



Best Practice for **BDoc Message Analysis**

mySAP CRM

Releases 3.0, 3.1, 4.0 and 5.0

Document Version 2 – October 13th, 2006

Best Practice - BDoc Message Analysis

Motivation:

This document will allow you to understand the role of BDoc messages in your system, to monitor their flow, and to react effectively to error situations.

As a pre-requisite you should be familiar with the content of the SAP training class CR500 ("Middleware Overview"), which explains the concepts of the CRM Middleware and the data flow.

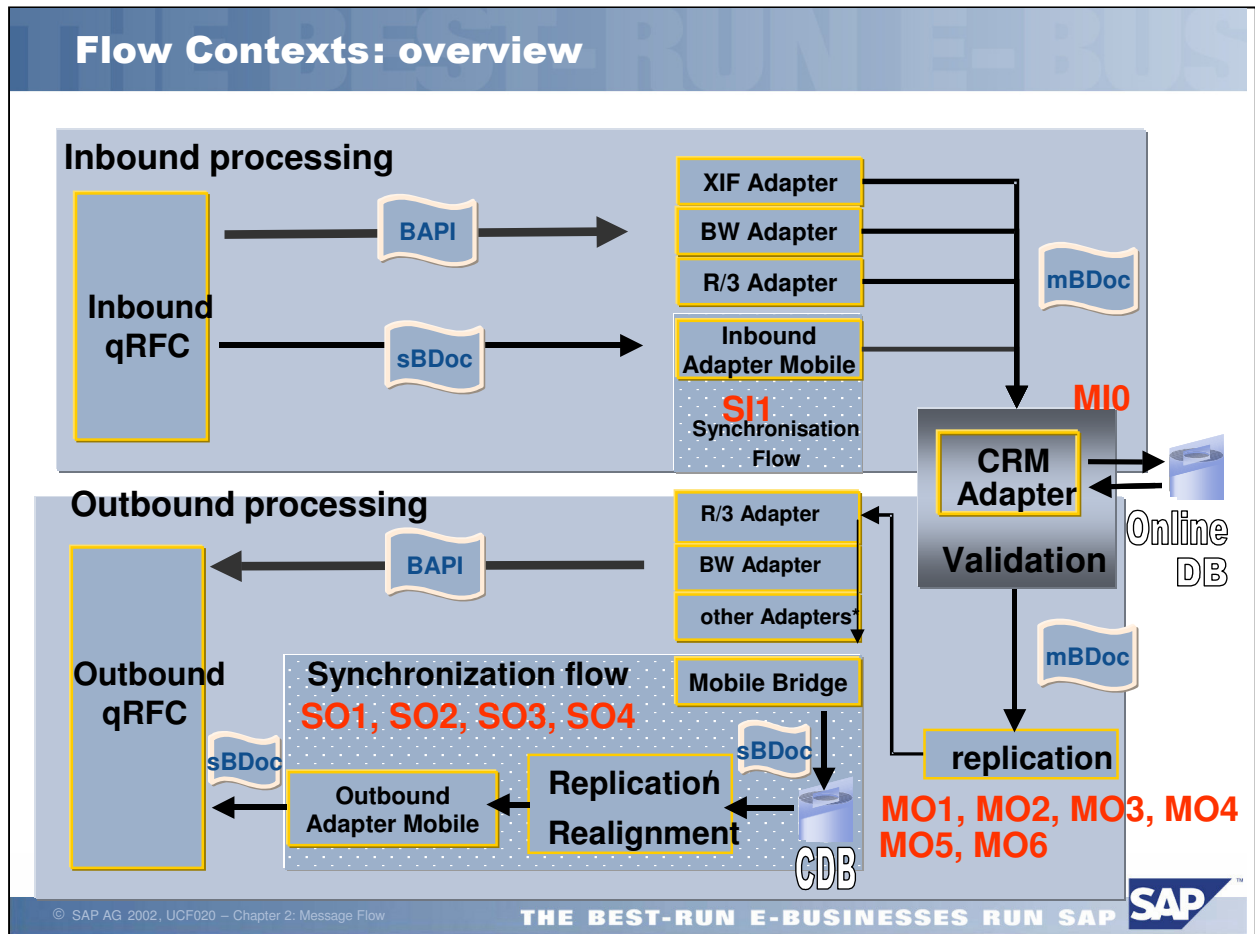
Furthermore SAP Support offers an onsite service called SMO-SA for CRM ("Solution Management Optimization - System Administration for CRM"). Detailed information is available at the SAP Service Marketplace at <http://service.sap.com/sysadmin>.

Table of contents:

1	Introduction	3
1.1	Middleware flow basics	3
1.1.1	Messaging flow	4
1.1.2	Synchronization flow.....	4
1.1.3	Flow configuration	5
1.2	Components of BDoc messages.....	6
2	Functional description of monitoring	8
2.1	Usage of the BDoc monitor	8
2.2	Error handling procedure	10
2.2.1	Basic consideration.....	10
2.2.2	BDoc message analysis procedure	11
2.2.3	Analysis of erroneous BDoc messages.....	12
2.2.4	Analysis of intermediate BDoc states.....	16
2.2.5	Analysis of final BDoc messages	20
2.2.6	Analysis of particular non-final BDoc states.....	22
2.2.7	Appendix 1: Classification of BDoc States	24
2.2.8	Appendix 2: Classification of Flow Contexts.....	24
3	Display unprocessed BDoc message summary.....	26
3.1	Handling procedure.....	26
4	Examples	28

1 Introduction

BDoc messages are used in your SAP CRM system as containers for the data that constitute a business process (application message, transaction).



*Other outbound adapters can be the XIF Adapter or the Groupware Adapter.

1.1 Middleware flow basics

The flow control for the BDoc messages distinguishes between synchronization flow (s-flow) and messaging flow (m-flow). Both flow types are used for inbound and outbound BDoc message processing. Inbound processing occurs, when a remote system (for example R/3 Backend system) posts data into the processing system (CRM Server). Outbound processing occurs, when the processing system (CRM Server) publishes or posts data to remote systems (for example R/3 Backend) or to „listeners“ (for example BW Adapter, Billing Engine). It can also be used if the CRM posts data to the CDB (Mobile Bridge) or other outbound adapters.

M- and s-flow consist of different steps, so called flow contexts. A flow context is defined as a sequence of services, which must be called within this flow context. The inbound flow consists of inbound flow contexts MI... (m-flow) or SI... (s-flow). For outbound flows the contexts are named as MO... (m-flow) or SO... (s-flow).

1.1.1 Messaging flow

Messaging flow is an infrastructure

- for passing data to CRM-Online applications,
- for processing data in the CRM-Online applications, and
- for distributing processed data to other destinations (e.g. R/3 Backend system or mobile clients consolidated database CDB) [so-called notifications]. This kind of data is processed in the CRM-Online application database and has to be forwarded to other data receivers, which are subscribed to this data and have to be informed or notified about these changes.

Messaging flow is used in all SAP CRM scenarios. In detail you can imagine the whole process as follows:

- Inbound BDoc messages are passed to a validation service. Usually this is the CRM-Online application. The validation service checks the content of the BDoc message (for example, for consistency with the customizing of the CRM system), in the same way as when a new object is created in a dialog transaction. If the BDoc message implies a suitable action, the validation service also performs the processing of the data. Generally, the call to a validation service that is not rejected, results in the outbound processing (notification).
- Whenever an application object is changed successfully, the application starts the outbound processing, notifying interested components about the change. The application creates a messaging BDoc and the Middleware's messaging flow dispatches this notification to potential receivers. Examples for application objects are: business partner, product, sales order, and conditions.

1.1.2 Synchronization flow

Synchronization flow is specialized for CDB based message processing in a CRM Mobile environment. It is only used in the Mobile Scenario.

