Message Control (CA-GTF-BS)

HELP.CAGTFBSMC

Release 4.6C

Copyright

Icons

Icon	Meaning
⚠	Caution
💬	Example
➡	Note
🧭	Recommendation
<SYN>	Syntax
💡	Tip

Contents

Message Control (CA-GTF-BS)

Purpose

The output or follow up processing of partner-dependent messages is automated via Message Control. The application then calls the Message Control via the specified interfaces.

You can use these interfaces to describe various data combinations along with the required processing. If one of these data combinations appears during processing, the appropriate processing is started (synchronously or asynchronously). You can define processing with several different methods, for example as a workflow, print output or as an IDoc dispatch for EDI.

Implementation considerations

SD and MM applications always use Message Control for message output. You carry out Customizing of the Message Control in the application’s Customizing. The Message Control is also sometimes described here as “output control“.

Integration

Message Control is a "service program" for other applications. It is initiated via ABAP function modules. An application that uses Message Control must

- Call the Message Control
- Transfer files which should be compared with the combinations defined in Customizing.
- Provide a processed program, that is called from the Message Control, if the transferred data corresponds with the combinations defined in Customizing

Features

The condition technique in the R/3 System is used to formulate the data combinations required. If the current application data corresponds to the conditions defined in Customizing, **message determination** was successful and created a message default. If the application supports it, the proposal can be changed (**processed**).

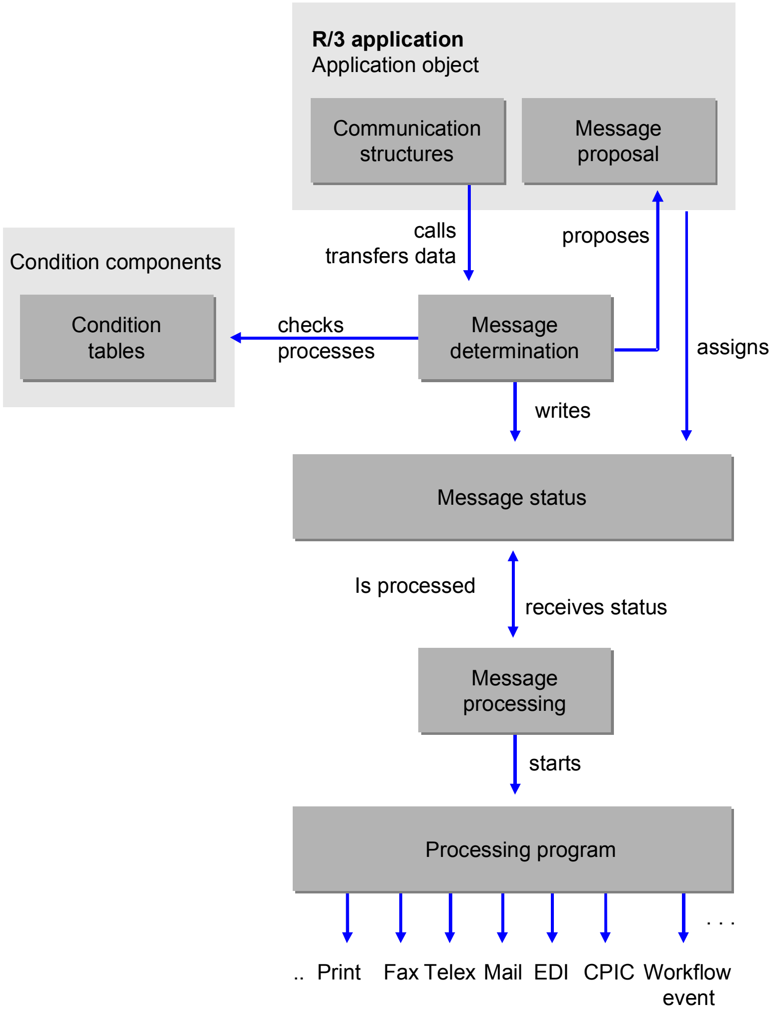
As a result the message is **processed** immediately or at a defined time. The processing status of the message is stored in a status record. If the application supports it, a processing log is written as well.

Message Control scenarios:

- The credit limit has been exceeded in a sales order, whereupon the block indicator is assigned. The Message Control should respond to the block indicator by printing out this customer’s account history.
- A document is released and receives a corresponding status in its data record. The Message Control should thereupon issue a release message.

There are also applications which can “switch off” message determination and directly generate the message default.

Message Control and connection to the R/3 application



See also:

Message Control Structure [Page 9]

The section describes the Message Control in Detail.

Message Control Customizing [Page 54]

You determine the rules in Customizing by which the system accesses the individual condition records.

Define messages as condition records in the application [Page 81]

You define concrete messages as condition records in the application.

Message Control connection to the application [Page 90]

The section is aimed at developers who want to install Message Control in an application.

Constraints

The Message Control only supports specific transmission media. For more information, see Transmission media [Page 83]

Message Control Structure

Definition

Considered as a **process in the system**, Message Control is subdivided into several sections. For **configuration** it contains so-called condition elements. Combined, this is the Message Control structure.

Use

Message Control is used by application programs to generate messages automatically from data sources (partners, company codes,...) and to send the messages at predefined times (for instance in paper format or as an e-mail).

Structure

Condition Components and Condition Records [Page 10]

The individual objects of the Message Control which are relevant in the message determination process and can be configured in Customizing or the application transaction are explained.

Message determination [Page 34]

This section describes how the condition component hierarchy is evaluated and how the condition tables are accessed. Different scenarios, which can be realized by special strategies when Customizing the condition component, are used as examples.

Message Default and Message Editing [Page 44]

The result of message determination is a message default that can be edited manually in the application, if this function is available in the application. If not, any messages found are forwarded directly for automatic processing.

Message processing [Page 45]

This explains how the message found can be processed.

Integration

Requirements for Applications [Page 91]

The Message Control is addressed by application programs via a defined interface. This section explains the various functions which must be available to the application.

Condition Components and Condition Records

Definition

The tool that Message Control uses for message determination is well known in other applications (for example price determination and account determination in SD): **condition technique**. The condition technique uses **condition components** that determine with which method and with which application data the message is searched for. The following two diagrams show the **hierarchy** for the condition components.

Condition Components: Sample Hierarchy [Page 11]

Condition Components: General Hierarchy [Page 12]

Structure

The individual condition elements are:

Application code [Page 13]

Field Catalog for Condition Keys [Page 14]

Procedure [Page 15]

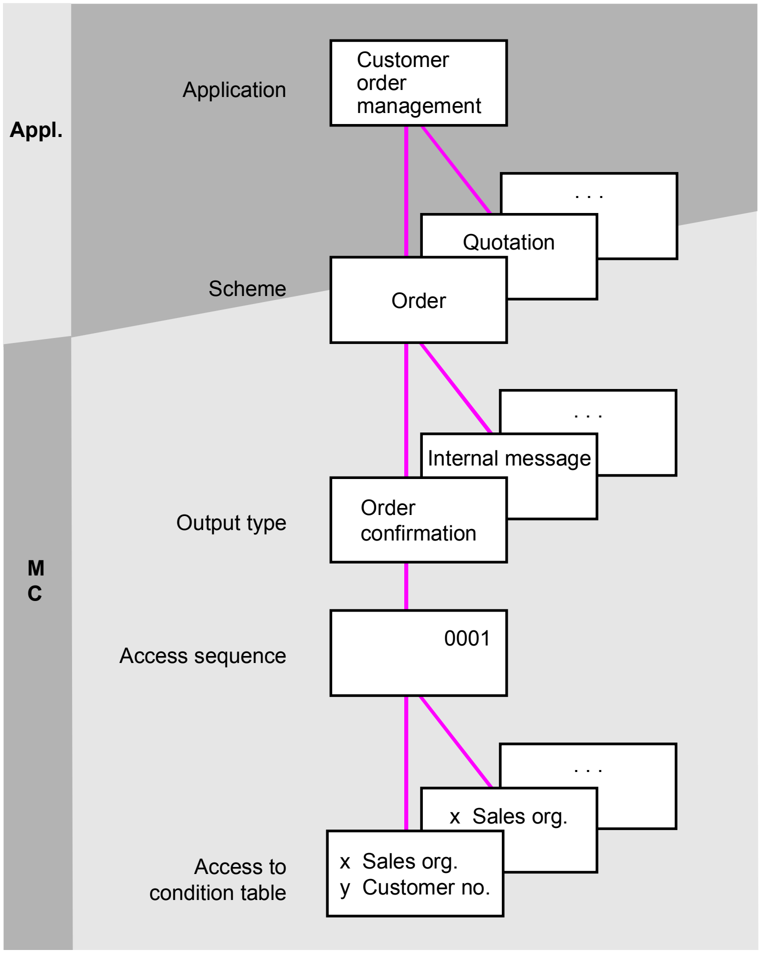
Output type [Page 19]

Access sequence [Page 23]

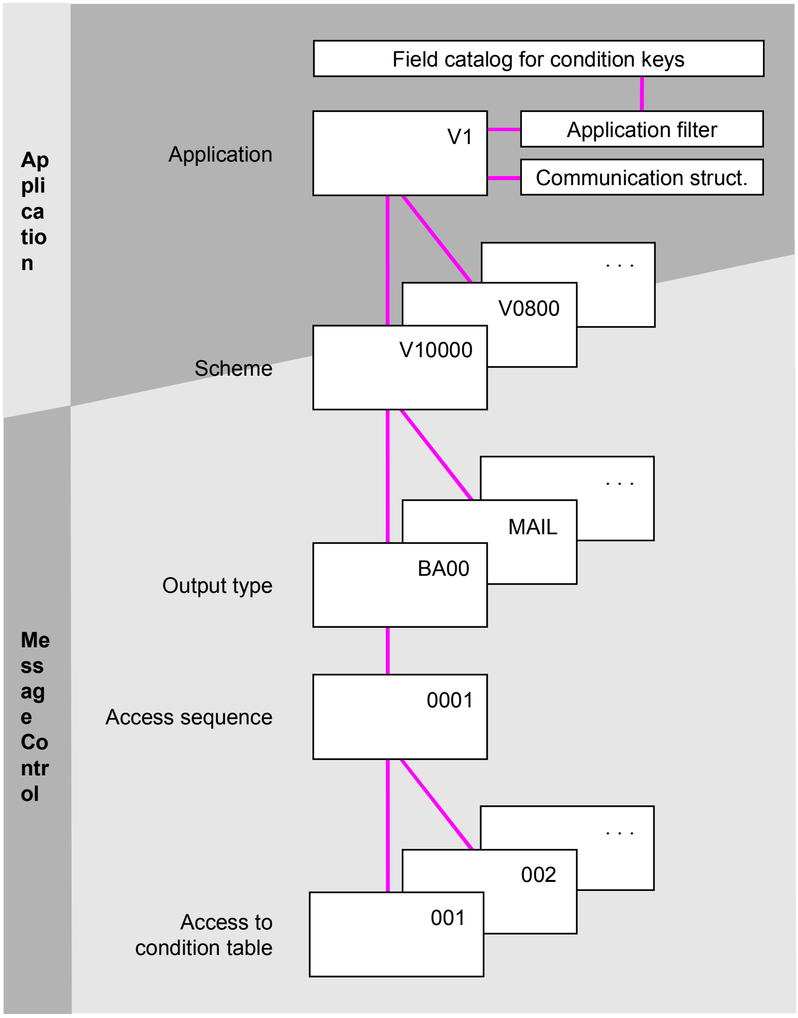
Condition table [Page 27]

Table access [Page 33]

Condition Components: Sample Hierarchy



Condition Components: General Hierarchy



Application Code

Definition

An application code is assigned to each individual application which uses Message Control (for example EF for ‘Purchasing’, V1 for ‘Sales’, MR for ‘Invoice Verification’).

Use

If an application calls Message Control, the application code is given first and the application can therefore be identified. The current application data is transferred in communication structures.

Field Catalog for Condition Keys

Definition

The field catalog contains all fields that can be selected as keys for Message Control condition tables. One **application filter** per application is created from this field catalog. This filter determines which fields can be selected for the relevant application (for example V1 for “Sales”).

Use

Defining the condition table

When creating a condition table [Page 56] you select the key fields from the “application filtered” field catalog which should be used in message determination for **table access**.

Defining the application filter

In order to define or extend the application filter, maintain Customizing table T681F: Use, for example, transaction NACE (*expert mode*) and choose *Edit* → *Field Catalog*. Select new fields from the communication structures [Page 92] for your application.

Extend field catalog

If you want to define a condition field for your application that does not yet exist in the field catalog, then before generation of the condition tables or the access sequences you must extend the field catalog (valid for all applications).

The field catalog for the message determination has the structure KOMB. This name is stored in table T681V. The uses of the condition technique (pricing, message determination, etc.) and their attributes are managed here. Every use of the condition technique has its own field catalog.

Procedure

Definition

A procedure (for example V10000) is a pool of output types (for example inquiry outputs, order outputs; an output type is also called a “condition type“).

An application determines one procedure from the current document type, which it transfers to message determination. A procedure is therefore a collection of output types from an application perspective.

Use

In Customizing you allocate output types to the procedure. For more information see define procedure [Page 63]

Integration

The procedure is the highest element in the condition component hierarchy. Whenever a procedure is called, all **output types** (next hierarchy level) are search for.

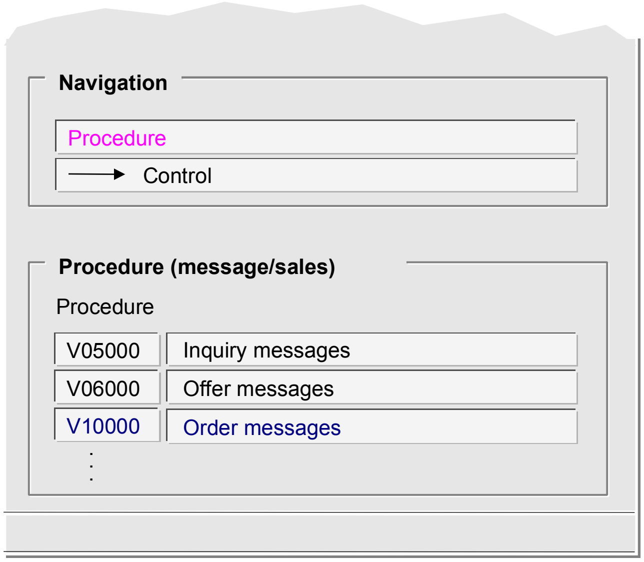
Condition Components: Sample Hierarchy [Page 11]

Defining the Procedure

1. You can select one of the two following alternatives:
   - Select the appropriate activity from Customizing in the relevant application
   - Use transaction NACE in *expert mode*. The following describes this alternative.
2. Select your application and select *Edit* → *Procedures*.
3. In change mode select *Edit* → *New entries.* In *Control* assign your procedure output types.
4. You can use an ABAP condition as an exit criterion for an output type. You define new conditions with transaction V/27.

