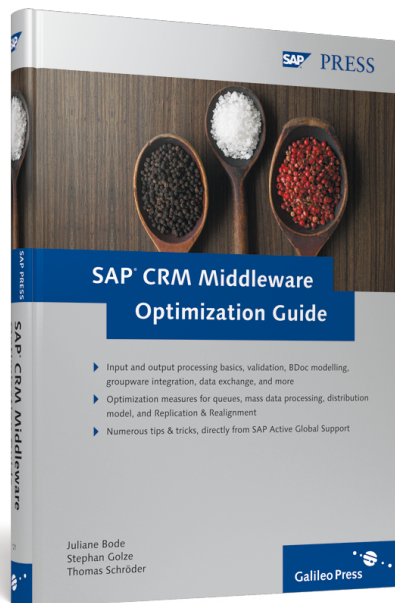


Juliane Bode, Stephan Golze, Thomas Schröder

SAP® CRM Middleware Optimization Guide



Contents at a Glance

| | | |
|-----------|---|------------|
| 1 | A Journey Through CRM Middleware | 25 |
| 2 | Inbound Processing and Validation | 37 |
| 3 | Outbound Processing | 89 |
| 4 | BDoc Modeling and Storage | 135 |
| 5 | Groupware Integration | 155 |
| 6 | Exchanging Customer-Specific Data | 169 |
| 7 | Introduction to Performance Optimization . | 187 |
| 8 | Inbound Queues | 197 |
| 9 | Replication Model and R&R | 235 |
| 10 | Outbound Queues | 269 |
| 11 | XML in Optimization | 297 |
| 12 | Reorganization | 307 |
| 13 | Performing Optimized Mass Changes | 317 |
| 14 | A Look Ahead | 359 |
| A | SAP CRM Middleware at a Glance | 369 |
| B | Glossary | 375 |
| C | Transactions and Menu Paths | 381 |
| D | Coding Examples for Chapter 6 | 385 |
| E | Analysis Roadmaps | 393 |
| F | Authors | 399 |

Contents

| | |
|--------------------|----|
| Preface | 13 |
| Foreword | 15 |
| Introduction | 17 |

Part I: Basic Principles of CRM Middleware

1 A Journey Through CRM Middleware 25

| | |
|---|----|
| 1.1 The Journey from R/3 to CRM | 25 |
| 1.2 The Journey from CRM to R/3 | 28 |
| 1.3 The Journey from CRM to a Mobile Client | 30 |
| 1.4 The Journey from a Mobile Client to CRM | 33 |

2 Inbound Processing and Validation 37

| | |
|---|----|
| 2.1 BDoc | 38 |
| 2.2 Flow | 40 |
| 2.2.1 Flow Contexts in Inbound Processing and Validation | 41 |
| 2.2.2 Displaying Flow Definitions | 42 |
| 2.2.3 Maintaining Flow Definitions | 43 |
| 2.3 Inbound Queues | 44 |
| 2.3.1 qRFC | 45 |
| 2.3.2 QIN Scheduler | 51 |
| 2.3.3 The qRFC Monitor for Inbound Queues | 56 |
| 2.3.4 Naming Conventions for Inbound Queues | 61 |
| 2.4 Inbound Adapters | 63 |
| 2.4.1 Adapter Framework | 63 |
| 2.4.2 Adapter Objects | 64 |
| 2.4.3 Mobile Inbound Adapter | 73 |
| 2.5 Validation | 79 |
| 2.6 Handling Errors When Processing BDoc Messages | 82 |
| 2.6.1 Error Handler | 82 |
| 2.6.2 Displaying and Processing Erroneous BDoc Messages | 83 |

3 Outbound Processing 89

| | |
|-----------------------------|-----|
| 3.1 Replication Model | 90 |
| 3.2 Messaging Flow | 100 |

| | | |
|-------|---|-----|
| 3.2.1 | CSA Queues | 101 |
| 3.2.2 | Replication Service and Outbound Adapters .. | 102 |
| 3.2.3 | Mobile Bridge | 104 |
| 3.2.4 | Flow Contexts in the Messaging Flow | 106 |
| 3.3 | Synchronization Flow | 107 |
| 3.3.1 | CDB Service | 107 |
| 3.3.2 | R&R Service | 108 |
| 3.3.3 | Mobile Outbound Adapter | 122 |
| 3.3.4 | Rejection Messages | 123 |
| 3.3.5 | Flow Contexts in the Synchronization Flow ... | 123 |
| 3.4 | Outbound Queues | 124 |
| 3.4.1 | qRFC | 125 |
| 3.4.2 | QOUT Scheduler | 125 |
| 3.4.3 | The qRFC Monitor for Outbound Queues | 128 |
| 3.4.4 | Naming Conventions for Outbound Queues .. | 132 |

4 BDoc Modeling and Storage 135

| | | |
|-----|---|-----|
| 4.1 | Definition and Structure of BDocs | 135 |
| 4.2 | Data Storage for BDoc Messages | 147 |

5 Groupware Integration 155

| | | |
|-----|---|-----|
| 5.1 | Server-Server Scenario in Groupware Integration | 155 |
| 5.2 | Analyzing Groupware Integration | 164 |

6 Exchanging Customer-Specific Data 169

| | | |
|-------|--|-----|
| 6.1 | Customizations Required in SAP R/3 to Enable the Import of Custom Data | 171 |
| 6.2 | Customizations Required in SAP CRM to Enable the Import of Data from SAP R/3 | 172 |
| 6.2.1 | Creating a CRM Online Table | 172 |
| 6.2.2 | Creating the mBDoc Type | 174 |
| 6.2.3 | Maintaining the Flow | 179 |
| 6.2.4 | Mapping the BAPI Container into the mBDoc Structure | 180 |
| 6.3 | Loading Data from the R/3 System | 183 |

Part II: Performance Optimization in SAP CRM Middleware

7 Introduction to Performance Optimization 187

| | | |
|-----------|---|------------|
| 8 | Inbound Queues | 197 |
| 8.1 | The Number of Inbound Queues Is Too High | 199 |
| 8.2 | The Number of Inbound Queues Is Too Low | 208 |
| 8.3 | Hardware Bottlenecks | 209 |
| 8.4 | Performance Analysis for Processing Single Records | 215 |
| | 8.4.1 Logical Destinations | 216 |
| | 8.4.2 Message Flow Statistics | 219 |
| | 8.4.3 CRM Middleware Trace | 228 |
| 8.5 | Dependencies Between Inbound Queues | 231 |
| 9 | Replication Model and R&R | 235 |
| 9.1 | Basic Principles | 236 |
| 9.2 | Optimizing the Replication Model | 239 |
| 9.3 | Parallelizing the Queues | 256 |
| 9.4 | Optimizing the Access Path | 259 |
| 9.5 | Special Optimization Options in R&R | 264 |
| 9.6 | Mass Changes | 265 |
| 10 | Outbound Queues | 269 |
| 10.1 | System Optimization | 274 |
| | 10.1.1 Optimizing the Disk Subsystem | 274 |
| | 10.1.2 Index Rebuild | 275 |
| | 10.1.3 Buffering qRFC Tables | 276 |
| | 10.1.4 RFC Server Groups | 277 |
| 10.2 | Minimizing the Number of Outbound BDocs | 277 |
| | 10.2.1 Packaged Messages | 278 |
| | 10.2.2 AC Extract | 281 |
| | 10.2.3 Strategies for Using the AC Extract | 284 |
| | 10.2.4 Data Collector | 295 |
| 11 | XML in Optimization | 297 |
| 11.1 | Exchanging Data Between SAP R/3 and SAP CRM via XML | 297 |
| 11.2 | Data Exchange Between the CRM Server and a Mobile Client via XML | 300 |
| 12 | Reorganization | 307 |
| 12.1 | Reorganizing the Middleware | 307 |
| 12.2 | Avoiding Object Links | 315 |

| | |
|---|------------|
| 13 Performing Optimized Mass Changes | 317 |
| 13.1 Optimal Parallelization of the Middleware | 318 |
| 13.1.1 Key Considerations for the Data | 320 |
| 13.1.2 Basic CRM System Considerations | 324 |
| 13.1.3 Summary of the Different Considerations | 325 |
| 13.2 Organizational Measures | 326 |
| 13.2.1 Information and Training | 327 |
| 13.2.2 Final Live Test | 328 |
| 13.2.3 System Preparations | 328 |
| 13.2.4 Monitoring | 329 |
| 13.2.5 Mobile Scenario | 329 |
| 13.3 System Monitoring and Analysis Roadmap | 330 |
| 13.3.1 Analyzing the Work Process Occupancy | 331 |
| 13.3.2 Analyzing the Queue Processing | 334 |
| 13.3.3 R&R Optimization | 337 |
| 13.3.4 CPU Bottleneck Analysis | 342 |
| 13.3.5 BDoc Error Analysis | 345 |
| 13.4 Immediate Actions When Things Really Go Wrong | 347 |
| 13.4.1 A Performance Bottleneck Has Occurred in the CRM | 347 |
| 13.4.2 Performance Collapse During Inbound Queue Processing | 351 |
| 13.4.3 Performance Collapse During Outbound Queue Processing – “ConnTrans Takes Forever!” | 353 |
| 13.4.4 The CommStation Occupies All Work Processes | 358 |
| 14 A Look Ahead | 359 |
| 14.1 Improvements in R&R | 359 |
| 14.1.1 Effects on the Parallelization Process Caused by the New Framework | 361 |
| 14.1.2 Processing Queue Entries in Blocks | 362 |
| 14.2 BDoc Merge | 364 |
| 14.3 Changes in the Administration Console | 365 |
| 14.4 bgRFC | 367 |

| | |
|---|------------|
| Appendix | 369 |
| A SAP CRM Middleware at a Glance | 369 |
| A.1 The Flow | 369 |
| A.2 Inbound and Outbound Processing | 370 |
| A.3 BDocs | 371 |
| A.4 Adapters and Services | 372 |
| B Glossary | 375 |
| C Transactions and Menu Paths | 381 |
| D Coding Examples for Chapter 6 | 385 |
| D.1 Extracting the Book Data from the R/3 System | 385 |
| D.2 Mapping the Extracted Data to the BAPI MTCS Container in SAP R/3 | 388 |
| D.3 Mapping in SAP CRM | 389 |
| D.4 Validating Received Book Data in SAP CRM | 391 |
| E Analysis Roadmaps | 393 |
| E.1 Checklist | 393 |
| E.2 Analysis Roadmaps | 394 |
| F Authors | 399 |
| Index | 401 |

Data sent to CRM is usually written into the inbound queues first and then further processed by CRM Middleware. It is not surprising, therefore, that inbound queues are the first place in CRM Middleware where optimization can begin.

8 Inbound Queues

If we consider the different systems that can be connected to CRM, the SAP R/3 system in particular is the one that can cause problems for CRM due to its large quantities of data. If you change a business object in R/3, the change is transferred directly to CRM and saved in an inbound queue. The inbound queue scheduler takes the data from the queue and transfers it to the R/3 Adapter (see Figure 8.1).

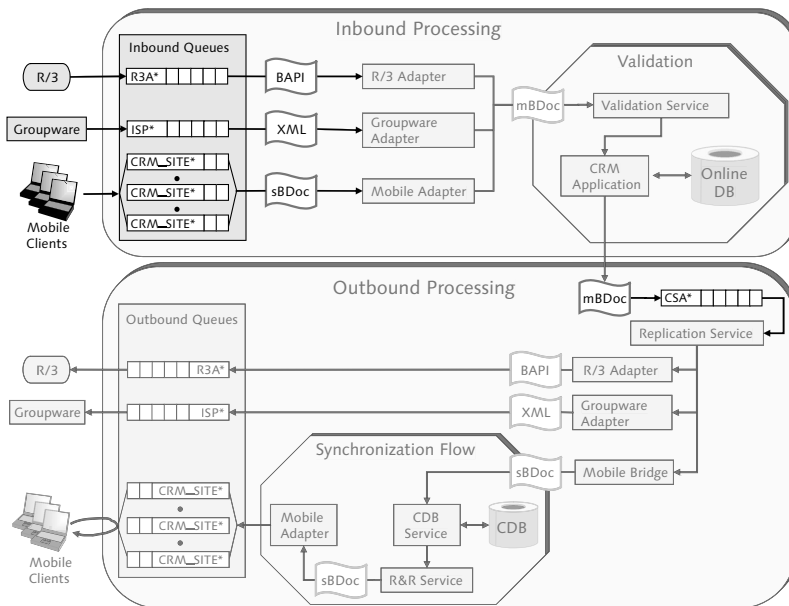


Figure 8.1 Inbound Queues in CRM Middleware

This delta supply of data works well in normal operations. However, if a connected system sends a large quantity of data faster than CRM can process it, the data accumulates in the inbound queue, and the corresponding repercussions, as was explained with examples in Chapter 7, *Introduction to Performance Optimization*, can occur if settings are incorrect. Therefore, the objective should be to optimize the inbound queue processing as much as possible.

All queues in CRM Middleware are based on *Remote Function Calls* (RFCs). RFC is therefore critically important for processing in Middleware. Inbound and outbound queues are implemented using *qRFC* (queued RFC), although Replication & Realignment queues are based on *trRFC* (transactional RFC) rather than qRFC. It is also important to note that, in addition to R3A*, ISP* and CRM_SITE* inbound queues, CSA* queues are also implemented as inbound queues as of CRM Release 4.0. Figure 8.2 shows CRM Middleware from the perspective of RFC. For all inbound queues, there is only one inbound queue scheduler.

