SAPphone
Telephony in R/3

Integrating R/3 with Your
CTI System -
Interface Description

SAPphone Interface Version 4.00A / 5.00A / 5.01ASP

Document Version 5.01A2

Copyright

History

Version	Date	Changes	
1.00	**April 7, 1997**		
4.00A1	**May 22, 1998**		
4.00A2	**May 29, 1998**		
4.00A3	**July 10, 1998**		
4.00A4	**September 22, 1998**		
5.00A1	**December 8, 1998**		
5.01A1	**March 24, 1999**	Chapter 10.1.18	Returncode 007 no longer valid, Returncode 052 and 053 now valid
		Chapter 10.1.19	Returncode 007 no longer valid, Returncode 052 and 053 now valid
		Chapter 10.1.20	Returncode 007 no longer valid, Returncode 052 and 053 now valid
		Chapter 10.1.23	Returncode 007 no longer valid, Returncode 052 and 053 now valid
		Chapter 10.1.26	Returncode 007 and 008 no longer valid
		Chapter 10.3.4	Returncode 051 replaced by returncode 020
		Chapter 11	New returncodes 052 and 053
		Chapter 5.2	Architecture with two RFC destinations will not be realized in the actual release
		Chapter 10.3.1	New import parameter LOGSYS
		Chapter 10.3.3	New export parameter ERR_DETAILED
		Chapter 10.3	Include description of functions: - SPS_DELETE_PDCALL - SPS_CAMPAIGNS_GET - SPS_ASSIGNED_CAMPAIGNS_GET
		Chapter 4.6	More detailed explanation of functionality
		Chapter 9.3	Constants for agent workmodes, reasons for inbound calls and call connection states now two digits (including leading zero)
		Chapter 7	Monitoring of external components
		Chapter 9.1	New structures for monitoring of external components
		Chapter 9.2	New data elements for monitoring of external components
		Chapter 9.3	New constants for monitoring of external components
		Chapter 10.6	Functions for monitoring of external components
		Chapter 10.4.1	Parameter CALLINFO deleted, info now in CALLDATA
		Chapter 10.4.4	Wildcard symbol for instance number 9999 instead of *
		Chapter 10.1.10	Parameter TYPE for SPS_GETQUEUES
		Chapter 10.1.11	Valid values for parameter WORKMODE
		Chapter 4.1	New Chapter: Registration
		Chapter 8	New Chapter: Certification
		Chapter 12	New Chapter: Delta between interface versions
		Chapter 9.2	Data element SP_INFO1 shortened to 50 character

Version	Date	Changes	
5.01A2	**April 26, 1999**	Chapter 4	Introduction: Call model from SAPphone perspective
		Chapter 4.6	Drop party to drop oneself from conference: second call still on hold
		Chapter 6.3	Keyname in example: REFOBJTYPE instead of OBJTYPE
		Chapter 8	- Call of any function should not result in break down, - R/3 wide telephony support for certification, - functions implemented in R/3 or gateway
		Chapter 9.1.14	New: structure SPH_QUEUES (data element SPH_QUEUES deleted)
		Chapter 9.1.15	New: SPH_CAMPGS
		Chapter 9.2	Column with valid values
		Chapter 9.3.9	Values for TYPE paremter for SPS_GETQUEUES
		Chapter 10	Introduction: structure of returncodes, import/export from implementers perspective
		Chapter 10.3.4	Parameter PDSTATISTC is EXPORT TABLES
		Chapter 10.4	Parameter CALLDATA is EXPORT/IMPORT TABLE in 10.4.2, 10.4.3, 10.4.4

Glossary

ACD	**A**utomatic **C**all **D**istribution
Agent	Someone who works in a call center.
API	**A**pplication **P**rogramming **I**nterface
Canonical numbers	Phone numbers in the form +CC (AREA) SUBSCRIBER-EXTENSION (e.g. +49 (6227) 34-1234)
CIC	**C**ustomer **I**nteraction **C**enter: R/3 work environment for Call Center Agents
CTI	**C**omputer **T**elephone **I**ntegration
Express message	Popup window that is displayed in a new session after the next user action in the R/3 System.
Telephony gateway	Software component that maps SAPphone functions and the corresponding functions in the CTI system.
IVR	**I**nteractive **V**oice **R**esponse
Main line	Direct connection from a phone to the phone network, without a connection to a phone system
Outside line digit	For extensions: digit that must be dialed in order to obtain an outside line.
PBX	**P**rivate **B**ranch e**X**change Phone system (extension phone system, telecommunications equipment)
Phone functionality	Functions that can be executed on the phone, such as initiating a call, ending a call, accepting a call, forwarding a call, etc.
Phone system	All the components that may be part of a telephony infrastructure, e.g. phone, PBX, phone network, etc.
Registration mode	Technique for calling RFC programs on external computers (see SAP RFC documentation).
RFC	**R**emote **F**unction **C**all Technology that can be used to call and execute functions or procedures on remote SAP systems or other computers (see SAP RFC documentation).
RFC component	The software that forms the foundation for communications via RFC.
SAPphone RFC interface	All the interfaces of functions that are either called in the R/3 System from external computers or are required on external computers by R/3 in order to provide the functionality available in SAPphone.
SAPphone server	A program provided by SAP that functions as the TAPI client and adapts the SAPphone RFC interface to TAPI.
Service provider	In connection with TAPI: telephony software that provides the phone functionality and can be used by a client.
Service number	Uniform phone number for calls to service functions (e.g. IVR-based information services), which can then be forwarded to individual extensions.
Start session	Technique for calling RFC programs on external computers (see SAP RFC documentation).
TAPI	**T**elephony **A**pplication **P**rogramming **I**nterface Interface defined by Microsoft for implementing phone integration applications

1 Introduction

1.1 *Document Version and R/3 Release*

The present document describes versions 5.00A and 5.01ASP of the R/3 SAPphone interface (see below). The interface version 5.00A is shipped with R/3 release 4.5B and is the first version that is part of the SAP Complementary Software certification program. The document also contains information about the SAPphone interface version 4.00A shipped with 4.5A. Changes between the two versions are highlighted. For a specification of earlier versions of the interface, please refer to version 1.00 of the current document .

1.2 *The R/3 Telephony Interface*

All communication between R/3 applications or the R/3 Customer Interaction Center on the one hand, and external CTI systems, Call Centers or PBX drivers on the other hand is via SAPphone, the R/3 telephony interface.
Throughout this document, any component which communicates with R/3 through the SAPphone interface on the one hand and a CTI system on the other, is referred to as a “telephony gateway”, see figure 1 below. “CTI systems” may range from “just” a CTI enabled PBX to globally distributed call center solutions.

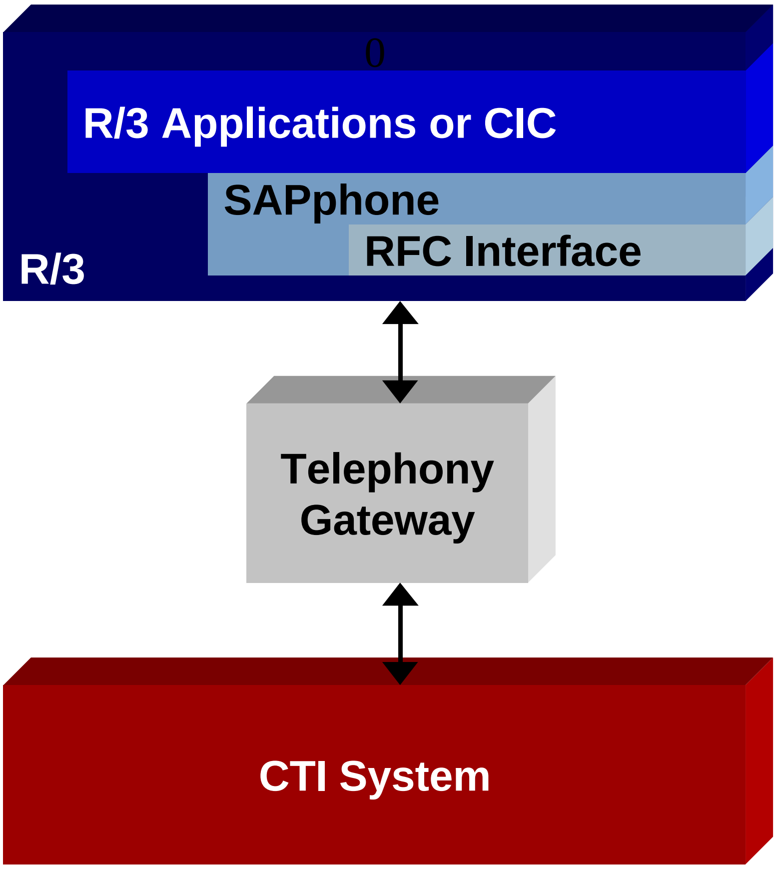


Figure 1: SAP R/3, telephony gateways and CTI systems

CTI systems may communicate with R/3 via their specific telephony gateway, provided by the vendor of the CTI system or some third party, or they may communicate via Microsoft TAPI with the SAP provided “SAPphone Server”. See chapter 3 for details.
Other interfaces, such as TSAPI, are not supported directly by SAP, but may be used in connection with third party CTI middleware.

1.3 SAPphone

SAPphone is designed to make the telephone an integral part of business processes, either in connection with individual R/3 applications, SAP Business Workflow or the SAP Customer Interaction Center (CIC) in a Call Center environment. SAPphone supports the following functions:

- Various telephony functions, such as initiate calls, transfer calls, etc (see list of function specifications in this document).
- Displaying incoming call information.
- Support for Call Center functions, such as Agent Login, etc.
- Support for campaigns (Predictive Dialing / Power Dialing).
- Branching to applications with caller data
- Entering memos for calls

SAPphone is does not perform the following functions:

- Processing and storing voice data (voice mail)
- Processing incoming calls that are made to a service number and assigning them to a specific processor (ACD functionality)
- Providing data from R/3 for incoming calls processed by an IVR system.