   If the condition is met, then the system does **not** search for messages of this output type. You can also define conditions in define access sequences [Page 58].
5. Save your entries.

The following illustration shows that procedure V10000 is declared for order messages. The application's Customizing defines when this procedure should be applied.

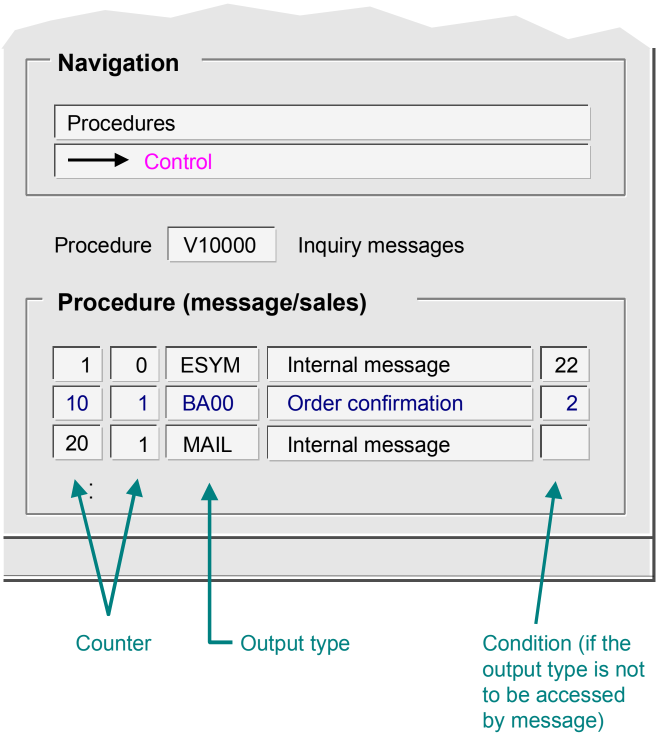


Procedure (message/sales)

Procedure

V05000	Inquiry messages
V06000	Offer messages
V10000	Order messages
:	

The following illustration shows the output types that are declared in procedure V10000. Subsequent explanations will concern output type BA00 (order confirmation). The counters defined in the procedure do not affect message determination because the procedure is always processed as a whole.



Navigation

Procedures

→ Control

Procedure V10000 Inquiry messages

Procedure (message/sales)

1	0	ESYM	Internal message	22
10	1	BA00	Order confirmation	2
20	1	MAIL	Internal message	

Counter
Output type
Condition (if the output type is not to be accessed by message)

In the example above, condition 2 means that the sales document must be checked for completeness. A message is only generated if the document is complete.

The code is:

```
form_kobed_002.
* Falls Verkaufsbeleg vollständig ist, soll Nachricht erzeugt werden
  sy-subrc = 0.
  if komkbv1-uvall ne 'C'.
    sy-subrc = 4.
    exit.
  endif.
  if komkbv1-costa ne space and
     komkbv1-costa ne 'C'.
     sy-subrc = 4.
     exit.
  endif.
  if not komkbv1-lifsk is initial.
     if komkbv1-lifsk ne tvls-lifsp.
       select single * from tvls where lifsp = komkbv1-lifsk.
   endif.
     if sy-subrc = 0 and
        tvls-spedr ne space.
       sy-subrc = 4.
       exit.
     endif.
 * Kein Druck, wenn Kreditsperre sitzt
   if   komkbv1-cmgst ca 'BC'.
     sy-subrc = 4.
     exit.
   endif.
 endform.
```

Output Type

Definition

An output type (or also a “condition type“) summarizes messages of the same meaning. It contains parameters that are valid for all its assigned messages, for example appropriate partner functions.

BA00 (order confirmation), MAIL (internal message), KRML (credit limit exceeded)

Use

For each transmission medium, one form routine in a program is assigned to the output type. This form routine is called for processing the message (message status record).

For transmission medium 7: SAPoffice mail [Page 86] you can assign the output type title and text of mails which are used as default values in the condition records, that is to say, in the messages.

For more information, see

Defining output types [Page 61]

Integration

Output types come after procedures in the condition component hierarchy. One **access sequence** (next hierarchy level) is assigned to the output type. In addition, one or more **condition tables** (last hierarchy level) can be assigned to an access sequence.

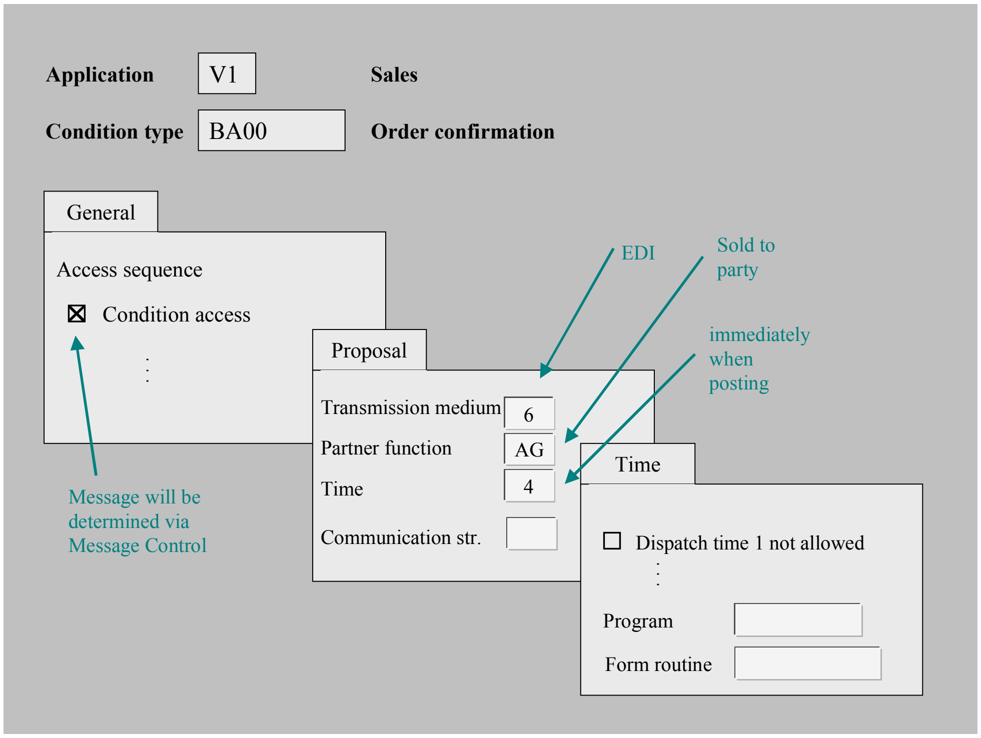
Condition Components: Sample Hierarchy [Page 11]

Defining Output Types

1. You can select one of the two following alternatives:
   - Select the appropriate activity from Customizing in the relevant application
   - Use transaction NACE in *expert mode*. The following describes this alternative.
2. Select your application and select *Edit* → *Output types*.
3. In change mode select *Edit* → *New entries.* You can also define default values for the condition records in your new definition.
4. Save your entries.

The following diagram shows that for order messages the output type BA00 (order confirmation) was defined with condition access (that is to say the message determination is called). Here you can also determine whether specific processing times [Page 49] should be excluded. In addition, control parameters for message determination and exits (that can occur during message determination) are declared for the transmission media "Print" and "Mail" (SAP*office)*.

Default values are declared for the condition record for output type BA00, for example transmission medium [Page 83] 6 (EDI). It makes no sense to create a communication strategy [Page 85] for this transmission medium. It makes sense for transmission medium 5: When creating a condition record for order acknowledgment, the communication strategy stored here would then be followed.



Application V1 Sales

Condition type BA00 Order confirmation

General

Access sequence

☒ Condition access

Proposal

Transmission medium 6

Partner function AG

Time 4

Communication str.

Time

☐ Dispatch time 1 not allowed

Program

Form routine

Defining Processing Program and Form

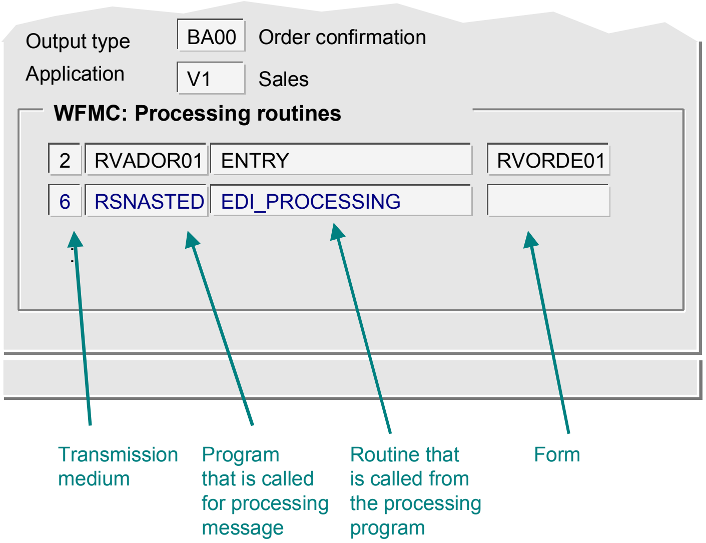
Prerequisites

You are in the maintenance dialog for output types [Page 61].

Procedure

1. Choose your output type (“condition type”) and choose *processing routine.*
2. Enter in change mode for the requested transmission medium a processing program, a form routine, and if applicable an SAPscript form. For a transmission medium that has not yet been entered, choose *Edit* → *New entries.*
3. Save your entries.

The following diagram shows a processing program in sales.



Output type BA00 Order confirmation

Application V1 Sales

WFMC: Processing routines

2	RVADOR01	ENTRY	RVORDE01
6	RSNASTED	EDI_PROCESSING	

Transmission medium

Program that is called for processing message

Routine that is called from the processing program

Form

Access Sequence

Definition

An access sequence is a **message determination** strategy used by the system to determine valid values for the **message default**.

Use

An access sequence

- Determines the order in which the condition tables are searched for messages. You can define that if the search is successful, no further access sequence condition tables will be processed (**Exclusive** indicator).
- Assigns fields in the communication structure to keys in condition tables (= **table access**). The contents of the communication structures are used to read the condition table and find a record (therefore one message). If the table is accessed successfully, the record may be edited in the condition table and becomes a **message status record** in the NAST table.

Defining Access Sequences with Accesses and Fields in Communication Structures [Page 58]

Integration

Condition Components: Sample Hierarchy [Page 11]

Defining Access Sequences

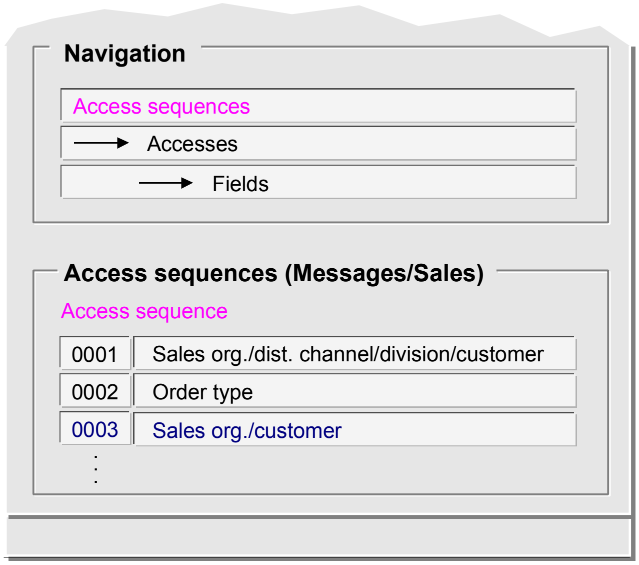
1. To define an access sequence, you can choose one of the two following alternatives:
   - Select the appropriate activity from Customizing in the relevant application
   - Use transaction NACE in *expert mode*. The following describes this alternative.
2. Select your application and select *Edit* → *Access sequences*.
3. In change mode you define new access sequences or change existing ones. This also involves the allocation of application fields to the key fields of the relevant condition table.

   The key is defined in message determination from the current values of the assigned application fields. The condition table is then read with this key. This is table access [Page 33].
4. You can use an ABAP condition for access for each condition table. You define new conditions with transaction V/27.

   If the condition is met, then the system does **not** access the condition table. You can also define conditions in Procedures [Page 63].
5. You can set the "exclusive" indicator for each condition table.

   If access to a table indicated as *exclusive* was successful, then the system does not search any further for messages in the access sequence. Instead it continues with the next output type defined in the procedure.
6. Save your entries.

The following illustration shows access sequences for sales documents. Subsequent explanations will concern access sequence 0003 of this list.



Navigation

Access sequences

→ Accesses

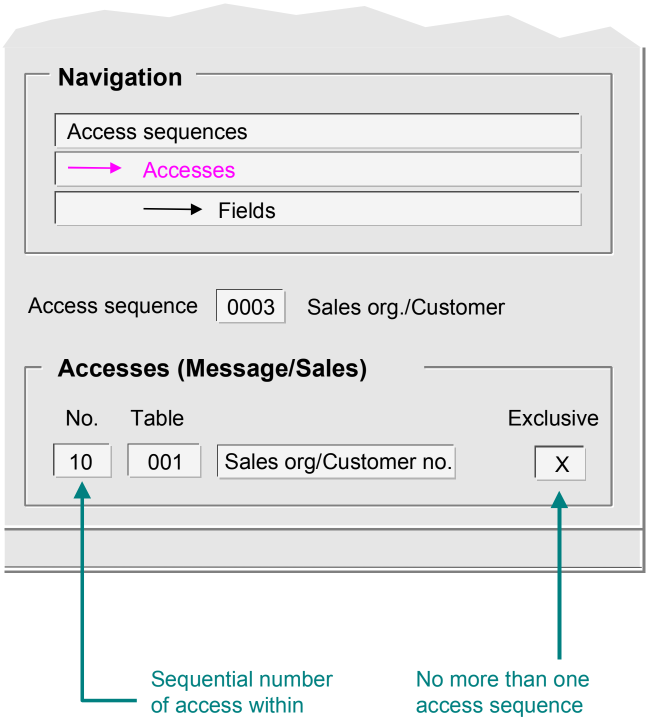
→ Fields

Access sequences (Messages/Sales)

Access sequence

0001	Sales org./dist. channel/division/customer
0002	Order type
0003	Sales org./customer
⋮	

The following illustration shows access sequence 0003 being used to access condition table 001. Here you define the accesses and their order in the access sequence.