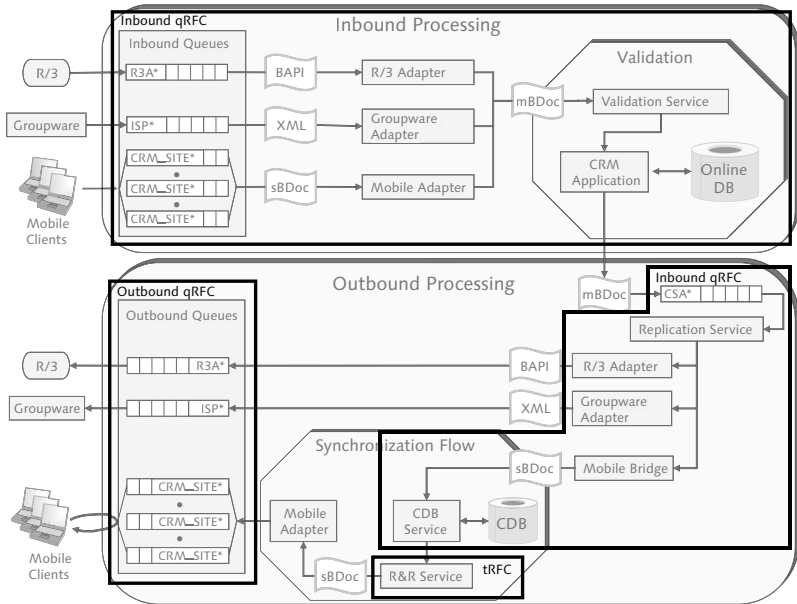


Figure 8.2 RFC in CRM Middleware

Chapter 2, *Inbound Processing and Validation*, contains a detailed description of inbound queues and qRFC.

Different reasons can account for why inbound queues are processed slowly:

Causes of problems

- ▶ So many inbound queues are created that the inbound queue scheduler experiences problems in processing these queues (see Section 8.1).
- ▶ Only a few queues are created, however, these contain a very high number of data records and the available resources in CRM cannot be used effectively (see Section 8.2).
- ▶ All the CRM system's work processes are occupied by the inbound queue processing and a resource bottleneck occurs, which slows down the processing (see Section 8.3).
- ▶ The inbound processing of individual data records is slow (see Section 8.4).
- ▶ Dependencies exist between inbound queues (see Section 8.5).

In the following sections, we will describe the different causes of problems for inbound queue processing and introduce possible solutions.

8.1 The Number of Inbound Queues Is Too High

The inbound queue scheduler ensures that the inbound queues are processed in parallel. To do this, it occupies all work processes available to it. However, the increasing number of queues impairs the performance of the scheduler. The deterioration in performance is clearly noticeable after 10,000 inbound queues. At worst, the performance is impaired to such a degree that it takes longer to assign the next queue entry to a free work process than it takes to post the entry itself. From a performance point of view, this means that the inbound queues are actually processed sequentially, in other words, they are no longer processed in parallel, and the available hardware resources are not used (see example below). The only way to prevent this is to limit the number of inbound queues in CRM.

Example

The company, BodeGolzeSchröder Inc. (hereafter abbreviated to "BGS"), has 2.7 million customers worldwide, 500,000 of which are in Germany and are serviced by 500 field sales representatives. BGS performs a realignment in R/3 once a year. The sales territories of the field sales representatives are reassigned during this realignment. This reorganization does not affect 40% of customers (200,000), since they have traditionally been serviced by the same field sales representative for years and the goal here is to retain this established customer relationship. The remaining 60% are assigned the new sales areas.¹ BGS has developed two reports for this purpose. The Z_DELETE_ASSIGNMENT report deletes the assigned employee responsible for a customer, and the Z_CALCULATE_NEW_ASSIGNMENT report uses a range of criteria to calculate which employee will be responsible for the customer in future. Reports Z_DELETE_ASSIGNMENT and Z_CALCULATE_NEW_ASSIGNMENT change 300,000 data records each. Due to the delta supply of data, 300,000 inbound queues with two entries each are generated in CRM. In addition to the day-to-day activities, CRM Middleware must also process the 600,000 BUPA_REL data records. In BGS's CRM system, the R/3 Adapter and validation usually need two seconds to process a BUPA_REL BDoc. Due to the extremely high number of inbound queues, the inbound queue scheduler needs 2.2 seconds to assign a queue entry to a work process. This results in an estimated inbound processing duration of $600,000 \times 2.2 = 1.32$ million seconds (which is equivalent to 15 days, 6 hours).²

Solution approaches

There are three approaches that you can adopt to reduce the number of inbound queues:

- ▶ Reduce the single records in R/3
- ▶ Change the naming for queues
- ▶ Use R/3 parameter CRM_MAX_QUEUE_NUMBER_DELTA (which ultimately also affects the naming of queues)

Reducing single records

The first approach involves reducing the single records generated by R/3. The company BGS (from our example) has this option. You could change the reports in R/3 to the effect that the COMMIT WORK is executed only after n data records, not after every single data record, and n objects are written into a BUPA_REL BAPI. If we pick 500 as the number to represent n , for example, the Z_DELETE_

1 In most cases, the result is the same as before, whereby the field sales representative retains most of his customers.

2 The impact of daily activities and the duration of outbound processing were not included in this estimate.

ASSIGNMENT report would only create 600, rather than 300,000, inbound queues. Unfortunately, this option is not always available. The other factor to bear in mind with this solution is that many (500) CSA queue entries can be generated from one R3A queue entry. Since CSA queues are also inbound queues in terms of CRM Middleware, they also “overload” the inbound queue scheduler.

The second approach consists of limiting the number of inbound queues by changing the *naming for queues*. R/3 queues for the delta supply of data in CRM have the following names by default: R3AD_<object part><object ID>. There is one entry in the CRMQ NAMES table for each object type. The “object part” element of the queue name is located in the QOBJPART field, and the BAPIFLD field indicates which field is used to fill the “object ID”. Figure 8.3 shows the entry for the BUPA_REL object. The R/3 outbound queue (and the CRM inbound queue) would have the queue name R3AD_BUPA12345 for the object 12345. Chapter 2, *Inbound Processing and Validation*, provides a detailed description of the naming conventions for the different inbound queues for delta loads, initial loads, and so on (R3AD*, R3AI*, etc.).

Changing the naming for queues

The screenshot shows the SAP R/3 interface for the 'Table CRMQ NAMES Display'. The table contains the following fields and values:

| Field | Value |
|------------|--------------------------|
| MANDT | 800 |
| CONSUMER | CRM |
| OBJNAME | BUPA_REL |
| SORT ORDER | |
| BAPISTRUCT | BSC_RELI |
| QOBJPART | BUPA |
| BAPIFLD | PARTNER1 |
| BAPIOFFSET | 0 |
| BAPIFLDLN | 10 |
| FTCHTABLE | |
| FTCHRELFID | |
| FTCHFID | |
| FLDOFFSET | 0 |
| LENGTH | 0 |
| INACTIVE | <input type="checkbox"/> |

Figure 8.3 R3AD Queue Naming; Table CRMQ NAMES in SAP R/3 (Transaction SE16)

By changing the naming for queues, several business partners are written into the same queue, rather than each business partner being written into its own queue. You control the maximum number of queues that can be generated and which object instances are written into a particular queue by determining which and how many positions from the object ID are important for the queue name. You use the `LENGTH` field to control the number of relevant positions for naming queues and the `FLDOFFSET` field to select the position.

For example, if the `LENGTH` field has the value 1 and `FLDOFFSET` has the value 9, this means that all object instances that have the same number on the tenth position are written into the same queue. You can therefore create a maximum of 10 queues.

The queue name is determined by the first object ID, for which a queue is generated, and does not change again until the queue has been processed completely (and disappears). The left-hand column of the table in the example below contains the R/3 object ID of the business partner. The same business partners are used in all four examples, but the values differ for the `LENGTH` and `FLDOFFSET` fields. The right-hand column of the table contains the name of the inbound queue, into which the business partner is written. As you can see from this example, the number of queues depends on the field values and IDs of the business partners.

Example

1. Setting: Standard Naming

Result: The six customers are written into six queues.

| R/3 Customer Number | CRM Queue Name |
|---------------------|-----------------------|
| 0000054601 | R3AD_CUSTOM0000054601 |
| 0000034541 | R3AD_CUSTOM0000034541 |
| 0000099421 | R3AD_CUSTOM0000099421 |
| 0000028421 | R3AD_CUSTOM0000028421 |
| 0000028422 | R3AD_CUSTOM0000028422 |
| 0000067822 | R3AD_CUSTOM0000067822 |

2. Setting: FLDOFFSET = 9 and LENGTH = 1

Result: The six customers are written into two queues.

| R/3 Customer Number | CRM Queue Name |
|---------------------|------------------------|
| 0000054601 | R3AD_CUSTOME0000054601 |
| 0000034541 | |
| 0000099421 | |
| 0000028421 | |
| 0000028422 | R3AD_CUSTOME0000028422 |
| 0000067822 | |

3. Setting: FLDOFFSET = 8 and LENGTH = 2

Result: The six customers are written into four queues.

| R/3 Customer Number | CRM Queue Name |
|---------------------|------------------------|
| 0000054601 | R3AD_CUSTOME0000054601 |
| 0000034541 | R3AD_CUSTOME0000034541 |
| 0000099421 | R3AD_CUSTOME0000099421 |
| 0000028421 | |
| 0000028422 | R3AD_CUSTOME0000028422 |
| 0000067822 | |

4. Setting: FLDOFFSET = 6 and LENGTH = 2

Result: The six customers are written into five queues.

| R/3 Customer Number | CRM Queue Name |
|---------------------|------------------------|
| 0000054601 | R3AD_CUSTOME0000054601 |
| 0000034541 | R3AD_CUSTOME0000034541 |
| 0000099421 | R3AD_CUSTOME0000099421 |
| 0000028421 | R3AD_CUSTOME0000028421 |
| 0000028422 | |
| 0000067822 | R3AD_CUSTOME0000067822 |

Changing the naming for queues Proceed as follows to change the naming for queues (see Figure 8.3):

1. In the CRMQNNAMES table *in R/3*, find the entry with the object name OBJNAME, for which you want to change the queue naming (Transaction SE16, or alternatively, Transaction SM30).
2. Enter the corresponding values for the **Field Offset** (FLDOFFSET) and **Field Length** (LENGTH) parameters.
3. Make sure that the total of the two values does not exceed the maximum length of the object ID. The maximum length of the object ID is specified in the BAPIFLDLEN field.

CSA queues Proceed as follows to change the naming for CSA queues (see Figure 8.4):

1. In the SMOFQFIND table *in CRM*, find the entry with the object name in the **BDoc Type** column, for which you want to change the queue naming (Transaction SM30, or alternatively, Transaction SE16).
2. Enter the values for the **Field Offset** (corresponds to FLDOFFSET) and **Internal Length** (corresponds to LENGTH) parameters.
3. The entries for all objects written into the same queue must be changed; in other words, all entries for which the values in the fields **Queue Object Part** and **Segment Field** are the same.

[>>] Note that you may only change the queue naming if the queues are empty.

Using the CRM_MAX_QUEUE_NUMBER_DELTA parameter

The third approach for reducing the number of inbound queues requires using the CRM_MAX_QUEUE_NUMBER_DELTA parameter. This parameter is maintained in the CRMPAROLTP and determines how many queues are created, irrespective of the settings in the CRMQNNAMES table. If the parameter is set, the ASCII values of the last three positions of the queue name are combined and calculated modulo CRM_MAX_QUEUE_NUMBER_DELTA. The result is a number that is lower than 1,000 and smaller than CRM_MAX_QUEUE_NUMBER_DELTA. The new queue name is created based on this number.

You can maintain the number of queues differently for each object type.³

³ As of PI_BASIS 2006.1, or if you have implemented SAP Note 944633.

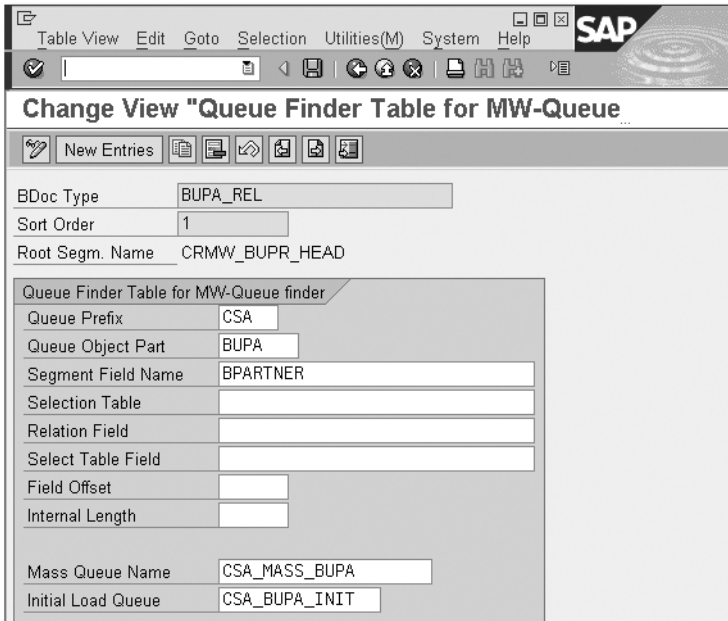


Figure 8.4 CSA Queue Naming; Table SMOFQFIND in CRM (Transaction SM30)

Proceed as follows to limit the number of R3AD queues (see Figure 8.5):

Limiting R3AD queues

1. Create a new data record for the CRMPAROLTP table *in R/3* (Transaction SM30).
2. Enter the value "CRM_MAX_QUEUE_NUMBER_DELTA" in the **Parameter name** (PARNAME) field.
3. Enter the name of the object type in the **Param. Name 2** (PARNAME2) field.
4. Enter the name for your CRM system in the **User** (CONSUMER) field.
5. Enter the number of queues in the **Param. Value** (PARVAL1) field.

