BC The SAP Communications Server

Release 4.6C

SAP
Copyright

© Copyright 2001 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft®, WINDOWS®, NT®, EXCEL®, Word®, PowerPoint® and SQL Server® are registered trademarks of Microsoft Corporation.

IBM®, DB2®, OS/2®, DB2/6000®, Parallel Sysplex®, MVS/ESA®, RS/6000®, AIX®, S/390®, AS/400®, OS/390®, and OS/400® are registered trademarks of IBM Corporation.

ORACLE® is a registered trademark of ORACLE Corporation.

INFORMIX®-OnLine for SAP and Informix® Dynamic Server™ are registered trademarks of Informix Software Incorporated.

UNIX®, X/Open®, OSF/1®, and Motif® are registered trademarks of the Open Group.

HTML, DHTML, XML, XHTML are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

JAVA® is a registered trademark of Sun Microsystems, Inc.

JAVASCRIPT® is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

SAP, SAP Logo, R/2, R/3, ABAP, SAP ArchiveLink, SAP Business Workflow, WebFlow, SAP EarlyWatch, BAPI, SAPPARTHE, Management Cockpit, mySAP.com Logo and mySAP.com are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other products mentioned are trademarks or registered trademarks of their respective companies.
**Icons**

<table>
<thead>
<tr>
<th>Icon</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>Caution</td>
</tr>
<tr>
<td>🗣️</td>
<td>Example</td>
</tr>
<tr>
<td>💡</td>
<td>Note</td>
</tr>
<tr>
<td>🤔</td>
<td>Recommendation</td>
</tr>
<tr>
<td>✨</td>
<td>Syntax</td>
</tr>
</tbody>
</table>
Operating the Communications Server ................................................................. 84
Starting the Communications Server ............................................................... 85
Closing the Communications Server ................................................................. 86
Notes on Testing and Problem Analysis .......................................................... 87
Connection to SAP Systems ............................................................................ 89
Overview ......................................................................................................... 90
Hardware and Software Requirements (SAP Systems) .................................... 91
The Communications Component SAP ............................................................. 92
SAP-specific Entries in the SAPcomm Configuration File ................................. 93
Character Sets and Code Pages ...................................................................... 101
Testing the Connection ................................................................................ 102
Notes on Problem Analysis .......................................................................... 105
Connection to ODEX ...................................................................................... 108
The Communications System ODEX ............................................................... 109
Hardware and Software Requirements for ODEX ........................................ 110
The Communications Component ODX ......................................................... 111
ODX-specific Entries in the SAPcomm Configuration File ......................... 112
SAP-specific Aspects for ODEX Operation .................................................... 115
Notes on Testing and Problem Analysis: ODEX .......................................... 117
Connection to EXPEDITE ............................................................................ 119
The Communications System EXPEDITE ..................................................... 120
Hardware and Software Requirements: EXPEDITE .................................... 121
The Communications Component EXP ......................................................... 122
EXP-specific Entries in the SAPcomm Configuration File ......................... 123
SAP-specific Aspects for EXPEDITE Operation ........................................... 125
Notes on Testing and Problem Analysis: EXPEDITE ................................. 126
BC The SAP Communications Server ............................................................ 127
The Central Communications System ............................................................. 128
Hardware and Software Requirements .......................................................... 129
The Communications Component SSW ........................................................ 130
CPI-C Configuration .................................................................................... 131
SSW-specific Entries in the SAPcomm Configuration File ......................... 132
SAP-specific Aspects of Soft*Switch Operation ........................................... 134
Notes on Testing and Problem Analysis: Soft*Switch ............................... 135
Connection to OSILINK X400 ...................................................................... 137
The OSILINK X.400 Communications System ............................................ 138
Hardware and Software Requirements for OSILINK X.400 ....................... 139
The CX4 Communications Component .......................................................... 140
CX4-specific Entries in the SAPcomm Configuration File ......................... 141
SAP-specific Aspects of OSILINK X.400 Operation .................................... 142
Notes on Testing and Problem Analysis: OSILINK X.400 ......................... 143
Connection to TOPCALL ........................................................................... 145
The TOPCALL Communications System ..................................................... 146
Hardware and Software Requirements for TOPCALL ............................... 147
The Communications Component TOP ......................................................... 148
TOP-specific Entries in the SAPcomm Configuration File ......................... 149
SAP-specific Aspects of TOPCALL Operation ............................................ 151
Notes on Testing and Problem Analysis: TOPCALL ................................. 152
Connection to GammaFax ................................................................. 153
The GammaFax Communications System ........................................ 154
Hardware and Software Requirements .......................................... 155
The GAM Communications Component ....................................... 156
GAM-specific Entries in the SAPcomm Configuration File ............. 157
Installation .................................................................................. 159
Connection via APPLI/COM(T.611) ............................................. 160
The File Interface APPLI/COM (T.611) ......................................... 161
Hardware and Software Requirements .......................................... 162
Communications Component APP ............................................... 163
Installation .................................................................................. 164
APP-specific Entries in SAPcomm Configuration File .................. 165
Interface Specification .................................................................. 167
APPLI/COM Directories ............................................................... 168
APPLI/COM Requests ................................................................. 170
Transmission Requests ................................................................ 171
Completed Transmission Requests .............................................. 174
Reception Requests .................................................................... 176
The Communications Component TST ........................................... 178
Basics ......................................................................................... 179
TST-specific Entries in the SAPcomm Configuration File ............. 180
SAP-specific Aspects for TST Operation ....................................... 181
TST-specific Log Entries ............................................................... 182
SAPcomm Development Library .................................................. 183
Overview .................................................................................... 184
Scope of Price List Component .................................................... 186
The Structure and Tasks of a Communications Component .......... 187
Message Flow in SAPcomm ......................................................... 189
Message and Status Processing in SAPcomm ............................. 191
Programming With the SAPcomm Development Library .......... 194
Initialization ............................................................................... 195
SEND, RECEIVE and FUNCTION Requests .................................. 196
Termination ................................................................................ 199
Creating a SAPcomm Grammar File ............................................ 200
Testing a Communications Component ....................................... 203
Description of the SAPcomm API .............................................. 204
Functions and Parameters of the SAPcomm API .......................... 205
Message Attributes .................................................................... 212
SAP Return Codes for Message Transmission ............................ 216
Compiling and Linking the Example Program .............................. 217
Example Configuration ............................................................... 218
Working With the Graphical Interface ........................................... 220
The Graphical Interface: Basics .................................................. 221
Displaying Channel-related Activities ........................................... 222
Using the SAPcomm Menu Bar .................................................... 224
The SAPcomm Directory Tree .................................................... 225
Overview .................................................................................... 226
The SAPcomm Root Directory ..................................................... 227
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>The SAPcomm System Directory</td>
<td>228</td>
</tr>
<tr>
<td>ETC Directory</td>
<td>229</td>
</tr>
<tr>
<td>The Component Directories</td>
<td>230</td>
</tr>
<tr>
<td>The SAPcomm Auxiliary Programs</td>
<td>232</td>
</tr>
<tr>
<td>Overview</td>
<td>233</td>
</tr>
<tr>
<td>Analyzing SAPcomm Parameter Files (scmactl)</td>
<td>234</td>
</tr>
<tr>
<td>Testing a Serial Interface (scmasyn)</td>
<td>235</td>
</tr>
<tr>
<td>Checking the Configuration Data (scmcfgcc)</td>
<td>236</td>
</tr>
<tr>
<td>Analyzing SAPcomm Log Files (scmlog)</td>
<td>239</td>
</tr>
<tr>
<td>SAPcomm Monitor (scmmon)</td>
<td>240</td>
</tr>
<tr>
<td>Moving Files (scmmove)</td>
<td>241</td>
</tr>
<tr>
<td>Performing Actions (scmperfm)</td>
<td>242</td>
</tr>
<tr>
<td>Testing the Network Link to the SAP System (scmping)</td>
<td>243</td>
</tr>
<tr>
<td>Displaying SAPcomm Files (scmqueue)</td>
<td>244</td>
</tr>
<tr>
<td>Reorganizing SAPcomm Files (scmreorg)</td>
<td>245</td>
</tr>
<tr>
<td>Stopping the Communications Server (scmstop)</td>
<td>246</td>
</tr>
</tbody>
</table>
BC The SAP Communications Server

The SAP communications server allows EDI, Mail and telecommunications documents to be exchanged between SAP Systems (R/2 and R/3) and external communication partners.

Basics [Seite 10]

Hardware and Software Requirements [Seite 20]

Installation [Seite 26]

The Configuration File SAPCOMM.CFG [Seite 35]

Configuration in the SAP System [Seite 64]

The Control Component SCB [Seite 73]

Connection to SAP Systems [Seite 89]

Connection to ODEX [Seite 108]

Connection to EXPEDITE [Seite 119]

Connection to OSILINK X400 [Seite 137]

Connection to TOPCALL [Seite 145]

Connection to GammaFax [Seite 153]

Connection via APPLI/COM(T.611) [Seite 160]

The Communications Component TST [Seite 178]

SAPcomm Development Library [Seite 183]

Working With the Graphical Interface [Seite 220]

The SAPcomm Directory Tree [Seite 225]

The SAPcomm Auxiliary Programs [Seite 232]
Basics

Naming Conventions [Seite 11]
Compatibility [Seite 12]
Functions of the Communications Server [Seite 13]
Implementation [Seite 14]
How the SAP Communications Server Works [Seite 15]
The Configuration File [Seite 16]
The User Interface [Seite 17]
The Communications Components [Seite 18]
Naming Conventions

This guide distinguishes between the actual communications server and its respective auxiliary programs. Wherever this distinction is not necessary for comprehension purposes, the term “SAPcomm” is used to represent the "communications server”, the respective auxiliary programs, and also all of the products connected with the communications server.

The names of the files created by the communications server, the SAPcomm programs and all delivered files contain only lowercase letters. This also applies even if the name is written in uppercase in this guide in order to indicate its importance.
Compatibility

R/3:
All R/3 Releases as of 2.0B are supported. In R/3 Systems of Release 2.0, SAPcomm must be connected to the R/3 System via an SAP Gateway of Release 2.1 or higher.

R/2:
SAPcomm Version 3.0 supports the entire SAPcomm-relevant functionality of R/2 Versions 4.4D-4.4F and 5.0B-5.0F, as well as the 4.3 Versions modified to correspond to these versions. Where necessary, please ask your SAP contact person which SAPcomm version supports extended functionality of later R/2 versions.
Functions of the Communications Server

The SAP communications server links SAP Systems (R/2, R/3 or a combination of both) with each other and with remote communications partners. Its tasks cover the facilitating of data exchange via different communications paths (X.400, and so on) and the provision of a standard interface to the SAP system for these different communications paths.

The SAP communications server is the basis for the following functions:

- exchange of EDI documents.
- exchange of mail documents.
- exchange of telecommunications documents (telex, teletex, telefax).

The SAP communications server also allows you to link components other than those supported by the standard SAP system or non-SAP systems to the SAP communications server via the SAPcomm Development Library, the interface of the SAP communications (see The SAPcomm Development Library [Seite 183]).

For information on which of these functions are supported on an “application level” in which SAP Systems from which release levels, refer to the following product information (obtainable from Basis Sales):

- SAPcomm - Communications Services in the SAP Systems
Implementation

The SAP communications server is a system of independent programs. It is not part of an SAP system, but runs independently of SAP systems in a UNIX or OS/2 environment. It communicates with one or more SAP systems via a CPI–C link.

It consists of the following components:

- component to control all activities (see The Control Component SCB [Seite 73]).
- user interface component;
- communications components to connect the communications systems performing the data transmission.

The communications server can run on the same hardware as an R/3 application server and/or an R/3 database server, or also run independently of them on separate hardware.
How the SAP Communications Server Works

The sending and receiving of data is carried out via the SAP communications server using the store-and-forward procedure.

For example, to send a series of EDI documents, the communications server must establish the link to the SAP system and "fetch" the EDI documents; they are loaded from the SAP system into the "communications server environment" and stored there. The control component of the communications server passes the EDI documents to the communications component (or possibly to several components in each case), which then sends the data at a date determined in the "communications server time plan".

Conversely, incoming documents are also first stored in the communications server environment and passed on to the SAP system at a later date.

The initiative for the data exchange between an SAP system and the communications server is therefore controlled by the communications server alone.

Please note that the SAP Communications Server automatically carries out only those actions for which scheduling instructions have been entered in the configuration file.
The Configuration File

The communications server requires specific information for the data exchange activities (creation of the link to an SAP system, passing on documents to a responsible communications component, and so on).

You must enter this information in the configuration file (see The Configuration File SAPCOMM.CFG [Seite 35]).

You also enter information regarding the temporal sequence of all data exchange activities in the configuration file SAPCOMM.CFG. These statements are stored internally in a "time plan".
The User Interface

The communications server supplies information on the status of the system via a graphical user interface. This user interface allows you to initiate data exchange and to check the activities of the SAP communications server via a menu. (See Working With the Graphical Interface [Seite 220]). However, the communications server can also be operated without its graphical user interface.
The Communications Components

Communications Systems

The connection of the SAP communications server to the data networks and the actual data transmission is carried out via special communications systems (for example, a file transfer program with an adapter card or telefax box). The connection of such a communications system to the communications server is carried out via the "communications components".

A separate communications component is responsible for every communications system to be connected as part of the SAP communications server.

This communications component consists of an independent program, which is started or ended by the control component of the communications server whenever the communications server is started or stopped.

Example: The communications component SAP (see next topic) is implemented by program SCMISAP or SCMISAP.EXE under OS/2. Other communications components are implemented analog by the corresponding programs (SCMIODX, SCMITOP, etc.).

Various vendors have adapted their communications solutions to SAPcomm requirements. For a list of such vendors, see the SAP brochure "Communication Services in SAP Systems", ordering number 50 010 054. This list is not guaranteed to be complete.

Please note that basic SAPcomm functions must always be obtained from SAP even if you want to use non-SAP communications components for SAPcomm.

R/2 only:

Non-SAP communications components for SAPcomm require the following SAP components:

- 837 (OS/2)
- 840 (UNIX)

Link With SAP Systems

The communications component "SAP" is responsible for data exchange between the communications server and one or more SAP systems.

Data exchange is based on CPI-C - regardless of whether the communications server is installed on the same hardware as an R/3 presentation server, application server or database server or on separate hardware.

All communications subsystems participating in data exchange (for example VTAM...) must be suitably configured. The configuration of these communications subsystems is not covered in this document. You will find configuration examples in SAP Communication: Configuration.

Please also read the documentation supplied by each manufacturer.

Link With Remote Communications Partners

The SAP communications server supports different communications systems when linking up with remote communications partners. You can check what these are in the individual sections on the communications components.

It is also possible to link other communications systems together via the communications server interface SAPcomm-API.

The communications server can be operated with any combination of these communications components. A subsequent upgrade with further components is possible at any time.
Hardware and Software Requirements

Operating Systems [Seite 21]
The Connection to SAP Systems [Seite 22]
Communications Components [Seite 23]
The Control Component [Seite 24]
Disk Capacity [Seite 25]
Operating Systems

The control component and the user interface of the communications server can be run on the following operating systems:

- AIX
- HP-UX
- OS/2 (as of Version 2.1)
- SINIX
- OSF/1
- Solaris (Sun)

Which of these operating systems meets your requirements depends largely on which communications component(s) you wish to install. The control component must be installed on the same hardware as all of the linked communications components.
The Connection to SAP Systems

The connection of the communications server to SAP Systems (R/2 and R/3) is based on CPI-C.

R/2 Systems

The hardware and software requirements are identical to those of a CUA server. For further details, please refer to the following SAP brochure: SAP-supported Network Products

Connecting the communications server to SAP systems requires the configuration of a logical unit LU 6.2 reserved for the communications server.

For more information, please refer to the following SAP documentation: SAP Communication: Configuration

R/3 Systems

The connection between SAPcomm and R/3 Systems is handled via the SAP Gateway (CPI-C Handler). For more information, please refer to the following documentation: SAP Communication: Configuration
Communications Components

For information on which operating systems the individual communications components can run under, refer to the sections on the communications systems. Information in this section only refers to the control component of the communications server and the connection to SAP systems.

The communications systems linked have more specific requirements - concerning, for example, the platform, operating system version or other hardware components. For further details, please refer to the respective manuals of the manufacturers.
The Control Component

Special requirements concerning types of hardware, operating system version and main memory depend on how you want to operate the communications server:

- **On a separate computer:**
  The same requirements apply as for R/2 Workstation Software. These requirements are described in the SAP documentation S40.2 and in the brochure *SAP-Supported Network Products*.

- **Together with an R/3 presentation server, an R/3 application server and/or an R/3 database server on a shared computer:**
  The same requirements apply as for the R/3 components.

**OS/2:** Please note that only Versions 2.1 and above are supported.
Disk Capacity

The disk capacity required mainly depends mainly on the following factors:

- How large your message volume is.
- The time period during which messages are buffered.
  How long must the communications server or other participating communications systems buffer messages?
  (This is determined, among other things, by the time plan of the communications server.)
- Whether you archive messages via the communications server.
- The extent to which you use the communications server’s tracing options.

A useful disk capacity in practice is 50 MB, but this can of course only be an approximate value.
Installation

Overview [Seite 27]
Installation Under UNIX [Seite 28]
Installation Under OS/2 [Seite 30]
Modifying the Configuration and Communications Files [Seite 31]
Installing and Configuring Communications Systems [Seite 33]
Preparation in the SAP System [Seite 34]
Overview

This section describes how to install the communications server on a workstation, which is already linked to a network environment.

Neither the installation and configuration of the network software nor the installation and configuration of the communications systems to be linked to the communications server are described in this manual. For further information, please read the appropriate manufacturer documentation and the SAP documentation *SAP Communication: Configuration*.

This section describes a new installation. If you have already installed an older version of the communications server, note the following before installing a new version:

- **SAPcomm as of Version 2.1:**
  - Save your old configuration file `sapcomm.cfg` and the side info file.

- **All SAPcomm versions before 2.1:**
  - The files created in the "old" communications server environment cannot be processed by the "new" communications server. Please ensure that all old files (outgoing and incoming documents, status messages) are first processed before you install the new version of the communications server. It is recommended that you delete the old version (including all files, after first making a data backup) before the installation or at the latest after starting up the new communications server.
Overview

Installation Under UNIX
This topic describes how to install SAPcomm on a UNIX platform, which is linked to an R/2 or R/3 System.

The installation program creates the SAPcomm directory tree [Seite 226] and copies the files from the CD-ROM to the various directories.

User Authorizations
The directories, programs and non-executable files must be created so that the user, under which the communications server is started, can carry out the following actions:

• start all executable files (the programs) of the SAPcomm call directory,
• read all the files of the profile directory,
• create directories in the SAPcomm root directory (write authorization),
• search for files in the SAPcomm root directory (read authorization for this directory).

Files created by the communications server are given the system-wide defined access mode (permissions). To restrict access authorizations to the files created by the server, use the UNIX command umask for the user under which the communications server is started.

MOTIF Parameters for SAPcomm
You can configure the SAPcomm user interface (the window which appears on the screen after starting the sapcomm program) using the procedure generally applicable to MOTIF programs. For example, if you want to use a font other than the default font for the SAPcomm window, insert the parameter

• sapcomm*labelFontList: <font name>
  This parameter affects all texts in menus and pop-up windows.
• sapcomm*FontList: <font name>
  This parameter affects all other SAPcomm texts.

You can change the size (sapcomm.height, sapcomm.width) and color of the window in exactly the same way. For further details on this, please read your Motif manual.

Installation with an R/2 System
An installation program is provided to install SAPcomm on a UNIX platform, which is linked to an R/2 System. It can be found on the delivery CD.

Installation with an R/3 System
To install SAPcomm on a UNIX platform, which is linked to an R/3 System, you have the following options:

Before R/3 Release 2.1D
To install SAPcomm on a UNIX platform, which is linked to an R/3 System earlier than Release 2.1D, please contact your SAP support representative.

After R/3 Release 2.1D
The installation is described in a separate document, sapcomm.install.doc. To install SAPcomm, use the script sapcomm.install.

Both of these files are included in the R/3 delivery package.

After R/3 Release 2.2B
The SAPcomm installation program can be called as an option in the R/3 installation menu.
Overview

Installation Under OS/2
The installation program contained on the CD-ROM installs the SAP Communications Server. The installation program creates the SAPcomm directory tree [Seite 226] under the root directory /sap/com.

It copies the files from the CD-ROM and modifies the OS/2 configuration file CONFIG.SYS. After installation, you will find the following files in the directory /sap/com/etc:

- The model configuration file sapcomm.cfg
  Copy it to the SAPcomm profile directory, if a SAPcomm configuration file not yet exits.

  ![Note]

  Modifications to SAPcomm configuration only take effect if they are performed on the configuration file stored in the profile directory.

- Files that are not essential for operation of SAPcomm (sample source file, etc.)

Instructions for using this program are contained in a separate guide, which is included.

Please note that for certain communications systems (ODEX...), the communications server must be installed on the same drive as the communications system. Further details on this can be found in the sections for the respective communications components.
Modifying the Configuration and Communications Files

The installation program creates the ETC Directory [Seite 229]. This directory contains the example configuration file sapcomm.cfg. Copy this file to the SAPcomm profile directory if no sapcomm.cfg file exists there. In this directory you will also find files that are not directly required for the operation of SAPcomm (example source code, etc.). Modifications to SAPcomm configuration only take effect if they are performed on the configuration file located in the profile directory.

After installation you must adapt the following files to suit your local requirements:

- configuration file sapcomm.cfg,
- sideinfo file,
- profile file sapcomm.pfi

You can now put the communications server into operation (the correct installation and configuration of all required network links as well as the linked communications programs is a prerequisite for this).

Please also read the topic Starting the Communications Server [Seite 85]

Configuration File sapcomm.cfg

The configuration file must be located in the SAPcomm profile directory. Maintain this file as described in the guidelines Configuration in the SAP System [Seite 64]. Please also note "Testing the Configuration Data [Seite 51]."

Side Info File

The communications server needs certain network-specific information to establish a link to an SAP system. It extracts this information from the parameter SYSTEM of the SAPcomm configuration file (communications component SAP), and from the side info file.

For a description of side info files, refer to the documentation SAP Communication: Configuration.

Example File for OS/2

The installation program copies a side info example file called SIDEINFO to the ETC Directory [Seite 229]. The installation program also generates (in the CONFIG.SYS file) the environment variable SIDE INFO with the value <Profile directory>/SIDEINFO. SAPcomm can only establish a connection with an SAP system if this environment variable correctly points to your side info file.

If you change the name or the directory of your side info file, you must also change the definition of the environment variable SIDE INFO accordingly.

If your system configuration file CONFIG.SYS already contains the definition of the environment variable SIDE INFO, this variable will be overwritten by the installation program.

The format of the side info example file corresponds to the description in the documentation SAP Communication: Configuration.

You can use the program CVSIDE to create a file with the new format from the "old" SAPCPIC.TBL file. To do this, enter the following command in the directory SAP:

    CVSIDE sapcpic.tbl > sideinfo

(You may need to specify the full path for CVSIDE.)

Please read the notes on the side info file in the section on the communications component SAP.
Overview

Profile File sapcomm.pfl
The profile file sapcomm.pfl is needed both for the UNIX and OS/2 environments. It is included with the delivery of SAPcomm.

OS/2:
During installation, the file is copied to the profile directory. You do not need to change this file.

UNIX:
It contains the value for the SAP parameter SAPSYSTEM. This value must be unique to SAPcomm - it may not already be assigned in another SAP profile file. You can choose any value you wish and replace the default with this new value.
Installing and Configuring Communications Systems

Before you can exchange data with your communications partners, you must:

- install your communications systems (ODEX, etc.) including the respective hardware and driver software (EICON, etc.)
- configure the installed communications systems.

Please refer to the respective guides for guidelines on the installation and configuration of your communications system and the respective adapter hardware and software.

Operation of the communications server requires the correct installation and configuration of your communications systems. Therefore, you must first test your communications system without starting the communications server.

The sections on the communications components explain how the communications server works together with your communications systems, and which SAP-specific aspects must be taken into account before configuration.

VAN Connection

When connecting to a VAN, specific preparations must be made during SAP communications server installation (registration, passwords, etc.). For further details on this, consult your VAN supplier and request the appropriate guides.
**Preparation in the SAP System**

You must make a number of preparations in every SAP System, to which the communications server is to be linked:

**R/2:**
- Create user
  Create the user, under which the communications server logs on to the R/2 System as follows:
  - as a CPI-C user (as of Release 5.0)
  - with authorization to maintain Tables EDCT and SKWD
  - other authorizations depending on the application
- Create Tables EDCT and SKWD
  Create Tables EDCT and SKWD in the SAP System in all clients, with which SAPcomm is to be linked. These tables exist only in client 0 in the standard delivery.
- For EDI as of Release 5.0F
  The SAPcomm-ID must be configured as a port (parameter IDSAPCOMM, see the section [Connection to SAP Systems](Seite_89)).

**R/3:**
- Create user
  Create the user, under which the communications server logs on to the R/3 System as follows:
  - as a CPI-C user
  - with the profile S_SKOM_SRV, which is contained in the R/3 shipment
The Configuration File SAPCOMM.CFG

Overview [Seite 36]
Structure of the Configuration File [Seite 37]

TRACE Parameter Group
Channels - Access to the Communications Components [Seite 44]
How are Documents Allocated to “Their” Channel? [Seite 46]

SAPcomm Schedule Statements [Seite 48]
Testing the Configuration Data [Seite 51]

Configuration Examples [Seite 52]
Overview

The configuration file is used to configure the SAP communications server. The parameters that you enter here allow you to determine:

- with which SAP systems the communications server can exchange data
- with which communications components you operate the communications server
- when the communications server carries out which activities
- routing:
  - which communications component "sends" a document and via which channel (Incoming messages: To which SAP system a document is forwarded.)

The configuration file is a text file. Maintain this file with an editor (for example, the SAP editor). Two examples of complete configuration files can be found at the end of this section.
Structure of the Configuration File

In the configuration file, parameters related by content are grouped together in parameter groups of a certain category. The category of the parameter group determines which parameters and which other parameter groups (nesting) the group can contain.

Only parameter groups of the category PROGRAM are permitted at the "upper" level. This means that the entire configuration file consists exclusively of parameter groups with the following structure:

```
PROGRAM <identifier>
(Parameter specification)
END-PROGRAM
```

Parameter groups of other categories have different start and end lines.

The parameter groups of the category PROGRAM enable you to specify the communications components with which the communications server is operated.

You must include a separate parameter group in the configuration file for every communications component - each with its respective identifier. You can look up the identifiers in the sections on the individual communications components.

The SCB Parameter Group

You must also include a PROGRAM parameter group in the configuration file in which you enter parameters for the control component of the communications server. This parameter group has the identifier "SCB" (the identification code stands for SAPcomm Basis).

The information in the following two sections "Parameter Specification" and "Referencing Parameter Groups" about syntax applies to the parameter group SCB and for all the parameter groups. However, the contents of the parameter group SCB differ from the remaining PROGRAM parameter groups. Parameter group SCB is therefore discussed separately in The Control Component SCB [Seite 73]

Parameter Specification

The parameters belonging to a parameter group are specified in the form of parameter lines:

```
PARAMETER-NAME PARAMETER-VALUE
```

or - when grouping several parameters into a block in the form of parameter groups (now as a "subgroup" of the current parameter group).

Either "ON" or "OFF" must be entered as a PARAMETER-VALUE for a series of parameters or parameter groups. This value activates or inactivates the respective parameter or parameter group.

Referencing Parameter Groups

Reference to parameter subgroups can be made elsewhere in the configuration file (an example of this is the parameter group DESTINATION specified below). The reference is made for every parameter whose PARAMETER-NAME is identical to the category of the parameter group which is to be referenced.

Example:

Referenced group: DESTINATION DEST1
Structure of the Configuration File

END-DESTINATION

Additional subgroup: ANOTHER_GROUP GR1

Ref. to group DEST1: DESTINATION DEST1

END-ANOTHER_GROUP

A referenced parameter group must always come BEFORE the parameter(s) that reference(s) it in the configuration file. Otherwise, the sequence of parameter lines and parameter groups is variable.

The PROGRAM Parameter Group

Parameter groups of the category PROGRAM have the following structure:

PROGRAM <identifier>

DESTINATION <destination name>

<destination parameter>

END-DESTINATION

SELECTION <selection name>

<selection parameter>

END-SELECTION

PERIODIC <period name>

<period parameter>

END-PERIODIC

PERFORM <perform name>

<schedule statements>

END-PERFORM

CHANNEL <channel name>

PERFORM <perform name>

<schedule statements>

END-PERFORM

<channel parameter>

END-CHANNEL

TRACE <ON|OFF>

TRACE_ALL <ON|OFF>

INIT <ON|OFF>

PGM_FLOW <ON|OFF>

DATA_FLOW <ON|OFF>

CS_RESPONSE <ON|OFF>

FILES <ON|OFF>

IPC <ON|OFF>

MEMORY <ON|OFF>

ROUTING <ON|OFF>

QUEUE_MSG <ON|OFF>

DEV_CTRL <ON|OFF>

SCI_CALL <ON|OFF>

SCI_PARAM <ON|OFF>

SCI_ATTR <ON|OFF>
The basic structure shown above is common to all parameter groups of the category PROGRAM. However, not all the parameters or parameter groups listed above need exist. Multiple instances of some of these groups - for example, CHANNEL, SELECTION - and also certain other parameters can exist.

The contents of the individual PROGRAM parameter groups can vary depending on the communications component. The following only describes the parameters which can or do occur in all PROGRAM parameter groups. The parameters specific to a certain communications component (these are the "component-specific parameters") are described in the section on the respective communications component.

Identifier
Every communications component connected to SAPcomm is identified by a three-digit alphanumeric identifier. By specifying this identifier as the "name" of a PROGRAM parameter group, you establish the reference to the respective communications component.

DESTINATION Parameter
Group together parameters that are required to create a link in parameter groups of the category DESTINATION. These parameters can then be referenced within the respective PROGRAM parameter group via the chosen destination name. The destination name consists of a maximum of ten alphanumeric characters and must be unique within its PROGRAM parameter group.

The specification of a destination parameter group is not a necessary requirement and depends on the participating communications component anyway. When, and in connection with which communications components, these parameter groups are necessary is explained in the sections on the individual communications components that are of interest to you.

SELECTION Parameter
You define groups of selection criteria for retrieval of documents from a target system or from a remote communications partner in parameter groups of the category SELECTION (for example, selective retrieval of outgoing documents from an SAP system).

The creation of a special SELECTION parameter group alone has no effect - only if it is referenced in a PERFORM parameter group with the action message RECEIVE do the selection criteria defined in it take effect. Please read "SAPcomm Schedule Statements [Seite 48]."

Enter any - alphanumeric - name with up to ten characters as the selection name. The only requirement is that all the selection names you use must be unique within their PROGRAM group.
Selection parameter
The actual selection parameters are specific to every communications component. Please read the individual sections on the communications components for information on which selection parameters you can specify in each case.

PERIODIC Parameter
You group together parameters for the execution times of action messages in parameter groups of the category PERIODIC. You can refer to these parameter groups from all PERFORM groups by specifying the period name. The contents of the PERIODIC parameter groups are explained in "SAPcomm Schedule Statements [Seite 48]."

PERFORM Parameter Group
Here you can use PERFORM parameter groups to specify the execution of communication components which support administrative functions (reorganization, and so on), and the times at which they are to be carried out. Please bear in mind that, in PERFORM groups that are direct subgroups of a PROGRAM parameter group, you can only enter statements for the execution of ADMINISTRATIVE functions. Statements for the execution of data transfer functions are only possible within CHANNEL parameter groups (see below). Please read the sections on the communications components for details of which administrative functions are supported by which communications components. The contents of the PERFORM parameter groups are explained below in "SAPcomm Schedule Statements [Seite 48]."

CHANNEL Parameter
Allocate "channels" to the communications component using the parameter groups of the category CHANNEL. Assign every CHANNEL parameter group ("every channel") a SAPcomm-wide unique channel name with a maximum of ten alphanumeric characters. Please note: The channel name that you assign is also used as a directory name (see The SAPcomm Auxiliary Programs). If your file system does not allow directory names of ten characters, you must limit yourself to correspondingly shorter channel names.
As explained in the section "Channels - Access to the Communications Components [Seite 44]," outgoing documents can only be forwarded to communications components via the channels configured here and incoming documents can also only be fetched from the communications components via channels. Specify when these actions - forwarding and fetching of documents - are to be carried out in the PERFORM parameter groups. The contents of the PERFORM parameter groups are explained in "SAPcomm Schedule Statements [Seite 48]."
The structure of a CHANNEL parameter group is described in "Channels - Access to the Communications Components [Seite 44]."

TRACE Parameter Group
The TRACE parameters are described in the section TRACE Parameter Group [Seite 42].

OWNSESSION Parameter
This parameter is only relevant if you use SAPcomm-API to develop your own communications component (only for the OS/2 version of SAPcomm). Specify OWNSESSION ON to start your component in a separate OS/2 session.

REQUEST_INTERVAL Parameter
Optional. Enter here at what time intervals the communications component is to be queried for status confirmations and incoming messages received by the connected communications system (in the SAPcomm user interface: "Local Request"). Default: 5 minutes.

**LOGFILE_EXPIRATION Parameter**

Optional. Enter here after how many days old log files created by the respective component are to be automatically deleted. If you do not enter this parameter, the respective log files are not deleted.

**ARCHIVE Parameter**

Optional. If this parameter is entered, the files sent via the respective communications component are not deleted but left on the workstation hard disk - under different names. These files are not deleted by the communications server.

**SAVE_RESPONSE_INFORMATION Parameter**

Optional. If this parameter is entered, "responses" created by the connected communications system (to the statements created by the communications server) are saved. Corresponding log file entries allow these archived responses to be retrieved for later analysis. These files are not deleted by the communications server.

**SENDREQUEST_BLOCKSIZE Parameter**

You can use this parameter to instruct a communications component to interrupt the transmission of outgoing messages to the connected communications system after a particular number of files.

After the interruption, the communications component is available for other tasks (e.g., the fetching of status messages). The transmission of outgoing messages is resumed once other activities (that may be waiting in a queue) have been processed, irrespective of schedule configuration.

