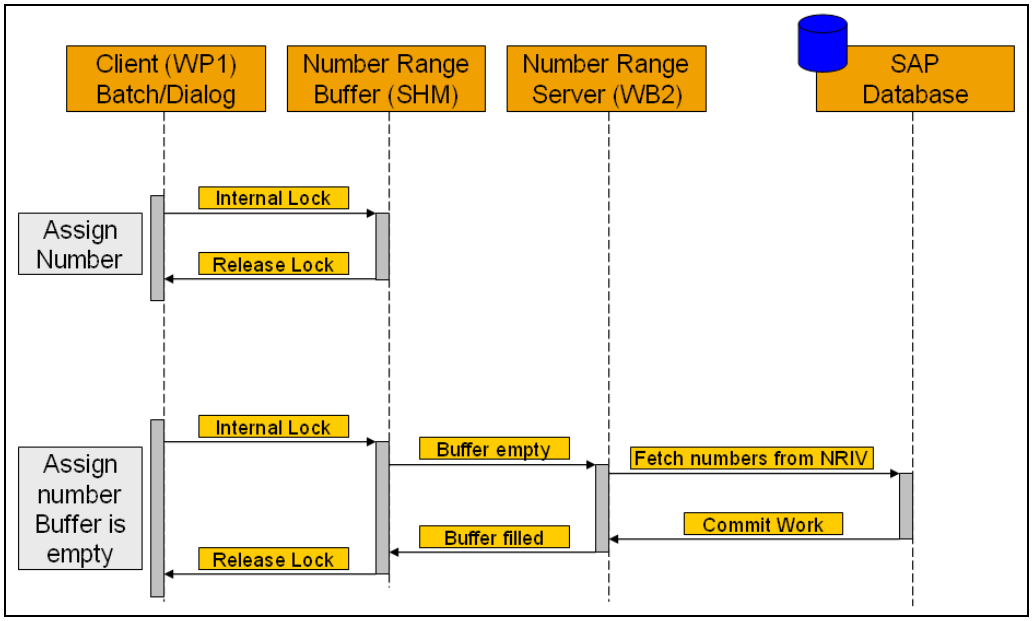


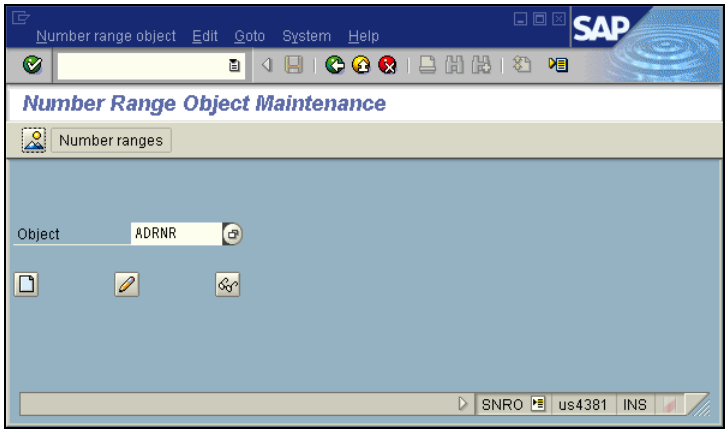
Figure 6-25 Process flow of the Number Range Server

We recommend to buffer the number range objects relevant during the initial download with the option ***Main Memory Buffering***. The disadvantage of this option is that gaps between numbers may exist if the CRM system consists of more than one application server that these gaps cannot be documented and numbers are not assigned chronologically. You may only set the main memory buffering if no legal requirement applies to the number assignment. You can display the buffer contents of the main memory using transaction SM56 (Display Number Range Buffer).

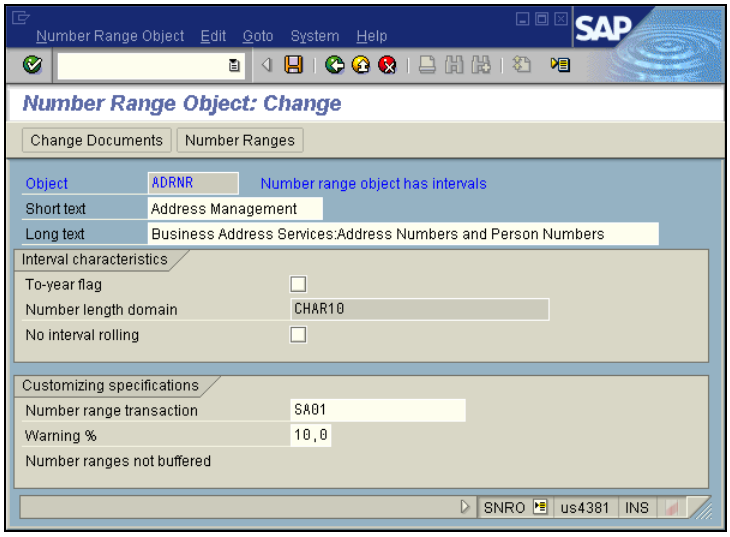
We recommend to buffer enough numbers in memory to allow the creation of about 500 to 1000 business objects per 5 seconds per application server in order to reduce the accesses to a number range interval to a reasonable level.

Figure 6-26 shows, how to setup buffering of a number range object in memory.

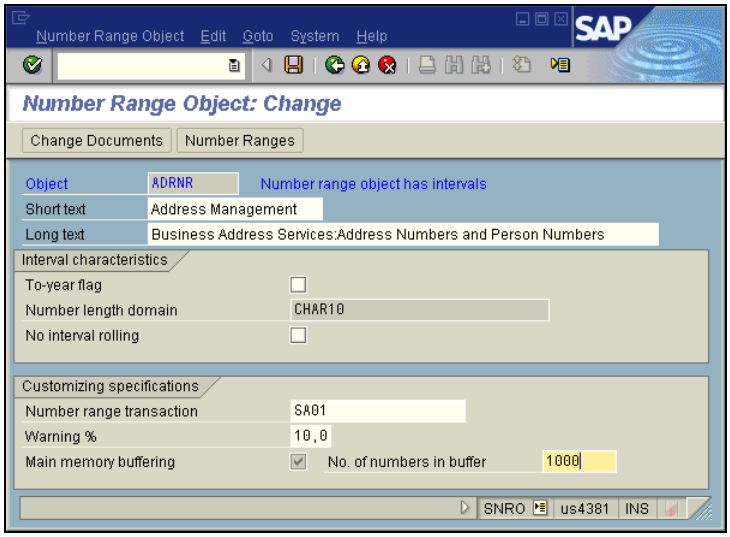
1. Call transaction SNRO (Number Range Object Maintenance) and enter the name of the number range object in the ***Object*** field.



2. Push the ***Change*** button.



3. Choose *Edit → Setup Buffering → Main Memory.* Enter an appropriate number of numbers in the ***No. of numbers in buffer*** field.



4. Push the ***Save*** button.

Figure 6-26: Setup Buffering of a Number range Object

Figure 6-27 shows the number range objects the application uses in the CRM system during the initial download and you should buffer in memory.

Adapter Object	Number Range Object	Description
BUPA_MAIN SI_CONNOBJ SI_POD	ADRNR	Address Numbers
SI_CONNOBJ SI_POD	CU_INOB	Internal object numbers
SI_CONTRACT	APPL_LOG	Application log
SI_CONTRACT	CRM_SALES	CRM Sales Transaction
BUPA_MAIN BUAG_MAIN	AENDBELEG	Change Document

SI_CONNOBJ SI_POD SI_CONTRACT	AENDBELEG	Change Document

Figure 6-27 Number Range Objects that can be buffered in Memory

6.15 Deactivation of Geocoding

The respective geographical longitude and latitude degrees are determined for the address data during geo-coding. You can this data, for example, to determine the distance between two business partners. By default, the geocoding is active even the service is not used in any business scenario.

If you do not use geocoding, you can globally deactivate the geocoding and the assigned geolocation persistence service. As a result, fewer database accesses are carried out thus reducing the load in the database. You can deactivate the geocoding and the assign service with the `GEOCODING_SET_GEOFLAG` report.

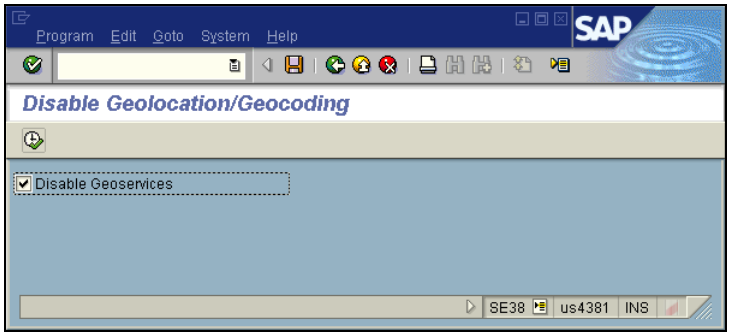


Figure 6-28 Deactivation of Geocoding

6.16 Database Performance

Analysis and improvement of a database performance is a complex subject. This chapter does not claim to provide the solution for all performance problems that may occur during an initial download. Instead, it offers a general overview and references to more detailed notes and documentation that could help to solve performance problems on the database. It is important that a database administrator accompanies the test cycles and analyses the database performance using the transaction ST04 (DB Performance Monitor) but also using database specific monitoring tool, such as, the Oracle Tuning Adviser. Apart analyzing the performance of database accesses, such as, SELECTS, INSERTS and UPDATES, you must pay special attention to congestions and bottlenecks on the database caused by many concurrent processes hitting and updating the same few tables at the same time.

You can use transactions ST03 and ST03N (Workload Monitors) to determine the extent to which accesses to the Oracle database are involved in the overall response time. You can determine this portion by using the **Database time** column. A high value indicates that database performance tuning will most likely improve the overall response time.

Alternatively, you may choose to start a first performance analysis on a very high level with the transaction SM50 (Process Overview). When a process remains for a longer time in the action SEQUENTIAL READ, this might give you already a first indication that a database problem exists and need to be resolved. In addition, extended runtimes identified in a SQL trace (transaction ST05) demands a further analysis.

The SEQUENTIAL READ action indicates a read access is executed in the database that may return more than one record. It is unimportant which accesses are used in the database and whether the main processing time is lost on the Oracle server (or, for example, in the network or the database interface). For more information, see SAP Note 618868.

However, only with an in-depth analysis you can identify the root of the performance problem, propose and implement options to resolve it. In a subsequent test, you must verify, whether the implemented change resolved the problem.

Do not change more than one parameter at a time. Otherwise, it will be impossible to evaluate the performance improvement caused by a specific parameter change.

6.16.1 Unloading the Database Server

You can configure easily an initial download with a high parallelization of the processing of inbound queues. It is even more easy, to increase the parallelization in adding more inbound queues (for more information, see chapter 6.5.1 *Request with multiple Inbound Queues*). As more processes access the database as more database resources are utilized to serve the concurrent requests. With an increasing number of concurrent processes, the database resources are more and more exhausted. Inevitably, the database server becomes the limiting factor for performance of the initial download.

Instead of executing inbound queues on the database server, you should process them only on the available application servers (for more information, see chapter 6.5.3 *Distribution of RFCs in the CRM System*) allowing the database to utilize all available resources to serve the incoming queries. Of course, this advice is only valid, if there is no resource sharing between the database server and the application servers. You should process inbound queues on the central instance (usually the central instance shares the system resources with the database server) only, if all available application servers are fully utilized and the database server still has resources left.

For the period of the initial download, you should consider to allocate temporarily additional system resources for as application servers to the CRM system in order to have enough system resources available to process the queues only on the application

servers thus making the system resources of the database server exclusively available just for database operations.

6.16.2 Refresh of Database Statistics

A table in the database of a SAP system is defined with a unique primary key and, if necessary, with additional secondary indexes. Just the existence of an index is not sufficient to allow the cost-based optimizer of the database to determine the best access path (execution plan, stored procedure etc.) to find and retrieve the requested rows from the tables most efficiently. In addition, the database must know how selective an index is. For this, the database gathers statistics in reading the rows and determining how many different values for each of the index fields exist. This information may change when the system is in production. Therefore, the periodic update (refresh) of the database statistics is among of the most important activities of the database housekeeping. Without updated statistics, the database either uses a wrong index or reads all rows of the table (full table scan). Obviously, the response time of the database will increase significantly with the number of rows in the table, if the database uses a wrong execution plan.

When you start an initial download into an already production system, the database statistics will be up to date. However, when you start an initial download into an empty system, a preceding analysis of the database statistics is not helpful because statistics are based on the information of existing rows in the table. Therefore, you will recognize that after some few minutes the response time for some tables will increase. This is because rows have been inserted in that table as a first result of the initial download and queries for the same table are not executed with an appropriate execution plan, for example, with a full table scan. It is important, that you monitor the database closely, for example, with the transaction `ST05` (SQL Trace) and manually refresh the statistics of the relevant tables if required. We recommended to create a check list to remember for which tables you must refresh the statistics. With such a check list, you can refresh the statistics shortly after you have started the initial download of one specific business object before even monitoring a deteriorating database performance. At the time, only comparably few rows exist in the table and the refresh finish quickly without consuming many database resources.

The age of the database statistics does not affect the performance. It is only important that statistics reflect correctly the current dataset in the table and that the cost-based optimizer always chooses the best access path based on the existing statistics. It is not a problem if tables with static contents contain statistics that are several years old. For more information, see SAP Note 588668.

You might monitor that the cost-based optimizer chooses a non-optimal access path even after an analysis of the statistics. This can happen, when a field that is part of the optimal index contains the same value for all rows thus making the field unselective. Even this special situation will change once the CRM system is in production, you cannot solve this problem just with an analysis of the statistics during the initial download. A solution could be, to add temporarily 10 – 20 rows with varying values in the particular field to the table using an unused client number for the `CLIENT` or `MANDT` field, for example client 999) and analyze the statistics again. Due to the added rows, the index becomes more selective and the cost-based optimizer chooses now the correct index. You should document that you added rows in a separate client. Of course, you should delete the rows once they are not needed anymore.

During the initial download, we therefore recommend to deactivate periodic jobs that refresh database statistic automatically. You should also deactivate a refresh triggered automatically by the database after a certain number of changes in a table. Often, the refresh of the statistics happens quite late leaving the performance on a low level for a long period affecting directly the duration of the initial download. You should refresh of the statistics only manually. This will also allow you to link a temporary drop of the performance to an ongoing analysis of the statistics. This is not possible when the refresh is triggered unattended in background.

6.16.3 Removing Unused Indexes

Removing unnecessary indexes can improve performance by the overhead during the mass updates during the initial download. In this way, you can also resolve contentions on an index in just removing the root of the contention: the index itself. Before removing an index, you must monitor its usage during the initial download in order to avoid performance problems of queries that require that specific index. You must rebuild the removed index after the initial download. Refer to your database manual, how to monitor the usage of indexes.

6.16.4 Optimization of Redo Logs (Oracle)

Before Oracle changes data in a data file it writes these changes to online redo logs. If the database crashes, you can restore the database from the last backup and then redo all changes as recorded in the online redo logs since the backup.

Instead of writing to the online redo logs immediately, Oracle records changes first in the memory (redo buffer) in order to reduce disk IO. The Oracle `LGWR` (log writer) flushes the changes of completed transactions from the redo buffer to the online redo logs.

Online redo log file are organized in groups (original online redo log and mirrored online red log) of which at least two must exist. The Oracle `LGWR` writes redo log entries to one group at a time. When this group is full, a log file switch occurs, making another group the current group. A log switch causes a checkpoint signaling the `DBWR` (database writer) to update the data files. The `DBWR` writes all modified database buffers from the redo buffer of the system global area to the data files.

If Oracle runs in archive log mode, the `ARCn` (archiver) Oracle archives the filled redo logs in order to allow a redo of all changes based on the redo log entries in the redo logs since the last backup. Of course, Oracle does not allow the overwriting of a redo log unless it is archived.

The performance of all online redo log operations has a huge impact on the overall database performance specifically during the initial download when a high number of database updates occur in a short period. Therefore, it is imperative that you monitor the redo log activities closely. You must pay special attention to

- Size of online redo logs to control the frequency of log switches and minimize system waits
- Optimize the online redo log disk to prevent bottlenecks. In high-update databases, no amount of disk tuning may relieve redo log bottlenecks, because Oracle must push all updates, for all disks, into a single redo location. To increase data security and also performance, the online redo log files must be mirrored and the original online and mirrored online redo log files must be located on different disks. Verify that your Oracle online redo log files are mirrored either at the database level or at the hardware level (RAID1).
- Determine the optimal size of the buffers related to redo logs, for example, the `log_buffer` parameter

As described above, you can completely recover a database operating in archive log from both instance and media failure using the (archived) redo log files. However, additional administrative operations are required to maintain the archived online redo logs. Operating a database in no-archive log mode has the advantage of not having to read the red log files and write them to their archive location. This increases the performance of the database slightly when the database is under high load due to the high volume of updates as during the initial download. The obvious cost of this is that you will only have limited options when recovering your database.

SAP does not support the no-archive log mode during normal production operation. If the database operates in no-archive log mode, the database can only be completely

recovered from instance failure but not from media failure. This means that, in case of media failure you will lose all work done after the last valid offline backup. The database can be backed up only while it is completely closed. However, if you choose to deactivate the archiving of redo log files during the initial download, you must create a backup of the database on completion of the initial download and reactivate the archive log mode.

6.16.5 Optimization of Rollback Segments (Oracle)

The Oracle database uses rollback segments to store necessary data to reverse, or undo, changes made by transactions. In dependency of the number of business per BDoc, each inbound queues can create big database transactions in updating many tables during the processing of a BDoc the database must be manage. This leads to special requirements regarding the sizing of the rollback segments. Ideally, the initial size of the rollback segments should be close to the optimal size. Otherwise, Oracle tries permanently to shrink the rollback segments affecting the overall database performance.

To identify contention for rollback segments first find out the number of times that processes had to wait for the rollback segment header and blocks. The `V$WAITSTAT` view contains this information. Compare the number of waits for any class with the number of logical reads during the same period (`V$SYSSTAT` view) If the number of waits for any class of waits is greater than 1% of the total number of logical reads then you should add more rollback segments.

Rollback segments should be isolated as much as possible by placing them in their own table space, preferably on a separate disk from other active table spaces. Oracle uses the `OPTIMAL` parameter to shrink back rollback segments to an optimal size after they have dynamically extended. The `V$ROLLSTAT` table can help in determining proper sizing of rollback segments:

6.16.6 Data Block Contention

During an initial download, the application updates some few database tables per download object massively, for example, the IBIN table during the initial download of connection objects. Due to the high level or parallelization of the initial download, “hot” data blocks can cause **block contention** that could lead to a serialization of the update operations thus affecting the database performance considerably. Therefore, you need to prepare the database for such a mass update. Refer to your database manual, how to monitor and resolve data block contentions. You may also use the transaction `ST04`, to monitor the buffer wait times in the database.

If you monitor contentions on table indexes due to massive updates, you may consider to **reverse indexes**. After a reversal, parallel changes to table entries with similar key values (for example, document numbers 1234, 1235, and 1236) are distributed to different blocks since the reverse document numbers (4321, 5321, 6321 in the above example) are now very different. As a result, waits due to locking situations on block level are less likely to occur (for more information, see SAP Note 915242). If key values that generally have even increments are inserted into an index, fragmentation effects (for more information, see SAP Note 771929) may occur. You can also avoid this effect by reversing the column values, because this causes the new entries to be distributed across all index leaf blocks.

Partitioning is the process of separating data into logical subsets. You cannot only partition tables but also indexes in all possible combinations. The basic advantage of partitioning is that many database change operations can be carried out on separate partitions (for more information, see SAP Note 722188) which can improve the performance significantly. On the other hand, you need to understand fully the feature before you use it and partition the table and indexes on the correct column for your application. Otherwise, you will not get the benefits of partitioning. Any performance gain will depend on how you choose the partition key. This can become very difficult since many business objects in the CRM system are

identified by a GUID (Global Unique Identifier) that is calculated using an algorithm defined by the Open Software Foundation and does not lead to consecutive numbers. You must assume that the expected performance gain is usually not worth the effort implementing and maintaining the partitions just because of the once-only activity of an initial download. Other database tuning options are usually more promising with a reduced effort to implement them.

6.16.7 Space Management

During tests with a high data volume, very often problems occur due to full table spaces on the database. This is because test systems have usually only limited disk space available compared to the disk space of the later production system. The limited disk space is not only counterproductive for a completion of a test cycle within a predefined period because full table space lead to an interruption of the tests due to a system downtime until more table space has been released to your test system. Moreover, limited disk space can lead to hot spots (disks with high I/O activities compared to the remaining disks of your storage subsystem).

You should have received an estimate of the initial and maximum size of the production database from your hardware partner, and sized the planned production database according to these values. We recommend that you set up the available disk space for the database of your test environment to the size expected after one year of operation or at least to the size that allows a completion of the initial load.