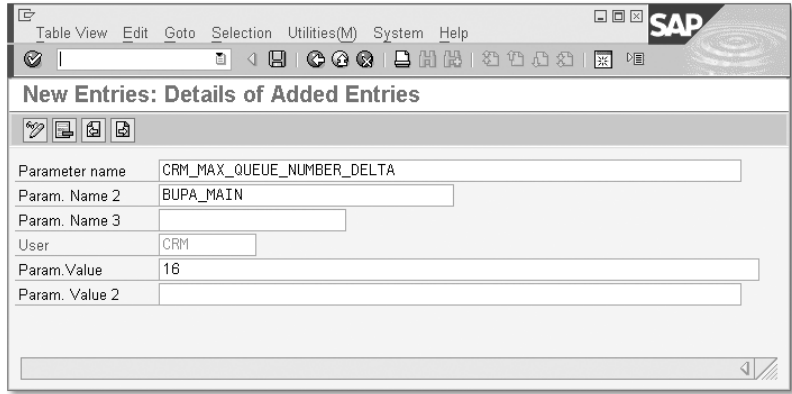


Figure 8.5 Table CRMPAROLTP in SAP R/3

Advantages and disadvantages

When you change the naming for queues, you can use the LENGTH field to determine the maximum number of queues that can be created, however, you cannot determine how many are actually created. The maximum number of queues is not only influenced by the value of the LENGTH field, but also by the type of object ID. Table 8.1 illustrates this dependency. However, it also shows that you can only limit the maximum number of queues in very granular steps. You cannot limit the maximum number of queues to 25, for example.

| Type of Object ID | Value of LENGTH | Max. Number of Queues |
|-------------------|-----------------|-----------------------|
| Numeric | 1 | 10 |
| | 2 | 100 |
| | 3 | 1,000 |
| Alphanumeric | 1 | 36 |
| | 2 | 1,296 |
| | 3 | 46,656 |

Table 8.1 Maximum Numbers of Inbound Queues Depending on the Type of Object ID and the LENGTH Parameter

The actual number of queues is influenced to a certain extent by the value of the FLDOFFSET field. Depending on the selection and status of the number ranges, certain positions in the object ID have the same value for all instances, whereas, for others, all values are found.

In contrast to changing the queue name, you can limit the maximum number of queues exactly by using the `CRM_MAX_QUEUE_NUMBER_DELTA` parameter. The type of object ID is not important in this case. Difficulties only occur if the last three positions of the ID are not distributed equally. This problem does not occur when you use the standard naming function (unless there are fewer than 1,000 objects). However, if you have implemented your own rules when assigning IDs, you may not be able to use the parameter as shown in the BGS example.

Example

To be able to identify a customer's country of origin immediately based on the customer number, all customer numbers at BGS end with a two-digit country code (external number assignment). When the `CRM_MAX_QUEUE_NUMBER_DELTA` parameter is used, a maximum of 10 queues are therefore created for the realignment in Germany.

You could say that the `CRM_MAX_QUEUE_NUMBER_DELTA` parameter is generally better suited to limiting queues than it is to changing the queue naming. To achieve optimum processing, it is worthwhile to distribute the data records across all queues as evenly as possible.

You may only change the queue naming or the `CRM_MAX_QUEUE_NUMBER_DELTA` parameter if all inbound queues are processed. Otherwise, data inconsistencies may occur between R/3 and CRM, as illustrated in the following example.

Attention

Example

Before you implement the change, all data records relating to customer A are written into queue A. After you make the change, they are saved in queue X. If you implement the change, even though queue A still contains data, two queues will exist for a certain period of time in the system that contain data from customer A. Queue A contains the older data records and queue X contains the newer data records. If queue X is processed before queue A, the newer data records supersede the older data records. Consequently, the newer data records are posted in CRM first and then the values in the older data records overwrite the newer values.

We recommend that you first deregister the outbound queues in R/3 and then wait until the R3AD inbound queues have been processed completely in CRM. Then make the change to the CSA queue naming

**Reducing SAP R/3
outbound queues**

and register the queues again. You will find it harder to change the naming of R3A queues, because you cannot prevent outbound queues from being created in R/3. You can therefore only implement a change within a maintenance window if no data is created in R/3, which is transferred to CRM.

8.2 The Number of Inbound Queues Is Too Low

In Section 8.1, we described how you could limit the number of inbound queues. We also mentioned that the actual number of inbound queues can be a lot smaller and data records are not necessarily replicated evenly across all queues. The following example highlights the problems that can occur if you limit the number of queues too much.

Example

BGS has decided not to change the report, but instead has chosen to change the queue naming. To avoid overloading the CRM system too much, the value 1 is selected for the LENGTH parameter and the value 7 is chosen for FLDOFFSET. BGS's CRM system consists of an application server and a database server with four CPUs each. There are 20 dialog work processes in each case on both servers. The CRM system should easily be able to process 10 inbound queues in parallel. BGS expects that 10 queues with 72,000 data records each will be created and that the processing will take 40 hours, based on $72,000 \times 2 \text{ seconds} = 144,000 \text{ seconds}$. After the mass change is started, it becomes apparent that the longest of the 10 queues contains 151,200 data records (i.e., 21 % of the changed customers have the number 3 as the last number before the country code in their customer number; 14 % have the number 8; 10 % have the numbers 7 and 5; 9 % have the numbers 6 and 9; 8 % have the numbers 2 and 4; 7 % have the number 0; and 4 % have the number 1). The processing for this queue takes 3.5 days, based on $151,200 \times 2 \text{ seconds} = 302,400 \text{ seconds}$.

In some circumstances, you can solve the problem described in the above example by analyzing the customer numbers and choosing the corresponding FLDOFFSET parameter. However, this analysis is very laborious and there is also no guarantee that it gives you an ideal value for FLDOFFSET. Alternatively, the maximum number of inbound queues can further increase. The crucial factor here is to

choose the correct balance between the number of inbound queues that is too low and too high.

8.3 Hardware Bottlenecks

In addition to the number of inbound queues, there are other parameters that you must take into account in order to guarantee optimum processing of inbound queues. This is demonstrated in the following example.

Example

For the next test, BGS selects the value 2 for the LENGTH parameter and the value 6 for FLDOFFSET. After the mass change starts, 84 inbound queues, rather than the expected 100, are generated in CRM with an average of 8,500 entries. The inbound queue scheduler uses the available resources to process the queues as quickly as possible, and occupies all dialog work processes. This results in an overload situation (CPU bottleneck). The CRM system is still busy processing only the inbound queues and no other work can be performed with the CRM system (this also applies in particular for the system administrator). The processing time of the individual data records also multiplies due to the CPU bottleneck.

The problems that BGS experienced in the last example were not caused by the number of inbound queues being too high, but due to the fact that the inbound queue scheduler occupied too many dialog work processes of the CRM system. The inbound queue scheduler doesn't have the necessary logic to check how heavily the system is already being utilized and subsequently to decide whether other queue entries can be processed, or whether there should be a break in the processing. All work processes available to the inbound queue scheduler are used for the processing until all inbound queues are empty.

You can only prevent this kind of overloading by limiting the number of dialog work processes that the inbound queue scheduler is allowed to occupy. For this purpose, the inbound queue scheduler is assigned to an *RFC server group*. You create and maintain RFC server groups in Transaction RZ12. Figure 8.6 shows a CRM system with three RFC server groups. The first group (without a name) is the standard group. This group is always used if there is no explicit assign-

RFC server group

ment to a group. The other two groups, **Queue_Scheduler** and **parallel_generators**, were created manually and one instance each was assigned to both groups. In principle, several instances can also be assigned to a group. In such cases, when a user logs on, the instance with the best response times is determined automatically and the user is logged on to this instance.

To create an RFC server group, start Transaction RZ12, and click the **New** button (**Edit • Create assignment menu option**). To display the resource assignment of an instance or to change it, double-click the corresponding row. A **Change Assignment** dialog window appears with the RFC parameter values of the instance.

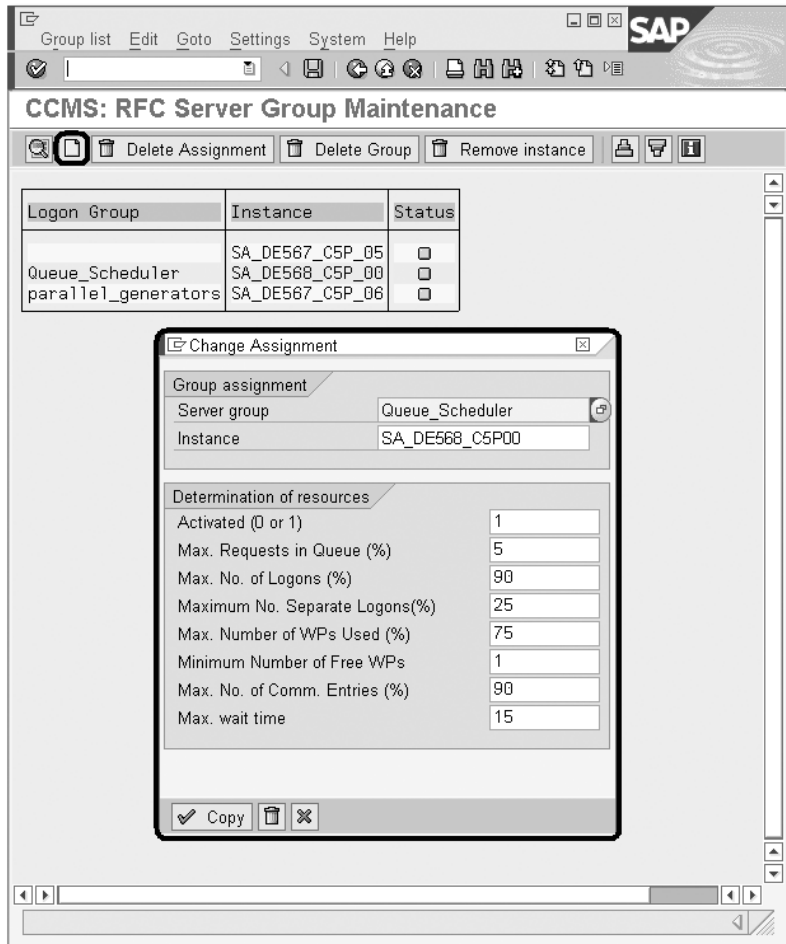


Figure 8.6 RFC Server Groups (Transaction RZ12)

The parameters are used as follows:

Parameters of the
RFC server group

▶ **Activated (0 or 1)**

Switch for activating the determination of resources. This should always have the default value 1 (= active).

▶ **Max. Requests in Queue (%)**

Quote for the number of maximum pending requests in the dialog waiting queue of the dispatcher, which is proportionate to the maximum length of the dispatcher request queue. The default value is 5.

▶ **Max. No. of Logons (%)**

This value specifies the maximum percentage of logons allowed to this instance by asynchronous RFCs (the total number of logons is contained in the `rdisp/tm_max_no` parameter). The remaining percentage continues to be reserved for dialog and HTTP users, i.e., if the number of logons exceeds this value, the caller will not be assigned any resources. The default value is 90.

▶ **Maximum No. Separate Logons (%)**

This value specifies the maximum percentage of logons allowed to this instance by asynchronous RFCs of a user (the total number of logons is contained in the `rdisp/tm_max_no` parameter). If the number of separate logons exceeds this value, the user is not assigned any further resources. Ideally, this value should not be greater than the **Max. No. of Logons (%)** parameter. The default value is 25.

▶ **Max. Number of WPs Used (%)**

Quote for the number of dialog work processes that a user is allowed to use. If the number of dialog work processes used exceeds this value, the caller is not assigned any further dialog work processes. This quote prevents all dialog work processes from being occupied by a user's RFCs. The default value is 75.

The system doesn't check the user name, which means that, if a user logs on to the system several times, each logon is viewed as a separate user, even though the user name is the same.

▶ **Minimum Number of Free WPs**

Quote for the number of dialog work processes that must be reserved for other users. If the number of free dialog work processes is less than the number specified in the quote, the caller is not assigned any dialog work processes. The default value is 1.

The value must always be smaller than the number of dialog work processes (`rdisp/wp_no_dia` parameter); otherwise, an RFC request cannot be processed.

► **Max. No. of Comm. Entries (%)**

Quote for the maximum number of communication entries of an instance that may be used by parallel RFCs (the total value of the communication entries is contained in the `rdisp/max_comm_entries` parameter). If the number of entries used exceeds this value, the caller is not assigned any resources. The default value is 90.

► **Max. wait time**

Maximum number in seconds that a work process can "remain in idle mode" if it doesn't receive any resources after the load on the system has been checked. The actual wait time is determined from the available resources. The fewer resources available, the longer the wait time.