1.4 SAPphone Interface Version Numbering System

To ensure the compatibility of SAPphone and the connected telephony software, or at least ensure suitable error handling in cases of incompatibility, the SAPphone interface is assigned version numbers. The version number has the following structure:

A**.**BBCDD

A = Numeric, increased with each incompatible interface change.
B = Numeric, increased for compatible changes (such as enhancements to functionality).
C = Letter, increased for SAPphone internal changes that have no effect on the RFC interface
D = Additional R/3 internal information

The version of the SAPphone interface implemented in a given R/3 release is defined in the program LPHONTOP. The version number of the SAPphone interface is in general not identical with the R/3 release name. The SAPphone interface numbers are:

R/3 release	SAPphone interface number
3.1G/H	**3.01A**
3.1I	**3.03A**
4.0A/B	**3.40A**
4.5A	**4.00A**
4.5B	**5.00A**
4.6A	**5.01ASP**

The telephony software connected to SAPphone must also be assigned a version. The version number must have the following structure:

A**.**BBCDDD

The values A, B and C should agree with the version of SAPphone the telephony software was created for or most recently adapted to. The values for D can be freely assigned by the telephony software vendor. They are not relevant for determining compatibility.

Compatibility is verified by the SAPphone function XCHGVERSION (see chapter 10.1.1). For the gateway to be compatible

- The value A must be identical in the telephony software version and the SAPphone version
- The value BB can be higher than in the SAPphone version. If the gateway does not require the new features of a higher BB value, the value BB can also be lower than in the SAPphone version.
- The value C has no effect on compatibility

2 Architecture

In addition to SAPphone, phone integration requires software and hardware components outside the R/3 System. The architecture describes which components are required, how they are connected with each other and how they are distributed (on the work center PC, on a telephony server, or distributed between both. The following alternative architectures are supported:

- Local connection
  - Without centralized telephony server
  - Additional hardware and software required at work center
- Client/server connection
  - With centralized telephony server
  - Additional software required at work center
- Centralized connection
  - With centralized telephony server
  - No additional components required at work center

The various options and the components involved are described below. To simplify things, only one telephony server and one PBX are illustrated in the diagrams below, although architectures with several telephony servers and/or PBXs are also supported. Installations that consist of a combination of the architectures illustrated below are also supported.

2.1 Local Connection

In the local connection, the so-called "work center solution", each work center PC is connected with the phone system via hardware components. The diagram below illustrates three possibilities:

1. Direct connection between the work center PC and the phone, e.g. through a model cable with a V24 interface
2. Connecting the work center PC to the phone network via a modem
3. Connecting the work center PC to the phone network via a plug-in board

Access to the phone network can also be provided in several ways:

A) Connection via a PBX
B) Direct connection to a main line

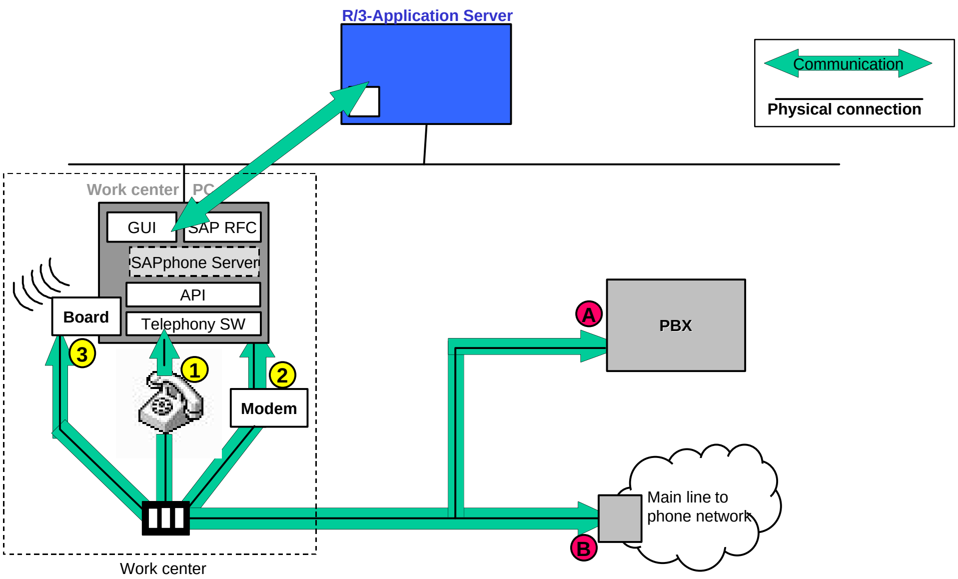


Figure 2: Local connection (3 possibilities)

In addition, each work center PC must be equipped with software components to establish the connection between the R/3 System and the phone system. For outgoing calls started from within an R/3 application, the

R/3 application server uses RFC functions to call the local software, which forwards the commands to the phone system components. For incoming calls, the phone system components report the call to the local software, which calls the corresponding functions on the R/3 application server.

2.2 Client/Server Connection

No hardware enhancements to the individual work center PCs are required for the connection using client/server technology. Instead, the central telephony server must be configured within the network, which provides access to the PBX. The installation can also encompass several PBXs and several telephony servers.

Such installations also enable the use of telephone products from other vendors, which are installed on the local PC, e.g. personal address books.

Both the work center PC and the telephony server must be equipped with additional software components. The following options are available:

- If the interface software that forms the connection between the external telephony software and SAPphone is not network-capable **-** that is, it cannot be installed on several distributed computers **-** the telephony software must also perform the communications via the local network. The interface software and a local component of the telephony software are installed on the work center PC, and the central component of the telephony software is installed on the telephony server (see Chapter 2.2.1).
- If the interface software is network-capable **-** i.e. can execute communications via the local network **-** only parts of this interface software need to be installed on the work center PC. The telephony software is installed on the telephony server, together with a central component of the interface software (see Chapter 2.2.2).

2.2.1 Connection without network enabled Interface Software

Work center PC:

- SAP RFC component for communicating with the R/3 System *and*
- Local telephony software for communicating with the telephony server *and*
- Interface for adapting the telephony software to the SAPphone RFC interface (either from a third-party vendor or the SAPphone Server)

Server:

- PBX-specific driver
- Central telephony software for connecting the individual work centers to the driver (via LAN)

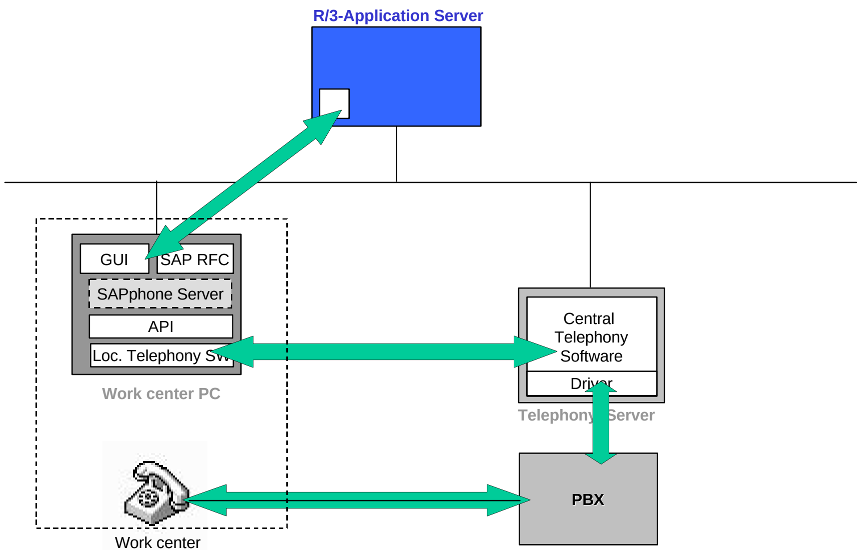


Figure 3: Client/server solution with telephony software component on the work center PC

2.2.2 Connection with network enabled Interface Software

Work center PC:

- SAP RFC component for communicating with the R/3 System *and*

- Interface for adapting the local API to the SAPphone RFC interface (either from a third-party vendor or the SAPphone Server) *and*
- Local API component

Server:

- Central API component *and*
- Central telephony software for communicating with the PBX

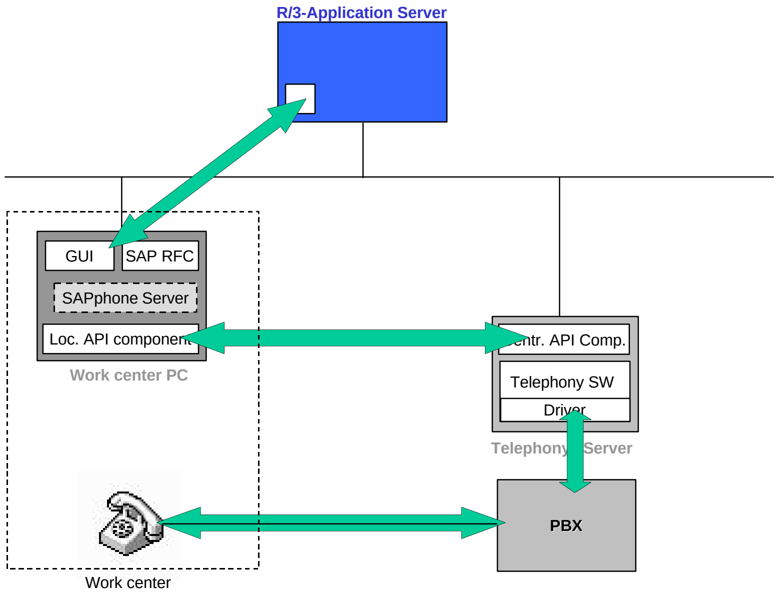


Figure 4: Client/server solution with connection via network-capable API

2.3 Centralized Connection

In a centralized architecture, the individual work center PCs do not require any additional hardware or software. The PBX is connected to a central telephony server. The telephony server communicates directly with the R/3 application server via the local network, without routing via the work center PC. To perform this task, the telephony server must be equipped with the following software components:

- Central RFC component for communicating with the R/3 application server *and*
- Interface for adapting the telephony software to the SAPphone RFC interface (either SAPphone Server or program from a third-party vendor)
- Central telephony software

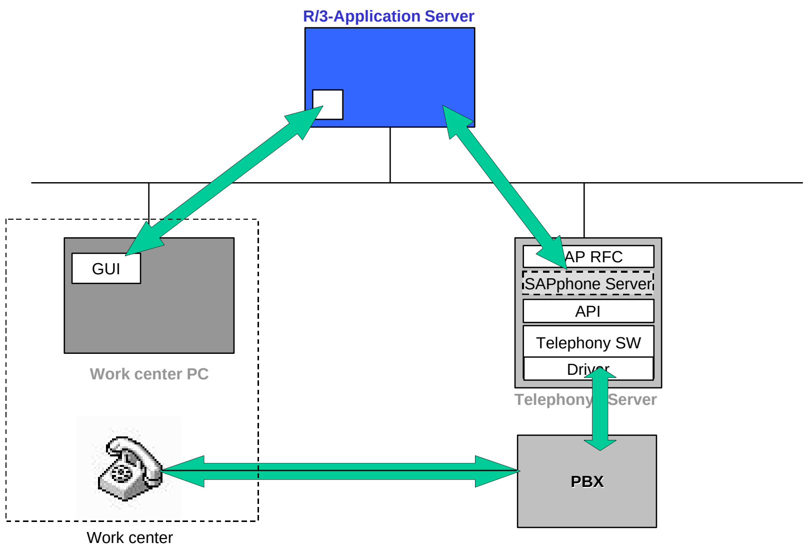


Figure 5: Architectures - centralized connection

3 Interface Technology

This chapter describes how the CTI system communicates with SAPphone. There are two possibilities:

- Via the SAPphone RFC interface
- Via the TAPI standard interface

The SAPphone RFC interface is the unique R/3 access point to the SAPphone functionality in R/3. As a result, the telephony software can always be connected directly to the SAPphone RFC interface.

Instead of communicating directly with the SAPphone RFC interface, an external telephony software system may communicate – via Microsoft TAPI – with the SAPphone server. The SAPphone server is shipped by SAP. It serves as a gateway between Microsoft TAPI (as a TAPI client) and the SAPphone RFC interface.

It is also possible to use both interfaces in parallel, for example, when different PBXs and service providers are used in a single installation. In this case, for example, one telephony server can communicate with SAPphone via TAPI, while another accesses the RFC interface directly.

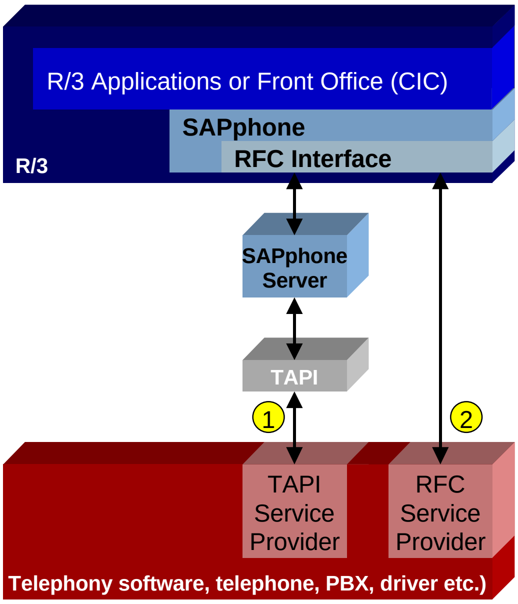


Figure 6: Interfaces

The following sections explain the technology involved, describe the two connection options, and list the components required for each.

3.1 Connection via SAPphone RFC Interface

To integrate RFC software in your own programs, use the RFC Software Development Kit for all operating system platforms supported by the R/3 System. The RFC SDK is a component of the SAPGUI Kit. For the associated descriptions, please refer to the following R/3 online documentation (paths as in R/3 rel. 4.5B), which is available on the Presentation CD:

1. R/3 online documentation on the RFC: R/3 library -> BC - Basis -> Basis Services / Communication Interfaces -> Remote Communications -> The RFC API *or* as a WinHelp file on the Desktop SDK after installing the CD
2. R/3 online documentation on the SAP Gateway: R/3 library -> BC - Basis -> Basis Services / Communitcation Interfaces -> BC – SAP Communication: Configuration -> SAP Gateway *or* the ReadMe file on the CD.

To connect the R/3 System and the PBX using the SAPphone RFC interface, the PBX-specific telephony software must be adapted to the SAPphone RFC interface. Depending on the architecture involved the required connection program, the telephony gateway, is installed either on the work center PC or on the central telephony server. This telephony gateway is not a SAPphone component. It merely converts the telephony software functions and commands into the SAPphone RFC library and vice versa.

RFC technology enables the starting and execution of functions – including R/3 ABAP functions and non-R/3 programs - on remote computers. Non-R/3 programs make use of this technology by calling functions from a SAP provided RFC library (C, C++, Java,..). DCOM technology is also supported.
RFC supports the following connections:

- Communication between R/3 systems
- Calling an external function from within an R/3 System
- Calling an R/3 function from within an external system

The connection between the R/3 application server and the external computer is based on TCP/IP. A general description of RFC functionality can be found in the documentation “Remote Communications”, Chapter: “The RFC API”.

3.1.1 Functions called from within R/3: activation type for RFC Calls

To develop programs/functions that are called via RFC from within R/3, the various RFC techniques for starting the program must be taken into account:

- Starting the program in registration mode **-** i.e. the program is started once at the beginning, registers with a SAP gateway under a program ID, and then waits for an RFC call. The call must have the same program ID. This ID is stored in the R/3 System in connection with an RFC destination, which is accessed during the RFC call to determine the external computer and the access path.
- Starting the program in start mode **-** i.e. the program is restarted for each RFC call and then terminated.

Programs that are installed on the central telephony server and called from there should always run in registration mode.
Programs that are installed on the work center PC should be called in start mode. The RFC destination specifies the access path to the RFC program on the work center PC. Only one needs to be defined for all the work centers together. However, if you want to use registration mode here as well (for performance reasons, for example), the program ID must be unique for each work center. This means that a separate RFC destination must be defined in the R/3 System for each work center, and the program ID on the work center PC must be configurable.

3.2 Connection via TAPI Interface

TAPI (Telephony API) is the interface that Microsoft has defined for telephony integration solutions. Many PBXs, CTI middleware products and Call Center systems either support TAPI or can be adapted to TAPI using products from third-party vendors. Earlier versions of TAPI (<= 2.0) are not network-capable, which means a TAPI service provider must be installed on every single desktop PC even in a client/server architecture (the TAPI service provider would then communicate via LAN with a central telephony server). With TAPI version 2.1, TAPI itself will perform communication via the local network.

In addition to the functionality implemented within R/3, SAPphone also includes an external software component, the SAPphone server. With this program, which can be installed either on the central telephony server or on the desktop PC, the SAPphone RFC interface is adapted to the TAPI standard interface, which means that any telephony software that supports TAPI can communicate with R/3.

The SAPphone Server supports TAPI Version 2.0 and later. A detailed description of the SAPphone server is contained in the document ´SAPphone Server 2.0 Usage of TAPI 2.1´.

When connected via the TAPI standard interface, the SAPphone server adapts the SAPphone RFC interface to TAPI. The following components must be installed outside the R/3 System:

- SAPphone server
- TAPI
- TAPI service provider

The distribution of the components for the various architectures is described below:

Local architecture
SAPphone server, TAPI, and the service provider are installed on the desktop PC.

Client/server architecture
Installed on desktop:

- SAPphone server
- TAPI
- If TAPI Version 2.0 is used: local service provider software. This local component will not be necessary in future TAPI versions.

Installed on telephony server:

- Service provider
- If future, network-capable TAPI versions are used: Central TAPI.

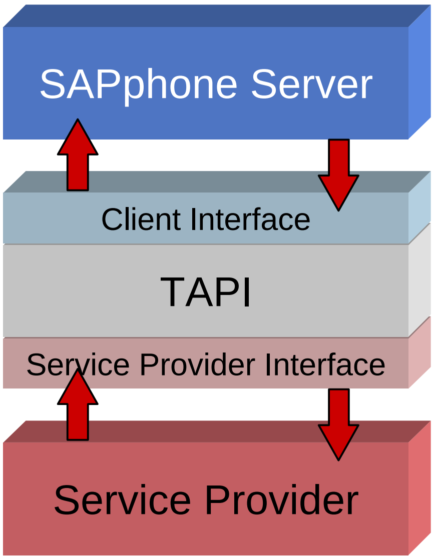


Figure 7: TAPI - Structure

Central architecture
SAPphone server, TAPI, and the service provider are installed on the telephony server.

4 Telephony functions

The following call model describes SAPphone's expectation regarding a telephony gateway's behaviour in terms of call control. This chapter explains e.g. the continuity of call handles required by SAPphone in scenarios like transfer calls or conference calls, the different possibilities to drop calls and the expected results e.g. when performing a drop party on a conference call. Whatever the call model inside the CTI system may be, the gateway should translate it to the model described here when communicating with SAPphone.

4.1 Registration

Some telephony gateways need information from R/3 about which extensions they have to support. They may use this information to register the extensions at the CTI system. R/3 does not require registration.

Telephony gateways that exclusively support users working with the R/3 Customer Interaction Center may use the following mechanism to get this information:
The gateway function SPS_REGISTER is called before any other telephony function. This function passes the extension and the IP address of the workstation the user is working on to the gateway. When the user closes the application and needs no more telephony support, the gateway function SPS_DEREGISTER is called.
If a gateway does not need this information from R/3, it should simply return a returncode = 0000 when these functions are called.

Telephony gateways that offer general telephony support for inbound and outbound calls for all R/3 applications cannot rely on these functions being called by the application. They have to support them **-** at least by returning a "successful" returncode **-** but the functions are not necessarily called before other telephony functions such as SPS_MAKE_CALL. These gateways should periodically call the R/3 function SPS_GET_LINES_PER_SERVER to receive a list of extensions that require telephony support. This list contains all extensions assigned to the gateway and, as of interface version 5.01ASP, information on whether or not a user is currently logged on and, if logged on, the IP address of the workstation. They can then register each extension internally based on this list.