6.16.8 Optimization of Storage Subsystem

A good performance of the storage subsystem is crucial for a good database performance. If you want to improve the performance of your existing storage subsystem, see ***SAP Service Marketplace*** at http://service.sap.com/stso for more information about ***SAP Storage Subsystem Optimization***.

6.16.9 Maintenance of CRM Middleware Tables

The tables for the business document (BDoc) message flow and the CRM Middleware trace can increase considerably and therefore require a lot of disk space. Of course, this may also cause a decrease in performance during the processing of BDoc messages. Figure 6-29 shows the BDoc message related tables. The CRM Middleware trace is saved in the SMWT_TRC table.

Table	Description
SMW3_BDOC	BDoc Message Store: Header
SMW3_BDOC1	BDoc Message Store: Message Body (classic part)
SMW3_BDOC2	BDoc Message Store: Message Body (extension part)
SMW3_BDOC4	BDoc: Validation Errors
SMW3_BDOC5	BDoc: Receivers with state
SMW3_BDOC6	BDoc: Receiver specific error segments
SMW3_BDOC7	BDoc: Receiver Root IDs
SMW3_BDOCQ	qRFC Queues in use by BDoc Message

Figure 6-29 Tables managing BDoc Messages

The reorganization of the CRM Middleware relevant tables can be performed by a daily background job that must run in all active clients. The scheduled job executes the SMO6_REORG2 program (for more information, see SAP Note 713173). This program carries out the reorganization of all relevant data of the CRM Middleware, such as:

- Reorganization of processed BDoc type messages

- Reorganization of trace data
- Reorganization of KeyGen data
- Reorganization of BDoc statistics
- Reorganization of Comm station statistics
- Reorganization of temporary job-data of the Administration Console
- Reorganization of scheduling data
- Reorganization of deleted sites, subscriptions, organizations
- Reorganization of the logs of the Subscription Assistant
- Reorganization of the temporary data for the realignment of inactive sites
- Reorganization of one-time requests
- Reorganization of the technical links in BDoc messages

You can schedule the background jobs with transaction SMWP (CRM Middleware Monitoring Cockpit).

Object links between mBdoc messages and mBdoc/sBdoc messages are held for 21 days in the standard job variant. For more information about objects links in mBDocs, see chapter 6.11 *Object Links in* BDocs (CRM System). If a business process finishes successfully, object link entries are no longer required. In our experience, a value of seven days is usually sufficient. If this is the case, set ***Reorganize for object links (BDocs) days to hold*** to seven days in your variant.

6.16.10 Maintenance RFC Tables

In a CRM system in transaction ST04 (Monitor Performance Database) *Reads / User calls* is too high (typically > 20) and in specific SQL statements the *Bgets/exec* and *Bgets/row* values are high although the correct index is used for a statement. Particularly during an initial download, the inbound and outbound queues grow and shrink very dynamically and the related tables in the CRM system increase and decrease their size dynamically. As a result, the indices associated with these tables degenerate quickly and have to be rebuilt regularly.

You need to check the storage quality of the identified index (*number of blocks* divided by *number of blocks used* when using the DB02 database). If it is less than 40%, you should perform an index rebuild. If you use an Oracle database, you may choose the COALECSE option as described in SAP Note 332677.

Figure 6-30 shows a list of tables that are in our experience particularly vulnerable to index degeneration. You should rebuild these tables on a regular basis, for medium sized installations at least once a week, for large installations daily.

Table	Description
SMW3_BDOC*	BDoc Message
SMWT_TRC	CRM Middleware trace
ARFCSSTATE	Description of ARFC Call Status (Send)
ARFCSDATA	ARFC Call Data (Callers)
ARFCRSTATE	Status of ARFC Calls on Receiver Side
QREFTID	Reference TIDs of Receiver List for qRFC
TRFCQOUT	tRFC Queue Description (Outbound Queue)
TRFCQIN	tRFC Queue Description (Inbound Queue)
TRFCQSTATE	Description of qRFC Call Conditions (Inbound Queue)
TRFCQDATA	qRFC Call Data (Inbound Queue)

SMOHJOBQ	Replication Site Queue
SMOHMSGQ	Replication Message Queue
SMOHSITEQ	Replication Site Queue
SMOFCDBHD	Header Table for comparable records coming from CDB
SMW0REL	Links between BDoc and application object
SRRELROLES	Object Relationship Service: Roles

Figure 6-30 Tables changing their size dynamically

In certain situations, the Oracle cost based optimizer (CBO) may take unfavorable decisions, although there are no CBO errors, for example, for tables in Figure 6-30. One reason for the calculation of a suboptimal access path can be if you choose an unfavorable time for analyzing statistics of these tables. Alternatively, you may choose to implement a script for the automatic postprocessing of critical statistic values using DBMS_STATS functions (for more information, see SAP Note 724545). This script adjusts individual statistic values, excludes the relevant tables from the statistics creation using the BRCONNECT tool (ACTIV = I in DBSTATC) and locks the statistics at Oracle level. The script is designed for systems with Oracle 10g or higher, but you can use it also for Oracle 9i or lower. For more information, see SAP Notes 1020260 and 756335.

6.17 Deactivation of BAdIs

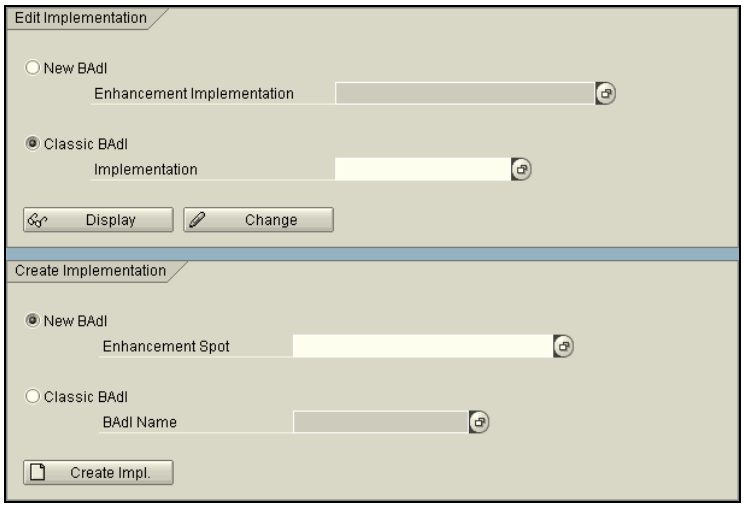
A BAdI (Business Add-In) is an object-oriented enhancement option, which makes it the most sophisticated enhancement type. The main characteristic of a BAdI is that it provides a mechanism to change the functionality of a well-defined business function (e.g. a BAPI) without making changes to the delivered source code.

There are BAdI implementations in the SAP standard delivery whose execution is not required in most business scenarios during an initial download. You may choose to deactivate such BAdI implementations temporarily to improve the performance of the initial download.

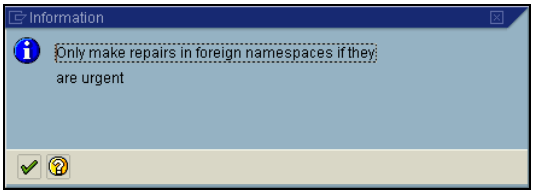
You should check with the CRM project team prior to a deactivation, whether you can deactivate a specific BAdI implementation. There might be project specific BAdI implementations that must remain active also during an initial download while you can deactivate other project specific BAdI implementations.

Figure 6-31 shows, how to deactivate a BAdI implementation.

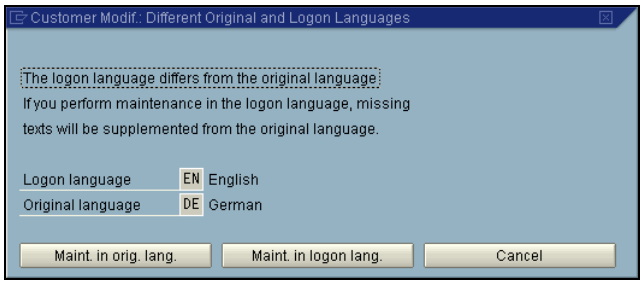
1. Call transaction SE19 (BAdI Builder).



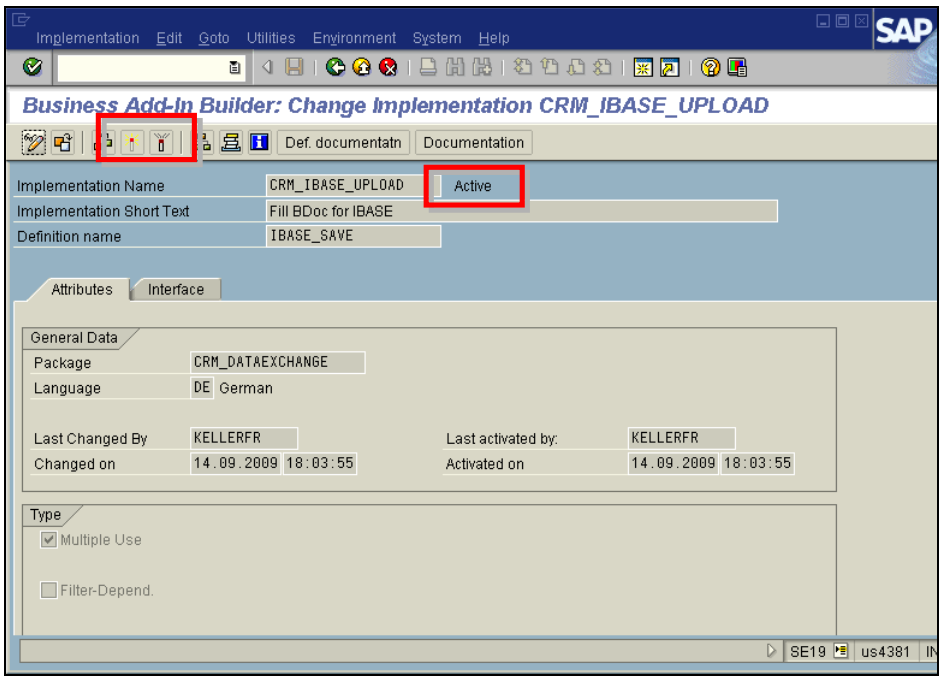
2. On the ***Edit Implementation*** sub screen choose either ***New BaAdI*** or ***Classic BAdI*** and enter the name of the BAdI implementation you want to deactivate.
3. Push the ***Change*** button.



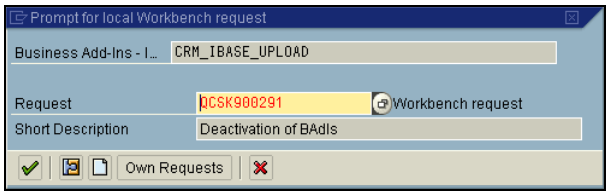
4. Push the ***Continue*** button



5. Push the ***Maintenance in original language*** button.



6. Push the ***Deactivate*** button.
7. Choose your transport request to save your change



8. Push the ***Continue*** button.

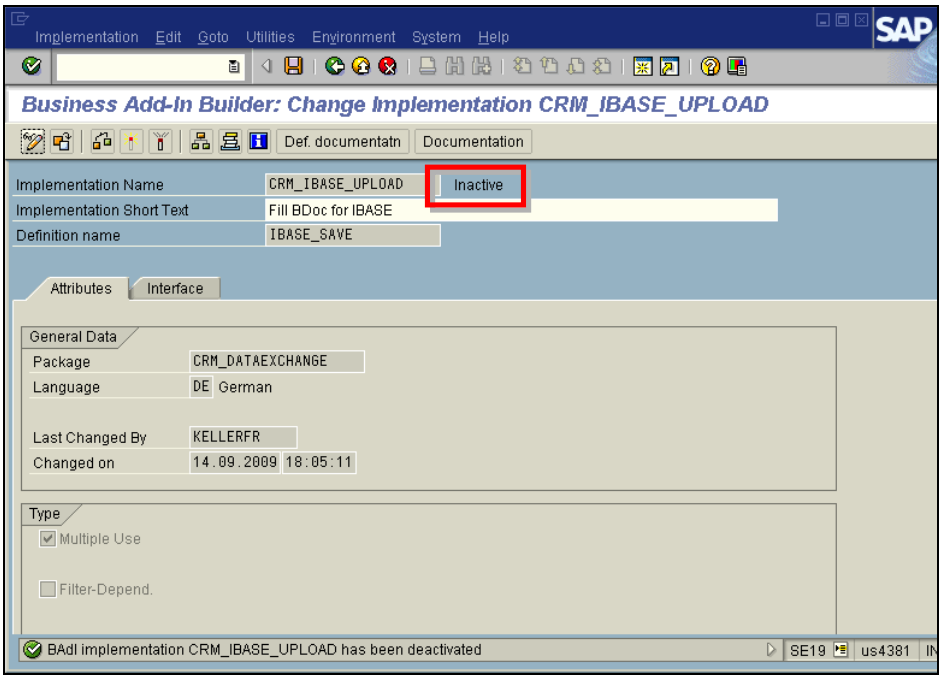


Figure 6-31: Deactivation of a BAdI Implementation

6.17.1 Business Partner (BUPA_MAIN)

Figure 6-32 shows a list of BAdI implementations you may choose to deactivate during an initial download of business partners (adapter object BUPA_MAIN):

BAdI	Implementation	Description
PARTNER_UPDATE	BUPA_OUTBOUND_CREATE	BP Outbound Processing
BUPA_GENERAL_EXPORT	FSBP_GENERAL_EXPORT	Export of General Data
	FSBP_PHONETIC_EXPORT	Update Phonetic Search Term
BUPA_ADDR_EXPORT	FSBP_ADDR_EXPORT	Export of Address References (BUT020) and Usages
BUPA_ADRREL_EXPORT	FSBP_ADRREL_EXPORT	Export of Address References for Relationships
BUPA_BANK_EXPORT	FSBP_BANK_EXPORT	Export of Checked Bank Data
BUPA_CCARD_EXPORT	FSBP_CCARD_EXPORT	Export of Payment Card Data
BUPA_ROLE_EXPORT	FSBP_ROLE_EXPORT	Export of Role Data
BUPA_CONTPREL_EXPORT	FSBP_CONTPREL_EXPORT	Export of Detailed Data for Contact Person Rel.
BUPA_INDSEC_EXPORT	FSBP_INDSEC_EXPORT	Export ofIndustry Data
BUPA_INTCOMP_EXPORT	FSBP_INTCOMP_EXPORT	Export of Detailed Data for Rel.Type Investment in Firm
BUPA_RELSHP_EXPORT	FSBP_RELSHP_EXPORT	Export of Management Table of the BP Relationships

Figure 6-32 BAdI Implementations for Business Partner for a Deactivation

Activate the deactivated BAdI implementation on completion of the initial download.

6.17.2 Connection Object (SI_CONNOBJ)

You may choose to deactivate the following BAdI implementation during an initial download of connection objects with adapter object SI_CONNOBJ:

- BAdI CRM_IBASE_UPLOAD (Fill BDoc for IBASE)

A deactivation is only possible if it is planned to perform an initial download to the CDB (mobile application) after the initial download from the ERP into the CRM system (for more information, see chapter 6.7 *Deactivation of Message Flow Processing*.

6.17.3 Point of Delivery (SI_POD)

You may choose to deactivate the following BAdI implementation during an initial download of point of deliveries with adapter object SI_POD:

- BAdI CRM_IBASE_UPLOAD (Fill BDoc for IBASE)

A deactivation is only possible if it is planned to perform an initial download to the CDB (mobile application) after the initial download from the ERP into the CRM system (for more information, see chapter 6.7 *Deactivation of Message Flow Processing*.

6.18 Deactivation of Change Documents

Many business objects are changed frequently during normal business operations. It is often useful and even necessary to be able to trace the changes made. This logging is carried out with change documents. Changes documents can be created for changes already made as well as for planned changes. To be able to log changes to a business object in a change document, an appropriate change document object is defined in the SAP system.

The application saves change documents asynchronously with only a very brief delay by special updating (UPD) work processes. This permits an improved performance for dialog transactions because the user can continue his work while the update of the database takes place in parallel.

The updating function causes additional database activities because the application passes the necessary update information through the VBHDR, VBDATA and VBMOD database tables to the updating work process. In addition, the creation of change documents in the CDHDR and CDCLS tables causes an additional database load thus reducing the available system resources for the initial download.

6.18.1 Business Partner (BUPA_MAIN)

During the replication of business partner, the application creates various change documents, for example, the BUPA_BUP, BUPA_ADR and ADRESSE2 change documents. You can suppress the creation of these change documents in implementing temporarily a function module. Figure 6-33 shows an implementation of such a function module you can use as a basis for an own implementation. This function module deactivates not only the creation of change documents but also status checks to improve the performance. You may choose to change the sample source code accordingly if you do not want to deactivate both. For more information, see SAP Note 493343.

```
FUNCTION Z_BUPA_NO_CHANGEDOC.
*"----------------------------------------------------------------------
*"*"Local Interface:
*"  IMPORTING
*"     REFERENCE(IV_X_SAVE) TYPE  CHAR1
*"  EXPORTING
*"     REFERENCE(STATUS) TYPE  CHAR1
*"  TABLES
*"      ET_RETURN STRUCTURE  BAPIRET2
*"  CHANGING
*"     REFERENCE(C_BP_STRUCT) TYPE  BUS_EI_EXTERN
*"----------------------------------------------------------------------
```

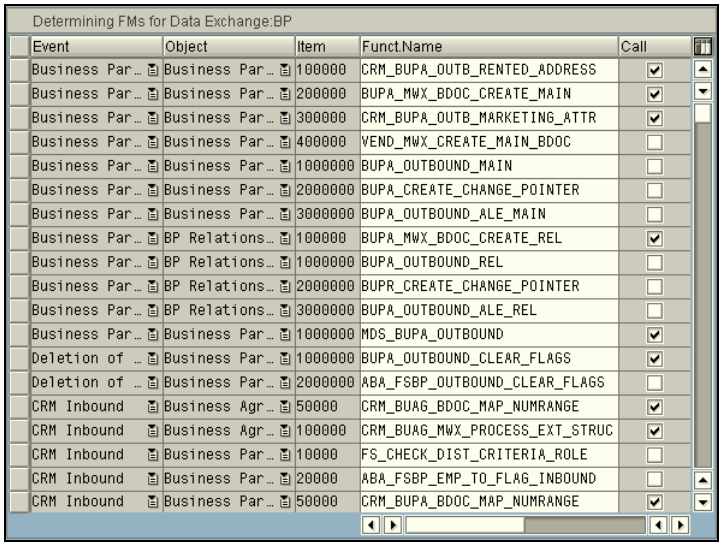


```
* suppress change documents
  CALL FUNCTION 'BUPA_MWX_BDOC_NO_CDS'.

* suppress status check
  CALL FUNCTION 'BUPA_STATUS_SUPPRESS_CHECK'
    EXPORTING
      iv_suppress       = 'X'.

  status = 'S'.

ENDFUNCTION.
```

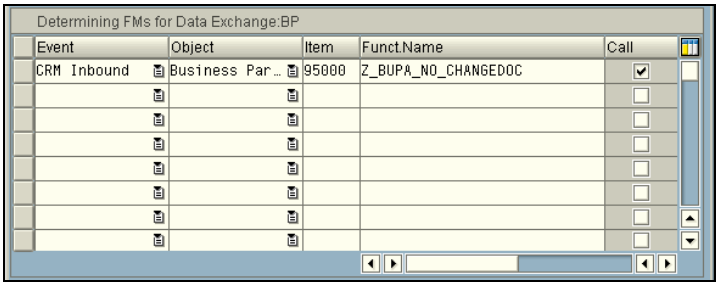
Figure 6-33 Implementation to suppress Change Documents for BUPA_MAIN

Figure 6-34 shows, how to add the function module to the call list of function modules relevant for the data exchange of business partners.

4. Call transaction CRMC_BUT_CALL_FU (Determining function modules for data exchange BP).



5. Push the ***New Entries*** button. Enter in the ***event*** field *CRM inbound (CRMIN)*, in the ***object*** field *Business Partner*, in the ***Item*** field a number < 100000, for example, 95000, in the FunctName field the name of the developed function module, for example, Z_BUPA_NO_CHANGEDOC and mark the ***Call*** parameter.