Profile parameters
of RFC server
group

All settings that you implement in Transaction RZ12 are immediately active, however, they are only saved up until the next time when you restart the instance. After you restart the instance, the settings are lost and the old values are active again. To save the values permanently, you must save them as profile parameters. Table 8.2 contains the names of the profile parameters for the individual values in Transaction RZ12.

| RZ12 | Profile Pparameters |
|------------------------------------|---|
| Activated (0 or 1) | <code>rdisp/rfc_use_quotas</code> |
| Max. Requests in Queue (%) | <code>rdisp/rfc_max_queue</code> |
| Max. No. of Logons (%) | <code>rdisp/rfc_max_login</code> |
| Maximum No. of Separate Logons (%) | <code>rdisp/rfc_max_own_login</code> |
| Max. Number of WPs Used (%) | <code>rdisp/rfc_max_own_used_wp</code> |
| Minimum Number of Free WPs | <code>rdisp/rfc_min_wait_dia_wp</code> |
| Max. No. of Comm. Entries (%) | <code>rdisp/rfc_max_comm_entries</code> |
| Max. wait time | <code>rdisp/rfc_max_wait_time</code> |

Table 8.2 Names of Profile Parameters

You assign the inbound queue scheduler to an RFC server group in Transaction SMQR. You select the menu option **Edit • Change AS group and enter the name of the RFC server group.**

Assigning an RFC server group

Figure 8.7 shows that the **Name of AS Group (DEFAULT = All)** parameter no longer has the "DEFAULT" value; it now has the "Queue_Scheduler" value instead.

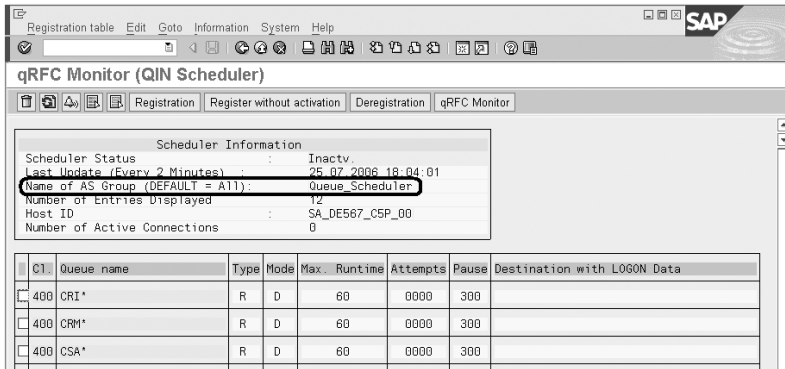


Figure 8.7 Assigning the RFC Server Group (Transaction SMQR)

The optimum setup of RFC server groups depends on the available hardware, volume of data and scenarios used. If you use a pure mobile sales scenario, online or Internet sales users must not be considered when the system is being optimized. In this case, almost all work processes can be made available to the RFC. In all other cases, you must limit the resources for the RFC in such a way that the users will still be able to continue using the system, even if the RFC load is high. If the CRM system has more than one application server, you may find it useful to set up the RFC server groups in such a way that the RFC processing is restricted to the resources of one application server, while the online users use a different application server. By adopting this approach, you can ensure that the users and the RFC will not disturb one another. This holds true especially if data is transferred from the backend, or from/to the mobile clients and the CRM application is used intensively (i.e., both scenarios occur concurrently). The disadvantage of this setup is that the RFC is restricted to one server if the RFC load is high (e.g., if you implement a mass change), even though resources are available on another server (e.g., during the night).

Optimum setting for the RFC server group

If there is only one application server, it is critical that you optimize the parameters of the RFC server group. The distribution of resources should reflect the actual load distribution – RFC load versus online load – in each case.

To prevent the inbound queue scheduler from using all work processes of the CRM system, you must set the **Max. Number of WPs Used (%)** and **Minimum Number of Free WPs** parameters correctly. The **Minimum Number of Free WPs** parameter determines how many work processes are available for the RFC and the **Max. Number of WPs Used (%)** parameter specifies how many work processes *one* RFC user is allowed to use.

If the system has sufficient dialog work processes, the **Minimum Number of Free WPs** parameter should have the value 3, rather than the default value 1. This ensures that you can still log on to this instance through the SAPGUI, even if the load on the system is very high. This recommendation applies only to a pure mobile sales scenario or a pure RFC instance. In all other cases, the value should be greater, and should also be based on the number of simultaneously active (“concurrent”) online users.

When determining the **Max. Number of WPs Used (%)** parameter, remember that the RFC is used not only by the inbound queue scheduler, but by other users as well (replication & realignment, outbound queue scheduler, external systems, and especially mobile clients through the DCOM station/.NET Connector). If you select a value that is too high, although the inbound queues will be processed quickly, the data will accumulate in other areas of the CRM Middleware because resources will not be available there. You can only achieve an optimum processing speed if all the system's components have sufficient resources. We discuss these dependencies in the CRM Middleware in more detail in Chapter 13, *Performing Optimized Mass Changes*.

Detailed documentation about configuring the system resources for parallel RFCs, tRFCs, and qRFCs is available under *SAP NetWeaver* in the SAP Help Portal (<http://help.sap.com>). Click **Search Documentation**. As Search string, enter “Configuration of System Resources for Parallel RFCs tRFC qRFC”. Choose **SAP NetWeaver** and the release and language that you want, and click **Search**.

8.4 Performance Analysis for Processing Single Records

SAP provides an entire range of statistical information and trace options for a performance analysis. Some of these are listed in Table 8.3.

| Transaction | Description |
|-------------|--|
| ST03N | Workload Monitor |
| STAD | Statistical Records |
| ST12 | Single Transaction Trace |
| SE30 | Runtime Analysis (ABAP Trace) |
| ST04 | Database Performance Analysis |
| DB02 | Database Performance: Tables and Indices |
| ST10 | Table Call Statistics |
| ST05 | Performance Analysis (SQL Trace) |
| SQLR | SQL Trace Interpreter |
| SMWMFLOW | Message Flow Statistics |
| SMWT | Middleware Trace |

Table 8.3 Performance Analysis Transactions

Except for the *Message Flow Statistics* (Transaction SMWMFLOW) and the *Middleware Trace* (Transaction SMWT), which we'll discuss in further detail, the analysis transactions are "standard" (i.e., not CRM-specific transactions that are found in every SAP system). Detailed descriptions about these transactions and the most suitable way that you can use them are already available; therefore, we don't want to repeat them here.⁴

Analysis transactions

However, we would like to point out how the performance analysis of CRM Middleware differs from the performance analysis in other SAP systems. The user who executes a transaction often acts as a *filter* for the performance analysis, in order to find the correct statistical data records or trace a particular action in the system. The "user" is not always identified very easily in CRM Middleware. For example,

⁴ We recommend that you refer to the book, *SAP Performance Optimization Guide: Analyzing and Tuning SAP Systems*, by Thomas Schneider, published by SAP PRESS (4th edition, 2005).

you will only be able to clearly establish which user is processing an inbound queue if a *logical destination* has been maintained. Otherwise, the queue processing is performed under the user ID that is currently processing a registered queue (this does not have to be the same queue).

Therefore, a queue entry is not necessarily processed further by the user ID that has written it into the queue; instead, it can be processed using a different user ID. If both of these users have different rights, a user with insufficient rights may try to process the queue entries of another user. The resulting errors are very difficult to analyze because, generally, they rarely occur and can seldom be reproduced.

8.4.1 Logical Destinations

We therefore urgently recommend that you create logical destinations for all inbound queues. There is little effort required, but the benefits are great. You must perform the following steps to create a logical destination:

1. For each queue, you should create one user, whose ID is used to process the queue, for example, *R3A_Inbound* for the R3A* queues. Start Transaction SU01, click the **New** button and create a user of the type "communication".
2. In Transaction SM59, check whether it has an internal connection of the type "NONE" and note the name of this connection.
3. Create a new internal connection in Transaction SM59. Select **Connection Type "L"** and enter the name of the internal connection "NONE" from Step 2 into the **Reference Entry** field on the **Technical Settings** tab (see Figure 8.8). Go to the **Logon & Security** tab and enter the user's logon information from Step 1 here (see Figure 8.9).
4. In Transaction SMQR, you can now assign the internal connection to the corresponding queue (see Figure 8.10).
 - ▶ Select the corresponding queue.
 - ▶ Click the **Registration** button.
 - ▶ In the **Queue Registration** dialog box, enter the ID of the internal connection from Step 3 into the **USERDEST** field.
 - ▶ Confirm your entries.

Figure 8.11 shows that the logical destination R3A_INBOUND is now assigned to all queues that start with the prefix "R3A".

Repeat these steps until you have assigned a logical destination (and therefore a separate user) to all inbound queues in Transaction SMQR. You can now see directly from the process overview (Transaction SM50) which work processes are working on particular inbound queues. The user also proves to be helpful in other transactions. For example, when analyzing BDoc errors (Transaction SMW01), you can use the user ID as a filter in the **User (Creator)** field if you want to select the corresponding mBDocs.

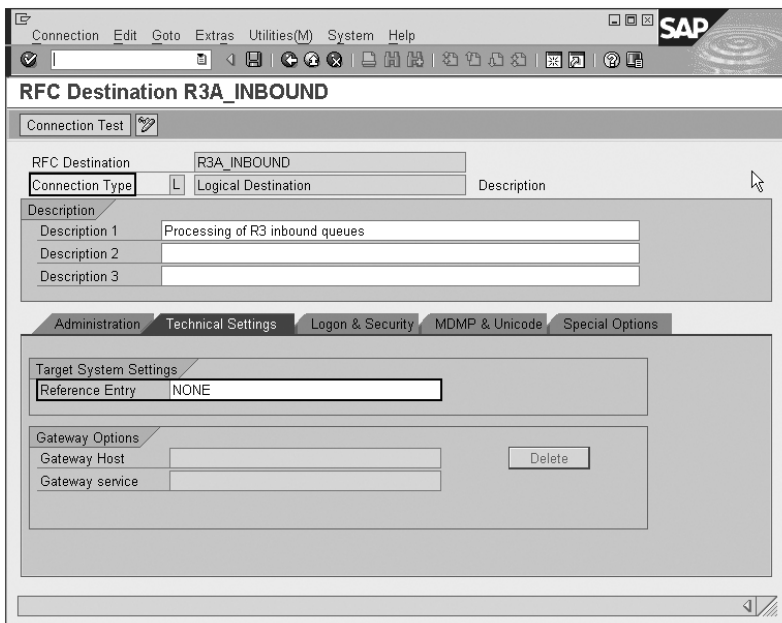


Figure 8.8 Creating a New Internal Connection of Type "L" (Transaction SM59)

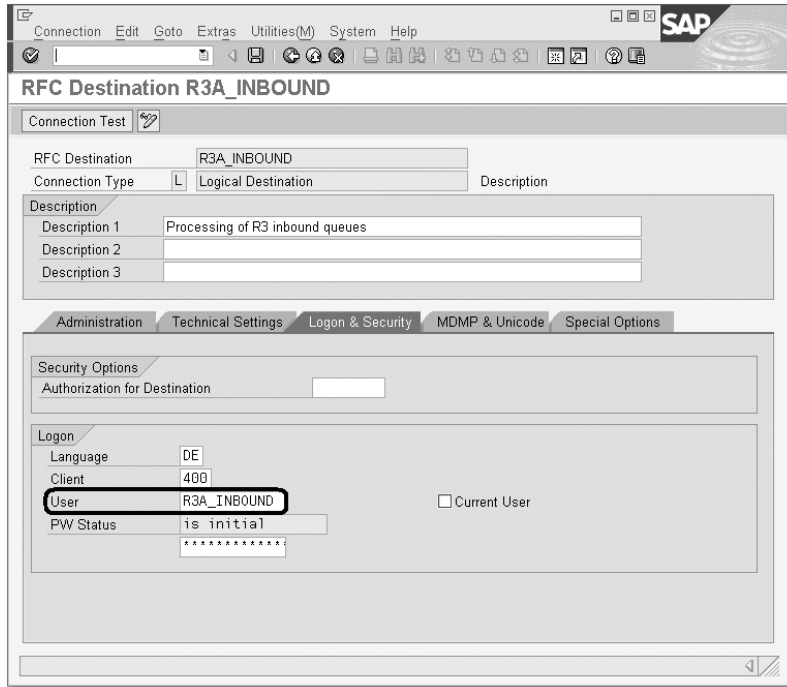


Figure 8.9 Assigning a User to the New Connection (Transaction SM59)

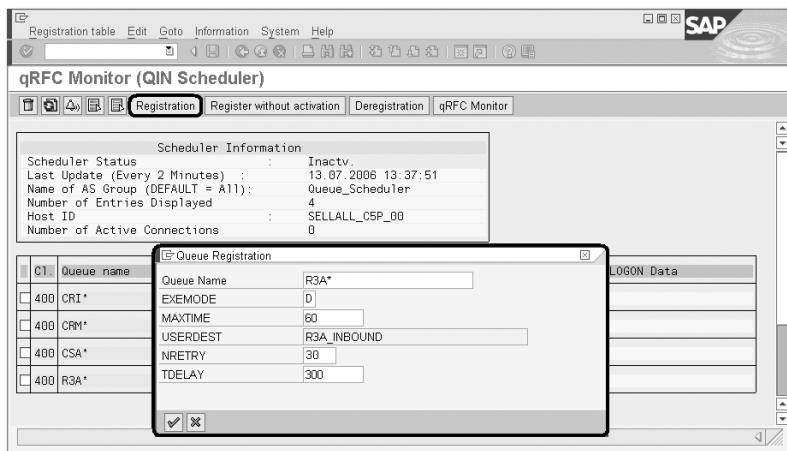


Figure 8.10 Entering a Logical Destination (Transaction SMQR)

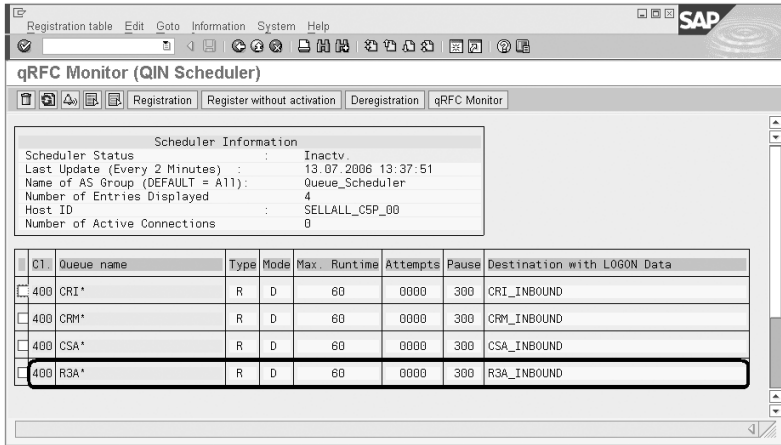


Figure 8.11 R3A Queue with Logical Destination (Transaction SMQR)

8.4.2 Message Flow Statistics

Message flow statistics are CRM-specific statistics that can be very helpful when analyzing performance. You can access the message flow statistics by using following specified path in the SAP Easy Access menu:

Architecture and Technology • Middleware • Monitoring • Message Flow • Display Message Flow Statistics

Alternatively, you can also use Transaction SMWMFLOW. Figure 8.12 shows the transaction's start-up window. You can choose between different statistics: the **Message / Service Kernel Application Statistics** and the **Message / Site / Queue Statistics**.