4.2 Inbound calls

4.2.1 Incoming Calls: displaying call information

SAPphone supports three methods by which incoming call information is reported to R/3:

1. The R/3 application "waits" for an incoming call ("Inwait mode").

   In this mode, the application will call the telephony gateway function SPS_WAITFORCALL on behalf of a particular extension. The called function should return control to the calling application only when a call for this extension comes in. Multiple calls may be active at any given moment for any extension.

2. The R/3 application actively requests information on incoming calls.

   In this mode, the R/3 application calls the telephony gateway function SPS_GETCALLSTATE, initiated, for instance, by an agent choosing a pushbutton like "Get incoming call".

3. The telephony gateway actively reports incoming call information to R/3

   In this mode, when a call comes in, the telephony gateway calls the SAPphone function SPS_NEW_CALL.

All three modes should be supported by the telephony gateway.

4.3 Consult

To initiate a consult, the active call must be put on hold, a connection to the new party established, then the held call returned to.
To transfer a call, a similar sequence of steps has to be performed: the active call must be put on hold, a connection to the new party must be established, then the held call must be transferred to this party.
To initiate a conference, the active call must be put on hold, a connection to the new party must be established, then the held and the active call must be merged into one conference call.

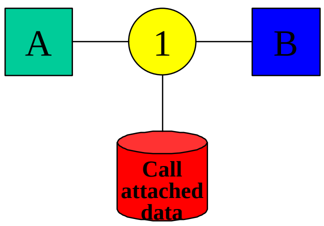
The function SPS_CONSULT combines the first two steps of all these scenarios. It performs two tasks: placing the active call on hold and establishing a connection to the new party.

After successfully calling SPS_CONSULT, the application has a held and an active call. From here, the application has several options:

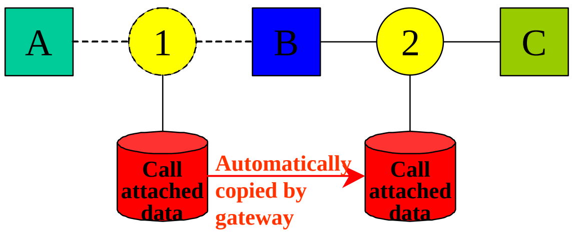
1. Drop the active call and still have the other call on hold to perform other steps (SPS_DROPCALL)
2. Alternate between the two calls by placing the active call on hold and returning to the held call (SPS_ALTERNATE)
3. Drop the active call and return to the held call as in a normal consult call (SPS_RECONNECT)
4. Transfer the held party to the new party as in a warm transfer (SPS_TRANSFER)
5. Merge the two calls into a conference call (SPS_CONFERENCE)

The gateway must support all of these steps after SPS_CONSULT has been called. The SPS_CONSULT parameter NEXT_STEP only indicates the next action which is **likely** to be called, conference or transfer, depending on which function the user initiated. The gateway should also support functions other than the NEXT_STEP function.
In situations where it is not possible to tell in advance what action will follow, the NEXT_STEP parameter will be set to <unknown>. It will also be set to <unknown> if the user initiated just a consult call. The SPS_CONSULT may still be followed by calling SPS_CONFERENCE or SPS_TRANSFER.

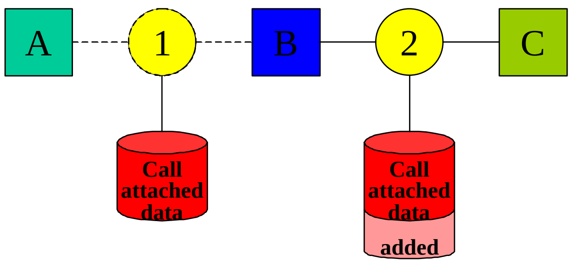
The gateway is responsible for copying call-attached data from the original call to the consult call. Please note: the ANI of the consult call must be set to the party that initiated the consult call, not to the original caller. If data is attached to the consult call, it should be copied back to the original call by the gateway (when completing conference, transfer or reconnect). SAPphone does not perform this function.



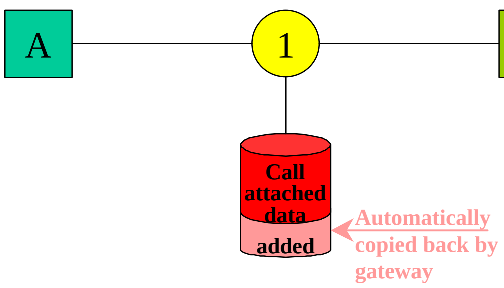
There is an active call between two parties with data attached to the call.



B initiates a consult call to C. The first call is placed on hold. Call-attached data is copied automatically from the first to the second call. The gateway is responsible for this.



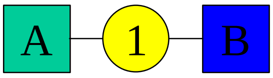
During the consult call, new data is attached to the call and added to the existing call-attached data.



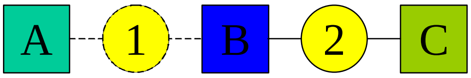
When the call is finally transferred, the consult call is ended and the first call is reactivated. The added call-attached data should then be copied back to the first call. Otherwise it is lost after the transfer.

4.4 Conference

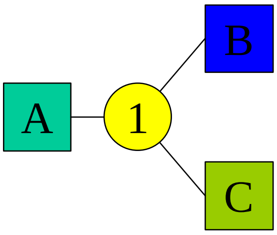
A conference always involves the following steps: put the active call (which may be a conference call) on hold, contact the new party and then initiate the conference.



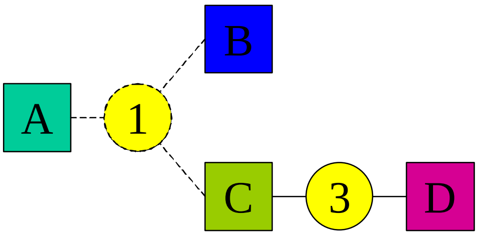
There is an active call between two parties. The call has information about ANI and DNIS and may have call-attached data.



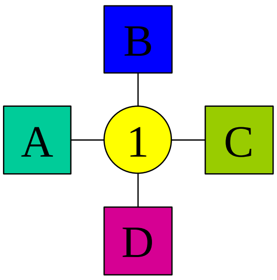
Party B first initiates a consult to C using SPS_CONSULT. This consult is a new call. Data attached to the original call is attached automatically to the new call by the gateway. To make sure that the new party can see the original ANI and DNIS, this information will be attached to the original call by SAPphone before initiating the consult.



When C agrees to participate in the conference, B initiates the conference using SPS_CONFERENCE. The original call then has three parties connected to it. All share the same call-attached data. For dropping parties from the conference, see chapter 4.6 Drop Calls.



When C wants to add another party to the conference, it first consults the new party D. This automatically places the conference call on hold.

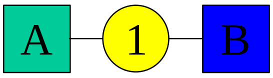


When D agrees to participate in the conference, C uses SPS_CONFERENCE to reactivate the conference and include the new party.

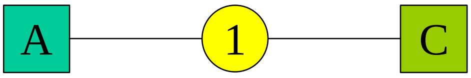
4.5 *Transfer*

There are two different ways to transfer a call: a one-step transfer (blind transfer), where the call is transferred directly to the new destination without prior checking if the new party really wants to accept the call, and a two-step transfer (warm transfer), where the transferring party first contacts the new party and then either transfers the call or returns to the held party.

4.5.1 One-step transfer

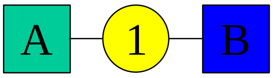


There is an active call between two parties. The call has information about ANI and DNIS and may have call-attached data.

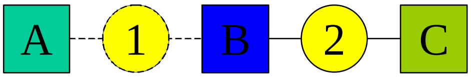


When party B initiates a one-step transfer using SPS_BTRANSFER, the call is transferred directly to the new party. Data attached to the call is available to the new party, because the original call is transferred. To make sure that the new party can see the original ANI and DNIS, this information will be attached to the call by SAPphone prior to transferring the call.

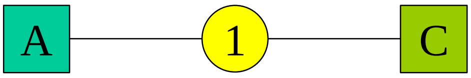
4.5.2 Two-step transfer



There is an active call between two parties. The call has information about ANI and DNIS and may have call-attached data.



To transfer the call, party B first initiates a consult to C using SPS_CONSULT. This consult is a new call. Data attached to the original call is attached automatically to the new call by the gateway. To make sure that the new party can see the original ANI and DNIS, this information will be attached to the original call by SAPphone before initiating the consult.

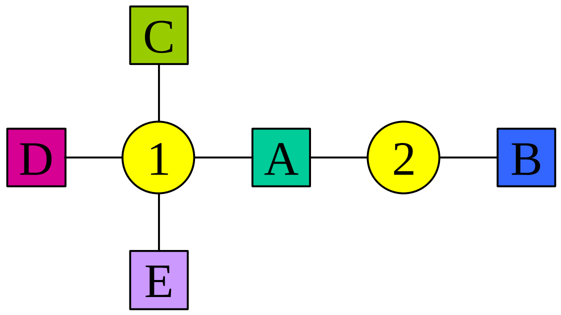


When the new party agrees to take over the call, B transfers the call using SPS_TRANSFER. C now sees the original call.

4.6 Drop Calls

There are three functions for dropping calls:
SPS_DROPPARTY to drop one party from a conference call
SPS_DROPCALL to drop a call completely
SPS_DROPALL to drop all calls for one extension

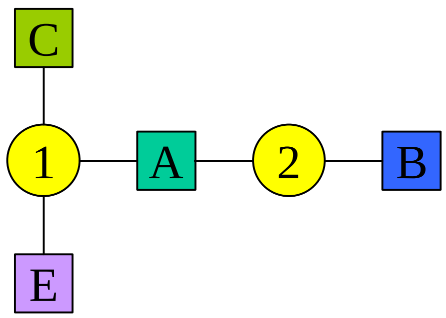
SPS_DROPPARTY is called, if the user wants to drop one party from a conference and keep the other parties in the conference. The party dropped may be the user himself or another party, if the CTI system allows the R/3 application to drop other parties (e.g. with call center supervisor rights).