6. Push the ***Save*** button.

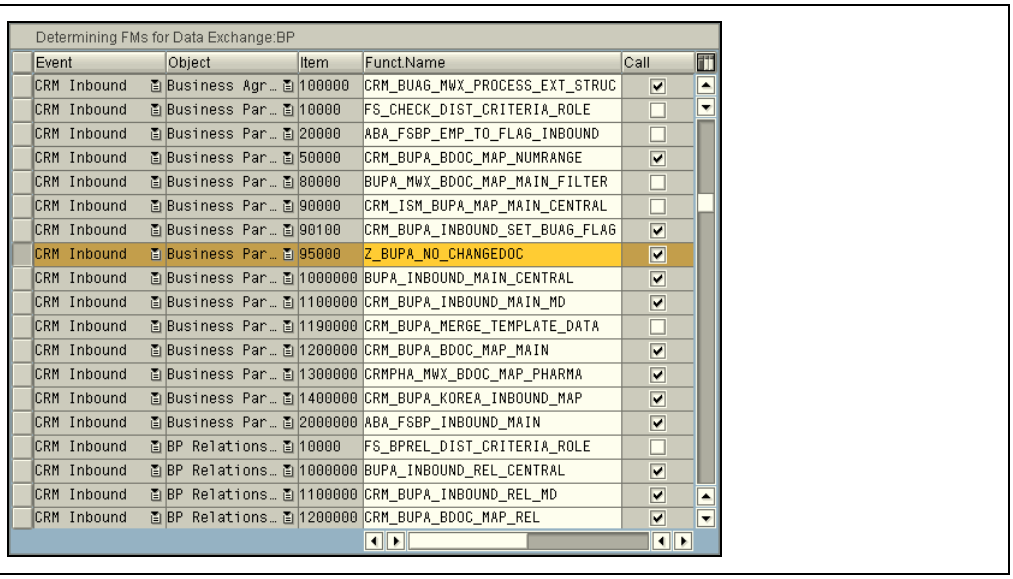


Figure 6-34: Adding a Function Module to suppress Change Documents for BUPA_MAIN

Remove the added function module from the call list on completion of the initial download or at least remove the flag of the ***Call*** parameter.

6.18.2 Business Agreement (BUAG_MAIN)

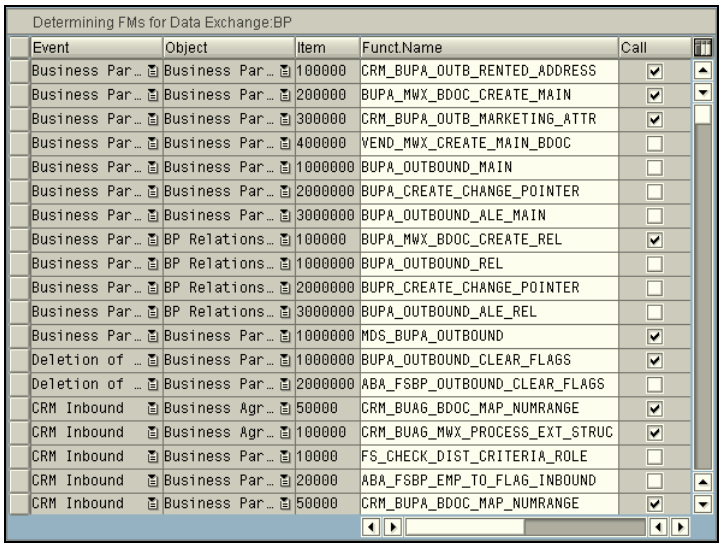
During the replication of contract accounts, the application creates various change documents when creating the respective business agreement on the CRM system. You can suppress the creation of these change documents, for example, the BUPA_FRG0130 change document, in implementing temporarily a function module. Figure 6-35 shows an implementation of a function module you can use as a basis for an own implementation. This function module deactivates not only the creation of change documents but also status checks in order to improve the performance. You may choose to change the sample source code accordingly if you do not want to deactivate both.

```
FUNCTION Z_BUAG_NO_CHANGEDOC.
*"----------------------------------------------------------------------
*"*"Local Interface:
*"  EXPORTING
*"     REFERENCE(EV_STATUS) TYPE  SMWVALSTAT
*"  TABLES
*"      CT_RETURN STRUCTURE  BAPIRET2
*"  CHANGING
*"     REFERENCE(CS_MESSAGE_EXT_STRUCT) TYPE
CRMT_BUAGS_EI_EXTERN
*"----------------------------------------------------------------------

* suppress change documents
  CALL FUNCTION 'BUPA_MWX_BDOC_NO_CDS'.

* suppress status check
  CALL FUNCTION 'BUPA_STATUS_SUPPRESS_CHECK'
    EXPORTING
      iv_suppress       = 'X'.

  ev_status = 'S'.

ENDFUNCTION.
```

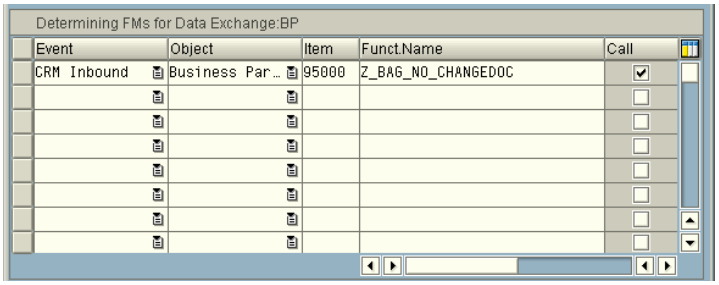
Figure 6-35 Implementation to suppress Change Documents for BUAG_MAIN

Figure 6-36 shows, how to add the function module to the call list of function modules relevant for the data exchange of business agreements.

1. Call transaction CRMC_BUT_CALL_FU (Determining function modules for data exchange BP).



2. Push the ***New Entries*** button. Enter in the ***event*** field *CRM inbound (CRMIN)*, in the ***object*** field *Business Agreement*, in the ***Item*** field a number < 100000, for example, 95000, in the FunctName field the name of the developed function module, for example, Z_BUAG_NO_CHANGEDOC and mark the ***Call*** parameter.



3. Push the ***Save*** button.

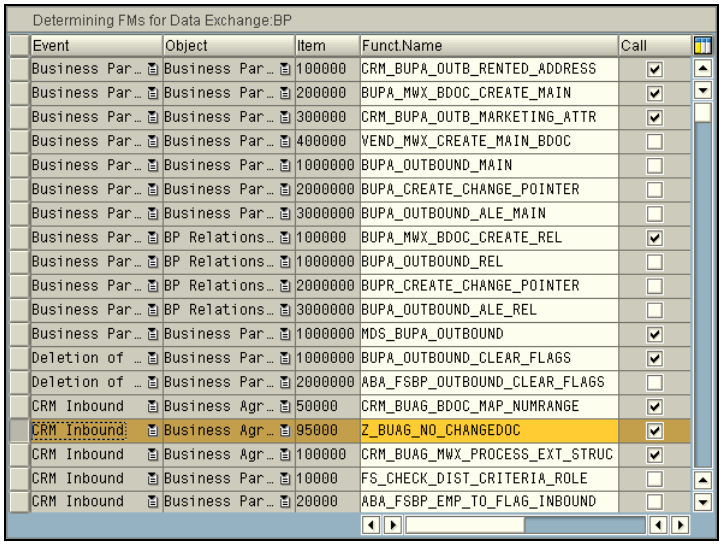


Figure 6-36: Adding a Function Module to suppress Change Documents for BUAG_MAIN

Remove the added function module from the call list on completion of the initial download or at least remove the flag of the ***Call*** parameter.

6.18.3 Connection Object (SI_CONNOBJ)

During the replication of connection objects, the application creates ADRESSE change documents when creating the address of the respective installed base in the CRM system. You can suppress the creation of these change documents in implementing temporarily the BADI_IBASE_SAVE_CHCK BAdI. Figure 6-37 shows an implementation of the CHECK_SAVE method you can use as a basis for an own implementation. If you must create change documents during the initial download, you can replace the code by the ABAP statement SET UPDATE TASK LOCAL. This forces the creation of the change documents in the actual work process instead of transferring control to an updating work process.

```
METHOD if_ex_badi_ibase_save_chck~check_save.

* suppress change documents for addresses
  CALL FUNCTION 'ADDR_SUPPRESS_CHANGE_DOCUMENTS'
    EXPORTING
      suppress_cd_type_1 = 'X'
      suppress_cd_type_2 = 'X'
      suppress_cd_type_3 = 'X'.

ENDMETHOD.
```

Figure 6-37 Implementation to suppress Change Documents for SI_CONNOBJ

Deactivate the implementation on completion of the initial download.

6.18.4 Point of Delivery (SI_POD)

During the replication of point of deliveries, the application creates ADRESSE change documents when creating the address of the respective installed base in the CRM system. You can suppress the creation of these change documents in implementing temporarily the BAdI BADI_IBASE_SAVE_CHCK. Figure 6-37 shows an implementation of the CHECK_SAVE method you can use as a basis for an own implementation. If you must create change documents during the initial download, you can replace the code by the ABAP statement SET UPDATE TASK LOCAL. This forces the creation of the change documents in the actual work process instead of transferring control to an updating work process.

Deactivate the BAdI implementation on completion of the initial download.

6.18.5 Contract (SI_CONTRACT)

During the replication of contracts, the application creates the CRM_ISUEXTA4 change documents. You can suppress the creation of these change documents in modify temporarily the CRM_ISUEXTA4_CHDOC_UP function module. Figure 6-38 shows a modification you can use as a basis for an own implementation.

```
FUNCTION CRM_ISUEXTA4_CHDOC_UP.
*"----------------------------------------------------------------------
*"*"Local Interface:
*"  IMPORTING
*"     VALUE(OBJECTID) LIKE  CDHDR-OBJECTID OPTIONAL
*"     VALUE(TCODE) LIKE  CDHDR-TCODE OPTIONAL
*"     VALUE(UTIME) LIKE  CDHDR-UTIME OPTIONAL
*"     VALUE(UDATE) LIKE  CDHDR-UDATE OPTIONAL
*"     VALUE(USERNAME) LIKE  CDHDR-USERNAME OPTIONAL
*"     VALUE(PLANNED_CHANGE_NUMBER) LIKE  CDHDR-PLANCHNGNR
DEFAULT
*"       SPACE
```



```
*"     VALUE(OBJECT_CHANGE_INDICATOR) LIKE  CDHDR-CHANGE_IND
DEFAULT
*"       'U'
*"     VALUE(PLANNED_OR_REAL_CHANGES) LIKE  CDHDR-CHANGE_IND
DEFAULT
*"       SPACE
*"     VALUE(NO_CHANGE_POINTERS) LIKE  CDHDR-CHANGE_IND DEFAULT
SPACE
*"     VALUE(UPD_ICDTXT) LIKE  CDPOS-CHNGIND DEFAULT SPACE
*"     VALUE(UPD_FLAG) LIKE  CDPOS-CHNGIND DEFAULT SPACE
*"  TABLES
*"      ICDTXT STRUCTURE  CDTXT OPTIONAL
*"      XTABLE STRUCTURE  VCRMA_ISUEXTA4
*"      YTABLE STRUCTURE  VCRMA_ISUEXTA4
*"----------------------------------------------------------------------

*<<<<< Start of Insert
  EXIT.
*<<<<< End of Insert

  CALL FUNCTION 'CRM_ISUEXTA4_WRITE_DOCUMENT' IN UPDATE TASK
    EXPORTING
      objectid                = objectid
      tcode                   = tcode
      utime                   = utime
      udate                   = udate
      username                = username
      UPD_ICDTXT_CRM_ISUEXTA4 = upd_icdtxt
      UPD_CRMA_ISUEXTA4       = upd_flag
    TABLES
      XCRMA_ISUEXTA4          = xtable
      YCRMA_ISUEXTA4          = ytable
      ICDTXT_CRM_ISUEXTA4     = icdtxt.

ENDFUNCTION.
```

Figure 6-38 Implementation to suppress Change Documents for SI_CONTRACT


Remove the modification on completion of the initial download.

6.19 Contract Replication

You can add customer-specific fields to the utility contract on the CRM system not only by means of the *Product Modeling Environment* (PME) but also by means of the *Easy Enhancement Workbench* (EEWB). This is the recommended approach if you want to use a simplified configuration for simple value storage with simple checks. The use of this lean configuration in combination with the optimized utility contract item has the following advantages over configuration with the *Product Modeling Environment* (PME):

- Significantly better performance
- Less database memory required
- Simplified error analysis
- Simplified configuration

The main disadvantage using the lean configuration is that you cannot use configurable products. For more information on the simplified configuration, see the document ***IS-U/CRM Contract Integration*** in the SAP for Utilities space of the ***SAP Service Marketplace*** (http://service.sap.com/). Choose SAP for Utilities – Product Information → SAP CRM for Utilities → Cookbook & Guidelines.

6.19.1 Transaction Type

When you create contracts (`ISU0` transaction type) with more than 20 line items (sites, points of delivery) then the system performs poorly. SAP recommends to use `ISUL` transaction type

for contracts to improve the performance. Starting with SAP CRM 7.0, so-called structured documents using lean transaction types are available for contracts.

To define which CRM transaction type the ERP system should use for the creation of contract items on the CRM system during the replication, access the SAP Customizing Implementation Guide (IMG) on the ERP system and choose *SAP Utilities → Customer Service → CRM Integration → Define Link to SAP CRM Transaction Types.*

For more information, see SAP Note 1158251.

6.19.2 Generic Data Validations for generated Customer Extensions

The system performs generic validations when processing value of the customer-specific fields you have generated with the Easy Enhancement Workbench (EEWB). To improve the performance considerably, you may choose to deactivate the generic data validation and implement own data validations in the `CHECK` method of the `CRM_GEN_EXT_BADI` BAdI. You can deactivate the generic validations for your generated extensions in setting the `NO_CHECK` indicator of the `CRMC_OBJECTS_GEN` table to X.

For more information, see SAP Note 1110305.

6.20 Parallelization of Data Migration and Replication

During a new implementation of the ERP system and the CRM system, the initial download is on the critical path of the cutover activities. The performance and, as a result, the duration of the initial download affects directly the required system outage period. Instead of treating the initial download as a discrete activity that is executed on completion of the data migration process into the ERP system, further minimization of the cutover period is possible in parallelizing the data migration activities into the ERP system and the initial download into the CRM system.

An execution of the migration and replication activities at the same time requires additional system resources. If the ERP and the CRM system are sharing the same resources, any replication process will consume resources that would not be available for the data migration process and, consequently, would extend the data migration window. In addition, dependencies between migration objects and replication objects limit the options for a parallelization of data migration and initial download. A well-tuned initial download process requires all system resources on the CRM system. Additionally, some system resources on the ERP system are consumed to select and retrieve the data on the ERP system and send them to the CRM system. When planning the data migration, you must not disregard the consumption of database resources just for the data selection on the ERP system even the resource consumption will be comparably small (5-10%).

Parallelization requires a good preparation and alignment of the related activities. A close collaboration is required between the data migration team, the team executing the initial download and the basis team monitoring the activities in the two involved systems.

You can implement either one of the following replication strategies or a mixture of those:

Strategy 1 The initial download starts on completion of the data migration of all replication relevant business objects. This is the standard strategy. The figure below shows the sequence of the activities starting with the data migration into the ERP system followed by an initial download of all the relevant business objects into the CRM system.

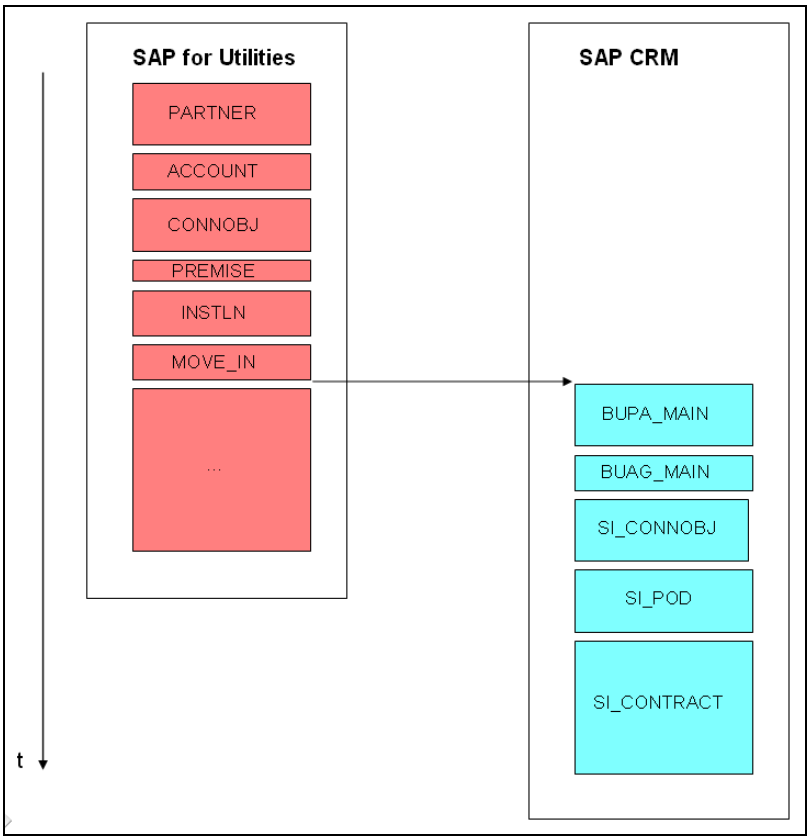


Figure 6-39 Parallelization Strategy 1

Strategy 2 The initial download of one kind of business object starts on completion of the data migration of that particular business object. Data migration is continued with another business object while the replication of the completely migrated business object is in progress. The figure below shows the sequence of the activities.

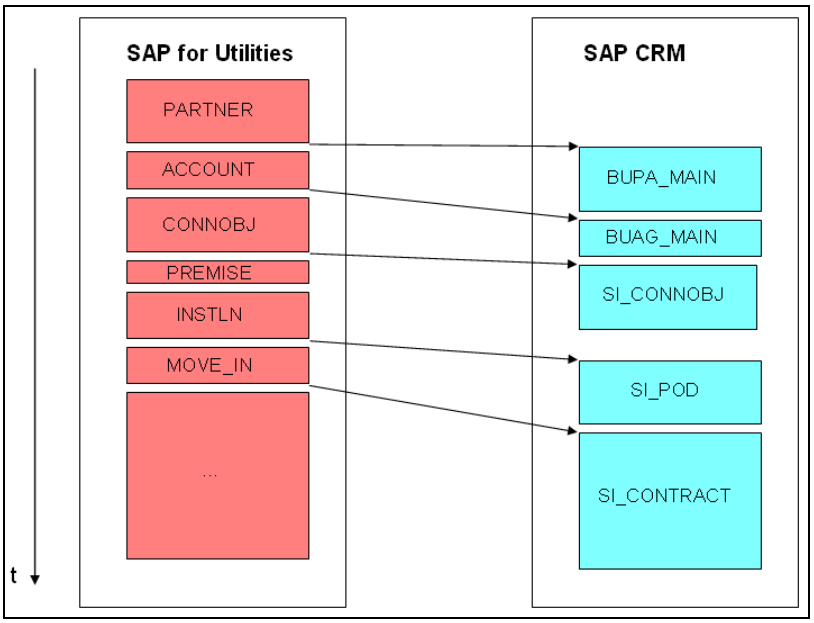


Figure 6-40 Parallelization Strategy 2

Strategy 3 The initial download of one kind of business object starts on completion of the data migration of only a subset of the same business object. Data migration is continued for the next subset of the same business object while the replication is in progress. The figure below shows the sequence of the activities.

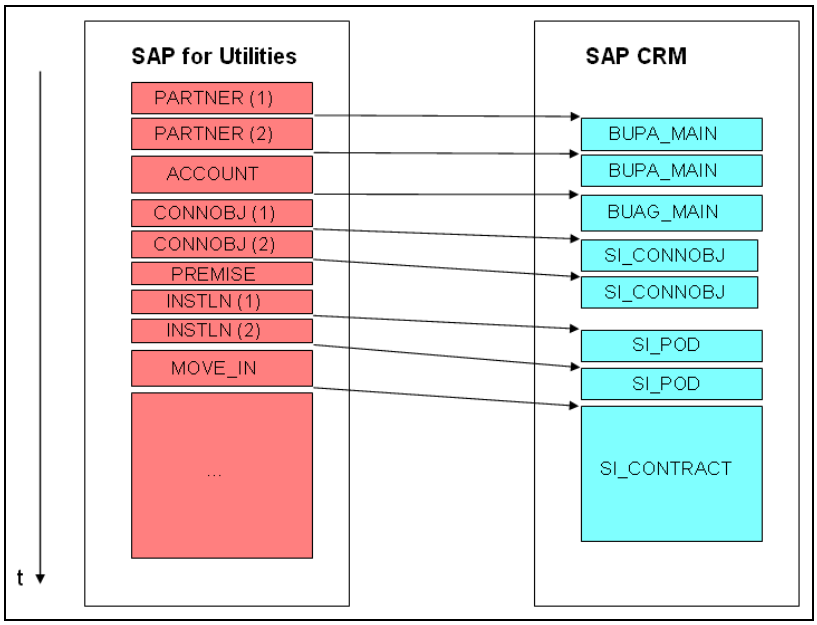


Figure 6-41 Parallelization Strategy 3

Strategy 4 The initial download starts on completion of the data migration of all replication relevant business objects of a set of business partners (for example, all data belonging to business partners of a region). Data migration is continued for the data of the next set of business partners while the replication of the business objects of the first set is in progress. The figure below shows the sequence of the activities.