Inbound s-BDocs can be processed in two ways:

- For synchronization BDoc types with assigned messaging BDoc type (the assignment is done in BDoc modeler, transaction SBDM) they are mapped to messaging BDocs, which are then passed to m-flow validation (see above). This m-BDoc message is not stored (persisted), but only exists during runtime to call the validation. After a successful validation the CRM application creates an outbound m-BDoc message.
- Otherwise they are passed to synchronization BDoc outbound processing straightaway. This would be the case for so-called "Mobile-only" objects, which have no correspondent object in CRM Online (only exceptional cases).

To trigger the synchronization outbound flow the mobile bridge for the messaging BDoc type must be activated, see chapter 1.1.3 "Flow Configuration". Outbound processing can be part of the "initial load" processing or notification (delta) processing. Initial load processing implies that data distribution to mobile clients is not active and only CDB needs to be updated using bulk CDB service. Notification processing implies that data distribution to mobile clients is active and receiver determination, realignment and extract processes need to run.

1.1.3 Flow configuration

Flow control is configured through entries in the following tables:

SMW3BDOCIF

Process functions with exactly one occurrence per BDoc type.

- **Inbound processing**

- for m-Flow (context MI0): validation function for m-BDoc in field VALIFUMO
- for s-Flow (context SI1): mapping from s-BDoc to m-BDoc in field MAPCLASS (a class with interface IF_SMW_MAP and method MAP_SYN2MSG)

- **Outbound processing**

- for m-Flow: alternative mobile bridge function in field MOBBRIDGE (compare table SMW3FDCUST)

SMW3FDSTD

Default [outbound] processing (independent of BDoc type). This table contains the default flow definitions in case SMW3FDBDOC and SMW3FDCUST do not contain entries overriding the default rules.

SMW3FDBDOC

BDoc type specific [outbound] processing. In case the default replication service and outbound adapter are not sufficient, it is possible to deliver additional services by including them in that table.

SMW3FDCUST

Client specific and BDoc type specific [outbound] processing.

This is the place to activate the mobile bridge (Field ACTIVE = "X"). In order not to override the default processing, mobile bridge function modules must be added in the following contexts:

- **MOA** = mBDoc Notification (additional calls)
- **MOB** = mBDoc Notification Multiple (additional calls)
- **MOC** = mBDoc Initial Load (additional calls).

At the customer site, only table SMW3FDCUST should be maintained.

Attention: The flag "Mobile Bridge" (MOBBRIDGE) should only be used when the standard mobile bridge function module should be overwritten by entries from table SMW3BDOCIF and field MOBBRIDGE.

Normally this is not required. Refer to the notes 629861 (CRM 4.0) and 632700 (CRM 3.X) to get more information about the standard mobile bridge delivered by SAP.

The evaluation logic for outbound processing is "from bottom to top" of these tables. First, customer enhancements in SMW3FDCUST are checked, then whether there are BDoc type specific entries in SMW3FDBDOC. If not, the standard services from SMW3FDSTD are taken.

Hint: You can check the flow definition for each BDoc type within transaction SMO8FD. You will get a list of all possible flow contexts with their assigned service functions modules (as assembled by the configuration in above tables).

1.2 Components of BDoc messages

To support this flow infrastructure the BDoc messages (or simply BDocs) are composed of:

- Exactly one message header,
- A list of message receivers (outbound processing),
- An error segment, e.g. a list of validation errors,
- A message body (classical part, s-BDocs and m-BDocs)
- A message extension (m-BDocs only),
 - This extension is supposed to hold the transaction data, whereas the classic part (message body) shall only contain fields used in the receiver determination by the replication component
- Root ID (only for BDoc message with single instance),
- A list of receiver specific error information.

The following table contains a detailed description of the different components of a BDoc message:

BDoc component	Where used	Purpose	Remarks
Message Header ^(*)	All BDocs	Message-ID, BDoc type, message type, flow context, validation state, BDoc state, queue name, links to original / reference or request BDocs, sender site, user (creator), root-ID, time stamps.	Exactly one per BDoc message
Message Body (Classic part)	All BDocs	s-BDoc: complete data segment m-BDoc: only fundamental fields for simple replication (receiver determination)	Build in BDoc Modeller: BDoc Overview -> Message Structure
Message Extension	m-BDocs only	Holds complete transaction data for m-BDoc	Build as complex DDic-structure, displayed in BDoc Modeller: BDoc Overview -> Related Data Type
Receiver List	Outbound m- and s-BDocs only	Determines receiving sites, internal sites as CRM and CDB are not listed	Result of Replication&Realignment (s-BDoc) or simple Replication (m-BDoc)
Root ID	Outbound intelligent distributed s-BDocs and optional for m-BDocs, Inbound s-BDocs after changes in Mobile Client, for inbound m-BDocs optional	Key of corresponding entry in application table (the GUID of the primary key field of the application instance)	Only for BDoc messages transporting a single application instance

Error Segment	All BDocs	Validation or technical errors	<p>For multiple receivers each receiver has an own error segment</p> <p>Validation errors occur in the inbound processing</p> <p>Receiver errors occur in the outbound processing</p> <p>Technical errors occur as well in inbound as in outbound processing</p>
---------------	-----------	--------------------------------	--

(*) See also SAP note 778354 for a technical description of the fields in the **BDoc Message Header** (table SMW3_BDOC). For monitoring and troubleshooting, please pay special attention to the parts of

- **BDoc links** (fields REFBDoc_ID, ORGBDoc_ID, REQBDoc_ID) where you can track a chain of several BDoc messages belonging together.

Example for a BDoc chain:

- Message "A" = s-BDoc inbound
- [Message "B" = m-BDoc inbound] - *not persisted!*
- Message "C" = m-BDoc outbound
- Message "D" = s-BDoc outbound

The BDoc links are now stored as following:

- Message "A" = s-BDoc inbound
 - REFBDoc_ID = <empty>
 - ORGBDoc_ID = <empty>
- Message "C" = m-BDoc outbound
 - REFBDoc_ID = "B"
 - ORGBDoc_ID = "A"
- Message "D" = s-BDoc outbound
 - REFBDoc_ID = "C"
 - ORGBDoc_ID = "A"

=> You see that each subsequent message contains a link to its predecessor (= reference BDoc) and its "root" (= original BDoc).

- **Queue name** (field QNAME)
 - Contains the name of the corresponding qRFC queue, where the BDoc message is temporarily stored.
 - If multiple queues are involved, see table SMW3_BDOCQ to get a list of all qRFC queue names for a single BDoc message ID.
 - In general, you can find valuable information regarding the usage of qRFC inbound and outbound queues in CRM (and their naming conventions) inside the SAP note 765236 ("FAQ: Queueing in CRM and R/3").

2 Functional description of monitoring

You can display and trace the flow of BDoc messages within the CRM Middleware. You can delete incorrect BDoc messages, trigger repeated processing of erroneous BDoc messages, and display the recipient list and receiver errors.

2.1 Usage of the BDoc monitor

Transactions: **SMW01 / SMW02**

Menu path:

CRM 4.0/5.0: *Architecture and Technology → Middleware → Monitoring → Message Flow
→ Display BDoc Messages / Display BDoc Messages Summary*

CRM 3.0/3.1: *Middleware → Monitoring → Message Flow
→ Display BDoc Messages / Display BDoc Messages Summary*

SMW01 has the following features:

a) Analysis

- Search for special error situations per BDoc ID, BDoc type, user, date, message ID, flow context, BDoc status and linked BDoc messages.
- Successfully processed messages appear with a **green light** (=final state), those still in process with a **yellow light** (=intermediate state), and those with an error condition with a **red light** (=error state).

b) Display functionality

- *Display the Middleware Trace*
CRM 4.0/5.0: Choose *BDoc Message > Display > BDoc Message Trace*
CRM 3.0/3.1: Choose *BDoc Message > Display > Middleware Trace*
- *Display the data content*
CRM 4.0/5.0: Choose *BDoc Message > Display > Classic part*
Choose *BDoc Message > Display > Extended part*
CRM 3.0/3.1: Choose *BDoc Message > Display > Data Segments*
Choose *BDoc Message > Display > Data Extended*

Both options are also available as XML output. It can be used in case the BDoc structure can no longer be read after structure changes.