Access sequence 0003 Sales org./Customer

Accesses (Message/Sales)

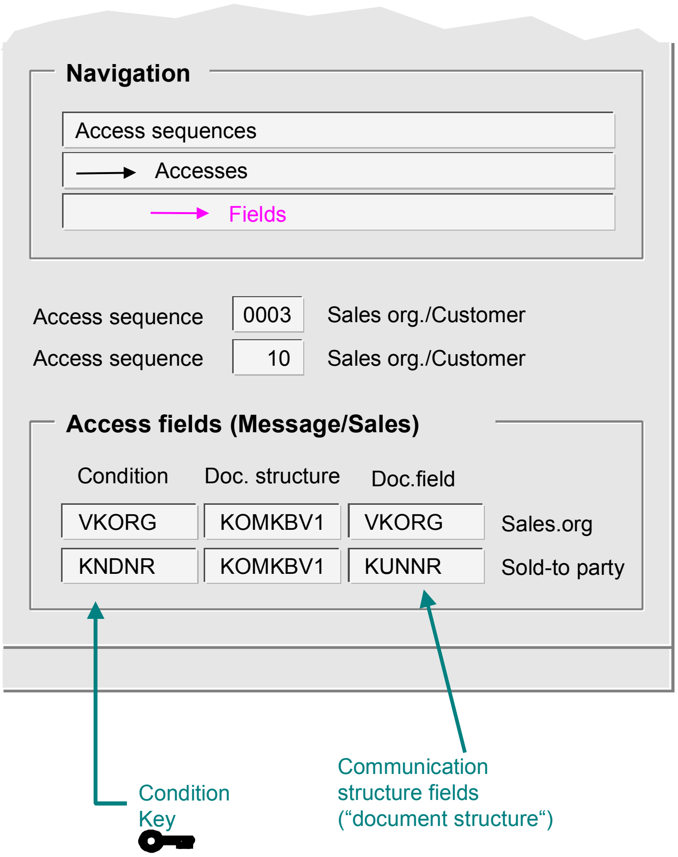
No.	Table		Exclusive
10	001	Sales org/Customer no.	X

Sequential number of access within access sequence

No more than one access sequence if access is successful:

The following illustration shows which fields are used to access the condition table. The sold-to party is transferred as the customer number (in contrast for example to the ship-to party). This **partner function** “sold-to party“ is defined as the standard default value for the condition records of the output type order confirmation (BA00).

Defining Access Sequences



Navigation

Access sequences

→ Accesses

→ Fields

Access sequence 0003 Sales org./Customer

Access sequence 10 Sales org./Customer

Access fields (Message/Sales)

Condition	Doc. structure	Doc.field	
VKORG	KOMKBV1	VKORG	Sales.org
KNDNR	KOMKBV1	KUNNR	Sold-to party

Condition Table

Definition

If the condition table access is successful, a specific action is executed in the condition technique. In the case of Message Control this action is the proposal of a message found in the condition table.

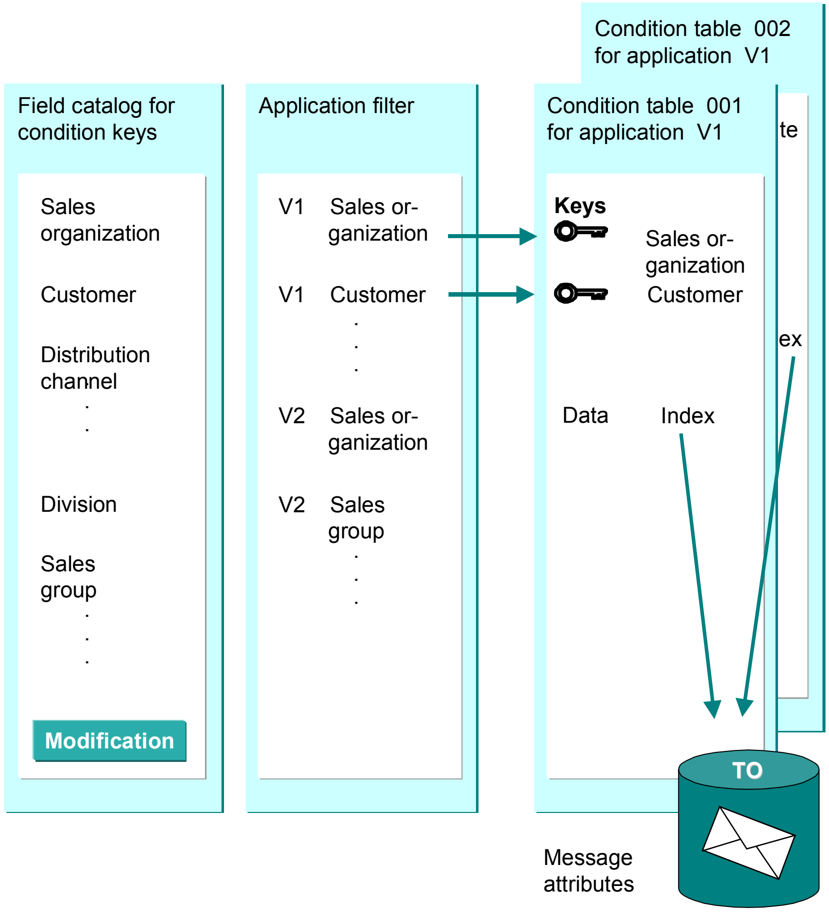
Structure

A condition table consists of one or more condition keys and a data part:

- The condition keys can be selected according to the data combination for which a message is to be found. The key fields come from the **field catalog**.
- The data section is dependent on the use of the condition technique (message/price/account determination) and consists of the fields which describe the message. In the case of **message determination** (that is if the condition table is used by the Message Control) the data part consists of an index for the table **message attributes** (NACH).
- The message attributes contain, for example, the processing time, the transmission medium and the language. When creating a message (that is to say when creating a corresponding condition record), this data is determined in the application.

The condition records of a condition table describe the values or field contents which lead to a message, as well as the attributes of this message. Depending on the application, the message found is proposed in a business process or forwarded for processing directly.

Condition Table



Message Control scenario: Condition keys

- A message is to be generated if the credit limit is exceeded in a sales order. A condition table must therefore be created for which the key is the appropriate lock indicator.
- A message is to be generated if the document status is changed. In this case, a condition table can be used which consists of a key from the status field containing the document information record.

Use

Defining Condition Records [Page 32]

Condition tables that have already been defined are filled in the application with condition records. SAP delivers several condition tables as standard.

Defining Condition Table [Page 56]

New condition tables can be defined in customizing. Their names are restricted to a customer name range.

Integration

Condition Components: Sample Hierarchy [Page 11]

Create Condition Table

1. To define a condition table, you can choose one of the two following alternatives:
   - Select the appropriate activity from Customizing in the relevant application
   - Use transaction NACE in *expert mode*. The following describes this alternative.
2. Select your application and select *Edit* → *Condition tables*.
3. Enter a new condition table.

   You receive the application view of the field catalog [Page 14].
4. Select the key fields of your new condition table by double-clicking on the field catalog.
5. Select *Condition* → *Generate*. Put your newly generated table in a transport request.

The following illustration shows condition table 001 which is supplied as standard and has *Sales organization* and *Customer* as its condition keys.

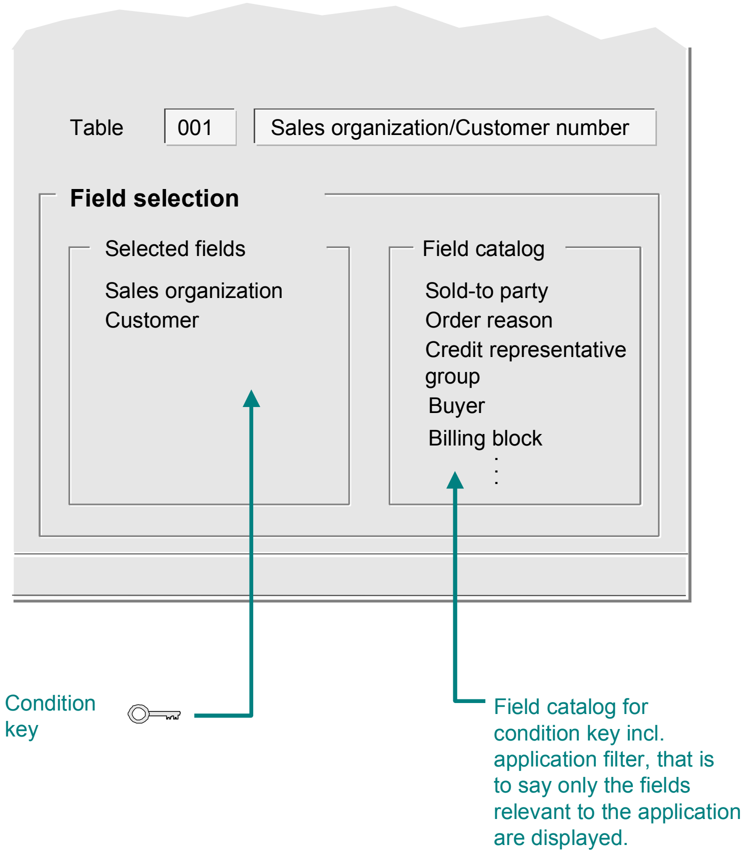


Table 001 Sales organization/Customer number

Field selection

Selected fields

Sales organization
Customer

Field catalog

Sold-to party
Order reason
Credit representative group
Buyer
Billing block
:

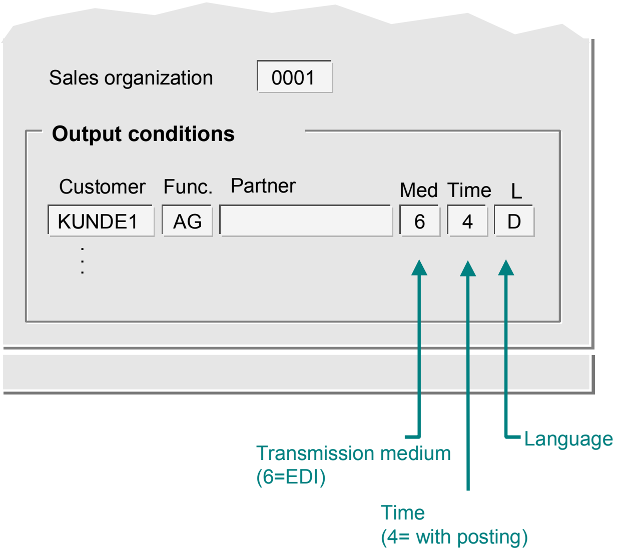
Condition key

Field catalog for condition key incl. application filter, that is to say only the fields relevant to the application are displayed.

Defining Condition Records

1. Declare the condition records in the application (often under *Messages*) or enter the transaction **NACE** (expert mode), select your application and choose *Condition records*.
2. Choose your requested output type.
3. Choose from the condition tables offered the one which has the requested condition key. Select *Execute.*
4. Enter your condition records and then save them.

You maintain condition records via the SD system menu. In SD, condition records used for message determination are declared as message master data. Other applications treat condition records as administration or Customizing data. The following diagram shows the message condition for the output type BA00 in Sales organization 0001. When CUSTOMER1 places an order, they receive an order confirmation via transmission medium EDI (6) immediately when the order is posted (time 4).



Sales organization 0001

Output conditions

Customer	Func.	Partner	Med	Time	L
KUNDE1	AG		6	4	D

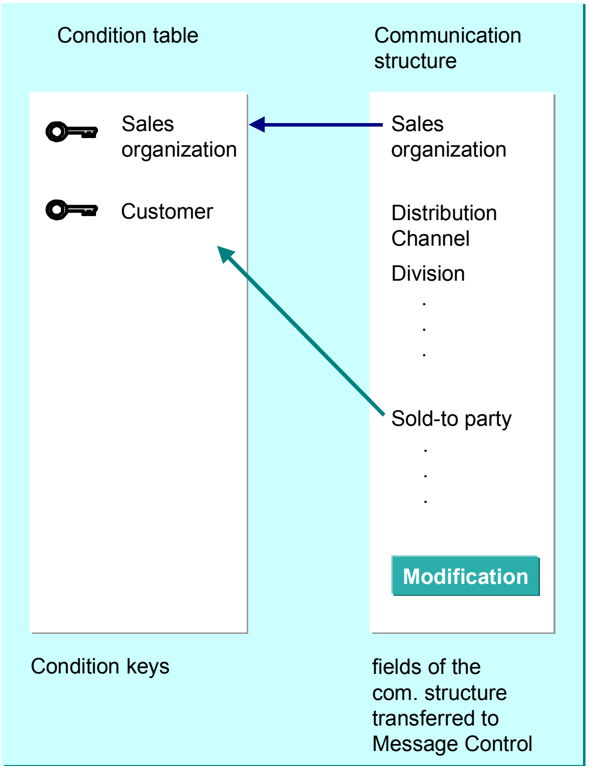
Condition Access

Definition

When a condition table is accessed the key fields are filled with values from the application fields. The condition table is then read with the specified key.

In the **definition** of access in Customizing, assign the condition keys of the condition tables to the access fields of the application (from the communication structure [Page 92]).

The following condition table has the key fields ‘sales organization’ and ‘customer’. These are filled using the contents of the application fields ‘sales organization’ and ‘sold-to party’.



Message Determination

Use

During message determination, the condition tables are searched for condition records according to the rules defined in Customizing (defining condition components) which correspond to the current application data.

The current application data is transferred to Message Control from the application in the communication structure [Page 92] . In addition, the application also transfers the relevant application code [Page 13] and a procedure [Page 15] as the highest condition components in the hierarchy.

When the system finds a message, the following scenario occurs:

- If the application provides functions for manual editing of messages, the message found is proposed to the user. The user can edit the message manually and make any changes required.
- The message (which may have been edited) is saved and is then immediately or later [Page 49] transferred to the processing program.

Prerequisites

The condition components and condition records [Page 10] must be defined.