Call overview for extension A

Handle	Party	Status
1	C	Connected
1	D	Connected
1	E	Connected
2	B	On Hold

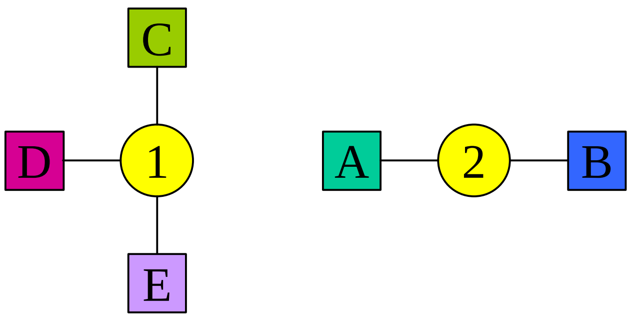
This is an example of a situation, **before** SPS_DROPPARTY is called. There are two calls active on extension A: one is a conference call with C, D and E, the other connection with party B is on hold.



Call overview for extension A

Handle	Party	Status
1	C	Connected
1	E	Connected
2	B	On Hold

After A has called SPS_DROPPARTY to drop D from call 1, A, C and E remain in the conference and B is still on hold.



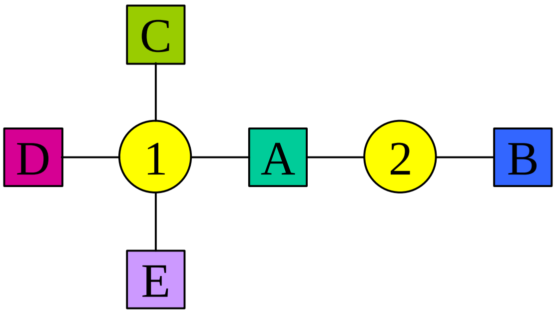
Call overview for extension A

Handle	Party	Status
2	B	On Hold

This is the situation **after** A called SPS_DROPPARTY to drop himself from call 1. The parties C, D and E are still connected to each other in a conference call. A is no longer part of the conference, but has still B on hold in call 2.

Note: If the CTI system does not allow the R/3 application to drop other parties, this function can only be used to drop oneself out of a conference. The conference itself should remain as long as there are at least two parties still connected. If the CTI system drops the whole conference when one party is dropped, then this function should be refused as “not supported”

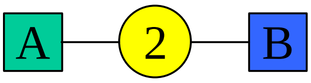
SPS_DROPCALL is used to drop a call completely. This function can be used for normal calls (between two parties) and for conference calls.



Call overview for extension A

Handle	Party	Status
1	C	Connected
1	D	Connected
1	E	Connected
2	B	On Hold

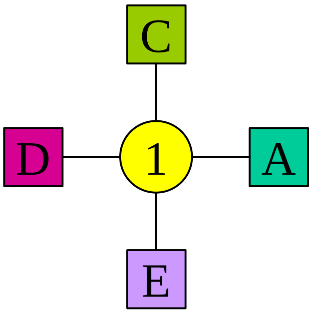
This is an example of a situation **before** SPS_DROPCALL is called. There are two calls active on extension A: one is a conference call with C, D and E, the other connection with party B is on hold.



Call overview for extension A

Handle	Party	Status
2	B	On Hold

After A has called SPS_DROPCALL for call 1, the complete conference call is dropped. None of the parties on the conference (A, C, D, E) are connected any longer, not even to each other. Call 2 is still active.



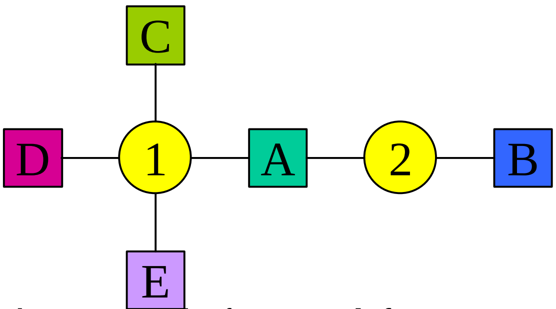
Call overview for extension A

Handle	Party	Status
1	C	Connected
1	D	Connected
1	E	Connected

This is the situation **after** A has called SPS_DROPCALL for call 2. The conference call is still active with all parties connected.

Note: Some CTI systems may not allow one party to drop a conference call completely. In this case SPS_DROPCALL on a conference call would have the same result as SPS_DROPPARTY with the user's party: The own party is no longer connected to the conference, but the conference call remains as long as there are at least two other parties connected.

SPS_DROPALL drops all calls at the user's extension. Internally, the gateway or the CTI system must drop every connection, although the list of connections is not specified in the function call. The gateway or the CTI system has to know which calls to drop.



Call overview for extension A

Handle	Party	Status
1	C	Connected
1	D	Connected
1	E	Connected
2	B	On Hold

This is an example of a situation **before** SPS_DROPALL is called. There are two calls active on extension A: one is a conference call with C, D and E, the other connection with party B is on hold.

Call overview for extension A

Handle	Party	Status

This is the situation **after** SPS_DROPALL is performed. All calls at extension A are dropped.
Note: As the conference call 1 is dropped using the same internal functionality as in SPS_DROPCALL, the same restrictions apply: if the switch or the CTI system doesn't allow a conference call to be dropped completely, the conference remains with the other parties still connected.

5 Predictive Dialing / Power Dialing

5.1 Definition of Terms

- **Predictive dialing / power dialing**: Two automated outbound dialing methods. Lists of → *planned calls* are downloaded to a → *predictive dialer / power dialer*, which then initiates the calls. From an R/3 point of view both methods behave in the same way. In the following both methods are abbreviated to „PD“.
- **Predictive dialer / power dialer:** CTI component for processing automated outbound dialing lists. A predictive dialer or power dialer automatically initiates outbound calls based on a list of planned calls. If a connection could be established, the dialer transfers the call to an available agent and reports the call to R/3. Abbreviated to “dialer” in the following.
- **Planned call:** An R/3 object which represents a planned outbound call.
- **PD list:** A list of planned calls. This list is created in R/3 and used to download planned calls to the dialer.
- **PD list entry:** An entry in a PD list. The entry consists of information on the planned call (e.g. telephone number(s), calling time, date, time zone, etc.).
- **PD call:** Outbound call that has been initiated by a dialer.
- **Campaign:** A container in the dialer into which PD lists can be downloaded. A campaign has parameters such as campaign ID (name), start and end date, attached agent groups, attached agents, etc. Several different campaigns can exist at the same time.

5.2 Architecture

An extension configured in R/3 can only access one telephony gateway at a time. This gateway must provide access to the CTI system supporting telephony functions as well as to the dialer.

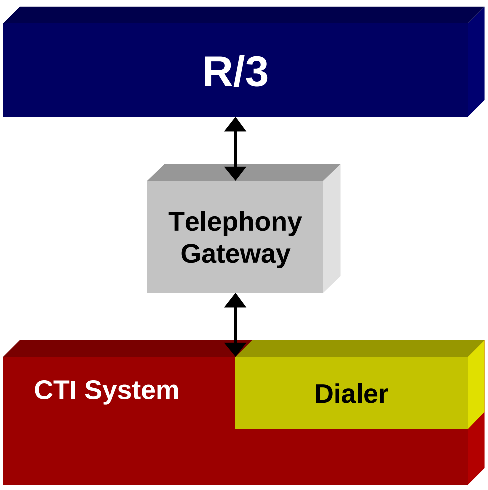


Figure 8: Predictive Dialing Architecture

5.3 Elements of Predictive Dialing / Power Dialing

To process a list of calls in a PD scenario, the following steps must be performed:

1. In the dialer: Create a campaign and define campaign parameters such as campaign ID (name), start time, end time, attached agent groups, attached agents, etc.
2. In R/3: Download one or several PD lists from an R/3 application into a campaign of the dialer via the gateway function SPS_PDLISTTRANSFER (10.3.1).
3. Start of campaign (i.e. of outbound dialing by the dialer) as soon as the start date and time have been reached.
4. Dialer initiates outbound calls. When a PD call can be established, the dialer transfers the call to a free agent and reports the call to R/3 in the same way as an inbound call is reported (see chapter 4.2).
5. Updating of planned calls in the dialer: there will be feedback to the dialer (e.g. that a planned call in the dialer can be marked as finished with the status „successful“). Updating and modifying is performed via the gateway functions SPS_PDLISTTRANSFER (10.3.1) and SPS_MODIFY_PDCALL (10.3.3) (please also refer to chapter 5.6 Modifying Planned Calls in the Dialer).

Steps 4 and 5 are performed until the whole campaign has been completed or the end date of the campaign has been reached.

While the campaign is running:
- further PD lists can be downloaded into this campaign and
- planned calls that have been downloaded already can be modified.

5.4 Relation between Planned Calls in R/3 and in the Dialer

In this paragraph the relationship between planned calls in R/3 and planned calls in the dialer is depicted on the basis of a virtual model. This model shows how the functions work together and what the function parameters mean. Please note that this virtual model does not propose a specific dialer design.

Virtual model: Each planned call in R/3, which has been downloaded to the dialer, corresponds to exactly one planned call in the dialer. A planned call (R/3) has a unique R/3 key and also each planned call in the dialer has a unique key, which will be called „dialer key" in this interface description. SAPphone maintains a mapping table for both unique keys.