- *Display a recipient list*
CRM 4.0/5.0: Choose *BDoc Message > Display > Errors/Receivers*
CRM 3.0/3.1: Choose icon 'receiver list'.
(internal receivers like CRM and CDB are not shown)
- *Display object links to other BDocs or application object*
CRM 4.0/5.0: Choose *BDoc Message > Display > Object Links*, or
push button "Links"
(if object link writing is switched off, use push button "Neighbours"; same
information, but collected on demand)
CRM 3.0/3.1: Push button "Links" (Ctrl+Shift+F6)
Hint: BDoc links are stored in tables SMW0REL and SRRELROLES (object type
'MIDMESSAGE') and will be deleted by the Middleware reorg job SMO6_REORG2. See also
SAP note 493156 for more information.

c) Corrective action

- *Reprocess failed BDocs*

CRM 4.0/5.0: Choose *BDoc Message* > *Process* > *Reprocess*

CRM 3.0/3.1: Choose *BDoc Message* > *Process* > *Retry to reprocess*

Warning:

Do not reprocess the BDoc messages without checking for the possibility of overlapping effects, like overwriting actual data! If newer BDoc messages for the same application object exist and have been fully processed, reprocessing an older BDoc message may cause the current data to be overwritten by a BDoc containing data from an older state. If available for the application object, you should also check the change documents, to see whether more recent changes than the date of the BDoc message have been made.

If you reprocess multiple BDoc messages, make sure the selection is sorted with ascending time stamps in order to keep the right sequence. The actual order of reprocessing corresponds to the order of the marked messages on the screen.

- *Delete failed BDoc messages*

CRM 4.0/5.0: Choose *BDoc Message* > *Process* > *Delete*

CRM 3.0/3.1: Choose *BDoc Message* > *Process* > *Mark as deleted*

Warning:

Do not delete BDoc messages without solving the problem! Deleting a BDoc message means that the receiving system(s) will not be updated, thus creating inconsistencies. Only delete BDoc messages when you are sure they are obsolete or can be recreated by sending or requesting the application object again.

2.2 Error handling procedure

2.2.1 Basic consideration

The processing of messages can occur as follows:

- The whole processing is successful.
- The receiver of a message refuses the data processing and returns error status and error information.
- The receiver of a message does not return a processing status.
- The receiver of a message processes the message and returns success status (undetected error).
- The processing of the message is abnormally ended due to a short dump.
- The processing of the message is abnormally ended due to a technical error.
- An adapter cannot map or communicate the message. The posting of the data to the receiver fails during outbound processing. The message for the respective receiver has an error state.

If a message has multiple receivers, each receiver might have a different processing state, like one receiver confirmed successful processing, but another receiver returned an error state. In such a case, please note that each receiver has its own error segment and can be reprocessed individually.

Failed BDoc messages must be analyzed and the errors have to be solved. The non-processing of the error messages will lead to inconsistencies in the system landscape, as the messages contain updates or new objects that have to be updated in the receiver system.

We can distinguish between errors in the inbound and errors in the outbound processing:

- ***Inbound processing:***

In this case, there are three different flow contexts:

SI0/SI1 (synchronization) and MI0 (messaging). Validation (E04) and technical errors (E01) can occur in these contexts.

The validation error is thrown by the validation service on the CRM Server application, which is also responsible for entering the corresponding error segments to the BDoc message. Mark the BDoc message and choose icon 'Errors' (CRM 3.0/3.1) or 'Show BDoc Msg errors/Receivers' (CRM 4.0/5.0) to get the error information of the local application when processing (validating) the message in inbound direction. Usually only error information is included for rejected messages.

Only inbound messages may have validation errors: M-BDoc messages may not be able to be stored in the CRM-Online database or s-BDoc messages can be rejected resulting in an outbound rejection message, which is then sent back to the Mobile Client.

- ***Outbound processing:***

After the BDoc message has successfully passed the validation phase, it is processed by the defined business logic and then passed to the outbound processing.

If an error occurs, error types are raised. Here the error types E01 / E02 are thrown, if an error occurs during the application outbound handling. CRM 3.0/3.1: In these cases choosing 'Errors' in transaction SMW01 returns no result. Instead display details of the error notifications by clicking the button 'Receivers' where you can select the button Error -> 'Long text' to get more information. It is also possible that the BDoc message passes the outbound m-flow successfully but throws an error afterwards. Then the error notification is provided in the outbound s-flow to the receiver, e.g. a technical error during update of the CDB.

Receiver specific error information can only be written for outbound messages, i.e. it can only be written for receivers of the message, because they have to get it.

2.2.2 BDoc message analysis procedure

For the analysis you have to choose the appropriate selection criteria when you call Transaction **SMW01** and then focus on specific detail levels:

- *Send / Update date or interval:* if you want to know the overall status of the data exchange within the middleware, enter the required send / update date or the interval (date and time when the message was sent / updated) and start 'Execute'. This information can also be retrieved from the monitoring cockpit.
- *BDoc message ID*
- *User (the person who created the message).* The user can either be a dialog user who works in CRM Online transactions and creates outbound messages, or can be an RFC user (e.g. between R/3 and CRM, internal RFCs or R&R user), or can be the Mobile employee (GUID of SMOMITABT-SFAMITABT)
- *Flow context*
- *Message validation type*
- *The identification of the sender and receiver site*
- Other selection criteria are possible for particular situations, that won't be described in this document.

For example, if you are only interested in BDocs in error status, type in 'E*' in the field 'BDoc state' (or check 'errors' in the selection screen of CRM 4.0). Be aware, that also BDoc messages that stay in an intermediate ('I*') or outbound ('O01') state for a longer time indicate an error situation.

As a nice alternative or even a preferred entry point for monitoring, you can use transaction **SMW02**, which summarizes BDoc messages and gives a good overview about the amount of BDoc messages as combination of BDoc type, state, flow context and receiver (site ID). By double-clicking on a summary line you will see the single BDoc messages by calling transaction SMW01.

For regular monitoring it is recommended to setup a selection variant which includes all BDoc states not equal F* (that is all non-final messages), e.g. by excluding pattern 'F*'.

BDoc Type (Generation Name)	<input type="text"/>
BDoc State	<input type="text" value="F*"/>
Flow Context	<input type="text"/>

Alternatively you can use transaction **SMW02A** which summarizes BDoc messages according to the inbound/technical or receiver errors, message class and message number. Here you can easily identify which BDoc message belong together by means of having the same error situation. Once you identified and solved a specific error message, you can select all BDoc messages having this same message in their error segments and reprocess them together.

2.2.3 Analysis of erroneous BDoc messages

The status information shown for each BDoc message provides the most important information for the administrator, as listed in table below:

BDoc Status	Flow Context	Details
E01		<p>Description</p> <p>Technical error. If any of the flow services fail a technical error is raised. Possible reasons for that error are:</p> <ul style="list-style-type: none"> - Data inconsistencies (e.g. between CRM and CDB) - Implementation and programming errors - Missing functionality (implementation problems)
	Inbound s-flow (S11)	<p>Detection: Select BDoc message, click button 'Errors' and 'Classical Data' Possibly the inbound s-flow definition (transaction SMO8FD) is not correct (services do not exist or have syntax error) or the rejection handling in case of validation errors is not correct.</p> <p>Solution: - If no validation errors and no receiver errors exist, retry to process the message. Otherwise (if no retry possible) correct the error, send the changes from the sender site (usually Mobile Client) again if possible or request the data from the leading system to the different receivers and mark the original failed BDoc message to "deleted".</p>
	Inbound m-flow (MI0)	<p>Detection: Select BDoc message, click button 'Errors':</p> <ul style="list-style-type: none"> - No validation function defined for BDoc type - Technical error occurred: Service (usually validation service), BDoc type <>, BDoc <> <p>Solution: Check in table SMW3BDOCIF if the validation module is entered for the BDoc type. If no retry is possible correct the error, request the message from the sender system and set the original failed BDoc to "deleted". If the error occurs during the call of the validation module, start the debugging mode, restart the BDoc processing and put a breakpoint at the validation module. This requires knowledge in the implementation of the validation module (and application logic). Look for notes related to that function module, if this is a SAP standard function.</p>
Outbound m-flow (MO1)	<p>Detection: Select BDoc message, click button 'Errors':</p> <ul style="list-style-type: none"> - Technical error occurred: Service <>, BDoc type <>, Message <> - Service that caused the error: <> <p>Solution: Find a solution to the errors that occurred, for example through SAPNet Note search or opening a SAPNet problem message on the corresponding application component. The right application component can be found in the corresponding properties of the package where the service is implemented (use transaction SE37 to display the attributes of that function module). Possibly there is something wrong inside the Mobile Bridge or in the following services of the s-flow (like CDB update). Re-process the message (this is allowed for testing purposes; be careful in the case of a production system, especially if the data in the BDoc message is not current) or request the data from the sender or from the original system (this is usually possible for master data but not always for transactional data).</p>	

	Outbound s-flow	<p>Detection: Select BDoc message, select button 'Errors' Search for short-dumps Possibly something went wrong in the services of the s-flow (like CDB update).</p> <p>Solution: Correct the error, if possible and retry to process the message (this is allowed for testing purposes). Be careful in a productive environment: in this case request the data from the sender (CRM) to the receiver (CDB) using the request mechanism.</p>
E02	Description	Partially send, receivers have errors. The flow determines the different flow services associated with the flow context. If no technical error occurred, the BDoc message state is set according to the receiver's state. The receiver is in an error state.
	Inbound s-flow	-- This situation does not occur at all.
	Inbound m-flow	-- This situation does not occur at all.
	Outbound m-flow	<p>Detection: Select BDoc message, click 'Receivers', Site type (example: SMOF_ERPSITE for R/3 backend) and site name (defined in the Admin Console) have the status 'Not processed' which results in this error state E02. Select 'Errors' for that receiver to find the reason for the incorrect processing.</p> <p>Solution: Solve errors and retry or request (using Root-ID or the business object ID from the extension data part of the BDoc message) from CRM Server to the receiver (Site name: an external system or R/3 backend). A typical error situation is a failed upload from CRM-Online to R/3. In the receiver-specific error segment you see the error messages which have been raised by the R/3 application.</p>
Outbound s-flow	<p>Detection: Select BDoc message, click 'Receivers', site type is SMW1 (Mobile Client). Site name is important; it is a mobile client site name. The state of the processing for this client is 'Error in outbound processing'. Reason for the error is displayed by clicking icon 'Errors' within the 'receivers' dialog. Details related to the error message are shown if you select 'Long text'.</p> <p>Solution: Solve errors and resend the data to the client. You can resend the data by reprocessing the BDoc message. However if the BDoc message is old, you should extract all the subscriptions related to that BDoc type (if data volume is low) for the mobile site or using the Root-ID to extract these single instances to the corresponding mobile client. Note 715387 permits the extraction of instances based on their Root-ID. As of CRM 5.0 you can extract single instances in transaction SMOELUTAB (lookup viewer).</p>	
E03	Description	BDoc cannot be read from DB. This error situation does occur if the BDoc message cannot be read anymore during its processing. This does not usually occur. Final state - no reprocessing possible.
E04	Description	BDoc validation error: during validation in the CRM application an error occurs. The reason could be incorrect customizing, a locked object or wrong application data.
	Inbound s-flow	-- not used, because if no mapping to m-BDoc message is provided, successful validation is assumed – for mobile s-BDoc only. If a mapping to m-BDoc occurred and the validation failed, a rejection is triggered, the inbound s-BDoc will get status F01 and an outbound rejection flow (context SO2) is started.

	Inbound m-flow	<p>Validation provided by service (CRM application server)</p> <p>Detection: Select the BDoc message, click on 'errors'. It shows the name of the validation function module and the error messages that have been raised by the CRM Online application.</p> <p>Solution: Solve the errors occurred, e.g. by correcting customizing, and reprocess the BDoc message. If reprocessing is not possible, start request for the business object included in the BDoc message, set the messages to processed. Requests can be started using the adapter framework request mechanism (transactions R3AR2, R3AR4, R3AR3). It is possible to debug the BDoc processing by setting a breakpoint at the validation module (module name can be found using transaction SMO8FD and BDoc type name) and then to restart the processing in SMW01.</p>
	Outbound m-flow	-- no validation errors possible
	Outbound s-flow	-- this error status does not occur in s-flow Validation errors in CRM application result into rejections in the s-flow with status F01, see chapter "Analysis of final BDoc states".
E05 (as of CRM 4.0)	<p>Description</p> <p>Inbound processing failed This status is set by the watchdog program RSMWFLOWWATCHDOG (transaction SMW3WD) for BDoc messages created in inbound processing. If these BDoc messages remain in intermediate state for a long time (time interval to be defined in the watchdog program), they are set to E05. Then it is possible to recognize them as error situations. For outbound BDoc messages refer to status E06</p>	
E06 (as of CRM 4.0)	<p>Description</p> <p>Outbound processing failed This status is set by the watchdog program RSMWFLOWWATCHDOG (transaction SMW3WD) for BDoc messages created in outbound processing. If these BDoc messages remain in intermediate state for a long time (time interval to be defined in the watchdog program), they are set to E06. Then it is possible to recognize them as error situations. For inbound BDoc messages refer to status E05</p>	
E07 (as of CRM 4.0)	<p>Description</p> <p>Conversion error This error situation occurs in case of using the XML data synchronization between CRM Server and Mobile Clients. This data exchange format is usually used for Unicode CRM Servers. The data extracted from the CDB is then transformed to XML data. If this transformation is not successful (this happens while calling the mobile outbound adapter within the outbound s-flow), a BDoc message is created in status E07.</p>	
E08 (as of CRM 5.0)	<p>Description</p> <p>Mapping error (replacement for status F05) This error status is always set if the data is transferred from a source system and an error occurs when mapping the BAPI structure to the corresponding BDoc. Caution: Usually the BDoc is incomplete and therefore must not be started either. Therefore no reprocessing is possible. However, the transaction that executes the mapping is rolled back into the respective qRFC inbound queue (showing queue status SYSFAIL). And this can be started again per SMQ2 after you have corrected the cause for the mapping error. Before release 5.0, the status F05 has been set in this case. However, BDocs with this status are easily overlooked. BDocs with the new status E08 are reorganized as a final status (they can also be set to deleted per SMW01 and then change into status F03). Solution: Correct the error situation and restart the corresponding inbound queue.</p>	

E09 (as of CRM 4.0/5.0 with note 835761 and for CRM 3.X with note 876855)	Description Update failure At an update failure after a successful validation, the inbound message is typically already in a final state F02 and the outbound message in intermediate state I04. With the report "Z_PROCESS_BDOCS_I04" from SAP notes 876855 (CRM 3.X) or 835761 (CRM 4.0/5.0) the inbound s-BDoc was converted into an error state E09. Advantage is, that in this state the inbound message does not get reorganized. More important, it also allows you to restart the inbound processing after the error situation has been resolved. However, in that case you need to delete the already pending update tasks from the previous run in transaction SM13. The previous outbound m-BDoc in state I04 gets state F03 and is obsolete.
---	---

If you solved the problem by requesting the message after correcting the error:

Do not forget to mark the BDoc message as 'deleted' as soon as you have solved the problem and started a request. This results in deleting the original failed BDoc message if the next background job "SMO6_REORG" (for CRM 3.X) or "SMO6_REORG2" (as of CRM 4.0 SP6 or note 713173) runs (the system consistency was established by requesting the object newly and deleting the failed one).