**Component-specific Parameters**

Other PROGRAM parameters specific to the respective communications component can exist in addition to the stated parameters of the parameter group PROGRAM. Please read the sections on the communications components for information on which individual parameters these are.
**TRACE Parameter Group**

If you specify the parameters of this optional group, information on run-time will be logged. You can use this information to fine-tune the SAPcomm schedule (see "Log Directory" in The SAPcomm Auxiliary Programs).

You can activate this group by specifying "ON" (default) or deactivate it by specifying "OFF". You can enter any combination of the following parameters (you can switch every individual parameter on by specifying ON or turn them off by specifying OFF. Specifying OFF has the same effect as if you had not entered the parameter at all).

**TRACE_ALL** <ON|OFF>
If this parameter is specified (with 'ON'), all the trace functions are activated. The specification of other trace parameters is then unnecessary.

**INIT** <ON|OFF>
If this parameter is entered, then information on the initialization phase of the communications component is logged.

**PGM_FLOW** <ON|OFF>
If this parameter is specified, then detailed information on the program execution of the communications component is logged.

**DATA_FLOW** <ON|OFF>
This parameter is only important for communications components that use a non-file-based type of interface to exchange data with their target system or the connected communications program. In this case, you can trigger the logging of exchanged data (or a part of it) by using the parameter DATA_FLOW.

**CS_RESPONSE** <ON|OFF>
If this parameter is entered, then information (for example on confirmations) supplied by the connected communications system is logged.

**FILES** <ON|OFF>
If this parameter is entered, information on the file-related activities of the communications components is entered into the log file.

**IPC** <ON|OFF>
If this parameter is entered, then information on the use of functions for process-to-process communication is entered into the log file by the communications component. This is only provided for internal SAP purposes.

**MEMORY** <ON|OFF>
If this parameter is entered, information on the administration of the memory (allocation, ..) is entered into the log file.

**ROUTING** <ON|OFF>
If this parameter is entered, information on the assignment of documents to the channels responsible for sending these documents is entered into the log file. Use this parameter if, for example, incoming documents are allocated to an "incorrect" SAP System - the log entries could help you to find an incorrect entry in the routing entries (the PROGRAM parameter group SCB).

**QUEUE_MSG** <ON|OFF>
If this parameter is entered, SAPcomm-internal messages are logged. This is only provided for internal SAP purposes.

**DEV_CTRL** <ON|OFF>
If this parameter is entered, information on the control of external devices (for example modems) is logged.
SCI_CALL <ON|OFF>
By specifying this parameter, you cause every call of one of the functions related to the activation of a communications component to be logged.

SCI_PARAM <ON|OFF>
By specifying this parameter, you cause every transfer of a parameter to the respective communications component (or from it) to be logged.

SCI_ATTR <ON|OFF>
By specifying this parameter, you cause every transfer of a message attribute to the respective communications component (or from this) to be logged.

SCI_PERF <ON|OFF>
If you specify this parameter, information on the duration of individual (PERFORM) activities or periods of SAPcomm inactivity is entered in the log file of the respective SAPcomm component.
Channels - Access to the Communications Components

The transfer of data from the control component to a communications component and from a communications component to the control component always takes place via the channels configured in the configuration file.

Sending Messages
A document is not assigned directly to the communications component that is to send it, but to a channel. The allocation of a document to a channel determines the following:
- the communications component that is responsible for sending;
- the time at which the document is transferred to the communications component for sending;
- parameters required by the communications component (dependent on the respective communications component);
- assignment to a destination configured for the respective communications component (dependent on the respective communications component).

Retrieving Messages
You insert statements into the configuration file, with which you trigger the communications server to retrieve messages from external communications partners (for example a VAN or SAP System to which the communications server is linked). This kind of statement does not refer directly to the communications component responsible, but to a channel. The channel determines the following:
- the communications component that is to retrieve the messages;
- the time at which these requests are to be carried out;
- the address from which the messages are to be retrieved (dependent on the respective communications component);
- selection parameters

Configuring Channels
You configure one or more channels for every communications component - that is, you enter one or more CHANNEL parameter groups within every PROGRAM parameter group in the configuration file.

CHANNEL Parameter Group
A CHANNEL parameter group has the following structure:

```
CHANNEL <channel name>

PERFORM <perform name>
  <schedule statements>
END-PERFORM

DESTINATION <destination name>
  <component-specific parameter>
END-CHANNEL
```

The parameters of the CHANNEL parameter group are explained below:

PERFORM Parameter Group
For every channel, you specify in one or more PERFORM parameter groups the time at which documents allocated to this channel are to be sent and, if necessary, the time at which data is to be retrieved.

The contents of the PERFORM parameter groups are explained in "SAPcomm Schedule Statements [Seite 48]".

**DESTINATION Parameter**

By specifying this parameter, you can link the current channel to a DESTINATION parameter group defined within the current PROGRAM group (for example, an SAP System). This assignment has the following effect: messages that are to be sent via this channel, and for which there is no recipient address (or at least none that is sufficient for sending), are sent to the address entered in the DESTINATION group.

The specification of this parameter is only useful for certain communications components (it is, for example, mandatory for the communications component SAP).

Please read the sections on the communications components in question for information on whether this parameter must be specified.

**Component-specific CHANNEL Parameters**

Other component-specific CHANNEL parameters can be entered in addition to the stated parameters. Details on the parameters can be found in the sections on the communications components.
How are Documents Allocated to “Their” Channel?

Every document to be sent is allocated to a channel (see previous topic "Channels - Access to the Communications Components [Seite 44]"). This assignment is made by the communications component that receives the document "from outside" (from the externally linked communications system or from an SAP System).

The (receiving) communications component determines the channel responsible for sending a document in one of the following two ways:

- the communications component is informed of the channel name by the (external) system from which the received document originates. This way of determining the channel name is only used for SAP-EDI and SAP Telecommunications outgoing documents. For further details on this, please read "SAP-EDI" and "Telecommunications from SAP Applications" below.

- The channel name is determined for every individual message using the routing statements stored in the SCB parameter group in the configuration file.. For further details, please read "The Routing Parameter Group [Seite 78] ".

You must store certain information in the SAP System so that SAP outgoing messages are allocated to the "correct" channels (either directly or via the routing statements in the configuration file). The following text explains how to do this in the areas EDI, Mail and Telecommunications.

The following topics describe how to assign the correct channel to a message to be exchanged in the following applications:

- SAP-EDI
- R/2 R/Mail and R/3 Office
- Telecommunication services for SAP documents

SAP-EDI

In the "partner communications addresses" of the SAP-EDI System, you allocate one or more "communications services" to every partner with whom you want to exchange EDI messages. Please enter the name of the channel via which you want to send the documents in question as a communications service.

In the case of incoming messages, the communications service allocated to the message, together with the name of the sender contained in the EDI message, serve to uniquely identify the sender. The communications service of the respective sender must therefore also always be transferred to the SAP-EDI System together with an incoming message.

Please read the guidelines in the sections on the communications components for details of how to allocate the communications service to an incoming message.

R/2 R/Mail and R/3 Office

On the transmission screen in the SAP mail system, you enter a "transmission type" for every message that you want to send to an external recipient. The transmission type determines the SERVICE parameter group (in the routing statements of the PROGRAM parameter group SCB), in which the responsible channel is determined (for example, "SSW" for Soft*Switch(SNADS) -> SERVICE parameter group is SSW, "X40" -> SERVICE parameter group is X400 ).

For further details on this, please read the topic "The Routing Parameter Group [Seite 78] " in the section on the control component.

46 April 2001
Telecommunications Services for SAP Documents

Documents that are created by SAP applications and are to be sent via the telecommunications services (telefax, telex, teletex) are transferred to the SAP spool system. In the course of this, they are assigned a “logical destination”. For details on how the selection of the destination is carried out, refer to “Telecommunications Tables” in “SAPcomm Administration in the SAP System”.

The identification of this destination (for example, "TF01" for telefax) also directly determines the name of the channel in the SAP communications server. An indirect assignment destination --> channel is not (currently) possible - that is, the SAPcomm channels for telecommunications services must have identical names to the logical destinations in SAP.
SAPcomm Schedule Statements

You specify when the communications server is to carry out which activities using the PERFORM parameter groups - if necessary, together with PERIODIC parameter groups. The parameters entered in these groups are stored internally as a “SAPcomm schedule”.

PERFORM parameter groups can be placed in two positions within a PROGRAM parameter group:

- As a direct subgroup of the PROGRAM parameter group
  This triggers the execution of PROGRAM-specific administrative functions, providing these are supported by the respective communications component.

- As subgroups of the CHANNEL parameter groups
  This triggers the transfer of documents to the respective communications component (to send) and the transfer of statements to the communications component to retrieve documents from remote communications partners.

PERFORM Parameter Group

A PERFORM parameter group has the following structure:

PERFORM <perform name>

ACTION <function>
DAY <weekday>
TIME <time of execution>
PERIODIC <period name>
SELECTION <selection name>
END-PERFORM

The specifications TIME and PERIODIC are not mandatory. If they do not appear in a PERFORM group, the actions specified in them are not carried out automatically, but can be started from the graphic SAPcomm user interface.

The parameters of the PERFORM parameter group are explained below:

Name of the PERFORM Parameter Group

Enter a name for the PERFORM group with a maximum of twenty characters. This name must be unique within its respective PROGRAM or CHANNEL main group. When starting actions from the graphic SAPcomm user interface, the list of “perform names” is displayed for you to select an action. It is therefore recommendable to assign meaningful names for selection purposes.

ACTION Parameter

You enter the function that the communications server is to carry out here. You must insert at least one ACTION parameter into a PERFORM parameter group. Several such parameters can be specified, depending on the communications component.

The functions which you can specify within a CHANNEL parameter group are:

SEND-SCHEDULE

Transfers the documents to the communications component that are not marked for night dispatch. The communications server checks at the times specified in the PERFORM group whether there are documents allocated to the present channel. If this is the case, these documents are transferred to the connected communications system. Once the transfer begins, it is the responsibility of the connected communications system to handle...
the sending of the documents. Please note that "sending" (from the view of the 
communications server) for the communications component SAP means transferring 
incoming documents to the SAP System!

**SEND-NIGHT**
Transfers the documents marked for night dispatch to the communications component.
Caution: these documents are only passed on for sending due to a "SEND-NIGHT" 
statement. If you have not specified such a statement for a certain channel, then 
documents allocated to this channel and marked for night dispatch remain in the 
communications server environment and are not sent! If you do not work with 
documents selected especially for night dispatch, then you need not enter a "SEND- 
NIGHT" statement.

**RECEIVE**
Creates the link to the remote communications partner and fetches supplied documents.
If the connection setup of the respective communications system is carried out 
asynchronously (relative to the communications component), the incoming documents 
are initially only transferred to the environment of the communications component. In a 
second step, the communications component is asked whether incoming documents 
exist. If this is the case, these incoming documents are transferred and allocated to 
"their" channel. The specification of this function is of course only useful in connection 
with communications components that can fetch active documents from a remote 
communications partner.

The functions that can be specified, if the PERFORM group is a direct subgroup of the 
PROGRAM parameter group, depend on the respective communications component. 
For details of which administrative functions (reorganization, and so on) are supported in 
each case, please read the sections on the individual communications components.

**DAY Parameter**
Optional. Enter this parameter if the function is only to be carried out on a certain day of 
the week. You can include several DAY parameter lines in a PERFORM group.
Permitted parameter values are: MONDAY, TUESDAY...

**TIME Parameter**
Optional. Enter in the format HH:MM (hours:minutes) the time (each day) at which the 
entered function is to be carried out. You can include several TIME parameter lines in a 
PERFORM group and also combine them with the PERIODIC time statements.

**PERIODIC Parameter**
Optional. You can refer here to a PERIODIC parameter group (see below) by using a 
predefined name <period name>. You can include several references to PERIODIC 
parameter groups in a PERFORM group and also combine them with the TIME 
statements.

**SELECTION Parameter**
Only affects the function RECEIVE. Enter the name of a SELECTION parameter group 
here, if you only want to retrieve documents from the remote communications partner 
that correspond to certain selection criteria (for example, requesting EDI documents from 
the SAP System that only contain certain message types). You can include several 
references to a SELECTION parameter group in a PERFORM parameter group.

**PERIODIC Parameter Group**
You refer to PERIODIC parameter groups in the PERFORM parameter groups if you want to 
make SAPcomm carry out an action not only once a day (parameter TIME), but also periodically.
SAPcomm Schedule Statements

In PERIODIC parameter groups, you can enter the intervals (specification in minutes) and in which part of the day the respective action is to be carried out.

You create the PERIODIC parameter groups as direct subgroups of the respective PROGRAM parameter group. The parameters of these parameter groups are explained below.

```
PERIODIC <period name>
    INTERVAL <length in minutes>
    FROM <starting time>
    UNTIL <finishing time>
END-PERIODIC
```

Name of the PERIODIC Parameter Group

You enter a name here with a maximum of twenty characters. You can refer to this name by specifying the PERIODIC parameter within the PERFORM groups.

INTERVAL Parameter

You enter here the intervals at which SAPcomm is to carry out a certain action ("whose" PERFORM group refers to the current PERIODIC group).

FROM Parameter

Optional. You can limit the period to which the PERIODIC statement refers to a part of the day. If you want to do this, enter the starting time here in the format HH:MM (hours:minutes).

UNTIL Parameter

Specify the end time here in the format HH:MM (hours:minutes). The finishing time can also be "smaller" than the starting time - that is, a period of a day from 20:00 to 7:00 is permitted.
Testing the Configuration Data

The configuration parameters stored in the configuration file sapcomm.cfg are not read directly from the control component of the communications server, but must first be checked, then converted into an SAPcomm-internal format and finally stored in the file sapcomm.bcf (binary configuration data). These functions are carried out by the SAPcomm auxiliary program scmcfgcc.

The control component of the communications server starts the auxiliary program independently if required - that is, if the binary configuration file sapcomm.bcf does not exist or the configuration file sapcomm.cfg has been changed.

On the other hand, you can also start this auxiliary program directly without starting the communications server - for example, to check a change made to the configuration file sapcomm.cfg, particularly if the following error message appears when starting SAPcomm: Binary configuration file can’t be generated.

For more information on scmcfgcc and problem analysis, see SAPcomm auxiliary programs under [Checking the Configuration Data (scmcfgcc) [Seite 236]
Configuration Examples

Configuration Example for Odex/PC and Expedite [Seite 53]

Configuration Example for TOPCALL [Seite 60]
Configuration Example for Odex/PC and Expedite

Example of a SAPcomm configuration file with Odex and Expedite. If one of these components are not installed, you must delete the corresponding PROGRAM parameter group (ODX or EXP). Refer also to the configuration example for the TOPCALL connection.

***********************************************************************
* PROGRAM SCB
*    Delete SCB log files after 10 days:
*       LOGFILE_EXPIRATION   10
*
*    Routing parameter group:
*       ROUTING SAPCOMM
*
* SERVICE OFTP
*    TO-CHANNEL       CH-SAPSYS1
*    END-SERVICE
*
* SERVICE EXPEDITE
*    TO-CHANNEL       CH-SAPSYS1
*    END-SERVICE
*
  END-ROUTING
*
  TRACE ON
  PGM_FLOW          ON
  FILES             ON
  ROUTING           ON
  END-TRACE
*
  END-PROGRAM
***********************************************************************
* *
*......PROGRAM parameter group SAP: Connection to SAP Systems
* *
PROGRAM SAP
*
  LOGFILE_EXPIRATION10
  IDSAPCOMM           SAPCOMM1
  ARCHIVE             ON
*
  TRACE ON
  PGM_FLOW           ON
  DATA_FLOW          ON
  CS_RESPONSE ON
  FILES             ON
  ROUTING           ON
***********************************************************************

April 2001
SCI_ATTR    ON
END-TRACE

DESTINATION SAPSYS1
SYSTEM     SAPPROD
    CLIENT  000
    USER    MYUID
    PASSWORD MYPASS
    LANGUAGE D
    RELEASE  R3-21D
END-DESTINATION

*        SAP selection parameter groups.
*
*        Select all EDI documents:
*
SELECTION SE-ALL
    APPLICATION EDI
END-SELECTION

*        Select all EDI documents to be sent to partners
*        beginning with LIE:
*
SELECTION SE-EDI
    APPLICATION EDI
    SELECT-OPTIONS SE1
        FIELD   PARTNUMB
        LOW     LIE*
        OPTION  CP
        SIGN    INCL
    END-SELECT-OPTIONS
END-SELECTION

*        PERIODIC parameter groups:
*
PERIODIC P3HOURS
    INTERVAL 180
    FROM 07:30
    UNTIL 19:00
END-PERIODIC

PERIODIC P15MIN
    INTERVAL 15
    FROM 07:30
    UNTIL 19:00
END-PERIODIC

*        Channel for upload and download:
*
CHANNEL CH-SAPSYS1

DESTINATION SAPSYS1

PERFORM SEND
   ACTION SEND-SCHEDULE
   PERIODIC P15MIN
END-PERFORM

* Define action "Get all outgoing EDI messages"
* Execute from Monday to Thursday at 18.00
*

PERFORM RCV-ALL
   SELECTION SE-ALL
   ACTION RECEIVE
   DAY MONDAY
   DAY TUESDAY
   DAY WEDNESDAY
   DAY THURSDAY
   TIME 18:00
END-PERFORM

* ...and earlier on Fridays:
*

PERFORM RCV-ALL_F
   SELECTION SE-ALL
   ACTION RECEIVE
   DAY FRIDAY
   TIME 15:00
END-PERFORM

* Messages for all partners LIE* are collected
* every three hours:
*

PERFORM RCV-EDI-LI
   SELECTION SE-EDI
   ACTION RECEIVE
   PERIODIC P3HOURS
END-PERFORM

END-CHANNEL

END-PROGRAM

***********************************************************************
* .........PROGRAM parameter group ODX: Connection to ODEXPC *
*
PROGRAM ODX

   LOGFILE_EXPIRATION 10
ARCHIVE ON
SAVE_RESPONSE_INFORMATION ON
*
* Only collect incoming documents from Odex,
* whose recipients are assigned in Odex to a
* "Local Code" beginning with 'E':
*
RECEIVER_LC E*
*
* All documents transferred from SAPcomm to Odex
* are given the Virtual File Name 'SAPCOMM':
*
VIRTUAL_FN SAPCOMM
*
* All documents received via Odex are assigned
* to the communications service 'ODEX':
*
DEF_COMMSERVICE ON
*
TRACE ON
  TRACE_ALL OFF
  PGM_FLOW ON
  DATAFLOW ON
  CS_RESPONSE ON
  FILES ON
  ROUTING ON
END-TRACE
*
* Define destinations for collecting messages
* from a VAN or directly from a partner
* :
*
DESTINATION LF01
*
* Supplier assigned the Local Code LOC1 in Odex
* :
*
  LOCALCODE LOC1
END-DESTINATION
*
* PERIODIC parameter groups:
*
PERIODIC P10MIN
  INTERVAL 10
END-PERIODIC
*
PERIODIC P180MIN
  INTERVAL 180
END-PERIODIC
*
Create a fast and a slow channel:

CHANNEL FAST

All collection actions via this channel reference the following destination:

DESTINATION LF01

PERFORM SEND-10
ACTION SEND-SCHEDULE
PERIODIC P10MIN
END-PERFORM

PERFORM RECV-LF01
ACTION RECEIVE
TIME 05:00
DAY SATURDAY
DAY TUESDAY
END-PERFORM

END-CHANNEL

CHANNEL SLOW

All documents transferred via this channel to Odex are given the priority 2:

PRIORITY 2

PERFORM SEND-180
ACTION SEND-SCHEDULE
PERIODIC P180MIN
END-PERFORM

END-CHANNEL

END-PROGRAM

**************************************************************
*......PROGRAM parameter group EXP: Connection to EXPEDITE *
*

PROGRAM EXP

LOGFILE_EXPIRATION 10
ARCHIVE ON
SAVE_RESPONSE_INFORMATION ON

April 2001
Configuration Examples

*       Definition of a 50 minute interval:
*
PERIODIC P50
   INTERVAL 50
END-PERIODIC
*
*       Definition of the SELECTION parameter group
*       CLASS-SAP to collect messages from
*       Information Exchange, whose CLASS attribute is
*       set to CLASS-SAP:
*
SELECTION CLASS-SAP
   RCV-CLASS RCVCLASS
END-SELECTION

TRACE ON
   TRACE ALL OFF
   FGM_FLOW ON
   DATA_FLOW ON
   FILES ON
   ROUTING ON
END-TRACE

CHANNEL   EXP-ACCT1
*
*       Set the IE CLASS attribute of all outgoing messages
*       sent via the channel EXPEDITE to SAP-EXP1:
*
*       Set the IE CLASS attribute of all outgoing messages
*       sent via the channel EXPEDITE to SAP-EXP1:
*
   PATH             d:/ie/acnt1
   SND-CLASS        SAP-EXP1
*
*       Send and receive documents on Fridays at
*       50 minute intervals:
*
PERFORM    SEND-RECEIVE
   ACTION      RECEIVE
   ACTION      SEND-SCHEDULE
   DAY     FRIDAY
   PERIODIC    P50
END-PERFORM
*
*       Collect daily at 10:00:
*
PERFORM    RECEIVE
   ACTION      RECEIVE
   TIME        10:00
END-PERFORM
*
*       Send daily at 20:00:
*
PERFORM    SEND
ACTION         SEND-SCHEDULE
TIME           20:00
END-PERFORM
END-CHANNEL

CHANNEL  EXP-ACNT2
*
*       Override the PROGRAM parameter PATH:
*
PATH          d:/ie/accnt2

PERFORM       RECEIVE10
ACTION        RECEIVE
ACTION        SEND-SCHEDULE
TIME           10:00
END-PERFORM

END-CHANNEL

END-PROGRAM
Configuration Example for TOPCALL

***************************************************************************************************
*      PROGRAM Parameter Group SCB
*      
*      PROGRAM SCB
*      Delete SCB log files after 10 days:
*      LOGFILE_EXPIRATION 10
*      
*      Routing parameter group:
*      ROUTING SAPCOMM
*      
*      SERVICE TLX
*      TO-CHANNEL CH-SAPSYS1
*      END-SERVICE
*      END-ROUTING
*      TRACE ON
*      PGM_FLOW ON
*      FILES ON
*      ROUTING ON
*      END-TRACE
*      END-PROGRAM

***************************************************************************************************
*      PROGRAM parameter group SAP: Connection to SAP Systems
*      
*      PROGRAM SAP
*      Delete SAP log files after 10 days:
*      LOGFILE_EXPIRATION 10
*      
*      All SAPcomm login activities for this instance
*      are stored in table SKWD under the following ID:
*      
*      IDSAPCOMM SAPCOMM1
*      
*      Normally rename and do not delete only
*      temporary files
*      (The system administrator is responsible
*      for reorganization):
*      
*      ARCHIVE ON
*      TRACE ON
*      PGM_FLOW ON
*      DATA_FLOW ON
*      CS_RESPONSE ON
*      FILES ON
*      ROUTING ON
*      SCI_ATTR ON
*      END-TRACE
*      

60 April 2001
Definition of the SAP destination.
Here, you determine to which SAP Systems
SAPcomm can build a connection:

DESTINATION SAPSYS1

The value of the following SYSTEM parameter
must point to a parameter group in the SIDEINFO
file, i.e. there must be an entry
DEST=SAPPROD:

SYSTEM     SAPPROD
CLIENT      000
USER        MYUID
PASSWORD    MYPASS
LANGUAGE    D
RELEASE     R3-21D
END-DESTINATION

SAP selection parameter groups
(download only).
Select all telecommunications documents:

SELECTION S TELE
APPLICATION TELE
END-SELECTION

Definition of PERIODIC parameter groups for
later reference in PERFORM parameter groups

PERIODIC P5MIN
INTERVAL 5
FROM     07:30
UNTIL    19:00
END-PERIODIC

PERIODIC P5MIN1
INTERVAL 5
FROM     07:32
UNTIL    19:00
END-PERIODIC

Channels of communications component SAP:

CpiC test channel:

CHANNEL   TEST1
DESTINATION SAPSYS1
TEST_CHANNEL ON

PERFORM  D_TEST
ACTION  RECEIVE
END-PERFORM
PERFORM  U_TEST
ACTION  SEND-SCHEDULE
END-PERFORM

END-CHANNEL
**Configuration Examples**

* Channel CH-SAPSYS1:

```
CHANNEL CH-SAPSYS1
```

* This channel is assigned to the SAP System (the above defined DESTINATION parameter group SAPSYS1).

```
DESTINATION SAPSYS1
PERFORM SEND
  ACTION SEND-SCHEDULE
  PERIODIC P5MIN
END-PERFORM
```

```
PERFORM RCV-ALL
```

* Download:

```
ACTION RECEIVE
```

* Collect all documents defined by the SELECTION parameter group:

```
SELECTION S_TELE
  PERIODIC P5MIN1
END-PERFORM
```

```
END-CHANNEL
END-PROGRAM
```

***************************************************************

* .......PROGRAM parameter group TOP: Connection to TOPCALL

```
PROGRAM TOP
```

* UNIX port for the V24 connection to Topcall:

```
PORT /dev/tty00
```

```
PERIODIC TCTIME
  INTERVAL 10
END-PERIODIC
```

* Channel TF01: All outgoing telecommunications documents assigned (in R/3) to the destination TF01 - and only these - are transferred via this channel to TOPCALL

* Generally, for each destination defined in R/3 (Table 164c) a SAPcomm channel of the same name must be created.

```
CHANNEL TF01
  PERFORM PUT-FAXES
  ACTION SEND-SCHEDULE
  PERIODIC TCTIME
  END-PERFORM
END-CHANNEL
```

```
TRACE ON
INIT ON
```
DATA_FLOW ON
CS_RESPONSE ON
FILES ON
ROUTING ON
SCI_PARAM ON
SCI_ATTR ON
END_TRACE

END_PROGRAM

*******************************************************************************
Configuration in the SAP System

**Overview** [Seite 65]

**Activating Logging (Table T164A/B)** [Seite 66]

**Checking Logon Times (Table TSKWD)** [Seite 67]

**Configuring Telecommunications Applications** [Seite 68]
Overview

This section shows you how to configure SAPcomm in the SAP System (R/2 or R/3). It also shows you how to test the configuration in the R/3 System.

A number of tables are important here, which effect all or some of the applications:

- All applications (EDI, Mail and telecommunications):
  
  Tables T164B and TSKWD

- Telecommunications applications:
  
  T164A, T164B, T1640, T164P, T164U, TMLVW, T164C and T164Y
Activating Logging (Table T164A/B)

In R/3: You maintain table T164A.
From Release 5.0C of R/2: You maintain table T164B.

In this table you can activate the Debugging parameter. This parameter lets you control whether the CPI-C data exchange between SAPcomm and the SAP system is logged. Please note that you should only activate logging for analysis purposes.

In this case, only information on the newest CPI-C link between SAPcomm and the SAP System is written to table TEDCT.
Checking Logon Times (Table TSKWD)

Table TSKWD exists in R/2 as of Release 5.0C, and in R/3 as of Release 1.1.
It maintains a separate record for different functions (EDI, mail, and so on) of when SAPcomm last logged on to the SAP system.
If you know the times, at which SAPcomm should log onto the SAP system (in accordance with the SAPcomm time plan in the configuration file), you can use table SKWD to check whether SAPcomm is keeping to its schedule.
You distinguish between different SAPcomm installations via the SAPcomm ID (see Overview [Seite 90]).
Configuring Telecommunications Applications

The behavior of the SAP system when dispatching and receiving telecommunication messages (telefax, telex, teletex) is controlled by a group of tables (164A, 164B, 164C, 164O, 164P, 164U, 164Y and MLVW).

Menu functions are available in R/3 to maintain these tables. These functions are in the menu Configuration, which you can call via transaction code SCOM or the following path:

Administration → Communication → SAPcomm

The configuration options can be divided into 2 categories:

- **Sending:**
  - Automatic selection of logical destination (and an SAP communications server) for the dispatch of a message and the automatic modification of the recipient's number.

- **Receiving:**
  - Automatic assignment of an incoming message to a user in the SAP system.
  - R/3 menu options: System parameters, Distribution list.

These R/3 menu options and the tables which form their basis (the parameters of which are valid for both R/3 and R/2) are described in detail in the sections which follow.

**System Parameters (T164A and T164B)**

In Tables T164A and T164B (R/2 and R/3-Releases 2.x only), you can make basic settings for sending and receiving messages:

- retention period of the messages in the spool
- format check of recipient's numbers (R/2 and R/3-Releases 2.x only)
- check code page (R/2 and R/3-Releases 2.x only)
- type of dispatch status report (R/3 only)
- process incoming messages (R/2 only)
- search area in incoming messages
- substitute recipient for incoming messages
- Test aid
- Spool formatting in the SAP System (R/3-Release 3.0 only)

**Server Location (T164O)**

Each logical destination, that you want to use for the dispatch of telecommunications messages must be defined here. A channel with the same name must also be created on an SAP communications server for every destination. For more details, refer to "How are Documents Allocated to "Their" Channel? [Seite 46]."

Enter the country ID of the location of the respective SAP communications server for every destination. This is used for automatic country code generation.

You can provide a comment for every destination.

**Server Selection (T164P)**

This table determines which destination is to be selected for message dispatch. The selection depends on:

- Telecommunications service (e.g. telefax; In R/3: TELEFAX, in R/2: TF)
•  Document type (currently not supported, always leave the field blank, R/3 only)
•  Country of the recipient (e.g. GB, R/3 only)
•  Area code of the recipient’s number (e.g. 071)
•  Communication device group (user group-specific, defined by parameter CDG, R/3 Release 3.0 only)

When dispatching a message, the entries are processed top down and the specified destination is used for the first matching entry.

The sort ID is used to sort the entries into the order you want.

**Server Exceptions (T164U)**

Special cases for automatically adapting the connection can be stored in this table. This allows a local area code to be removed if, for example, it is the same as the local area code in which the SAP communications server is situated, or a special cost-saving abbreviated dialing code is used in local areas.

Depending on the destination and the telecommunications service, the following are specified:

•  Initial digits of a recipient's number to be replaced
•  A series of digits to replace the deleted digits (it can also be empty).

When dispatching a message, the entries are processed top down and the initial digits of a recipient's number are replaced at the first match.

The field "Key" is a sort indicator and is used to sort the entries into the order you want.

**Country Code (TMLVW, T005K/T005I/T005J)**

In R/2 and in R/3 Releases 2.x, the country codes for telefax, telex and teletex are contained in table TMLVW. From Release 3.0 of R/3, these dialing codes are contained in tables T005K/T005I/T005J.

You carry out maintenance as follows: **Administration ➔ Communication ➔ SAPcomm, Configuration ➔ Country code or - from 3.0 also - Country code exceptions tel.**

All combinations derived from the location countries of SAP communications servers and all the countries of the possible message recipients must be entered here.

When dispatching a message, the country code found under the respective sending and target countries - depending on the service - is set before the recipient's user number. The telefax service contains a special feature: an area code for the target country can be entered in field "1". If the recipient's number starts with this number, it is removed when inserting the country code. (for example: "0" in Germany, "9" in Spain ... )

**Assignment (T164C, R/3 only)**

Output devices (destinations) can be assigned to SAP communications servers in this table. On each server, you can make the following specifications in the SAPcomm configuration:

•  Server indicator (Parameter IDSAPCOMM)
•  Selection of DESTINATION (including generic)

The assignment of destinations to a SAPcomm server is as follows:

•  If you specify a DESTINATION, the server only receives messages for this destination.
•  If a DESTINATION is not selected, the server receives messages for all destinations entered for its indicator (IDSAPCOMM) in table T164C.
•  If this server indicator is not listed in T164C, the server receives messages for all destinations beginning with T.
All destinations assigned to a SAPcomm server must also be created as the CHANNEL of a fax/telex component in the configuration of this server.

**Distribution List (T164Y)**

This controls the distribution of incoming telecommunication messages for users in the SAP system.

Every entry defines a search term. The beginning of every incoming message is scanned for these terms. The first and final line of this initial area are set in the system parameters. The actions defined in this entry are carried out when the first match is made:

- Printer, on which the message is to be output. The message is then printed automatically.
- The name of a SAP user or list of recipients, to whom the message is to be sent by Mail.
- Recipient type (R/3 Release 3.0 only)
- Specification as to whether the message is to be sent as an express mail.

You can sort the entries into the order you want via a sort indicator.

**Special Features in R/3**

Faxes/telex dispatch from R/3 entails printing the message on special devices.

For this, you must perform the following configurations:

1. Create SAP spool output device
   
   Output devices are set up for telefax or telex using SAP Spool Administration.

2. Set SAPscript printer driver
   
   SAP applications (MM, SD...) and SAPoffice print to these fax/telex output devices via SAPscript. Therefore, these devices must also be configured in SAPscript.

3. Maintain telecommunications tables
   
   Unlike normal printing to paper, the application does not directly specify the required output device for fax/telex dispatch; It only determines the device type i.e. TELEFAX or TELEX.

   A suitable output device is automatically determined via special tables. These tables must be maintained.

   Additional tables are required to modify the fax/telex number according to the local details of the SAPcomm location (country code, area code etc.)

   For this, use transaction code SCOM, or choose

   Administration→Communication→SAPcomm

   Choose the menu options of the Configuration menu to maintain the various tables.

   A description of these tables is given above.

**Create SAP Spool Output Device**

To determine a spool output device, use transaction code SPAD.

Please note the following:

- Choose any name you like for the output device, e.g. TF01.

If the name does not begin with the letter T (=telecommunications), or if there are other output devices with begin with T and are not fax/telex output devices, you must also enter the name of this output device in Table T164C.
Configuring Telecommunications Applications

- **Device type**: It must correspond to the communications subsystem used with SAPcomm. If, in Table TSP09, you marked the field ABAP/4 for the driver TELE (see the following section “Set SAPscript Printer Driver”), you must always state TOPCALL for the device type.

- **Formatting host**: Any, it is not analyzed, e.g. host on which SAPcomm is installed.

- **Host printer**: Any, it is not analyzed, e.g. SAPcomm or TOPCALL.

- **Device class**: F for telefax, T for telex.

- **Access method**: X (=SAPcomm)

- **LPQ format**: Any, it is not analyzed, e.g. B

- **Conversion flag**: Select NO

- **Location and message**: Any entries

- **SAP title page, SAP lock printer, operating system prompt**: Select NO.