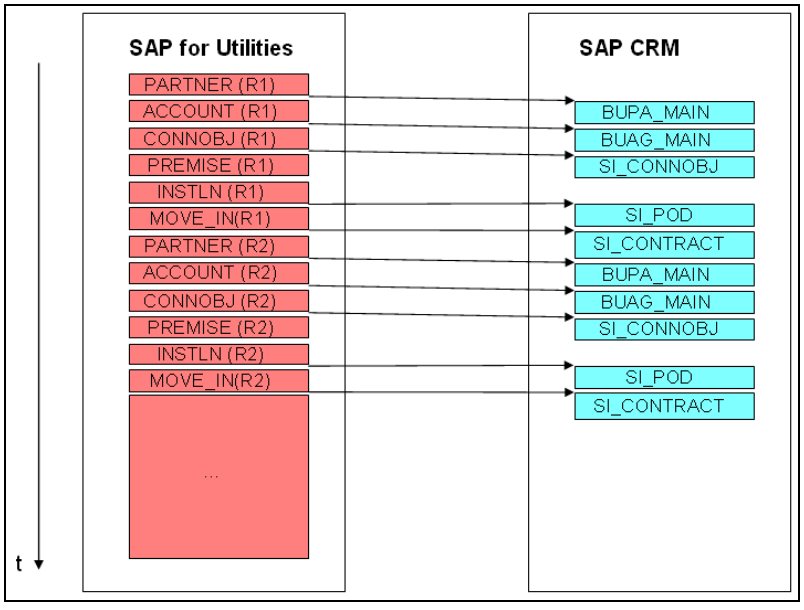


Figure 6-42 Parallelization Strategy 4

There are two important settings on the ERP system that control the automatic outbound processing of BDocs through the CRM Middleware:

- Transaction BF11 (Business Transaction Event Application Indicator): BC-MID application is active or inactive (***application active*** indicator). You can leave the CRM Middleware on the ERP system inactive until you initiate the first initial download of a business object into the CRM system. Once you activate the CRM Middleware, the application in the ERP system initiates an undesired delta download when creating a replication relevant business object for example, during data migration. Except for the standard strategy, you

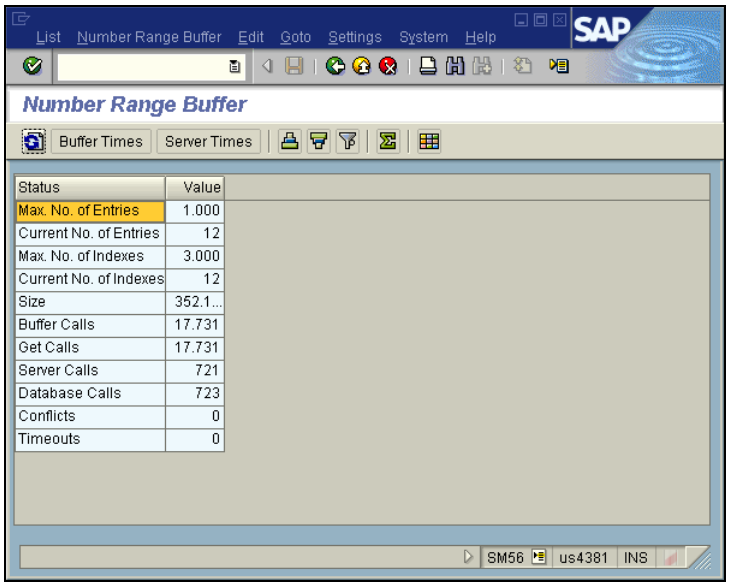
must activate the CRM Middleware from the beginning to allow the execution of an initial download.

- IS Migration Workbench (ISMW): CRM_REPL_OFF control parameter: This parameter controls whether the replication of the migrated business object into the CRM system with a delta download should be suppressed when the CRM Middleware is active. For more information, how to configure this parameter, see the ***Guideline IS Migration Workbench*** in the SAP for Utilities space of the ***SAP Service Marketplace*** (http://service.sap.com/). Except for the standard strategy, you must use CRM_REPL_OFF control parameter to suppress the automatic replication of the created business objects during data migration. For more information, see chapter 6.22 *Phased Migration*.

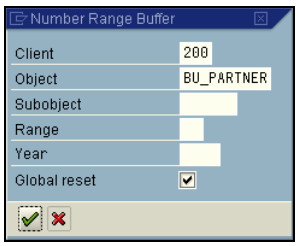
Before you start the initial download, you must define the lower and upper limit of the intervals of each of the requests on the ERP system. For the lower limit of the first request, check the highest used identification number of the already existing business objects (see Figure 5.9 for a list of tables in the ERP system). If you do not determine the intervals correctly, you will either not replicating business objects or replicating business object that already have been replicated into the CRM system. A re-replication of business objects with requests is possible at anytime but extends the replication period.

Most of the number range objects of the replication relevant business objects are buffered in memory due to performance reasons (for more information about number range object buffering, see chapter 6.14 *Buffering Number Range Objects in Memory*). The assignment of consecutive numbers to business objects cannot be guaranteed if processes are executed on more than one application server because every application servers maintains its own buffer of free numbers. Therefore you must delete the numbers that are still available in the number buffers on all application servers prior to start the data migration of the related business objects. For example, for must delete the reset number range object BU_PARTNER prior to start data migration of business partners. This prevents a lower number being assigned to a new business partner if not all numbers of one application server have been used. Figure 6-43 shows, how to delete entries in the number range buffer.

1. Call transaction SM56 (Number Range Object Maintenance) and enter the name of the number range object in the ***Object*** field.



2. Choose *Number Range Buffer* → *Reset*. Enter the ***Client*** and the name of the number range object in the ***Object*** field. In addition, select the ***Global Reset*** indicator to delete the buffer entries on all application servers.



3. Push the ***Continue*** button.

Figure 6-43 Delete Entries in the Number Range Buffer

6.21 Preparation of Contract Replication (SI_CONTRACT)

A migrated contract does not contain any CRM product information. In production, the users create utility contracts in the CRM system. Then the contracts are replicated automatically into the ERP system together with a CRM product that has been allocated to the contract on the CRM system. On the ERP system, the master data generator (MDG) processes the BDoc data, determines a master data template by means of the product name, and subsequently processes the template. A master data template describes how to create the required business master of a utility contract.

Before you start the initial download of migrated contracts, you must allocate a product to each contract. SAP delivers the ECRM_GENERATE_EVERH report to determine and allocate a suitable product to a utility contract and creates an entry in the EVERH table. The report dies the determination based on the tariff information of the related installation and the available master data templates. The following fields and options are available to select contracts and execute the report as part of the data migration process:

Field	Description
VTR_AB	Lower limit of the contract interval
VTR_BIS	Upper limit of the contract interval
GEN_FROM	Move-out date (for example, 31.12.9999)
GENERATE	'X'
SIMULATE	A list of the intended database updates is display; no database updates are performed.

Figure 6-44 Parameter of the ECRM_EVERH_CREATE Report

You can choose between three options to mark a contract regarding an automatic replication to CRM with a delta load:

Field	Description	XREPLCNT Field
GENERATE	Marks contracts to be initially replicated only with a request load (mass tariff change)	C
MASSDATA	You choose this option when preparing an initial download of migrated contracts as part of the data migration process. All selected contracts will be marked as to be replicated	M
ISUONLY	Marks contracts as never to be replicated to CRM with neither an initial load nor a delta load.	O

Figure 6-45 Options to execute the ECRM_EVERH_CREATE Report

Figure 6-46 shows the selection screen of this report. The ***VTR_AB*** and ***VTR_BIS*** fields allow a parallel execution of the report based on distinct intervals.

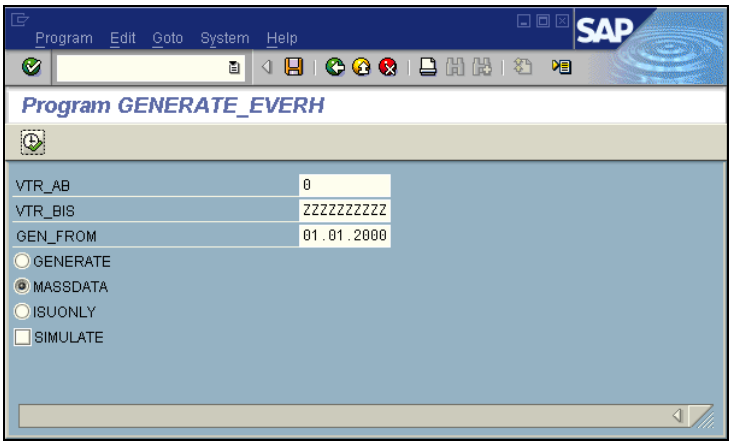
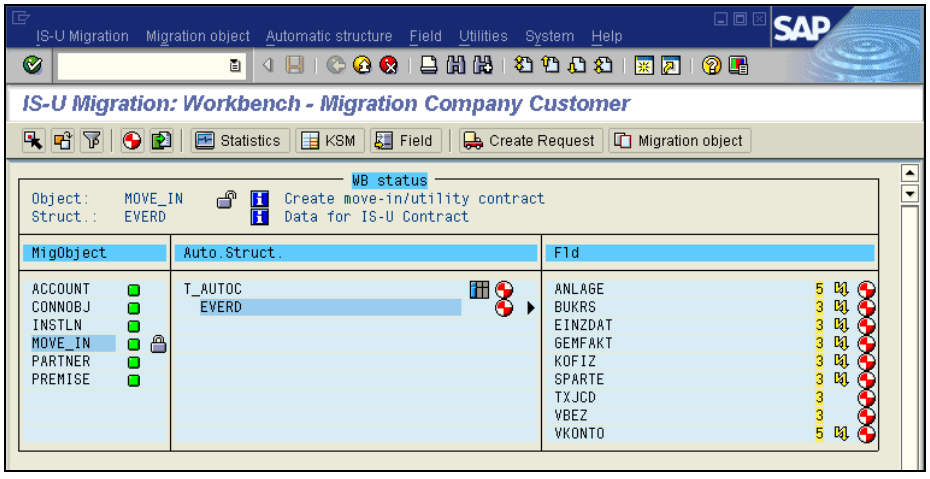


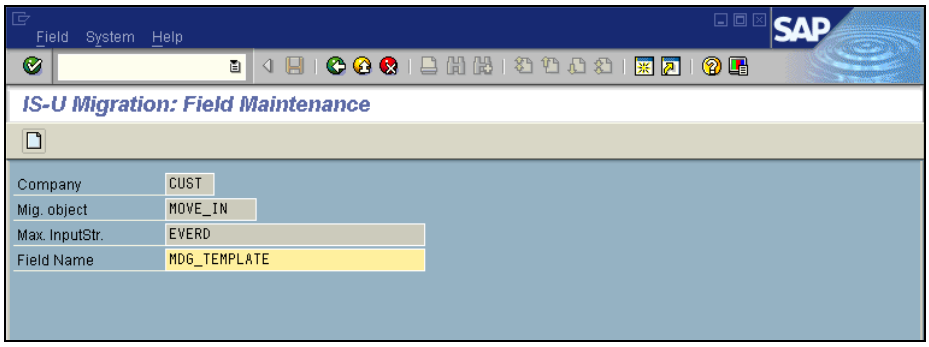
Figure 6-46 Selection Screen of the ECRM_EVERH_CREATE Report

Instead of using the ECRM_GENERATE_EVERH report, you may choose to create the necessary entries in the EVERH table during the migration of contracts into the ERP system. This is possible, if you already know the product name of the contracts when creating the import file for the contracts. Figure 6-47 shows some important changes to the Customizing of the MOVE_IN migration object in the ERP system that you can use as a basis for an own development. The objective is to transfer the name of the master data template in the migration import file and create the entry in the EVERH table when processing the data. For more information about the IS Migration Workbench, see the ***Guideline IS Migration Workbench*** in the SAP for Utilities space of the ***SAP Service Marketplace*** (http://service.sap.com/).

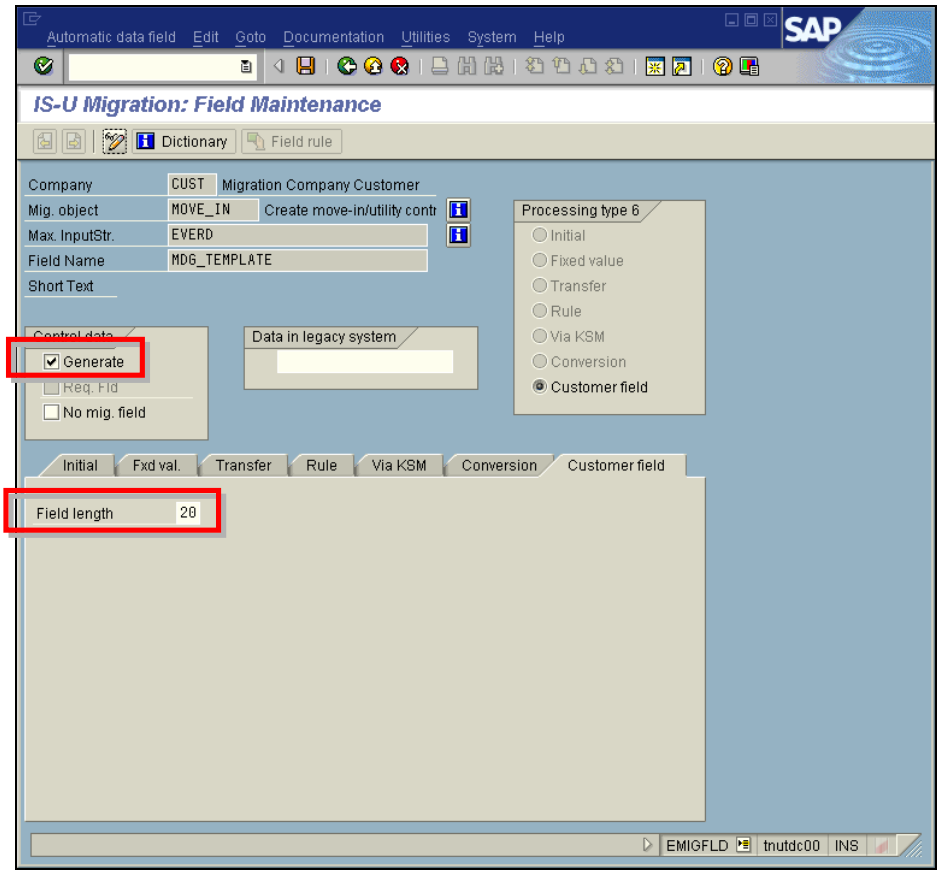
1. Call transaction EMIGALL (IS Migration Workbench) and mark the MOVE_IN migration object. Then mark the EVERD structure with a double-click.



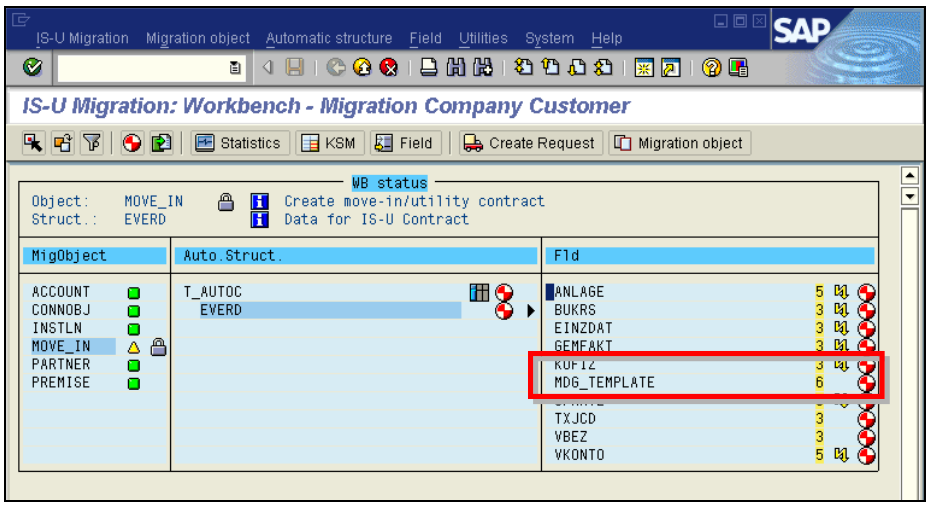
2. Choose *Field* → *Create*. Enter the name of the new field in the ***Field Name*** field.



3. Push the *Create* button.
4. Select the ***Generate*** indicator and enter 20 (length of the template field) in the ***Field length*** field.



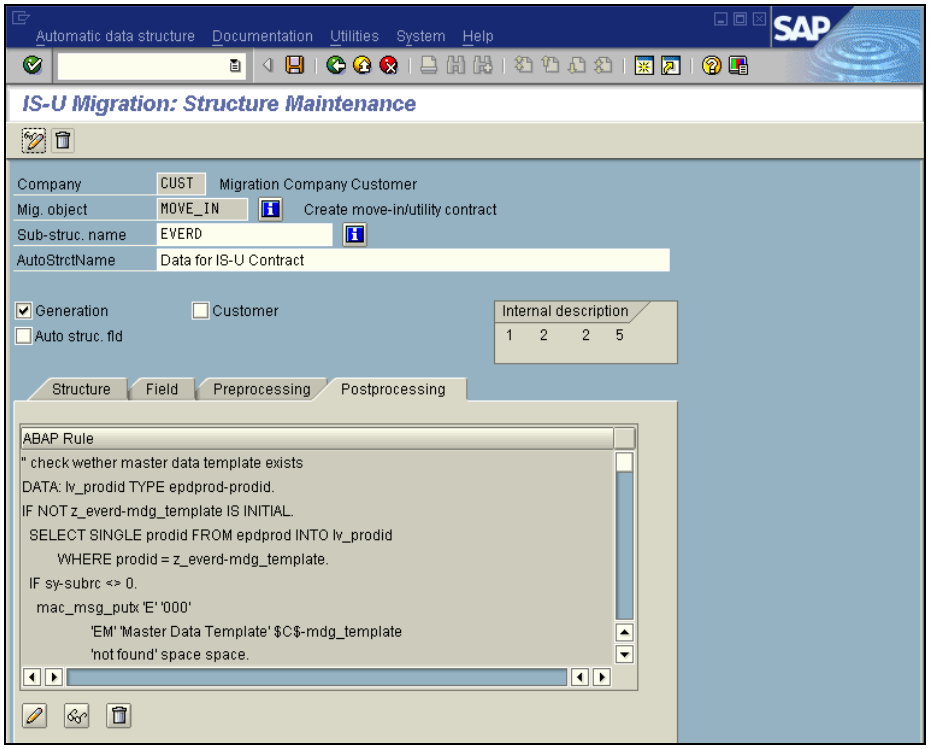
5. Save the field Customizing and return to the main screen of the IS Migration Workbench to display the newly created field. The new field is part of the customer structure and you can transfer a value for this field in the import file.



6. Choose *Automatic Structure* → *Change*. On the structure maintenance screen choose the *Postprocessing* sub screen. Then push the *Change* button and enter the following ABAP code. The ABAP code checks whether the transferred master data template exists and determines the related product and product GUID.