Further information: SAP Note 526853 contains some tips on how to deal with BDocs in error state.

2.2.4 Analysis of intermediate BDoc states

BDoc messages in intermediate state (I01, I02, I03 and I04) are created while the Middleware flow is processing the BDoc changes in CRM Online, to the R/3 Backend, in the CDB or to and from the different data receivers (Mobile Clients, Groupware Server, external system). As long as they are no short dumps or coding errors, the state of these BDoc messages should change after a short period of time to a final state (F* or E* state).

Pay special attention to the connection between a BDoc message in the flow (as seen in transaction SMW01) and the corresponding entry in a qRFC queue (as seen in transactions SMQ1 and SMQ2). Usually the intermediate states are related to the storage of a BDoc message in a qRFC queue. The queue is always processed asynchronously and can therefore create delays in the BDoc processing.

BDoc Status	Flow context	Details
I01	Description	<p>Received (intermediate state).</p> <p>The message reached the CRM system in the qRFC-queue. This means the data is posted into the inbound queue for inbound processing. A short dump (e.g. caused by a coding error or timeout) occurred. The queue entry will be deleted automatically.</p>
	Inbound s-flow	<p>Detection: Check short dump related to the validation module in the inbound processing. You can also reprocess the BDoc message to reproduce the short dump.</p> <p>Solution: After having removed the cause of the problem, you can reprocess the BDoc message (if you can guarantee that these changes contain current data and won't overwrite newer changes). If the BDoc message is old, you can decide to extract the same instance (using the Root-ID) from the CDB and send it to the corresponding Mobile Client (note 715387 contains a description how to proceed). In this case the changes done by the Mobile Client are removed and replaced by the data as they are stored in the CDB. You can also check manually if the BDoc changes contained in this BDoc message are already in CRM Online and then adjust the data in CRM Online manually or reprocess the messages or mark it to 'deleted'.</p>
	Inbound m-flow	<p>Detection: Check SMW01 for status, check short dump (ST22)</p> <p>Solution: Correct the problem and reprocess the message. Same procedure as for inbound s-flow.</p>
	Outbound m-flow	<p>Detection: Check SMW01 for status</p> <p>Solution: Occurs only in very special cases, if a commit occurred in the asynchronous step. Open a SAPNet problem message, after solving do a request load for the affected objects.</p>
	Outbound s-flow	<p>Detection: Check SMW01 for status</p> <p>Solution: Occurs only in very special cases, if the s-flow is called with a commit. Open a SAPNet problem message, after solving start a request from CRM Online to the CDB for the affected objects.</p>

102	<p>Description</p> <p>Written to qRFC Queue (intermediate state). An entry is created in a qRFC queue to enable processing of the m-BDoc message in the receiver system. Such queues start usually with the prefix CSA*. They store the notification message created by the application, until it gets asynchronously processed by the Middleware. In CRM 3.0/3.1 this queues are either outbound queues (standard delivery) or inbound queues (possible with certain support package level). As from CRM 4.0 the CSA* queues are always qRFC inbound queues. Although logically it is an outbound flow, technically inbound queues are used because they can be scheduled more flexible which is better for performance reasons.</p>
	<p>Inbound s-flow</p> <p>-- Not relevant for inbound s-flow</p>
	<p>Inbound m-flow</p> <p>-- Not relevant for inbound m-flow</p>
	<p>Outbound m-flow</p> <p>For a short time this BDoc status is perfectly normal as long as it can be found in the corresponding queue (double-click the queue name for the BDoc message and you will display the queue). If it remains for a prolonged period, it has to be analyzed.</p> <p>Reason a) Outbound/inbound queue in CRM System is stopped or deregistered, or its processing is very slow. It is also possible that the queue has been deleted manually. If you have used the watchdog report, the state will change after a certain time interval automatically to E06. Detection: <ul style="list-style-type: none"> - Queue stoppages can be caused by manually locking a qRFC queue (status STOP in SMQ1/SMQ2) - by queue failures (status SYSFAIL in SMQ1/SMQ2) - by unregistered queue (status READY in SMQ1/SMQ2 but no processing occurs) - or by special circumstances, such as mass queues (e.g. CSA_MASS_BUPA) waiting for higher priority delta queues to process, that is queues depending on each other - Queue does not exist anymore because deleted manually Solution: analyze suspicious outbound/inbound queues using SMQ1/SMQ2. Check the registration in the outbound and inbound queue scheduler using SMQS and SMQR. Check the activities described in note 624921. If this, and SAP Note search does not solve the problem, open a SAPNet message under component BC-MID-RFC.</p> <p>Reason b) If the inbound queues are empty, it is likely that someone has deleted the queue content manually Detection: Check SMQ2 for Queues or System log (SM21) for deleted queue entries. Solution: Select classical and extension data. If the BDoc state is not equal to "<> can no longer be read due to structure changes" then find out the object instances involved in this BDoc message, request them from the reference system and then set the BDoc message to deleted. If queue is not empty -> wait until queue is processed!</p> <p>Before opening a SAPNet message, please check if note 527481 or 574774 applies, and ensure that qRFC resources are configured according to note 74141 and that the latest qRFC version has been obtained as described in note 438015.</p> <p>In CRM 3.0, if you have not yet implemented the note 529764 (use of inbound queues for processing the CSA queues), start the outbound queue monitor (SMQ1) and enter the queue name; otherwise (CRM 3.0, SP12 or higher, CRM 4.0/5.0) start the inbound queue monitor (SMQ2) and enter the queue name. If no entries are displayed, then the queue entry has been deleted manually. If you are sure that the BDoc message has been created</p>

		<p>some minutes ago and contains the last updated information related to the business object, you can retry it and the BDoc message is reprocessed directly (without asynchronous step in a CSA* queue). If the BDoc message is old and you are not sure that it contains the latest update, then you should request the whole business object from CRM to CDB, from CRM to R/3 or from R/3 to CRM to ensure the data consistency. It depends if you are dealing with master data or transaction data. Master data can be modified in all the systems and could be requested from any source system to another target system. Starting requests for transaction data is critical, as there are some business objects that cannot be requested if they belong to a specific scenario or have a particular original system.</p> <p>If the queue does exist and contains some entries, one of them could be assigned to your BDoc message. Check whether this CSA queue is not running and restart it if required. To find out the corresponding queue entry, use the send time from the BDoc header and compare it with the queue entry '1.date' and '1.time'.</p> <p>If the queue does exist and contains some entries, one of them could be assigned to your BDoc message. Check whether this CSA queue is not running and restart it if required. To find out the corresponding queue entry, use the send time from the BDoc header and compare it with the queue entry '1.date' and '1.time'.</p>
	Outbound s-flow	-- Not relevant for outbound s-flow
I03	<p>Description After qRFC step (intermediate state). The outbound flow processing of the m-BDoc is started and the entry in the inbound queue is removed.</p>	
	Inbound s-flow	-- Not relevant for inbound flow
	Inbound m-flow	-- Not relevant for inbound flow
	Outbound m-flow	<p>Detection: Check the Queue for uploading the data to the R/3 Backend system (R3AU*) or Groupware (ISP*), check if there are any short dumps related to the processing of the BDoc messages (dumps could have occurred in the Mobile Bridge or in the R/3 outbound adapter)</p> <p>Solution: Analyze the dump or error situation to determine the cause. Once this has been corrected you can reprocess the message. Before reprocessing, make sure that there are no more recent changes in the application object (i.e. by looking into the change documents).</p>
	Outbound s-flow	-- Not relevant for outbound s-flow
I04	<p>Description BDoc stored before update task (intermediate state). BDoc Message was not written to qRFC queue, yet. Search for update tasks in transaction SM13. The sender creates a BDoc message to post data to the receivers. The application calls the messaging flow.</p>	
	Inbound s-flow	-- Not relevant for inbound s-flow
	Inbound m-flow	-- Not relevant for inbound m-flow