After pressing enter, an additional field appears:

- **Destination host**: Any, it is not analyzed, e.g. host on which SAPcomm is installed.

### Set SAPscript Printer Driver

To set the SAPscript printer driver, maintain Table TSP09.

For this, use Transaction code SM31 or choose

**System → Services → Table Maintenance.**

In R/3 Releases 2.x, there must be a X for the entry TELE in the **Driver** column in the **ABAP/4** column.

In R/3 Release 3.0, there should be an empty field (‘ ’, SPACE) for the entry TELE in the **Driver** column in the **ABAP/4** column. This is the prerequisite for format processing of fax documents defined by the device type. You should mark the **ABAP/4** field for the driver TELE only if minimum formatting (end of line, line feed) suffices for your purposes. The device type must then be TOPCALL (or a defined type with another name).

### Testing the Telecommunications Configuration in R/3

To test the configuration, proceed as follows:

1. Check the consistency of the table entries:

   Choose **Administration → Communication → SAPcomm, Utilities → Check.**

2. Test the device assignment and number modification:

   For this, choose transaction code SKRT or the menu paths

   **Administration → Communication → SAPcomm, Routing → Test.**

   After you specify a recipient’s number and a country ID, an output device (destination) must be found, and the number to be dialled must be modified accordingly.

3. Create a simple message:

   Choose **Administration → Communication → SAPcomm, Utilities → Send test fax.**

4. Create a test message

   - Choose **Office → Outbox, Document → Create**

   - Create a type RAW document and save it.

   - Specify target country and recipient number.

   - Choose **Mail → Send, Document → Send direct ext.**

   → Fax.... Fax the message.
There must now be a new entry in the spool database. To check it, choose transaction code SP01 or

System → Services → Print Requests

The output status of the new entry is Waiting.
The Control Component SCB

Overview [Seite 74]

SCB-specific Entries in the SAPcomm Configuration File [Seite 75]

PERFORM Parameter Group [Seite 77]

The Routing Parameter Group [Seite 78]

Operating the Communications Server [Seite 84]

Notes on Testing and Problem Analysis [Seite 87]
Overview

The control component is the central component of the SAP communications server:

- It is the first component activated after the start of the communications server.
- It handles the starting of the individual communication components.
- It carries out the orderly system shutdown.
SCB-specific Entries in the SAPcomm Configuration File

The identifier of the control component is SCB. You configure the control component in the same way as the other components of the communications server (the communications components). You therefore insert (just) one SCB parameter group into the configuration file SAPCOMM.CFG.

This parameter group has the following structure:

```
PROGRAM SCB

  IGNORE_CC_ABORT <ON|OFF>
  IGNORE_TIME_PLAN <ON|OFF>
  OPEN_LOG_WINDOW <ON|OFF>

  PERFORM <Program>
  TIME 10:00
  TIME 20:00
  ACTION /usr/sap/com/start.scqueue

END-PERFORM

ROUTING SAPCOMM
  SERVICE <Name>
  <SERVICE-Parameter>
END-SERVICE

END-ROUTING

TRACE <ON|OFF>
  <TRACE-Parameter>
END-TRACE

END-PROGRAM
```

The individual parameters in sub-groups of this group are explained below:

**IGNORE_CC_ABORT**
Optional. If this parameter is entered (with ON), then the termination of an activated communications component does not automatically result in the termination of all the activities of the communications server. This parameter is only used for problem analysis and should not be entered into normal operation.

**IGNORE_TIME_PLAN**
Optional. If this parameter is entered (with ON), all entries in the SAPcomm time plan are ignored. The communications server then only carries out actions based on the user input via the user interface. This parameter is only used for testing purposes and must not be activated in productive operation.

**OPEN_LOG_WINDOW**
Only OS/2: If this parameter is entered and you start the communications server via the user interface, then the log window is opened automatically at the start of the communications server.

**PERFORM**
The PERFORM parameters are described in [PERFORM Parameter Group](Seite 77).

**ROUTING**
The ROUTING parameters are described in [The Routing Parameter Group](Seite 78).

**TRACE**
The TRACE parameters are described in [TRACE Parameter Group](Seite 42).
You can instruct the control component to start any programs controlled by the time plan. For this, you create one or more PERFORM parameter groups in the SCB program parameter group, in which you specify the names of the programs to be executed as ACTION parameters. The program is executed asynchronously by the control component. Program start is entered in the SCB log file, however program output and program termination code are not.

If the program to be called generates screen output, and you want to store this output in a file, start the program indirectly: The control component starts a shell script, which in turn starts the program to be called, and in the course of this redirects the program output to a file (see example below).

Note that the program to be executed must not require any user input.

You would like to print out the number of files waiting for processing within SAPcomm twice a day, and store the printout in the file /usr/sap/com/load.

First, you create the shell script /usr/sap/com/start.scmqueue. It contains, for example, the following two lines (HP-UX):

date        >>/usr/sap/com/load
scmqueue -l >>/usr/sap/com/load

Add the following parameter group to parameter group PROGRAM SCB in the SAPcomm configuration file sapcomm.cfg:

PERFORM START SCMQUEUE
TIME    10:00
TIME    20:00
ACTION  /usr/sap/com/start.scmqueue
END PERFORM

Note

OS/2:
As the program specified as the ACTION parameter is executed from a separate command interpreter (cmd.exe), SAPcomm cannot recognize whether it could even be started. Even in this case, no error message in entered in the SCB log file.
The Routing Parameter Group

The assignment of an incoming or outgoing document to a channel is made in the ROUTING parameter group in the PROGRAM SCB group of the configuration file SAPCOMM.CFG (exceptions: SAP EDI and SAP telecommunications outgoing documents).

In addition to the fundamental purpose of channel assignment, the ROUTING parameter group is also used to determine the category of incoming documents (MAIL or EDI) for which no type has been determined a priori (such as documents received via X400).

The problem of assigning a document to its responsible channel - the routing - has already been discussed in the topic How are Documents Allocated to “Their” Channel? [Seite 46] in the section on the configuration file. This topic describes the details of the routing statements that have to be maintained in the configuration file.

Example of a ROUTING Parameter Group

The following example illustrates the principle of default channel assignment and specialization (override). For an explanation of the APPLICATION parameter, refer to “Classification Groups” (SERVICE X400) under the section “SERVICE Parameter Groups” below.

**ROUTING SAPCOMM**

SERVICE X400

ORGANIZATION SALES

OU1 EDI

TO-CHANNEL SAP1

APPLICATION EDI

END-OU1

TO-CHANNEL SAP1

APPLICATION MAIL

END-ORGANIZATION

OU1 EDI

TO-CHANNEL SAP-EDI

APPLICATION EDI

END-OU1

TO-CHANNEL SAP2

APPLICATION MAIL

END-SERVICE

SERVICE OFTP

TO-CHANNEL SAP-EDI

END-SERVICE

END-ROUTING

The following assignments are defined for X400 and OFTP documents by these routing statements.

X400 documents are assigned to different channels:

- All X400 documents which have the component "Organization" set to "SALES" in the recipient address are allocated as follows:
  - If their "Organizational Unit 1" (OU1) is set to "EDI", the application EDI is allocated to them and they go to channel SAP1.
Otherwise, they are allocated the application MAIL (and they are forwarded to the same channel "SAP1").

It is not necessary to start component specification with the component with the highest rank (COUNTRY) (followed by the next lower, and so on; if the components of a higher rank are missing, then they are taken to be matches during allocation.)

- All other X400 documents, in which the component "Organizational Unit 1" (OU1) is set to "EDI" in the recipient address, are allocated to the channel "SAP-EDI".
- All other X400 documents are allocated to the channel "SAP2".

All incoming documents (OFTP documents) received via the communications component ODX are allocated to the channel "SAP-EDI".

**SERVICE Parameter Groups**

The ROUTING parameter group consists of one or more SERVICE parameter groups. Each of these parameter groups contains the routing statements that refer to “their” service.

This means that you, for example to forward incoming ODEX messages to one or more SAP systems, must set up a parameter group SERVICE OTP. The parameter group SERVICE SAP is not responsible for these messages.

Valid SERVICE parameter groups and their application fields are:

- **SERVICE X400** documents with recipient addresses in X400 format (communications components X40, SSW...).
- **SERVICE SNADS** documents with recipient addresses in SNADS format.
- **SERVICE SAP** documents with recipient addresses in the SAP format (for example, mail messages between SAP systems).
- **SERVICE OFTP** documents sent or received via the ODX communications component.
- **SERVICE EXPEDITE** documents sent or received via IBM Expedite.
- **SERVICE BCS** documents received via the BCS communications component.
- **SERVICE FAX** fax documents.
- **SERVICE TLX** telex documents.
- **SERVICE TTX** teletex documents.

Every SERVICE parameter group consists of the following elements:

- Optional parameter groups (classification groups):

  They define subsets of addresses for all messages belonging to the current SERVICE and allocate channels to these subsets for further processing. These parameter groups are called classification groups.

  The classification groups have SERVICE-dependent types (that is: address components - COUNTRY, ADMID, PRMD, ORGANIZATION, and so on for SERVICE X400) and can also contain sub-classification groups themselves.

- Reference to a default channel. All messages belonging to the current SERVICE that are not entered by one of the above parameter groups are forwarded to this channel.
The Routing Parameter Group

Example of a classification group:

<table>
<thead>
<tr>
<th>COUNTRY</th>
<th>DE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADMD</td>
<td>DBP</td>
</tr>
<tr>
<td>TO-CHANNEL</td>
<td>CHAN1</td>
</tr>
<tr>
<td>APPLICATION</td>
<td>MAIL</td>
</tr>
<tr>
<td>END-ADMD</td>
<td></td>
</tr>
<tr>
<td>TO-CHANNEL</td>
<td>CHAN2</td>
</tr>
<tr>
<td>APPLICATION</td>
<td>MAIL</td>
</tr>
<tr>
<td>END-COUNTRY</td>
<td></td>
</tr>
</tbody>
</table>

You must specify a default channel.

You will find an explanation of the APPLICATION parameter in the section on the X400 SERVICE Group below.

The structure of the SERVICE parameter group, and the classification groups contained in it, is hierarchical in the following sense:

- Documents of a common SERVICE are grouped together initially and allocated to a (default) channel.

- This assignment can be overridden, by grouping together documents with a common "next lowest" address component (for example: SERVICE -> COUNTRY -> ADMD and so on for X400) and allocating them to a (possibly different) channel.

- A more specific channel assignment can be achieved by repeated application of this process (that is, the definition of parallel or deeply nested subparameter groups).

The hierarchy between parallel subparameter groups (that is, those which have a common "main group") is implemented using ranking (predefined in SAPcomm):

The group ADMD, for example (for SERVICE X400), has a "higher" rank than the group PRMD. As a consequence, the relationship of the document to one of the ADMD groups is checked first when allocating a given document to a channel. If the document can be allocated to an ADMD group, then all routing information entered outside of this ADMD group is irrelevant for this document. Only if no assignment to an ADMD group is possible, is a check made whether the document can be allocated to a "lower" group - for example, a PRMD group.

Classification Groups

The categories of the respective classification groups are entered for every SERVICE parameter group in the following section.

The sequence of the list reflects its hierarchical order: the category ADMD, for example, (for SERVICE X400) is entered BEFORE the category PRMD because it has a higher rank (see above).

Please remember that parameter groups for the entered categories are optional and - if they are entered at all - can also occur in multiple instances (parameter groups ADMD A1, ADMD A2...).

SERVICE X400

X400 documents are classified according to the components of their recipient address. In addition to a channel for further processing, X400 documents must also be allocated to an application field (application). You must therefore also include the parameter APPLICATION with the values EDI or MAIL at every place where you enter the parameter TO-CHANNEL (this, of course, has the result that a recipient cannot be an EDI and MAIL recipient at the same time - but in practice this is not an important restriction).
The categories of the classification groups are:
COUNTRY - ADMD - PRMD - ORGANIZATION - OU1 - OU2 - OU3 - OU4 - LASTNAME - FIRSTNAME.

SERVICE SNADS
SNADS documents are classified according to the components of their recipient address. In addition to a channel for further processing, SNADS documents must also be allocated to an application field (Application). Please read the topic on SERVICE X400 for more details on this topic.

The categories of the classification groups are:
DGN (distribution group name) - DEN (distribution element name).

SERVICE SAP
(As of R/2 Release 5.0E or R/3 Release 2.1): Documents in which the recipient address is entered in the SAP format are classified as follows:

- LU name of the target system.
  The LU name is a field of a line in Table XCOM, which is referenced by the “Symbolic Destination” field of the external address.
- Client
- User name
  These are fields that you maintain when creating an external address in the SAP R-Mail (or Office) System.

The categories of the classification groups are: LU, CLIENT und USER.

SERVICE OFTP
OFTP documents are routed depending on the local code of the recipient configured in ODEXPC, and not on the EDI code of the recipient.

Example:
```
SERVICE OFTP
  RECEIVER-LC  SAP1
    TO-CHANNEL  CHAN1
  END-RECEIVER-LC
    TO-CHANNEL  CHAN2
  END-SERVICE
```
Explanation: All incoming OFTP messages, which are addressed to the recipient with the ODEX local code SAP1, are routed to channel CHAN1. All other OFTP incoming documents are routed to channel CHAN2.

Classification group type:
RECEIVER-LC.

SERVICE EXPEDITE
EXPEDITE documents are classified by the recipient specified in the document (Example for EDIFACT: UNB segment), and by their IBM-IE class.

In the SERVICE parameter groups, the recipient is either specified as parameter ALIAS or as a parameter pair ACCOUNT, USERID:

- ALIAS:
  For documents whose (recipient) EDI codes do not contain blanks or periods;
  ACCOUNT, USERID
  For documents whose (recipient) EDI codes contain a blank or a period.

In this case, the
The Routing Parameter Group

first part of the EDI code is referenced with parameter ACCOUNT, the second part with parameter USERID.

CLASS
Value of the (IE)CLASS attribute of the message
The classification groups are of the type ALIAS, ACCOUNT, USERID and CLASS.

Examples:
1. All incoming IE message should be routed to the channel CH-ALL:

   SERVICE EXPEDIT
   TO-CHANNEL    CH-ALL
   END-SERVICE

2. All incoming IE messages, whose (recipient) EDI codes consist of a pair "<ACCOUNT> <USERID>", with <ACCOUNT>="MYACCNT", should be routed to channel CH-MYACCNT, all other incoming IE messages are routed to channel CH-DEFAULT:

   SERVICE EXPEDIT
   ACCOUNT MYACCNT
   TO-CHANNEL    CH-MYACCNT
   END-ACCOUNT
   TO-CHANNEL    CH-DEFAULT
   END-SERVICE

3. All incoming IE messages, whose (recipient) EDI codes consist of a pair "<ACCOUNT> <USERID>", with <ACCOUNT>="MYACCNT", should be routed as follows:
   - If their CLASS attribute is set to MYCLASS, they are routed to the channel CH-MYCLASS,
   - Otherwise, they should be routed to the channel CH-MYACCNT.

   All other incoming IE messages should be routed to the channel CH-DEFAULT:

   SERVICE EXPEDIT
   ACCOUNT MYACCNT
   CLASS MYCLASS
   TO-CHANNEL    CH-MYCLASS
   END-CLASS
   TO-CHANNEL    CH-MYACCNT
   END-ACCOUNT
   TO-CHANNEL    CH-DEFAULT
   END-SERVICE

   Note: you can also use, for example, USERID instead of CLASS.

4. All incoming IE messages, whose (recipient) EDI codes consist of a pair "<ACCOUNT> <USERID>", with <ACCOUNT>="MYACCNT", should be routed to the channel CH-MYACCNT.

   All incoming IE messages, whose (recipient) EDI codes are set to "ALIASNAME" (and contain no blanks), should be routed to channel CH-ALIAS.

   All other incoming IE messages should be routed to the channel CH-DEFAULT:

   SERVICE EXPEDIT
   ACCOUNT MYACCNT
   ALIAS ALIASNAME
   TO-CHANNEL    CH-ALIAS
   END-ALIAS
TO-CHANNEL          CH-DEFAULT
END-SERVICE

The sender of the message must include the recipient information as a value of the IE attribute RECEIVER. Check this in the files created by Expedite BASEOUT.MSG. Your Expedite/OS2 installation must be at least version 4.1.

If these requirements are not fulfilled, use only the parameter CLASS.

SERVICE FAX
FAX documents are classified by their fax number. Only the routing statements for incoming documents must be maintained for FAX.
The classification groups are of the category: NUMBER.

SERVICE TLX
TLX documents are classified according to their telex number. Please remember that only the routing statements for incoming documents have to be maintained for TLX.
The classification groups are of the category: NUMBER.

SERVICE TTX
TTX documents are classified according to their teletex number. Please remember that only the routing statements of incoming documents have to be maintained for TTX.
The classification groups are of the category: NUMBER.

SERVICE BCS
BCS documents are classified by the parameter values for BANK and USER-ID.
Operating the Communications Server

The communications server is designed for unattended 24-hour operation. Most of the files it creates are also deleted again automatically. The auxiliary program `scmreorg` is provided to delete or save the remaining files ("archived" transmission files etc., see The SAPcomm Auxiliary Programs [Seite 232]. No further maintenance operations are necessary.

Additional topics:
- Starting the Communications Server [Seite 85]
- Closing the Communications Server [Seite 86]
Starting the Communications Server

You can operate the SAP communications server with or without the graphical interface.

You start the control component directly to operate the communications server without the graphical interface. To do this, you enter the following at the command level of your operating system:

```
scmbasis
```

You can also activate this program using alternative mechanisms, such as automatic start during system start.

To operate the communications server with the graphical interface, you start the program responsible for creating the SAPcomm window. To do this, you enter the following at the command level of your operating system:

```
sapcomm
```

The program `sapcomm` sets up the SAPcomm window and starts the control component (program `scmbasis`). For details on the program `sapcomm`, refer to Working With the Graphical Interface [Seite 220].

After Starting

After starting the program, the control component of the communications server is activated first. It carries out the following steps:

- loads the binary configuration data (file sapcomm.bcf)
- starts the communications components entered in the configuration file

Note: The file `sapcomm.bcf` is automatically created from the source file of the configuration data (`sapcomm.cfg`). If, however, error messages appear when starting the communications server (“Binary configuration file can’t be generated” or “Binary configuration data invalid”), use the auxiliary program `scmcfgcc` to analyze these problems. Please read the short description of this program in The SAPcomm Auxiliary Programs [Seite 232].

Further program execution is determined by the time plan statements that you entered in the configuration file `sapcomm.cfg`. 


Closing the Communications Server

You close the communications server in one of the following ways:

- From the graphical interface
- By calling the auxiliary program `scmstop` (see The SAPcomm Auxiliary Programs [Seite 232])

For an orderly termination, the control component must wait for the activities of every individual communications component to be terminated.

This can lead to a long wait before shutdown, particularly if synchronous data transmission is in progress - for example, in the case of the communications components SAP or EXP. In this case, simply wait. The shutdown process runs automatically once initiated. The shutdown process cannot be cancelled! After it has been initiated, no new activities can be started - neither by the time plan nor by user input.

However, it is possible to end the activity of the communications server "uncleanly" (from within the graphical interface or via the help program `scmstop`), regardless of whether shutdown was initiated beforehand or not. In this case, all communications components are ended "compulsorily", without regard for activities that may currently be in process. Only use this method in emergencies - data loss cannot be excluded as a result!
Notes on Testing and Problem Analysis

A separate log file is created in each component of the communications server.

When analyzing errors or problems, always consult the log files of the communications
components concerned. You will find details on log files under "The Log Directory" in Checking
the Configuration Data (scmcfgcc). Also consult the log files of the communications programs
you are using.

You can determine the scope of the log entries via the TRACE parameter group TRACE
Parameter Group [Seite 42] in the configuration file. For details on the TRACE parameter group,
refer to the section "The PROGRAM Parameter Group" in the Chapter "The Configuration File
SAPCOMM.CFG".

The error messages in the SAPcomm log files generally contain key words which, if required,
assist you when consulting the respective guides (for your operating system, communications
program, and so on).

Please also take note of the comments in the sections on the individual communications
components regarding targeted problem analysis.

When checking log entries, ensure that you take note of the following points:

• Which files have been created?
  (Switch on TRACE parameter FILES)
• Which files have been passed on to the communications component or from this to the
  communications program?
• Which files have been exchanged between SAPcomm and the SAP System?

You should archive certain files for later analysis (e.g. response files of the connected
communications programs). For this, use the configuration parameter
SAVE_RESPONSE_INFORMATION and ARCHIVE.

Error Entries in the SAPcomm Log Files

Entries in the SAPcomm log files are divided into the following categories:

I  Information
W  Warning
E  Error

Most of the entries are self-explanatory when considered together with the specific error
situation.

The following section gives an explanation of some of the log entries in the log files. An
explanation of log entries specific to a certain communications component can be found in the
sections on the individual communications components.

E0000 Error opening <file1>

File1 <file1> could not be opened.

If this file was to be opened for writing, the error entry may indicate an authorization
problem (UNIX). Check whether the user, under which you started the communications
server, has write authorization in the directory in which the file was to be created.

If the file was to be opened for reading, then it can also - as above - be an authorization
problem. Furthermore, the error could also be traced back to an earlier file
reorganization (renaming or deleting files).

E0001 Error renaming <file1> to <file2>

April 2001  87
Notes on Testing and Problem Analysis

File <file 1> could not be renamed as <file2>.
Check whether file <file2> already exists (one reason for this could be an earlier abnormal termination of the communications server, the communications component or the entire workstation). If this is the case, then rename this file.
If the file contains data to be sent, first check whether this file has already been sent by your communications program. If this is not the case, initiate sending manually.

**E0002 Error running program <program>**
Program <program> could not be started.
Check whether the program is installed on the hard disk of your computer, and whether the environment variable PATH points to the directory in which the program is installed.

**E0130 Cannot route <Recipient-Info>**
No channel could be allocated to the document in order to process it further.
Please check your entries in the ROUTING parameter group. For more details, refer to The Routing Parameter Group [Seite 78].

**Binary configuration file can't be generated**
The configuration file sapcomm.cfg contains invalid entries.
Correct this file using the auxiliary program scmcfgcc (see The SAPcomm Auxiliary Programs [Seite 232]) and the notes in “The Configuration File SAPCOMM.CFG”.

**Binary configuration data invalid**
The binary configuration data in the file SAPCOMM.BCF could not be loaded.
Recreate this file using the SAPcomm auxiliary program scmcfgcc (see The SAPcomm Auxiliary Programs [Seite 232]). After you have located and corrected the problem, restart the communications server.
Connection to SAP Systems

Overview [Seite 90]

Hardware and Software Requirements (SAP Systems) [Seite 91]

The Communications Component SAP [Seite 92]

SAP-specific Entries in the SAPcomm Configuration File [Seite 93]

Character Sets and Code Pages [Seite 101]

Testing the Connection [Seite 102]

Notes on Problem Analysis [Seite 105]
Overview

The communications server is connected to SAP Systems via CPI-C links. You can configure it to communicate with several R/2 and/or R/3 Systems and enable (EDI or Mail) data to be exchanged between these systems.

You can check which versions of the R/2 or R/3 System can be connected to SAPcomm under the section “Basics” in the topic Compatibility [Seite 12].

The communications server can run either together with an R/3 database server, application server or presentation server on the same piece of hardware or apart from all linked SAP Systems on a separate piece of hardware.
Hardware and Software Requirements (SAP Systems)

Communication between the communications server and an SAP System is based on CPI-C. The particular protocol (TCP/IP...) used as a basis depends on your network environment and is not important as far as configuring the SAP communications server is concerned. This document does not cover technical network aspects of the link between the communications server and SAP Systems, for example:

- Which network components (VTAM, Communication Manager, and so on) are required or are supported,
- Configuration of network components,
- Configuration of gateways,

You will find information on this in the the following documentation:

- SAP Communication: Configuration
- SAP-supported Network Products (SAP Brochure)
- Documentation on your network software
The Communications Component SAP

The connection of the communications server to SAP Systems is handled by the communications components SAP.

EDI Incoming Documents

When dealing with incoming EDI documents, certain information required to assign the incoming document to an SAP partner number must always be transferred to the SAP system, and which is maintained partner-specifically in the SAP system as a "communications service" (for incoming documents).

The "communications service" transferred to the SAP system (and which must be maintained there appropriately) depends on the respective communications component via which the incoming document reached the communications server.

Please take note of the comments on the incoming EDI documents and communications service made in the respective sections on the communications components.
SAP-specific Entries in the SAPcomm Configuration File

SAP is the identifier of the communications component responsible for the connection to SAP Systems. You must include the following parameter set in the configuration file SAPCOMM.CFG:

```
PROGRAM SAP
    <Parameter specification>
END-PROGRAM
```

The parameters of this group are explained below. A general description of the PROGRAM parameter group can be found in Structure of the Configuration File [Seite 37].

An example of a PROGRAM parameter group for the component SAP can be found in Model Configuration [Seite 218].

Component-specific PROGRAM Parameters

The component-specific PROGRAM parameters are listed below.

**IDSAPCOMM <SAPcomm identifier>**

Optional. Enter an identifier of up to ten alphanumeric characters. This identifier is passed on to the SAP System by the communications server. Several instances of SAP communications servers in SAP Table SKWD are distinguished by means of the identifier configured here.

**R/2:**

EDI outgoing messages (from Release 5.0F):

The SAPcomm identifier must be configured as the "port" in the R/2 System.

**SM_LOG <Level>**

Optional. You can use this parameter to activate a special trace function in the SAP communications component. This trace function is used for internal SAP problem analysis. You should only use this parameter after consulting your SAP support representative.

Trace levels:

- 0: no trace
- 1: errors only (default)
- 2: internal status change
- 3: maximum trace

This function is only implemented for the SAP communications server under UNIX. When using the SAP communications server under OS/2, the parameter SM_LOG has no effect, message 1008 in the SAPcomm log indicating this.

**CONVERT <conversion file>**

Optional. Only for the connection to R/2 Systems. If you specify this parameter, the specified conversion file is used for data exchange with all R/2 Systems. Exception: Data exchange with the R/2 Systems, to which you have assigned a DESTINATION-specific conversion file with parameter DCONVERT (see above).

For more information, read the topic Character Sets and Code Pages [Seite 101].

**SIDE_INFO <path and file name>**

The communications server references an entry in the sideinfo file via the parameter SYSTEM when establishing the link with an SAP System. If you want to override the
SAP-specific Entries in the SAPcomm Configuration File

search path for this file, then enter the complete path of the desired sideinfo file here (with file names). For more information on the side info files, read the following guide:
SAP Communication: Configuration

The path specification of the sideinfo file via the parameter SIDE_INFO is only implemented for the SAP communications server under UNIX. When using the SAP communications server under OS/2, this parameter has no effect. The sideinfo file to be used should be defined via the environment variable SIDE_INFO in this case. This variable is set automatically by the installation.

**RCV_TIMEOUT <number of seconds >**

By setting the RCV_TIMEOUT parameter and/or the CRCV_TIMEOUT CHANNEL parameter, you can instruct SAPcomm to automatically cancel “hanging” CPI-C connections.

If you do not set this parameter, the “hanging” CPI-C connection causes the associated SAPcomm component to be blocked for other tasks.

If you want to have different timeouts for different channels, you can set the CHANNEL parameter CRCV_TIMEOUT in addition to RCV_TIMEOUT.

This overrides the timeout set by the PROGRAM parameter RCV_TIMEOUT for the appropriate channel. However the CRCV_TIMEOUT parameter only takes effect AFTER SAP System logon.

The timeout for CPI-C connection setup is set exclusively by the PROGRAM parameter RCV_TIMEOUT.

Configuration hint:

Use the PROGRAM parameter RCV_TIMEOUT to configure a relatively short timeout (around 15 seconds) mainly for the logon to SAP Systems. Use the CHANNEL parameter CRCV_TIMEOUT, on the other hand, to configure longer timeouts adjusted to the respective SAP System and respective application (Mail, EDI or TELE). If you do not want automatic cancellation of a connection after logon, simply set the the CHANNEL parameter to a very high value (36000 = 10 hours).

The definition of the timeout times via the parameters RCV_TIMEOUT and CRCV_TIMEOUT is only implemented for the SAP communications server under UNIX. When using the SAP communications server under OS/2, these parameters have no effect. In this case, a general timeout should be set via the environment variable CPIC_TIMEOUT.

**PERFORM Statement**

No statements to execute administrative actions for the communications component SAP need to be entered in the configuration file SAPCOMM.CFG.

**DESTINATION Parameter Group**

You must include a separate parameter set of the category DESTINATION in the PROGRAM parameter set SAP for every individual SAP System and every individual client with which the communications server is to exchange data.

Specify an SAP System to which the communications server is to be linked as follows:

```
DESTINATION <Destination name>
```

```text
SYSTEM <Sideinfo entry>
CLIENT <number>
```
SAP-specific Entries in the SAPcomm Configuration File

END-DESTINATION

The individual parameters of the category DESTINATION that you specify in a program group are explained below.

**SYSTEM**

Mandatory. The communications server must have access to a certain set of network-specific information in order to establish a CPI-C link for an SAP System.

It gets this information from the so-called sideinfo table. For more information on sideinfo files, please refer to the following documentation:

*SAP Communication: Configuration*

The sideinfo table must contain a separate entry for every SAP System that is to be accessible via a CPI-C link. Each of these entries can in turn be identified by a respective key word. Specify the key word that belongs to the entry in the sideinfo table allocated to your SAP System as "sideinfo entry".

**CLIENT**

Mandatory. Specify the number of the client with which the communications server is to exchange data here.

**USER**

Mandatory. Enter the name with which the communications server can log on to the SAP System.

Please see the hints in the section *Configuration in the SAP System [Seite 64]*.

**PASSWORD**

Mandatory. Enter the password appropriate to the "user ID" here.

**LANGUAGE**

Mandatory. Enter the language identifier with which the communications server is to log on to the respective SAP System.

**RELEASE**

Only to be entered for SAP update levels before 5.0C (R/2) or 2.1 (R/3). In such cases, you enter the category of the SAP System (R/2 or R/3) as well as the update level in the form "<category>-<update level>" here - for example, R/2 50b, R/3 11h, and so on.

**REPORT**

Optional. Enter the name of the ABAP report that is to be started by the communications server after creating the link to the SAP System.

Only enter this parameter if you want to use a report other than the one provided by SAP in the standard system for data exchange between the communications server and the SAP System.

**TRANSPORT**

Optional. For OS/2 only. You must only specify this parameter if you want to change the default access method for SAP connections (in accordance with your network...
SAP-specific Entries in the SAPcomm Configuration File

configuration).
Default: Access to R/2 via SNA, to R/3 via TCP/IP

DCONVERT <conversion file>
Optional. Only affects the connection to R/2 Systems. If you specify this parameter, the
specified conversion file is used for data exchange with the SAP system (in
DESTINATION). The parameter DCONVERT overrides the parameter CONVERT if
specified (see below).
For more details, read the section “Character Sets and Code Pages”.

CPIC_BUFFER_LEN <buffer length>
Optional. Only for R/2, ignored in R/3.
Use this parameter to set the maximum size of blocks sent via CPI-C.
Only use this parameter when working with IMS and if the buffers of the input/output
message queue are smaller than 4096. Permitted values are 300 to 4096 bytes.
If the parameter is not specified, the default value 4096 is used.
If you set the CPIC_BUFFER_LEN to a different value, one of the following requirements
must be met:
• R/2 Release: at least 5.0F
• Correction K103962 was imported
As an alternative to this correction, you can also set the value of MAX_SIZE in the
structure STREAM in report RSEINCA1 to the value of CPIC_BUFFER_LEN. If you
choose this path, you must regenerate all reports that include RSEINCA1.

SECURITY <NONE | SAME | PROGRAM>
Optional. Only applies to connection to R/2 host systems with security systems
(TopSecret, RACF, etc.).
Default value NONE.
This parameter has no influence on SAP System logon. This is done with the USER and
PASSWORD parameters in the DESTINATION group.
NONE:
No user information is passed to the security system.
SAME:
Only the user name is passed to the security system.
PROGRAM:
The user name and the ID are passed to the security system.
The user name and password are obtained from the values for USER and PASSWORD of the
DESTINATION parameter group. If the host user and SAP user are different, you can set the
values that are passed to the host security system with the environment variables
SAPCOMM_USR and SAPCOMM_PASS.

DESTINATION K50_000
SYSTEM K50
CLIENT 000
USER SAPCOMM1
PASSWORD XXXXXXXX
LANGUAGE D
RELEASE R2-50G
SECURITY    SAME
END-DESTINATION

**SELECTION Parameter Group**
Here you define groups of selection criteria for retrieving outgoing documents from an SAP System.

The parameters of the SELECTION parameter group:

**APPLICATION <Application name>**
Mandatory. Enter TELE, MAIL or EDI as application names depending on whether the parameters of this group refer to the selection of telecommunications, mail or EDI documents.

**Further parameters:** The actual selection parameters are specific to each individual application. The selections options are dependent on the components in the SAP System and are oriented towards SELECT OPTIONS in ABAP/4.

As a principle, each SELECTION group is made up of the parameter APPLICATION and one or more SELECT OPTIONS groups.

In each of these SELECT OPTIONS groups, the user states the mandatory parameters FIELD and LOW, as well as HIGH, SIGN and OPTION as required. The name of the SELECT OPTIONS group is not significant.

**SELECT-OPTIONS <Name>**
- **FIELD** <Field name>
- **LOW** <Name>
- **HIGH** <Name>
- **SIGN** <INCL|EXCL>
- **OPTION** <ABAP/4 options: GE,...>

**END-SELECT-OPTIONS**

The default values for SIGN and OPTION are INCL and EQ.

It is possible to put together any combination of SELECT OPTIONS groups with the allowed field names per application in one respective SELECTION group.

**Application MAIL**
Allowed field names for application MAIL.

**SDART <transmission type>**
Possible values for <transmission type> are:
- X40 Selects messages for X400 addressees
- SSW Selects messages for SNADS addressees
- RML Selects messages for other SAP Systems

You want to define a SELECTION group via which you can choose all mail documents to other SAP systems. You name the relevant parameter group SAP-MAIL, for example. The SELECTION group then appears as follows:

```
SELECTION       SAP-MAIL
    APPLICATION MAIL
    SELECT-OPTIONS RMAIL
        FIELD SDART
        LOW RML
    END-SELECT-OPTIONS
END-SELECTION
```
Application TELE

Allowed field names for application TELE.