Features

Message determination: Processing condition component hierarchies [Page 35]

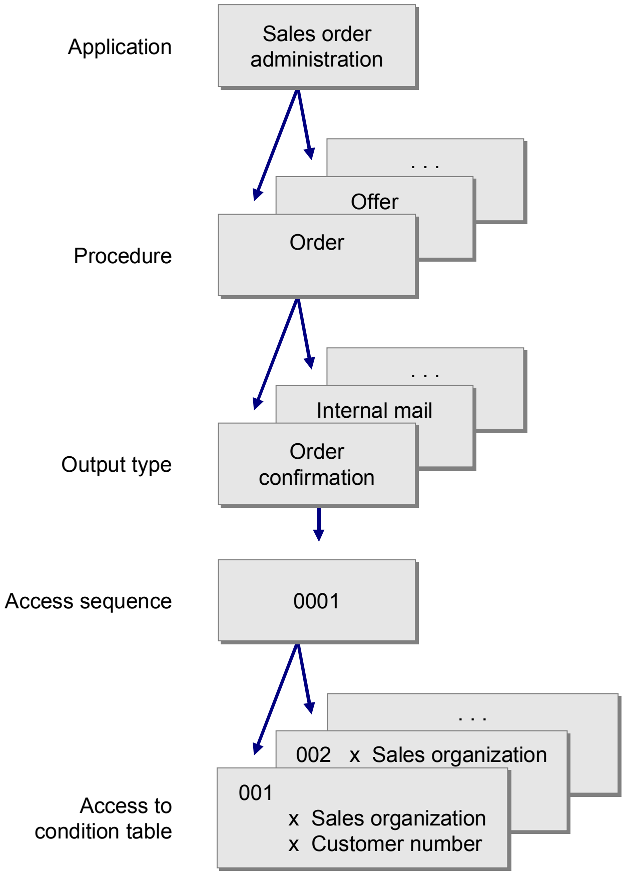
Condition table access [Page 37]

Activities

For each object (document), if you want to...	You should read the section entitled...
send not more than one message to each partner	Strategy:"Exclusive Access Sequence" [Page 38]
send several messages to a partner in various functions or languages	Strategy:"Inclusive Access Sequence" [Page 39]
send several messages to a partner in the same function and language	Strategy “procedure” [Page 41]

Example Message Determination: Processing Condition Component Hierarchies

If a business application transfers a procedure for message determination to Message Control, all output types in the procedure are searched. The following diagram shows the output types *Order confirmation* and *Internal mail* in the procedure *Order*.



During message determination for an output type, the access sequence is processed, i.e. all condition tables in this access sequence are accessed sequentially.

In the above example, there are two condition tables (001 and 002) for the access sequence 0001 and therefore for the message *Order confirmation*.

The order confirmation access sequence refers successively to the tables listed below with the corresponding keys:

Table	Key	Indic. “Exclusive”

Example Message Determination: Processing Condition Component Hierarchies

Condition table 001	Sales organization customer number	According to strategy:set or not set
Condition table 002	Sales organization	No difference in example

Condition table 001 therefore contains more specific conditions (sales organization / customer number) than condition table 002 (only sales organization) for the sending of an order confirmation. In the case of an exclusive exit strategy for table 001, table 002 is no longer accessed when a condition from table 001 has been met. As there is not a third table behind table 002 in the access sequence, it makes no difference in this example if the exit strategy for this table is inclusive or exclusive.

You can assign the following condition components with conditions written in ABAP:

- Output type

  The value of the system return code at the end of the condition determines whether the condition table of this output type is accessed. If the result is negative (Return code <> 0), processing in the procedure continues, that is to say, the next output type in the procedure is processed.

- Condition Table Access

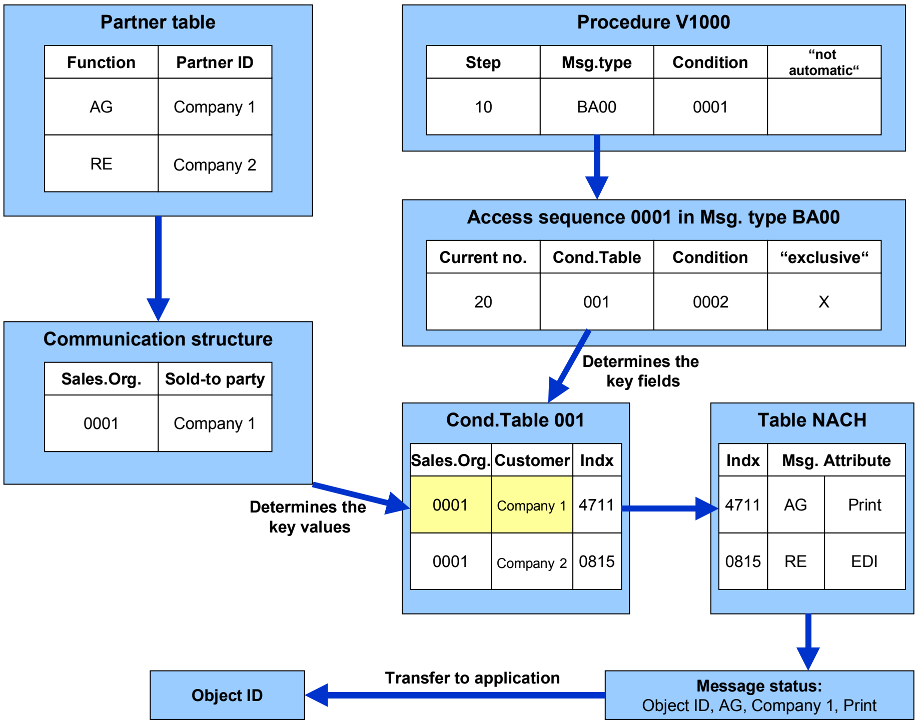
  If the system return code is <> 0 at the end of the condition, processing of the access sequence continues.

Message determination: Condition Table Access

Order confirmations are searched for in the following diagram:

1. Message determination begins in the procedure V1000 with the output type BA00 and the assigned access sequence 0001.
2. In the access sequence the key of the condition table 001 is determined with the sales organization 0001 and the sold-to party “company 1”. The condition table is read with this key.
3. If a condition record is found, the system proposes a message default with the attributes found in table NACH.
4. The message default (which may have been edited manually) is stored after document posting as a record in table NAST. The message is given a unique number for identification purposes.

Condition access example



Partner table

Function	Partner ID
AG	Company 1
RE	Company 2

Procedure V1000

Step	Msg.type	Condition	“not automatic“
10	BA00	0001	

Access sequence 0001 in Msg. type BA00

Current no.	Cond.Table	Condition	“exclusive“
20	001	0002	X

Communication structure

Sales.Org.	Sold-to party
0001	Company 1

Cond.Table 001

Sales.Org.	Customer	Indx
0001	Company 1	4711
0001	Company 2	0815

Table NACH

Indx	Msg. Attribute	
4711	AG	Print
0815	RE	EDI

Determines the key fields
Determines the key values
Object ID
Transfer to application
Message status:
Object ID, AG, Company 1, Print

Strategy "Exclusive Access Sequence"

Consider the following **access sequence**

Condition table	**Key**	**Indic. “Exclusive”**
001	Sales organization customer number	set
002	*Sales organization*	not set

with access to the following two **condition tables**:

Condition table 001

Sales organization	**Customer number**	**Partner function**	**Transmission medium**	**Language**	**Processing time**
0001	Company1	Sold-to party	2 Fax	English	Immediately
0002	Company2	Sold-to party	6 EDI	German	Night

Condition table 002

Sales organization	**Partner function**	**Transmission medium**	**Language**	**Processing time**
0001	Sold-to party	1 Print	German	Immediately
0002	Sold-to party	1 Print	English	Immediately

This definition of the condition components leads to the following scenario at message determination:

In sales organizations 0001 and 0002, the order confirmation is normally printed in English or German immediately at order entry, and is addressed to the sold-to party. If, however, the sold-to party in sales organization 0001 is “Company 1”, an order entry will cause the order confirmation to be faxed in English, and not printed in German. If the sold-to party in sales organization 0002 is “Company 2”, it will cause a German order confirmation to be sent by EDI, but it will not be printed in English.

Condition table 001 is evaluated before condition table 002. A diagram illustrating this example can be found in: Message determination: Processing condition component hierarchies [Page 35]

Setting the exclusive indicator in condition table 002 will not change the process. This is because in example table 002 the last table is the access sequence.

Strategy "Inclusive Access Sequence”

Consider the following **access sequence**

Table	Key	Indic. “Exclusive”
001	Sales organization customer number	not set
002	*Sales organization*	not set

with access to the following two **condition tables**:

Condition table 001

Sales organization	Customer number	Partner function	Transmission medium	Language	Processing time
0001	Company1	Sold-to party	2 Fax	English	Immediately
0002	Company2	Sold-to party	6 EDI	German	Night

Condition table 002

Sales organization	Partner function	Transmission medium	Language	Processing time
0001	Sold-to party	1 Print	German	Immediately
0002	Sold-to party	1 Print	English	Immediately

This definition of the condition components leads to the following scenario at message determination:

In sales organizations 0001 and 0002, the order confirmation is normally printed in German immediately at order entry, and is addressed to the sold-to party. If, however, the sold-to party in sales organization 0001 is “Company 1,” a fax is also sent in English. If the sold-to party in sales organization 0002 is “Company 2,” an order confirmation will also be sent in German via EDI that night.

⚠

Documents of **one** output type which are sent at the same time must differ from each other with regard to at least one of these attributes (key for the table of the Message status [Page 47]):

- Partner function
- Partner
- Language

If the attributes are not different and multiple processing is still required, several output types must be defined in the procedure.

The “inclusive access sequence” strategy has the results described above because the message status records found at the same time have different language attributes.

Strategy "Inclusive Access Sequence”

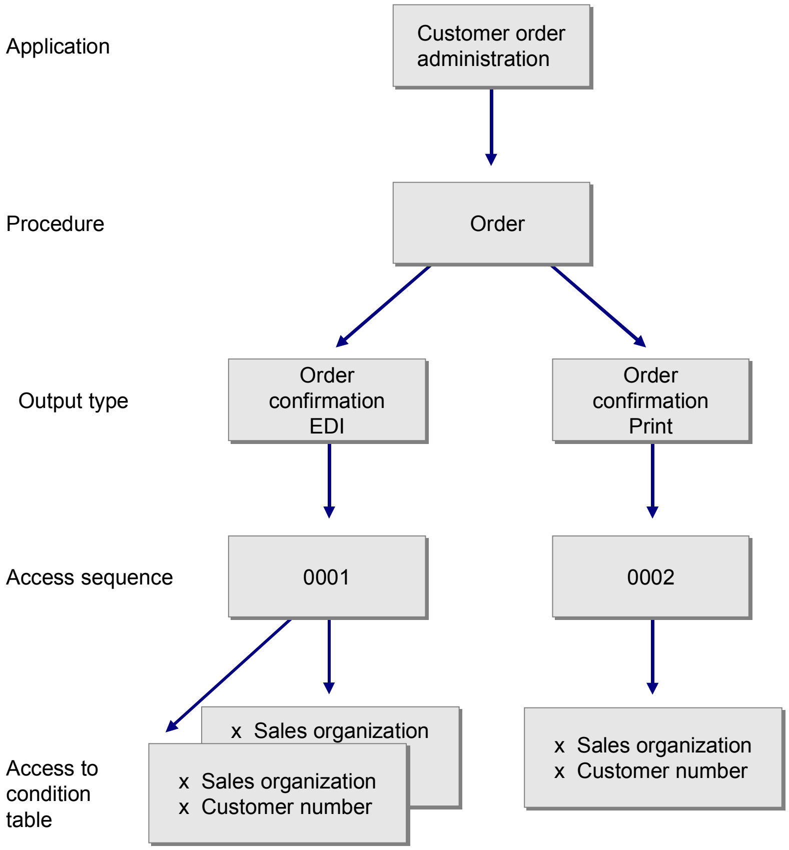
Condition table 001 is evaluated before condition table 002 at message determination. A diagram illustrating this example can be found in: Message determination: Processing condition component hierarchies [Page 35]

Strategy Procedure

If messages with no difference in partner function, partner number, or language are to be sent for an object, different output types must be used in Message Control.

If, for example, a sold-to party is supposed to receive an order confirmation both in printed form and via EDI, this means that a message for an object (customer order) must be sent to a specific partner and in a specific language but using different methods.

The condition technique - and therefore the Message Control - requires different output types to represent this situation. These are linked in the procedure, where you cannot assign an ‘exclusive indicator’. For example, you can define the condition components displayed in the following diagram:



In the example, the sold-to party “Company 3”, for which transfer via EDI has been newly defined, should receive the order confirmation via EDI. In the initial phase, however, the confirmation should also be printed and sent by mail. The following condition record is created for the output type “order confirmation EDI”:

Example for condition record EDI

Sales organization	Customer number	Partner function	Transmission medium	Language

0001	Company 3	Sold-to party	EDI	German

An additional condition record is then created for output type "order confirmation print":

Example for condition record EDI

Sales organization	Customer number	Partner function	Transmission medium	Language
0001	Company 3	Sold-to party	Print	German

When the test phase with sold-to party “Company 3” is complete, the condition record can be deleted. Subsequently, only EDI is used for communication.

Partner Determination

Definition

During partner determination, a **table of partners** relating to an application object, in addition to the communication structures, is transferred to Message Control by an application. This table contains a list of **partner functions** and **partner numbers**. For further information on customer partners (application Sales and Distribution) see Partners in customer master record [Ext.]

Use

In this case, you must only enter the partner function in the **condition record** since the partner number for the corresponding partner function is taken from the table of partners transferred to the message status record during **message determination**. The recipient of the message is therefore known.

However, it is also possible to define a partner number in the condition record. This number overrides the values found in the table of partners.

➡

A special feature is derived from partner function AP (contact partner): You can define one or more of such contact partners in the customer master record, which are given a unique number but not executed as separate master records. If such a contact partner is then transferred in message proposal as a communication partner (field PARNR), then the processing programs are terminated, which assume the existence of a master record for the communication partner: This applies, for example, for transmission medium 6 (EDI, program RSNASTED).

Message Default and Message Processing

Use

A message default is made in the application transaction (for example purchase order entry). You can change the proposed message manually (edit) before the object (for example order document) is posted and the message is processed automatically.

Prerequisites

Your application must call the corresponding functions of the Message Control. There are also applications which transfer the found message directly for processing without offering any manual processing options. The message is therefore also called the „message default“, although you cannot change the message in any way.