Outbound m-flow	<p>Detection: Check if there are any update terminations which occurred in the update task by using transaction SM13. Check if there are inbound messages related to that outbound message, double-click 'Original BDoc' field to display the original inbound message. Typically the inbound message (s-BDoc in flow context SI1 or m-BDoc in flow context MI0) is already on a final state F02, i.e. the validation was successful but the asynchronous update failed. The original inbound message in final state might have been already reorganized.</p> <p>Solution: Solve the problem with the open update tasks and then execute them again in SM13. Check whether after successful reexecution in SM13 the BDoc state has been changed. If not, or if you don't find any open updates, but you do find a corresponding original message, proceed as following:</p> <ul style="list-style-type: none"> • Variant 1: Try to reprocess the inbound message to check if the newly created outbound message has also status I04. If BDoc is processed, no retry of the failed BDoc is allowed. Set the message as deleted. Reprocessing the related inbound message is sometimes not possible, because the state is final. In this case copy the inbound message using transaction SMW19 and set the new state of the copied message to I01. Then you can reprocess it. • Variant 2: Use the solution from SAP notes 876855 (CRM 3.X) or 835761 (CRM 4.0) to convert the inbound s-BDoc into an error state E09. Advantage is, that in this state the inbound message does not get reorganized. More important, it also allows you to restart the inbound processing after the error situation has been resolved. However, in that case you need to delete the already pending update tasks from the previous run in transaction SM13. <p>When there are no open update tasks anymore, check whether the data changes according to the inbound BDoc message have actually been posted to the CRM online application tables. If the changes are present (i.e. a new application object was successfully created), you should not use variant 2, because you might get a problem of posting the same data again (i.e. getting an "insert duplicate record" situation).</p>
Outbound s-flow	-- Not relevant for outbound s-flow

2.2.5 Analysis of final BDoc messages

All successfully processed BDoc messages in final status will be deleted from the system by the middleware reorganization job SMO6_REORG or SMO6_REORG2 (refer to note 713173 to get details about both reorganization jobs). Default setting is after seven days.

BDoc Status	Flow context	Details
F01	Description Rejected (fully processed)	
	Inbound s-flow	Detection: Validation error of an update coming from the mobile client on the CRM Application that leads to a rejection to the same Queue (Receivers). Select BDoc message, click 'Receivers' (to which mobile client is it related) and 'Errors' and find out reason for the error. Mobile users find the rejection message in the Inbox tileset and must reenter the changes and synchronize the data. This inbound BDoc message will create another outbound message (Context SO2 and state F04).
	Inbound m-flow	-- Not relevant for inbound m-flow
	Outbound m-flow	-- Not relevant for outbound m-flow
	Outbound s-flow	-- Not relevant for inbound m-flow
F02	Description Confirmed (fully processed) The validation of the data in CRM Online is successful and the CRM Online tables are updated. The queue name displays the inbound queue which was used to process the data on the CRM Server. The sender site name is useful to know the originator of the message. If the request BDoc Message ID is filled, that means that this BDoc message (MI0) originated from an outbound message from CRM Server and is a response to a request sent by the CRM Server. This response created an outbound message; it also calls the synchronization flow. If the request BDoc ID is empty, this means that this BDoc message contains data sent by an external system (R/3 Backend).	
	Inbound s-flow	Validation successful. Always true if no mapping to m-BCDoc.
	Inbound m-flow	Validation in CRM application successful
	Outbound m-flow	Notification, direct send or initial load has been sent successfully to one or more receivers
	Outbound s-flow	Notification, direct send or initial load has been sent successfully to one or more Mobile Sites
F03	Description Set to processed (fully processed) After setting a BDoc message manually to processed, it can not be restarted anymore. For inbound processing this usually means that the CRM Server application did not receive the update of an application object. For outbound processing the receiving application (like R/3 or Mobile) did not receive the update of the CRM Server application. Thus an inconsistency exists between sender and receiver, which must be corrected, e.g. by recreating/resending the change or by requesting the application object in its current state.	
	Inbound s-flow	Manually set to processed (was in error or intermediate status)
	Inbound m-flow	Manually set to processed (was in error or intermediate status)
	Outbound m-flow	Manually set to processed (was in error or intermediate status)
	Outbound s-flow	Manually set to processed (was in error or intermediate status)

F04	Description Confirmed (fully processed by all receivers). The flow determines the flow services associated with the flow context. If no technical error occurred, the BDoc status is set according to the receiver's state. If all the receivers are in success state or no receivers exist, the BDoc message state is set to F04. This enables qRFC queue LUWs.	
	Inbound s-flow	-- Not relevant for inbound flow
	Inbound m-flow	-- Not relevant for inbound flow
	Outbound m-flow	Notification to multiple receivers
	Outbound s-flow	Notification to multiple receivers
F05	Description Information (no processing)	
	Inbound s-flow	Detection: Select BDoc Message, click 'Errors', read the long text
	Inbound m-flow	Detection: Select BDoc Message, click 'Errors', read the long text Can happen if the validation detects an error situation that does not allow a reprocessing (e.g. mapping error or filtering problem). Instead of state E04 the BDoc message gets state F05. There might be a qRFC inbound queue in state SYSFAIL as well. When restarting the queue entry, a new BDoc message will be created. As of CRM 5.0 the state F05 is converted into state E08. To prevent that for temporary problems (like locked application object) you can implement note 958552.
	Outbound m-flow	Detection: Select BDoc Message, click 'Errors', read the long text
	Outbound s-flow	Detection Select BDoc Message, click 'Errors', read the long text

2.2.6 Analysis of particular non-final BDoc states

BDoc Status	Flow context	Details
T01 (3.0) / R01 (4.0/5.0)	Description These states are used to cover specific situations and especially to avoid automatic rejections to mobile clients. T01: Temporary lack of resources in application layer , refer to note 594388 R01: Retry after temporary error , refer to note 653301	
	Inbound s-flow	Reason: the s-BDoc message is mapped to a m-BDoc message (which is not stored in the BDoc message store and cannot be viewed in SMW01), then the internal m-BDoc message is passed to the validation module of the CRM application. The validation can lead to a temporary error situation. Here are some examples: Reason a): "Processing of document with GUID <> is canceled" Reason b): "Transaction <> is being processed by user <>" or "Product is locked by user <>" The user got a lock on the 'object' in CRM when a change was made in mobile client. Solution: Select BDoc message, then 'errors'. Unlock the corresponding transactions or objects and repeat the processing, if the queue still exists. In CRM 4.0 the reprocessing is done automatically according to a specific customizing entry described in note 653301. This retry won't cause any inconsistencies as the BDoc messages are retried many times and the other subsequent changes are waiting in the inbound queue CRM_SITE*. Note: In CRM 3.0 and 3.1 the state T01 can also occur when the inbound logic according to note 594388 has been applied. Instead of starting a rejection flow after a validation error for an inbound s-flow, the inbound BDoc message can be forced into a T01 state. This enables reprocessing on server side instead of rejecting the message and sending it back to the Mobile Site.
	Inbound m-flow	Relevant for m-flow but BDoc messages are displayed in inbound s-flow (see above)
	Outbound m-flow	-- Not relevant for outbound flow
	Outbound s-flow	-- Not relevant for outbound flow
O01	Description Sent to receivers (not all have confirmed). The remote system must still return an inbound status to update the receiver's state in the sender system. This status is set if no receiver in error state exist, but there is a receiver in 'processing state' (detailed description of the receiver state follows)	
	Inbound s-flow	-- Not relevant for inbound flow
	Inbound m-flow	-- Not relevant for inbound flow