DEST <Destination>

This selection is possible as of R/3 Release 2.1. <Destination> is the name of a spool destination. This enables a subset of the selection to be selected via Table T164C.

You want to define a SELECTION group via which you can select all fax documents of the output device (= spool destination) TF01. You name the relevant parameter group FAX-01, for example. The SELECTION group then appears as follows:

```
SELECTION  FAX-01
  APPLICATION  TELE
  SELECT-OPTIONS  FAX01
    FIELD  DEST
    LOW  TF01
  END-SELECT-OPTIONS
END-SELECTION
```

Application EDI

Allowed field names for EDI.

COMMTYPE <communication service>
TRNSLSET <EDI allocation set>
COMPANYC <company code>
PARTTYPE <partner type>
PARTNUMB <SAP partner number>
TCODE <transaction code>
DOCMTYPE <document type>
DOCMVAR <document variant>
DOCMNUMB <document number>
FUNC_ACK <functional acknowledgments>
MESTYP <message type>
MESCOD <message variant>
MESFCT <message function>

Please note:
You cannot use the field name FUNC_ACK together with other field names in a download session. When you use FUNC_ACK, LOW ON must also be specified.

For the precise meaning of these selection criteria, please refer to the field documentation of the ABAP/4 Report RSESPL01 (F1 documentation). In particular, you will learn which of the above SELECT-OPTIONS are supported by your SAP System.

You want to define a SELECTION group via which you can select all outgoing EDI documents destined for partners "between" ALPHA and GAMMA (with the exception of partner BETA) and which also belong to the partner category "vendor" and Transaction TE21 (purchase orders). You name the corresponding parameter set - for example, ORDERS-TE21. The SELECTION group then looks as follows:

```
SELECTION  ORDERS-TE21
  APPLICATION  EDI
  SELECT-OPTIONS  NUMB1
    FIELD  PARTNUMB
```
CHANNEL Parameter Group

The CHANNEL parameter sets that you create for communications component SAP can contain the following parameters:

DESTINATION <destination name>

Mandatory. Enter the destination name under which you stored the parameters for a specific SAP System. All data exchange activities via the current channel refer to this SAP System. You can only specify one DESTINATION per channel.

PERFORM <PERFORM name>

Mandatory. Only one action (RECEIVE,...) can be specified in each PERFORM parameter group.

There must always be a reference to a SELECTION for the RECEIVE action. Selection of the APPLICATION is sufficient here.

TEST_CHANNEL <ON | OFF>

Refer to Testing the Connection [Seite 102].

CRCV_TIMEOUT <number of seconds>

This parameter overrides the timeout set by the PROGRAM parameter RCV_TIMEOUT for the appropriate channel. However, the CRCV_TIMEOUT parameter only takes effect AFTER SAP System logon.

If you are using a communications component that supports incoming faxes, you must configure the following CHANNEL parameter:

FAX_SPOOL_DIR <directory>

Name of the directory (from SAPcomm point of view), in which incoming faxes are stored.

FAX_SHARE_DIR <directory>

Name of the FAX_SPOOL_DIR from the point of view of the R/3 terminals.

If the directory for storing faxes (FAX_SPOOL_DIR) has another name from the point of view of the R/3 terminals (PCs, at which R/3 users are working) - this is a LAN
configuration matter - then you must configure this name using the FAX_SHARE_DIR parameter. This can also be a logical path name defined in R/3.

Fax reception is only possible with R/3 as of Release 3.0. With fax reception, an info record is created in SAPoffice, which contains the names of the files under which a received fax was stored. It is then possible to access these files from SAPoffice (display via viewer, forward, etc.).
Character Sets and Code Pages

This topic is only relevant if individual characters are converted incorrectly when data is exchanged between SAPcomm and an SAP System (special characters, diaereses, etc.).

In the SAPcomm environment (on the workstation), data is represented in the SAP code page 1103 (IBM code page 00697/00850). This code page is usually different from a code page used in an SAP system, which means that when data is exchanged between SAPcomm and an SAP system, the data must be converted from one code page to the other.

**Special Features in R/3:**

When data is exchanged with an R/3 System (SAP code page 1100 (ISO 8859-1)), the conversion is performed in the R/3 System if necessary. This conversion is not described here.

**Special Features in R/2:**

When data is exchanged with an R/2 System, the data conversion is performed on the workstation. This process normally assumes that the data in the SAP system is represented in the SAP code page 0100 (IBM 00697/00273). If your situation is different, you must modify the conversion on the workstation.

To do this, generate one or more of your own conversion files and let SAPcomm - using the SAPcomm configuration parameter CONVERT or DCONVERT - use the desired conversion files for data conversion.

There are two ways of creating your own conversion file:

- Modify an existing conversion file using an editor
- Use the program GENCPTAB (OS/2 only) to create a conversion file.
  For example, when you enter "GENCPTAB 850 273" the program generates a conversion file to convert data in code page 850 to data in code page 273.
Testing the Connection

To check the network connection between the communications server and an SAP system, you can use the SAPcomm auxiliary program `scmping`.

You can also use `scmping` to perform the following tests:

- Run test for programs, which were started in the SAP system by the communications server
- Validity of the logon data stored in the configuration file

A description of the auxiliary program `scmping` can be found in *The SAPcomm Auxiliary Programs [Seite 232]*.

Perform the test with `scmping` before you test data exchange between the component SAP of the communications servers and a SAP System.

Activation of the functionality for transfer of test data from/to an SAP system is bound to an entire channel, not to a PERFORM parameter group. You first define a channel as a "test channel" by specifying the CHANNEL parameter TEST_CHANNEL (with ON).

There are three ways to test the applications (EDI, Mail, telecommunications) in the SAP system (not the network connection), in particular to test incoming processing:

- Create an outgoing message "for" a channel via which documents are transferred into the SAP System.
- Create an outgoing message for any channel and "redirect" this message to a channel via which documents are transferred into the SAP System. For this, use program `scmmove` (see *The SAPcomm Auxiliary Programs [Seite 232]*).
- Create documents using "communications component" TST (only telecommunications documents for R/3).

By creating or modifying standard texts for the test (for RECEIVE), or - for a SEND action - the files generated by a RECEIVE action before sending, you can examine problems in the area of data conversion (see above).

Example configuration of a test channel:

```
CHANNEL TEST_CH
DESTINATION    SAP1
TEST_CHANNEL   ON

PERFORM DOWNLOAD_TEST
  ACTION RECEIVE
END-PERFORM

PERFORM UPLOAD_TEST
  ACTION SEND-SCHEDULE
END-PERFORM

END-CHANNEL
```

Always create a separate channel for the test.

Never change a production channel to a test channel because the data, which you have loaded into the SAP system as test data, cannot be loaded subsequently into a production SAP environment.

Do not create any incoming files "manually" - you could put the number assignment system of the communications server into an undefined state.
Transmission of Test Data From the SAP System to SAPcomm

In this test, you load the data stored in a standard SAP word processing text onto the communications server.

The standard text must be created under the following name in the SAP System:

R/2:  40001
R/3:  S_EDI_TEST

If you cannot for any reason use the predefined text number, choose another and then change the routine GET_TEST_TEXTNR of ABAP/4 report RSECPIC0 correspondingly.

Carry out the following steps for the test:

- Start the communications server (if it is not already running) and choose
  Perform → Transfer.
- Choose the channel provided for the test and the PERFORM group DOWNLOAD_TEST.

You should find the standard text of the channel used for the test back in the data directory.

If this is so, the test was successful.

If this is not so, try to localize and eliminate the error using the topic Notes on Problem Analysis [Seite 105].

Each RECEIVE action, initiated via a test channel, loads the standard text on the workstation with the communications server, and stores it in the file directory assigned to this channel.

Transmission of Test Data From SAPcomm to the SAP System

In this test, you transfer files from the communications server to the SAP System and store them in SAP Table EDCT.

Each SEND action, initiated via a test channel, loads the data stored in the directory assigned to this channel into the SAP System, and stores it in Table EDCT. Do not generate these files; Use the files created previously through a RECEIVE.

The Upload test presupposes that test data already exists in the test channel directory (e.g. from a successfully executed Download test). If test data is not available, you will merely get the following entries in the log file of component SAP, providing that you have set the TRACE parameter SCI CALL to ON:

- I0600 Call SendFiles
- I0601 Back from SendFiles

Carry out the following steps for the test:

1. Activate the debugging mode by setting the Debugging parameter in Table 164b of the SAP System to the value X.
2. Start the communications server (if it is not already running), and choose Perform → Transfer.
3. Choose the channel provided for the test and the PERFORM group UPLOAD_TEST.

The communications server now transfers predefined data (“START OF TEST DATA....END OF TEST DATA’) to the SAP System. Here, the data transferred is stored in Table EDCT. Check whether the data has been correctly stored in Table EDCT.

If this is so, the test was successful. Deactivate the debugging mode via Table 164B.

If this is not so, try to localize and eliminate the problem using the topic Notes on Problem Analysis [Seite 105].
Notes on Problem Analysis

For an efficient and successful analysis of communication problems between the communications server and an SAP System, it is fundamentally important to find out in which communication phase the problem occurs. Always try to localize the problem according to the following categories:

- **Connection Setup Fails**
  - The attempt to establish the connection fails locally through the communications subsystem (e.g. *Communications Manager* of the SAPcomm workstation).
  - The attempt to establish the connection receives a negative or no response from the target system.

- **Connection Termination Immediately After Setup**
  Breakdown or termination of the connection occurs immediately after the connection is established.

- **Connection Termination After Data Transmission**
  Connection termination occurs after the transmission of some data.

This first localization of the problem can, in most cases, be carried out using the error entries in the log file of the communications component SAP.

Only in the first two of the cases stated above is it generally a purely network problem - and even then, it is almost always a configuration problem.

These error categories are explained in the following topics.

### Connection Setup Fails

The attempt to establish the connection fails locally through the communications subsystem (e.g. *Communications Manager* of the SAPcomm Workstation) or receives a negative or no response (Cases 1 and 2).

The SAP basis and network software (Communications Manager, and so on) provide special utilities for the analysis of network problems:

- **Data trace at the SAP basis software level**
  For more information, read the section "Error Analysis" in the documentation *SAP Communication: Programming*

- **Data traces at network software level**
  For this, read the Online Documentation.

If necessary, consult the relevant guides in order to analyze the trace files (for example, the IBM manual "Programming Services and Advanced Problem Determination for Communications" and the literature it refers to).

Only the "commonest" problems in the host-workstation link are discussed in the following topics.

### Error CPI-C: CMINIT

(Entry in the log file of the communications component SAP.)

This entry indicates a problem relating to the sideinfo file.

First check whether SAPcomm is using the correct sideinfo file: Entry I1720 of the log file, which is created by component SAP, tells you which file SAPcomm is using as the sideinfo file.

Then check whether the following configurations match:

- **SYSTEM entry in the configuration file** *sapcomm.cfg* and the corresponding entry in the side info file (see *Modifying the Configuration and Communications Files* [Seite 31].)
Notes on Problem Analysis

- Configuration of your network software and entries in the side info file

**Error CPI-C: CMALLOC**

This entry indicates that the system has identified the partners (LUs - Logical Units) between which a link is to be set up, but either the local or the remote LU is not ready for connection setup. Check:

- Whether your local LU is busy - for example, due to a previous link to another partner LU or via the SAP CUA Interface (if necessary, deactivate existing "sessions" that are no longer required).
- Whether the SAP System is available on the host.
- Whether the attempt to set up the connection failed on your workstation, or whether the host responded negatively to the attempt to set up the connection. This cannot be determined from the SAP log files. Use the trace options of your communication subsystem (the Communications Manager...) instead.

**Error WS-host synchronization: Logon failed OR:**

No CM_SEND_RECEIVED after logon

This entry indicates that the link to the SAP System was created, but the ABAP/4 program "demanded" by the communications server could not be started there. Check:

- Whether the values for client, user ID, password and, if necessary, ABAP/4 program entered into the configuration file are correct,
- Whether the ABAP/4 programs RSECPIC0 and RSECPIC1 can be executed when called.
  (Start the programs online in the SAP System. No processing takes place as a result - if an error occurs, however, you receive an appropriate message).
- The SAP System log.

**Error CPI-C: CMRECV (or CMSEND):**

This entry generally indicates that an abnormal termination occurred in the SAP System during the data transmission phase. Check:

- The SAP System log,
- If necessary, the SAP EDI log file on the host (in the menu option Information System of the SAP EDI menu).

**Connection Termination Immediately After Setup**

Immediately after the connection is established, it breaks down or terminates.

One of the following problems is usually present:

- Values specified for user, password or client are invalid.
- The ABAP/4 program called - or one of the subprograms called by the ABAP/4 program - does not exist, is incorrect or has an obsolete load phase.

**Connection Termination After Data Transmission**

After the transmission of some data the connection terminates.

One of the following problems is usually present:

- One of the programs responsible for the processing of data loaded on the SAP System or to be loaded on the workstation, does not exist, is incorrect or has an obsolete load phase.
- During data processing, a runtime error occurred as a result of authorization problems, data access (file full,...) or timeout.
In all the cases given (apart from incorrect client) you will find notes in the log file of the SAP System (System log, RSSLOG), which will either offer direct help on eliminating the error or will give additional information for further problem analysis.

**Error Entries in the Log File of the Component SAP**

The following section lists the error log entries specific to the communications component SAP, sorted according to error number, and explains each of them. You will find explanations of general error log entries in *Notes on Testing and Problem Analysis* [Seite 87].

**E0107 Download data incomplete**

*W0601 Message incomplete ==> Message Lost*

The data transferred to SAPcomm from the SAP System is incomplete. The data was archived (the name of the archive file is entered in the log file). Check data for possible transfer errors.

**E0108 Priority invalid**

The priority entered in the data transferred to SAPcomm from the SAP System is invalid. The default priority "SCH" (=data is to be sent according to the time plan) was used.

**E0112 Negative host acknowledgement from host for <File>**

The data contained in file <file> and transferred from SAPcomm to the SAP System could not be processed correctly. Check the incoming data in the SAP System for errors using the SAP transactions (display the EDI log, and so on).

**E1532 Security UserID has invalid size = <Length>**

The user name stated for the security system used (e.g. RACF) is of invalid length. It must be between 1 and 8 places. Check the settings in sapcomm.cfg and those of your environment variable SAPCOMM_USR, as applicable.

**E1542 Security Password has invalid size = <Length>**

The password stated for the security system used (e.g. RACF) is of invalid length. It must be between 1 and 8 places. Check the settings in ssapcomm.cfg and those of your environment variable SAPCOMM_PASS, as applicable.

**E1699 No SELECTION for download! Change sapcomm.cfg**

You have defined a PERFORM group in the component SAP with the ACTION RECEIVE, without stating a SELECTION group. Therefore, no messages are transmitted via this PERFORM to SAPcomm. Insert a SELECTION group in the relevant PERFORM.

**E1702 Channel <Name> unknown**

The channel stated in the data transferred from the SAP system to SAPcomm is not contained in the configuration file SAPCOMM.CFG.

Check the following:

- settings in the SAP system
  (field *Communications service* for EDI messages, etc.)
- channels configured for SAPcomm in *The Configuration File SAPCOMM.CFG* [Seite 35]

**E1756 Negative acknowledge <Error code> from SAP system**

An error occurred in the SAP system when processing an incoming message or status message transmitted from SAPcomm to the SAP system. For more information about this error and its cause, see the SAP system log.
Connection to ODEX

The Communications System ODEX [Seite 109]
Hardware and Software Requirements for ODEX [Seite 110]
The Communications Component ODX [Seite 111]
ODX-specific Entries in the SAPcomm Configuration File [Seite 112]
SAP-specific Aspects for ODEX Operation [Seite 115]
Notes on Testing and Problem Analysis: ODEX [Seite 117]
The Communications System ODEX

The ODEX product line consists of communications programs that support data transmission in accordance with the Odette file transfer protocol (OFTP) based on X.25. They allow file transfer both in point-to-point connections with your business partners and via all value added network services (VANs) supporting OFTP. If in doubt, please consult your VAN contact person.

The ODEX product line is produced by Data Interchange, Peterborough, England.
Hardware and Software Requirements for ODEX

ODEX is available in different versions and for different platforms.

The SAP communications server supports the connection of ODEX on the following system platforms only:

- **OS/2**
  - Only ODEX Version 1.5.1 for OS/2
  - (and compatible upgrades)

- **HP-UX**
  - ODEX/UNIX

At the time of printing, no other version can be connected to the ODEX interface of the communications server.

The communications programs of the ODEX product line require an X.25 adapter board and the appropriate driver software for connection to an X.25 network. For details on this and guidelines on the installation and configuration of ODEX, please refer to the ODEX document or speak to your ODEX sales contact person.
The Communications Component ODX

The connection of the communications server to the communications system ODEX is handled by the communications component ODX.

You can send both EDI files and any other files via the communications component ODX. Here, "EDI files" have data which complies with an EDI standard supported by ODEX (Edifact...). All other files are "Non-EDI files".

When sending non-EDI files, ensure that address information can be passed to ODEX.

There are two ways to do this:

- From Release 5.0D of R/2: In the SAP EDI Menu ("Communications addresses") assign the "Communications ID" X40 to communications services intended for the sending of "Non-EDI files" via ODEX. In the relevant X400 addresses, maintain the DDA field (Domain Defined Attributes) with the ODEX local code of the addressees (the remaining X400 address fields are ignored).

  The "Communications service" is also the name of a SAPcomm channel which is assigned to the communications component ODX (as in the case of "normal" EDI files).

- Fixed assignment of a SAPcomm channel to an addressee configured in ODEX.

  Please read the explanations of the MODE and DESTINATION channel parameters.

All documents received via ODEX are handled as EDI documents.

The communications component ODX communicates with ODEX via a command file stored in the ODX work directory and the response file ODEXPC.RSP created by ODEX. The following steps must be followed to execute an action via ODEX:

- The communications component generates the command file. This file contains commands corresponding to the action to be initiated.

- The communications component calls ODEXBAT.EXE and transfers the command file to this program.

- ODEXBAT.EXE processes the command file and generates the response file ODEXPC.RSP.

- The communications component evaluates the response file ODEXPC.RSP. If an error occurs, ODEXPC.RSP is renamed and a corresponding message entered into the log file. This ensures that the error messages from ODEX are not overwritten. Please refer to the ODEX document for help with the analysis of the ODEX response files.

Please remember that ODEXBAT itself does not perform data transmission but only receives corresponding orders, checks them and processes them for the driver program ODEDRV01.

Data transmission - both sending and picking up - is carried out asynchronously at the communications server. This means, for example, that ODEXBAT has not yet sent the data after it has processed a transmission request, and that no incoming documents have arrived yet after it has processed a fetch order!
ODX is the identifier of the communications component responsible for connection to the communications system ODEX. You therefore add the following parameter group to the configuration file SAPCOMM.CFG:

```
PROGRAM ODX
    <Parameter specification>
END-PROGRAM
```

The parameters of this group are explained below. A general description of the PROGRAM parameter group can be found in [Structure of the Configuration File Seite 37] .

**Component-specific PROGRAM Parameters**

You can include the following parameters in the PROGRAM parameter set ODX:

- **RECEIVER_LC** <Odex Local Code>
  
  You can instruct the communications server to accept only those documents received for ODEX that are addressed to particular recipients.
  
  Please remember that this selection affects all of the documents received by ODEX, and not only those retrieved using a RECEIVE time plan entry!
  
  For example: enter "RECEIVER-LC SAPS" if you only want to accept documents addressed to a recipient who is configured for ODEX, for example, as a file node with the local code SAPS.
  
  On the other hand, you can enter "RECEIVER-LC SAP*", if you want to receive documents addressed to recipients who are configured for ODEX as file nodes with local codes starting with SAP.
  
  Please take note of the difference between receiver and local sender codes! If required, specify several of these internal file nodes as recipients in order to make an accurate selection!
  
  If you do not enter the parameter RECEIVER-LC, the communications server transfers all incoming documents from ODEX.

- **VIRTUAL_FN** <Name>
  
  If you enter this parameter, the "Virtual File Name" in outgoing messages is set to <name>. Default: SAP EDI.

- **DEF_COMMSERVICE** <ON | OFF>
  
  Optional. If you specify this parameter and set it to ON, the communications service "ODEX" is assigned to all ODEX incoming messages. Otherwise the communication service is copied from the Odex field Company Code (as described in the following topic "[SAP-specific Aspects for ODEX Operation Seite 115]").

- **STATUS_ON_XMIT** <ON|OFF>
  
  This parameter only concerns outgoing documents. Documents sent via ODEX are normally only marked as sent when a receipt (EERP) has come from the recipient.
  
  You can instruct the communications server not to wait for this receipt, but to mark documents as sent when transmission has been completed correctly (i.e. particularly when the document may not yet be with the receiver, but has first arrived at a VAN).
  
  For this, set the parameter STATUS_ON_XMIT to ON.
ODX-specific Entries in the SAPcomm Configuration File

**IGNORE_INCOMING_MESSAGES**  <ON|OFF>
If you do not want to process incoming messages, set this parameter to ON.
Default value: OFF

**BATCH_CHECKING_ONLY**  <ON|OFF>
Set this parameter (to ON) only if messages coming in via ODEX are not stored on the same drive as SAPcomm.
Default value: OFF

**DELETE_ODX_FILES_SENT**  (ON/OFF)
Optional. If the OFF value is set, dispatched files are not deleted in ODEX. These files will be kept in ODEX until they are deleted by an ODEX reorganization.
Default value: ON (dispatched files are automatically deleted)

**USE_SENDER_LC_FOR_ROUTING**
Optional. If the ON value is set, incoming messages are assigned to their target channel via the ODEX Local Code of the sender. The relevant parameter in the ROUTING parameter group keeps its name ("RECEIVER-LC").

**PERFORM Parameter Group**
No statements are entered in configuration file SAPCOMM.CFG to execute ODX-specific administrative actions for the communications component ODX.
Only one ACTION parameter is allowed in each of the ODX-PERFORM parameter sets.

**DESTINATION Parameter Group**
You only need to include DESTINATION parameter sets in the PROGRAM parameter set in the following cases:

- You want to retrieve incoming messages from a remote communications partner.
- You want to send files as "Non-EDI files", and the respective addresses are not to be passed, along with the files, to the communications component ODX.

In these cases, you must create a DESTINATION parameter set for each network node (in the sense of the ODEX user directory) for which you want to set up a link. Each of these groups contains the following parameter:

**LOCALCODE**  <Odex Local Code>
Here, you specify the local code which is assigned to the network node in the ODEX user directory.
You can then reference the destination defined here - and therefore the respective Odex local code - through the CHANNEL parameter DESTINATION.

**SELECTION Parameter Group**
No SELECTION parameter needs to be specified for the communications component ODX.

**CHANNEL Parameter Group**
The CHANNEL parameter sets that you create for the communications component ODX can contain the parameters described below:

**DESTINATION**  <destination name>
This parameter can be used in connection with both incoming and outgoing messages:

1. Incoming messages
   Enter the parameter DESTINATION if you want to set up links to remote communications...
partners via the present channel to retrieve incoming documents (and not, or not only, to send documents).

If so, you must enter as a parameter value the destination name under which you stored the local code of the communications partner. You can only enter one DESTINATION per channel!

2. This parameter is used in connection with outgoing messages to determine local codes of receivers of documents, which are to be sent as "Non-EDI files" and do not have a receiver X400 address.

**PRIORITY**  \(<values \, 0-9, \, default \, value \, 5>\)

Optional: The value entered here is passed to ODEX for priority control. It does not affect document control within SAPcomm.

**MODE**  \(<EDI \, | \, TEXT>\)

This parameter only affects outgoing documents. It determines whether outgoing documents are passed to ODEX as EDI files or Non-EDI files via the respective channel. If you do not specify this parameter, MODE EDI is assumed.

If you specify MODE TEXT, the recipient address of a document to be sent is determined as follows:

First the communications component ODX checks whether it has received an X400 address along with the document.

- If so, the X400 address field DDA (Domain Defined Attribute) must contain the local code of the recipient configured in ODEX. All other X400 address fields are ignored.
- If this is not so, the local code of the recipient configured in ODEX is determined using the CHANNEL parameter DESTINATION.
SAP-specific Aspects for ODEX Operation

Bear in mind the following notes when installing and configuring ODEX:

- Install on the same drive as the communications server (OS/2 only).
- Enter all the partners with which you want to exchange EDI messages into the ODEX user directory (see ODEX manual).
- Files sent via by ODEX are only marked as "sent" when a positive receipt (EERP) has come from the recipient.
- When defining a network node, enter 'A' in the field "When to send EERP".
- For every EDI partner from whom EDI messages are received, enter their "communications service" (from the SAP EDI partner agreements of the SAP System) into the ODEX user directory. Namely:
  - if the EDI partner is defined as a file node, you enter the communications service into field Company Name of this file node;
  - if the EDI partner is defined as a message node, you enter the communications service in the field Company Name of the associated network node.
  - the entry starts in the first column. The subsequent text must be separated by one or more blanks.

You have entered the communications service MARKIII on the host for your vendor LIEF001. LIEF001 is entered as a message node in your ODEX user directory. This message node refers to the network node NETNODE.

The entry in field Company Name of network node NETNODE in the ODEX user directory then has the following appearance:

MARKIII - Network node for GE access

The entry in field Company Name of the message node has no significance as far as SAP software is concerned.

Every entry of a network node in the ODEX user directory (see the ODEX manual) contains a parameter Attempts called so far. This parameter is incremented by one for every failed attempt ODEX makes to set up the connection. If it reaches the value of the parameter Max Retries in the same entry, ODEX halts the attempt to set up the connection.

This situation is not recognized by the communications server.

Remedy

The parameter Attempts called so far must be reset to 0. This is done each time the communications server issues an order to ODEX to retrieve incoming messages for the network node (entry RECEIVE in the configuration file SAPCOMM.CFG). You can therefore use the time plan to ensure that ODEX does not finally give up its attempt to set up the connection.

Operating ODEX

The program ODEXBAT (called by the communications component ODX) does not perform data transmission itself but makes data available to the actual data transfer program ODEDRV01. Data is only transferred - and incoming documents can only be received - if ODEDRV01 is running.
SAP-specific Aspects for ODEX Operation

You must therefore start Program ODEDRV01 in a separate session before or after starting the communications server. Never terminate the program, but rather halt its execution by pressing PF3. Please read the ODEX manual for more details on this subject.

You can also use ODEX (at the same time as the communications server) to send and receive documents from the ODEX menu or from other programs. Please bear in mind:

- **Sending:**
  Do not use a *Virtual File Name* used by SAPcomm (see the explanation on the PROGRAM parameter VIRTUAL-FN).

- **Receiving:**
  Use the configuration parameter RECEIVER-LC (see above) to set the communications server to only receive those documents from ODEX that are intended for it. If you do not enter this parameter, the communications server transfers all the incoming documents from ODEX!
Notes on Testing and Problem Analysis: ODEX

Problems during interaction between the communications server and ODEX mostly fall into one of the two following categories:

- **Orders refused**
  Send or fetch orders are refused by ODEXBAT;

- **Orders not carried out**
  Send or fetch orders are accepted but not carried out

**Orders Refused**

Every rejection of an order by ODEXBAT is noted in the ODX log file. Typical log entries in these cases are:

- E0202 FTP File name missing.
- E0502 Error Odex function (...).

In addition to these log entries, you will also find messages in the ODX log file stating that the ODEX response file was saved (and under which name). Examine this (text) file - it contains further information on why ODEXBAT could not process the order.

**Orders Not Carried Out**

If you cannot find a message stating that ODEXBAT rejected an order and yet no data transmission takes place, then please use the functions supplied in the ODEX user program (menu options File processing, Maintenance, Audit/Log, and so on) to locate and correct the problem.

**General Notes**

Finally, another two general notes on ODEX functions that support you when working with ODEX (configuration, testing, problem analysis...):

- You can monitor the data transmission and switch the logging of certain actions on or off at the run-time of the driver program ODEDRV01.

  All important actions (connection setup, data transmission, acceptance of calls,...) are entered into the ODEX log file. The ODEX user menu has options for displaying and printing the log file.

- You can mark the "Nodes" (Network,...) in ODEX as internal in order to test the sending and receiving of documents, and the addressing scheme in particular. Documents that are addressed to an internal node are not passed on to X.25 software for sending, but are supplied by ODEX as incoming documents. You do not need X.25 hardware and software for this operation.

**Error Messages in the ODX Log**

The following lists error log entries specific to the communications component ODX, sorted by error number, and explains each of them. General error log entries are explained in Notes on Testing and Problem Analysis [Seite 87]

**E0114** No corresponding control file for ODEX status report.

  The communications component ODX could not process all the send status information supplied by ODEX. This error can generally be traced back to the fact that ODEX was also used to send data manually. It does not interrupt the transmission and receiving processes.

**E0201** ODEX information on outgoing file <name> is incomplete.
Notes on Testing and Problem Analysis: ODEX

The ODEX response file ODEX.RSP is incomplete. It has been archived (for the name of the archive file, please refer to the ODX log file). Check the messages written to this file.

**E0202 FTP file name missing in ODX response file**

The ODEX response file could not be processed correctly. It has been archived (name in the ODX log file). Check the messages written to this file. This error generally indicates that the ODEX user directory has not been maintained correctly.

**E0503 Length of CommService invalid: <no.>. Incoming message ignored.**

The Company Code field of the sender of an incoming message does not have a valid communications service.

Remedy: The only way you can pass the document to SAPcomm is by setting the ODX PROGRAM parameter DEF COMMSERVICE to ON, and restarting SAPcomm. This parameter affects ALL Odex incoming messages.
Connection to EXPEDITE

- The Communications System EXPEDITE [Seite 120]
- Hardware and Software Requirements: EXPEDITE [Seite 121]
- The Communications Component EXP [Seite 122]
- EXP-specific Entries in the SAPcomm Configuration File [Seite 123]
- SAP-specific Aspects for EXPEDITE Operation [Seite 125]
- Notes on Testing and Problem Analysis: EXPEDITE [Seite 126]
The Communications System EXPEDITE

In this document, the name EXPEDITE refers to the IBM expEDite OS/2 Base communications program. This product allows data to be exchanged between an OS/2 workstation and the IBM Information Exchange, via the IBM Information Network.
Hardware and Software Requirements: EXPEDITE

You need the OS/2 version of the SAP communications server in order to connect to EXPEDITE. The workstation can establish a connection to the Information Network either directly or via the mainframe. It is possible to use an existing leased line from the mainframe to the Information Network. The mainframe product expEDite/MVS is not required for this.

The technical network and organizational aspects of connecting EXPEDITE to the Information Exchange are not covered in this document. Please consult your IBM contact person for this information.
The Communications Component EXP

The connection of the communications server to EXPEDITE is handled by the communications component EXP.

This communications component is used exclusively for the exchange of EDI documents. Documents received via EXP and forwarded to an SAP System always arrive via the EDI component of the SAP System and cannot be processed as MAIL documents.

The communications component EXP communicates with EXPEDITE via the command file BASEIN.MSG and the response file BASEOUT.MSG created by EXPEDITE. The following steps are required in order to execute an action via EXPEDITE:

- The communications component EXP generates the "command file" BASEIN.MSG. This file contains the necessary commands for the action to be initiated.
- The communications component starts EXPEDITE and waits until EXPEDITE completes the execution of its functions.
- EXPEDITE meanwhile processes the command file, sets up the link to the IBM Information Exchange, carries out the data transmission, and finally generates the response file BASEOUT.MSG.
- The communications component evaluates the response file BASEOUT.MSG. If an error occurs, this file is renamed and a appropriate message is entered into the EXP log file. This ensures that the error messages issued by EXPEDITE are not overwritten. For assistance in analyzing these files, please refer to the EXPEDITE guide.
EXP is the identifier of the communications component responsible for the connection to the EXPEDITE communications system.

You therefore include the following parameter set in the configuration file SAPCOMM.CFG:

```
PROGRAM EXP
  <Parameter specification>
END-PROGRAM
```

The parameters of this group are explained below. A general description of the PROGRAM parameter group can be found in Structure of the Configuration File [Seite 37].

**PERFORM Parameter Group**

No statements need to be entered in configuration file SAPCOMM.CFG to execute EXP-specific administrative functions for the communications component EXP.

**DESTINATION Parameter Group**

No DESTINATION parameter sets need to be entered in configuration file SAPCOMM.CFG for communications component EXP.

**SELECTION Parameter Group**

You define selection criteria here to retrieve incoming documents from the Information Exchange. As parameters of a SELECTION parameter set, enter:

```
RCV-CLASS <class name>
```

Specify a name with a minimum of one and a maximum of eight alphanumeric characters. Only documents belonging to the class with this name are transferred to the communications server from the Information Exchange if you reference this SELECTION parameter set in your PERFORM statements.

You can define several SELECTION groups and reference them in different PERFORM statements. However, you can only reference a maximum of one SELECTION group in each individual statement.

If you do not reference a SELECTION group, then incoming documents of class 

```
#E?
```

are retrieved by Information Exchange.

Example of a SELECTION group to select incoming messages of the class SAP:

```
SELECTION SAP-EDI
  RCV-CLASS SAP
END-SELECTION
```

**CHANNEL Parameter Group**

The CHANNEL parameter sets which you create for the communications component EXP can contain the parameters explained in the following section:

```
SND-CLASS <class name>
```

Optional. You specify a name here that contains between one and eight alphanumeric characters. Documents sent to the Information Exchange via this channel have this name as their "class". Default class name: Blank.

```
TRANSLATE_TABLE <file name>
```

Optional. Using the TRANSLATE_TABLE entry EXPEDITE, you can allow a code translation other than the standard ASCII-EBCD code translation to be performed for all documents sent to the Information Exchange via this channel. The <filename> (no
EXP-specific Entries in the SAPcomm Configuration File

extension!) is forwarded to EXPEDITe as a TRANSLATE component of the SENDEDI command.

**PATH** <path name>

Mandatory. Enter the directory of your Expedite installation containing the file Basein.pro, which is specific to each SAPcomm channel.