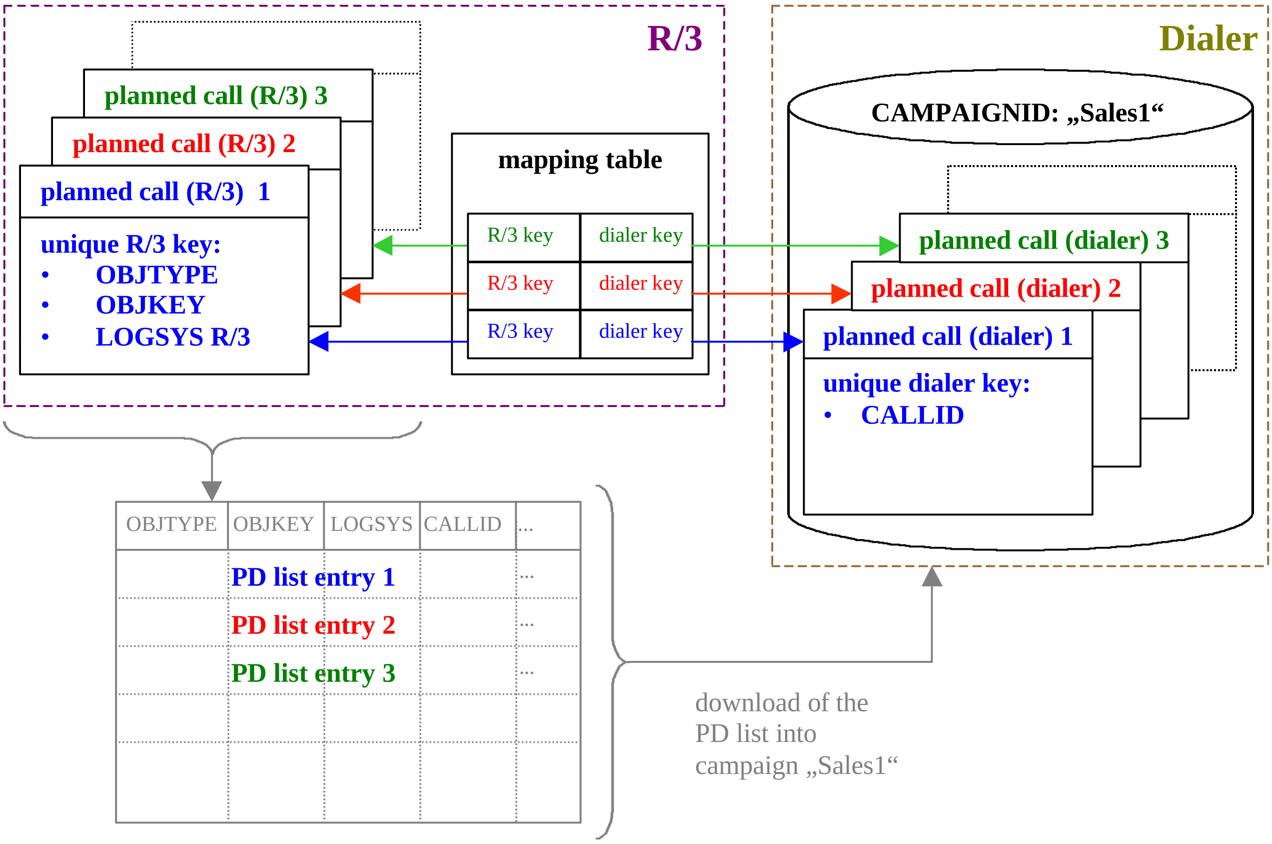


Figure 9: Relation between planned calls in R/3 and in dialer and download of planned calls (R/3) via a PD list

5.5 List Transfer

Planned calls which exist in R/3 are downloaded to the dialer in a PD list via the gateway function SPS_PDLISTTRANSFER (10.3.1). Each PD list entry consists of information on the planned call (telephone number(s), calling time, date, time zone, etc.), the unique R/3 key and a field for the unique dialer key (field „CALLID", see structure SPH_PDCALL, 9.1.8). If a corresponding planned call exists in the dialer, i.e. if the planned call has been downloaded before, SAPphone will fill the dialer key field by using the mapping table.

On list transfer the dialer is expected to check the field for the unique dialer key of each PD list entry. The following two cases have to be distinguished:

- For PD list entries with an empty dialer key field, the dialer has to create a new planned call (dialer) with a new unique dialer key.

- If the dialer key field is filled, the dialer has to update the corresponding planned call (dialer) according to the attributes of the PD list entry.

As a consequence of this procedure, each planned call (R/3) corresponds exactly to one planned call (dialer) and each planned call (dialer) has a unique dialer key.

To enable SAPphone to maintain the mapping table, at some point the dialer has to pass the dialer keys back to R/3. Again two cases have to be distinguished:

- Either: The dialer keys are passed back to R/3 immediately on return of the function SPS_PDLISTTRANSFER (please note, that the PD list is transferred via a TABLES parameter, which can be used in both directions: import and export).

- Or: The dialer keys are passed back to R/3 later, which may be preferable e.g. for performance reasons. Then the dialer has to call the SAPphone RFC function SPS_PD_STATUS (10.3.2) to pass the dialer keys back to R/3.

5.6 Modifying Planned Calls in the Dialer

Planned calls in the dialer can be modified either by downloading a PD list via the gateway funtion SPS_PDLISTTRANSFER (10.3.1) (several planned calls at the same time) or via the gateway function SPS_MODIFY_PDCALL (10.3.3) (a single planned call). In both cases SAPphone determines the unique dialer key of a planned call (dialer) via the mapping table and transfers this key to the dialer. The dialer is expected to modify the corresponding planned calls (dialer).

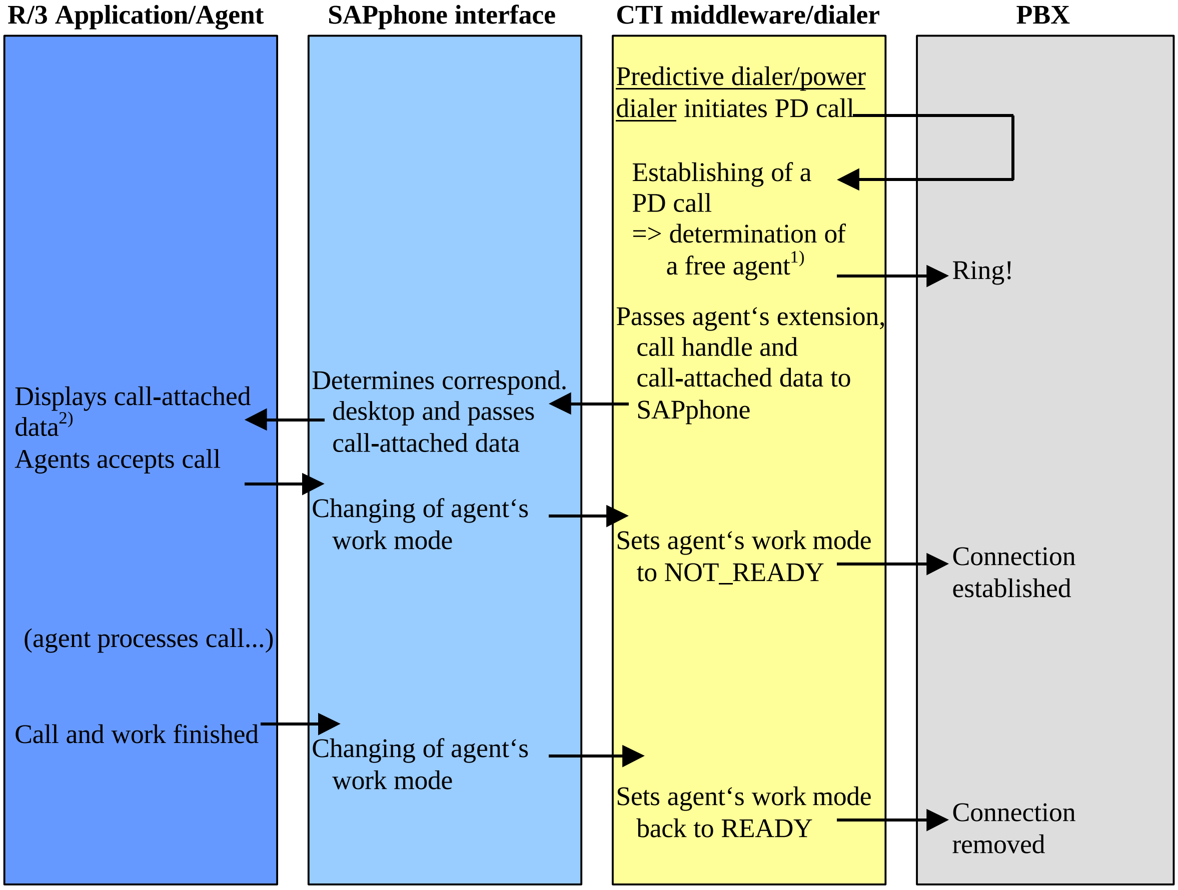
5.7 Reporting PD calls to R/3

PD calls are reported to R/3 by the gateway as inbound calls (see chapter 4.2). In addition, the following attributes of a planned call (dialer) are transferred as call-attached data (see chapter 6):

OBJNAME	INST.	KEYNAME	VALUE	
PDCALL	01	PD_CALL		Must be the 1st entry; no value has to be specified
PDCALL	01	OBJTYPE		
PDCALL	01	OBJKEY		
PDCALL	01	LOGSYS		
PDCALL	01	CALLID		
PDCALL	01	ADDRNUMBER		
PDCALL	01	PERSNUMBER		
PDCALL	01	ADDR_TYPE		all fields of structrue SPH_PDCALL (9.1.8)
PDCALL	01	TELNO1		
PDCALL	01	STARTTIME1		
PDCALL	01	ENDTIME1		
PDCALL	01	TIMEZONE1		
PDCALL	01	:		
PDCALL	01	FINISHED		
PDCALL	01	PDCSTATE		
PDCALL	01	CAMPAIGNID		Parameter CAMPAIGNID from function SPS_PDLISTTRANSFER (10.3.1)

Figure 10: Table „CALLDATA“ of functions SPS_NEW_CALL and SPS_WAITFORCALL

The following diagram shows an example for reporting (via function SPS_NEW_CALL) and processing of a PD call. Dependent on the call center solution, there may be differences at several points.