	Outbound m-flow	<p>Detection: Check the RFC user and connection in R/3 Backend. Are there short dumps or other error indicators on the receiving system? Check the whole loop (outbound qRFC queue in CRM, short dump or update termination in R/3, outbound qRFC queue in R/3 and inbound qRFC queue in CRM for the created delta download).</p> <p>Solution: Select BDoc Message, go to 'Receivers', Status: Sent. Check the inbound Queues from R/3 Backend system (R3AD*), check if they are registered and if they are failing for any reason. Check whether the delta events are active in the receiving R/3 system for the used object class (transaction R3AC4).</p> <p>Note: If the Mobile Bridge is active, the successful call of the Mobile Bridge function module counts as a confirmation from the Mobile scenario. There is no further feedback from the subsequent synchronization flow or even distribution to the mobile sites.</p>
	Outbound s-flow	-- Not relevant for outbound flow
D01	<p>Description To be processed (Debug) This status is set manually to enable debugging the BDoc messages in SMW01. This is usually done during the development phase of a new BDoc type or flow process and is not used in any productive system. Refer to note 432661 for details on how to set this status in the coding.</p>	
	Inbound s-flow	-- Not relevant for inbound flow
	Inbound m-flow	<p>Reason: Temporarily locked due to debugging.</p> <p>Solution: If the BDoc message is new and you can guarantee that it contains current changes, you can reprocess it. If this BDoc message was only used for debugging purposes and it not relevant anymore, you can set the state to 'deleted' (ensure that no data inconsistencies are possible).</p>
	Outbound m-flow	<p>Reason: Temporarily locked due to debugging.</p> <p>Solution: If the BDoc message is new and you can guarantee that it contains current changes, you can reprocess it. If this BDoc message was only used for debugging purposes and it not relevant anymore, you can set the state to 'deleted' (ensure that no data inconsistencies are possible).</p>
	Outbound s-flow	-- Not relevant for outbound flow

Messages in non-final status must be processed (mark as 'deleted' or reprocess) according to the situation. The main target is not to affect the consistency of the whole system landscape (take into account the sequence of changes). Such inconsistencies are very difficult to detect at a later stage in the production environment.

Note: Here is additional information related to the processing state of the message for the receiving site. As of CRM 4.0 the following states are allowed:

- **Sent:** Message has been sent to the receiver. Waiting for a response. See above for status O01 in outbound m-flow.
- **Processed:** Message has been sent and successfully processed by the receiver.
- **Error in outbound processing:** Message could not be sent to the receiver due to errors in outbound processing (mapping, configuration, technical errors).
- **Not Processed:** The message has been sent to the receiver but the receiver could not process the message (error situation).

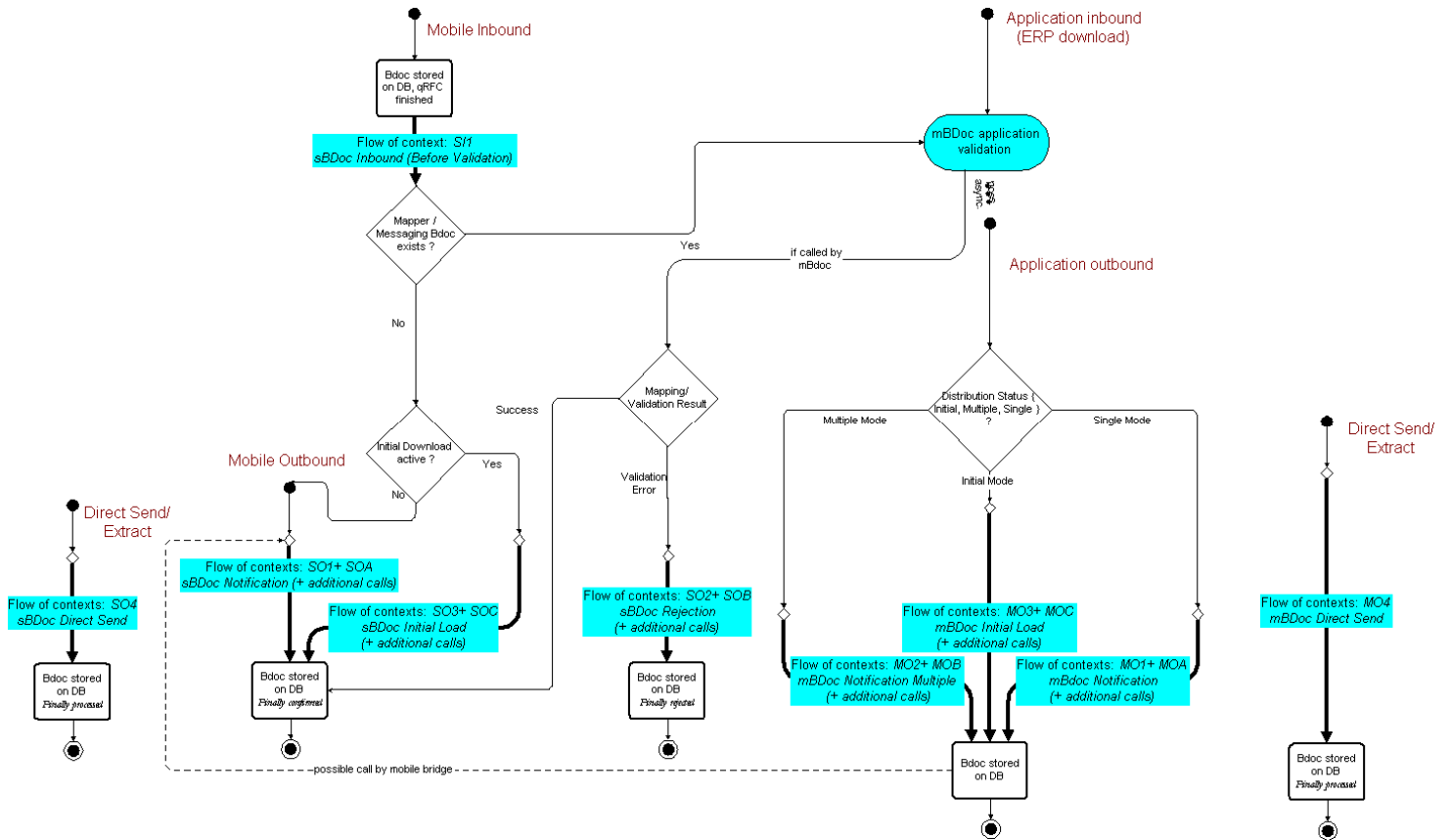
2.2.7 Appendix 1: Classification of BDoc States

BDoc State	BDoc State Description	Final state	Error state	Retry allowed
D01	To be processed (Debug)			X
E01	Technical error (incomplete)		X	X
E02	Partially send, receivers have errors		X	X
E03	BDoc cannot be read from DB	X	X	
E04	BDoc validation error		X	X
E05	Inbound processing failed		X	X
E06	Outbound processing failed		X	X
E07	Conversion error		X	X
E08	Mapping error	X	X	
E09	Update failure		X	X
F01	Rejected (fully processed)	X		
F02	Confirmed (fully processed)	X		
F03	Set to processed (fully processed)	X		
F04	Confirmed (fully processed by all receivers)	X		
F05	Information (no processing)	X		
I01	Received (intermediate state)			X
I02	Written to qRFC Queue (intermediate state)			X
I03	After qRFC step (intermediate state)			X
I04	BDoc stored before update task (intermediate state)			
O01	Sent to receivers (not all have confirmed)			X
R01	Retry after temporary error		X	X
T01	Temporary lack of resources in application layer		X	X

2.2.8 Appendix 2: Classification of Flow Contexts

Con-text	Flow Context Description	s-Flow	m-Flow	In-bound	Out-bound	Standard	Custom
SI0	sBDoc Validate	X		X		X	
SI1	sBDoc Inbound (Before Validation)	X		X		X	
SO1	sBDoc Notification	X			X	X	
SOA	sBDoc Notification (additional calls)	X			X		X
SO2	sBDoc Rejection	X			X	X	
SOB	sBDoc Rejection (additional calls)	X			X		X
SO3	sBDoc Initial Load	X			X	X	
SOC	sBDoc Initial Load (additional calls)	X			X		X
SO4	sBDoc Direct Send	X			X	X	
MI0	mBDoc Validate		X	X		X	
MO1	mBDoc Notification		X		X	X	
MOA	mBDoc Notification (add'l calls)		X		X		X
MO2	mBDoc Notification Multiple		X		X	X	
MOB	mBDoc Notification Multiple (add'l calls)		X		X		X
MO3	mBDoc Initial Load		X		X	X	
MOC	mBDoc Initial Load (add'l calls)		X		X		X
MO4	mBDoc Direct Send		X		X	X	
MO5	mBDoc Post Request		X		X	X	
MO6	mBDoc Post Rejection		X		X	X	
MOF	mBDoc Post Rejection (add'l calls)		X		X		X

Usage of Flow Contexts:



(This graphic can be viewed in your CRM system by running program RSMWDISPLAYCNTXT)

3 Display unprocessed BDoc message summary

Functional description

Display the total number of unprocessed BDoc messages, grouped by BDoc type and client. You can use this information to check the cross-client system status for monitoring purposes, or before you install a support package.