**Component-specific PROGRAM Parameters**

**ACK_EXPECTED** <ON|OFF>

Optional. If this parameter is entered (with ON), outgoing messages are only marked as "sent" when a "delivery acknowledgement" has arrived from the IBM Information Exchange (Default: After transfer to the Information Exchange).

This parameter is only used with Expedite/OS2 Version 4.1 onwards. If you are using an older version of Expedite, ACK_EXPECTED is ignored.

**DEF_COMMSERVICE** <ON | OFF>

Optional. If you do not specify this parameter (or if you set it to ON), the "communications service" of incoming messages is always set to "EXPEDITe".

If you specify "DEF COMMSERVICE OFF", the communications service of the incoming messages is determined as follows:

- If the IBM Information Exchange parameter CLASS of the incoming message is set, its value is used as the "communications service" of the message.
- Otherwise, the communications service of the message is set to "EXPEDITe".

**EDIONLY** <ON | OFF>

This parameter determines the form in which the RECEIVEEDI command in the file BASEIN.MSG is passed to IEBASE by SAPcomm (refer to RECEIVEEDI in "Using Expedite Base /2 Message Commands" of the IBM Expedite Manual):

The command RECEIVEEDI is passed to IEBASE without the addition of "EDIONLY(y)

- The version of IEBASE is older than 4.1.
- The parameter EDIONLY is set to OFF AND the selection parameter RCV-CLASS for the current RECEIVEEDI command is set to a value of your choice.

Only messages of this class are then fetched from Information Exchange.
SAP-specific Aspects for EXPEDITE Operation

Bear in mind the following notes when installing and configuring EXPEDITE:

- Install it on the same drive as the communications server.
- The communications server does not create the EXPEDITE configuration file BASEIN.PRO. You must create this file yourself following the rules in the EXPEDITE manual and then configure EXPEDITE. The IEPATH parameter must be set in the SESSION command in this file.
- IBM IE error messages in clear text.
  Information Exchange passes error messages to SAPcomm as so-called system messages. These messages cannot be passed onto the SAP system at present. SAPcomm stores these messages in the `system.msg` file, located in the file directory specified by the PATH parameter of the respective channel.
  The messages are for your information only. The file is not deleted by SAPcomm. When the information is of no further use to you, delete the file. If need be, it can be recreated by SAPcomm.
- The "communications service" assigned to the incoming documents.
  Unless you specify otherwise, the "communications service" assigned to the incoming documents is "EXPEDITE". You can make SAPcomm set the communications service dynamically from the respective incoming message. To do this, read the explanations of the PROGRAM parameter DEF COMMSERVICE (see above).
- Working with various Information Exchange Accounts:
  Installation of the Expedite programs in a PATH directory:
  You must create a separate directory for each account for the the BASEIN.PRO file. Each of these directories is assigned to one or more EXP channels via the CHANNEL parameter PATH.

Operating EXPEDITE

The control component of the communications server starts the communications component EXP in a separate session (separate OS/2 window) directly after the program start.

This, in turn, starts EXPEDITE each time a data exchange with Information Exchange is to take place according to the time plan. Data transmission is performed synchronous to the communications component - that is, each time it issues an order to EXPEDITE, the communications component waits until EXPEDITE has carried out the order.

During this time, you should under no circumstances halt the communications server!
Never stop the communications component yourself. It is halted automatically when you shut down the communications server.
Notes on Testing and Problem Analysis: EXPEDITE

If EXPEDITE completes execution with a SESSION code that indicates an error (please read the EXPEDITE manual for information on this topic), then the communications server responds to it in one of the following ways:

- If the error occurred when sending data and points to an error in the data (error group "EDI errors" in Appendix B of the EXPEDITE manual), then the data to be sent is deleted from the workstation and an error status for the respective messages is reported to the host.
- In all other cases, the communications server assumes it is a network error. Four further attempts are made to complete the initiated data transmission using the EXPEDITE restart facility. If this does not succeed, the problem must be corrected by hand. This situation is indicated by "INACTIVE" in the column STATUS on the SAPCOMM screen window (see Working With the Graphical Interface [Seite 220]).

For more information, refer to the EXPEDITE documentation.

Error Messages in the EXP Log

The following section lists the error log entries specific to the communications component EXP and explains each of them. Explanations for general error log entries can be found in Notes on Testing and Problem Analysis [Seite 87]

E0206 IE response: Parameter <...> not found in <file>.
A problem occurred when interpreting file <file> created by EXPEDITE. The file is archived by the communications server. Send this file on floppy disk to the SAP Hotline.

E0207 Invalid response file line <...>
A problem occurred when interpreting the response file created by EXPEDITE. The file is archived by the communications server. Send this file on floppy disk to the SAP Hotline.

E0400 IE response: Command <Name> return code <...>
The function <Name> specified was not performed by the IBM Information Exchange System (remote target system, not local IBM program EXPEDITE). The EXPEDITE response file for the error message was archived. Examine the information contained therein to find out the exact cause of the error (Receiver not known...)

E0402 No connection with IE. Retry limit reached. Restart SAPcomm.
To continue data exchange with IBM Information Exchange, you must eliminate the problems or errors which may exist with the connection to IBM Information Exchange, and restart SAPcomm.
BC  The SAP Communications Server

The SAP communications server allows EDI, Mail and telecommunications documents to be exchanged between SAP Systems (R/2 and R/3) and external communication partners.

Basics
Hardware and Software Requirements
Installation
The Configuration File SAPCOMM.CFG
Configuration in the SAP System
The Control Component SCB
Connection to SAP Systems
Connection to ODEX
Connection to EXPEDITE
Connection to Soft*Switch
The Central Communications System [Seite 128]
Hardware and Software Requirements [Seite 129]
The Communications Component SSW [Seite 130]
CPI-C Configuration [Seite 131]
SSW-specific Entries in the SAPcomm Configuration File [Seite 132]
SAP-specific Aspects of Soft*Switch Operation [Seite 134]
Notes on Testing and Problem Analysis: Soft*Switch [Seite 135]
Connection to OSILINK X400
Connection to TOPCALL
Connection to GammaFax
Connection via APPLI/COM(T.611)
The Communications Component TST
The SAPcomm Development Library
Working With the Graphical Interface
The SAPcomm Directory Tree
The SAPcomm Auxiliary Programs

April 2001  127
The Central Communications System

With its product Soft*SwitchCentral, the company Soft*Switch provides the ability to exchange electronic post with mail systems from different producers: PROFS, ALL-IN-ONE, X.400, cc:Mail, MHS, UNIX mail and others.

It is based on an IBM mainframe program which runs under MVS or VM. It has a directory to convert addresses between the different connected systems and conversion programs which convert different file formats from one format into another. Furthermore, it contains communications modules to address mail systems via SNADS, PROFS or SMTP. Other mail systems are addressed with a dedicated gateway workstation (X.400, cc:Mail, Novell's MHS, ...). The SAPcomm server acts as a gateway for Soft*Switch.
Hardware and Software Requirements

Connection to Soft*Switch is currently only possible via an OS/2 workstation. This requires the communications software of OS/2 Version 2.1 (or above).

The Soft*Switch Central installation must include the SNAPI server module. Version V3L380P2 (or a newer version) is required for Soft*Switch Central.

The communications programs of the Soft*Switch product line require a separate CPI-C link between SAPcomm and Central. Since the SAP host connection via the communications component SAP also needs a CPI-C link, at least two LUs must be configured on the OS/2 workstation.

The following Soft*Switch manuals are required:
- Soft*Switch Central: Administration and Operation Guide (for MVS: Pub.No.1406, for VM Pub.No 1475)

To find the relevant paragraphs in the Soft*Switch documentation, you must know that SAPcomm is functionally a SNAPI client with P1 (i.e. gateway) from the viewpoint of Soft*Switch. LU6.2 is used as the transport means.

Notes on configuration of Communications Manager can be found in:
- SAP Communication: Configuration
- SNAPI Client Library/OS/2 Guide (Soft*Switch Pub.No 1903)
The Communications Component SSW

The connection of the communications server to the communications system Soft*Switch-Central is handled by the communications component SSW.

This communications component is used to exchange MAIL or EDI documents.

You must assign the “application type” to documents received via SSW, as well as a channel for further processing. The assignment determines whether the incoming documents are directed to the MAIL or EDI component of the SAP System.

For this, use the Routing statements. For information on Routing statements, refer to the section "The Routing Parameter Group [Seite 78]."

The communications component SSW communicates with Soft*Switch via a CPI-C link. This link is included when booting the component and remains that way until the component is shut down.

The individual messages and status confirmations are supplied via this link with the SNAPI log. If a message reaches Central, the following steps are carried out:

- The responsible mail system is determined.
- The message is converted, if necessary.
- The address is converted, if necessary.
- The message is forwarded to the relevant mail system.

Status messages are processed in the same way.
CPI-C Configuration

The communications component SSW must include a CPI-C link to Central. It accesses the Communications Manager directly. SAP network software is not used for this. The entries usually found in the sideinfo file are therefore made directly in the SAPcomm configuration file SAPCOMM.CFG.

As with all CPI-C links, the following elements must be configured (the names in *italics* are required later both in the SAPcomm configuration file SAPCOMM.CFG and in the Soft*Switch configuration member CDINPUT):

- A second LU *luname* must be defined in VTAM for the workstation.
- This LU *luname* must be configured in the Communications Manager.
- One partner LU must be configured in the Communications Manager. The "Fully qualified PLU name" and the "Uninterpreted PLU name" *pluname* must correspond to the VTAMAPPLID parameter in the ADDLOCALNODE command in the Soft*Switch configuration file CDINPUT. *luname* must be used as LU name for this PLU.
SSW-specific Entries in the SAPcomm Configuration File

SSW-specific Entries in the SAPcomm Configuration File

SSW is the identifier of the communications component responsible for the connection to the communications system Soft*Switch.

You must therefore include the following parameter set in the configuration file SAPCOMM.CFG:

```latex
PROGRAM SSW
  <Parameter specification>
END-PROGRAM
```

The parameters of this group are explained below. A general description of the PROGRAM parameter group can be found in Structure of the Configuration File [Seite 37].

PERFORM Parameter Group

No statements need to be entered in the configuration file SAPCOMM.CFG for communications component SSW in order to execute Soft*Switch-specific administrative actions.

DESTINATION Parameter Group

No DESTINATION parameter sets need to be entered in the configuration file SAPCOMM.CFG for the communications component SSW.

SELECTION Parameter Group

No SELECTION parameter sets need to be entered for the communications component SSW.

CHANNEL Parameter Group

No CHANNEL parameters

Component-specific PROGRAM parameters

You can include the following parameters in the PROGRAM parameter set SSW:

- **CLIENTRGN nodergn**
  Optional. You can specify the 'Routing Group Name' of the SSW component from the viewpoint of Soft*Switch using `nodergn`. `nodergn` must have the same value as the NODERGN parameter in the ADDREMOTENODE command in the Soft*Switch configuration file CDINPUT.

- **CLIENTREN noderen**
  Mandatory. You enter the 'Routing Element Name' of the SSW component from the point of view of Soft*Switch using `noderen`. `noderen` must have the same value as the NODEREN parameter in the ADDREMOTENODE command in the Soft*Switch configuration file CDINPUT.

- **CLIENTDGN dgn**
  Mandatory. You enter the 'Distribution Group Name' of the SSW component from the point of view of Soft*Switch using `dgn`. `dgn` must have the same value as the DGN parameter in the ADDDGNINFO command in the Soft*Switch configuration file CDINPUT.

- **ACBNAME luname**
  Mandatory. You enter the name of the 'Logical Unit' with which the SSW component will communicate with Soft*Switch using `luname`. `luname` must match the values of the following parameters:
- LUNAME in the ADDREMOTENODE command in the Soft*Switch configuration file CDINPUT
- ACBNAME of the VTAM definition of the workstation
- Logical Unit in the configuration of the Communications Manager
- LU according to the definition of the Partner Logical Unit in the configuration of the Communications Manager

**MODENAME**   modename

Mandatory. You enter the 'Mode Table Entry' that is used in the VTAM definition of ACBNAME using modename. modename must have the same value as the MODENAME parameter in the ADDREMOTENODE command in the Soft*Switch configuration file CDINPUT.

**PLUNAME**   pluname

Mandatory. You enter the name of the partner LU with which the SSW component will communicate with Soft*Switch using pluname. pluname must have the same value as the VTAMAPPLID parameter in the ADDLOCALNODE command in the Soft*Switch configuration file CDINPUT.

**PASSWORD**   snapipassword

You enter the password with which the communications component SSW will log onto Central using snapipassword. snapipassword must have the same value as the SNAPIPASSWORD parameter in the ADDREMOTENODE command in the Soft*Switch configuration file CDINPUT.

**DEBUG_MSK**   mask

Optional. The SNAPI Client Library used in the communications component SSW can generate a large number of log entries. This log information is divided into the same classes as in the central log (see Administration and Operations Guide). Each class is triggered via a bit. mask is determined by adding the following decimal values:

1  ERROR       Error
2  AUDIT       Message processing
4  SHOWBUF     Hexadecimal dump of the sent and received data
16 DEBUG       Development
32 GENERAL     Process activity
64 WARNING     Warnings

ERROR, WARNING, GENERAL and AUDIT are activated if this parameter is omitted. This corresponds to a value of 99 for mask. It is sufficient for productive operation. During the configuration, DEBUG and SHOWBUF can be activated. Activating the DEBUG and SHOWBUF traces during productive operation is not recommended.
SAP-specific Aspects of Soft*Switch Operation

Bear in mind the following notes when configuring Soft*Switch Central:

- The CPI-C link is always established from the workstation. The parameter AUTOACQUIRE in the Soft*Switch configuration file CDINPUT must be set to N.
- The R/MAIL and SAPmail users can only be reached via their X.400 addresses. This means that either all of these users must be entered into the central directory or the Soft*Switch-AUTOREGISTRATION option for the SAPcomm-Client must be activated for the registration of these users.
- R/MAIL and SAPmail only support pure ASCII and binary files. Consequently, only ASCI and PCDATA are possible as a DOCTYPE.
- **EDI Incoming Documents:**
The recipients of EDI incoming documents must be configured in Soft*Switch as X400 recipients. The routing statements for these recipients in the SAPcomm configuration file must - in addition to the TO CHANNEL statement - contain the APPLICATION EDI statement, or the respective EDI incoming documents will be handled as Mail incoming documents.

The "communications service" allocated to the incoming documents (see the section on the communications component SAP) is "SSW". It cannot be changed.

Example CDINPUT file for the NDBATCH program for the configuration of Central:

```plaintext
ADDDGINFO NODEREN=SCMISSW, *same value as in ADDREMOTENODE
  DGN=SCMISSW, *same value as in ADDREMOTENODE
  DEN=*, *fixed value
  DOCTYPE1=ASCI, *fixed value
  DOCTYPE2=PCDATA *if desired
```

```plaintext
ADDDGINFO NODEREN=SCMISSW, *same value as in ADDREMOTENODE
  DGN=SCMISSW, *same value as in ADDREMOTENODE
  DEN=*, *fixed value
  DOCTYPE1=ASCI, *fixed value
  DOCTYPE2=PCDATA *if desired
```

```plaintext
ADDREMOTENODE NODEREN=SCMISSW, *same value as in sapcomm.cfg
  GATEWAYTYPE=SNAPI, *fixed value
  NAMEOPTION=SNADS, *fixed value
  SNAPIPASSWORD=PASS, *same value as in sapcomm.cfg
  SENDERAUTOREG=Y, *Y or N possible
  AUTOACQUIRE=N, *fixed value
  LUNAME=LTRIC13, *same value as in sapcomm.cfg
  MODENAME=LU62TST1, *same value as in sapcomm.cfg
  SENDERUSIZE=2048,
  SENDPACING=3,
  RECEIVEUSIZE=2048
  RECEIVEPACING=3,
  SENDUSIZE=2048
  RECVUSIZE=2048
  GATEWAYTYPE=SNAPI, *fixed value
  NAMEOPTION=SNADS, *fixed value
  SNAPIPASSWORD=PASS, *same value as in sapcomm.cfg
  SENDERAUTOREG=Y, *Y or N possible
  AUTOACQUIRE=N, *fixed value
  LUNAME=LTRIC13, *same value as in sapcomm.cfg
  MODENAME=LU62TST1, *same value as in sapcomm.cfg
  SENDERUSIZE=2048,
  SENDPACING=3,
  RECEIVEUSIZE=2048
  RECEIVEPACING=3,
  SENDUSIZE=2048
  RECVUSIZE=2048
```

Example CDINPUT file for the NDBATCH program for the configuration of Central:
Notes on Testing and Problem Analysis: Soft*Switch

Problems which occur during the interaction of the communications server and Soft*Switch normally belong to one of the following categories:

- CPI-C link cannot be set up.
- Messages cannot be sent.
- An SAP user cannot be reached by an external system.

Measures for eliminating errors in these three categories are explained below.

**CPI-C Link Cannot be set up**

If this is the case, carry out the following measures:

- Make sure that Central is running (in ROSCOE, TSO,...)!
- Make sure that the SNA coupling LAN <-> HOST is functioning (in the Communications Manager under Message or Status).
- Check the configuration: VTAM, Communications Manager, SAPCOMM.CFG, Central CDINPUT. The specified values must agree.

**Messages Cannot be Sent**

If the errors E1101.....API069E occur in the log file of the communications component SSW or error API069E in the central log, then there is a problem with the sender.

You will find other entries in the central log file that make it easier to assign them to one of these cases below:

- AUTOREGISTRATION is switched off (value 'N').
  Please register the user in the Soft*Switch directory (with NDBATCH or by MAIL to NDADMIN).
- The LASTNAME of the user was not entered.
  Please maintain the external addresses of the users in SAP. The external address must contain a LASTNAME (see R/2 document S09.2)
- There is already a user with this NATIVE address and this LASTNAME in the Central Directory but with other X.400 fields.
  The reason for this is that the external address of the user or the external address of the user $$SYSTEM was changed without performing a directory update.
  Please update the directories. You can either enter the SAP external address (see R/2 document S09.2) or maintain the Soft*Switch directory (with NDBATCH or by mail to NDADMIN).
- The SAP user has no authorization to send messages to a certain mail system or to a certain mail user.
  The security option of Soft*Switch is active (Central Access Control Support).
  Please check the configuration of Central using Administration and Operations Guide.
  If the recipient can be reached via X.400, the configuration or security settings of the X.400 gateway can also be relevant: X.400 Gateway Planning, Installation, and Operations Guide and Installing Connections to Other MTAS.
- The Soft*Switch configuration does not match the SAPcomm configuration.
  Switch on the DEBUG and SHOWBUF parameters in SAPCOMM.CFG and in Central.
  Send relevant sections from CDINPUT, SAPCOMM.CFG and the log files from Central and the workstation to the SAP or Soft*Switch Hotline.
SAP-specific Aspects of Soft*Switch Operation

SAP User Cannot be Reached by External System

In such a case, carry out the following measures:

- The SAP user is not known in the Soft*Switch directory, or the user is known, but his external address does not match that in the directory.
  
  Both are diagnosed by checking the Soft*Switch directory (with NDBATCH or by mail at NDQUERY). Please update the directories. You can either maintain the SAP external address (guide S09.2) or maintain the Soft*Switch directory (with NDBATCH or by mail to NDADMIN).

- The external sender has no authorization to send messages to the SAP System.
  
  The security mechanisms of Soft*Switch are active (Central Access Control Support). Please check the configuration of Central using the \textit{Administration and Operations Guide}. If the sender is an X.400 user, the configuration or security settings of the X.400 gateway can also be relevant: \textit{X.400 Gateway Planning, Installation, Operations Guide} and \textit{Installing Connections to Other MTAS}.

- The Soft*Switch configuration does not match the SAPcomm configuration.
  
  Activate the DEBUG and SHOWBUF parameters in SAPCOMM.CFG and in Central. Send the relevant sections from CDINPUT, SAPCOMM.CFG and the log files from Central and the communications component SSW to the SAP or Soft*Switch Hotline.

General Notes

Finally, two more notes that will support your work with Soft*Switch:

- When problems occur, always check the central log file on the host and the log file of the communications component SSW on the workstation.

- There are two forms in R/MAIL which support queries to NDQUERY and NDADMIN to assist in maintaining or modifying entries in the Soft*Switch directory. These forms generate messages in the formats described in the \textit{Users' Guide to Services}. Component 180 "Forms" is required in order to use forms in the SAP System! Carry out the following steps:
  
  - Maintain the external addresses of NDADMIN and NDQUERY in the SAP R/MAIL System.
  
  - Enter these addresses in TMLFO for the SSWF forms.
  
  - Select one of these forms in Mail (OK: nmail and then PF11).
Connection to OSILINK X400

The OSILINK X.400 Communications System [Seite 138]

Hardware and Software Requirements for OSILINK X.400 [Seite 139]

The CX4 Communications Component [Seite 140]

CX4-specific Entries in the SAPcomm Configuration File [Seite 141]

SAP-specific Aspects of OSILINK X.400 Operation [Seite 142]

Notes on Testing and Problem Analysis: OSILINK X.400 [Seite 143]
The OSILINK X.400 Communications System

OSILINK X.400 is a complete message handling system based on the CCITT standard X.400(88). It includes the components Message Transfer Agent (MTA), User Agent (UA) and Message Store (MS).

It is produced by CoCoNet, Computer Communication Networks GmbH, Duesseldorf, Germany.
Hardware and Software Requirements for OSILINK X.400

OSILINK X.400 is implemented according to the X/Open standards for UNIX System V, and therefore requires the UNIX version of the SAP communications server.

OSILINK X.400 provides communications facilities both in WANs (via X.25 or ISDN) and LANs (via TCP/IP). The system is generally set up with an existing transfer interface (XTI). Technical network aspects are not covered here. Please consult CoCoNet directly if you have any questions.
The CX4 Communications Component

Connection of the communications server to the communications system OSILINK X.400 is handled by the communications component CX4.

The communications component represents an application (Message Transfer Client) as far as the OSILINK X.400 MTA is concerned. It exchanges messages and status information with the MTA via an interface (MTAPI).
CX4-specific Entries in the SAPcomm Configuration File

CX4 is the identifier of the communications component responsible for the connection to the communications system OSILINK X.400.

You must therefore include the following parameter set in the configuration file SAPCOMM.CFG:

```
PROGRAM CX4
  <Parameter specification>
END-PROGRAM
```

The parameters of this group are explained below. A general description of the PROGRAM parameter group can be found in Structure of the Configuration File [Seite 37].

PERFORM Parameter Group

No statements need to be entered in the configuration file SAPCOMM.CFG for the communications component CX4 to execute OSILINK X.400-specific administrative actions.

DESTINATION Parameter Group

No DESTINATION parameter sets need to be entered in the configuration file SAPCOMM.CFG for the communications component CX4.

SELECTION Parameter Group

No SELECTION parameter sets need to be entered for the communications component CX4.

CHANNEL Parameter Group

No OSILINK X.400-specific parameters need to be entered in the CHANNEL parameter sets that you create for the communications component CX4.

Component-specific PROGRAM Parameters

You must include the following parameters into the PROGRAM parameter set CX4:

- **MAIL_DIR** `<MTA root directory>`
  Mandatory. This parameter must contain the path specification for the installation directory of the OSILINK X.400 MTA. It corresponds to the environment variable of the OSILINK X.400 MTA with the same name and must be set to the same value!

- **MTC_NAME** `<MTC name>`
  Mandatory. This parameter must contain the name of the application (Message Transfer Client) under which the SAP communications server is managed in the OSILINK X.400 MTA. This name must also be maintained in the configuration of the OSILINK X.400 MTA under "Available MTCs".
SAP-specific Aspects of OSILINK X.400 Operation

Bear in mind the following notes when installing and configuring OSILINK X.400:

- A separate application (Message Transfer Client) must be configured in the OSILINK X.400 MTA for the SAP communications server.

Proceed as follows:

1. Enter a (user defined) name in the administration program (mta.adm) using the function "Configure local MTA" / "More" under "Available MTCs".

2. To receive messages, you define all X.400 addresses that are to be allocated to the SAP communications server and enter in turn the respective application names under "MTC/GWC name" using the Function "Managing users".

3. When booting OSILINK X.400 MTA, you must start a separate receiver process for this application via the command:

   lrec $MAIL_DIR <MTC name>

- **Incoming EDI documents**: The "communications service" allocated to the incoming documents is "CX400". It cannot be changed.

For more information on communications services, refer to The Communications Component SAP [Seite 92].
Notes on Testing and Problem Analysis: OSILINK X.400

Problems arising during the interaction of the communications server and OSILINK X.400 can be divided into two categories:

- **No link from communications component CX4 to OSILINK X.400 MTA.**
  This is mostly caused by incorrect installation or configuration of the OSILINK X.400 application for SAPcomm:
  - Parameter MAIL_DIR or MTC_NAME maintained incorrectly
  - Application (MTC) not created in OSILINK X.400 MTA
  - OSILINK X.400 Receiver Process (lrec) not started for the application
  - X.400 recipient address not allocated to the SAPcomm application

- **No sending or receiving of messages via OSILINK X.400 MTA.**
  This is mostly caused by incorrect installation or configuration of the OSILINK X.400 MTA or the subordinate transport layer:
  - Partner MTAs not or incorrectly maintained in OSILINK X.400 MTA or in the transport layer
  - Incorrect or incomplete routing information in the OSILINK X.400 MTA

For problems in the second category, please see your OSILINK X.400 MTA guide for advice or consult CoCoNet directly.

CoCoNet also offers administrator courses covering this topic. These courses are recommended for your OSILINK X.400 MTA support staff.

**Error Entries in the CX4 Log**

The following section lists error log entries specific to the communications component CX4 and explains each of them.

A description of general error log entries can be found in [Notes on Testing and Problem Analysis][1] [Seite 87].

- **E1101**  Missing Parameter <par>
  The specified parameter in the program group CX4 is missing from the configuration file.

- **E1102**  Init of MTAPI failed, RC=<err>
  The start-up of the MTA interface was rejected with error code <err>.

- **E1301**  Service <serv> not supported
  A message for a communications service other than X.400 was transferred to component CX4.

- **E1351**  Creation of new message failed, RC=<err>
  The creation of a new message was rejected by the MTA interface with error code <err>.

- **E1352**  Error <err> setting attribute <nr> to <text>
- **E1353**  Error <err> setting attribute <nr> to <wert>
- **E1354**  Error <err> setting originator to <addr>
- **E1355**  Error <err> setting recipient to <addr>
- **E1356**  Error <err> setting body (type <nr>) to <file>
  The setting of the specified feature for the current message was rejected by the MTA interface with error code <err>.

- **E1358**  Submission of message failed, RC=<err>
Notes on Testing and Problem Analysis: OSILINK X.400

Transfer of the current message was rejected by the MTA interface with error code <err>.

**E1359  Deletion of message failed, RC=<err>**
Deletion of the current message to be sent was rejected by the MTA interface with error code <err>.

**E1371  Address too long adding <name>=<text>**
The entered address element could no longer be added to an O/R address.

**E1372  Invalid format of address part <name>**
The entered address element of an O/R address has an invalid format.

**E1373  Address part <name> too long**
The entered address element of an O/R address is too long.

**E1501  Receipt of message failed, RC=<err>**
Retrieval of a received message was rejected by the MTA interface with error code <err>.

**E1502  Invalid content type <nr>, message destroyed**
Message contents of the category <nr> were received. This category is not supported by component CX4. The message was not processed.

**E1509  Deletion of message failed, RC=<err>**
Deletion of the current incoming message was rejected by the MTA interface with error code <err>.

**E1552  Error <err> reading attribute <nr>**
**E1554  Error <err> reading originator**
**E1555  Error <err> reading recipient**
**E1556  Error <err> reading body part**
Reading the entered feature for the current message was rejected by the MTA interface with error code <err>.

**E1601  Invalid body part type <nr>**
A message part of the category <nr> was received. This category is not supported by component CX4. The category was changed to "binary".
Connection to TOPCALL

The TOPCALL Communications System [Seite 146]
Hardware and Software Requirements for TOPCALL [Seite 147]
The Communications Component TOP [Seite 148]
TOP-specific Entries in the SAPcomm Configuration File [Seite 149]
SAP-specific Aspects of TOPCALL Operation [Seite 151]
Notes on Testing and Problem Analysis: TOPCALL [Seite 152]
The TOPCALL Communications System

TOPCALL is an independent computer system that allows different DP systems to access the telecommunications services telefax, telex and teletex. The system is constructed modularly and can contain different configuration levels: the type and number of post office lines and DP systems that can be connected can be configured relatively freely.

It is produced by TOPCALL INTERNATIONAL, Vienna, Austria. Sales and distribution is handled by numerous subsidiaries/sales partners worldwide.
Hardware and Software Requirements for TOPCALL

The workstation on which SAPcomm runs (UNIX or OS/2) requires a free V.24 interface for the connection to TOPCALL.

The TOPCALL system requires an asynchronous V.24 interface for the connection to SAPcomm.

The version of the TOPCALL operating system TCOSS must be 5.22 or higher.

💡

The TOPCALL communications system connection described here, via V.24 connection using the communications component TOP, is an old procedure which is still supported, but no longer recommended.

A more recent procedure for TOPCALL connection can be implemented using the communications component APP via the APPLI/COM interface, see Connection via APPLI/COM(T.611) [Seite 160]. Only using this type of connection is it possible to receive faxes, or to send them in postscript format.
The Communications Component TOP

The connection of the SAP communications server to TOPCALL is handled by the communications component TOP. Data exchange between the communications component TOP and the TOPCALL system is performed via a serial asynchronous V.24 line. The communications component TOP transfers messages to be sent to TOPCALL and retrieves the messages held in TOPCALL at regular intervals.

The TOPCALL communications system connection described here, via V.24 connection using the communications component TOP, is an old procedure which is still supported, but no longer recommended.

A more recent procedure for TOPCALL connection can be implemented using the communications component APP via the APPLI/COM interface, see Connection via APPLI/COM(T.611) [Seite 160]. Only using this type of connection is it possible to receive faxes, or to send them in postscript format.
TOP-specific Entries in the SAPcomm Configuration File

TOP is the identifier of the communications component responsible for the connection to the communications system TOPCALL.

You must therefore include the following parameter set in the configuration file SAPCOMM.CFG:

```
PROGRAM TOP
  <Parameter specification>
END-PROGRAM
```

The parameters of this group are explained below. A general description of the PROGRAM parameter group can be found in [Structure of the Configuration File][Seite 37].

**PERFORM Parameter Group**

No statements need to be entered in the configuration file SAPCOMM.CFG for the communications component TOP to execute TOPCALL-specific administrative actions.

**DESTINATION Parameter Group**

No DESTINATION parameter sets need to be entered in the configuration file SAPCOMM.CFG for the communications component TOP.

**SELECTION Parameter Group**

No SELECTION parameter sets need to be entered for the communications component TOP.

**CHANNEL Parameter Group**

The CHANNEL parameter sets that you create for the communications component TOP can contain the parameters explained in the following topic:

- **TCCHANNEL** `<TOPCALL channel name>`
  
  Optional. You determine the logical channel to which messages are to be sent in TOPCALL using this parameter. If the parameter is not entered, the messages are sent to the following TOPCALL channels depending on the telecommunications service:
  
  - for telefax: F:
  - for telex: X:
  - for teletex: T:

- **MASK** `<TOPCALL mask>`
  
  Optional. You determine the screen that is to be used for the dispatch of messages in TOPCALL using this parameter. If the parameter is not entered, the standard screen set for the corresponding TOPCALL channel is used.

**Component-specific PROGRAM Parameters**

You can include the following parameters in the PROGRAM parameter set TOP:

- **DRIVE** `<TOPCALL drive>`
  
  Optional. You specify here the drive on which the messages are to be stored in TOPCALL. If this parameter is not specified, drive B is the default.

- **SERVER** `<Server abbreviation>`
  
  Optional. This parameter determines the first character of the file name under which messages are stored in TOPCALL. As a result, several SAP communications servers can be connected to a TOPCALL. If the parameter is not entered, the TOPCALL files start with "0".

- **PORT** `<SAPcomm V.24 port>`
TOP-specific Entries in the SAPcomm Configuration File

Optional. You specify here the UNIX device name of the V.24 port used by SAPcomm on this computer. If the parameter is not entered, "/dev/tty00" is used.

**BAUD**  
<V.24 baud rate>

Optional. You enter here the transmission speed to be used for TOPCALL on the V.24 line.
Permissible values are: 1200, 2400, 4800, 9600 and 19200.
If the parameter is not entered, 9600 baud is used.
SAP-specific Aspects of TOPCALL Operation

Bear in mind the following notes when installing and configuring TOPCALL:

- The asynchronous V.24 interface to TOPCALL must be configured with the following transmission parameters:
  - Transmission speed must correspond to the BAUD parameter
  - Even parity
  - 8 data bits
  - 2 stop bits
  - XON/XOFF log
- The logical TOPCALL channel via which SAPcomm is connected must be configured with active confirmations; the TOPCALL standard screen +Y must be used for this purpose.
  To speed up the data transfer, SAP recommends you to remove the variable for the message text ($X$) from the TOPCALL screen +Y.

Operating TOPCALL

The communications component TOP carries out the following actions:

1. Sending messages:
   - The message is created in a file D:S<N> in TOPCALL.
     This is:
     D   TOPCALL drive (parameter DRIVE)
     S   SAPcomm abbrev. (parameter SERVER)
     <N> internal message no. (assigned by SAPcomm)
     The TOPCALL author field is currently always filled with the value "SAPcomm".
   - A transmission request with the number specification N=KK:MM<N> is sent to TOPCALL:
     KK   TOPCALL channel (parameter TCCHANNEL)
     MM   TOPCALL mask (parameter MASK)
     <N>  Recipient's number (fax,telex,teletex).

2. Receiving messages and confirmations:
   Incoming messages and/or status confirmations are requested from TOPCALL by the RETRIEVE command, and then received, evaluated, and finally confirmed.
Notes on Testing and Problem Analysis: TOPCALL

Problems that occur during the interaction of the communications server and TOPCALL (messages not sent, confirmed or received) can normally be traced back to an interrupted V.24 link.

In this case, it is advisable to check the communication with auxiliary program scmasyn. To do this, close down the SAP communications server and at the UNIX command level enter:

```
scmasyn .i
```

If the link is in order, then after a short transmission time you receive the TOPCALL table of contents displayed on the screen.

Error Entries in the TOP Log

The following section lists error log entries specific to the communications component TOP and explains each of them.