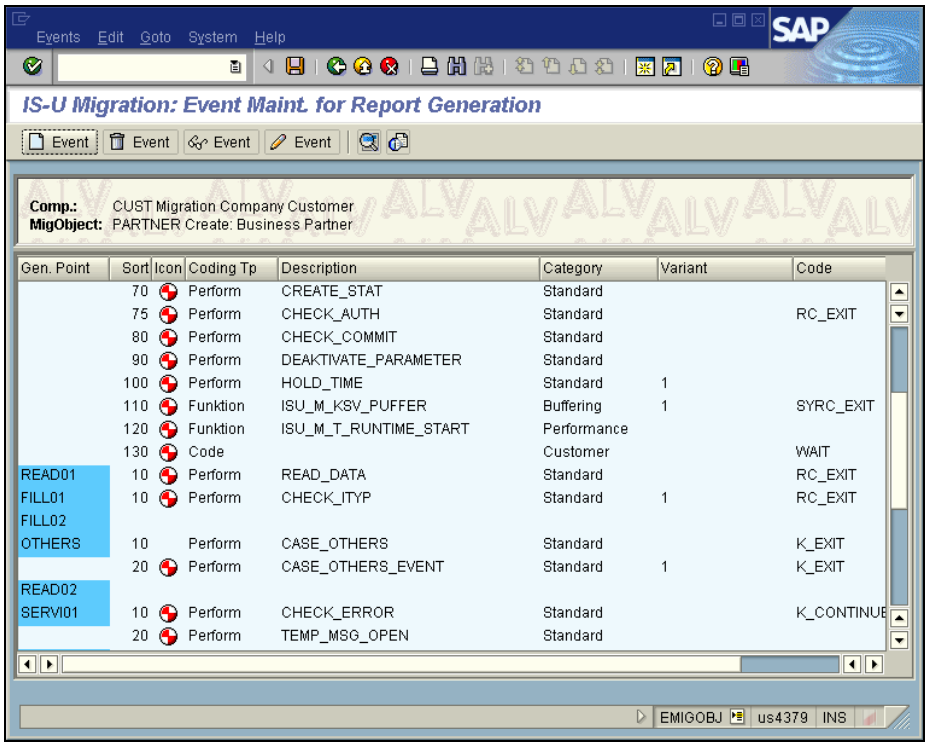
```
" check wether master data template exists
DATA: lv_prodid TYPE epdprod-prodid.
IF NOT z_everd-mdg_template IS INITIAL.
  SELECT SINGLE prodid FROM epdprod INTO lv_prodid
          WHERE prodid = z_everd-mdg_template.
  IF sy-subrc <> 0.
    mac_msg_putx 'E' '000'
                 'EM' 'Master Data Template' $C$-mdg_template
                  'not found' space space.
    mac_jump 6.
  ENDIF.
ENDIF.
```

7. Push the *Save* button to save the postprocessing code.

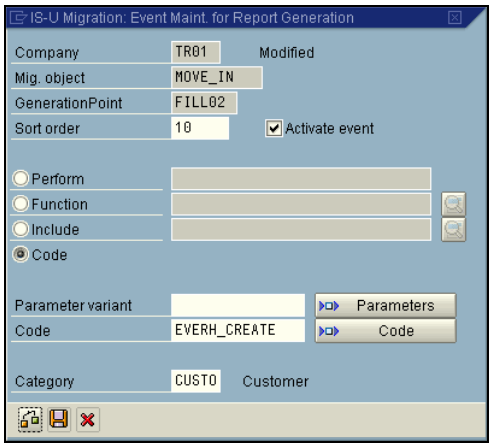


8. Save the structure Customizing and return to the main screen of the IS Migration Workbench.

9. Choose *Migration Object* → *Change*. On the migration object maintenance screen, choose *Utilities* → *Events*.



10. Choose the FILL02 ***generation point*** and push the *Create Event* button. Enter *10* in the ***Sort Order*** field and select the ***Activate event*** indicator. Then choose the ***Code*** radio button and enter *EVERH_CREATE* in the ***Code*** field.



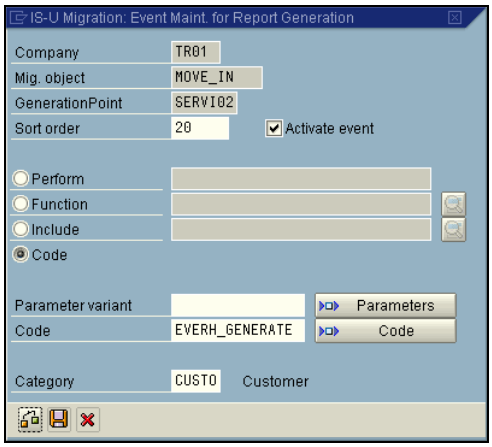
11. Push the *Code* button and enter the following ABAP code. Then push the *Save* button. The ABAP code creates the EVERH entries in the database.

```
" data definition
DATA: lt_everh TYPE TABLE OF everh WITH KEY anlage,
      wa_everh TYPE everh.

" fill internal table EVERH entry with contract data
IF jump-knz = space.
  CLEAR wa_everh.
  wa_everh-vertrag     = everd-vertrag.
  wa_everh-bis         = everd-auszdat.
  wa_everh-xreplcntl   = 'M'.
  wa_everh-gen_date    = sy-datum.
  wa_everh-ab          = everd-einzdat.
  wa_everh-anlage      = everd-anlage.
```



```
  IF NOT z_everd-mdg_template IS INITIAL.
    SELECT SINGLE crm_product crm_product_guid
           FROM epdprod
           INTO (wa_everh-crm_product, wa_everh-product_guid)
           WHERE prodid  = z_everd-mdg_template.
  ENDIF.
  wa_everh-prodid      = z_everd-mdg_template.
  wa_everh-erdat       = sy-datum.
  wa_everh-ernam       = sy-uname.
  APPEND wa_everh TO lt_everh.
ENDIF.
```

12. Push the *Save* button to save the new event.
13. Choose the SERVI02 ***generation point*** and push the *Create Event* button. Enter *20* in the ***Sort Order*** field and select the ***Activate event*** indicator. Then choose the ***Code*** radio button and enter *EVERH_GENERATE* in the ***Code*** field.



14. Push the *Code* button and enter the following ABAP code then push the *Save* button.

```
" data definition for default product
DATA: lv_crm_dummy_product TYPE esales_crm_product,
      lv_crm_dummy_product_guid TYPE guid16.

" get the default product
CALL FUNCTION 'ISU_CRM_GET_DUMMY_PRODUCT'
  EXPORTING
    x_service        = space
  IMPORTING
    y_dummyproduct = lv_crm_dummy_product
    y_product_guid = lv_crm_dummy_product_guid
  EXCEPTIONS
    others         = 1.

" determine the CRM product GUID for each contract
LOOP AT iever INTO j_ever.

  READ TABLE lt_everh INTO wa_everh
                      WITH KEY anlage = j_ever-anlage.
  IF sy-subrc = 0 AND NOT wa_everh-crm_product IS INITIAL.
    IF wa_everh-crm_product = lv_crm_dummy_product.
      wa_everh-product_guid = lv_crm_dummy_product_guid.
    ENDIF.
  ENDIF.

  " complete the entry of the EVERH table and save it
  wa_everh-vertrag = j_ever-vertrag.
  wa_everh-bis = j_ever-auszdat.
  INSERT everh FROM wa_everh.

ENDLOOP.

REFRESH lt_everh.
```


15. Push the *Save* button to save the new event.
16. Return to the main screen of the IS Migration Workbench and generate the load program.

Figure 6-47 Additional Customizing of the MOVE_IN migration object

6.22 Phased Migration

6.22.1 Motivation

In a phased data migration, you want to replicate technical and business master data in stages from the ERP system into the CRM system. Between the migration phases, the system is in production and the users create master and transactional data in the CRM system that is replicated automatically into the ERP system by means of the delta replication. During any migration phase, a replication of a business object should take place only after you have not only created the object itself but also all required changes to this object in order to avoid the need of a replication of the same business object twice.

In subsequent stages, besides migrating new business objects, you want to migrate also changes to business objects that have already been used in production. A change could be, for example, the creation of a new address of a business partner.

In the first stage, an initial download is the most efficient way to replicate all required business objects from the ERP system into the CRM system. Starting with stage two, the following options are available for a replication of business objects into the CRM system:

Option 1 You deactivate the delta replication during data migration into the ERP system and replicate all business objects into the CRM system with an initial download regardless whether they already have been replicated into the CRM system before or you migrated them just in this stage. This is an option, if the additional volume of already replicated business objects is manageable. The advantage of this option is the execution of a simple initial download disregarding whether one or all of the already replicated business objects have been updated in the current phase.

Option 2 You activate the delta replication activated during data migration. The import jobs of the IS Migration Workbench create BDocs for the created or changed business objects and sends it to the CRM system (the number of business objects per BDoc usually depends on the used commit buffering value of the migration object). The advantage of this approach is that the replication into the CRM system takes place in parallel to the data migration into the ERP system. The disadvantage is that it is difficult to control the load not only in the ERP system but also in the CRM system by the import jobs of the data migration activity. This is because an import job on the ERP system compares as a CRM request with multiple inbound queues on the CRM system. This may lead to an overload in the CRM system if too many import jobs are running in the ERP system together with too many inbound queues in the CRM system. On the other hand, the execution of too few import jobs on the ERP system may lead to an extended migration window due to a reduced overall throughput of the data migration process.

Option 3 You separate import files containing the data for the creation of business objects from import files containing the data to update already existing business objects. Then you perform the data migration in two separate steps:

- During the migration of new business objects, you deactivate delta replication and replicate the created business objects subsequently with an initial download with a high performance into the CRM System. The filters settings of the required requests correspond to the number range of the newly created business objects.
- During the migration of changes to already existing business objects, you activate the delta replication. This is a feasible option if the volume of updates and their replication is manageable. The alternative is to deactivate the delta replication also for the

required updates of business objects and replicate them into the CRM system in a subsequent step.

You will need to take a decision per business object and based on the volumes of already existing business objects, number of business objects to be updated and the available cutover window. Usually, you do not want the delta replication happening automatically during data migration due to performance issues when migrating and replicating at the same time. On the other hand, you want to avoid a re-replication of all business objects after you have changed during data migration only very few out of many already replicated business objects.

However, what are the best strategies to replicate not only the newly created business objects you can identify easily by their number range but also only changed business objects? How can you identify and replicate them with an initial download? What is the most efficient approach to reduce the migration window? Do both ERP and CRM system share the same system resources, such as, CPUs and memory? Particularly, if both ERP and CRM systems share the same system resources, the options are limited because any data migration activity affects the replication performance into the CRM system and vice versa.

There is no unique answer to these questions. It is important to understand that any concurrent data migration and replication activity increases the complexity of the process thus makes it more difficult to test. The next chapters highlight known challenges and possible solutions and provide further details for a replication of the following business objects:

- Business partner (adapter object BUPA_MAIN)
- Contract account (adapter object BUAG_MAIN)
- Business partner contacts (adapter object SI_BCONTACT)
- Business partner relationship (adapter object BUPA_REL)
- Connection object (adapter object SI_CONNOBJ)
- Point of delivery (adapter object SI_POD)
- Contract (adapter object SI_CONTRACT)

One of the key parameter for a data migration and replication into already productive systems is the `CRM_REPL_OFF` control parameter in the IS Migration Workbench (ISMW). With the `CRM_REPL_OFF` control parameter, you can deactivate any delta replication activity for the business objects you are creating or changing during data migration with the ISMW without changing any further parameter for a deactivation of the CRM Middleware. The parameter can be set per migration object. Figure 6-48 shows the an example with different settings for the migration objects that create business objects (ACCOUNT and PARTNER migration objects) and that changes an existing business object (ACCOUNTCHA and PARTNERCHA migration objects). A definition for the PARTNER and ACCOUNT migration objects is not required because the standard setting of the `CRM_REPL_OFF` control parameter is X (*replication off* or *replication deactivated*).

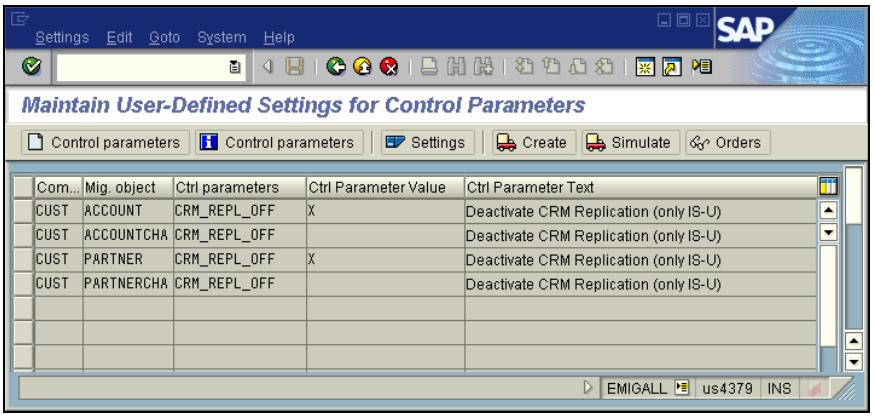


Figure 6-48 Usage of `CRM_REPL_OFF` Control Parameter in the ISMW

6.22.2 Business Partner

In the SAP standard Customizing, there is no field available to create a request filter in the CRM system to select and replicate only business partners that have been changed on a specific date or by a specific user. Appropriate fields in the BUT000 table (header table of business partners), such as, CHUSR (last user to change object) or CHDAT (date on which the business partner was last changed) are not available as filter fields are missing and need to add them to the SMOFFILFLD table (Allowed fields for Filter definition) in the CRM system first. You maintain the SMOFFILFLD table with transaction SM30 (Maintain Table View). Figure 6-49 shows the definition of the new selection field CHDAT (date on which the business partner was last changed).

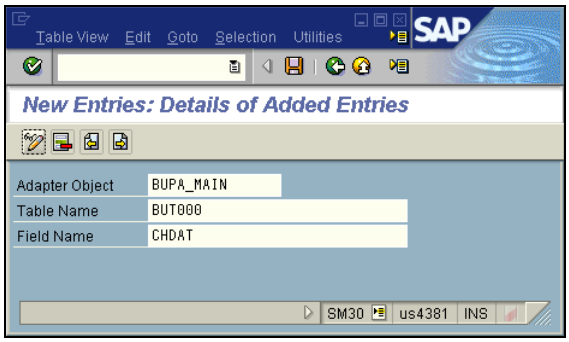


Figure 6-49 Definition of a new Field for a Request

Figure 6-50 shows a request that uses this new field in its filter. With the newly defined filter field you can define a filter selecting only business partners that have been updated on a specific date (for example, 1st of March 2009 that is 20090301 in the system internal date format yyyymmdd). Additionally, you can add a number range to allow a concurrent execution of more than one request. No further Customizing changes are required if you add only fields of the BUT000 table because the extractor on the ERP system has been developed to select any business partners only based on filters related to the BUT000 table.

You should check before starting the initial download whether a new table index is required based on the filter fields on the BUT000 table on the ERP system. Otherwise, you may experience performance problems during selection of business partners according to the filter settings.

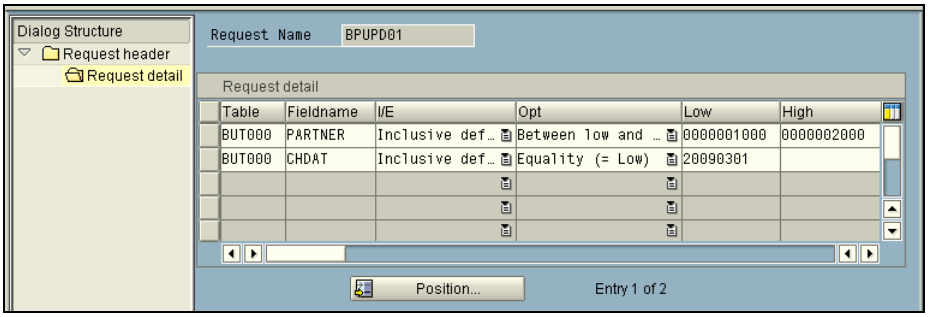


Figure 6-50 Filter of a Request to replicate only updated Business Partners

If a selection of the relevant business partners based on the BUT000 table is not sufficient, you need to develop an own extractor module on the ERP system. You can use the standard extractor for business partners (CRS_BUPA_MAIN_GET_ID_LIST function module) as a reference. Figure 6-51 shows a custom extractor that can be used as a basis for an own development.


```
FUNCTION ZCRS_BUPA_MAIN_GET_ID_LIST.
*"----------------------------------------------------------------------
*"*"Local Interface:
*"  IMPORTING
*"     REFERENCE(I_BAPICRMDH1) TYPE  BAPICRMDH1 OPTIONAL
*"  TABLES
*"      ET_BAPIIDLIST STRUCTURE  BAPIIDLIST
*"      IT_RANGE STRUCTURE  CRMSELSTR
*"----------------------------------------------------------------------

* type definition
  TYPES: BEGIN OF ty_bp_key,
           partner TYPE bu_partner,
           partner_guid TYPE bu_partner_guid,
         END OF ty_bp_key.
  TYPES: BEGIN OF ty_ksv_key,
           newkey TYPE bu_partner,
         END OF ty_ksv_key.

* data definition
  DATA: BEGIN OF lt_addrnum OCCURS 0,
           addrnumber TYPE ad_addrnum,
        END OF lt_addrnum.
  DATA lv_addr_guid TYPE bu_address_guid.
  DATA lt_bp TYPE TABLE OF ty_bp_key.
  DATA ls_bp TYPE ty_bp_key.
  DATA lt_ksv TYPE TABLE OF ty_ksv_key.
  DATA ls_ksv TYPE ty_ksv_key.
  DATA ls_bapiidlist TYPE bapiidlist.
  DATA lt_where_cond TYPE where_tab OCCURS 0 WITH HEADER LINE.

* check whether TEMKSV table is included in filter
  LOOP AT it_range WHERE TABLE = 'TEMKSV'.
    EXIT.
  ENDLOOP.
  IF sy-subrc = 0.

*   create where clause
    REFRESH et_bapiidlist.
    CALL FUNCTION 'CONVERT_SELECT_INTO_WHERE'
      EXPORTING
        select_tablename = 'TEMKSV'
      TABLES
        in_select_fields = it_range
        out_where_cond   = lt_where_cond
      EXCEPTIONS
        wrong_input_data = 1
        OTHERS           = 2.
    IF sy-subrc = 0.

*     select entries from TEMKSV
      SELECT newkey FROM temksv INTO TABLE lt_ksv
        WHERE (lt_where_cond).

*     return if lt_ksv is initial.
      CHECK sy-subrc = 0.

*     select business partners
      SELECT partner partner_guid FROM but000 INTO TABLE lt_bp
        for all entries in lt_ksv
        WHERE partner = lt_ksv-newkey.

*     return if lt_bp is initial.
      CHECK sy-subrc = 0.

*     fill bapiidlist
```



```
      LOOP AT lt_bp INTO ls_bp.
*       Check whether BP have a valid GUID already
        IF ls_bp-partner_guid IS INITIAL.
          CALL FUNCTION 'SYSTEM_UUID_CREATE'
            IMPORTING
              uuid = ls_bp-partner_guid.
          UPDATE but000
            SET partner_guid = ls_bp-partner_guid
            WHERE partner = ls_bp-partner.
        ENDIF.
        ls_bapiidlist-id = ls_bp-partner_guid.
        APPEND ls_bapiidlist TO et_bapiidlist.
      ENDLOOP.

*     do the same for the BP addresses
      SELECT addrnumber INTO TABLE lt_addrnum FROM but020
         FOR ALL ENTRIES IN lt_bp WHERE
            partner = lt_bp-partner AND
            address_guid IS NULL.
      LOOP AT lt_addrnum.
        CALL FUNCTION 'SYSTEM_UUID_CREATE'
          IMPORTING
            uuid = lv_addr_guid.
        UPDATE but020
          SET address_guid = lv_addr_guid
          WHERE addrnumber = lt_addrnum-addrnumber.
      ENDLOOP.
      COMMIT WORK AND WAIT.
    ENDIF.
  ENDIF.

ENDFUNCTION.
```

Figure 6-51 Implementation to select Business Partners based from TEMSKV Table

The ZCRS_BUPA_MAIN_GET_ID_LIST function module selects business partners based on the entries in the TEMKSV table. The TEMKSV table is updated with the legacy system key and SAP key of the created or updated business object during data migration with the PARTNER or PARTNERCHA migration object of the IS Migration Workbench. You must register your custom extractor at ***Time*** *Extract* on the ERP system with transaction COM_BUPA_CALL_FU (Determining Function Modules for Data Exchange). To support still requests with a filter based on the BUT000 table, you should add also the standard extractor (CRS_BUPA_MAIN_GET_ID_LIST function module). Figure 6-52 shows the registered extractors.

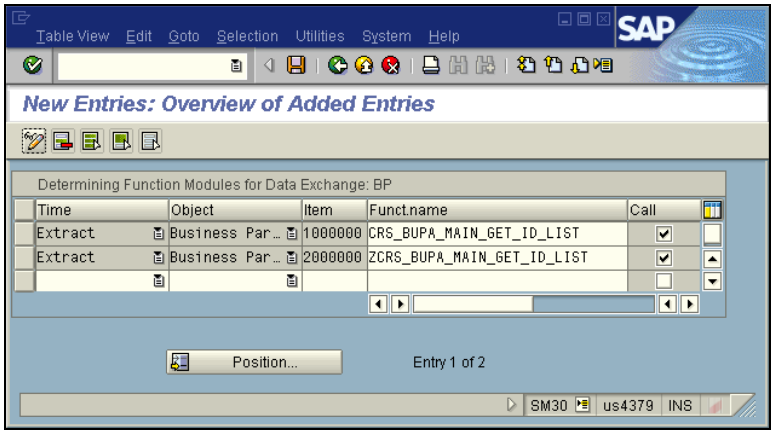


Figure 6-52 Custom Extractor in COM_BUPA_CALL_FU Transaction

In addition, you must add the fields FIRMA (Migration Company), OBJECT (Migration Object) and, if required, OLDKEY (Legacy System Key) to the allowed field list for BUPA_MAIN adapter object in the SMOFFILFLD table in the CRM system as shown in Figure 6-53.