Features

The created message default contains the **recipient** of the message, the **transmission medium** and the **processing time** for the subsequent (automatic) processing. You can change these parameters in the transaction application.

You can display a **determination analysis** showing whether messages were found and if so where it was in the procedure. You can display a message in detail which will show which application data determined the condition table key.

After document posting the processed message default is stored as a Message Control record [Page 47] in table NAST.

In SD a sales order is placed in sales organization 0001 with sold-to party CUSTOMER1. The system suggests a purchase order acknowledgment (output type BA00) that should be sent by EDI.

The determination analysis shows that the key of the condition table consisting of the fields “sales organization” and “customer” is determined with the application fields “sales organization” (0001) and “sold-to party” (CUSTOMER1). The condition table is read with this key, a condition record is found and proposed in SD as an EDI purchase order acknowledgment.

Message Processing

Use

For output (for example EDI dispatch) the message found is processed. This involves, for example, **formatting** (for printing). The current status of message processing can be seen in **message status**. **Processing time** can be used to set when message processing should be triggered.

Features

The Message status record [Page 47] (table NAST) contains information about when to start processing programs and which parameters are used. When the processing program has been called, a processing status is written. You can therefore obtain an overview of the processing status of the messages (or conditions) valid for this object on the basis of an application object.

Subsequent processing [Page 49]

Interface to processing programs [Page 50]

Processing log [Page 51]

Formatting SAPscript messages [Page 52]

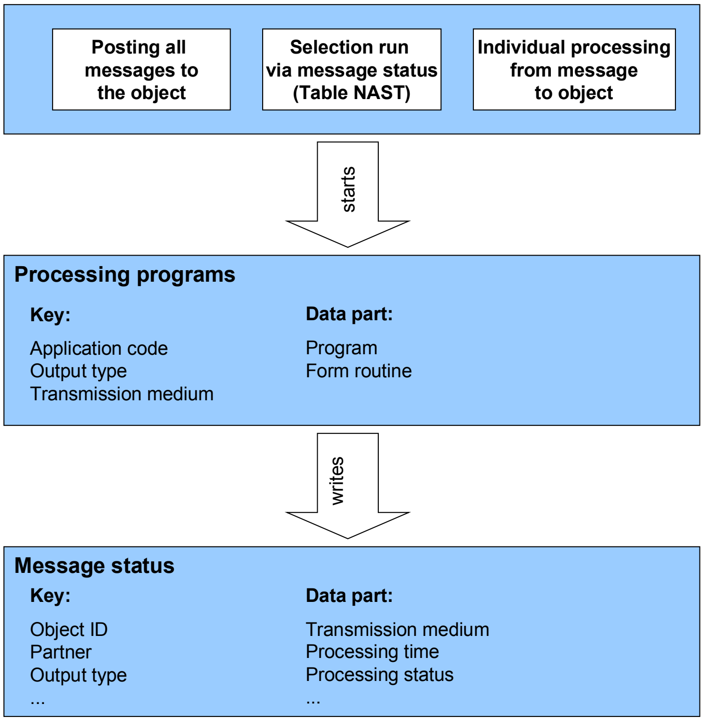
Formatting mails [Page 53]

Message Control scenario: Processing

- A particular customer's credit limit being exceeded should cause the account history to be printed out. In this case, the processing program is the print formatting and output.
- A release message for a document is implemented in a processing program as a call to Business Communication Interface.

The following diagram shows the various processing options:

Message Processing



Posting all messages to the object

Selection run via message status (Table NAST)

Individual processing from message to object

Processing programs

Key:

Application code
Output type
Transmission medium

Data part:

Program
Form routine

Message status

Key:

Object ID
Partner
Output type
...

Data part:

Transmission medium
Processing time
Processing status
...

Message Status Record

Definition

Table record which contains the message settings, in particular their **processing status**.

Structure

The status record logs

- When it was written to the database (key)
- To which object (document) and to which output type the message belongs to (key)
- Which partner the message should go to (key)
- Which language the message is written in (key)
- Via which transmission medium the message was transmitted or should be transmitted with
- Whether the message should be processed immediately or later [Page 49]
- Whether processing was successful: If processing is successful, the processing status is set to 1, if an error occurs it is set to 2. If it is set to 0 the message is not yet processed, therefore, for example, not yet printed or sent electronically.
- Whether the message is new or it concerns a change message

Further settings generally depend on the transmission medium.

The application can "block" the message with an indicator (field SNDDR): It can then only be processed when the field is empty again. This is currently used in purchasing, whereby documents can be subject to a release procedure: They must then first be released by agents responsible, before the corresponding messages can be sent.

The application sets or deletes the indicator, in which during Message Processing [Page 95], for example, it calls the function module RV_MESSAGES_MODIFY before the module RV_MESSAGES_UPDATE from the Message Control Function Group [Page 94].

Use

After document posting, a status record is generated for each (possibly edited) message default. From the application transaction, you can view the process status of your messages, for example by choosing *Header → Messages* from a purchasing document display. If processing was successful, for example, a green traffic light is displayed. By choosing *Additional specifications* you can display additional fields of a status record.

The message status records are found in table NAST. You can display this table with the usual tool. Choose *Tools → ABAP Workbench → Data Browser → Display table contents.*

Subsequent Processing

Use

If a **processing time** other than 4 is used, the message is not processed when the application document is updated but rather later on.

Activities

You determine the processing time in the condition record.

For subsequent processing use the report RSNAST00 or a specific task function to start the processing. The following connections with the processing time exist:

- Time 1: Use the ABAP/4 Development Workbench to start RSNAST00 online.
- Time 2: Start RSNAST00 as a background process at a certain time (*Tools → Administration, Jobs → Job-Definition)*
- Time 3: Call task function (for example a report). If you are using SAP*script* to process the message, you can also preview it here.

RSNAST00 processes all message status records which meet the selection criteria and for which the selected requested time for message processing or interval has been reached or passed. Requested time for message processing or interval are fields in the Message Control record of table NAST and are not entered in the standard system (Caution: The requested time for message processing are other fields than the dispatch time!) To set a dispatch time or interval, a routine in a report can be defined as the **attribute of an output type** so that this routine runs whenever message determination encounters the output type.

Interface to Processing Programs

Use

You can define your own processing programs which communicate with Message Control via a defined interface. This interface consists of

- Table NAST (message status records) as a **global** parameter
- A return code and a screen flag which are transferred to the called form routine in the processing program.

Activities

The return code (type like sy-subrc = INT4) should be changed by the routine and the screen flag (US_SCREEN of type c) should be transferred.

The return code has the value 999 at time of call. If this value does not change, the calling program assumes that either the form routine or the program could not be found in TNAPR. The called form routine should therefore deliver a return code which does not have the value 999, if possible, and has the value 0 in successful cases. The program RSNAST00 writes the message status based on this return code (table NAST). See also section: Message Status Record [Page 47]

If US_SCREEN is set to 'X', the screen is the output device. If the value is set to ' ' (SPACE), the message is printed.

Assign a transmission medium to your processing program and store it with the output type (table TNAPR). For more information, see

Define processing program and form [Page 22]

Special feature in transmission medium 5: Communication Strategy

The processing program must analyze the Communication strategy [Page 85] and determine the correct dispatch medium. The include program RVADOPFO can be included with this.

Processing Logs

Use

If the application supports it, individual processing steps are logged. You can display the processing log from the application.

When creating a standard order a purchase order acknowledgment shall be sent via EDI to partner CUSTOMER1. When the order is created and the message is generated, you can display the processing status via the processing log.

Prerequisites

The processing program of the application must call function module NAST_PROTOCOL_UPDATE. The log is **extended** by each call. For more information, see

Message Control connection to the application [Page 90]

Formatting SAP*script* Messages

Use

The output of messages in printed form is controlled by SAP*script* for the following Transmission media [Page 83]:

- Print
- Fax
- Communication strategy [Page 85] if containing print or fax

For this purpose, they are formatted as SAP*script* forms.

Activities

In addition to processing programs and processing routines, the required SAP*script* form must be defined for your application and output type. You make the entry in the definition of the output types for transmission medium 1. For more information, see

Defining output types [Page 61]

Processing Mails

Use

The output of messages as *mail* (transmission medium 7) is controlled by Business Communication Interface [Ext.] Default texts can be defined in Message Control as mail headers (mail attributes, for example title and name).

Features

You can enter a title and a standard text for the notifications for each output type and language. This standard text can include **fields** as parameters which are replaced with the current data (**field values**) for the application object at runtime (if message determination is run for the object).

The fields can belong to Message Control (incl. Communication Structure [Page 92]) or to the calling application.

In addition to the text, a mail can contain a process (for example a transaction or report) (**executable mail**). This process can be started in connection with the mail. Like the text, you can set parameters in the processing call.

Activities

Executable mail

The processing is defined in the **condition record**. Enter transmission medium 7 and select *Communication.* The corresponding maintenance screen appears. Via *Goto* → *Execute parameter* you can, for example, define executable reports or SET/GET-parameters. To maintain the condition records read section

Message definition in the application [Page 81]

> We recommend using SAP Business Workflow instead of the executable mails (transmission medium 9 or T).

Using fields (parameters)

If the parameter fields belong to the calling application, a routine in a program which is executed during message determination must be entered in Customizing as an **attribute of the output type**. The routine must belong to the same function group or module pool from which the Message Control is called by the application. This routine is executed during message determination and the parameters can be replaced by the current values. For more information, see

Defining output types [Page 61]

Message Control Customizing

Purpose

In Customizing for the relevant application, you define the condition components for message determination. Conversely you define the actual messages in the application transaction as condition records. For more information, see

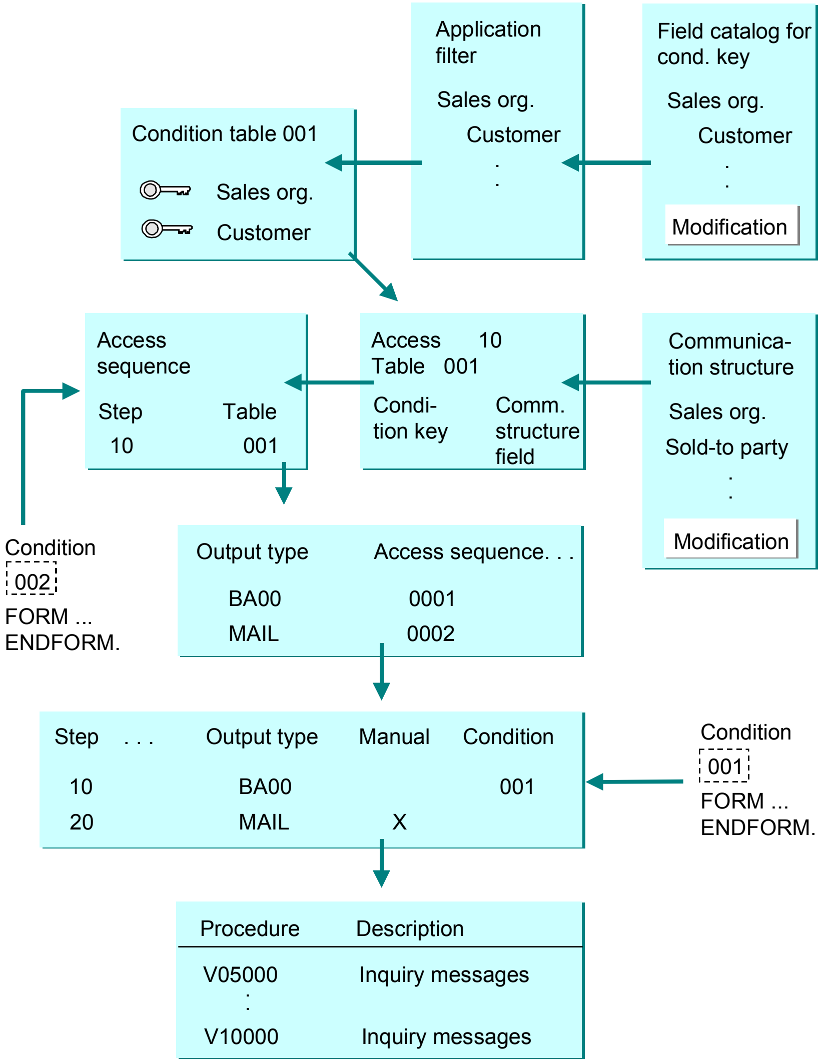
Message definition in the application [Page 81]

Process flow

Keep to the order of the following steps:

1. Defining Condition Table [Page 56]
2. Define the access sequence [Page 58]
3. Define output types [Page 61]
4. Define procedure [Page 63]

The following diagram illustrates this process. Please note that the condition components are defined from the lowest to the highest hierarchy level. Details for a condition element are explained in its Object documentation [Page 10] . Check if SAP standard settings meet your requirements.



Application filter

Sales org.
Customer
.
.

Field catalog for cond. key

Sales org.
Customer
.
.
Modification

Condition table 001

Sales org.
Customer

Access sequence

Step	Table
10	001

Access 10
Table 001

Condition key	Comm. structure field

Communication structure

Sales org.
Sold-to party
.
.
Modification

Condition
002
FORM ...
ENDFORM.

Output type	Access sequence. . .
BA00	0001
MAIL	0002

Step	. . .	Output type	Manual	Condition
10		BA00		001
20		MAIL	X	

Condition
001
FORM ...
ENDFORM.

Procedure	Description
V05000	Inquiry messages
:	
V10000	Inquiry messages

Create Condition Table

6. To define a condition table, you can choose one of the two following alternatives:
   - Select the appropriate activity from Customizing in the relevant application
   - Use transaction NACE in *expert mode*. The following describes this alternative.
7. Select your application and select *Edit* → *Condition tables*.
8. Enter a new condition table.

   You receive the application view of the field catalog [Page 14].
9. Select the key fields of your new condition table by double-clicking on the field catalog.
10. Select *Condition* → *Generate*. Put your newly generated table in a transport request.

The following illustration shows condition table 001 which is supplied as standard and has *Sales organization* and *Customer* as its condition keys.