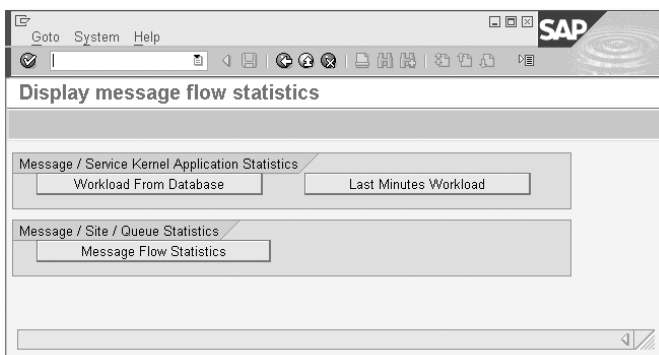


Figure 8.12 Start-Up Window of Transaction SMWMFLOW

Switching statistics on/off

The writing of statistics can be switched on and off. Therefore, prior to a performance analysis, you should check to ensure that the statistics are written. First, select **Goto • Activate statistics** from the menu.

In the next window that opens, you can switch the kernel application statistics and site/queue statistics on and off independently of each other (see Figure 8.13):

► **Message / Service Kernel Application Statistics**

When you click the **Kernel application statistics** button, the window that you see in Figure 8.14 opens. To ensure that the statistics are written for the CRM Middleware, you should mark the checkbox in the **Middleware Message Hub Statistic** row in the **active** column. To make sure that the kernel statistics are also actually activated, you must have already scheduled the `SAP_COLLECTOR_FOR_PERFMONITOR` background job.

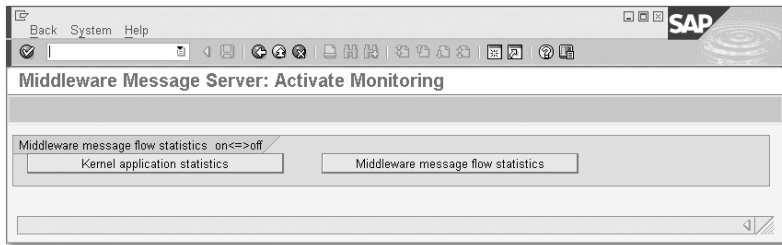


Figure 8.13 Activating Statistics

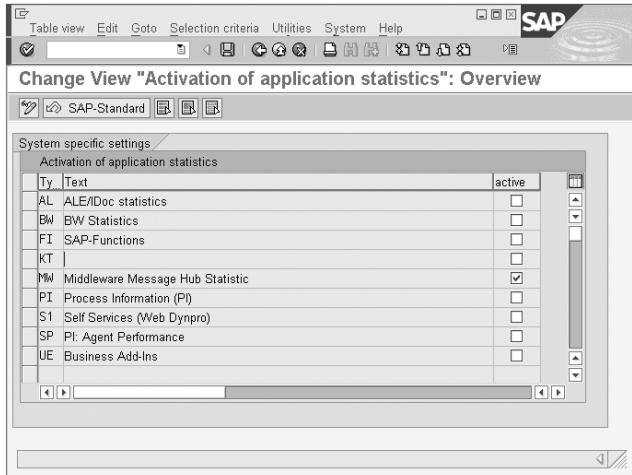


Figure 8.14 Activating Kernel Application Statistics

► Message / Site / Queue Statistics

When you click the **Middleware message flow statistics** button (see Figure 8.13), the window that you see in Figure 8.15 opens. Both the **Monitoring Message Flow** and the **Collector** should be switched on for an analysis.

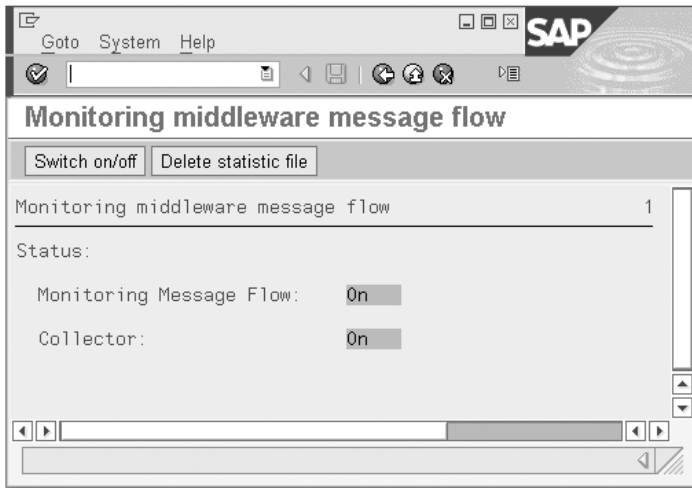


Figure 8.15 Switching Site/Queue Statistics On

Workload Statistics

To display the *workload statistics*, click one of the two buttons **Workload From Database** or **Last Minutes Workload** in Transaction SMWMFLOW (see Figure 8.12).

When you click the **Last Minutes Workload** button, the current load of the last x minutes of the CRM instance on which you are working is displayed. You can define how big “x” is yourself.

Last minutes

When you click the **Workload From Database** button, “historical” data is displayed. You can display statistics of one instance or all instances and choose between different time intervals.

Historical data

In both cases, the layout of the results window is identical. You receive data about the number of BDocs per type that were processed in the time period, as well as data about the processing time, CPU time, wait time, database time and the number of Kbytes requested. You also receive additional information about the total time and the average time for all time data (see Figure 8.16). Since

the values in milliseconds are used for the internal calculation, but the total time is specified in seconds, this may lead to rounding variances. This occurs in particular with relatively small values or with few statistics records.

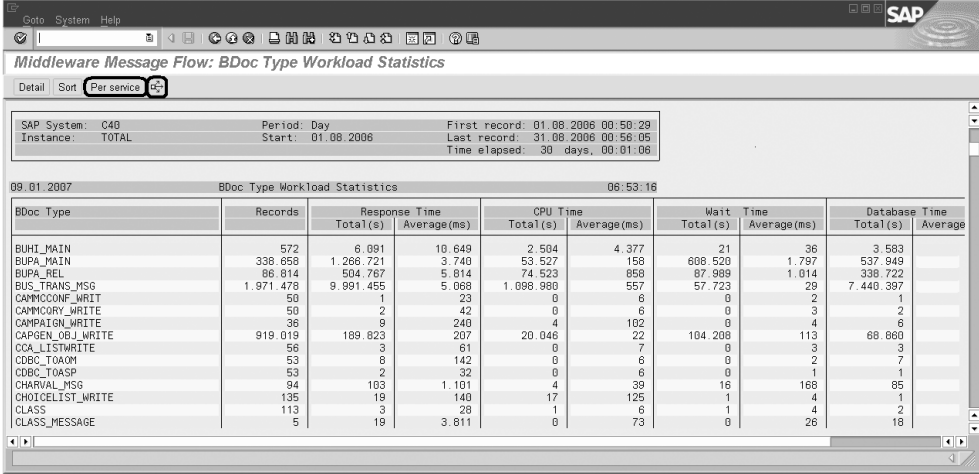


Figure 8.16 Workload Statistics (Transaction SMWMFLOW)

Workload statistics "per service"

You can receive additional detailed information about the processing times of a particular BDoc type by selecting a row and clicking the **Per service** button. The services that were called for processing a BDoc type are then listed, and the response time, CPU time, wait time, and database time are specified for each service (see Figure 8.17).

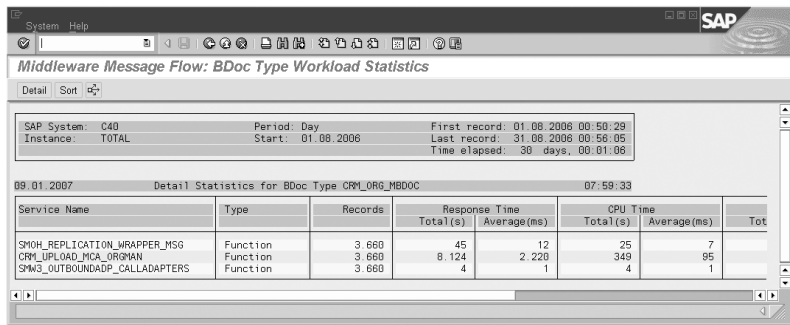


Figure 8.17 Workload Statistics Per Service (Transaction SMWMFLOW)

BDoc type hierarchy

You can obtain information about the *BDoc type hierarchy* by selecting a BDoc type and clicking the **where-used list** button to the right

of the **Per service** button (see Figure 8.16). Figure 8.18 shows an example of the `BUS_TRANS_MSG` BDoc type. Information about the different times that were required for processing a BDoc or a service is also displayed. You will also receive a range of information about the generated BDocs.

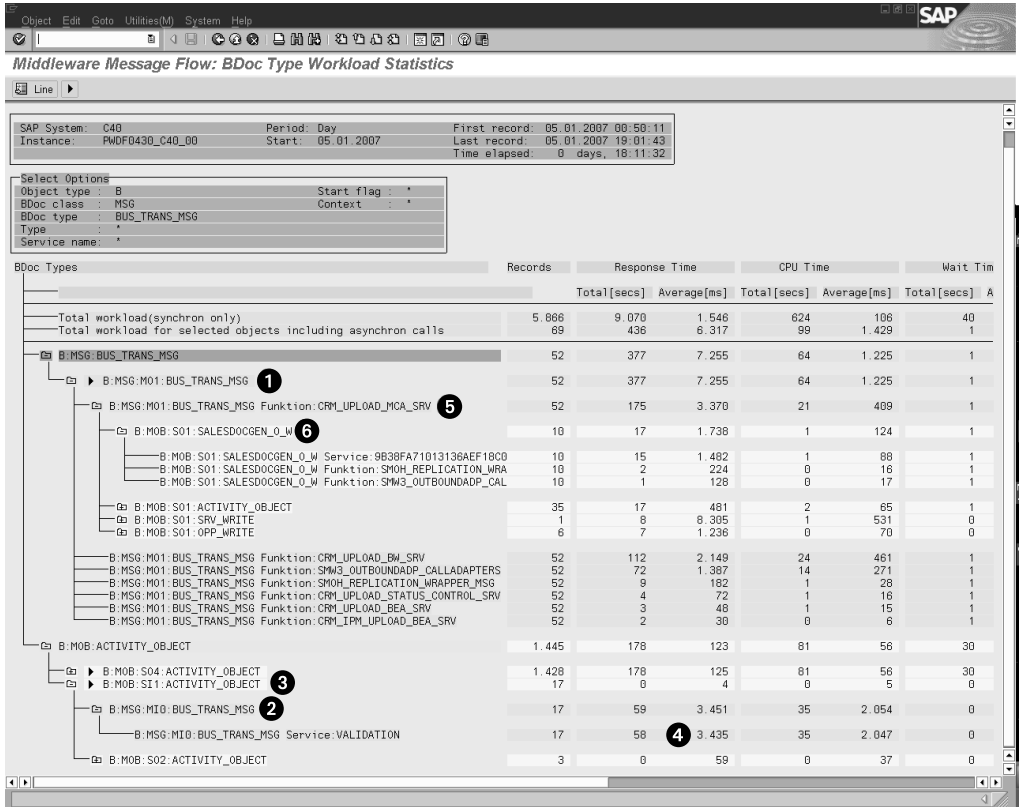


Figure 8.18 Workload Statistics – BDoc Hierarchy (Transaction SMWFLOW)

In row **1**, you can see that 52 mBDocs of the `BUS_TRANS_MSG` type were generated with the `M01` flow context. In addition to these 52 mBDocs, another 17 `BUS_TRANS_MSG` mBDocs were generated with the `M10` flow context (see row **2**). Only the `VALIDATION` service was called for these 17 BDocs. You can identify the relevant predecessor and successor BDocs based on the tree structure. A total of 17 `ACTIVITY_OBJECT` sBDocs were processed with the `S11` flow context (see row **3**) and the 17 `BUS_TRANS_MSG` mBDocs were generated as a result. From a business point of view, this means that 17

activities were transferred from mobile clients to the CRM Server during the day and validated there. The validation took 3.435 ms on average for each mBDoc (see row ④).