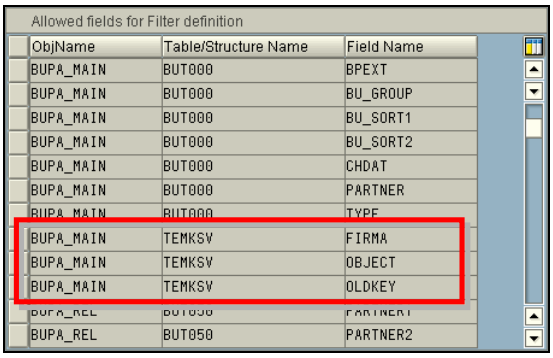


Figure 6-53 Definition of a new Fields for a Filter based on the TEMKSV Table

Then, you can create a request with transaction R3AR2 (Define CRM Request) referring to the entries in the TEMKSV table to select only business partners that have been updated during data migration with the PARTNERCHA migration object. Figure 6-54 shows the filter definition of such a request.

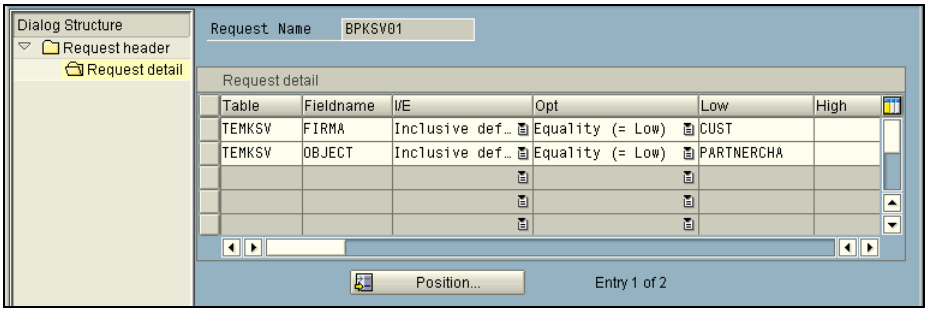


Figure 6-54 Filter to replicate only BP updated with PARTNERCHA Migration Object

Do not define a filter for a request based on both BUT000 and TEMKSV tables. The reason for this is that both extractors are executed separately combining in this way the filters for both tables with an OR statement. The volume of selected business partner could be higher than desired.

6.22.3 Business Partner Relationship

The problem to identify and replicate only updated business partner relationships with an initial download is the same as for a replication of updated business partners except that the volume of changed or new business partner relationships usually is quite small. Therefore should always consider the option to re-replicate all relationships if the volume is manageable.

In the SAP standard Customizing, there is no field available to create a request filter in the CRM system to select and replicate only business partners that have been changed on a specific date or by a specific user. Appropriate fields in the BUT050 table (header table of business partner relationships), such as, CHUSR (last user to change object) or CHDAT (date on which the business partner relationship was last changed) are not available as filter fields are missing and need to add them to the SMOFFILFLD table (Allowed fields for Filter definition) in the CRM system first. You maintain the SMOFFILFLD table with transaction SM30 (Maintain Table View). Figure 6-55 shows the definition of the new selection field CHDAT (date on which the business partner relationship was last changed).

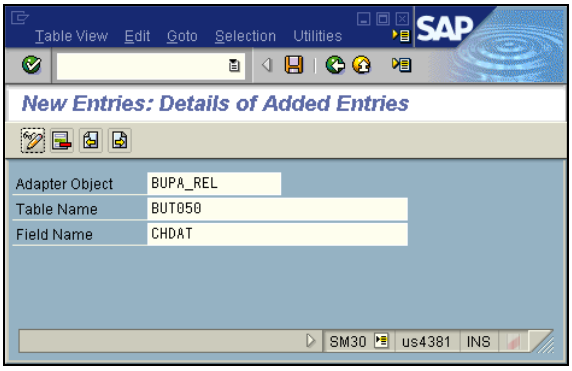


Figure 6-55 Definition of a new Field for a Request

You should check before starting the initial download whether during the replication a new table index based on the filter fields is required on the BUT050 table on the ERP system to perform the data selection efficiently.

If a selection of the relevant business partner relationships based on the BUT050 table is not sufficient, you need to develop an own extractor module on the ERP system for this based on the standard extractor for business partners (CRS_BUPA_REL_GET_ID_LIST function module). Figure 6-51 shows a custom extractor for business partner that can be used as a basis for an own development. The implementation for a selection and replication of business partner relationships is very similar.

You must register your custom extractor at ***Time*** *Extract* in the COM_BUPA_CALL_FU table on the ERP system with transaction SM30 (Maintain Table View). To support still requests with a filter based on the BUT050 table, you should add also the standard extractor (CRS_BUPA_REL_GET_ID_LIST function module). In addition, you must add the fields FIRMA (Migration Company), OBJECT (Migration Object) and, if required, OLDKEY (Legacy System Key) to the allowed field list for BUPA_REL adapter object in the SMOFFILFLD table in the CRM system. Then, you can create a request with transaction R3AR2 (Define CRM Request) referring to the entries in the TEMKSV table to select only business partner that have been created during data migration with the PART_REL migration object:

6.22.4 Business Partner Contacts

You have two options to deactivate temporarily the replication of business partner contacts during data migration:

- Choose the ***Suppress Delta Data Comparison*** indicator in the Central Mapping Settings for business partner contacts. In the SAP Customizing Implementation Guide (IMG) choose *SAP Utilities* → *Customer Service* → *CRM Integration* → *Contact Replication* → *Make Central Mapping Settings*. Figure 6-56 shows the IMG activity.

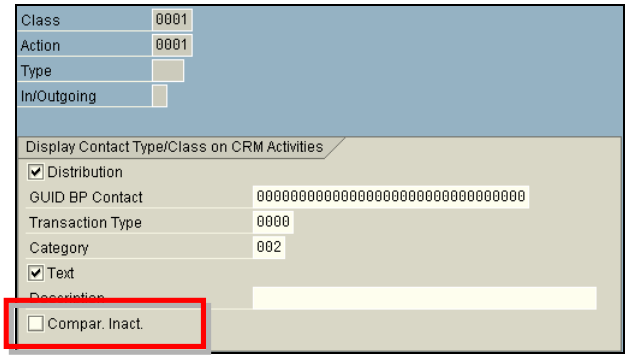


Figure 6-56 Suppress Delta Data Comparison Indicator

- Use the CRM_REPL_OFF control parameter in the IS Migration Workbench (ISMW) to deactivate any delta replication activity for the business partner contacts you are creating.

You replicate business partner contacts with requests based on filters that refer to the business partner contact number in the BCONT table and an appropriate number range. If you use the CRM_REPL_OFF control parameter you may choose to use a filter based on the BPCT_CRM_GUIDS table and the REPLCNT field as shown in Figure 6-57.

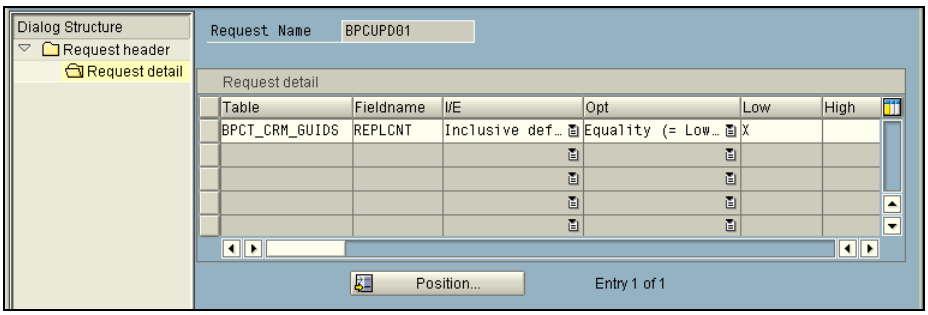


Figure 6-57 Filter to replicate only BP contacts updated with BPCONTACT Migration Object

6.22.5 Contract Account

The problem to identify and replicate only updated contract accounts with an initial download is the same as for a replication of updated business partners.

In the SAP standard Customizing, there is no field available to create a request filter in the CRM system to select and replicate only contract accounts that have been changed on a specific date or by a specific user. Appropriate fields in the FKKVKP table (header table of contract accounts), such as, AENAMP (last user to change object) or AEDATP (date on which the contract account was last changed) are not available as filter fields are missing and need to add them to the SMOFFILFLD table (Allowed fields for Filter definition) in the CRM system first. You maintain the SMOFFILFLD table with transaction SM30 (Maintain Table View). Figure 6-58 shows the definition of the new selection field AEDATP (date on which the contract account was last changed).

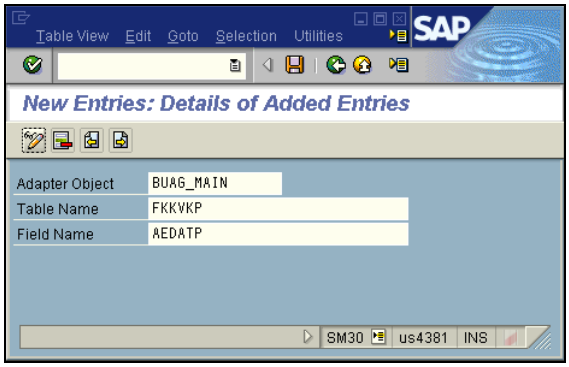


Figure 6-58 Definition of a new Field for a Request

Figure 6-59 shows a request that uses this new field in its filter. With the newly defined filter field you can define a filter selecting only contract accounts that have been updated on a specific date (for example, 1st of March 2009 that is 20090301 in the system internal date format yyyymmdd). Additionally, you can add a number range to allow a concurrent execution of more than one request. No further Customizing changes are required if you add only fields of the FKKVKP table because the extractor on the ERP system has been developed to select any contract accounts only based on filters related to the FKKVKP table.

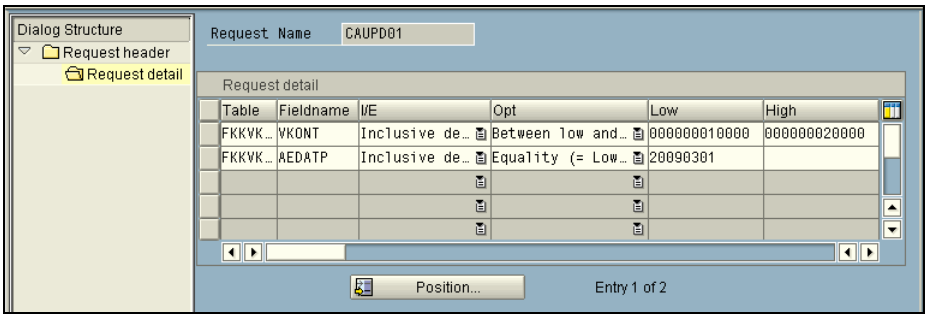


Figure 6-59 Filter of a Request to replicate only updated Contract Accounts

You should check before starting the initial download whether during the replication a new table index based on the filter fields is required on the FKKVKP table on the ERP system to perform the data selection efficiently.

If a selection of the relevant contract accounts based on the VKKVKP table is not sufficient, you need to develop an own extractor module on the ERP system. You can use the standard extractor for contract accounts (CRS_BUAG_MAIN_GET_ID_LIST function module) as a reference. The approach for an implementation of a customer-specific extractor module of contract accounts is the same as for an customer-specific extraction module of business partners except that you must use the BUAG_MAIN adapter object instead of the BUPA_MAIN adapter object. Refer to chapter 6.22.2 *Business Partner* for more details.

6.22.6 Connection Object

Usually, there is no need to migrate changes to connection objects that already exist in the production system. After a go-live, the SAP system becomes the leading system for connection objects. If there is any change to a connection object in the SAP system, you must synchronize the legacy system to reflect the change also in the legacy system.

Therefore, the simplest strategy is to deactivate the delta replication with the CRM_REPL_OFF control parameter of the IS Migration Workbench and perform an initial download with CRM requests and appropriate number ranges for the newly created connection objects. Before starting the data migration, you should reset the number buffer as shown in Figure 6-43 to simplify the determination of the number ranges of only the newly migrated connection objects.

Alternatively, you may choose to use a filter based on the ECRM_TEMP_OBJ table and the XREPLCNT field as shown in Figure 6-60.

The standard extractor of connection objects resets the XREPLCNT field to space immediately after the selection of the connection objects regardless whether during further processing the related BDoc could be created and send or not. Therefore, we advice to test the initial download for a single connection object with a filter based on the related identifier prior to initiate the initial download based on the ECRM_TEMP_OBJ table.

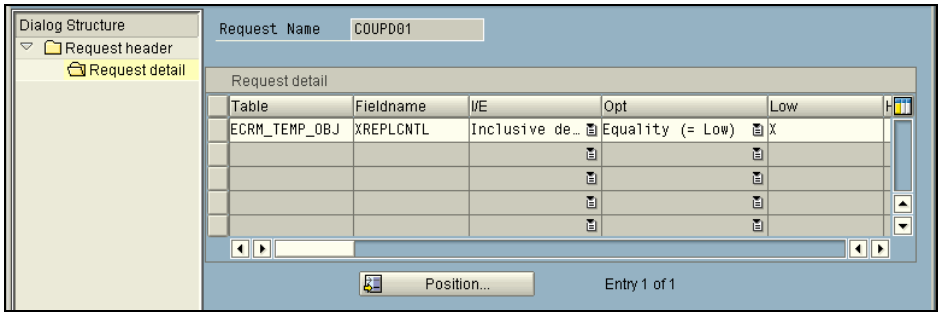


Figure 6-60 Filter to replicate only COs created with CONNOBJ Migration Object

6.22.7 Point of Delivery

During data migration into the ERP system, deregulated thus replication relevant points of delivery (PoDs) are created when migrating a utility installation with the INSTLN migration object. In the ERP system, each PoD is identified by a GUID (Global Unique Identifier) that is calculated using an algorithm defined by the Open Software Foundation and does not lead to consecutive numbers derived from a number range object. This makes it impossible to define a range covering a list of newly created PoDs. This is the reason why by default a request to download PoDs is based on the identifier of connection objects, to which a PoD is linked through its related installation.

This issue is not causing any problems during the first initial download when you have to replicate all PoDs based on all newly migrated connection objects. In this scenario, you can create requests based on the range of identifiers of the related connection objects. This method is feasible, when you migrate new utility installations fully or partially based on already replicated connection objects in a second cutover phase. The question is which of the existing connection objects must be selected to replicate the newly added PoDs into the CRM system? One option is to replicate all PoDs regardless whether they have already been replicated or not. This is an option, if the additional volume of PoDs is manageable. The advantage will be the execution of a simple initial download considering all PoDs based on all existing connection objects. However, what are the options if the volume of newly created PoDs is small compared to the already replicated PoDs and only limited time is available to replicate these PoDs? One solution could have been to create a list of the PoDs you need to replicate and use this list in the filter settings of a request download. The problem is that for a request you cannot create a filter as a list of identifiers. In addition, ranges of identifiers cannot be used either because PoDs are identified by a GUID and not by subsequent numbers as indicated before.

You can use the CRM_REPL_OFF parameter in the IS Migration Workbench (ISMW) when implementing a phased migration and replication strategy. If the CRM_REPL_OFF parameter is set to X, the application saves for each migrated PoD its identifier in the ECRM_TEMP_OBJ table in the ERP system. Figure 6-61 shows the filter settings of a request to replicate PoDs based on the list in the ECRM_TEMP_OBJ table. You can run only one single request with such a filter.

The request download reads identifiers of the PoDs in the ECRM_TEMP_OBJ table and determines the related connection objects. Based on this list of connection objects, the relevant installations and their PoDs are determined and replicated. Due to technical reasons, the request replicates already replicated PoDs again.

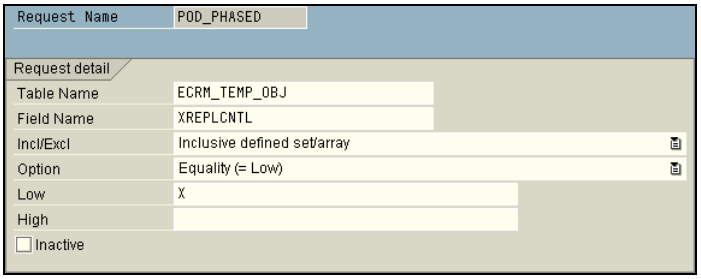


Figure 6-61 Filter Settings for a phased PoD Replication

6.22.8 Contract

After a go-live, the SAP system becomes the leading system for contracts. An update of an SAP contract due to a change in non-SAP system after the go live is unlikely. If there is such a process, an immediate synchronization with an interface is in place that makes an update of a contract in a second data migration phase unnecessary.

Therefore, the simplest strategy is to deactivate the delta replication with the CRM_REPL_OFF control parameter of the IS Migration Workbench and perform an initial download with CRM requests and appropriate number ranges for the newly created contracts. Before starting the data migration, you should reset the number buffer as shown in Figure 6-43 to simplify the determination of the number ranges of the newly migrated contracts.

Alternatively, you may choose to use a filter based on the ECRM_TEMP_OBJ table and the XREPLCNT field as shown in Figure 6-62.

The standard extractor of connection objects resets the XREPLCNT field to space immediately after the selection of the connection objects regardless whether during further processing the related BDoc could be created and send or not. Therefore, SAP advices to test the initial download for a single contract with a filter based on the related identifier prior to initiate the initial download based on the ECRM_TEMP_OBJ table.

As already described in chapter 6.21 *Preparation of Contract Replication (SI_CONTRACT)*, you must create entries in the EVERH table for each migrated contract prior to start the initial download. Figure 6-62 shows the filter settings of a request to replicate contracts based on the XREPLCNTL field in the EVERH table. You can run only one single request with such a filter. It is not possible to use number ranges to define and execute multiple requests concurrently

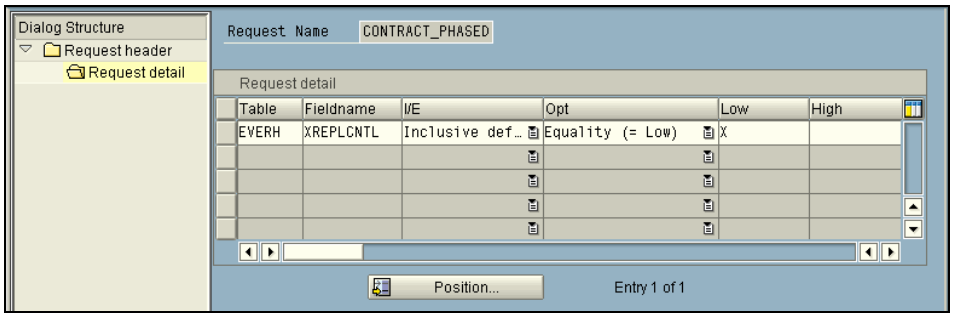


Figure 6-62 Filter Settings for a phased Contract Replication

6.23 Custom fields

6.23.1 Motivation

You can create for all business objects customer-specific fields. If the same fields exist in both ERP and CRM system, the initial download transfers customer-specific fields together with the standard fields automatically from the ERP system into the CRM system and vice versa. Of course, you must implement the enhancements correctly according to the related configuration guides, for example, in using the Easy Enhancement Workbench (EEWB).

Frequently, customer-fields are created and maintained only in the CRM system and must be populated for each replication relevant business object with an appropriate value as part of the overall initial download procedure. The following options are available to finish the initial download with the customer fields populated with their correct values:

Option 1 You perform an initial download disregarding the customer-specific fields with the consequence that the initial download creates the business objects and does not touch the customer-specific fields this leaving these fields with a space or initial value. On completion of the initial download, you update the customer-specific fields with an own report on the CRM system.

For this, you must create an import file that contains the values of the customer-specific fields per business object and its identifier. Your program reads the file, identifies the related business object by the transferred identifier and updates the object in the CRM system with a respective BAPI or any other standard method.

Option 2 During the initial download, you transfer the values of the CRM customer-specific fields in special containers in the BDoc together with the values of the standard fields. On the CRM system, you can transfer the values from the BDoc directly to the customer-specific fields.

For this, you transfer the values of the CRM customer-specific fields in the import file of the corresponding migration object into the load program of the migration. The load program saves the information together with the identifier of the migrated business object in a custom table. With a suitable BAdI implementation on the ERP system, you read the custom table during the initial download and populate the container of the BDoc. With a corresponding BAdI implementation on the CRM system, you populate the customer-specific fields with the information of the BDoc container while processing the BDoc. The implemented BAdIs remain activated only during the initial download.