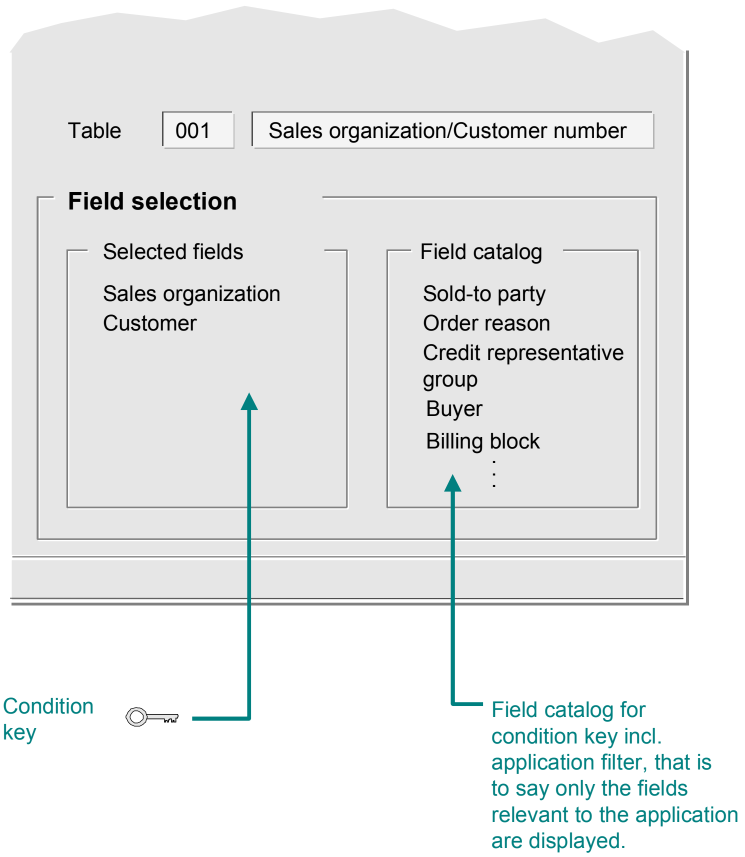


Table 001 Sales organization/Customer number

Field selection

Selected fields

Sales organization
Customer

Field catalog

Sold-to party
Order reason
Credit representative group
Buyer
Billing block
⋮

Condition key

Field catalog for condition key incl. application filter, that is to say only the fields relevant to the application are displayed.

Defining Access Sequences

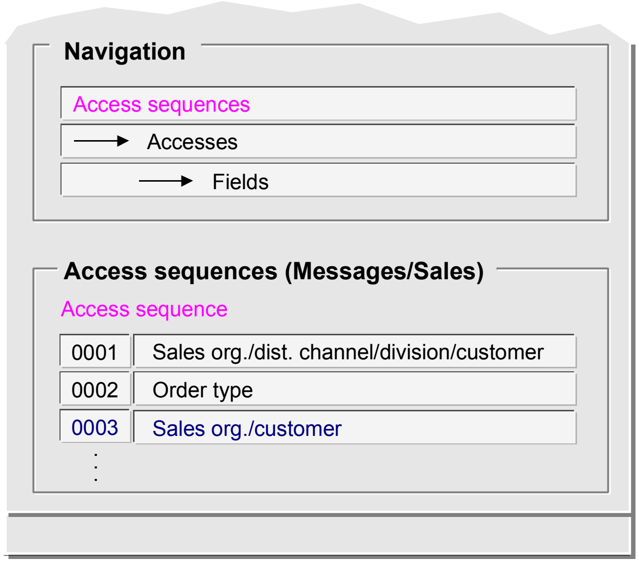
7. To define an access sequence, you can choose one of the two following alternatives:
   - Select the appropriate activity from Customizing in the relevant application
   - Use transaction NACE in *expert mode*. The following describes this alternative.
8. Select your application and select *Edit* → *Access sequences*.
9. In change mode you define new access sequences or change existing ones. This also involves the allocation of application fields to the key fields of the relevant condition table.

   The key is defined in message determination from the current values of the assigned application fields. The condition table is then read with this key. This is table access [Page 33].
10. You can use an ABAP condition for access for each condition table. You define new conditions with transaction V/27.

    If the condition is met, then the system does **not** access the condition table. You can also define conditions in Procedures [Page 63].
11. You can set the “exclusive” indicator for each condition table.

    If access to a table indicated as *exclusive* was successful, then the system does not search any further for messages in the access sequence. Instead it continues with the next output type defined in the procedure.
12. Save your entries.

The following illustration shows access sequences for sales documents. Subsequent explanations will concern access sequence 0003 of this list.



Navigation

Access sequences

→ Accesses

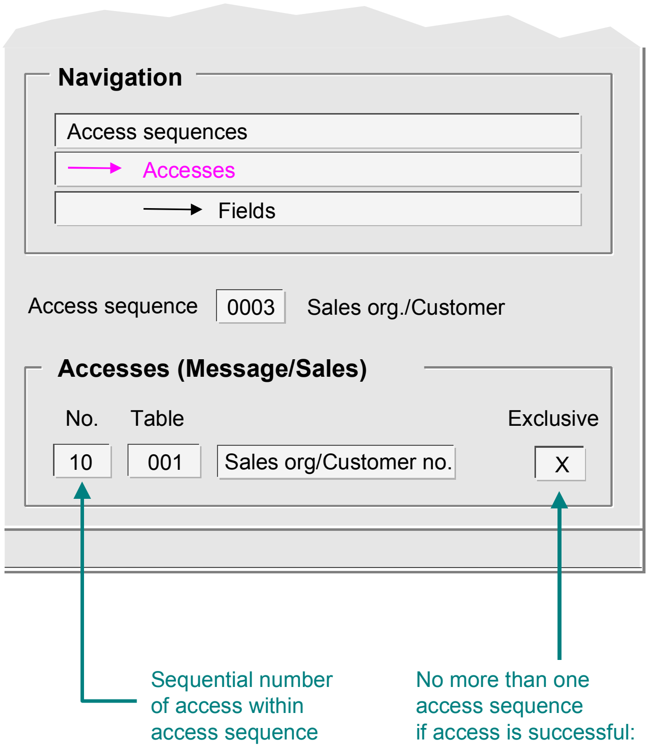
→ Fields

Access sequences (Messages/Sales)

Access sequence

0001	Sales org./dist. channel/division/customer
0002	Order type
0003	Sales org./customer
:	

The following illustration shows access sequence 0003 being used to access condition table 001. Here you define the accesses and their order in the access sequence.



Navigation

Access sequences
→ Accesses
→ Fields

Access sequence 0003 Sales org./Customer

Accesses (Message/Sales)

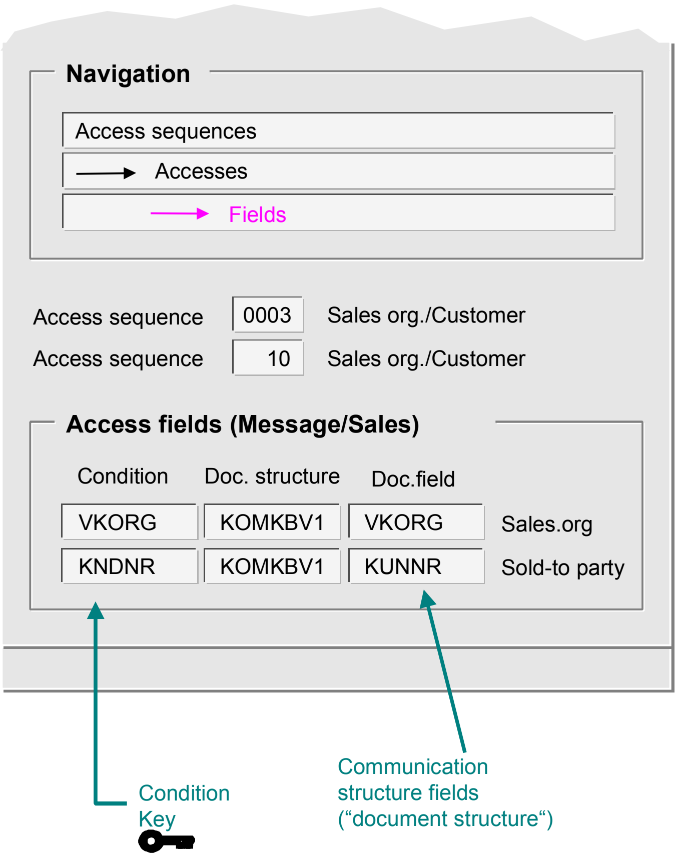
No.	Table		Exclusive
10	001	Sales org/Customer no.	X

Sequential number of access within access sequence

No more than one access sequence if access is successful:

The following illustration shows which fields are used to access the condition table. The sold-to party is transferred as the customer number (in contrast for example to the ship-to party). This **partner function** “sold-to party“ is defined as the standard default value for the condition records of the output type order confirmation (BA00).

Defining Access Sequences



Access sequence 0003 Sales org./Customer

Access sequence 10 Sales org./Customer

Access fields (Message/Sales)

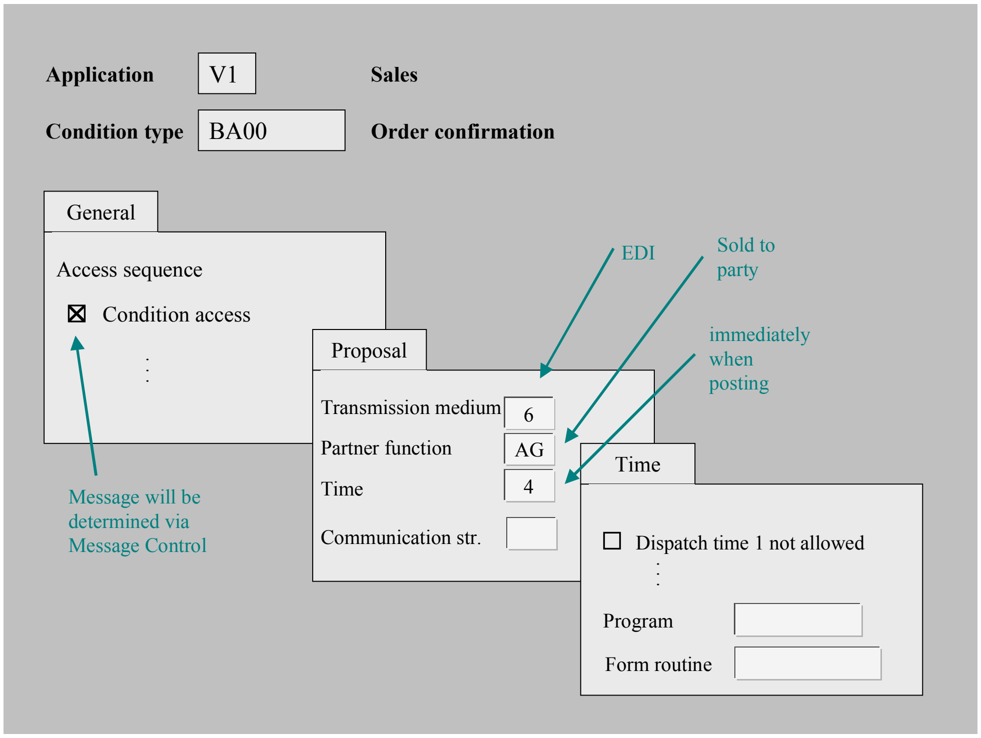
Condition	Doc. structure	Doc.field	
VKORG	KOMKBV1	VKORG	Sales.org
KNDNR	KOMKBV1	KUNNR	Sold-to party

Defining Output Types

5. You can select one of the two following alternatives:
   - Select the appropriate activity from Customizing in the relevant application
   - Use transaction NACE in *expert mode*. The following describes this alternative.
6. Select your application and select *Edit* → *Output types*.
7. In change mode select *Edit* → *New entries.* You can also define default values for the condition records in your new definition.
8. Save your entries.

The following diagram shows that for order messages the output type BA00 (order confirmation) was defined with condition access (that is to say the message determination is called). Here you can also determine whether specific processing times [Page 49] should be excluded. In addition, control parameters for message determination and exits (that can occur during message determination) are declared for the transmission media “Print” and “Mail” (SAP*office)*.

Default values are declared for the condition record for output type BA00, for example transmission medium [Page 83] 6 (EDI). It makes no sense to create a communication strategy [Page 85] for this transmission medium. It makes sense for transmission medium 5: When creating a condition record for order acknowledgment, the communication strategy stored here would then be followed.



Application V1 **Sales**

Condition type BA00 **Order confirmation**

General

Access sequence

☒ Condition access

Proposal

Transmission medium	6
Partner function	AG
Time	4
Communication str.	

Time

☐ Dispatch time 1 not allowed

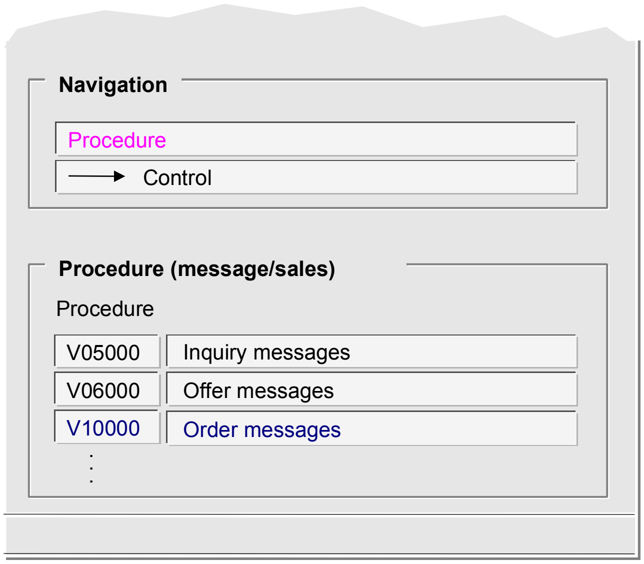
Program	
Form routine	

Defining the Procedure

6. You can select one of the two following alternatives:
   - Select the appropriate activity from Customizing in the relevant application
   - Use transaction NACE in *expert mode*. The following describes this alternative.
7. Select your application and select *Edit* → *Procedures*.
8. In change mode select *Edit* → *New entries.* In *Control* assign your procedure output types.
9. You can use an ABAP condition as an exit criterion for an output type. You define new conditions with transaction V/27.

   If the condition is met, then the system does **not** search for messages of this output type. You can also define conditions in define access sequences [Page 58].
10. Save your entries.

The following illustration shows that procedure V10000 is declared for order messages. The application's Customizing defines when this procedure should be applied.