[1]: each agent possesses a defined work mode
[2]: alternatively a task could be started (e.g. by using a corresponding business object)

5.8 Statistics

R/3 can retrieve statistical information either for a whole campaign or for an individual agent by using the gateway function SPS_STATCAMPAIGN (10.3.4).

5.9 Inbound call before PD call

Scenario: A customer calls into the call center, before the planned call (dialer) for this customer is performed by the dialer.

The following cases have to be distinguished:

- The dialer and the agent do not check if a planned call for this customer exists in the dialer. The planned call will be performed by the dialer.
- The dialer does not check, but the agent checks if a planned call for this customer exists in the dialer. The agent can prevent the planned call by modifying its status (setting to „finished“).
- The dialer checks for an existing planned call. The dialer can display this inbound call as if it happened as a planned call. The agent can handle both the inbound call and the planned outbound call. After this the planned call (dialer) is set to „finished“.

6 Call-attached data

6.1 Purpose

When talking about call-attached data we are not talking about call handle or telephone numbers. Call-attached data is application data related to a call. It could e.g. be the customer ID of the calling party or the number of an order on which the customer wants information

The data can be collected in the following ways:

- in an IVR session, where the caller enters some key information, e.g. his customer number, before the call is transferred to an agent
- during the call, e.g. a new order is created by a call center agent or a service notification is manually selected by the called user and attached to the call
- generated within the call center software outside of R/3

Call-attached data can be sent in two directions:

- Into R/3, when an incoming call arrives
- Out of R/3, when a call is initiated or transferred to another party

Within R/3 call-attached data is used to fill in applications on incoming or transferred calls in advance:

- entering the customer number in an IVR session can help to identify the caller if caller identification based on the phone number is unlikely (e.g. because ISDN is not available)
- attaching a service notification number already selected manually to a call that is transferred to another agent allows the service notification to be displayed without new selection.

6.2 System Architecture

Call-attached data is not stored within R/3, but in the external telephony software that is connected to SAPphone. It is provided to R/3 either by request (R/3 calls functions within the external system to read or modify the data) or together with an incoming call in the form of a tables parameter.

When refering to a call center with an IVR and other components involved, call-attached data can be stored in several places:

- In the IVR
- In the Call Center Software
- In a Telephony gateway

Currently an extension configured in SAPphone can only access one component at a time, so that the component providing the call control functionality must be the same that provides call-attached data, but it doesn't necessarily have to be the same component that stores the data.

To make the solutions described above more visible, the following diagrams show some possibilities for the architecture of a call center, with focus on where call-attached data is stored and who provides this data to SAPphone. The box ‚Call Center' in the diagrams could also be a telephony gateway without special call center functionality. It simply stands for the main component providing the call control functionality to SAPphone.

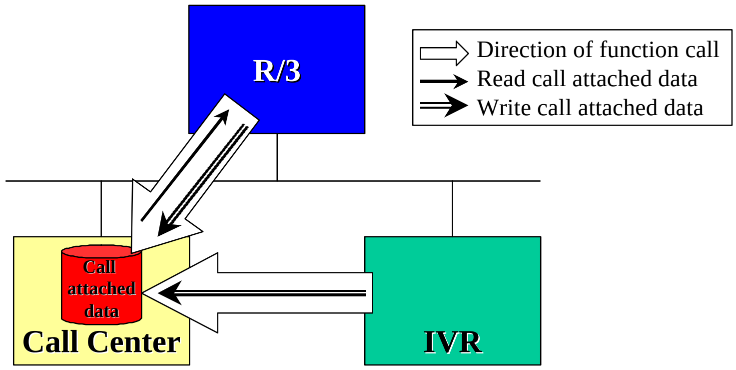
Call-attached data can be provided to R/3 in two ways:

- Via a set of special function modules to read and modify call-attached data
- Via a table parameter in the function modules to send inbound calls to R/3, to make consult calls (thus allowing to forward call-attached data with call transfers, conference calls and consult calls) and to initiate outbound calls.

This paragraph about call-attached data mainly covers the first way, the set of specialized functions. Therefore the following diagrams do not include the flow of call data or call control. They only show the data flow and function call direction concerning call-attached data

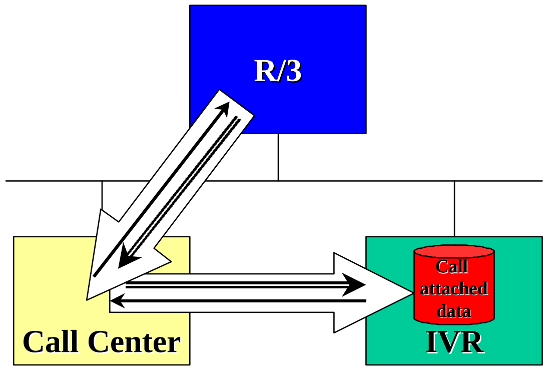
6.2.1 Data in call center, single access

R/3 in connection with call center software and an IVR. Communication takes place only between R/3 and the call center. Call-attached data is stored in the call center.



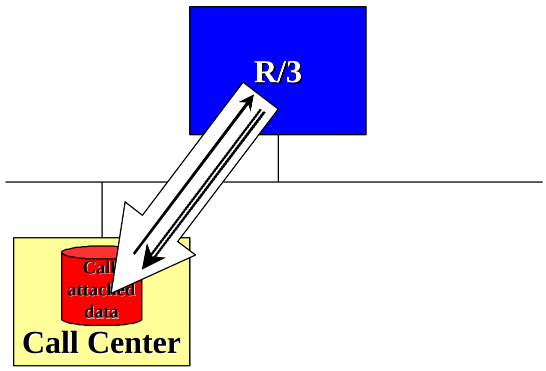
6.2.2 Data in IVR, single access

R/3 in connection with call center software and an IVR. Communication takes place only between R/3 and the call center. Call-attached data is stored in the IVR.



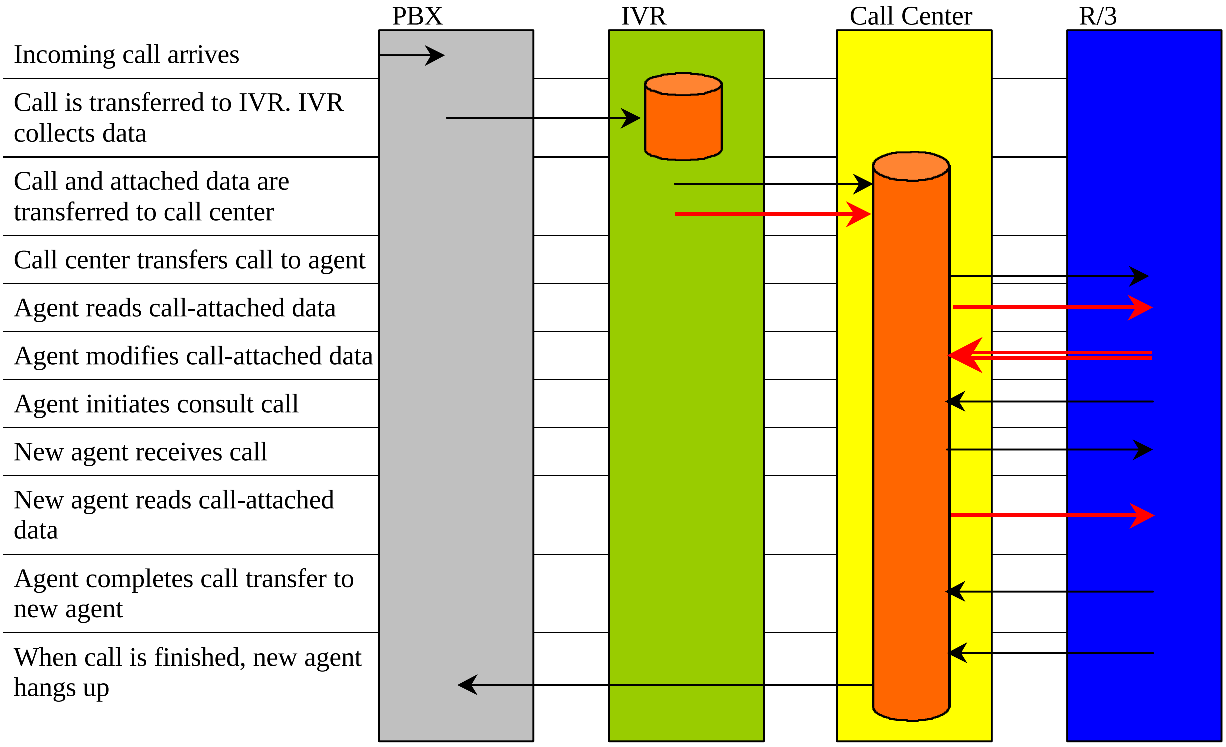
6.2.3 Data in Call Center, no IVR

R/3 in connection with call center software. No IVR is involved. Communication takes place between R/3 and the call center. Call-attached data is stored in call center.
Call-attached data collected during the call within R/3 or within the call center software can be transferred to another agent or an outbound call can be accompanied by call-attached data collected prior to the call.



6.2.4 Call and data flow

The diagram below shows how the call and call-attached data can be processed within a system with IVR and call center, when call-attached data is stored within the call center and communication takes place only between R/3 and the call center. Similar diagrams could be drawn for the other architectures.



6.3 Data Structure

Call attached data is transported as a table, but the data is organised in a tree-like structure. The information itself is contained in key-value-pairs. Multiple key-value-pairs, preferably those that are needed to identify one business object, can be put together in an object instance e.g. if there is more then one key field. Multiple object instances belong to one object (for more detail see chapter 9.1.6 SPH_IOCONT). The object is used to categorize information.

The following example should explain the structure of call-attached data:

Call attached data as tree structure...