The 52 BUS_TRANS_MSG mBDocs with the MO1 flow context do not have a predecessor BDoc. This means that the objects were created in CRM Online itself and were not sent from SAP R/3 or a mobile client. The average response time of 7.255 ms (see row ①) for a BUS_TRANS_MSG mBDoc is only provisionally significant, since a BUS_TRANS_MSG mBDoc can contain different business objects, the processing of which has different degrees of complexity. Underneath row ①, you can see the services and functions that were called when the BUS_TRANS_MSG mBDocs were being processed. (Note that the sequence in the tree structure does not correspond to the actual sequence. The actual sequence is contained in Transaction SMO8FD.)

Row ⑤ contains the processing times of the *Outbound Flow Service for Mobile Clients* (CRM_UPLOAD_MCA_SRV function module). In the mobile scenario, the mBDoc is mapped to one or more sBDocs. Here, you receive information about which sBDocs were generated when the 52 BUS_TRANS_MSGs were processed. You can subsequently also tell, from the names of the sBDocs, which business objects are "hiding" in the 52 BUS_TRANS_MSG mBDocs. When you open the tree under row ⑤, you see that the 52 BUS_TRANS_MSG mBDocs contain 10 orders (SALESDOCGEN_O_W sBDoc), 35 activities (ACTIVITY_OBJECT sBDoc), one service order (SRV_WRITE sBDoc), and six opportunities (OPP_WRITE sBDoc). When you open the tree structure under these SO1 BDocs (e.g., SALESDOCGEN_O_W in row ⑥), you receive information about the different services and functions that were called when the sBDocs were being processed.

Summary If the processing times of an object are not good enough, the statistical data in Transaction SMWMFLOW enables you to analyze exactly which service or function and area (database, CPU) is slow. Based on this information, you can then use a SQL or ABAP trace to determine the database statement or coding segment where time is being lost.

Message Flow Statistics

To display the *message flow statistics*, click the **Message Flow Statistics** button in Transaction SMWMFLOW (see Figure 8.12). Due to

the decoupling of the inbound and outbound processing, the processes may run on different instances. The statistics for the inbound and outbound processing are therefore listed separately. Within the inbound or outbound processing, you can display the total load (**Total**) or the load distribution across the individual instances. You can also select a time interval (day or week). Figure 8.19 shows an example of message flow statistics. The system in this example has over five instances (us0091_Q5C_91 to us4399_Q5C_91) and daily statistics from January 01, 2007 to January 10, 2007.

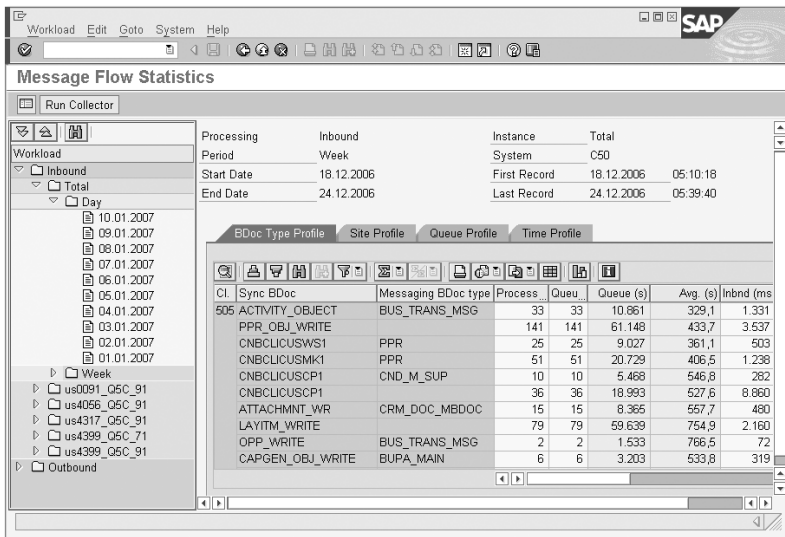


Figure 8.19 Message Flow Statistics (Transaction SMW/MFLOW)

The different queue and processing times are displayed on the right-hand side of the window. Depending on which tab you select, the figures for each BDoc type, site, or queue are summarized. In the **Time Profile** tab, you will find information about how many BDocs were processed in a particular period and how long the processing took. If you select a day as the time interval, the **Single Records** tab will also be displayed. The data for each BDoc is listed here individually.

To make it easier for you to understand the data, we have provided a list of the column headers on the **BDoc Type Profile** tab and their descriptions in Table 8.4 (inbound processing) and Table 8.5 (outbound processing). In Figure 8.20 (inbound processing) and Figure

8.21 (outbound processing), we have illustrated where the times are measured within the Middleware.

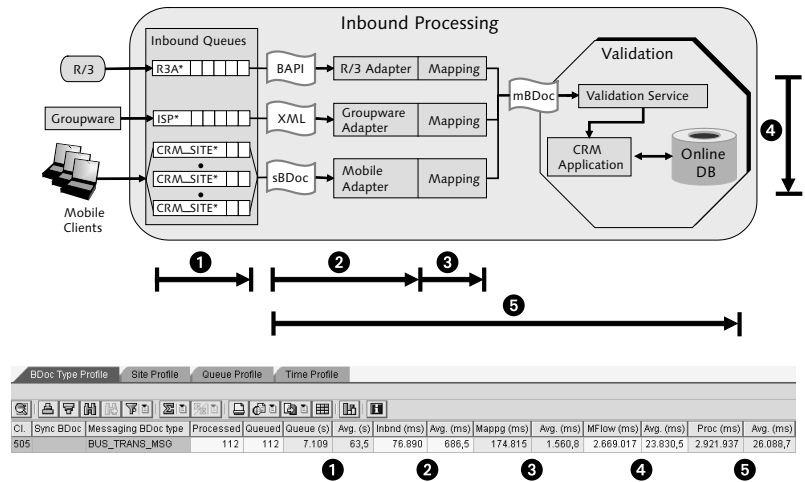


Figure 8.20 Times for Inbound Processing (Transaction SMWMLFLOW)

Times for inbound processing

| Column title | Description |
|---------------------|--|
| Sync BDoc | Name of synchronization BDoc type |
| Messaging BDoc type | Name of messaging BDoc type |
| Processed | Number of BDocs of this type processed in total |
| Queued | Number of BDocs of this type in the inbound queues |
| Queue(s) | Total wait time in seconds in the inbound queues |
| Avg. (s) | Average wait time in seconds in the inbound queues |
| Inbnd. (ms) | Total processing time in milliseconds in the inbound adapter |
| Avg. (ms) | Average processing time in milliseconds in the inbound adapter |
| Mappg. (ms) | Total processing time in milliseconds for mapping the BDocs |
| Avg. (ms) | Average processing time in milliseconds for mapping the BDocs |
| Mflow. (ms) | Total processing time in milliseconds in the message flow |
| Avg. (ms) | Average processing time in milliseconds in the message flow |

Table 8.4 Descriptions of Columns in Inbound Processing (Transaction SMWMLFLOW)

| Column title | Description |
|--------------|--|
| Proc. (ms) | Total processing time in milliseconds (without the wait time in the inbound queue) |
| Avg. (ms) | Average processing time in milliseconds (without the wait time in the inbound queue) |

Table 8.4 Descriptions of Columns in Inbound Processing (Transaction SMWMFLOW) (cont.)

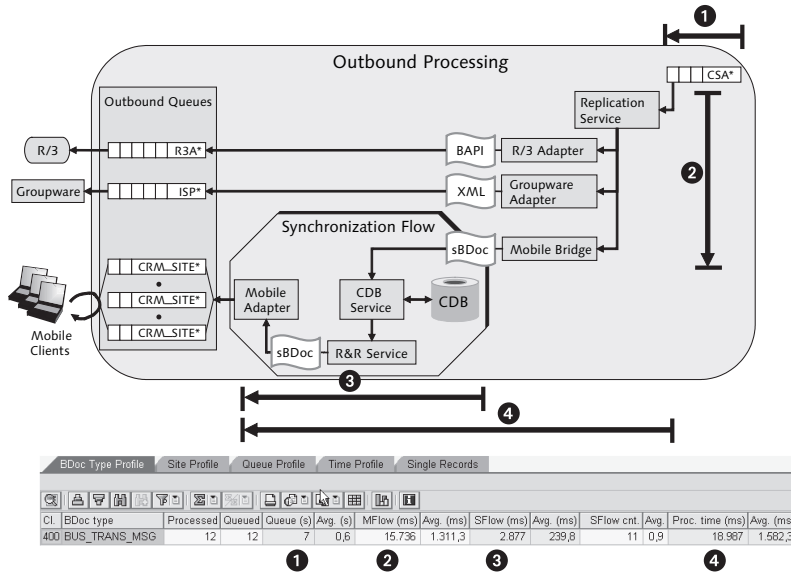


Figure 8.21 Times for Outbound Processing (Transaction SMWMFLOW)

| Column title | Description |
|--------------|---|
| BDoc type | Name of BDoc type |
| Processed | Number of BDocs of this type processed in total |
| Queued | Number of BDocs of this type in the inbound queues |
| Queue(s) | Total wait time in seconds in the inbound queues |
| Avg. (s) | Average wait time in seconds in the inbound queues |
| MFlow. (ms) | Total processing time in milliseconds in the message flow |
| Avg. (ms) | Average time in milliseconds in the message flow |

Times for outbound processing

Table 8.5 Descriptions of Columns in Outbound Processing (Transaction SMWMFLOW)

| Column title | Description |
|-----------------|--|
| SFlow. (ms) | Total processing time in milliseconds in the synchronization flow |
| Avg. (ms) | Average time in milliseconds in the synchronization flow |
| SFlow. cnt. | Number of BDocs of this type in the synchronization flow |
| Avg. | Number of BDocs in the synchronization flow divided by the number of processed BDocs |
| Proc. time (ms) | Total processing time in milliseconds (without the wait time in the inbound queue) |
| Avg. (ms) | Average time in milliseconds (without the wait time in the inbound queue) |

Table 8.5 Descriptions of Columns in Outbound Processing (Transaction SMWM-FLOW) (cont.)

8.4.3 CRM Middleware Trace

Setting trace levels

The *CRM Middleware Trace* enables you to obtain additional information that is written into the Middleware during processing. The CRM system not only allows you to switch the writing of the Middleware trace on and off, it also enables you to define the area where the trace is written and its granularity. For example, you can specify that the trace is restricted to errors and warnings in the message flow (**Trace level: Warning**), whereas you want all information (**Trace level: Detail Level 2**) to be written during the generation (see Figure 8.22). To maintain a trace level, go to **Architecture and Technology • Middleware • Monitoring • Message Flow • Set up Middleware Trace** from the SAP Easy Access menu or by using Transaction SMWTAD.

Trace levels

The following trace levels are available:

- ▶ **Level 0: Error**
Only serious errors are reported.
- ▶ **Level 1: Warnings**
Only situations that can lead to an error are reported.
- ▶ **Level 2: Service Flow**
A note is made of all services that are processed in the Middleware.
- ▶ **Level 3: Detail Level 1**
Additional information is written about the executed programs and modules.

► **Level 4: Detail Level 2**

Program-specific information is written.

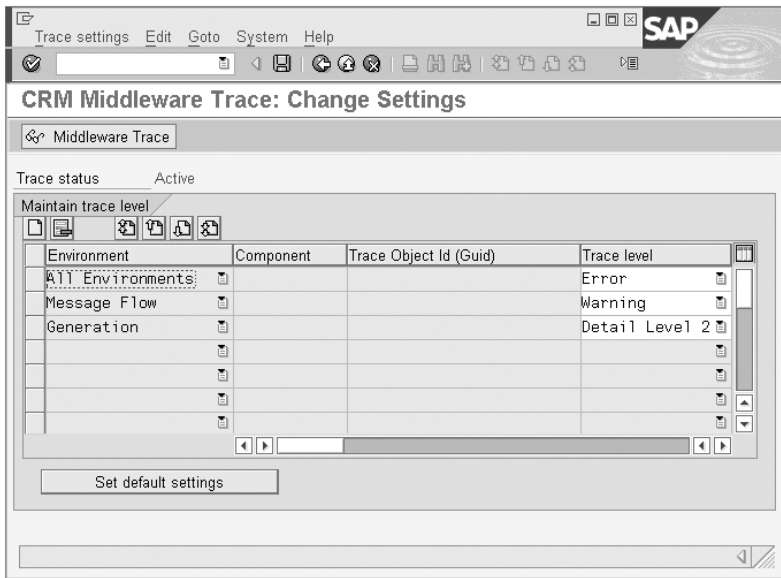


Figure 8.22 Settings for Middleware Trace (Transaction SMWTAD)

The standard setting in the live system is Level 1. Levels 3 and 4 are intended for developers.

For performance and disk space reasons, you should delete old traces on a regular basis. SAP provides the SMO6_REORG2 report for reorganization purposes, which you can also use (among other things) to delete traces.

Chapter 12, *Reorganization*, contains information about this report, and reorganization in general.

You can access the trace itself by using the following path in the SAP Easy Access menu: Displaying a trace

Architecture and Technology • Middleware • Monitoring • Message Flow • Display Middleware Trace

Alternatively, you can also use Transaction SMWT. Figure 8.23 shows the selection screen that appears when you start the transaction.