Explanations for general error log entries can be found in Notes on Testing and Problem Analysis [Seite 87].

E1001 Error opening <port>, errno=<..>
Error when accessing the UNIX device file <port>. The value errno specifies the cause - for example: port does not exist.

E1002 Error resetting serial port, errno=<..>
Error during the attempt to access the V.24 interface in reading mode or in writing mode. The value errno specifies the cause - for example: missing authorizations.

E1003 Error resetting TOPCALL, rc=<..>
No or incorrect response from the TOPCALL box to a reset command.

E1004 Error closing serial port, errno=<..>
Error when closing the V.24 interface. The value errno specifies the cause.

I1101 <TOPCALL transmission command>
Transmission request to the TOPCALL box.

E1102 Error sending to TOPCALL, rc=<..>, errno=<..>
No or incorrect response from the TOPCALL box to a send command.

I1201 <TOPCALL incoming transfer file>
File name of one of the confirmations or incoming messages received by the TOPCALL box.

E1202 Error receiving from TOPCALL, rc=<..>, errno=<..>
No or incorrect response from the TOPCALL box to a fetch command.

I1301 Confirm last query to TOPCALL
Confirmation of the last incoming TOPCALL transfer file (log entry x1201).

I1302 Delete <file> in TOPCALL
Deletion of the message to be sent stored under <file> in the TOPCALL box.
Connection to GammaFax

The GammaFax Communications System [Seite 154]
Hardware and Software Requirements [Seite 155]
The GAM Communications Components [Seite 156]
GAM-specific Entries in the SAPcomm Configuration Files [Seite 157]
Installation [Seite 159]
Connection to GammaFax

The GammaFax Communications System
GammaFax is a PC product for sending and receiving faxes. It consists of hardware (fax cards, up to 16 of which can be mounted in a PC), and associated software for DOS and OS/2. It is manufactured by the firm DIALOGIC.
Hardware and Software Requirements
For GammaFax connection, you require the OS/2 version of the SAP communications server.
Hardware: From 1 to 16 GammaFax CP boards as required, and the corresponding number of telephone connections
Software: GammaFax for OS/2 5.1
The GAM Communications Component

The connection of the communications server to GammaFax is handled by the GAM communications component.

It places transmission requests in the "pending list" of the GammaFax queue, and analyzes the "sent list" of the GammaFax queue in order to receive status confirmations. Faxes received are taken out of the "received list" of the GammaFax queue and forwarded to an SAP System (from R/3 3.0).

The GammaFax queue manager, which transmits data between the GammaFax queue and the GammaFax cards, must be started parallel to the SAP communications server.

The GammaFax menu interface allows you to display, parallel to this and at all times, the contents of the individual lists of the GammaFax queue, as well as send faxes manually or from other applications.

The SAP communications server is able to recognize the transmission requests it creates from the GammaFax User ID, which has the following character string:

XXXiiiiiiSSScccNNNNNuuuuuuuuuuuu

The elements of the character string have the following meaning:

XXX Any SAPcomm selection made by the user, (see parameter SERVER below)
     Default value: SAP
ii.. Serial SAPcomm message number
SSS SAP System, from which the message is coming
ccc The client in this system
NN.. Spool number in the SAP System
uu.. The SAP user name of the message sender
GAM-specific Entries in the SAPcomm Configuration File

The CHANNEL Parameter Group:

**DIAL_PREFIX**  **<Dial string>**

Optional. Here you can specify a string of digits or special characters that you dial before the fax number of the recipient. You can, for example, set the exchange number for a private branch exchange (usually “0”).

**RETRY_COUNTER**  **<Dial retries>**

Optional. Here you can set the maximum number of dial attempts that the GammaFax software is to perform for each message. If this parameter is not specified, the number of dial attempts is three.

**RETRY_DELAY**  **<Dial interval>**

Optional. This parameter specifies the time allowed (in minutes) before a new dial attempt is made. If the parameter is not specified, the interval between two consecutive dial attempts is 15 minutes.

**CSID**  **<Sender ID>**

Optional. When the fax is transmitted, the character string defined in this parameter (max. 20 characters long) is sent to the recipient as the sender ID. The recipient can then, for example, print it out in the fax header line or keep it in his/her transmission log.

**CD_TIMEOUT**  **<Wait for response time>**

Optional. This parameter specifies the maximum wait time (in seconds) for the response tone of the recipient’s fax machine, after the number has been dialled. If the parameter is not specified, the maximum wait time for connection setup is 45 seconds.

**COVER_PAGE**  **<Cover page file>**

Optional. Here you can specify the name of a file to be used as the GammaFax cover page.

**HEADER_LINE**  **<Header line>**

Optional. The character string defined in this parameter (max. 20 characters long) is output on the header line of each page when the fax is transmitted. If this parameter is not specified, the GammaFax standard header line is used.

**MODEM_ID**  **<Fax card>**

Optional. You can specify in this parameter that faxes sent via this channel are forwarded to a particular GammaFax card (if there are several mounted). You can also specify, for example, the telephone line used, as well as other fax card-specific parameters (font, ECM mode etc.). The fax card description must match the description in the GammaFax configuration file (GFAX.$DC). (Parameter CHANNEL, e.g. GFAX1.1 GFAX1.2 GFAX1.3 etc.) If this parameter is not specified, the next fax card available is used to dispatch this message.
GAM-specific Entries in the SAPcomm Configuration File

Component-specific PROGRAM parameters:

**SERVER**  
<Server abbreviation>  
Optional. This parameter defines the first three characters of the user ID, under which transmission requests are placed in the GammaFax queue. This enables several instances of the GAM component to find the messages that belong to them, and therefore access a common GammaFax installation. If the parameter is not specified, the character string “SAP” is used.

**LOGO_DIRECTORY**  
<directory>  
Optional. This parameter contains the path specification for a directory containing graphic files in TIFF format. They can be inserted as logos in the outgoing faxes of each sender.

**SIGN_DIRECTORY**  
<directory>  
Optional. This parameter contains the path specification for a directory containing graphic files in TIFF format. They can be inserted as captions in outgoing faxes. The sender can only access a file if its name corresponds to his/her SAP user name.
Installation

First install the hardware (mount the GammaFax CP boards consulting the GammaFax documentation).

Then install the GammaFax for OS/2 software (min. 5.1). The installation program is a DOS program: Depending on the configuration of your DOS windows under OS/2, it is sometimes recommendable to boot the PC with a DOS diskette, and to restart OS/2 after successful installation.

We recommend that you execute the automatic identification of the mounted fax cards - that is offered during installation - so that the correct type of GammaFax CP board is used.

Answer the question regarding background operation with "No"; The GammaFax queue manager runs as a separate process under OS/2.

After you have finished the installation, perform the following tasks:

- In the CONFIG.SYS file (not AUTOEXEC.BAT !!), set the environment variable GFAX to the path of the GammaFax directory.
- Check the path specification contained in the command files GFQM.CMD and GF.CMD in the GammaFax directory. Correct it if necessary or remove the statement SET GFAX=... entirely.
- Configure the start of the GammaFax queue manager when starting the PC by adding the statement `START %GFAX%GFQM` to the file STARTUP.CMD.

Restart the PC and test whether the installation is correct by dispatching a fax using the GammaFax menu interface (GF). The transmission request must disappear from the "pending list" and appear in the "sent list" with status 0000.

Install the SAP communications software (if not yet available on the PC).

Add a new group to the configuration file:

```
PROGRAM GAM
...
END-PROGRAM
```

Fill this group with the usual parameters and parameter groups (CHANNEL, PERFORM, ACTION, PERIODIC, ..), as well as the GammaFax-specific parameters required (see list above).

Program SCMIGAM.EXE, which belongs to the GAM component, uses two dynamic GammaFax libraries (GPIDLL.DLL and CEPPORTW.DLL). They are installed in directory `/sap_ws/dll`, together with the other SAPcomm programs.

Fax documents may contain formatting instructions (e.g. page break). They begin with the SAP Escape sign (hexadecimal 1C). The character set conversion file of the OS/2 Communications Manager sometimes has to be adapted so that this sign can reach the GAM component from R/2 Systems unchanged. (You can use your own conversion files with parameters CONVERT and DCONVERT in the SAP component, see Connection to SAP Systems [Seite 89].)

In the EBCDIC/ASCII conversion file, the first 16 lines define the conversion from ASCII to EBCDIC, the following 16 lines define the conversion from EBCDIC to ASCII. In this second section, set the position for 1C (line 18, columns 25+26) to "1C".

April 2001  159
Connection via APPLI/COM(T.611)

The File Interface APPLI/COM (T.611) [Seite 161]

Hardware and Software Requirements [Seite 162]

Communications Component APP [Seite 163]

Installation [Seite 164]

APP-specific Entries in SAPcomm Configuration File [Seite 165]

Interface Specification [Seite 167]
The APPLI/COM interface was brought out in 1990 by Deutsche Bundespost TELEKOM and was adopted by the ITU-T (then known as CCITT) in revised form in 1992 as Recommendation T.611. The interface defines a procedure for transferring messages between a “Local Application” (hereinafter referred to as LA) and a “Communication Application” (hereinafter referred to as CA). The LA is the source/destination of documents, the CA being responsible for sending and receiving documents via various electronic communications services.

The SAP communications server can be implemented as an LA. Various communications systems from different manufacturers can hence be used as CA, and connected to SAPcomm. The operation of the APPLI/COM interface via SAPcomm is, however, only partially supported by the ITU-T recommendation. It does not meet all the points of the specification and also has some SAP-specific characteristics. Therefore, not all systems conforming to T.611 available on the market can be used as CA, but only those which conform to the SAP-specific characteristics. There is a full description of the SAP version of the APPLI/COM interface at the end of this chapter under Interface Specification [Seite 167].

Here is a selection of systems which support the SAP version of the APPLI/COM interface. (The list is not complete. Please refer to the manufacturer of the relevant communications system, as required.)

- **CAESAR** CAE ELECTRONICS, Stolberg, Germany
- **FAXINATION** FENESTRAE, Leidschendam, Netherlands
- **NVS** PP-COM, Alsdorf, Germany
- **TOPCALL** TOPCALL INTERNATIONAL, Vienna, Austria
- **TWINFAX** MPS SOFTWARE, Kirchheim, Germany

The communications system TOPCALL can also be connected to SAPcomm via an asynchronous serial link (V.24), instead of via the APPLI/COM interface, see Connection to TOPCALL [Seite 145].

This older connection procedure is still supported, but no longer recommended. Only with the newer procedure via the APPLI/COM interface, can faxes be received and also sent in postscript format.

In the past it was also possible to connect the CAESAR communications system to SAPcomm via the communications component CAE. This component is no longer supported. From now on, only use the component APP described below.
Hardware and Software Requirements

You can use the OS/2 version or the UNIX version of the SAP communications server for connecting communications systems via the APPLI/COM interface.

The SAP communications server (LA) and the communications system (CA) can be used on the same computer or on separate computers (even with different operating systems). Messages are exchanged between LA and CA via files in shared directories. The directories can be created on the LA computer or the CA computer or a file server. When LA and CA are installed on different computers, it must therefore be possible to access via NFS.

An additional module may be necessary for the communications system to support the APPLI/COM interface and/or handle SAP-specific characteristics. For more information, please refer to the respective system manufacturer.
Communications Component APP

The operation of the APPLI/COM interface via the SAP communication server is assumed by the communications component APP, enabling fax and telex messages to be sent and received. The communications component APP works as LA as far as APPLI/COM is concerned. It can serve one or more CAs simultaneously.

A set of directories is required per CA for message exchange between LA and CA. These directories must have read and write access from LA and CA. The following directories are required:

- **COM_JOB** For the storage of transmission requests to be processed.
- **COM_ACK** For the storage of processed transmission requests and reception notifications (= reception requests).
- **COM_ERR** For the storage of incorrect transmission requests.
- **COM_OUT** For the contents of messages to be sent.
- **COM_IN** For the contents of messages received.
Installation

Requirements
First establish the file system for the shared directories between LA and CA and the names of the directories. Create the directories.
It is recommended (but not absolutely mandatory) to create the five directories required per CA below a higher-level directory.
If LA and CA are to run on different computers, ensure that the shared directories have read and write access from both computers. (It may be necessary to install NFS software for this).

Communications system
Install the communications system (CA) including any necessary additional modules for supporting the APPLI/COM interface and/or handling the SAP-specific characteristics. For more information, please refer to the respective system manufacturer.

SAPcomm
Install the SAP communications server, if not already available.
Create a new group in the configuration file:

```plaintext
  PROGRAM   APP

  . .

  END-PROGRAM
```

Fill this group with the usual parameters and parameter groups (CHANNEL, PERFORM, ACTION, PERIODIC, etc), and the desired APPLI/COM-specific parameters, see [APP-specific Entries in SAPcomm Configuration File](#) .

Fax documents may contain formatting instructions (e.g. page break). These start with the SAP-Escape character (hexadecimal 1C). It may be necessary to modify the character set conversion file of the OS/2 Communication Manager for this to get from R/2 systems to the component APP. (With the parameters CONVERT and DCONVERT in the SAP component, you can use your own conversion files, see [Connection to SAP Systems](#) .)

The first 16 lines in the EBCDIC/ASCII conversion file stipulate the conversion from ASCII to EBCDIC, the next 16 lines the conversion from EBCDIC to ASCII. In this second section, set the position for 1C (line 18, columns 25+26) to "1C". 
APP-specific Entries in SAPcomm Configuration File

Component-specific PROGRAM parameters

No component-specific PROGRAM parameters should be declared for the communications component APP.

Parameter group DESTINATION

For every communications system (CA) to be connected, create a parameter group of type DESTINATION in the PROGRAM parameter group APP.

Establish the APPLI/COM-specific features here.

The information potentially required for optional parameters depends on the requirements of the respective communications system. For more information, please refer to the system manufacturer.

The individual parameters which can be declared in a program group of type DESTINATION are shown below:

1. **COM_JOB**: <Directory>
   - Mandatory. Path specification for COM_JOB directory.

2. **COM_ACK**: <Directory>
   - Mandatory. Path specification for COM_ACK directory.

3. **COM_ERR**: <Directory>
   - Mandatory. Path specification for COM_ERR directory.

4. **COM_OUT**: <Directory>
   - Mandatory. Path specification for COM_OUT directory.

5. **COM_IN**: <Directory>
   - Mandatory. Path specification for COM_IN directory.

6. **SYNC_JOB**: <Command>
   - Optional. With this parameter, it is possible to define an OS/2 or UNIX command which is executed after each time a transmission request is created. The CA can hence be informed that there are requests for transmission. This parameter is not necessary for CAs which automatically scan the directory COM_JOB.

7. **SYNC_ACK**: <Command>
   - Optional. With this parameter, it is possible to define an OS/2 or UNIX command which is executed before each time status messages or incoming messages are searched for. It is hence possible to request the CA to provide these messages. This parameter is not necessary for CAs which automatically put status messages and incoming messages into the directory COM_ACK.

8. **JOB_EXT**: <File extension>
   - Optional. States the (3-character) extension to the file name under which transmission requests are to be put into the directory COM_JOB. If this parameter is not declared, “app” is used.

9. **ACK_EXT**: <File extension>
   - Optional. States the (3-character) extension to the file names under which status messages and incoming messages are to be searched for in directory COM_ACK. The usual wildcard characters “*” and “?” can also be used. If this parameter is not declared, “app” is used.
APP-specific Entries in SAPcomm Configuration File

**DIAL_PREFIX <Dial sequence>**  
Optional. This enables you to state a sequence of digits or special characters, which is dialed before the actual number receiving the fax. This means, for example, that the number for getting an outside line in a PBX ("0" in most cases) can be set.

**CONVERT_ID <Format identification code>**  
Optional. This makes it possible to define the format identification code which can be stated via the APPLI/COM parameter “CONVERT_ID”. The LA informs the CA of the format in which the message to be sent is available.

There are two possibilities for messages from SAP systems:

The message is available in a text format (for SAP Systems R/2 and R/3 up to Release 2.2 the only possibility). In this case, the LA always uses the format identification code “ASCII”, the parameter CONVERT_ID being ignored.

The message is available in a printer format (SAP System as of Release 3.0), established in R/3 via the spool device type used. In this case, the LA uses the format identification code stated with the parameter CONVERT_ID. If this parameter is not declared, the LA uses “VOID”.

The decision as to which printer format (postscript, HP-PCL5, etc) is used, i.e. which spool device type is stated in R/3 (POSTSCPT, HPLJ4, etc) and which value is stated for the parameter CONVERT_ID (PS, PCL, etc), depends on which formats are supported by the CA and which format identification codes are expected on the APPLI/COM interface.

**TYPE_ID <Transmission type>**  
Optional. States the transmission type which can be set via the APPLI/COM parameter “TYPE_ID”, and with which messages are to be sent by the CA. If this parameter is not declared, “STD” (= “Standard”) is used.

**Parameter Group CHANNEL**  
The CHANNEL parameter groups which you create for the communications component APP must contain the following component-specific parameter:

**DESTINATION <Destination>**  
Mandatory. Here you state the name of a previously created DESTINATION parameter group and hence establish the CA with which messages are to be exchanged via this CHANNEL.
Interface Specification

The sections below describe in detail the version of the APPLI/COM file interface, as used by the communications component APP of the SAP communications server.

APPLI/COM Directories [Seite 168]
APPLI/COM Requests [Seite 170]

This information is not required for normal operation of SAPcomm with a communications system connected via APPLI/COM, not even for installation and configuration. This section is intended for those persons who want to adapt a communications system to SAPcomm via APPLI/COM, i.e. primarily for the manufacturers of these systems.
APPLI/COM Directories

Data is exchanged between LA (SAPcommm communications component APP) and CA (communications system) via files in shared directories. The following directories are used. The names stated are symbolic descriptions. The actual path names for these directories must be set in the SAPcommm configuration (see APP-specific Entries in SAPcommm Configuration File [Seite 165]) and on the communications system.

**COM_JOB**

The LA writes transmission requests into this directory. The requests only contain send attributes, the actual message content being stored in separate files in the directory COM_OUT. Each transmission request is stored in its own file, the extension of these files being configurable in SAPcommm (parameter JOB_EXT, see [APP-specific Entries in SAPcommm Configuration File [Seite 165]).

The CA must read, remove and process the transmission requests from this directory. It can also remove the files with the contents of the messages in the directory COM_OUT.

The CA can check for new requests at regular intervals without being requested, or be informed by SAPcommm via an operating system command call (parameter SYNC_JOB, see [APP-specific Entries in SAPcommm Configuration File [Seite 165]).

**COM_ACK**

The CA saves transmission requests which have been processed (successfully or not) into this directory.

The LA will read these requests, remove them, and generate status confirmations to SAP Systems from them. The files with the contents of the initial messages in the directory COM_OUT can also be removed by the LA.

The CA can store reception requests in this directory. These requests also contain only reception attributes. The message contents must be stored in separate files in the directory COM_IN.

The LA will read these requests, remove them and forward them to recipients in SAP Systems. The files with the message contents in the directory COM_IN are also removed by the LA.

The LA searches in the directory COM_ACK only for files which have a particular extension, this being configurable in SAPcommm (parameter ACK_EXT, see [APP-specific Entries in SAPcommm Configuration File [Seite 165]).

The CA can put processed transmission requests or reception requests in the directory COM_ACK without being requested, or be requested to do so by SAPcommm via an operating system command call (parameter SYNC_ACK, see [APP-specific Entries in SAPcommm Configuration File [Seite 165]).

**COM_ERR**

Transmission requests which cannot be processed because of their content (e.g. because of syntactic errors), are removed from the directory COM_JOB by the CA and stored in this directory.

These requests are not processed any further by the LA.

**COM_OUT**

The LA stores the message contents for transmission requests here.

**COM_IN**

The CA stores the message contents for reception requests here.
APPLI/COM Requests

The transmission and reception requests are normal text files in ASCII character set. Each line (except the header) consists of the following: a parameter name, possibly a sequence of blank characters, a colon, and the parameter value directly after the colon without separating blank characters.

The parameters required for each of the three types of request are described below:

Transmission Requests [Seite 171]
Completed Transmission Requests [Seite 174]
Reception Requests [Seite 176]
Transmission Requests

Transmission requests are created by the LA in COM_JOB and read and processed by the CA. They contain the following parameters:

FUNCTION
This parameter defines the type of request. The value for transmission requests is always SENDACK.

REQ ID
This parameter contains a unique number for this transmission request. The number is allocated by SAPcomm. The CA must refer to this number in completed transmission requests.
Maximum length: 8 characters.

LA ID
This parameter describes the source of the message: the name of the SAP System, and the client in which the message was generated. The CA can use this information, for example for cost assignment.
Maximum length: 16 characters.

USERKEY
This parameter contains an ID for the message, which was allocated in the SAP System in which the message was generated. This ID is only unique within this SAP System. As SAPcomm serves several SAP Systems simultaneously, several transmission requests may contain identical values in USERKEY (but different values in LA ID).
The parameter has the following structure: spool number (5 characters), period, job number (2 characters).

USERINFO
This parameter contains the user name of the sender of the message in the SAP System. The user name is only unique within this SAP System. As SAPcomm serves several SAP Systems simultaneously, there may be several persons with identical values in USERINFO (but different values in LA ID).
Maximum length: 12 characters.

SERVICE
This parameter states the communications service via which the message is to be sent. Possible values are:

- FX3  Fax group 3
- TLX  Telex

ADDRESS
This parameter contains the fax or telex numbers of the recipient of the message.
Maximum length: 30 characters.

SENDTIME
This parameter states the point in time when the CA is to send the messages. Possible values are:

- URGENT  Send immediately with high priority
- IMMEDIATE  Send immediately
- TARIFF1  Send at cheap night rate
- YY-MM-DD-hh:mm  Send at time stated
Transmission Requests

FILELIST
This parameter specifies the content of the message. The value of the parameter is made up of the following comma-separated fields:

- Name of the message content file. The name is stated in the form of a complete path, the file, however, is always in directory COM_OUT. As, for the installation of LA and CA on separate computers with different operating systems, the path specifications for the directory COM_OUT can be different for LA and CA, the CA should only evaluate the basic file name and extend it with the path specifications for the directory COM_OUT from its own configuration.
  Maximum length: 128 characters.

- Format of message content. Possible values are:
  ASCII Message is only text, which contains only printable characters in ASCII character set, and the control codes CR/LF (hex 0D/0A) at line ends and FF (hex 0C) at page breaks.
  VOID Message already prepared in a printer language (e.g. postscript) in creating SAP System. Instead of the value VOID, a different value can also be used which stipulates the printer format more precisely (e.g. PS). This should be set in the SAPcomm configuration (parameter CONVERT_ID, see APP-specific Entries in SAPcomm Configuration File [Seite 165]).
  Maximum length: 10 characters.

- Transmission type with which the message is to be sent by the CA. The standard transmission is labeled STD. The value can be changed in the SAPcomm configuration (parameter TYPE_ID, see APP-specific Entries in SAPcomm Configuration File [Seite 165]).
  Maximum length: 10 characters.

STATUS
This parameter is for completed transmission requests and has 2 blank characters.

ERROR
This parameter is for completed transmission requests and has 85 blank characters.

CIL
This parameter is for completed transmission requests and has 72 blank characters.

COMID
This parameter is for completed transmission requests and has 24 blank characters.

COSTS
This parameter is for completed transmission requests and has 80 blank characters.

Transmission request for text message from SAP System “ABC”, client “005”, from user “BONDJ” to fax number 001-206-8828080, to be sent immediately:

I*APPLI/COM*1996*SAP*
FUNCTION: SENDACK
REQ-ID : 123456
LA-ID : ABC005
USERKEY : 09876.01
USERINFO:BONDJ
SERVICE:FX3
ADDRESS:0012068828080
SENDTIME:IMMEDIATE
FILELIST:/usr/sap/com/appli/out/00123456.001,ASCII,STD,
STATUS:
ERROR:
CIL:
COMID:
COSTS:
Completed Transmission Requests

Completed transmission requests are stored by the CA in COM_ACK and read and processed by the LA. They must contain the following parameters. All additional parameters in a completed transmission request which are not listed here are ignored by the LA.

**FUNCTION**

This parameter defines the type of request. The value for completed transmission requests is always SENDACK.

**REQ ID**

This parameter contains a unique number for the processed transmission request. The number must correspond to the value in the transmission request.

**ERROR**

This parameter states the ultimate success or failure of the transmission request processing. The value of the parameter is made up of the following two slash-separated fields:

- error code (as in list below, 4 characters)
- error text (explanatory text, maximum 80 char)

Maximum length : 85 characters.

List of valid error codes and the resulting status of the message in the SAP System:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>Successful</td>
<td>Sent</td>
</tr>
<tr>
<td>5001</td>
<td>Service not supported</td>
<td>Invalid address</td>
</tr>
<tr>
<td>6019</td>
<td>TYPE ID invalid</td>
<td>Configuration error</td>
</tr>
<tr>
<td>6020</td>
<td>CONVERT ID invalid</td>
<td>Configuration error</td>
</tr>
<tr>
<td>9000</td>
<td>Transmission error</td>
<td>Transmission error</td>
</tr>
<tr>
<td>9001</td>
<td>Connection interrupted</td>
<td>Transmission error</td>
</tr>
<tr>
<td>9002</td>
<td>Connection occupied</td>
<td>Occupied</td>
</tr>
<tr>
<td>9003</td>
<td>Connection not established</td>
<td>Recipient not reached</td>
</tr>
<tr>
<td>9006</td>
<td>Transmission error (transport)</td>
<td>Transmission error</td>
</tr>
<tr>
<td>9007</td>
<td>Transmission error (session)</td>
<td>Transmission error</td>
</tr>
<tr>
<td>9008</td>
<td>Recipient terminates connection</td>
<td>Transmission error</td>
</tr>
<tr>
<td>9009</td>
<td>Recipient refuses connection</td>
<td>Recipient not reached</td>
</tr>
<tr>
<td>9010</td>
<td>Receiving device not compatible</td>
<td>Device not compatible</td>
</tr>
<tr>
<td>9011</td>
<td>Transmission error (document)</td>
<td>Transmission error</td>
</tr>
</tbody>
</table>

All other error codes lead to "unknown error" status in the SAP System.

**CIL**

This parameter states the sender and recipient IDs, as well as the actual transmission time. The value of the parameter is made up of the following slash-separated fields:

- Recipient ID (CSID), 24 characters, may be filled with blank characters
- Sender ID (CSID), 24 characters, may be filled with blank characters
- Date and time of transmission, format : YY-MM-DD-hh:mm
- Additional information, e.g. number of pages (ignored by LA)

**COMID**

This parameter contains a unique ID for the send operation, allocated by the CA. Maximum length : 24 characters.
COSTS
This parameter contains the costs accrued through this transmission request. This information can be either in the country-dependent small currency unit (without decimal point) or in the large currency unit (with decimal point and all decimal places).
Maximum length: 80 characters, maximum numerical value: 999999.

FILELIST
This parameter is optional in completed transmission requests. If the parameter is contained in the request, the LA deletes the file stated in this parameter (if the CA has taken the parameter from the transmission request, this is the file with the message content). If the completed transmission request does not contain this parameter, the CA must deal with the deletion of the message content file itself.

Completed transmission request for a text message from the SAP System “ABC”, client “005”, from user “BONDJ” to fax number 001-206-8828080, processed by the CA under the ID “0345Fk12”. The transmission was terminated by the distant device and gave rise to costs of 2.88 DM.
The parameters LA ID, USERKEY, USERINFO, SERVICE, ADDRESS, SENDTIME and STATUS were ignored by the LA in the example.

FUNCTION: SENDACK
REQ-ID: 123456
LA-ID: ABC005
USERKEY: 09876.01
USERINFO: BONDJ
SERVICE: FX3
ADDRESS: 0012068828080
SENDTIME: IMMEDIATE
FILELIST:/usr/sap/com/appli/out/00123456.001,ASCII,STD,,
STATUS: -
ERROR: 9008/transmission terminated by distant fax machine
CIL: +1 206 882 8080 ms /+49 6227 34 1282 sap /95-11-24-18:13
COMID: 0345Fk12
COSTS: 2.88 DEM
Reception Requests

Reception requests are created by the CA in COM_ACK and read and processed by the LA. They must contain the following parameters. All additional parameters in a reception request which are not listed here are ignored by the LA.

FUNCTION
This parameter defines the type of request. The value for completed reception requests is always RECEIVE.

SERVICE
This parameter states the communications service via which the message was received.
Possible values are:
- FAX for fax (group 3 or 4)
- FX3 for fax group 3
- FX4 for fax group 4
- TLX for telex

SUBADDR
This parameter states the recipient's number. This can either be the whole fax/telex number, or just the individual extension number of the recipient. This number can be used in the SAPcomm configuration, to establish the SAP System, to which the message is to be forwarded (PROGRAM group SCB, parameter group ROUTING, SERVICE group FAX or TLX, parameter NUMBER, see The Routing Parameter Group [Seite 78]).
Within the SAP System (as of R/3 Release 3.0), this number can be used to assign the recipient(s) of the message.
Maximum length : 30 characters.

CIL
This parameter states the sender and recipient IDs, as well as the actual transmission time. The value of the parameter is made up of the following slash-separated fields:
- Recipient ID (CSID), 24 characters, may be filled with blank characters
- Sender ID (CSID), 24 characters, may be filled with blank characters
- Date and time of transmission, format : YY-MM-DD-hh:mm
- Additional information, e.g. number of pages (ignored by LA)

COMID
This parameter contains a unique ID for the receive operation, allocated by the CA.
Maximum length : 24 characters.

FILELIST
This parameter specifies the content of the message. The value of the parameter is made up of the following, comma-separated fields:
- Name of message content file. The files must always be in the directory COM_IN. The name may be stated in the form of a complete path, but is this case the LA will only evaluate the basic file name and extend it with the path specifications for the directory COM_IN from the SAPcomm configuration. The file name is compiled by the LA in lower case letters. The CA should therefore never generate files whose names contain upper case letters.
Maximum length : 128 characters.
- Format of the message content. Possible values are:
Reception Requests

ASCII  (Only allowed for telex messages.) The message is only text, which contains only printable characters in ASCII character set. The parameter FILELIST can only be declared once in this case, i.e. the message can consist only of a content file.

TIFF  (Only allowed for fax messages.) The message consists of graphics coded as TIFF G3 or G4. If the messages is made up of several single-sided TIFF files, the parameter FILELIST must be declared several times.

• Transmission type with which the message was received by the CA. Is ignored by the LA.

Reception request for fax message from sender “+1 206 882 8080” to recipient “+49 6227 3459 1624”, consisting of three pages.

I*APPLI/COM*1996*SAP*
FUNCTION:RECEIVE
SERVICE :FX3
SUBADDR :1624
CIL :+49 6227 3459 1624 /+1 206 882 8080 ms /95-11-24-19:57
COMID :0345Wp09
FILELIST:/usr/sap/com/appli/in/00815p01.tif, TIFF, STD,
FILELIST:/usr/sap/com/appli/in/00815p02.tif, TIFF, STD,
FILELIST:/usr/sap/com/appli/in/00815p03.tif, TIFF, STD,
The Communications Component TST

Basics [Seite 179]

TST-specific Entries in the SAPcomm Configuration File [Seite 180]

SAP-specific Aspects for TST Operation [Seite 181]

TST-specific Log Entries [Seite 182]
Basics

The component TST is a diagnosis tool which you can use to simulate the behaviour of a communications component.

Furthermore, the source code in the delivery version of SAPcomm can be used as an example code for using the SAPcomm interface (see The SAPcomm Development Library [Seite 183]).

When dispatching messages via channels that are allocated to this component, it functions as a quasi "black hole" which receives the messages, ignores them and returns a (configurable) transmission report.

Files can be created to simulate incoming documents that are processed as received messages and then forwarded to other SAPcomm components. Such messages are always flagged as telematic messages. The TST component does not support the generation of EDI or mail documents.
TST-specific Entries in the SAPcomm Configuration File

TST is the identifier of the test component.

You must therefore include the following parameter set in the configuration file SAPCOMM.CFG:

```
PROGRAM TST
  <Parameter specification>
END-PROGRAM
```

A general description of this file can be found in the configuration file (see The Configuration File SAPCOMM.CFG). The TST-specific features of this parameter specification are explained below.

**PERFORM Parameter Group**

No statements need to be entered in the configuration file SAPCOMM.CFG for communications component TST to execute TST-specific administrative actions.

**DESTINATION Parameter Group**

No DESTINATION parameter sets need to be entered in the configuration file SAPCOMM.CFG for communications component TST.

**SELECTION Parameter Group**

No SELECTION parameter sets need to be entered for communications component TST.

**CHANNEL Parameter Group**

The CHANNEL parameter sets that you create for communications component TST can contain the parameters explained below:

**SENDRC**

`<return code>`

This parameter specifies the return code that is to be confirmed as send status when dispatching messages.

Value "0" results in a positive confirmation (sent successfully), all other values represent a negative confirmation ("error in sending").

**RECEIVE**

`<receive file>`

A file name used to simulate incoming documents can be specified using this parameter. This file is treated as a received message and is forwarded for each event with a receive query (PERFORM .. ACTION RECEIVE). The file is not deleted! Incoming documents are always treated as telex messages, other applications and services are under development.

**RECIPIENT**

`<number>`

The recipient <number> is allocated to incoming messages simulated by TST and is used to route these messages to an SAP System, for example. Please read the model configuration at the end of The SAPcomm Development Library [Seite 183].

**Component-specific PROGRAM Parameters**

No PROGRAM parameters need to be entered in configuration file SAPCOMM.CFG for communications component TST.
SAP-specific Aspects for TST Operation

This version of component TST is a minimal solution to enable a simple installation and represent an example program that is as uncomplicated as possible (see The SAPcomm Development Library [Seite 183]). You should therefore bear the following limitations in mind for TST operation:

- The component does not store messages intermediately. Therefore, a message is only to be created and/or sent in the SAP System after the previous message has been completely processed - that is, the send status report has been received. Dispatching several messages at once results in a status report being created for only one message.

- Received messages can only be of the category telex. Consequently, the message contents are searched and forwarded to a printer and/or SAP mail users via processing in the SAP System (see Configuration in the SAP System [Seite 64]).
TST-specific Log Entries

The following lists the log entries specific to the communications component TST and explains each of them. Explanations for general log entries can be found in Notes on Testing and Problem Analysis [Seite 87].