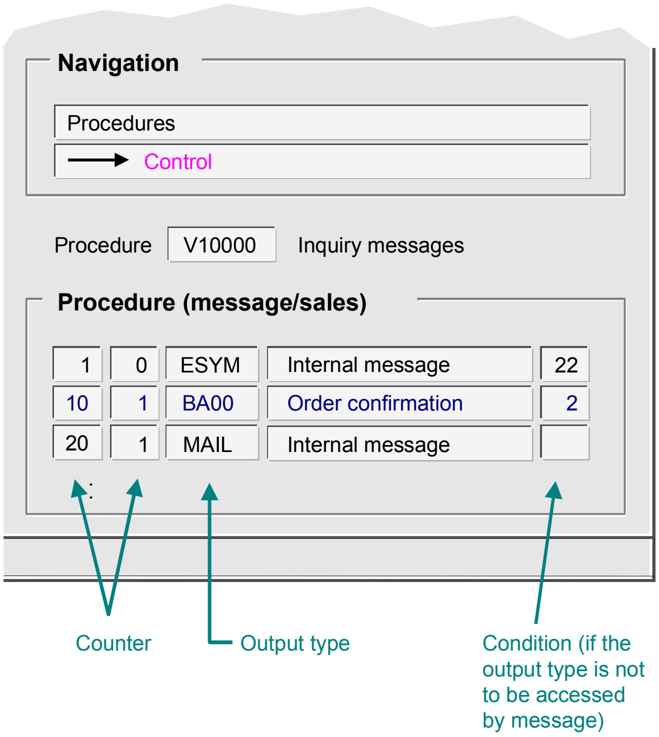
Procedure (message/sales)

Procedure

V05000	Inquiry messages
V06000	Offer messages
V10000	Order messages
:	

The following illustration shows the output types that are declared in procedure V10000. Subsequent explanations will concern output type BA00 (order confirmation). The counters defined in the procedure do not affect message determination because the procedure is always processed as a whole.

Defining the Procedure



Navigation

Procedures

Control

Procedure V10000 Inquiry messages

Procedure (message/sales)

1	0	ESYM	Internal message	22
10	1	BA00	Order confirmation	2
20	1	MAIL	Internal message	

Counter
Output type
Condition (if the output type is not to be accessed by message)

In the example above, condition 2 means that the sales document must be checked for completeness. A message is only generated if the document is complete.

The code is:

```
form_kobed_002.
* Falls Verkaufsbeleg vollständig ist, soll Nachricht erzeugt werden
  sy-subrc = 0.
  if komkbv1-uvall ne 'C'.
    sy-subrc = 4.
    exit.
  endif.
  if komkbv1-costa ne space and
     komkbv1-costa ne 'C'.
     sy-subrc = 4.
     exit.
  endif.
  if not komkbv1-lifsk is initial.
     if komkbv1-lifsk ne tvls-lifsp.
       select single * from tvls where lifsp = komkbv1-lifsk.
   endif.
     if sy-subrc = 0 and
        tvls-spedr ne space.
       sy-subrc = 4.
       exit.
     endif.
 * Kein Druck, wenn Kreditsperre sitzt
   if   komkbv1-cmgst ca 'BC'.
     sy-subrc = 4.
     exit.
   endif.
 endform.
```

Checking Message Control Customizing

The examples in these guidelines use parameters which are delivered as standard by SAP. They are intended to help users understand the condition component hierarchy in Customizing. The examples relate to EDI applications using the IDoc interface.

Message Control Customizing in Purchasing (MM): Example - Purchase order [Page 67]

Message Control Customizing in Sales (SD): Example - Order confirmation [Page 72]

Summary of Message Control Customizing [Page 79]

This section summarizes the parameters for MM examples.

Message Control Customizing in Purchasing (MM): Example - Purchase Order

Message Control Customizing can be reached via the Reference IMG: from the R/3 initial screen, select *Tools* → *Business Engineer* → *Customizing, Implement. projects* → *Display SAP Ref. IMG*

From the Reference IMG, select *Materials Management* → *Purchasing* → *Messages* → *Output Control*

Condition tables and access sequences (MM) [Page 68]

Checking message types (MM) [Page 69]

Checking procedures (MM) [Page 70]

Checking assignment of partner functions to message types (MM) [Page 71]

Condition Tables and Access Sequences (MM)

1. Open the node *Condition table* and execute the function *Define for purchase order.* Select *Messages: Display Condition Table purchase order.*
2. Select table “27” and press *Enter*. The following fields should be defined as key fields
   - *Purchase organization* and
   - *Vendor*

   Values are entered in these key fields at message determination. Go back to the Reference IMG display.
3. Open the node *Access Sequences* and execute the function for purchase orders.
4. Check the following entries in the corresponding columns of table 27:
   - View *Access Sequences*: name "DocType/PurchOrg/Vendor" for access sequence 0001. Select this line in *table control*.
   - View *Accesses*: condition table “27” and name “Purchasing Output Determination: Purchase Organization/Vendor for EDI”.
   - View *Accesses*: no condition, the “Exclusive” field is selected.
   - View *Fields*: conditions “EKORG” and “LIFNR”, column “Direct value source” is left blank.
5. Leave this function. Continue with the section Checking output types (MM) [Page 69]

Checking Output Types (MM)

1. Open the node *Output Types* and execute the function *Define output types for purchase orders*
2. Select the function *Maintain output types: Purchase order*
3. View *Output type*: Check the following entries:
   - Output type “NEW”
   - Description “Purchase order”
4. Leave this action and select the action *Fine-Tuned Control:Purchase Order*. Check the following entries:
   - CTyp (condition type = output type) “NEW”
   - Operation (print operation) “1”
5. Leave this action. Continue with the section Checking procedures (MM) [Page 70]

Checking Procedures (MM)

1. Open the node *Message procedures* and execute the function *Define for purchase order.*
2. Select *Maintain message determination procedure: Purchase order* from the dialog box.
3. View *Procedures*: Check the following entries in the corresponding table columns:
   - Procedure “RMBEF1”
   - Description “Purchase order header”
4. Select this line in *Table control*.
5. View *Control*: Check the following entries in the corresponding table columns:
   - Step “10”
   - Counter “1”
   - CTyp “NEW”
   - Description “Purchase order”
   - Condition “101”
   - The field *Manual only* is not selected.
6. Leave this function and select “Assign Procedure to Purchase Order”
7. Check the following entries in the corresponding table columns:
   - Use “B” (messages)
   - Application “EF” (purchasing: purchase order)
   - Description “Purchase order”
   - Procedure “RMBEF1”
   - The fields “Z” (processing time for change messages) and “P” (new partner function determination) are not selected.
8. Leave this function. Continue with the section Checking assignment of partner functions to output types (MM) [Page 71]

Checking Assignment of Partner Functions to Output Type (MM)

1. Open the node *Partner functions per output type* and execute the function *Define for purchase order.*
2. Check the following entries in the corresponding table columns:
   - Message “NEW”, (transmission) medium 1 (print), function “LF”.
   - Message “NEW”, (transmission) medium 6 (EDI via IDoc), function “LF”.
3. Leave this function. If all condition components are available as described, Message Control is configured for a purchase order from purchasing. If electronic purchase orders are also to be sent via EDI, the program for **message processing** in table TNAPR should be checked. In this case, the routine EDI_PROCESSING in program RSNASTED must be saved for output type NEW (condition key), application EF and transmission medium 6.

Message Control Customizing in Sales (SD): Example - Order Confirmation

From the Reference IMG, select *Sales and Distribution* → *Basic Functions* → *Output Control* → *Output determination* → *Output determination using the condition technique* → *Maintain output determination for sales documents*

Checking condition tables (SD) [Page 73]

Checking output types (SD) [Page 74]

Checking access sequences (SD) [Page 75]

Checking assignment of output to partner functions (SD) [Page 76]

Checking procedures (SD) [Page 77]

Checking procedure assignments (SD) [Page 78]

Checking Condition Tables (SD)

1. Execute the function *Maintain condition tables*. Select *Maintain condition table for sales documents.*
2. Select table “1” . The following fields should be defined as key fields:
   - *Sales organization*
   - *Customer*
3. Values are entered in these key fields at message determination. Go back to the single action display. From the action *Field catalog output sales documents*, you can still display or maintain the **application filter**. Go back to the Reference IMG display.
4. Continue with the section Checking output types (SD) [Page 74]

Checking Output Types (SD)

1. Execute the function *Maintain output types*.
2. View *Output type*: Check the following entries:
   - Output type *BA00*
   - Name “Order Confirmation”
3. Leave this function and go back to the Reference IMG display. Continue with the section Checking access sequences (SD) [Page 75]

Checking Access Sequences (SD)

1. Execute the function *Maintain access sequences*.
2. Check the following entries in the corresponding table columns:
    - View *Access Sequences*: descrip. *SalesOrg/Customer* for access sequence (AcSq) 0003.
    - View *Accesses* for access sequence 0003: condition table “1” and description “Sales org/Customer number”. no condition, *Exclusive* field is selected.
    - View *Fields*: conditions “VKORG” and “KNDNR”.
3. Leave this function and go back to the Reference IMG display. Continue with the section Checking assignment of output to partner functions (SD) [Page 76]

Checking Assignment of Messages to Partner Functions (SD)

1. Execute the function *Assign output types to partner functions*.
2. Check the following entries in the corresponding table columns:
   - Output type “BA00”, (transmission) medium 1 (print), name “Sold-to party”
   - Output type “BA00”, (transmission) medium 6 (EDI), name “Sold-to party”
3. Leave this function and go back to the Reference IMG display. Continue with the section Checking procedures (SD) [Page 77]

Checking Procedures (SD)

1. Execute the function *Maintain output determination procedure*.
2. View *Procedures*: Check the following entries in the corresponding table columns:
    - Procedure “V10000”
    - Description “Order Output”
3. Select this procedure (select line in *Table control*)
4. View *Control*: Check the following entries in the corresponding table columns:
    - Step “10”
    - Counter “1”
    - CTyp “BA00”
    - Name “Order Confirmation”
    - Condition “2 (routine for order confirmation)”
    - The field *Manual only* is not selected.

    ➡

    The source text for condition “2” can be displayed via the action *Define conditions* at the same level as the node *Output determination*.
5. Leave this function and go back to the Reference IMG display. Continue with the section Checking procedure assignments (SD) [Page 78]

Checking Procedure Assignments (SD)

1. Execute the function *Assign output determination procedure*. Select the sub-function *Allocate sales document header*.
2. Check the following entries in the corresponding table columns:
   - Sales type (VArt) “TA” (document type “sales order“)
   - Output procedure (Out.pr.) “V10000”
   - Description “Order Output”
3. Leave this function. If all condition components are available as described, Message Control is configured for an order confirmation as a print form. If electronic purchase orders are also to be sent via EDI, the program for **message processing** in table TNAPR should be checked. In this case, the routine EDI_PROCESSING in program RSNASTED must be saved for output type BA00 (condition key), application V1 and transmission medium 6.

Summary of Message Control Customizing

The following tables describe all the settings required for Message Control in EDI using examples: “Purchase order” (MM), “Request” (MM) and “Delivery schedule” (MM). The parameters for the example “purchase order” (MM) are also described in Message Control Customizing in Purchasing (MM): Example Purchase Order [Page 67] checked.

Parameters for “Purchase order” (MM, Company IMG node *MM → Purchasing → Messages → Output control*)

IMG node	Field	Parameters
Condition table messages	Table	27, table fields: EKORG (purch. org.) and LIFNR (vendor)
Access sequences	Access sequence	0001
	Description	DocType/PurchOrg/Vendor
	Seq. no.	5
	Condition table	27
	Description	Message determination purchasing: Purch. org./vendor for EDI
	Condition	0
	“Exclusive”	select
	Conditions	EKORG and LIFNR (2 entries)
	Direct value source	No entry
Output Types	Otyp (output type)	NEW
	Operation	1
	Ctyp (condition type)	NEW
	Description	new
	Operation	1
Message determination procedures	Procedure	RMBF1
	Description	Purchase order
	Step	10
	Counter	1
	Ctyp (condition type)	NEW
	Description	new
	Condition	101
	“manual only”	empty
	V (use)	B (messages)
	Application	BF
	Description	Purchase order
	Z,P (checkboxes)	empty

Parameters for “RFQ” (MM, Company IMG node *MM → Purchasing → Messages → Output control*): Only differences in relation to the parameters for “Purchase order” are displayed.

IMG node	Field	Parameters
Message determination procedures	Procedure	RMBEA1
	Description	Purchasing RFQ
	Application	EA
	Description	Purchasing RFQ

Parameters for “Delivery Schedule” (MM, Company IMG node *MM → Purchasing → Messages → Output control*): Only differences in relation to the parameters for “Purchase order” are displayed.

IMG node	Field	Parameters
Output Types	Otyp (output type)	LPET
	Ctyp (condition type)	LPET
	Description	Delivery schedule
	Operation	5
Message determination procedures	Procedure	RMBEL1
	Description	Purchasing delivery schedule
	Step	10
	Ctyp (condition type)	LPET
	Description	Delivery schedule
	Condition	105
	Application	EL
	Description	Purchasing delivery schedule

Message Definition in the Application

Purpose

In the application you determine the values for which a message should be found. You define the message attributes, for example the processing time or the language.

From a technical point of view, you thus create condition records.

When an application calls message determination, condition records are searched for which correspond to the current application data. The search proceeds according to the hierarchy of the defined condition components. When a search is successful, a message with attributes from the corresponding record from table NACH is proposed.

Process flow

If you create a condition record from the application, the system runs through the condition components as with message determination (see diagram below). The application transfers the relevant application code (for example V1 for “Sales“) and the selected output type. Via a screen, you can select your required condition table according to the condition key.

Some transaction screens are dependent on the condition key and provide access to all the Dictionary data which the system associates with the condition key, e.g. documentation or entry help.

After selecting the condition table you create the actual condition record with the message attributes of table NACH. Apart from the **language,** please note the following important condition record fields:

Transmission media [Page 83]

Processing time [Page 49]

Partner determination [Page 89]

Message Control scenario: Condition record

- A message proposal is generated if a block indicator is assigned in a sales order. When creating the condition records required, a screen is displayed automatically which contains the field *Block indicator*.
- A message default is generated when there is a specific document status. When creating the condition records required, a screen is displayed automatically which contains the field *Document status* and contains its documentation.