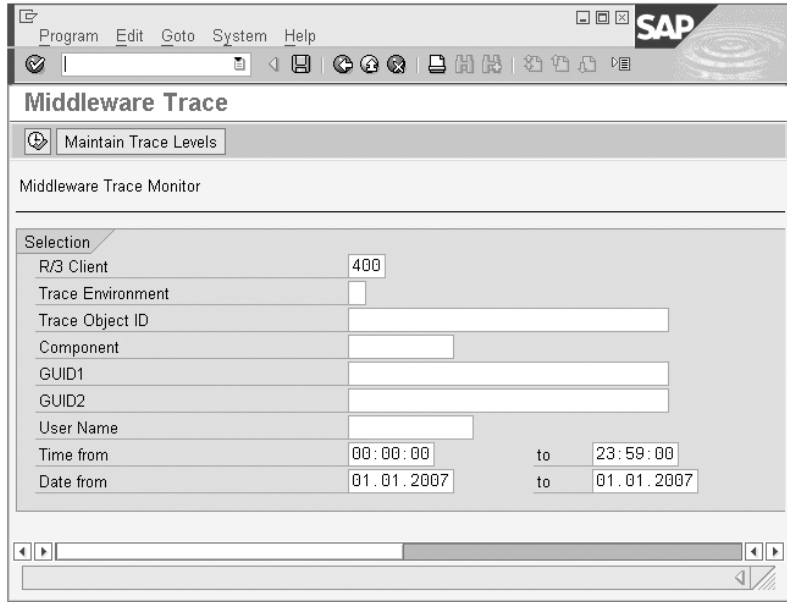


Figure 8.23 Selection Screen for Middleware Trace (Transaction SMWT)

In accordance with the selection criteria you enter, a list of the available traces in the system is displayed. You can display the trace by double-clicking the corresponding row (see Figure 8.24).

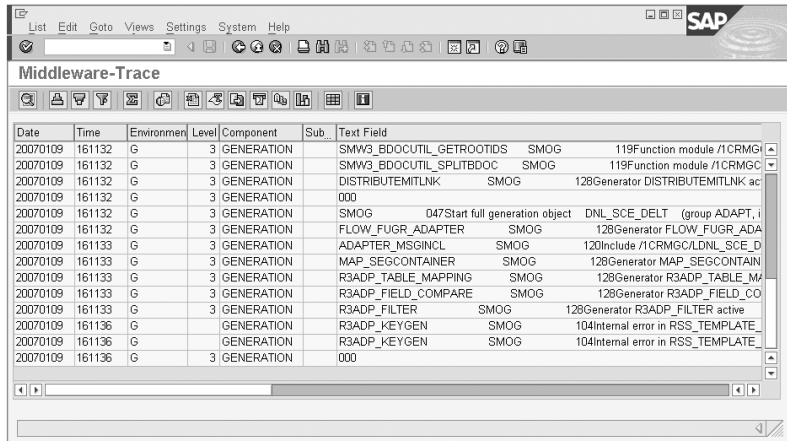


Figure 8.24 Middleware Trace

The trace itself contains a range of information: in addition to the **Date**, **Time**, **Environment** and trace level (**Level**), the **Text Field** and **Trace GUID** in particular are displayed. **Text Field** contains the trace message (maximum of 250 characters) and **Trace GUID** contains the GUID of the trace object. If you're searching for the trace messages for a particular BDoc, for example, you can search according to the GUID of the BDoc in the **Trace GUID field**. Alternatively, you can enter the BDoc GUID as your selection criterion in the **GUID1** field; only the traces that contain messages for the BDoc will subsequently be displayed (see Figure 8.23). In the case of BDocs in particular, you can also jump directly to the trace from Transaction SMW01.

To do this, select the BDoc in Transaction SMW01 and click the **Middleware Trace** button, which is highlighted (see Figure 8.25).

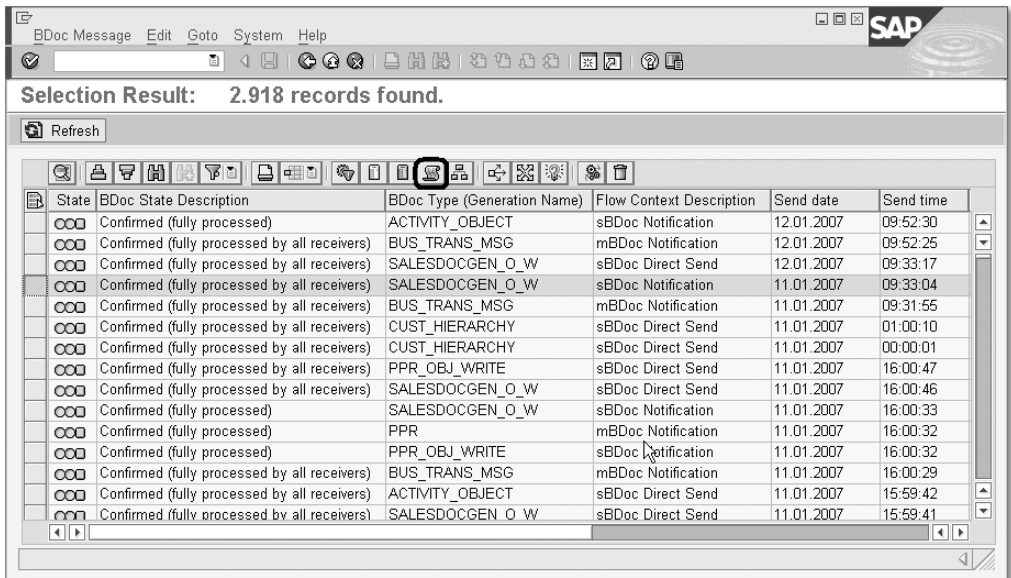


Figure 8.25 Jumping to the Middleware Trace from Transaction SMW01

8.5 Dependencies Between Inbound Queues

You cannot always process CSA inbound queues in parallel on an unrestricted basis. Dependencies may occur between individual queues for some object types, which means that the processing for a

queue must wait until one or more entries of another queue are processed. The following example illustrates this problem, based on a download of business partner relationships from SAP R/3.

Example

A request for all business partner relationships is started in a CRM system. The number of R3AR_BUPA* queues is limited to three and there is no limit on the number of CSABUPA* queues. A BUPA_REL BDoc from R/3 contains all relationships of a business partner (to simplify matters, let's assume that all business partners have four relationships). If an RFC record is being processed in a R3AR queue, the individual relationships are written into different CSA queues (depending on the business partner GUID in the relationship). Therefore, this results in five CSA queue entries (and consequently the corresponding CSA queues), even though only one mBDoc is generated. If the mBDoc is being processed, you see in Transaction SMQ2 that the five queues have the *running* status; while in Transaction SM50, you see that only one work process is busy.

The BDoc can only be processed if all CSA queue entries are in the first position of the queue in question, since otherwise, the queue entries would be superseded.

If the number of CSA queues is not restricted, they usually have one entry. Two or more entries will only appear in a CSABUPA queue if two business partners have a relationship to one another or to a common third party. Figure 8.26 shows that both business partners 1111 and 2222 have a relationship to business partner 4112. If both entries in the R3AR_BUPA1111 and R3AR_BUPA2222 queues are processed, two entries are written into the CSABUPA4112 queue. Only one entry is written into all other CSABUPA* queues. The BUPA_REL BDocs in the CSA queues for business partners GP1111 and GP3333 can be processed immediately and in parallel, since the relevant entries are situated in the first position in the CSA queues. The BUPA_REL BDoc for business partner GP2222 cannot be processed immediately, because the GP2222-GP4112 relationship is not in the first position in the CSABUPA4112 queue. All queues that have a business partner relationship to GP2222 as the first entry (CSABUPA412*) get the *waiting* status, since they can only be processed after the GP1111-GP4112 business partner relationship has been processed in the CSABUPA4112 queue.

If there is no limit to the number of CSA queues (as described in the example), situations only rarely occur where queues have to wait for each other, or so many queues are created that this does not affect the processing speed. It is a different situation if the number of CSA

queues has been severely limited. The CSA queues generally have more entries and the number of queues waiting for each other increases, as explained in the following example.

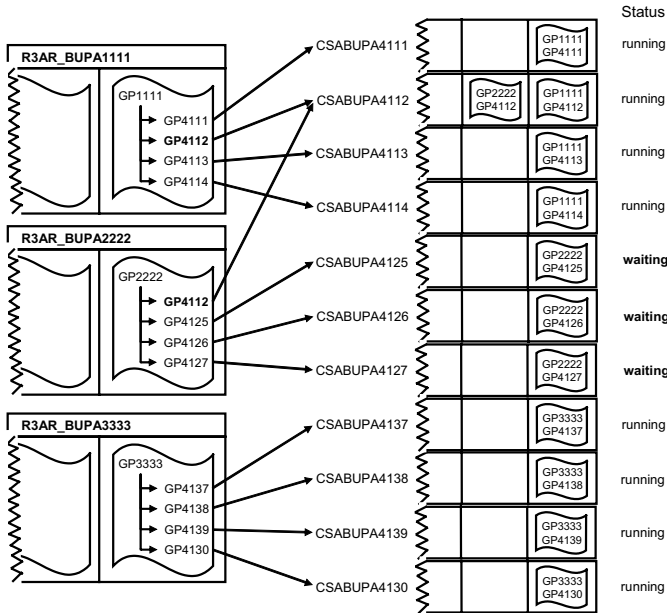


Figure 8.26 Dependencies Between CSA Queue Entries

Example

The prerequisites in this example correspond to those from the last example, however, the only difference in this case is that the number of CSABUPA* queues has been limited to 10.

Figure 8.27 shows that the relationship between GP2222 and GP4127 and the relationship between GP3333 and GP4137 are written into the same queue due to the different queue naming.

Consequently, the BUPA_REL BDoc of business partner 3333 can only be processed if the GP2222-GP4127 entry of queue CSABUPA4127 (i.e., the BUPA_REL BDoc of business partner 2222) has been processed. However, this entry can only be processed if the first entry GP1111-GP4112 of queue CSABUPA4112 (i.e., the BUPA_REL BDoc of business partner 1111) has been processed. In other words, this means that the BUPA_REL BDocs are processed sequentially, even though 10 CSA queues exist.

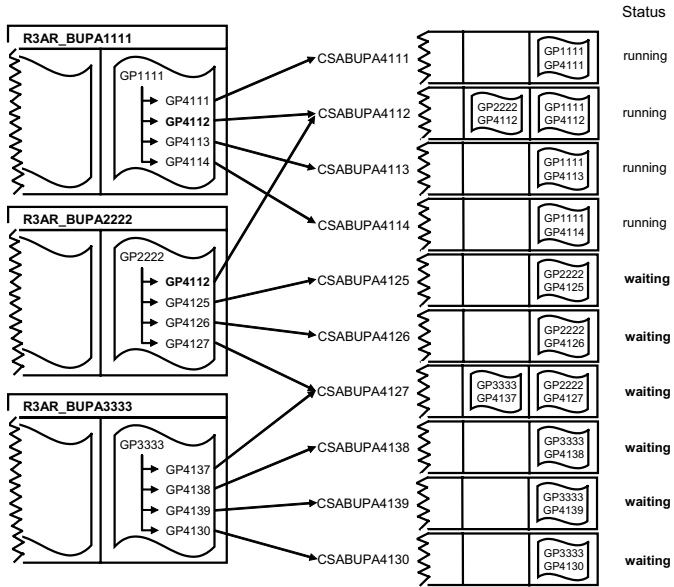


Figure 8.27 Dependencies with Limited Number of CSA Queues

Index

.NET Connector 45

A

AC extract 119, 281
 bulk extract 283
 unfiltered extract 283
AC_EXTRACT queue 111
Adapter 63
Adapter framework 63
Adapter object 64
 activating and deactivating 68
 assignment to a BDoc type 65
 block size 65
 business object 64
 condition object 64
 customizing object 64
 filter settings 72
 initial flow context 69
 mapping module CRM → R/3 73
 object class 67
 parent objects 73
 tables/structures 70
Administration console 31, 90
 improvements 365
 wizard 98
Analysis roadmap
 CPU bottleneck analysis 345, 397
 inbound queue processing 336, 337, 395
 R&R optimization 341
 work process analysis 334, 395
ARFCRDATA, Table 47, 50
ARFCRSTATE, Table 46, 47
ARFCSDATA, Table 46, 47, 49
ARFCSSTATE, Table 46, 47

B

Background RFC 367
BAPI structure 25
BAPIMTCS 171
BDoc 37, 38
 assignment of segment and database 143
 class 135

error status 148
 final Status 149
 instance 140
 interim status 148
 release 146
 robust data storage 152
 sendbits 147
 standard field 147
 static WHERE clause 144
 status 147
 task type 147
 type 140
BDoc instance 40, 140
BDoc link
 adjacent functionality 316
 avoiding 315
 reorganization 314
BDoc merge 364
BDoc message 39
BDoc Modeler 140
BDoc release, WHERE clause 146
BDoc statistics, reorganization 309
BDoc store 75, 79
BDoc type 26, 39
 lock 146
bgRFC → see Background RFC 367
Block size 278
Bulk extract 283
Bulk message 123
Business document 38
Business object, adapter object 64

C

CDB 31, 107
CDB service 31, 107, 108
Checkbox structure 175
Classic part → see mBDoc 137
Communication monitor, reorganization 310
Condition object, adapter object 64
Confirmation message 122
ConnTrans 32, 74
 transfer duration 272
Consolidated database → see CDB 31
CP_CODEPAGE 302