I1001 Sending file <filename>
Receiving a message to be sent. The message is logged with this entry, but then ignored and not sent.

I1002 Confirm last message sent
Generating a send status report for the last received message to be sent (log entry I1001). The return code in this report was defined with parameter SENDRC.

I1003 Receiving file <filename>
Generating an incoming message from file <filename>. The name of this file was defined with parameter RECEIVE.
SAPcomm Development Library

Overview [Seite 184]
Scope of Price List Component [Seite 186]
The Structure and Tasks of a Communications Component [Seite 187]
Message Flow in SAPcomm [Seite 189]
Message and Status Processing in SAPcomm [Seite 191]
Programming With the SAPcomm Development Library [Seite 194]
Creating a SAPcomm Grammar File [Seite 200]
Testing a Communications Component [Seite 203]
Description of the SAPcomm API [Seite 204]
Compiling and Linking the Example Program [Seite 217]
Example Configuration [Seite 218]
Overview

The SAPcomm Development Library allows you to connect communication systems for EDI, Mail and telecommunications data transfer to SAPcomm.

To perform this kind of connection, you basically create your own SAPcomm communications component using the SAPcomm Development Library.

From the point of view of the SAPcomm control component, this communications component behaves like an SAP communications component:

- It can be configured through its "own" PROGRAM parameter group in the SAPcomm configuration file sapcomm.cfg
- It is started and ended by the SAPcomm control component.

The SAPcomm Development Library provides the following options:

- You can instruct SAPcomm to transmit certain (or all) messages to a SAPcomm component you have developed yourself rather than to an SAP communications component. The SAPcomm component would then forward these messages to a particular communication system you are implementing, for example.
- You can pass messages to SAPcomm via the SAPcomm Development Library and then instruct SAPcomm to forward these messages to any configured communications component.
- You can instruct SAPcomm to start your program for the actions you wish at configured times and independent of the messages.

Before developing your own component and integrating it into SAPcomm, familiarize yourself with the details of message processing within SAPcomm. Proceed from the overview of the sections of documentation on the SAPcomm Development Library below.

Then carry out the following steps:

1. Use the routines provided in the SAPcomm Development Library (SciGetParam, etc.) to program a communications component.
   - The program must be stored in the SAPcomm call directory and must adhere to the following naming convention:
     `scmi<identifier><extension>`
   - Identifier:
     - Any combination of three letters (lower case), that has not yet been allocated.
     - The IDs that have so far been allocated are: adm, cae, cmt, cx4, exp, gam, odx, sap, scb, ssw, top, tst, xpp.
     - To register an ID for a new SAPcomm communications component at SAP, consult your SAP contact person.
   - Extension:
     - "exe": for OS/2 only
2. You yourself determine the configuration parameters (PROGRAM, CHANNEL, PERFORM etc.) that your component requires. You define these parameters in a "grammar file" called `scmi<identifier>.grm`, and store this grammar file in the SAPcomm profile directory. For more information, read the section Creating a SAPcomm Grammar File [Seite 200].
3. You enter your component in the SAPcomm configuration file by including a PROGRAM parameter group `<identifier>` with the respective parameters defined in your grammar file.
   - If you have carried out these steps successfully, SAPcomm will start your component along with the other configured components during the start phase, and activate it at the times you have configured.
To program a communications component, you can use the following compilers:

**UNIX:** ANSI-C Compiler
**OS/2:** The SAPcomm Development Library was created with the IBM Compiler C Set ++

If you get the error `gethostname: unresolved external` when linking the communications component, create a “dummy” function called `gethostname` (with an empty function part) in your program. This function is only used in the SAP communications component.
Scope of Price List Component

The SAPcomm Development Library includes the following components:

- `scmi.h`  
  Header file for OS/2 and UNIX. It contains the data structure and function declarations.

- `scmiux.o`  
  UNIX version of the development library

- `scmio2.lib`  
  OS/2 version of the development library

- `scmitst.c`  
  Model program for using the development library

The SAPcomm Development Library is available for all platforms listed under Operating Systems. To operate SAPcomm in connection with R/2 and a SAPcomm communications component that was not created by SAP, you require one of the following price list components:

- 837 (OS/2)
- 840 (UNIX)
The Structure and Tasks of a Communications Component

This topic describes where in the SAPcomm architecture a communications component belongs, which tasks it is responsible for, and which functions are at its disposal for their execution.

The SAPcomm communications components are the actual access points of external communication systems to SAPcomm. One SAPcomm communications component is always assigned to each communication system (it can in turn "serve" several communication systems). Each communications component communicates "outwards" with one or more external communication systems, and "inwards" with the SAPcomm control component (and not directly with other communications components).

Corresponding to its two-way communication task ("outwards" and "inwards"), every communications component consists of two parts linked together in an independent program:

- **Internal part**: identical for all communication components
- **External part**: specific to the connected communication system

The SAPcomm API is the interface between the internal and external part of a communications component.

The SAPcomm Development Library consists of the coding of the "internal part" of a communications component with its routines for communicating with the SAPcomm control component. It also contains the "main" function of the communications component.

The "external part" of a communications component contains the actual access to the external communication system connected.

The internal and external parts contain the following variables and functions:

**Variable that must be defined in the external part:**
- SciVersion  Version of the header file used

**Functions that must be made available by the external part:**
- SciInit  Initialize external part at the start of the program
- SciFinit  End external part
- SciOpen  Start a processing sequence
The Structure and Tasks of a Communications Component

SciClose End a processing sequence
SciSendMsg Send a message
SciRecvMsg Fetch a message
SciFunction Additional functions

Functions available in the internal part (of the development library):
SciGetParam Read a configuration parameter
SciGetAttr Read a message attribute
SciSetAttr Set a message attribute
SciTrace Write entries to log files
SciActivity Inform on present activity
SciFileOpen Open a file
SciFileClose Close a file
SciFileRename Rename a file
SciFileRemove Delete a file

Therefore, using the SAPcomm Development Library to create a new communications component means programming the “external part” of a communications component with the routines required by the SAPcomm API, and linking it with the “internal part” (of the SAPcomm Development Library) in an independent program.
Message Flow in SAPcomm

This topic describes the stages, through which documents and their status information must pass, on their way from one communication system to another (and back again), and the “trigger” that is necessary to pass each individual stage.

To be able to use the functions of the SAPcomm Development Library correctly, you must have an understanding of each stage in this process.

Between generation of a SAPcomm message by a SAPcomm communications component (for example, an outbound fax document from an SAP System or an X400-incoming message) and issuing of a “send status” confirmation to this communications component, the following four stages are passed:

1) Message is picked up from a connected communication system (source system)
2) Message is passed to a connected communication system (target system)
3) Send status information is picked up from the target system
4) Send status information is forwarded to the source system

For information on programming the relevant details for these four stages, refer to Programming With the SAPcomm Development Library [Seite 194].

The communications component does not carry out any of these four steps on its own initiative: Communications components only become active when they receive a request from the control component to perform a particular action.

How Requests are Triggered

The following documentation explains which events lead to the control component sending requests to a communications component, and which functions of the external part are called as a result:

1) At the start and end of the program
   The control component sends INIT or FINIT requests. These requests result in the call of the SciInit and SciFinit functions of the external part.

2) The control component sends “local” RECEIVE requests at regular intervals.
   These requests result in the call of the SciOpen function with the call parameter SCI_OPE_LPRC (Local Processing), as well as subsequent calls of the SciRecvMsg and SciClose functions. You can control the frequency of RECEIVE requests with the configuration parameter REQUEST_INTERVAL.

3) If one or more documents are generated with priority SCI_SCH_IMMEDIATE, the control component sends a SEND request to the communications component(s) responsible for forwarding.
   This request results in the call of the SciOpen functions with the call parameter SCI_OPE_SEND, as well as subsequent calls of the SciSendMsg and SciClose functions.

4) The control component sends requests for actions configured for the communications component to be executed.
   These requests are triggered either by time plan statements in the SAPcomm configuration file or actions performed by the user via the graphical user interface or calling the help program scmperf.
   The connection between the configured actions and the resulting calls of functions of the external part is as follows:
Message Flow in SAPcomm

- The SEND-SCHEDULE and SEND-NIGHT actions result in SEND requests. The internal part first checks whether messages or status information exist in the current channel. If not, the external part is not activated. Otherwise the SciOpen function is called with the call parameter SCI_OPE_SEND, and the SciSendMsg and SciClose functions are called as a result. The difference between the two actions is that SEND-NIGHT only takes into account messages created with priority SCI_SCH_NIGHT, while SEND-SCHEDULE takes all the others into account.

- The RECEIVE action results in “non-local” RECEIVE requests. These requests result in the call of the SciOpen function with the call parameter SCI_OPE_RRCV (Remote Receive), as well as subsequent calls of the SciRecvMsg and SciClose functions.

- Other actions, which were defined specially for the communications component, result in FUNCTION requests.
  The SciFunction function is called, and the name of the action is passed to it as the call parameter. It is up to the external part how it wants to respond to these SciFunction calls.

Programming (in the external part) the processing sequences initiated by the call of the SciOpen function is described under Programming With the SAPcomm Development Library [Seite 194].
Message and Status Processing in SAPcomm

This section provides information on the character set used in SAPcomm - in text documents, for format control in fax documents and for message ID allocation.

- Message formats
- Message IDs
- Processing of send status information

Message Formats

Character Set

In the SAPcomm environment, texts are displayed in the IBM PC Multilingual character set (IBM code page 00697/00850). Outbound text messages (body type SCI_BDT_TXT) are passed in this code page to the external part, and incoming text messages must be passed from the external to the internal part in this code page. If conversions are necessary, they must be made in the external part.

Format of Fax Documents (Outbound)

Outbound fax documents are passed to the external part as text messages or binary messages. They contain text in both cases - only character set and format control are different.

The range of options for including format control in outbound fax documents depends on the SAP System in which the document is generated (R/2 or R/3, Release level).

The different options are described below:

R/2 (all Releases supported by SAPcomm)

- Type of message “body”: SCI_BDT_TXT (text message).
- Display of end of line: CR-LF (OS/2) or LF(UNIX)
- Display of page break: ">SU" at beginning of line (Rel. 5.0F), or "\x1c SPAGE" at any place (from Rel. 5.0G)

Character conversion into other code pages can be carried out using codepage conversion files (refer to the description of the CONVERT and DCONVERT parameters of the SAP communications component).

R/3: Releases 2.1* and 2.2*

- Type of message “body”: SCI_BDT_TXT (text message).
- Display of end of line: CR-LF (OS/2) or LF(UNIX)
- Display of page break: "\x1c SPAGE"

The code page cannot be changed.

R/3: Release 3.0A onwards

There are two options:

1) Data is passed via the SAPcomm API, as in R/3 Releases 2.1 and 2.2 (see above).
2) Format control in the SAP System (“R/3 3.0 session”, see below).

You can choose between these two options in table TSP09 (throughout the R/3 System):

| TSP09: TELE entry, ABAP/4 driver = 'X' => | R/3 2.1/2.2 session |
| TSP09: TELE entry, ABAP/4 driver = '' => | R/3 3.0 session |

R/3 3.0 Session:

- Type of message “body”: SCI_BDT_BIN (binary file).
Message and Status Processing in SAPcomm

- Character set and format control are set in the SAP System (with transaction SPAD) for the fax system connected.
- SAPspool formatting generates a binary file with a format adapted to the fax system. Thus, depending on the requirements, HPCL3, HPCL5, NEC, EPSON, PostScript, Prescribe, or proprietary printer languages can be generated for fax systems not supported in the standard.

Note that the “output device” associated to the fax in the “R/3 3.0 session” must be a “SAPcomm output device” (see topic “Configuration in the SAP System”).

The standard R/3 delivery (Release 3.0) contains “descriptions” of the “device types” SAPCOMMF and GAMMAFAX. These generate data in the IBM PC Multilingual character set (IBM 000697/00850, SAP 1103) with CR-LF(OS/2 and UNIX) and CR-FF.

Once they have been created in a SAP System, device type descriptions can be exported to the (UNIX) file system by the SAP program RSTXSCRP, and imported to other SAP Systems by the same program (RSTXSCRP).

Format of Fax Documents (Inbound, R/3 only, as of Release 3.0)

The recipient can display inbound fax documents directly from the SAP System (using display software included in the R/3 delivery).

For this, they must be passed to the internal part as follows:
- in TIFF format
- with a body part for each page (see message attribute SCI_A_MBODY)
- with the body part type SCI_BDT_FAX (inbound fax)

Message ID (SAPcomm Message/Org/Ext Ids)

Messages are marked by the message attributes SCI_A_MID (SAPcomm message ID), SCI_A_ORGID (originator generated ID) and SCI_A_EXTID (external ID). The purposes of this are as follows:
- Management of messages within SAPcomm
- Support for the management of messages within the connected communication systems
- Assignment of send status information to the respective message

The meaning of these attributes is as follows:

SCI_A_MID
Allocated automatically when a SAPcomm message is generated. It forms the first six characters of the names of all files belonging to the message (in this way, they are marked as belonging to a common message).

SCI_A_ORGID
The ID of the message in the source communication system. Set explicitly by the external part when a (new) message is passed to SAPcomm (within processing a RECEIVE request), if the connected communication system, from which the message has come, has allocated an ID and reported it to the external part.

Example: The external part of the SAP communications component sets the “Origid” for outbound fax documents generated in the SAP System to the number of the corresponding spool document.

SCI_A_EXTID
The ID of the message in the target communication system. It can only be set "subsequently" for outbound messages, that is, when the first status information is generated for the message (within processing a RECEIVE request).
Example: The “Extid” of a SAPcomm message sent via X400 is the X400 Message ID.

**Processing Send Status Information**

Send status information is exchanged via the SAPcomm API as a type SCI_SOM_STAT message (message attribute SCI_A_SOM). SAPcomm supports - in the sense of X400 - two-level status messages, which can be distinguished by the following values of the SCI_A_RETCODE status attribute:

- SCI_DELIVERED / SCI_NOT_DELIVERED
- SCI_RECEIVED / SCI_NOT_RECEIVED.

**The Procedure for Status Processing:**

The internal part of a communications component keeps an attribute file (stored in the “processing directory”) for each message passed to the external part for forwarding to the target communication system. The attribute file is necessary for the processing of send status information that refers to this message. There are two different cases:

- No status information (or “notifications”) required for the message to be sent (the attributes cDelReq and cRecReq have the value SCI_NRQ_NONE for each recipient).
  
  In this case, the attribute file is deleted after the message is passed to the external part. Subsequent setting of the send status is not possible.

- DELIVERY or RECEIPT notifications are required for at least one recipient (attribute cDelReq or cRecReq has the value SCI_NRQ_FULL).
  
  In this case, the attribute file is kept until the last required notification has been passed from the external to internal part (of the SAPcomm component belonging to the target communication system). It is then automatically deleted by the internal part. Additional status information settings for this message are not possible. If there are any remaining attribute files (e.g. one of the Receipt notifications required does not arrive), they must be deleted in another way (e.g. using program scmreorg, manually or by SAPcomm through the time plan).

Value SCI_NRQ_NEG for attributes cDelReq and cRecReq is currently not supported. Please do not use it.
Programming With the SAPcomm Development Library

From the point of view of the external part, the basic program flow can be divided into three stages:

- Initialization [Seite 195]
- SEND, RECEIVE and FUNCTION Requests [Seite 196]
- Termination [Seite 199]
Initialization

The internal part calls SciInit (once at the start of the program). Then the external part queries the internal part regarding the values of “its” PROGRAM configuration parameters, and stores the values in static variables. The internal part calls the function SciGetParam once for each configuration parameter.

SciGetParam( "PROGRAM", "MYPROGPARAM1", buff, len_buff);

The external part also performs initializations (checks availability, logon etc.) specific to the communication system connected.
SEND, RECEIVE and FUNCTION Requests

Depending on the events that occur (see section "Message Flow in SAPcomm"), the control component repeatedly sends requests to the communications component. The different types of requests result in the following processing logic ("message" stands for both "document" and "status information"): 

SEND Request:
The SEND request results in all documents waiting to be dispatched (in the current channel, and taking into account the SCI_A_SCHEDULE scheduling attribute), and all status messages waiting to be processed, being passed to the external part.

Flow:
1) The internal part calls SciOpen with the call parameter SCI_OPE_SEND (start of a SEND processing sequence).
   - The external part queries the CHANNEL-specific configuration parameters by calling the SciGetParam function, performs operations initiating dispatch (if necessary) and finally executes a "return".
2) The internal part repeatedly calls SciSendMsg (for each message).
   - From the SciSendMsg function, the external part first calls SciGetAttr for reading the message attribute SCI_A_SOM:
     NrChars = SciGetAttr( SCI_A_SOM, buffer, len_buffer);
   - Depending on the result (in "buffer") - SCI_SOM_DOCU or SCI_SOM_STAT - the message to be processed is either a document to be dispatched or status information to be processed.

In the first case, the document-specific attributes (see "Functions and Parameters of the SAPcomm API") are now read by the repeated call of SciGetAttr (SCI_A_MID, SCI_A_ORGID, SCI_A_SUBJECT, SCI_A_MORG, SCI_A_MRECIP (repeatedly), SCI_A_MBODY (repeatedly) and others depending on document type).

In the second case, the attributes belonging to the status information are queried accordingly (or "return" with SCI_OK is executed immediately, if status information is not to be edited).

If the document, to which the status information refers, had several recipients, this message may also contain status information that refers to several (but possibly not all) recipients.

Therefore, the recipients and the recipient-related attributes (SCI_A_RETCODE, SCI_A_RETTEXT, SCI_A_DELDATE, SCI_A_DELTIME) are queried in a loop (repeated query of SCI_A_MRECIP until SCI_NOMORE is returned).

- Finally, the operations necessary for further processing of the message are performed, followed by "return" with SCI_OK.
3) The internal part calls SciClose (end of processing sequence).

RECEIVE Request:
The RECEIVE request gives the external part the opportunity to pass inbound messages and status information (for documents transferred earlier in a SEND processing sequence) to the internal part.

Flow:
1) The internal part calls SciOpen with the call parameter SCI_OPE_LPRC or SCI_OPE_RRCV (start of a processing sequence).
Only for SciOpen with SCI_OPE_RRCV: The external part queries the CHANNEL-specific configuration parameters by calling the SciGetParam function (there is no “current” channel for SCI_OPE_LPRC, and therefore no CHANNEL-specific parameters). After all the necessary operations for initiating the RECEIVE process have been performed, "return" is executed with SCI_OK.

If there are no messages or status information to be passed to the internal part, "return" is performed with SCI_NOMORE. In this case, the SciRecvMsg and SciClose functions are not called.

2) The internal part repeatedly calls SciRecvMsg (for fetching each message)

- If the external part has no document or status information to pass to the internal part, it executes "return" with SCI_NOMORE (then the internal part no longer calls SciRecvMsg, but skips to step 3) below).

- If inbound documents or status information (from the connected communication system) do exist, the external part starts to pass the message. By calling the SciSetAttr function, it sets the SCI_A_SOM attribute, namely with the value SCI_SOM_DOCU if a document (inbound fax..) is to be passed, or the value SCI_SOM_STAT if status information is to be passed (for a document transferred earlier). Then the internal part repeatedly calls SciSetAttr to set a message attribute in each case. Finally, SCI_A_EOM is set (check return code, see topic "Functions and Parameters of the SAPcomm API"). As soon as all attributes (for which the external part has values) are set for a message, "return" is executed with SCI_OK.

- Setting status attributes: To assign status information to the relevant transmission document, the attribute SCI_A_MID must be set. It is also possible to set the attribute SCI_A_EXTID, so that additional status information based on the attributes SCI_A_ORGID and SCI_A_EXTID can also be assigned alone (without SCI_A_MID). For performance reasons, the attribute SCI_A_MID should always be specified, providing the SAPcomm message ID is known.

- Generating documents (SCI_SOM_DOCU): To be able to assign the generated message to “its” channel based on the recipient address via the routing implemented in SAPcomm (or to be more precise, to a channel of the communications component responsible for forwarding to the target system), the address type (SCI_A_RAT) - in addition to the address itself - must be set. The following address types are currently supported: SCI_RAT_X400, SCI_RAT_SNADS, SCI_RAT_SAP, SCI_RAT_FAX, SCI_RAT_TTX, SCI_RAT_TLX, SCI_RAT_OFTP, SCI_RAT_EXP, SCI_RAT_BCS.

- Generating documents, attribute SCI_A_OCHANNEL: If the external part of your component expects send status information for documents which it passes to the internal part, you should only pass the documents in a processing sequence initiated with SciOpen and the call parameter SCI_OPE_RRCV. Only then is it certain that the message attribute OCHANNEL is correctly set (it is set by the internal part). Status information, on the other hand, can be set in every RECEIVE processing sequence.

- Generating documents, attribute SCI_A_CORRDATA: You can use this attribute to link any data you want to a message, which you can access again for all later corresponding status information.

3) The internal part calls SciClose (end of processing sequence).

**FUNCTION Request**

**Flow:**
The internal part calls **SciFunction**. The external part executes the required functions, generally dependent on the call parameter. The value returned with „return“ is simply entered in the log file; it does not have any other importance.
Termination

The internal part calls SciFinit (once at the end of the program). Then the external part performs actions specific to the communication system connected (logoff, etc.).
Creating a SAPcomm Grammar File

Every SAPcomm communications component has its own grammar file. Comparably to a Data Dictionary, this file consists of the definitions of all parameters that can be used to configure the communications component.

In particular, these definitions determine:

- The names of all parameters allowed
- The attributes of these parameters

(Each or mandatory, can be specified once or several times, value range).

The grammar files are ASCII text files. You can create and maintain them with any text editor. You should not change a grammar file that has been correctly created unless a program change to the communications component makes this necessary.

The actual configuration of the component is performed exclusively in the SAPcomm configuration file.

Every grammar file should at least define the configuration parameters described in the section *The Configuration File SAPCOMM.CFG [Seite 35]*. You can do this with the following template (the identifier of the new component to be created is XPL (example)). Use the grammar file *scmitst.grm* of the SAPcomm component TST as a model, and change it accordingly.

```
PGT XPL

R PERIODIC AL 10

F INTERVAL MI MO EF
F FROM TI OO EF
F UNTIL TI OO EF

ER

# Definitions of communication component independent parameters:
#

F TRACE ON OO
F TRACE_ALL ON OO EF
F INIT ON OO EF
F PGM_FLOW ON OO EF
F DATA_FLOW ON OO EF
F CS_RESPONSE ON OO EF
F FILES ON OO EF
F IPC ON OO EF
F MEMORY ON OO EF
F ROUTING ON OO EF
F QUEUE_MSG ON OO EF
F DEV_CTRL ON OO EF
F SCI_CALL ON OO EF
F SCI_PARAM ON OO EF
F SCI_ATTR ON OO EF
F SCI_PERF ON OO EF

EF
```
The following definition lets the user configure calls of the External Part function SciFunction. Define the admissible function parameter values in the ACTION entry below! If your component is not designed to handle calls of SciFunction, delete the following definition.

```
# F PERFORM AL 10 MP
  F DAY CO MONDAY TUESDAY WEDNESDAY THURSDAY FRIDAY SATURDAY SUNDAY EC OP EF
  F TIME TI OP EF
  F PERIODIC AL 10 OP EF
  F ACTION CO DEFINE HERE YOUR szFunction

PARAMETERS
  EC MP EF

EF

# F REQUEST_INTERVAL MI OO EF
F LOGFILE_EXPIRATION NU 3 OO EF
F SAVE_RESPONSE_INFORMATION ON OO EF
F ARCHIVE ON OO EF
F SENDREQUEST_BLOCKSIZE NU 2 OO EF
F OWNSESSION ON OO EF

# INSERT HERE the definitions of those PROGRAM parameters that are specific to your communication component!

# F CHANNEL AL 10 MP
  F PERFORM AL 10 MP
    F DAY CO MONDAY TUESDAY WEDNESDAY THURSDAY FRIDAY SATURDAY SUNDAY EC OP EF
    F TIME TI OP EF
    F PERIODIC AL 10 OP EF
    F ACTION CO RECEIVE SEND-SCHEDULE SEND-NIGHT

# ADD more "actions", if you want to support CHANNEL-dependant calls of SciFunction (see the comment on the previous PERFORM definition).

# EC MP EF

# INSERT HERE the definitions of those CHANNEL parameters that are specific to your communication component!

# EF

EPGT
```

At both places marked **INSERT HERE**, include the definitions of parameters required by your communications component.

For each parameter, include an entry with the following format:
Creating a SAPcomm Grammar File

F <parameter name> <value range> <use> EF

Parameter name:
Text field of any length. A distinction is made between upper and lower case: „Par1“ and „par1“ are two different parameters.

Value range:
The options for specifying value ranges are as follows:
1) Alphanumeric: AL <LEN>
2) Numeric: NU <LEN>
3) Minutes (0-24 hrs): MI
4) Time: TI
5) Switch („ON“, „OFF“): ON
6) Constant: CO <list> EC

Use:
This is where you determine whether a parameter is optional or mandatory, and whether it can occur several times or only once:
1) OO optional, can be specified once at most
2) OP optional, can be specified several times
3) MO mandatory, always specified once
4) MP mandatory, specified at least once

To test your grammar file, place it in the SAPcomm profile directory, rename any other grammar files that may exist (change file extension), and then start the program scmcfgcc with the option -g. If the grammar file is syntactically correct, the program reports the following: ‘Parsing grammar file: <Name>‘, and finally ‘No configurations file ???‘.
Testing a Communications Component

This topic describes the utilities available within SAPcomm for testing your component. Almost the entire functionality of a SAPcomm component can be tested without connecting SAPcomm to an SAP System.

Depending on the type of component you are testing, SAPcomm can be operated in one of the following test constellations:

- Only with the component you are testing
  Two or more channels are assigned to the component. SAPcomm Routing is set so that every message created by the component you are testing is routed to that component.

- With the component you are testing and the TST component
  Depending on the purpose of the test, SAPcomm Routing is set so that message output, message input or both functions of the new component are tested. You may have to modify the source program `scmitst.c` (contained in the delivery of SAPcomm) in accordance with the aims of the test (other address type of the messages generated, generate messages with several bodies, etc.), and then regenerate it.

It is recommended that you deactivate the SAPcomm time plan (configuration parameter `IGNORE_TIME_PLAN`) for the test, and start all activities via the graphical interface or the help program `scmperfm`.

Working With a Debugger

If you want to analyze your component with a debugger, which cannot be appended to active processes, or analyze the start phase of your component, proceed as follows:

1) Add your component to the SAPcomm configuration file, set the SAPcomm configuration parameter `IGNORE_CC_ABORT` to "ON", and start SAPcomm.
2) Terminate your component using the program `scmstop` (entry: `scmstop -c<identifier>`).
3) Start your component in the debugger.

Log Files

SAPcomm writes a separate log file for each component, including the SAPcomm control component (see topic "The SAPcomm Directory Tree"). The log file of your component is the most important utility you have for checking that your component is correct.

- You can generate reference log entries by starting SAPcomm with the component TST alone, and letting the component “receive” and “dispatch” documents.
- To analyze the log files, use the selection options of the help program `scmlog`.

The format of (almost) all log entries is as follows:

```
<time>  <channel> <message ID>    <category>  <text>
```

The columns `<channel>` and `<message ID>` are only filled if the log entry refers to a particular channel or a particular SAPcomm message ID.

Files generated by SAPcomm: Use the log files to check which files SAPcomm generates in connection with a SEND or RECEIVE action. Use program `scmactl` to check the contents of the message attribute files.
Description of the SAPcomm API

The following topics describe all the functions of the SAPcomm API, their call parameters and return codes, and the message and status attributes used in SAPcomm:

Functions and Parameters of the SAPcomm API [Seite 205]
Message Attributes [Seite 212]
SAP Return Codes for Message Transmission [Seite 216]
Functions and Parameters of the SAPcomm API

This topic describes the individual functions and parameters of the SAPcomm Development Library.

---

SciVersion

```c
char * SciVersion ;
```

This variable must be declared in the external part, and should be initialized with the value SCI_VERSION defined in the header file. In this way, the internal part checks the compatibility of the header and object file used.

---

SciInit

```c
int SciInit  (void) ;
```

Request to initialize external part. The configuration parameters required must be read by calls of SciGetParam.

```c
return SCI_OK : Initialization successful
SCI_FATAL : Initialization could not be performed
```

---

SciFinit

```c
int SciFinit (void) ;
```

Request to terminate external part.

```c
return SCI_OK : Termination successful
SCI_FATAL : Error terminating
```

---

SciOpen

```c
int SciOpen  (int iMode ) ;
```

The external part is notified of the start of a processing sequence.

**iMode** Type of processing sequence. Bit mask, in which the following values can be set:

- `SCI_OPE_SEND` : Send messages to external system
- `SCI_OPE_LPRC` : Fetch messages from external system
- `SCI_OPE_RRCV` : Set up connection between external system and remote address (destination) = active fetching of messages
Functions and Parameters of the SAPcomm API

return SCI_OK : Processing sequence can be started
SCI_RETRY : Processing is currently not possible
SCI_ERROR : Type of processing is not possible
SCI_FATAL : System error, no more SciOpen

The following is executed within the subsequent processing sequence:

a) If SCI_OPE_SEND is set: The function SciSendMsg is called once for each message to be sent.

b) If SCI_OPE_LPRC or SCI_OPE_RRCV is set: The function SciRecvMsg is called until no more messages are returned.

The function SciClose is called at the end of the processing sequence. The time interval between two processing sequences is controlled by the SAPcomm time plan.

If, for example, it is not possible to permanently keep connections to external systems, they can be set up and closed within the SciOpen and SciClose functions. In cases where the external systems are always addressable by local connections, these functions will generally be empty, i.e. will only consist of "return (SCI_OK);" or will only query the configuration parameters required.

---

SciClose

int SciClose (void) ;
The external part is notified of the end of the current processing sequence.

return SCI_OK : Processing sequence ended successfully
SCI_ERROR : Error in the processing sequence
SCI_FATAL : System error, no more SciOpen

---

SciSendMsg

int SciSendMsg (void) ;
Request to send a message. The attributes of the message must be queried via corresponding SciGetAttr calls.

return SCI_OK : Send request created successfully
SCI_RETRY : Error, retry may be successful
SCI_ERROR : Error, no more attempts
SCI_FATAL : System error, no more SciSendMsg

SciRecvMsg

```c
int SciRecvMsg (void) ;
```

Query whether messages (documents or status) exist. Which of the two cases exist, as well as the attributes of the message, must be specified via corresponding SciSetAttr calls.

```c
return
SCI_OK : Document or status existed
SCI_NOMORE : No message available
SCI_RETRY : An error occurred that can be eliminated
SCI_FATAL : System error, no more SciRecvMsg
```

SciFunction

```c
int SciFunction (char * szFunction ) ;
```

Request to execute an administration function. In the SAPcomm time plan, you can define the time you want a function to be called. You can define the meaning of the functions for each communications component (dependent on the abilities and requirements of the external system).

```c
return
SCI_OK : Function performed successfully
SCI_RETRY : An error occurred that can be eliminated
SCI_FATAL : System error, no more SciFunction
```

R/3: Releases 2.1 and 2.2

The function SciFunction is always called as follows:

```c
iFunction = SCI_FUN_REORG
```

OS/2 and R/3, Release 3.0 onwards

When the function SciFunction is called, the parameter szFunction has the current value defined in the ACTION parameter of the SAPcomm time plan.

For example, the following time plan entry as a subgroup of a CHANNEL parameter group results in the function SciFunction being called with szFunction = "CHECKLINE" at 8.00.

```c
PERFORM OWNFUNC
  ACTION     CHECKLINE
  TIME       08:00
ENDPERFORM
```

For more information on time plan statements, refer to SAPcomm Schedule Statements [Seite 48].
### SciGetParam

```c
int SciGetParam (char * szGroup,
                char *  szParam,
                char *  pParam,
                int     iMaxSize) ;
```

Delivers the value of a configuration parameter, which is specified in the SAPCOMM.CFG file for the respective communications component.

- **szGroup**: Type of parameter group, in which the parameter being searched for is defined (PROGRAM, CHANNEL, ...).
- **szParam**: Name of the required parameter.
- **pParam**: Pointer to the buffer, in which the value is to be stored.
- **iMaxSize**: Maximum number of characters (= size of buffer - 1).

#### Return Values
- If >= 0 : Number of transferred characters (without `\0`)
- **SCI_NOMORE**: No more values available
- **SCI_BUFERR**: Buffer too small
- **SCI_NOTFOUND**: Parameter not set
- **SCI_FATAL**: System error

If the parameter value is longer than the buffer provided, the error code SCI_BUFERR is returned; the entire length of the buffer is however filled (iMaxSize characters + `\0`). This means it is also possible to read and evaluate only the beginning of a parameter.

In the case of parameters that can be specified several times, SciGetParam must be called until the value SCI_NOMORE is returned.

### SciGetAttr

```c
int SciGetAttr ( int   iAttr,
                SCI_P_ATTR pAttr,
                int     iMaxSize ) ;
```

Delivers the value of an attribute of the current message to be sent. This function must only be called during processing of a call of SciSendMsg.

- **iAttr**: Required attribute (for possible values, see "Message Attributes [Seite 212"]
- **pAttr**: Pointer to the buffer, in which the value is to be stored.
- **iMaxSize**: Maximum number of characters (= size of buffer - 1).

#### Return Values
- If >= 0 : Number of transferred characters (without `\0`)
- **SCI_NOMORE**: No more values available
- **SCI_BUFERR**: Buffer too small
- **SCI_NOTFOUND**: Attribute not set
- **SCI_UNKNOWN**: Unknown attribute
To transfer an attribute value completely into the buffer, the data types defined for each attribute in scmi.h should be used for the buffer declaration (see "Message Attributes [Seite 212]"). If the attribute value is longer than the buffer provided, the error code SCI_BUFERR is returned; the entire length of the buffer is however filled (iMaxSize characters + "\0"). This means that it is also possible to read and evaluate only the beginning of an attribute value.

Attributes that may have several values assigned to them, e.g. SCI_A_MRECIP (recipient of a Mail message), must be queried by multiple calls of SciGetAttrib. If no more values are found, the function delivers SCI_NOMORE.