The main advantage of option 1 is that you separate the initial download and the update of the customer-specific fields into two steps that you can execute successively. However, the main disadvantage of this option is that the replication period is extended significantly because one business object is always touched twice: once during the initial download for a creation and subsequently a second time during its update. The execution of the custom program can take even more time than the initial download because you must read a business object from the database first before updating it and saving the updated version in the database. Your report must also allow parallel processing and a good error handling.

The main advantage of option 2 is that the implementation is simple and usually does not affect significantly the performance of the initial download.

The following chapters show how to implement this approach for various replication relevant business objects.

6.23.2 Easy Enhancement Workbench (EEWB)

The Easy Enhancement Workbench (EEWB) is a development tool with which you can extend SAP applications (called business objects) in a simple manner.

Wizards help you to define Customer enhancements to a Business Object. The Workbench then does all development work for the user; databank tables, screens and application logic are created automatically. Finally, the customer enhancement is included in the SAP standard. This also allows users without ABAP knowledge the simple possibility of extending the SAP standard.

An extension created using the Easy Enhancement Workbench does not differ technically from one created manually. In both cases transportable ABAP objects are created and the same Customer Exits, Business Transaction Events or BAdIs are implemented. The difference lies exclusively in the manner in which the objects required are created. Template objects that are adapted to the extension definition and created by a generator achieve the automation offered by the EEWB. The functionality of the EEWB is therefore only available for specially prepared business objects, mainly from the CRM environment. The type of extension is also predefined. In most cases, you have the possibility to add user-defined tables or fields. Often the extension takes place system-wide. For example, when extending a business object on the CRM system, the data exchange to the connected ERP system can be extended and a new table can also be created in the ERP system. The system landscape must be set up in order to be able to use system-wide generation.


You must regenerate the definition of the BDocs that are related to a particular enhancement after you have created or changed an enhancement. The regeneration can lead to incompatibilities when processing still unprocessed BDoc data with the newly generated BDoc. Therefore, you must ensure that all affected BDocs have been processed successfully before you transport a new enhancement into your production system. This will prevent problems when processing data based on the previous BDoc version.

For more information about the EEWB, see SAP Note 494966.

6.23.3 Custom fields Technical Objects

As of CRM Release 4.0, you can add customer-specific fields to the technical business objects connection object and point of delivery. Currently, the ERP system supports customer-specific fields for connection objects and other master data, but not for points of delivery. If you have fields for the point of delivery, you must ‘reroute’ these to other master data or store them in customer-specific tables. After you have defined the customer-specific set categories in order to save your data in the CRM system you implement the data exchange as follows:

- Implement the `FILL_BAPIMTCS` method of the `IBSSI_SEND_TO_DWN` BAdI for the data exchange of technical objects in the ERP system. The implementation consists out of two parts:
  - Selection of the data from the customer-specific table you have populated during the data migration for the related customer-specific fields on the CRM system
  - Move of the retrieved values of the customer-specific fields into the `BAPIMTCS` CRM Middleware transaction container structure

  The data of the related connection object specific data is available in the `BAPI_EHAU` structure while you find the data of the point of delivery in the `BAPI_EUI` structure.
- Implement the `READ_AND_DELETE_FROM_BAPIMTCS` and `FILL_SETTYPES` method of the `IBSSI_RECEIVE_TO_DWN` BAdI for the data exchange of technical objects in the CRM system.

A sample code is available for all abovementioned BAdIs. To display the sample code choose *Goto* → *Sample Code* → *Display* in transaction *SE18*. When creating the

implementation, use the FLT_VAL parameter to differentiate between connection objects ('0101') and point of deliveries ('0102').

For more information on customer-specific fields of technical business objects in SAP CRM, see the document ***Replication of Technical Objects (IS-U/CRM)*** in the SAP for Utilities space of the ***SAP Service Marketplace*** (http://service.sap.com/). Choose SAP for Utilities – Product Information → SAP CRM for Utilities → Cookbook & Guidelines.

You can use the COM_BUPA_SET_TO_BAPIMTCS function module in the ERP system to move the customer-specific data into the BAPIMTCS structure. On the CRM system, you can use the BUPA_BAPIMTCS_TO_SET function module to unpack the data from the BAPIMTCS container. This is possible because the DDIC definition of the BAPIMTCS structure is the same for a replication of technical objects and business partners.

6.23.4 Custom fields Contract

Many projects enhance the contract on the CRM system by customer-specific fields and tables by means of the Easy Enhancement Workbench (EEWB). The CRM_DOWNLOAD_FILL_DATA_NEW method of the ECRM_CRM_DOWNLOAD BAdI on the ERP system allows you to add customer-specific values into the TXY_CONTRACTCONF data container of the SI_CONTRACT BDoc for a replication of customer-specific fields from the ERP system into the CRM system. Figure 6-63 shows an implementation of the CRM_DOWNLOAD_FILL_DATA_NEW method you can use as a basis for an own development to select and map customer-specific data from a customer-specific table into the TXY_CONTRACTCONF data container. In the example, the customer-specific table was filled during data migration of contracts with the MOVE_IN migration object.

```
METHOD if_ex_ecrm_crm_download~crm_download_fill_data_new.

  DATA : wa_contractconf TYPE ecrm_crmcontractconf,
         wa_contractpos TYPE ecrm_crmcontractpos,
         wa_cust_data TYPE zmig_si_contract.

* Initialisation
  CLEAR: wa_contractconf.

* select data from the custom table
  SELECT SINGLE * FROM zmig_si_contract INTO wa_cust_data
                 WHERE vertrag = x_contractpos-ever-vertrag.
  IF sy-subrc = 0.

*   add customer-specific field ZCRM_FIELD_01 to data container
    wa_contractconf-ref_guid   = x_contractpos-everh-contractpos.
    wa_contractconf-ref_kind   = 'B'.
    wa_contractconf-attr_name  = 'ZPARA_NAME_ZCRMFIELD01'.
    wa_contractconf-kind       = 'E'.
    wa_contractconf-elemlength = '10'.
    wa_contractconf-type       = 'C'.
    wa_contractconf-value     = wa_cust_data-zcrm_field_01.
    APPEND wa_contractconf TO txy_contractconf.

*   add customer-specific field ZCRM_FIELD_02 to data container
    wa_contractconf-ref_guid   = x_contractpos-everh-contractpos.
    wa_contractconf-ref_kind   = 'B'.
    wa_contractconf-attr_name  = 'ZPARA_NAME_ZCRMFIELD02'.
    wa_contractconf-kind       = 'E'.
    wa_contractconf-elemlength = '10'.
    wa_contractconf-type       = 'C'.
    wa_contractconf-value     = wa_cust_data-zcrm_field_02.
    APPEND wa_contractconf TO txy_contractconf.

  ENDIF.
```



```
ENDMETHOD.
```

Figure 6-63 Implementation to map Customer-specific Data in SI_CONTRACT BDoc

On the CRM system, you must maintain a configuration that allows an automatic mapping of the values in the TXY_CONTRACTCONF data container to the customer-specific fields you defined with the EEWB. To access the configuration in the SAP Customizing Implementation Guide (IMG) choose *Customer Relationship Management → Industry-Specific Solutions → Utility Industries → Transactions → Settings for Contract Replication → Maintain Customer Enhancements*.

You can access the utilities specific IMG path directly with the transaction CRM_ISU_IMG.

To allocate a new field, proceed as follows:

1. Choose the ***New Entries*** button
2. Give the new fields a name in the ***Parameter Name*** field. The attribute names (WA_CONTRACTCONF-ATTR_NAME field) you use in your implementation of the CRM_DOWNLOAD_FILL_DATA_NEW method must be the same as the parameter names you define in this step.
3. Enter a descriptive text In the ***Text*** field
4. In the ***Business Transaction Components*** field, select the business transaction component of the relevant set type.
5. Enter the name of the customer-specific field in the ***Field Name*** field.

Once you have made these settings, the new field is functional. You can make the further settings for each field, such as, defining the visibility of the field on the screen. After you saved the configuration, the standard application will recognize the field automatically during the processing of the SI_CONTRACT BDoc in the CRM system.

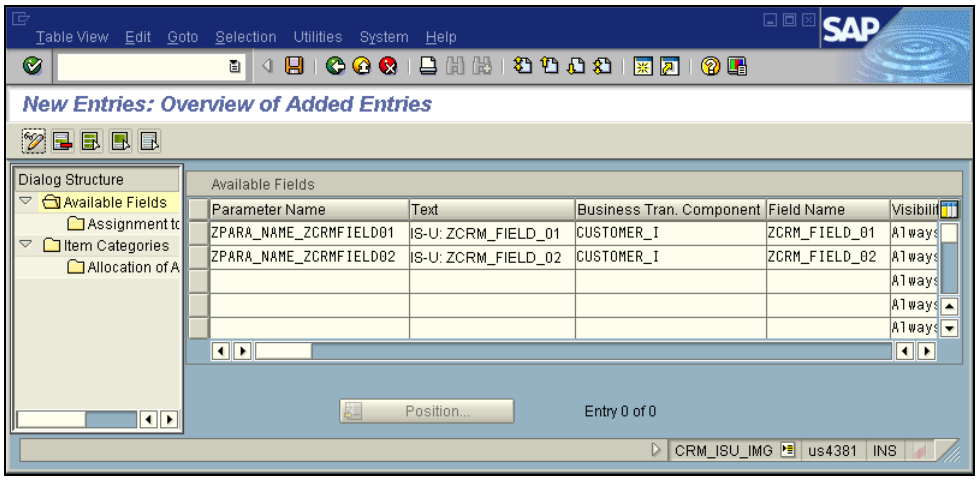


Figure 6-64 Allocation of New Fields in Customizing for CRM

6.23.5 Custom fields Business Partner

There are two corresponding function modules available on both the ERP and CRM system to pack customer-specific into a BDoc of the BUPA_MAIN adapter object and to unpack the data in the receiving system:

- ERP system: COM_BUPA_SET_TO_BAPIMTCS and COM_BUPA_ BAPIMTCS_TO_SET
- CRM system: BUPA_SET_TO_BAPIMTCS and BUPA_BAPIMTCS_TO_SET

Despite their different names, the function of the function modules in both systems is the same. The (COM_)BUPA_SET_TO_BAPIMTCS function modules packs the customer-specific data any data structure into the BAPIMTCS CRM Middleware transaction container structure of the BUPA_MAIN BDoc while the (COM_)BUPA_BAPIMTCS_TO_SET unpacks the data from the data container and moves it to the given data structure.

During an initial download, you transfer your customer-specific data in the BAPIMTCS data container on the ERP system. For this you need to develop an own function module. Figure 6-65 shows a function module that you can use as a basis for an own development to select and map customer-specific data from a customer-specific table into the BAPIMTCS data container. In the example, the customer-specific table was filled during data migration of business partners with the PARTNER migration object.

```
FUNCTION ZCRM_BUPA_COMM_TO_BAPIMTCS
*"----------------------------------------------------------------------
*"*"Local Interface:
*"  IMPORTING
*"     REFERENCE(IS_BP_EXTERN) TYPE  BUS_EI_COM_EXTERN
*"  CHANGING
*"     REFERENCE(CT_BAPIMTCS) TYPE  COM_BUPA_BAPIMTCS_T
*"     REFERENCE(CV_NEXT_FREE_CURRDNO) TYPE  CURRDNO DEFAULT 1
*"----------------------------------------------------------------------
*
* The ZCUSTOMER_BUPA is a customer-specific table with the
* following fields:
* MANDT        Client                    Key field
* PARTNER      BP number                 Key field
* ATNAME       Characteristic Name       Key field
* ATVALUE      Characteristic Value
* KLASSE       Class number
*"----------------------------------------------------------------------

* local data definition
  DATA: BEGIN OF lt_zcustomer_bupa,
           lt_bupa_data TYPE TABLE OF zcustomer_bupa,
        END OF lt_zcustomer_bupa.
  DATA: lv_partner TYPE gc_partner,
          lv_partner_guid TYPE gc_partneg.

* Initialization
  lv_partner = is_bp_extern-header-object_instance-bpartner.
  lv_partner_guid = is_bp_extern-header-object_instance-bpartnerguid.

* select data from the custom table
  SELECT * FROM zcustomer_bupa
           INTO TABLE lt_zcustomer_bupa-lt_bupa_data
           WHERE partner = lv_partner.

* map data into the BAPIMTCS container
  IF sy-subrc = 0.
    CALL FUNCTION 'COM_BUPA_SET_TO_BAPIMTCS'
      EXPORTING
        iv_partner_guid = lv_partner_guid
        iv_application  = 'ZCUSTOMER_BUPA'
      CHANGING
        ct_bapimtcs     = ct_bapimtcs
        cv_currdno      = cv_next_free_currdno
        cs_set          = lt_zcustomer_bupa.
  ENDIF.

ENDFUNCTION.
```

Figure 6-65 Implementation to map Customer-specific Data in BUPA_MAIN BDoc

You must register your function module at ***Time*** *Mapping of Complex Structure to BAPIMTCS Structure (TOMTC)* on the ERP system with transaction COM_BUPA_CALL_FU (Determining Function Modules for Data Exchange). To support still requests with a filter based on the BUT000 table, you should add also the standard extractor (CRS_BUPA_MAIN_GET_ID_LIST function module). Figure 6-52 shows the registered function module.

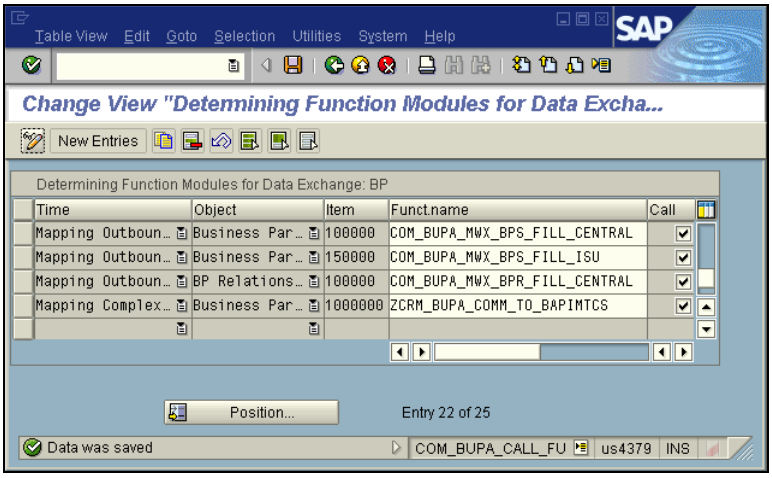


Figure 6-66 Custom Data Mapper in COM_BUPA_CALL_FU Transaction

On the CRM system, you must develop a complementary function module that unpacks the data from the BAPIMTCS container and moves the unpacked data to the related fields of the CRM business partner. Figure 6-67 shows a function module that can be used as a basis for an own development retrieve the transferred customer-specific data from the BAPIMTCS data container.

```
FUNCTION ZCRM_BUPA_BAPIMTCS_TO_COMM.
*"----------------------------------------------------------------------
*"*"Local Interface:
*"  CHANGING
*"     REFERENCE(CT_BAPIMTCS) TYPE  CRMT_BUPA_BAPIMTCS_T
*"     REFERENCE(CS_BP_EXTERN) TYPE  BUS_EI_EXTERN
*"     REFERENCE(CT_ERROR_TABLE) TYPE  BUS_BAPIRET2_T
*"----------------------------------------------------------------------

* type definition
  TYPES: BEGIN OF ty_zcustomer_bupa,
           mandt   TYPE mandt,
           partner TYPE partner,
           atname  TYPE atnam,
           atvalue TYPE atwrt,
           class   TYPE ,
         END OF ty_zcustomer_bupa.

* local data definition
  DATA: BEGIN OF lt_zcustomer_bupa,
           lt_bupa_data TYPE TABLE OF ty_zcustomer_bupa,
        END OF lt_zcustomer_bupa.
  DATA: lv_partner TYPE gc_partner,
        lv_partner_guid TYPE gc_partneg,
        wa_zcustomer_bupa TYPE ty_zcustomer_bupa.

* Initialization
  lv_partner = cs_bp_extern-header-object_instance-bpartner.
  lv_partner_guid = cs_bp_extern-header-object_instance-bpartnerguid.

* map data into the local structure
  CALL FUNCTION 'BUPA_BAPIMTCS_TO_SET'
```



```
    EXPORTING
      iv_partner_guid = lv_partner_guid
      iv_application  = 'ZCUSTOMER_BUPA'
    CHANGING
      ct_bapimtcs     = ct_bapimtcs
      cs_set          = lt_zcustomer_bupa.

* move the customer data into the CRM business partner
  LOOP AT lt_zcustomer_bupa-lt_bupa_data INTO wa_zcustomer_bupa.
*   move data to the customer fields in the CS_BP_EXTERN structure
  ENDLOOP.

ENDFUNCTION.
```

Figure 6-67 Implementation to map Customer-specific Data in BUPA_MAIN BDoc

You must register your function module at ***Time*** *Mapping of BAPIMTCS structure to Complex Structure (MTCTO)* on the CRM system with transaction CRMC_BUT_CALL_FU (Determining Function Modules for Data Exchange).

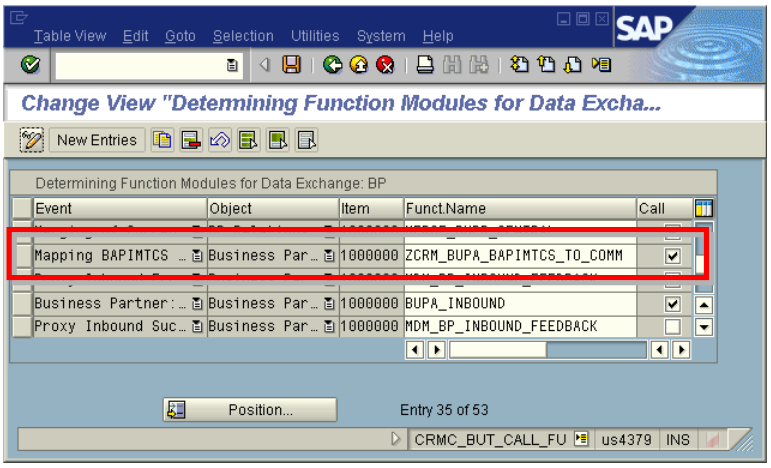


Figure 6-68 Custom Data Mapper in CRMC_BUT_CALL_FU Transaction

For more information how implement enhancements of the business partner data exchange, see SAP Note 774662. For more information and examples how to implement a both way data exchange, see SAP Notes 877668. You can also use the implementation of the CRM_BUPA_FINSERV_TO_BAPIMTCS and function module and the CRM_BUPA_BAPIMTCS_TO_FINSERV function module in both the ERP and CRM system as another example. For more details, see SAP Note 873771.