You will also find an example from the SD in the following section:

Defining Condition Records [Page 32]

Message Definition in the Application

Diagram: System activities when creating a condition record in the application

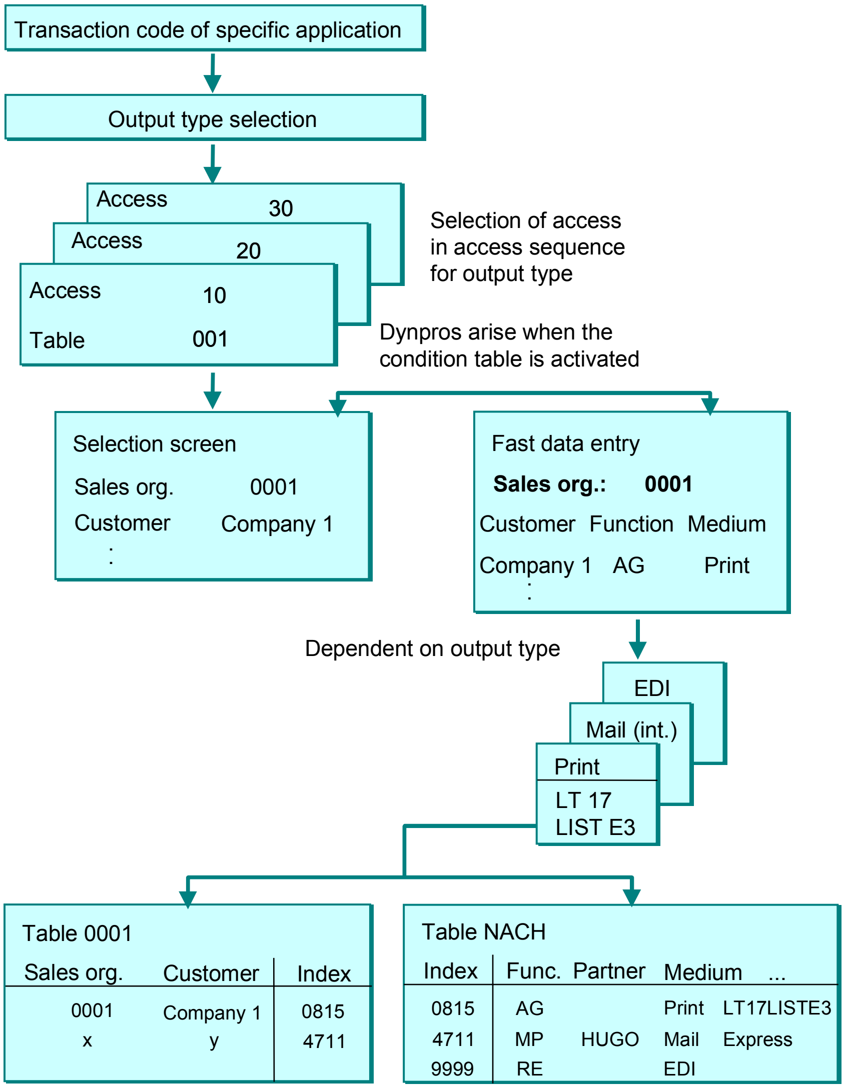


Table 0001

Sales org.	Customer	Index
0001	Company 1	0815
x	y	4711

Table NACH

Index	Func.	Partner	Medium	...
0815	AG		Print	LT17LISTE3
4711	MP	HUGO	Mail	Express
9999	RE		EDI	

Transmission Media

Definition

The transmission media determine the type of processing. The following transmission media are available:

1 Print output [Page 84]

2 Fax

4 Telex

5 Sending externally with a communication strategy [Page 85]

6 EDI: The partner must be defined here as an EDI partner (see Partner profiles [Ext.])

7 Sending a mail [Page 86]

8 Special function

9 Workflow event [Page 87]

A Distribution (ALE) The partner is derived from the distribution model

T Workflow task [Page 88]

You can use the special functions for your own programs. For example, you could call a warehouse management system via CPI-C (Common Programming Interface-Communication).

Integration

Table TNAPR contains information about which **processing programs** should be transferred for the relevant transmission medium for the dispatch of the message. New processing also requires corresponding entries.

- Messages which are to be sent as mails are linked with a program which calls the Business Communication Interface.
- Messages which are to be printed or faxed are linked with a program which contains both a call to the program which determines the print data and the name of the SAP*script* form to be used.
- Messages which are to be processed via the communication strategy are linked to a program which contains the Include RVADOPFO for evaluation of this strategy.

Transmission Medium 1: Print

Definition

Print output includes

- formatting the message after a SAP*script* form or a function module
- output to the spool, from where the data goes either immediately or later to the selected printer

Use

You can set the printer in the condition record, so that the message can be printed without dialog (this is particularly important in immediate processing [Page 49]). Choose *Communication* in the condition record and enter the printer as the *output device* in the subsequent screen. Here, you also have the possibility to enter a function module (*Layout module*) in place of a SAPscript form [Page 52] for print formatting.

Transmission Medium 5: Communication Strategy

Definition

The communication strategy is used to define the sequence in which the system searches for communication methods (Internet, fax, X.400) for a partner. The communication methods must be supported by SAP*connect*. The first communication method for which an address exists in central address management is used.

The communication methods supported by SAP*connect* are not the same as the transmission media supported by Message Control.

Use

Communication strategies are used to automatically select the preferred communication method for contacting one particular partner.
You can create a communication strategy in Message Control Customizing by selecting *Cross-application Components* → *General Application Functions* → *Message Control* The sequence is defined by positions.
In the condition record for your transmission medium 5, select *Communication* and save your communication strategy from the next screen.

You want to send advertising material to all customers, by e-mail in each case if possible. Those customers who do not have an e-mail connection should receive the material via fax. Those customers who have neither an e-mail connection nor a fax connection should receive printed material by mail.

Integration

Save the addresses and communication methods in central address management. Select *Tools* → *Business Communication, Office* → *Addresses* → *External Companies/Contact Person.* New communication methods can be created by selecting *Additional Communication.*

You can define a communication strategy per output type in Customizing, which the system should propose when creating a condition record. For more information, see

Defining output types [Page 61]

Transmission Medium 7: Mail

Definition

You can create a BOR object of type MESSAGE [Ext.] with a condition record with transmission medium 7. You can store a standard text in the message object. This text can be changed in message editing and is sent as a mail in message processing.

Use

The object type MESSAGE offers all the send options available. For example, you can send your messages to an internal distribution list or to external Internet addresses. You can append notes or attachments to your document or assign the status *Reply required* to the document.

In the condition record for your transmission medium 7, select *Communication* and define your object in the usual way from the following screens. If a standard text already exists in the object (for example from an earlier release), you jump directly to the notes maintenance function. The text maintenance or text display functions can be called by selecting *Document* → *Change* or *Document* → *Display*

For more information (executable mail, parameters) see

Processing mails [Page 53]

Integration

The message object is created with the method (‘interface’) IFCREATE, which is appended to the object type MESSAGE. The Business Workplace maintenance screens are then called and the condition record display disappears. For more information about creating a Business Workplace document, read the following documentation:

Creating a document [Ext.]

Transmission Medium 9: Workflow Event

Definition

In message processing an event is stopped, which starts a workflow as an event receiver. For more information, see

Event [Ext.]

Use

An event can be used by various defined workflows. The workflow management is also available when the event is used. They are therefore more flexible than in transmission medium T.

Choose *Communication* in the condition record and enter the BOR Object type [Ext.], for which the event should be triggered, as well as the event itself. The object type must be assigned your application, the event on the other hand the object type. The “correct“ workflows are specified and started from the object type and event as event receivers.

Integration

As with all transmission media you need a processing program [Page 50]. The program must generate the event. At the same time, it can call the function module SAP_WAPI_CREATE_EVENT.

Transmission Medium T: Workflow Task

Definition

A workitem is generated here in message processing, that is defined by a single step task (for example the standard task TS20000019): The task is “instantiated”. For more information, see

Tasks and Task Groups [Ext.]

Use

You can use standard tasks for example, if several transactional users should be informed, of which one should then execute the activity (therefore the work item, for example creating a follow-up document). A more flexible method of workflow processing is available, in which you only send an event as a message (Transmission medium 9).

Choose *Communication* in the condition record and enter the tasks which should be instantiated. The BOR Object type [Ext.], which is defined in this task, must be assigned to your application.

Integration

As is the case for every transmission medium, you also need a Processing program [Page 50] for workflows. This program must generate the work item. At the same time, it can call the function module SAP_WAPI_START_WORKFLOW.

Partner Determination

Definition

During partner determination, a **table of partners** relating to an application object, in addition to the communication structures, is transferred to Message Control by an application. This table contains a list of **partner functions** and **partner numbers**. For further information on customer partners (application Sales and Distribution) see Partners in customer master record [Ext.]

Use

In this case, you must only enter the partner function in the **condition record** since the partner number for the corresponding partner function is taken from the table of partners transferred to the message status record during **message determination**. The recipient of the message is therefore known.

However, it is also possible to define a partner number in the condition record. This number overrides the values found in the table of partners.

➡

A special feature is derived from partner function AP (contact partner): You can define one or more of such contact partners in the customer master record, which are given a unique number but not executed as separate master records. If such a contact partner is then transferred in message proposal as a communication partner (field PARNR), then the processing programs are terminated, which assume the existence of a master record for the communication partner: This applies, for example, for transmission medium 6 (EDI, program RSNASTED).

Message Control Connection to the Application

Purpose

In the course of a new development you can integrate Message Control in an application.

⚠

New developments in customer systems display a modification which is not protected when there is a release upgrade! There are however applications which already use Message Control and allow upwards compatible customer enhancements. Also see section “The communication structure [Page 92] “.

Prerequisites

You have R/3 developer authorization.

Process flow

1. Clarify the requirements for applications [Page 91].
2. Consider if you want to extend field catalog [Page 14]. You should only really do this if the fields already available for the condition access do not suffice.
3. Define your communication structure [Page 92].
4. Program the function calls [Page 95] at the appropriate point in the application program. Also note the section function group of the Message Control [Page 94].
5. Insert functions for Customizing the Message Control [Page 54] in the R/3 IMG structure for your application.

Requirements for Applications

Features

SAP applications which are connected to Message Control must provide the following functions:

- Formulate condition components for the **message determination** in Customizing.
- Call the message determination.
- Optional: generate a **message default** from the message found for the user, who can **edit** this message proposal manually.
- **Process** a message at a certain time.

For more information see the following sections:

Message determination [Page 34]

Function calls [Page 95]

Interface to processing programs [Page 50]

Communication Structure

Definition

When the Message Control is called, application data is placed in the communication structures. This application data is required for message determination.

Use

The structures are required because the condition technique must recognize the dictionary attributes of the fields determining the proposal. Only then can the Message Control generate the programs required for the access.

Every application has one or two communication structures and function modules which fill these structures. In order to support customer enhancements, the fill modules should contain *user-exits* Communication structures and the fill modules are subject to a naming convention (see below). As well as the communication structures, the function modules must also fill the partner tables (see the function calls [Page 95]).

Connectivity with the application filter

As all condition keys are fields of the application filter [Page 14] in the given application, these should be compiled from fields from the corresponding communication structures. Therefore you avoid type conflicts in condition access [Page 37].

Structure

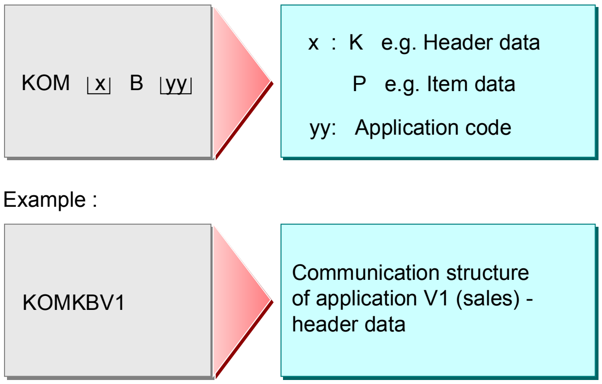
In the case of application objects which are subdivided into several hierarchy levels (header, item, sub-item, and so on) and for which the messages also refer to the lower hierarchy levels, you can create two separate communication structures.

The two communication structures have the following contents:

- Communication structure 1
  contains general fields
- Communication structure 2
  Consists of fields of the lowest hierarchy level to which messages are assigned

Naming convention

The conventions for naming communication structures are as follows:



KOM x B yy

x : K e.g. Header data

P e.g. Item data

yy: Application code

Example :

KOMKBV1

Communication structure of application V1 (sales) - header data

Similarly, the naming conventions for the function modules are KOMxByy_FILL and COMMUNICATION_AREA_KOMxByy. For more information, see

Function calls [Page 95]

Message Control Function Group

Definition

Message Control can only be called via the Application Programming Interface (API), function group V61B.

Use

The API eliminates the need to enter table definitions for the messages in the calling application. If, however, the message data is required in the calling application, a function module can be used to read and return the data.

The API is called for the following functions:

- Message determination
- Editing the message proposal
- Message processing

If necessary, other function modules or dialog modules can be called from the function group.

Function Calls

Use

The application only calls the Message Control via the API.

Features

- Read messages
  - RV_MESSAGES_READ

    Existing message defaults are imported from table NAST into the internal table of the API.
- Message determination
  - COMMUNICATION_AREA_KOMxByy

    call the function module KOMxByy_FILL, which places the application data in the communication structure KOMxByy. These function modules must be written by the application programmer.

    Naming convention: x = K (for header) or P (for Position), yy = application code [Page 13].
  - MESSAGING

    process the condition component hierarchy and execute the condition access.
- Editing the message proposal
  - COMMUNICATION_AREA_KOMxByy
  - MESSAGING
  - RV_MESSAGES_MAINTENANCE

    The message default is determined and displayed on screen.
- Message processing
  - COMMUNICATION_AREA_KOMxByy
  - Number assignment for application documents
  - MESSAGING
  - RV_MESSAGES_RENAME
  - RV_MESSAGES_UPDATE

    Current data is entered in the communication structure (also described as the “communication block”). The message default is determined and sent (dependent on processing time [Page 49])
  - NAST_PROTOCOL_UPDATE

    The processing log is written. This last function module is called by the processing program, which the application stores as an attribute of the output type [Page 22].

Activities

Data declarations

The application should transfer your current partners to Message Control (see partner determination [Page 89]). Define the structure of the partner table in the calling program with the command

```
INCLUDE VMSGPART.
```

The data declarations must also be changed in the program KOMBSTRU in order to declare the communication structures of a new application in Message Control.

Assign values to the communication structure and the partner tables

The function modules KOMxByy_FILL, which enter the communication structures KOMxByy with the application data and write the current partners in the partner tables, must be put in their own functions group, for example VCOM. You should adhere to the SAP enhancement concept through the installation of *user-exits*.

Put the function modules COMMUNICATION_AREA_KOMxByy in the function group V61B.