Usage

Transaction: **SMW03**

Menu path:

CRM 4.0/5.0: *Architecture and Technology → Middleware → Monitoring → Message Flow → Display unprocessed BDoc Message Summary*

CRM 3.0/3.1: *Middleware → Monitoring → Message Flow → Display unprocessed BDoc Message Summary*

3.1 Handling procedure

Proceed as follows:

- Enter a client, BDoc type, and BDoc status.
- Choose an existing ABAP List Viewer (ALV) layout and click Execute.

There are different reasons for unprocessed BDoc Messages:

- Wrong customizing
- Missing 'required' fields
- s-BDoc, m-BDoc errors
- RFC connection problems
- Middleware specific problems
- Application specific problem related to data or to customizing

The resolution of unprocessed messages is based on the following actions:

- Solve the problem and reprocess the message
- Reprocess the BDoc message
- Set to processed
- Request the data from the reference system and set message status to processed

What are the possible actions to put the “unprocessed” messages into a final state?

Function	Description
Mark message as deleted/processed	Deletes a message, SMW01
Retry to process message	Processes message further, SMW01
Define, start and monitor single requests for load	R3AR2, R3AR4, R3AR3
Synchronize objects	R3AS4, R3AM1 (for customizing objects only)
Process aborted BDoc Messages	SMW20 This transaction (report RSMWAPP01) allows a mass re-processing of erroneous BDoc messages. However, it does not check for logical correctness (e.g. overtaking of older changes). So you should rather reprocess BDoc messages manually in a controlled way.
Unlock data in source or target system and repeat	CRM Application

Which tools/functionalities are available for analyzing the problem?

Receivers	SMW01: Displays a list of the receivers of a message
Errors	SMW01: Displays error segments
Middleware Trace	SMW01: Displays records for individual steps
Classic Data	SMW01: Data segments of a message
Extension Data	SMW01: Extension of a message (only for m-BDoc messages)
Outbound queues	SMQ1 (see also outbound queue scheduler SMQS)
Inbound Queues	SMQ2 (see also inbound queue scheduler SMQR)
Mobile clients In- and outbound Queues	SMQ1, SMQ2
Short Dumps	ST22 (+- 15 minutes), select update User
Links	SMW01: Links between messages and object types
DIMA	SDIMA: Compare and adjust data between two data sources
Update requests	Transaction SM13 to check if there are any hanging update requests
System log	SM21: Check for "User xxx is deleting inbound/outbound queue xxx" to see if queues were manually deleted.

4 Examples

Example 1:

You want to create a new Business partner as employee.

Select messages from SMW01 by entering the send date and/or dialog user who created this business partner.

Message in m-flow is the original m-BDoc passed to the CRM application. Select the m-BDoc message and click button 'Links' to get the list of corresponding s-BDocs. s-BDocs are created only if the mobile bridge is active.

In case the state of the m-BDoc is E02 (not all the receivers have confirmed), select BDoc message and click button 'Receivers' to get details of the error. The flow controller is waiting for the confirmation from one or more receivers. The site name and the RFC destination of the R/3 Backend system are then displayed with the corresponding state. If the state is 'Sent' possible reason is that the RFC Destination does not work.

The column "Original BDoc message" shows for each s-BDoc message the corresponding m-BDoc message. In CRM 3.0/3.1 it is possible to choose this field to show the original BDoc message. In CRM 4.0/5.0, it is necessary to change the layout (by adding the column to display the Original BDoc message).

In column "Reference BDoc message" the message ID of the predecessor message is shown.

Column "Root ID" contains the key of the BDoc instance, if the message contains information on exactly one BDoc instance.

To solve the problem you can use one of the following methods:

1. Select the BDoc message, then press the button 'Receivers' and retry the processing for the waiting receivers or
2. Define and start a request from CRM to the R/3 Backend System to upload the data object defined by the field Root ID or get the business partner number from the classic part of the data.
3. If the message is not current anymore and you know that it is solved already (by a request or initial load), mark the BDoc message as deleted. To get detailed information on the BDoc specific flow processing, select the BDoc message and press the button 'Middleware Trace'. To enable tracing the flow with the highest trace level, define the following middleware parameters in transaction SMWTAD for CRM 4.0/5.0 or refer to note 206439 for CRM 3.0/3.1 (high trace level has a negative impact on the overall performance, reset it after having finished your analysis):

Environment	Message Flow
Trace-Level	Trace-Level 2

Example 2:

BDoc Type: BUPA_MAIN

Status: E02, Context: MO1

Select BDoc message, then press button 'Links', all the corresponding s-BDocs are displayed.

Select BDoc message, then button 'receivers', if the receiver's state is "sent" that means that the CRM Server is waiting for a confirmation from this consumer (receiver). Start SMQ1 and select destination name, a queue R3AUBUPA<id> is there and could be in error status CPICERR:

1. Restart the queue after the reason of problem is solved (adjust RFC connection and RFC user and password!)
2. If no queue does exist, that means that it was deleted manually. Define and start a request to get the current status from CRM Server to R/3 or retry the message processing if you are sure that this is the last change or for test purposes.

Select BDoc message, then press button 'Receivers', if the receiver has the state: "Error outbound in processing", select button 'errors' to get detailed information and implement the solution to the problem. After solving the problem, retry to process the BDoc message or extract the data from the CRM Server and send it to the receiver. In this case do not forget to delete the BDoc message.

Example 3:

BDoc Type: CRM_DNL_ACT_H Status: E02, Context: SO1

Select BDoc message, then press button receivers, the site name, site ID and state error "outbound processing" are displayed, Press 'Errors' to get the message "Sending of an empty message is not possible", press 'long text' that contains the following instructions:

Determine the component that sent the empty message. Report the error to the owner of that component. Determine what the message content should have been. Possibly the system, in which the change was originated, and the receiver system may be desynchronized. You need to synchronize the data between the systems.

If you have not done any changes to this BDoc type, open a SAPNet message to report that issue to SAP.

Example 4:

BDoc Type: ZCR550_10_MESG Status: E01, Context: MIO

Select BDoc message and press button 'Errors':

No validation function defined for BDoc type /CRMGEC/ZCR550_10_MESG. Check in table SMW3BDOCIF if any validation function is defined to this BDoc.

Example 5:

Validation error E04, m-flow inbound

Select BDoc Message, then press button 'Errors' and solve all the problems, sort the messages in the chronological order, then restart the processing of the messages or define and start a request for these BDoc types.

It may be useful to use debugging for this type of error.

Select BDoc message in SMW01

Enter /h, click 'reprocess BDoc message' and enter a breakpoint at the validation module, which causes the error and which is listed in the last line of the error segments. To find out the validation service name for a specific BDoc, start transaction SMO8FD, enter the BDoc name and execute. The validation module is the first service in context MIO. Validation services do only exist for messaging BDoc types, as synchronization BDoc types are first mapped to messaging BDoc types and then sent to the CRM Online validation. Usually you will need some application knowledge to find out the reasons for the errors during the validation.