SciSetAttr

```c
int SciSetAttr (int   iAttr,
                 SCI_P_ATTR  pAttr ) ;
```

Sets the value of an attribute of the current message to be processed.

a) Call during processing of SciRecvMsg:

Attributes can be set for a transmission document or for a status message for any document that has already been sent.

b) Call during processing of SciSendMsg:

Attributes can only be set for the status message for the current document to be sent.

iAttr Required attribute

(for possible values, see "Message Attributes")

pAttr Pointer to the buffer, in which the value is stored

return

- SCI_OK : Successful
- SCI_BUFERR : Value specified is too long
- SCI_UNKNOWN : Unknown attribute
- SCI_FATAL : System error

SciTrace

```c
void SciTrace (long  lClass,
               char  cType,
               int  iMsgNr,
               char * szFormat,... ) ;
```

Writes a trace entry in a SAPcomm log file.

lClass Class of the entry (for possible values, see below)

cType Type of entry:

- 'E' = Error
- 'W' = Warning
- 'I' = Information
- 'X' = Extended information

iMsgNr Number of the entry (0 to 999 reserved for SAP)

szFormat Text of the entry

(Format according to the C function "printf")
Functions and Parameters of the SAPcomm API

Date, time and name of the communications component are added to the entry at the beginning of the line, and a "newline" (\n') is added at the end of the line (therefore this should not be contained in the error text).

The trace class (lClass) defines the contents area of the entry. Permitted values are:

- **SCI_TRC_CONFIG**: Configuration information
- **SCI_TRC_PGM_FLOW**: Function/program calls and return
- **SCI_TRC_DATA_FLOW**: Data passed/received
- **SCI_TRC_CS_RESPONSE**: Responses/reactions of external systems
- **SCI_TRC_FILES**: File operations
- **SCI_TRC_IPC**: Inter-process communication
- **SCI_TRC_MEMORY**: Storage operations
- **SCI_TRC_ROUTING**: Assignment/forwarding of documents/status
- **SCI_TRC_QUEUE_MSG**: Queue messages
- **SCI_TRC_DEV_CTRL**: Control of external devices
- **SCI_TRC_SCI_CALL**: SAPcomm API function calls
- **SCI_TRC_SCI_PARAM**: SAPcomm API parameter values
- **SCI_TRC_SCI_ATTR**: SAPcomm API attribute values

Type 'E', 'W' and 'I' entries are always written in the log file. Type 'X' entries, on the other hand, are only written if their class is activated in the configuration file in the TRACE group.

The entry number (iMsgNr) for non-SAP developments must be at least 1000.

```c
SciTrace (SCI_TRC_PGM_FLOW, 'E', 1125, "Program %s not executable, retcode=%d", sPgm, iRc);
```

**SciActivity**

```c
void SciActivity (int iAct);
```

Notifies the internal part of a change in the current activity status. This allows the user interface to optically display the current actions of the individual communications components. The call of this function is optional.

- **iAct**: <current action>

Values permitted for iAct are:

- **SCI_ACT_IDLE**: No activity
- **SCI_ACT_CALL**: Outgoing call
- **SCI_ACT_INCOMING**: Incoming call
- **SCI_ACT_CONNECTED**: Connection made
- **SCI_ACT_RECEIVE**: Receive data
SCI_ACT_SEND: Send data
SCI_ACT_ACK: Wait for confirmation
SCI_ACT_SYNCH: Send a confirmation


FILE * SciFileOpen (char * path,
                     char * type);
int SciFileClose (FILE * stream);
int SciFileRename (char * oldname,
                   char * newname);
int SciFileRemove (char * path);

These functions correspond to the following standard C functions. The meaning of the function parameters is identical:
- fopen
- fclose
- rename
- remove

When one of these functions is called, the corresponding standard C function is called, and additional actions, such as writing trace entries, storage space checks or similar, may be performed.

It is recommended that you always use these functions rather than the original standard C functions: The handling of file operations complies with SAPcomm, and there are better options for problem analysis.
## Message Attributes

Note the following points:

- All structures must be initialized (with Hex-0) before every use. (For example, the recipient structure SCI_MRECIP for each individual recipient.)
- **Message attribute MRECIP**
  
  The MRECIP field cDeliver must be marked with a cross (SCI_TRUE), so that an inbound message is in fact forwarded to the (MRECIP) recipient.

Every attribute is defined by a constant SCI_A_xyz. The maximum length permitted (without the final "0") is defined by a constant SCI_L_xyz. The data type SCI_xyz defines a character field of the length SCI_L_xyz+1.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Use with:</th>
<th>Document Status</th>
<th>Multiple-Call</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCI_A_SOM</td>
<td></td>
<td>Set Get</td>
<td>(M=Mandatory)</td>
</tr>
<tr>
<td>EOM</td>
<td></td>
<td>M</td>
<td>(O=Optional)</td>
</tr>
<tr>
<td>MID</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MBODY</td>
<td></td>
<td>M</td>
<td>O</td>
</tr>
</tbody>
</table>

Defines the message type. Possible values are:
- SCI_SOM_DOCU = actual document
- SCI_SOM_STAT = status confirmation

Thus, you can find out in SciSendMsg whether a document or status is to be sent.

This must be the first attribute set in SciRecvMsg.

End of message. This attribute must always be the last attribute set in SciRecvMsg.

A value does not have to be assigned (pAttr=NULL).

SAPcomm-unique message ID defined by the internal part. The value requested in SciSendMsg must be reset in SciRecvMsg for status confirmations for identification of the message concerned (as the first attribute after SCI_A_SOM, before all other attributes).

Switches to the next body part of a message. Value of the attribute is a reference to an SCI_MBODY structure.

CAUTION: The present version of the SAPcomm API only supports text, Fax3 and binary type body parts. In particular, "forwarded IPMs" and therefore nested messages are currently not implemented.
### Message Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>M</th>
<th>O</th>
<th>O</th>
<th>O</th>
<th>M</th>
<th>O</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEBODY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>End of the body part list. A value does not have to be assigned (pAttr=NULL).</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MORG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Creator of the message. Value of the attribute is a reference to an SCI_ADDRESS structure.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MFORWARDER</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sender of the message. Value of the attribute is a reference to an SCI_ADDRESS structure.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAUTHORIZER</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Person who authorized the sending of the message. Value of the attribute is a reference to an SCI_ADDRESS structure.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MRECIP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Recipient of the message. Value of the attribute is a reference to an SCI_MRECIP structure. This attribute is only a multiple attribute for mail services. If, on the other hand, the first recipient has address type fax or telex, the attribute can only be called once!</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MerERECIP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>End of the recipient list. A value does not have to be assigned. (pAttr=NULL).</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ORGID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A unique ID allocated by the external system that generated the message.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUBJECT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Title of the message.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRIORITY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Priority of the message: 1..9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMPORTANCE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Importance of the message: 1..9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>APPLICATION</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type of message contents. Possible values are: SCI_APP_EDI, SCI_APP_MAIL, SCI_APP_TELE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COMMTYPE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAP EDI communication service.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CHANNEL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Target channel via which the document is to be sent.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OCHANNEL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

April 2001  213
Message Attributes

Channel via which the status message is to be sent.

**SCHEDULE**

Send time for forwarding within SAPcomm:
- SCI_SCH_IMMEDIATE = immediate
- SCI_SCH_NIGHT = at the night time configured
- SCI_SCH_SCHEDULE = according to the time plan

Send time for the external system:
- SCI_SCH_IMMEDIATE = immediate (default)
- SCI_SCH_NIGHT = at night rate
- SCI_SCH_SPECIFIED = specified in

Send date (only for SENDSCHED = SCI_SCH_SPECIFIED).

Send time (only for SENDSCHED = SCI_SCH_SPECIFIED).

Expiry date.

Specification of an earlier message to which the present message is responding.

SAP return code of the message transmission (for possible values, see below).

Original return code or error message of the external system.

Charges for message transmission.

Date of the message transmission

Time of the message transmission.

Unique ID allocated by the external system that has received the message.

214  April 2001
Message Attributes

Separate ID reported by the recipient. Value of the attribute is a reference to an SCI_ADDRESS structure.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Mandatory</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOSTNAME</td>
<td></td>
<td></td>
<td>Name of the host on which the external system (SAP), which generated the message, is running.</td>
</tr>
<tr>
<td>SAPSYSTEM</td>
<td>M</td>
<td>M</td>
<td>Name of the SAP System that generated the message.</td>
</tr>
<tr>
<td>SAPCLIENT</td>
<td>M</td>
<td>M</td>
<td>Client in the SAP System, in which the message was generated.</td>
</tr>
<tr>
<td>CORRDATA</td>
<td>O</td>
<td></td>
<td>Archive numbers assigned to the message in the SAP System (only for SCI_APP_EDI)</td>
</tr>
<tr>
<td>ECORRDATA</td>
<td>O</td>
<td></td>
<td>End of archive number list. A value does not have to be assigned (pAttr=NULL)</td>
</tr>
</tbody>
</table>
SAP Return Codes for Message Transmission

SCI_SUCCESSFUL  The message was transferred correctly.
SCI_CONFIG_ERROR A configuration exists which is unsuitable for message transmission.
SCI_INV_MSG_ATTR  A message attribute is not allowed or is missing.
SCI_MSG_FORMAT  The format of the message contents is not allowed.
SCI_MSG_TOO_LARGE The message is too large.
SCI_TOO_MANY_RECIP There are too many recipients.
SCI_INV_REC_ADDRESS Invalid recipient address.
SCI_INV_ORG_ADDRESS Invalid creator address.
SCI_INV_FOR_ADDRESS Invalid address (forwarder).
SCI_INV_AUT_ADDRESS Invalid address (authorizer).
SCI_NOT_AVAILABLE A component required for message transmission is not available.
SCI_ROUTING_ERROR Error finding connection route to recipient.
SCI_LINE_BUSY The connection to the recipient was busy for all dial attempts.
SCI_NO_CONNECTION A connection could not be made to the recipient.
SCI_ANSWERBACK A differing ID was reported by the recipient.
SCI_DEVICE_INCOMP The end-device of the recipient is not suitable for message transmission.
SCI_TRANSM_ABBORT Transmission of the message was cancelled.
SCI_UNKNOWN_ERROR Unknown error.
Compiling and Linking the Example Program

The source code for the SAPcomm component TST is contained in the SAPcomm delivery package. You can use this example program (scmitst.c) to study the use of the functions of the SAPcomm Development Library.

You generate a SAPcomm communications component from the example program by compiling it and linking the generated object file with the SAPcomm Development Library.

**Under UNIX**

First compile program scmitst.c (in ANSI mode, here under HP-UX):
```
c c -Aa -c scmitst.c
```
Then link the object file generated with the SAPcomm programming library scmiux.o:
```
c c -o scmitst scmitst.o scmiux.o
```

**Under OS/2**

The compiler and linker statements for OS/2 (for the C compiler IBM C Set ++) are as follows:

First compile program scmitst.c:
```
icc -c -Sm -Gm+ -J- scmitst.c
```
Then link the object file generated with the SAPcomm programming library scmio2.lib:
```
icc /B"PM:VIO /A:4 /ST:8192" scmitst.obj scmio2.lib
```
Example Configuration

This topic contains a complete example configuration for the example program for a connection to an SAP System.

*******************************************************
* SAPcomm control component *
*******************************************************
PROGRAM SCB
  ROUTING SAPCOMM TLX SERVICE TLX NUMBER 12345 CHANNEL C-CLIENT1
  END-NUMBER CHANNEL C-CLIENT3
  END-SERVICE
  END-ROUTING
  TRACE ON FILES ON
  END-TRACE
END-PROGRAM

*******************************************************
* Communication component TST *
*******************************************************
PROGRAM TST
  PERIODIC PM-10MINUTES
    INTERVAL 10
    FROM 13:00
    UNTIL 17:00
  END-PERIODIC
  TRACE ON DATA_FLOW ON CS_RESPONSE ON
  END-TRACE
  CHANNEL TSTGOOD SENDRC 0 PERFORM SEND0 ACTION SEND-SCHEDULE PERIODIC PM-10MINUTES
  END-PERFORM
  END-CHANNEL
  CHANNEL TSTBAD SENDRC 1 PERFORM SEND1 ACTION SEND-SCHEDULE PERIODIC PM-10MINUTES
  END-PERFORM
  END-CHANNEL
  CHANNEL TSTIN RECEIVE testrec.txt RECIPIENT 12345 PERFORM RECV ACTION RECEIVE PERIODIC PM-10MINUTES
  END-PERFORM
  END-CHANNEL
END-PROGRAM
************ Communication component SAP ************

PROGRAM SAP

SELECTION SELECT-TELE
APPLICATION TELE
END-SELECTION

PERIODIC PM-20MINUTES
INTERVAL 20
FROM 13:00
UNTIL 17:00
END-PERIODIC

DESTINATION D-CLIENT1
SYSTEM TESTSYS
RELEASE R2-50B
CLIENT 1
USER CPIC4
PASSWORD HELLO
LANGUAGE D
END-DESTINATION

DESTINATION D-CLIENT3
SYSTEM TESTSYS
RELEASE R2-50B
CLIENT 3
USER CPIC4
PASSWORD HELLO
LANGUAGE D
END-DESTINATION

CHANNEL C-CLIENT1
DESTINATION D-CLIENT1
PERFORM UPLOAD
ACTION SEND-SCHEDULE
PERIODIC PM-20MINUTES
END-PERFORM

PERFORM DOWNLOAD
ACTION RECEIVE
TIME 23:00
SELECTION SELECT-TELE
END-PERFORM

END-CHANNEL

CHANNEL C-CLIENT3
DESTINATION D-CLIENT3
PERFORM UPLOAD
ACTION SEND-SCHEDULE
PERIODIC PM-20MINUTES
END-PERFORM

END-CHANNEL

END-PROGRAM
Working With the Graphical Interface

- The Graphical Interface: Basics [Seite 221]
- Displaying Channel-related Activities [Seite 222]
- Using the SAPcomm Menu Bar [Seite 224]
The Graphical Interface: Basics

You can operate the SAP communications server with or without the graphical interface.

To run the communications server with the graphical interface, the following environment is required:

UNIX: Motif
OS/2: Presentation Manager

Program sapcomm is responsible for setting up and managing the “SAPcomm window”.

You can use this program as follows:

- Start the SAP communications server (refer to Starting the Communications Server [Seite 85])
- Start the graphical interface during operation.
  (In this case, program sapcomm will not start the control component.)

Any errors that occur when starting the graphical interface, or during its operation, are logged in the file sapcomm.err. This file is created in the work directory of the control component SCB.

Any errors affecting the control component or other components of the communications server are logged in the individual log files of the communications server.

The graphical interface of the communications server allows you to:

- Display channel-related activities
  At present, you can display up to 15 channels. If you want to configure more than this, you must use SAPcomm without the graphical user interface.

- Use the SAPcomm menu bar
  With the menu bar, you can change certain configuration parameters dynamically, as well as start the activities of the communications server defined in the configuration file sapcomm.cfg.
Displaying Channel-related Activities

This topic describes the individual elements of the information display for channel-related activities.

Example:

<table>
<thead>
<tr>
<th>CHANNEL</th>
<th>STATUS</th>
<th>OUT</th>
<th>IN</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP1</td>
<td>ACK</td>
<td>--.--</td>
<td>15:24-15:26</td>
</tr>
<tr>
<td>IE-NET</td>
<td>CONNECTED</td>
<td>--.--</td>
<td>--.--</td>
</tr>
<tr>
<td>MARKIII</td>
<td>READY</td>
<td>15:22-15:28</td>
<td>--.--</td>
</tr>
<tr>
<td>MAIL</td>
<td>READY</td>
<td>--.--</td>
<td>--.--</td>
</tr>
</tbody>
</table>

The individual columns of the SAPcomm window can be explained as follows:

**CHANNEL Column**
A separate line is assigned to every configured channel in the SAPcomm window. This column displays the channel name.

**STATUS Column**
Displays the actions of the communications component for each channel. Information on data exchange with the target system is displayed here for communications components that carry out data transfer synchronously (for example, SAP). For all other communications components, only information on the interaction between the communications component and the respective linked communications program is displayed.

The STATUS values have the following meaning:

- **READY**
  The communications component is waiting. It is ready to receive a new request from the control component.

- **CONNECTED**
  Only for synchronous data transfer. The communications program has set up the link to the target system.

- **SEND**
  The communications component is editing a request to sending data.
  In the case of synchronous data transfer, this means that data transfer is in progress. During non-synchronous transfer, it merely means that the communications component reprocesses the order for the linked communications program and forwards it.

- **CALL**
  The communications component processes a request to set up a link to a remote communications partner to request incoming messages.
  In the case of synchronous data transfer, this means that the request process is in progress. During non-synchronous transfer, it merely means that the communications component reprocesses the order for the linked communications program and forwards it.

- **CHECK**
  The communications component evaluates the response file generated by the communications program.

- **SYNCH**

April 2001
The communications component must interrupt its activity with the other components of the communications server for synchronization reasons and then waits.

**ACK**
Only for synchronous data transfer. The communications component has transferred data to the target system and is waiting for a confirmation.

**ADMIN**
The communications component carries out an internal administrative function.

**INACTIVE**
The communications program assigned to the respective channel is not ready (for example, ODEX: the driver ODEDRV01 is inactive; ExpEDite/OS2: the link to the Information Exchange cannot be established despite restart attempts - manual intervention necessary).

**OUT Column**
This channel displays information about the times of transfer activities via the individual channels.
Left time specification:
Time that a transfer request was last processed.
Right time specification:
Next time at which a transfer request will be transferred to the communications component.

**IN Column**
This channel displays information about the times of receive activities via the individual channels.
Left time specification:
Last time at which a document was received from the relevant communications component.
Right time specification:
Next time at which an order for requesting incoming messages will be transferred to the communications component.
Using the SAPcomm Menu Bar

The SAPcomm menu bar allows you to:

- change certain SAPcomm configuration parameters dynamically
- start the communications server activities defined in the configuration file sapcomm.cfg, regardless of the time plan.

To do this, you activate one or more of the following submenu options:

→ **Perform**
  → **Administration**
  Start administrative functions. Using two selection lists, you can first select a communications component and then an administrative function configured for the selected component (a PERFORM parameter set).
  
  → **Transfer**
  Start data transfer functions. Using two selection lists, you can first select a channel and then a function configured for the selected channel (a PERFORM parameter set).

→ **Configuration**
  → **Log Window**
  Changeover switch. Open or close a window for the current display of SCB log entries (OS/2 only).
  
  → **Logging**
  Set logging behavior. You can first select a SAPcomm component using a selection list and then the logging level for this component.
  
  → **Log QMessages**
  Changeover switch. Log information which is sent to the control component from other components of the communications server (internal SAP use).
  
  → **Change Log File Font**
  Change the font used for the online display of the SCB log entries (OS/2 only).
  
  → **Time Plan**
  Enter the current time plan in the SCB log file.

→ **Exit**
  → **Shutdown**
  Shut down the communications server. Please read Closing the Communications Server [Seite 86].
  
  → **Abort**
  Abnormal termination of the communications server activities. Please read Closing the Communications Server [Seite 86].
The SAPcomm Directory Tree

Überblick

ETC-Verzeichnis [Seite 229]

Overview [Seite 226]

The SAPcomm Root Directory [Seite 227]

The SAPcomm System Directory [Seite 228]

The Component Directories [Seite 230]
Overview

The files belonging to SAPcomm (programs, configuration files, log files, transmission files) are stored in a directory tree structure. The individual directories are created during installation and some are created by SAPcomm itself.

The structure of the directory tree is as follows:
The SAPcomm Root Directory

The root directory is called:

**UNIX:** /usr/sap/com  
**OS/2:** \sap\com

The "relative" names of the subdirectories (relative to the root directory) are identical for OS/2 and UNIX. For example: "SYS" (see below) is the relative name of the SAPcomm system directory and is therefore called:

**UNIX:** /usr/sap/com/SYS  
**OS/2:** \sap\com\SYS

The names of the remaining subdirectories of the SAPcomm root directory are constructed in a similar way.
The SAPcomm System Directory
Name: SYS.
The system directory does not contain any files, just subdirectories: The SAPcomm program directory and the SAPcomm profile directory.

The SAPcomm Program Directory
Name: exe.
The program directory does not contain any files, just subdirectories: the debug directory, the production directory and the SAPcomm call directory.

The Debug Directory
Name: dbg
This directory is not currently used.

The Production Directory
Name: opt
This directory is only used by SAPcomm in UNIX versions.
The production directory contains:
- Programs belonging to the communications server (sapcomm, scmbasis, etc.)
- The SAPcomm auxiliary programs (scmlog, etc).
- All components belonging to SAPcomm-API.

The Call Directory
Name: run
The SAPcomm call directory contains:
UNIX: Links to the programs stored in the production directory;
OS/2: The programs and components mentioned in the above topic on the production directory.

The Profile Directory
Name: profile.
The profile directory contains:
- The configuration file sapcomm.cfg
- The profile file sapcomm.pfl
- All grammar files belonging to the individual components
- Information on SAPcomm (supplements to documentation, and so on).
ETC Directory
After installation, you will find the following files in the ETC directory:

- The model configuration file `sapcomm.cfg`
  Copy it to the SAPcomm profile directory, if a SAPcomm configuration file not yet exits.

  ![Note](image)

  Modifications to SAPcomm configuration only take effect if they are performed on the configuration file stored in the profile directory.

- Files that are not essential for operation of SAPcomm (sample source file, etc.)
The Component Directories

Name: the **three-character identifier** (in lower case) appropriate to the respective component.

In each case, there are separate directories for every component of the communications server (that is, the control component and communications components). You do not need to create these directories: They are created automatically by the communications server, if they do not already exist.

These directories do not contain any files, just the following three subdirectories.

**The Work Directory**

Name: **work**.

After the program start, every component makes "its" work directory the current directory. Files are temporarily buffered here, if necessary.

This directory contains the following subdirectory:

**The Work Archive Directory**

Name: **arc**.

Files saved for the purpose of later analysis are stored in the work archive directory. Example: Response files with error messages created by the connected communications program.

**The Log Directory**

Name: **log**.

A separate log file is written by every component of the communications server. This file is created in the log directory belonging to the respective component under the following name:

    Lyymmdd.kkk

with yy=year, mm=month, dd=day and kkk=identifier of the component. At the end of the day, the current log file is closed and a new one (with a correspondingly changed name) is created.

You can make the communications server automatically delete old log files.

For more information, read the notes on the configuration parameter LOGFILE_EXPIRATION in Structure of the Configuration File [Seite 37].

The entries in the log files are in text format and contain the respective time of the entry, the category (I=Information, W=Warning and E=Error), an indicator and text.

You can specify the range of the log entries in the configuration file yourself using the TRACE parameter sets.

You can use the usual operating system commands or editor programs to display and analyze the log files. Furthermore, a display and search program specially designed for the SAPcomm log files is provided in program **scmlog** (see The SAPcomm Auxiliary Programs [Seite 232]).

**The Data Directory**

Name: **data**.

Every communications component has just one data directory. However, this directory itself does not contain any files, but a separate channel directory for every channel allocated to the communications component and the processing directory.

**The Processing Directory**

Name: **busy**.
Files for messages that have been transferred to the linked communications program for
sending, but for which there is no final confirmation ("sent" or "error"), are stored in the
processing directory.

The Channel Directories
Name: The channel name entered in the configuration file (in lower case).
Every document to be sent is uniquely allocated to a channel ("incoming documents" are also
generally considered "to be sent" from the point of view of the communications server). The
document is stored in the channel directory allocated for this channel.
This directory contains the following subdirectory:

The Channel Archive Directory
Name: arc.
Files with send dates, and associated files, which are not deleted after sending but saved due to
a corresponding entry in the configuration file, or files for which an error occurred and are
therefore saved for later analysis, are stored in the channel archive directory.
The SAPcomm Auxiliary Programs

Overview [Seite 233]
Analyzing SAPcomm Parameter Files (scmactl) [Seite 234]
Testing a Serial Interface (scmasyn) [Seite 235]
Checking the Configuration Data (scmcfgcc) [Seite 236]
Analyzing SAPcomm Log Files (scmlog) [Seite 239]
SAPcomm Monitor (scmmon) [Seite 240]
Moving Files (scmmove) [Seite 241]
Performing Actions (scmperfm) [Seite 242]
Testing the Network Link to the SAP System (scmping) [Seite 243]
Displaying SAPcomm Files (scmqueue) [Seite 244]
Reorganizing SAPcomm Files (scmreorg) [Seite 245]
Stopping the Communications Server (scmstop) [Seite 246]
Overview

SAPcomm auxiliary programs support your work with the SAP communications server. The description of these programs here is limited to the main applications. Not all call options or call parameters are listed.

A help option -h (for example, `scmlog -h`) is available for all programs. You can use this to display the full list of call options and call parameters, as well as current changes.
Analyzing SAPcomm Parameter Files (scmactl)

The program *scmactl* checks the relevant parameter file for every specified SAPcomm message ID and displays its contents:

- Routing information (via which CHANNEL the message arrived, or to which CHANNEL the message went or goes).
- Sender address (if relevant).
- Recipient address or list
- Mail attributes

For information about how to activate these functions when calling the program *scmactl*, please read the possible entries help (option *-h*).

The program *scmactl* supports the localization of problems in message forwarding. Interpreting the displayed data on the basis of existing documentation is only possible up to a certain point. Comprehensive interpretation can only be carried out fully at SAP.
Testing a Serial Interface (scmasyn)

This program checks that external devices are correctly connected via a serial, asynchronous V.24 interface (RS232). To do this, commands entered via the keyboard or existing files can be sent to the device manually.

Data received is stored in a file. Reception ends as soon as no character has been received for 3 seconds. The data is then displayed on the screen.

You can set the relevant transmission parameters (UNIX device names and transmission speed) via options, or take them from the SAPcomm configuration (parameters PORT and BAUD). Please note the possible entries help for scmasyn (option -h).

Example: To test the correct link to a TOPCALL device connected to /dev/ttys/ttyc04, you can retrieve the contents of the TOPCALL drive B:

scmasyn ..i,d=b
Checking the Configuration Data (scmcfgcc)

Program `scmcfgcc` tests the syntax and contents of configuration data in the file `sapcomm.cfg`. To do this, it reads the grammar files in the SAPcomm profile directory for the installed communications components and creates the binary configuration file `sapcomm.bcf` required for the operation of the communications server.

The communications server starts program `scmcfgcc` automatically if the file `sapcomm.bcf` does not exist or is "older" than the file `sapcomm.cfg`.

However, you should start the program manually in the following cases:

- The communications server finds a configuration error (SCB log file) and cannot be booted. (Error message: "Binary configuration file can't be generated")
- You are editing the configuration file `sapcomm.cfg` and make relatively significant changes which you want to check before starting the communications server.

Start the program by entering the following command on the command line:

```
scmcfgcc
```

First, the program reads in all the grammar files from the SAPcomm profile directory and displays them. Then, it reads in the configuration file `sapcomm.cfg` and attempts to generate the binary configuration file `sapcomm.bcf` from this.

If `scmcfgcc` ends without error messages, your SAPcomm configuration file is formally correct and you can start SAPcomm. The output of `scmcfgcc` is typically as follows in this case:

```
Grammar checker alpha version 0.009
Parsing grammar file : /usr/sap/com/SYS/profile/scmiapp.grm
Parsing grammar file : /usr/sap/com/SYS/profile/scmicx4.grm
Parsing grammar file : /usr/sap/com/SYS/profile/scmiodx.grm
Parsing grammar file : /usr/sap/com/SYS/profile/scmisap.grm
Parsing grammar file : /usr/sap/com/SYS/profile/scmiscb.grm
Parsing grammar file : /usr/sap/com/SYS/profile/scmitop.grm
Parsing grammar file : /usr/sap/com/SYS/profile/scmitst.grm
Start to parse configuration file /usr/sap/com/SYS/profile/sapcomm.cfg
```

In the event of an error, the configuration compiler gives tips for correcting configuration problems via error messages. You should take particular note in the error messages of the difference between the two following error types:

- Error in one of the grammar files.
  The configuration compiler displays which grammar files contain errors, but you cannot solve this kind of problem at once. Initially, you can only remove (rename) the incorrect grammar file, delete all entries belonging to the communications component in the file `sapcomm.cfg`, and restart the configuration compiler. Please consult the SAP Hotline about the incorrect grammar file.

- Error in configuration data.
  Take careful note of the specific error messages issued by the configuration compiler when correcting your entries in the configuration file `sapcomm.cfg`. See below for a list of the possible error messages and their causes.

### Configuration Compiler Error Messages

Potential error messages are issued from `scmcfgcc` after the line

```
Start to parse configuration file /usr/sap/com/SYS/profile/sapcomm.cfg
```
In the event of several error messages, only the first message is significant generally, the others being mostly consequent errors. Blank lines and comment lines are included in the line numbers declared.

**SYNTAX error - Resynchronization**
Can't parse token *Text* at line *Line number*
A variable with the name *Name* has already been declared
No sapcomm.bcf file generated
Possible causes:
- you have used the expression *Text* in the declared *Line number*, which is not allowed there. There may be a spelling mistake.
- before the *Line number* declared, you have forgotten to end a parameter group, e.g. with END-PROGRAM, END-DESTINATION, etc.

**Invalid input format (parameter *Value* at line *Line number**)
No sapcomm.bcf file generated
Possible causes:
- you have declared an inappropriate *Value* for a parameter with a specific quantity of allowed values (e.g. ACTION or DAY). There may be a spelling error.

**Error : SC001 : Parameter required:** *Parameter*
**SYNTAX error - Resynchronization**
Can't parse token *Text* at line *Line number*
A variable with the name *Name* has already been declared
No sapcomm.bcf file generated
Possible causes:
- you have not declared a mandatory *Parameter*.

**Error : SC002 : Parameter ambiguous:** *Parameter*
A variable with the name *Name* has already been declared
No sapcomm.bcf file generated
Possible causes:
- you have declared a *Parameter* which can only be declared once several times.

**No entry *Value* for parameter *Parameter* was found in the global table**
A variable with the name *Name* has already been declared
No sapcomm.bcf file generated
Possible causes:
- you have referred to a non-existent parameter group *Value* for a *Parameter*. There may be a spelling error.

**The routing channel *Channel* on line *Line number* is not declared**
A variable with the name *Name* has already been declared
No sapcomm.bcf file generated
Possible causes:
• in the parameter group ROUTING, you have declared a *Channel* for a parameter TO-CHANNEL, which does not exist as a parameter group CHANNEL. There may be a spelling mistake.
Analyzing SAPcomm Log Files (scmlog)

The program `scmlog` supports the analysis of log files generated by SAPcomm. To specify the log file, enter one of the following during the program call:

- The three-character identifier of the relevant communications component (in this case, `scmlog` independently accesses the log file for the current date)
- File and path name.

The program `scmlog` supports different selection criteria, such as:

- Error entries
- Warnings
- The logging categories that can be specified in the SAPcomm configuration file
- Entries that were generated at a specified time

Only the last eleven selected entries are displayed. You can, however, also use `scmlog` as a log monitor, i.e. to display each entry as it is created.

For more information on how to specify the functions entered here when calling the program `scmlog`, refer to the possible entries help (option `-h`).
SAPcomm Monitor (scmon)

You can use this program to display a part of the storage area used by the communications server in an edited form. The displayed data can only be partially interpreted on the basis of the present documentation. Full interpretation can only be carried out internally at SAP.
Moving Files (scmmove)

The program `scmmove` moves files generated by SAPcomm (transmission and parameter files) from a source directory to a target directory and thus supports the following functions:

- "Extraction" of all files to be sent via a specified SAPcomm channel and their storage in a non-SAP directory on the hard disk or a floppy disk. This function also allows you to send documents if the network link to your communications partner is interrupted.

- "Regeneration" of archived documents. Documents that cannot be sent are stored in the archive directory assigned to their channel. When the cause of the send failure has been corrected, these documents can be released for a further send attempt using the program `scmmove`.

Using `scmmove` can interfere with the operation of the communications server. Therefore, you should start this program only if the communications server is not active.

Using the possible entries help (option -h), you can view how the individual functions of the program are activated.
Performing Actions (scmperfm)

With program `scmperfm` you can make SAPcomm perform actions configured in the SAPcomm configuration file (PERFORM parameter groups).

This is the equivalent of using the SAPcomm graphical user interface to instruct actions.
Testing the Network Link to the SAP System (scmping)

The program `scmping` allows you to check whether the parameters relevant to your local network (for example, in the Communications Manager) are maintained correctly.

The program establishes a CPI-C link with an SAP System, logs on to that system and cancels the link again - without exchanging data.

At the program start, you determine which SAP System is accessed by entering a DESTINATION parameter set from the configuration file `sapcomm.cfg` as a call parameter.

For example: you start the program with the option `-D SAP1`, if SAP1 is a DESTINATION parameter set containing the necessary specifications (SYSTEM, etc.) for the setup with an SAP System.

You can override some of these parameters with call options. Display the program's possible entries help for details.
Displaying SAPcomm Files (scmqueue)

You can use the program scmqueue to display an overview of files currently being processed by SAPcomm.
Reorganizing SAPcomm Files (scmreorg)

The program `scmreorg` deletes out-of-date files generated by SAPcomm. The files to be deleted are selected by various options:

- Minimum age in days
- Channel
- Component
- File type (archived files,...)

If in doubt, use the option `-l` (list). This option does not delete the selected files, but merely creates a list of files to be deleted.
Stopping the Communications Server (scmstop)

You can use this program to shut down the communications server "cleanly" or stop the server with all active communications components. For further details on shutting down the server "cleanly", refer to Closing the Communications Server [Seite 86].

Always use the program scmstop to terminate the communications server if you are running it without the graphical interface.

Please note that the program scmstop merely sends a shutdown command to the communications server. It does not necessarily mean that the communications server has also already ended. You must check this via the log file of control component SCB. (To do this, use the program scmlog described above.)

If one of the SAPcomm components is blocked, you cannot exit SAPcomm. Abort SAPcomm using the program scmstop (with appropriate wait time) and restart SAPcomm specifying the -a option (for aborted).

You can also terminate individual communications components separately. To do this, you enter the three-character identifier of the component when calling the program scmstop. This option is exclusively provided for the event that the control component is locked or for some other reason cannot terminate one of more of the active communications components.

For information on how the functions entered here must be specified when calling the program scmstop, please refer to the possible entries help (option -h).