6.23.6 Custom fields Business Partner – Use Case Marketing Attributes

The previous chapter describes how to transfer customer-specific information from the ERP system into the CRM system using the BUPA_MAIN BDocs. A typical use case is the migration of the marketing attributes of a business partner. If the legacy system holds the marketing attributes of a business partner, you can migrate them into the ERP system into a temporary customer-specific table using the IS Migration Workbench and replicate them during an initial download into the CRM system together with the general data of a business partner. You can now create the marketing attributes during the BDoc processing with an implementation based on SAP Note 609236. The code snippet in Figure 6-69 can be used as a basis for an own implementation.

```
FUNCTION ZCRM_MKTBP_CRMOU_WRITE.
*"----------------------------------------------------------------------
*"*"Local Interface:
*"  IMPORTING
*"     REFERENCE(IV_X_SAVE) TYPE  C DEFAULT 'X'
*"  EXPORTING
```



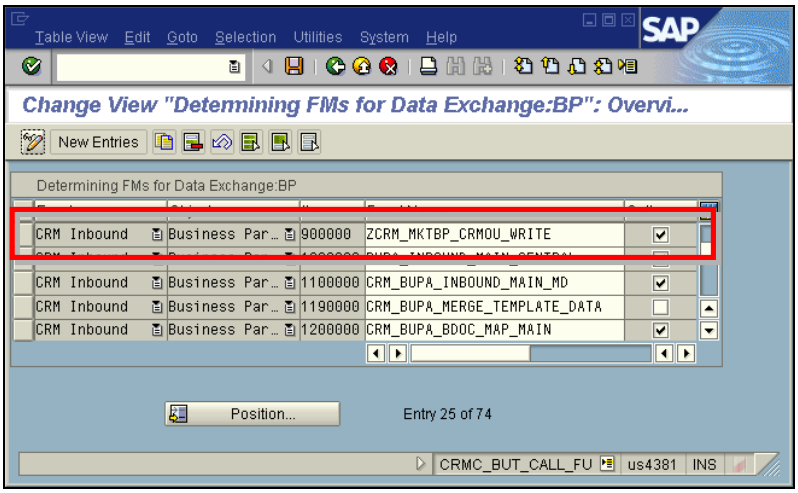
```
*"     REFERENCE(STATUS) TYPE  SMWVALSTAT
*"  CHANGING
*"     REFERENCE(C_BP_STRUCT) TYPE  BUS_EI_EXTERN
*"     REFERENCE(ERROR_SEGMENTS) TYPE  SMW_ERRTAB
*"----------------------------------------------------------------------

* local data definition
  DATA: lt_bp_alloclist TYPE crmt_mktbp_alloc_bp_tab,
        wa_bp_alloclist TYPE LINE OF crmt_mktbp_alloc_bp_tab,
        wa_allocvalues TYPE LINE OF crmt_mktpfchr_attribute_list.
  DATA: lv_partner TYPE gc_partner,
        lv_partner_guid TYPE gc_partneg,
        wa_zcustomer_bupa TYPE ty_zcustomer_bupa.
  DATA: lt_return   TYPE TABLE OF bapiret2,
        ls_return   TYPE bapiret2.

* initialization
  lv_partner      = c_bp_struct-header-object_instance-bpartner.
  lv_partner_guid = c_bp_struct-header-object_instance-bpartnerguid.

* mapping transferred attributes to API parameter
* mapping transferred attributes to API parameter
  LOOP AT gt_zcustomer_bupa-lt_bupa_data INTO wa_zcustomer_bupa
                                         WHERE partner = lv_partner.
    wa_bp_alloclist-partner_guid = lv_partner_guid.
    wa_bp_alloclist-profile_template_id = wa_zcustomer_bupa-class.
    wa_allocvalues-atname = wa_zcustomer_bupa-atname.
    wa_allocvalues-atvalue = wa_zcustomer_bupa-atvalue.
    APPEND wa_allocvalues TO wa_bp_alloclist-allocvalues.
    APPEND wa_bp_alloclist TO lt_bp_alloclist.
  ENDLOOP.

* call API to create marketing attributes
  CALL FUNCTION 'CRM_MKTBP_CREATE_LIST_BPS_OB'
* EXPORTING
*   IV_AUTHORITY_CHECK       = ' '
*   IV_ACTIVITY              = '03'
    TABLES
      it_bp_alloclist          = lt_bp_alloclist
      et_return                = lt_return.

* error handling
  status = 'S'.
  LOOP AT lt_return INTO ls_return.
    CALL FUNCTION 'BUPA_MWX_BDOC_BUILD_ERR_MESS'
      EXPORTING
        iv_obj_id   = c_bp_struct-header-object_instance-bpartnerguid
        is_bapiret2 = ls_return
      IMPORTING
        et_err_msg  = error_segments.
    IF ls_return-type CA 'XEA'.
      status = 'A'.
    ENDIF.
  ENDLOOP.

ENDFUNCTION.


Top-Include of the function group:
FUNCTION-POOL ZCRM_MKTBP.

* type definition
TYPES: BEGIN OF ty_zcustomer_bupa,
          mandt   TYPE mandt,
          partner TYPE partner,
          atname  TYPE atnam,
          atvalue TYPE atwrt,
          class   TYPE klasse_d,
```



```
        END OF ty_zcustomer_bupa.

* local data definition
DATA: BEGIN OF gt_zcustomer_bupa,
         lt_bupa_data TYPE TABLE OF ty_zcustomer_bupa,
      END OF gt_zcustomer_bupa.
```

Figure 6-69 Implementation to process Marketing Attributes

You must register your function module at ***Time*** *CRM Inbound (CRMIN)* on the CRM system with transaction CRMC_BUT_CALL_FU (Determining Function Modules for Data Exchange).



In addition, you need to adjust the ZCRM_BUPA_BAPIMTCS_TO_COMM function module in Figure 6-67 to save the received information in the global memory of the function group. From there the ZCRM_MKTBP_CRMOU_WRITE function module can pick it up during the further processing of the business partner and process the marketing attributes.

```
FUNCTION ZCRM_BUPA_BAPIMTCS_TO_COMM.
*"----------------------------------------------------------------------
*"*"Local Interface:
*"  CHANGING
*"     REFERENCE(CT_BAPIMTCS) TYPE  CRMT_BUPA_BAPIMTCS_T
*"     REFERENCE(CS_BP_EXTERN) TYPE  BUS_EI_EXTERN
*"     REFERENCE(CT_ERROR_TABLE) TYPE  BUS_BAPIRET2_T
*"----------------------------------------------------------------------

* type definition
  TYPES: BEGIN OF ty_zcustomer_bupa,
           mandt   TYPE mandt,
           partner TYPE partner,
           atname  TYPE atnam,
           atvalue TYPE atwrt,
           class   TYPE klasse_d,
         END OF ty_zcustomer_bupa.

* local data definition
  DATA: BEGIN OF lt_zcustomer_bupa,
           lt_bupa_data TYPE TABLE OF ty_zcustomer_bupa,
        END OF lt_zcustomer_bupa.
  DATA: lv_partner TYPE gc_partner,
        lv_partner_guid TYPE gc_partneg,
        wa_zcustomer_bupa TYPE ty_zcustomer_bupa.

* Initialization
  lv_partner = cs_bp_extern-header-object_instance-bpartner.
  lv_partner_guid = cs_bp_extern-header-object_instance-bpartnerguid.

* map data into the local structure
```



```
  CALL FUNCTION 'BUPA_BAPIMTCS_TO_SET'
    EXPORTING
      iv_partner_guid = lv_partner_guid
      iv_application  = 'ZCUSTOMER_BUPA'
    CHANGING
      ct_bapimtcs     = ct_bapimtcs
      cs_set          = lt_zcustomer_bupa.

* move the customer data into the global memory
*  APPEND lines of lt_zcustomer_bupa-lt_bupa_data TO
*                  gt_zcustomer_bupa-lt_bupa_data.

ENDFUNCTION.
```

Figure 6-70 Implementation to unpack Marketing Attributes from BUPA_MAIN BDoc

7 Repository Tables

7.1 Tables ERP System

Table	Description
CRMKEY	Contains all R/3 keys and corresponding keys on the CRM Server
CRMOBJECT	Corresponds to SMOFOBJECT on the CRM Server
CRMOBJTAB	Corresponds to SMOFTABLES on the CRM Server
CRMFILTAB	Corresponds to SMOFFILTAB on the CRM Server
CRMPAROLTP	This parameter file contains control parameters. The RFC-relevant parameters have been transferred to CRMRFCPAR
CRMQNAMES	Table for defining the names of queues
CRMCONSUM	Contains the applications (consumers) that receive data from the R/3 backend
CRMRFCPAR	Used to define the target systems

7.1.1 Important Parameters in CRMPAROLTP Table

Table Maintenance
You maintain the CRMPAROLTP table with transaction SM30 (Maintain Table View)

CRM_MAX_NO_QUEUES_PER_REQUEST
The parameter CRM_MAX_NO_QUEUES_PER_REQUEST controls the parallel processing of the LUWs in concurrent inbound queues per request. You can execute the data selection in the source system in parallel by defining and starting several requests. You can adjust the level of parallel processing also online.

Field	Parameter Value
Parameter Name	CRM_MAX_NO_QUEUES_PER_REQUEST
Param. Name 2	Name of adapter object (for example BUPA_MAIN)
Param. Name 3	Maximum number of parallel processes of initial load type
User	CRM
Param.Value	Maximum number of parallel inbound queues
Param. Value 2	

CRM_FILTERING_ACTIVE
The parameter CRM_FILTERING_ACTIVE controls the filtering of business objects during a request or delta load. The filter has been set with transaction R3AC1 (Adapter Object Overview).

Field	Parameter Value
Parameter Name	CRM_FILTERING_ACTIVE
Param. Name 2	Name of object class (for example, BUPA)
Param. Name 3	Name of adapter object (for example, BUPA_MAIN, BUPA_REL, BUAG_MAIN)
User	CRM
Param.Value	Activation (for example, X to activate filtering)

Param. Value 2	

7.1.2 Important Parameters in CRMRFCPAR Table

Table Maintenance
You maintain the CRMRFCPAR table with transaction SM30 (Maintain Table View)

Parameters

The CRMRFCPAR table controls which data is sent to which consumer. As a result, the data can be sent to recipients that are configured differently (it can be sent to receiving systems, such as, the Internet Pricing and Configurator (IPC) as well as the CRM server).

When you maintain this table, you can configure the loading behavior of defined objects and destinations precisely (transaction SM59).

The decision whether an entry in CRMRFCPAR table is relevant for the current data exchange is made for each client based on the ***Consumer***, ***Object Name***, and ***Download*** fields and the discard data indicator. The other fields are not relevant for your decision whether the entry in the CRMRFCPAR table should be used.

Entries in CRMRFCPAR table are always evaluated separately for each RFC connection that is entered. This means an entry for a specific object name (for example, CUSTOMER_MAIN) overrides only a generic entry (object name "*") if the same RFC connection is specified for both entries. This also applies if the same consumer is specified for different RFC connections. When you maintain the table entries, you should avoid specifying different RFC connections for a consumer, even if this is possible. Otherwise, for example, the RFC connection to the system to which the data should be sent, cannot be clearly identified during the initial load.

Field	Meaning
CONSUMER	Consumer: Receiving application, for example CRM, IPC or SAP Enterprise Buyer (EBP). The english term that is normally used is 'Consumer'. The consumer is transferred via the CRMCONSUM table. The CRM consumer (if this exists) is transferred and activated automatically. The relevant documentation describes how to maintain other consumers (for example, IPC) or you must find this out from the SAP component that is responsible.
OBJNAME	Object Name: Name of the adapter object, for example, BUPA_MAIN. This field often contains the wildcard character "*". As a result, this entry is valid for all objects of the specified consumer.
RFCDEST	RFC Destination: RFC destination of the target system (as defined in transaction SM59)
DOWNLOAD	Download: Load types for which this table entry should be valid. These are described with a single character. The system differentiates between D (delta load), I (initial load, synchronization load), R (request load), E (Error messages regarding data loads for the back end system), O (Other error messages (O) (for example, error messages from an initial load) or * (all load types except for E and O). An error that occurs frequently is that entries are generated for the initial load and/or delta load, but not for the error messages.
RFC_QUEUE	Queue name: If you want to use a predefined queue name instead of the ERP outbound queue name that is usually determined, enter the queue name in this field. If this field contains an entry, the queue name is no longer determined in the "normal" way. This field is normally blank and you should fill it only after consulting SAP because an entry affects how the data is processed and it hinders SAP support. Do not enter a wildcard character "*" under any circumstances.
RFC_IN_QUE	Inbound queue: Again, do not enter a wildcard "*" character in this field. Specify

	the queue name in this field if a certain queue name is to be used for the inbound queue. If there is an entry in this field, the queue name is no longer determined in the "normal" way. Therefore, this field is normally blank.
BAPINAME	BAPI name: If a receiver module is to be called in the target system that is different from the standard module BAPI_CRM_SAVE, enter the name of the module in this field.
INFO	Information: An explanatory short text for this table entry.
INACTIVE	Definition inactive: If you want this table entry to be ignored, set this indicator to 'X'. It is used for the internal evaluation of the CRMRFCPAR table.
DISCARDDAT	Discard data indicator: If this indicator is set to 'X', the data is rejected before it is written to the outbound queue of the R/3 system. This means that destinations that meet the conditions of this entry do not receive any data. This method can be used to prevent the data distribution for an object or entire system. This lock entry always overrides "regular" entries (it always has priority). See SAP Note 482444 for details, before you set this indicator. Also note that some business objects always expect a reply from the ERP system after a data change on the CRM server. Therefore, do not set the "Discard data" indicator in combination with delta load "D" for these objects.
USE_IN_Q	Inqueue indicator: If this indicator is set, the inbound queue in the target system is used. This is the recommended default behavior if a BBPCRM Server is connected. If the indicator is not set, the data is stored in the outbound queue in the back-end system and a Remote Function Call (RFC) is used to call the module stored in the ***BAPINAME*** field directly from there.
SEND_XML	Send data in XML format: The value "M" is used for an adapter-specific transfer mode. Like the XML transfer, it can be used for mixed system landscapes, but it is much faster than the XML transfer. "M" is therefore the recommended setting. If this parameter is set to "X", the system transfers the data in the adapter in the (adapter-specific) XML format (this does not correspond to the general XML standard). Regardless of the format used for the data transfer within the adapter, the data in the target system is mapped back to the original format that was transferred from the sender before the data is available for further processing. Choosing the correct format can improve the performance or prevent transfer errors, however, it does not influence the application functions.
HOLD_DATA	Obsolete entry (for more details, see SAP Note 443898).
CRM_REL	CRM release: Contains the CRM version of the target system if this is a CRM/EBP server. If this is not a CRM system or if the release is lower than CRM 3.0, this field remains blank.
LAST_UPD	Last update: Contains the date on which the CRM_REL field was last checked and updated.

The ***Discard data*** and ***Data record inactive*** fields are often mixed up. The effects of such a mix-up can be serious. When the ***Data record inactive*** indicator is set, the CRMRFCPAR table entry is ignored. However, when the ***Discard data*** indicator is set, the corresponding RFC destination cannot contain any data which meets the conditions of the particular entry in the CRMRFCPAR table - other entries are overridden (the target system will not receive data, even if there are another suitable entries in the CRMRFCPAR table).

7.2 Tables CRM System

Table	Description
SMOFDSTAT	Defines the Initial Load Status for each download object. Values are: Waiting, Running, Started, Comparing, Done, Abort. The status is set automatically.
SMOFEVOBJ	Contains the list of all events supported by the CRM Server
SMOFFILFLD	Contains the fields that are allowed to be filtered
SMOFFILSAP	Contains the filter conditions predefined by SAP that are included in table SMOFFILTAB
SMOFFILTAB	Defines filter conditions for each table for the initial download and delta download.
SMOFINICON	Controls the flow contexts, source and target of initial data transfers
SMOFINICUS	Controls the repeated data transfer
SMOFMAPTAB	The reference table between CRM server table names and R/3 backend table names.
SMOFOBJCLA	Contains the object classes
SMOFOBJECT	Main table of the Adapter Repository. It contains all the objects relevant for the data transfer. A reference relationship exists to the transaction repository via TRANSID.
SMOFOBJPAR	Contains the parent objects of objects
SMOFOBJCTT	Contains long texts for SMOFOBJECT.
SMOFPARSFA	Standard control table for parameters
SMOFQFIND	Control table to determine queues
SMOFREQTAB	Contains filter conditions for data requests from the R/3 backend
SMOFRELTAB	Defines special relationships between tables
SMOFSUBTAB	Contains subscribed mapping functions
SMOFTABLES	Defines the list of tables belonging to a given Adapter object
SMOFUPLMAP	Mapping function modules for transferring data from CRM to a R/3 backend

7.2.1 Important Parameters in SMOFPARSFA Table

Table Maintenance

You maintain the SMOFPARSFA table with transaction SM30 (Maintain Table View)

MAX_PARALLEL_PROCESSES

The parameter MAX_PARALLEL_PROCESSES controls the number of parallel processes of an initial load type that can run in parallel. Processes of initial load type are Initial Load, Request Download and Data Integrity Manager (DIMa) instances. If you want to start more than one request at the same time, you need to define this parameter accordingly.

Field	Parameter Value
PARSFAKEY	R3A_COMMON
PARNAME	MAX_PARALLEL_PROCESSES
PARVAL1	Maximum number of parallel processes of initial load type

MW_NO_INMBDOC_OBJ_LINKS, MW_NO_OUTMBDOC_OBJ_LINKS

The MW_NO_INMBDOC_OBJ_LINKS and MW_NO_OUTMBDOC_OBJ_LINKS parameters deactivate the creation of a link between an mBDoc and the related object.

Field	Parameter Value
PARSFAKEY	FLOW
PARNAME	MW_NO_INMBDOC_OBJ_LINKS
PARVAL1	X

Field	Parameter Value
PARSFAKEY	FLOW
PARNAME	MW_NO_OUTMBDOC_OBJ_LINKS
PARVAL1	X

MW_NO_INMBDOC_OBJ_LINKS, MW_NO_OUTMBDOC_OBJ_LINKS

The MOBBRIDGE_NO_LINKS parameter deactivates the creation of a link between an mBDoc and an outbound sBDoc, if there is an active mobile bridge.

Field	Parameter Value
PARSFAKEY	FLOW
PARNAME	MOBBRIDGE_NO_LINKS
PARVAL1	X

MOBBRIDGE_NO_LINKS

The MOBBRIDGE_NO_LINKS parameter deactivates the creation of a link between an outbound mBDoc and an outbound sBDoc, if there is an active mobile bridge.

Field	Parameter Value
PARSFAKEY	FLOW
PARNAME	MOBBRIDGE_NO_LINKS
PARVAL1	X

MW_NO_MBDOC_LINKS

The MW_NO_MBDOCS_LINKS parameter deactivates the creation of a link between an inbound mBDoc and an inbound sBDoc, if there is an active mobile bridge.

Field	Parameter Value
PARSFAKEY	FLOW
PARNAME	MW_NO_MBDOC_LINKS
PARVAL1	X

8 Important SAP Notes

8.1 General Notes

Note	Description
721844	IS-U CRM Integration - Performance Tips & Tricks
721762	FAQ: Replication of technical objects
715528	IS-U CRM integration - initial download of SI_POD
594953	Initial download performance IS-U/CCS - mySAP CRM
702735	Deactivation of IPC in CRM
429423	CRM Release: General analysis of the initial Load
770849	Generating requests automatically from list

8.2 Notes related to Requests

Note	Description
788822	Filtering does not work
544001	Entry of filter conditions for adapter objects
510192	Description and maintenance of table CRMRFCPAR
426159	Adapter: Running requests in parallel
635697	Activating/deactivating creation of BDoc messages
788871	Adapter specific entries in table SMOFPARSFA

8.3 Notes related to CRM Middleware

Note	Description
1270396	ERP Outbound: Parallelization of confirmations from CRM
779664	Consistency check of qRFC queues with deletion
765236	FAQ: Queueing in CRM and R/3 (queue names)
769113	Delete unprocessed LUWs in the qRFC
713173	Update of the CRM Middleware reorganization
527481	tRFC or qRFC calls are not processed
429423	CRM Release: General analysis of the initial Load
375566	Large number of entries in tRFC and qRFC tables
366869	HOLD/EXECUTED/WCONFIRM entries in ARFCRSTATE
366807	Significance of background jobs ARFC
356228	qRFC occupies all work processes on OLTP or the CRM system
350176	CRM/EBP: Performance improvement during exchange of data
324545	tRFC/qRFC: Table ARFCRSTATE increasing fast
872533	FAQ: CRM Middleware
763680	CSA* qRFC queues occupy all work processes on CRM

431345	CRM Server: Analysis data exchange CRM->R/3
483301	Suppressing leading zeros in queue name
1095263	Parameter: REMOVE_LEADING_ZEROS is object-dependent
426159	Adapter: Running requests in parallel
356228	qRFC occupies all work processes on the R/3 backend or the CRM system
369007	qRFC: Configuration for the QIN Scheduler
390592	qRFC Monitoring

8.4 Notes related to Debugging

Note	Description
656823	FAQ: Tips and tricks for transfer debugging

8.5 Notes related to important Corrections

Note	Description
1051445	qRFC scheduler does not use all available resourcen
936257	Initial/synchronization load starts repeatedly
929078	Initial load remains in running status
844758	Request with error stays in the status Running
805918	Parallel processing request during load into an R/3
707820	Performance: Object links and RSRLDREL
1393664	ISUCRM: Grouping for same header does not work
1381356	ISUCRM: Item grouping not work in initial and request load
1325334	Grouping of contract into single header not necessary in ERP

8.6 Notes related to Database Maintenance

Note	Description
206439	Reorganization of tables in CRM Middleware
536414	Reorganization of BDoc tables, trace and log tables
801474	Reorganization of Table BDOC_TRACK MONBDOCT (Display Program for Table BDOC_TRACK) RBDOCTBL (Reorganize BDOC Tracking Table)
706478	Preventing Basis tables from increasing considerably
350176	Performance Improvement during Initial Download
932975	Oracle statistics for RFC tables
1020260	Delivering Oracle statistics
3809	Oracle: Changing the size of the redo log files
619188	FAQ: Oracle wait events
863417	FAQ: Oracle Database Archive modes and redo logs
8438	Orcacle: Activating/deactivating ARCHIVELOG mode

8.7 Notes related to Optimization of Storage Subsystem

Note	Description
793113	FAQ: Oracle I/O-Konfiguration
912905	FAQ: Storage systems used with MaxDB
987961	FAQ: SQL Server 2005 E/A-Performance

8.8 Notes related to Client and System Copy

Note	Description
897804	CRM Client Copy 5.0: Additional information
765018	Problems with logical system during data exchange (after system copy)
564435	Additional information on CRM System copy
757093	FAQ: Analyzing CRM Client Copy problems
577778	Postprocessing after client copy
379436	CRM link to R/3 OLTP: Change of the R/3 OLTP system