Creating a database table 172
 Creating an object class 181
 CRM_MAX_QUEUE_NUMBER_DELT
 A, parameter 204, 207
 CRMPAROLTP, Table
 CRM_XML_BACKGROUND_PROCES
 SING_ON 299
 CRMPAROLTP, table
 number of inbound queues 204
 CRMQNAME, Tabelle
 FLDOFFSET, Feld 202
 CRMQNAME, Table 61, 204
 LENGTH, field 202
 CRMRFCPAR, Table
 XML control 297
 CRMSUBTAB, Table 67, 171
 CRS_FIRST_DOWNLOAD_TRIGGER
 171
 CSA queue 29, 101
 Current state message 122
 Customizing object, adapter object
 64

D

Data collector 253, 281, 295
 reorganization 312
 Data Integrity Manager 134
 Data transfer
 delta 63
 initial 63
 DB statistics 261
 DBSTATC, Table 262
 Default pool 276
 Deletion message 123
 avoiding 247
 Delta data transfer 63
 Destination
 excluding 127
 parameters 127
 registering and deregistering 127
 DIMa → see Data Integrity Manager
 134
 Disk subsystem 274
 Distribution
 bulk 238
 intelligent 237
 intelligent, switching to bulk 242
 intelligent, without filter criteria 241

Distribution model 31
 Dynamic mapping 76

E

Error Handler 78, 81, 82
 EXEMODE, Parameter 54
 Extended Markup Language → see
 XML 297
 Extension part → see mBDoc 137
 EXTRACT queue 111
 EXTRACTBLK queue 111
 Extractor 171

F

Flow 37, 40
 context 40, 179
 definition 42

G

Generated mobile inbound adapter
 74
 Generic mobile inbound adapter 74
 GNRWB, Transaction 304
 Groupware adapter
 GWA_01 156
 GWA_02 156
 Groupware connector → see Group-
 ware integration 158
 Groupware integration
 analysis 164
 client-client scenario 155
 data queue, primary 160
 data queue, secondary 160
 folder, private 156
 folder, public 156
 groupware adapter 156
 groupware connector 158
 groupware connector proxy 158
 MapBox 157
 MapBox, log files 164
 MapBox, RFC destination 158
 MBMANDTSTORE, table 162
 overview 155
 payload interface 158
 server-server scenario 155
 system queues 160, 164

trace of internal SyncPoint 165
trace of payload interface 166
userlist.xml 162
 GWI → see Groupware integration
 155

H

Hardware Bottleneck 209

I

Inbound adapter 26, 37, 63, 69
 Inbound processing 37
 data from R/3 27
 mobile client data 35
 Inbound queue 26, 33, 37, 44, 48
 dependencies 231
 deregistering 54
 details 59
 entries 60
 of mobile client 32
 overview 56
 parameters 54
 reduce number of queues 200
 registering 52
 slow processing 199
 status 58
 Inbound queue name
 data from mobile clients 61
 data from SAP R/3 61
 for CSA queues 101
 Inbound queue scheduler 45, 51
 activating 52
 performance 199
 status 51
 Inbound scheduler 51
 Index fragmentation 262
 Index quality 262
 Initial data transfer 63
 Initial load 183
 Integration model
 message exchange 136
 synchronization 136
 Interlinkage 98, 248

J

Java Connector (JCo) 45

K

KEEP pool 276
 Kernel application statistics 220

L

Logical Unit of Work 47
 Lookup table 31, 90, 252
 LUW 47

M

MapBox → see Groupware integration
 157
 Mapping 75
 BAPI container in mBDoc 180
 dynamic 76
 mBDoc to sBDoc 31
 static 75, 139
 Mapping function module 26
 Mapping method 34
 Mass change 188
 planned 326
 unplanned 326, 327
 MAX_PACKAGE_SIZE, parameter
 278
 MAXTIME, Parameter 54, 353
 mBDoc 26, 38, 135
 classic part 137
 creating the classic part 177
 creating the extension part 174
 extension part 137
 MBMANDTSTORE, table 162
 Message flow statistics 224
 kernel application statistics 220
 Middleware message flow statistics
 221
 switching on/off 220
 Messaging BDoc → see mBDoc 26, 38
 Messaging flow 101
 Middleware message flow statistics
 221
 Middleware trace 228
 displaying a trace 229
 Reorganization 309
 setting trace levels 228
 Mobile adapter 107
 Mobile application BDoc 136

Mobile bridge 30, 90, 102, 104
 Mobile inbound adapter 34
 generated 74
 generic 74
 Mobile outbound adapter 31, 122

N

Naming for queues
 advantages/disadvantages 206
 Neighbour functionality 315
 NRETRY, Parameter 54

O

Online database 26, 37
 Outbound adapters 69, 90, 102, 104
 Outbound processing 89
 for mobile clients 32
 for R/3 29
 Outbound qRFC with recipient list
 269
 benefits 270
 Outbound queue 29, 31, 48, 89, 124
 displaying an overview 128
 displaying details 130
 displaying entries 131
 in R/3 25
 of mobile client 33
 status 129
 Outbound queue name
 additonal ones 134
 data to mobile clients 132
 data to R/3 132
 Outbound Scheduler 125

P

Parallel processing, optimizing the
 middleware 318
 Payload interface → see Groupware
 integration 158
 Performance analysis 215
 SMWMFLOW 219
 SMWT 228
 Publication 93

Q

QIN Scheduler → see Inbound queue
 scheduler 51
 QOUT Scheduler 125
 activating 126
 status 126
 QREFTID, Table 49
 qRFC 45, 48, 125
 monitor for inbound queues 56
 qRFC Monitor
 for Outbound Queues 128
 qRFC monitor for inbound queues 45
 Query BDoc → see Mobile application
 BDoc 136
 Queue naming
 changing the naming for queues 201
 Queue, stopping 257
 Queued RFC → see qRFC 45

R

R&R 285
 definition 235
 DEPENDENCY queue 360
 distribution-relevant fields 238
 internal optimization 264
 new improvements 360
 new queue framework 360
 optimizing a mass change 267
 parallel processing of queues 361
 parallelizing queue processing 256
 processing queues in blocks 362
 queues 236
 realignment 235
 replication 235
 replication wrapper 238
 R&R queue 111
 displaying 111
 starting and stopping 114
 status 113
 R&R queue demon 112
 starting and stopping 113
 status 113
 R&R queue framework 110
 R&R service 31, 107
 R/3 outbound adapter 29
 R3AC1, Transaction 64, 182
 R3AC3, Transaction 64, 180

- R3AC5, Transaction 64
 - R3AC6, Transaction
 - controlling the reorganization process* 310
 - queue parallelization* 256
 - R3AM1, Transaction 183
 - R3AR2, Transaction
 - one-time request* 314
 - R3AS, Transaction 183
 - REALIGN queue 111
 - Realignment 31, 109
 - Rejection message 78, 123
 - Remote Function Call → see RFC 45
 - Reorganization
 - BDoc links* 314
 - BDoc messages* 308
 - BDoc statistics* 309
 - data collector* 312
 - key generation* 309
 - middleware trace* 309
 - of data for sites that cannot be activated* 313
 - request* 314
 - SAP_MW_REORG, variant* 308
 - SMO6_REORG* 308
 - SMO6_REORG2* 308
 - SMW3* tables* 308
 - standard variant* 308
 - statistics of the CommStation sessions* 310
 - subscription agent* 313
 - Replication 31, 108, 114
 - bulk* 115
 - intelligent* 116
 - Replication & Realignment 287
 - Replication & Realignment → see R&R
 - Replication & Realignment service → see R&R service 31
 - Replication model 90
 - optimization* 239, 248
 - optimizing interlinkages* 248
 - optimizing the bulk publications* 240
 - Replication object 90
 - Replication object type 91, 92
 - bulk* 96
 - Dependent* 97
 - Intelligent* 96
 - Simple Bulk (MESG)* 95
 - Simple Intelligent (MESG)* 96
 - Simple Intelligent (SYNC)* 96
 - Replication service 29, 102
 - mBDoc* 103
 - sBDoc* 114
 - Replication wrapper
 - mBDoc* 103
 - sBDoc* 114
 - RFC 45
 - RFC libraries 45
 - RFC server group 209, 257
 - assign* 213
 - create* 210
 - optimal number* 325
 - parameters* 211
 - profile parameters* 212
 - RFC Software Development Kit 45
 - RSANAORA, Report 263
 - RSRLDREL, program 314
 - RSTRFCQD, Report 355
 - RSTRFCQDS, Report 355
 - RZ12, Transaction 210
- ## S
-
- SBDM, Transaction 74, 77, 140, 177
 - sBDoc 30, 38, 135
 - block size* 278
 - structure* 136
 - Scheduler, tRFC 46
 - SDIMA, Transaction 134
 - SDK, (RFC) Software Development Kit 45
 - SE11, Transaction 172
 - Segment, field assignment 136
 - Service, generating 182
 - Setting up a logical destination 216
 - Site 93
 - deactivating* 244
 - mass deactivation* 246
 - Site type 94
 - site deactivation not supported* 313
 - Sizing 277
 - SM50, Transaction
 - occupancy of the work processes* 332
 - SM59, Transaction 54, 126
 - setting up a logical destination* 216
 - SMO 275
 - SMO8FD, Transaction 42, 79
 - SMO9_KYTBL, Table
 - reorganization* 309

- SMOE_BULK_SITE_ACTIVATION,
Report 285
- SMOEAC, Transaction 61, 90, 132
AC extract 281
XML optimization 303
- SMOECK, Transaction 241
- SMOEGENDET, Table 100
- SMOEGENHEA, Table 100
- SMOEGENLOG, Table
reorganization 313
- SMOEJOBID, Table
reorganization 310
- SMOFFILTAB, Table 72
- SMOFINICON, Table 69
- SMOFOBJCLA, Table 68
- SMOFOBJECT, Table 64, 65, 68
- SMOFOBJPAR, Table 73
- SMOFPARSFA, Table 301, 310
deactivating mBDoc links 315
processing R&R queues in blocs 362
- SMOFPARSFA, table
block size 278
- SMOFQFIND, Table 204
- SMOFQNAMEs, Table 132
- SMOFSUBTAB, Table 73
- SMOFUPLMAP, Table 73
- SMOGEN, Transaction 182, 244
- SMOHILTP, Table 99
- SMOHJOBQ, Table 259
- SMOHLUBULK, Table 96, 238
- SMOHMSGQ, Table 111, 259
optimizing the access path 259
- SMOHMSGST, Table 259
- SMOHPUBL, Table 93
- SMOHQTAB, Table 61, 132
- SMOHQUEUE, Transaction 111, 113,
114, 236, 243
AC extract 282
block size 278
Stop queue 353
- SMOHREPOBJ, Table 92
- SMOHSGQST, Table 111
- SMOHSITEID, Table 94
- SMOHSITEQ, Table 111, 259
- SMOHSUBSCR, Table 95
- SMOHSUBSIT, Table 95
- SMOJDC, Transaction 255
- SMOJDCPROC, Table
reorganization 312
- SMQ1, Transaction 128
- SMQ2, Transaction 56
- SMQR, Transaction 51
changing RFC server group 213
MAXTIME 353
setting up a logical destination 216
- SMQS, Transaction 125
- SMW01, Transaction 83, 316
jumping to the Middleware trace
231
- SMW01, transaction
DEBUGMODE 152
- SMW1SPRVDR, Table 313
- SMW3*, Table
reorganization 308
- SMW3BDOCIF, Table 43, 79, 179
- SMW3FDBDOC, Table 44, 105
- SMW3FDBDOC, Transaction 44, 105
- SMW3FDCUST, Table 44
- SMW3FDCUST, Transaction 44
- SMW3FDIF, Transaction 44, 76, 79,
179
- SMW3FDSTD, Table 44
- SMW3FDSTD, Transaction 44
- SMWMCOMM, Transaction 310
- SMWMFLOW, Transaction 219
message flow statistics 224
workload statistics 221
- SMWMSESSHT, Table 310
- SMWMSESSIN, Table 310
- SMWT, Transaction 229, 309
- SMWT_TRC, Table
reorganization 309
- SMWTAD, Transaction 228
- SPRO, Transaction 82
- ST03N, Transaction 309, 310
- ST06, Transaction
Idle time 342
Load average 344
- Static mapping 75
- Storage Area Network 274
- Storage quality 276
- SUBCHECK queue 111
- Subscription 94
changing the assignment of sites 120
- Subscription agent 99
reorganization 313
- Subscription generators 99
- Synchronization 63
- Synchronization BDoc → see sBDoc
38

Synchronization flow 90, 107
 System landscape 187
 heterogeneous 297
 homogeneous 297
 inhomogeneous 297
 System monitoring 330
 System Optimization Services 275

T

TDELAY, Parameter 54
 TID 47
 Transaction Identifier 47
 Transactional Remote Function Call →
 see tRFC 45
 Transactional RFC → see tRFC
 tRFC 45, 46
 TRFCQDATA, Table 50
 TRFCQIN, Table 50
 TRFCQOUT, Table 49, 50
 TRFCQSTATE, Table 50
 TRFCRSTATE, Table 50
 TRFCSDATA, Table 50
 TRFCSSTATE, Table 50
 TSMW3_STAT, table 147

U

Upload 63
 USERDEST, Parameter 54

V

Validation 37, 79, 179
 data from R/3 27
 mobile client data 35
 Validation service 26, 79

W

WHERE clause → see BDoc 144
 Wizard 98
 Workload statistics 221
 switching on/off 220

X

XML 297

between CRM Server and a mobile client 300
between R/3 and CRM 297
code page conversion 300
data exchange between R/3 and CRM
 297
data exchange with mobile clients
 300
decoupling the application from the conversion 299
performance increase for mobile client data exchange 300
restrictions regarding optimization
 305
synchronous RFC 298

Z

ZAP message 123, 284