	Object	***Instance***	***Key***	***Value***
Call				
	CIC_CLIPBOARD			
		0001		
			REFOBJTYPE	ORDER
			OBJKEY	0000000082
		0002		
			REFOBJTYPE	SERVICENOTE
			OBJKEY	00009944632
	KEYVALUE			
		0001		
			CUSTOMER	0856951
		0002		
			REASON	NEWORDER

....and its representation in the table:

Object name	Object instance	Key name	Value
CIC_CLIPBOARD	0001	REFOBJTYPE	ORDER
CIC_CLIPBOARD	0001	OBJKEY	0000000082
CIC_CLIPBOARD	0002	REFOBJTYPE	SERVICENOTE
CIC_CLIPBOARD	0002	OBJKEY	00009944632
KEYVALUE	0001	CUSTOMER	0856951
KEYVALUE	0002	REASON	NEWORDER

As the data is interpreted within R/3, certain object names and key names have to be used. The following table shows all object names and key names that must be set by the telephony gateway when attaching data e.g. from an IVR or a planned call information to a call. Additional object names and key names are used, but this data is set and read only from SAPphone. The gateway should only store and provide the data when requested.

These are the valid object names:

CIC_CLIPBOARD	Contains business objects and structured fields that are displayed in the CIC clipboard
PDCALL	Identifies a planned call and contains all relevant data. The data comes originally from R/3 and is downloaded to the gateway with the PD-functions (see chapter 5.5, 5.6)
KEYVALUE	Contains unstructured, simple key-value-pairs (not supported in R/3 rel. 4.5A)

Each of these objects has a list of valid key names:

Objectname	**Key name**	**Type of value**	**Description**
CIC_CLIPBOARD	REFOBJTYPE OBJKEY	CHAR10 CHAR70	Type and key of a business object. The object type must be known in the R/3 business object repository. The object key must have the correct structure, dependent on the object type.
	REFSTRUCT REFFIELD VALUE	CHAR30 CHAR30 CHAR255	Field of structure and value. Structure name and field name must be known in the R/3 data dictionary, the value must have the correct type and length, dependent of the structure-field.
PDCALL	PD_CALL	CHAR1	Must be available to signal that the call is a planned call
	OBJTYPE OBJKEY LOGSYS	CHAR10 CHAR70 CHAR10	Type, key and logical system for a business object that is connected to the planned call. This could e.g. be a customer contact. Based on this object, the application that handles the call gets all the information it needs (e.g. what campaign, who is called etc)
	CALLID	CHAR30	ID that is assigned by the campaign managing software to the planned call. This is not the handle of the initiated call.
	ADDRNUMBER PERSNUMBER ADDR_TYPE	CHAR10 CHAR10 CHAR1	Address ID of remote party: Company address, personal address and address type.
	TELNO1 STARTTIME1 ENDTIME1 TIMEZONE1	CHAR30 CHAR6 CHAR6 CHAR6	Phone number to dial for the planned call. A phone number here includes a time window defined by start time, end time and time zone. The call can only be made in this time window.

	TELNO2 STARTTIME2 ENDTIME2 TIMEZONE2	CHAR30 CHAR6 CHAR6 CHAR6	1. alternative phone number for the planned call
	TELNO3 STARTTIME3 ENDTIME3 TIMEZONE3	CHAR30 CHAR6 CHAR6 CHAR6	2. alternative phone number for the planned call
	FINISHED	CHAR1	This flag signals whether a planned call is completed or not. It is set by the software managing the campaign (the dialer) after a call has been established successfully, but the application can change this flag e.g. if the call must be repeated because the wrong person was on the line (see chapter 5.6 Modifying Planned Calls in the Dialer). It is part of call-attached data for information purposes.
	PDSTATE	CHAR3	This field is set by the application only and should be stored by the dialer to allow statistics. It is part of call-attached data for information purposes.
	CAMPAIGNID	CHAR30	ID of the campaign
	STARTDATE ENDDATE	CHAR8 CHAR8	Start and end date of the time period in which the planned call should be performed
KEYVALUE	Any character string is allowed		

6.4 Processing call-attached data

Call attached data is stored in the external telephony software. To read or modify the data from R/3, the following function modules are used.

SPS_GETCALLINFO	Reads all call-attached data of a single call and returns a list in the structure described above.
SPS_SETCALLINFO	Sets all call-attached data of a single call. The function replaces the existing call-attached data with the list of data received from R/3.
SPS_APPENDCALLINFO	Adds call-attached data to the list. The function adds the list of data received from R/3 to the list of existing call-attached data.
SPS_DELETECALLINFO	Deletes call-attached data from the list. The function deletes the data specified in the import tables parameter from the list of existing call-attached data.

These functions must be provided by the external telephony software and are called by R/3. Each of the functions receives the call handle as import parameter, because call-attached data is always related to a single call. For the interface of these functions, please see chapter 10.4).

In addition to the above mentioned functions for actively reading call-attached data, the application

- receives the call-attached data on an inbound call (SPS_NEW_CALL 10.2.2, SPS_WAITFORCALL 10.1.11),
- can send call-attached data with an outbound or consult call (SPS_MAKE_CALL 10.1.15, SPS_CONSULT 10.1.16).

6.4.1 Processing call-attached data in 4.5A

The functions described in the chapter above are available from 4.5B onwards. In 4.5A (and only there) we have a different set of functions to exchange call-attached data between R/3 and the external telephony system.

SPS_CREATEINFO	This function receives the object name (the category) and creates an (empty) info-object. It generates and returns the object instance number.
SPS_SETINFO	This function receives the object name, object instance and key-value-pairs and stores the data in the corresponding info object.

For the interface of these functions, please see chapter 10.5 Functions for call-attached data in interface version 4.00A.
If a telephony gateway wants to support R/3 rel. 4.5A and 4.5B, it has to provide all functions. As these functions are called from R/3, the gateway does not have to know which version it is dealing with as long as it provides all functions.

7 Monitoring of external components

The SAPphone interface includes the definition of functions which allow for monitoring telephony gateways and CTI systems (via their gateways) from within R/3. In this chapter, whenever gateway traces and status information are mentioned, this includes traces and status information pertaining to those CTI system components that are connected to R/3 via their telephony gateway. It is up to the gateway to decide on the extent of CTI system information provided to R/3.
The following methods of monitoring are supported:

1. Gateway traces in R/3
   An administrator or a remote support person can
   - Set and reset the trace level of telephony gateways.
   - Display the gateway trace in R/3.
   - Request additional information about gateway specific trace entries.

2. Monitor gateway status in R/3
   The R/3 alert monitor calls the gateway periodically to retrieve its status. In addition the administrator may check the gateway status anytime.

To provide all data needed for these methods of monitoring, the following functions have been defined. For further information regarding interfaces see chapter 10.6.

SMON_TRACE_SET	To set the trace level of a component either to ON or to OFF (see chapter 7.1). The function must be implemented in the gateway and is called by R/3.
SMON_TRACE_UPLOAD	To upload the trace of a component to R/3. The function must be implemented in the gateway and is called by R/3.
SMON_TRACE_EXPLAIN	To obtain further explanation of gateway specific return codes. The function must be implemented in the gateway and is called by R/3.
SMON_COMP_STATE_RETRIEVE	To obtain a list all components belonging to the gateway with their current status and the information, whether or not they provide trace information. The function must be implemented in the gateway and is called by R/3.

7.1 Working with the gateway trace

The gateway must be able to distinguish between two sets of trace levels: ON and OFF:

- ON means that the user gets all the information he needs to be able to solve problems.

- OFF is the actual trace level that was set on the gateway directly.

An R/3 user can switch between these two trace levels using the function **SMON_TRACE_SET**.
When the function **SMON_COMP_STATE_RETRIEVE** is called, the gateway must return the current trace state for each component. The following values are allowed:

0	The gateway component is not able to provide any trace information to R/3.
1	The trace level of the gateway component is the original level that the administrator has set on the gateway directly (OFF)
2	The trace level was set to ON.

The gateway can only report '2', when the trace was switched ON from within R/3 using the function SMON_TRACE_SET. It must report '1', when the last call of SMON_TRACE_SET set the flag to OFF or the trace level was changed manually on the gateway.

7.2 Monitoring the gateway status

A gateway can consist of multiple components (e.g. for inbound calls, outbound calls, campaigns etc). Trace and status information can be retrieved for each component individually. To obtain status information on all components, the function **SMON_COMP_STATE_RETRIEVE** is called. For each component it returns the name, the current status, additional information on the status and the current trace state.

The status of a component must have one of the following values:

0	No problems at all
1	Component is switched off (no error)
2	Warning: component has heavy load
3	Problem occurred. Administrator interaction necessary
4	Unknown status

8 Certification

The SAPphone RFC interface is part of SAP's Complementary Software Program (CSP). A telephony gateway can be certified as compatible with this interface. The certificate can be obtained in one of three categories:

- Basic telephony
- Call center telephony
- Call center telephony and Predictive dialing

The following chapters list the function modules that have to be supported for each category. The support of these functions is tested in the certification test. It is however strongly recommended, that the gateway does not terminate or close the RFC connection to R/3 when any other function is called. It should return an exception with the message ‘Function not supported’ instead.

The list of function modules also indicates whether a function is implemented in R/3 and can be called by the gateway or must be implemented in the gateway and is called by R/3.

Chapter 4.1 describes the registration mechanisms supported by R/3 and the differences between R/3 wide telephony support and telephony support for users exclusively working in the R/3 Customer Interaction Center. A gateway must offer R/3 wide telephony support for certification.

8.1 Basic telephony

The gateway supports all general and telephony specific functions, which are the following:

Function	Described in chapter	Implemented in gateway	Implemented in R/3
XCHGVERSION	**10.1.1**	**X**	
XCHGPARAMS	**10.1.2**	**X**	
SPS_REGISTER	**10.1.3**	**X**	
SPS_DEREGISTER	**10.1.4**	**X**	
SPS_GETCALLSTATE	**10.1.9**	**X**	
SPS_WAITFORCALL	**10.1.11**	**X**	
SPS_CANCELWAIT	**10.1.12**	**X**	
SPS_ANSWER	**10.1.13**	**X**	
SPS_DEFLECT	**10.1.14**	**X**	
SPS_MAKECALL	**10.1.15**	**X**	
SPS_CONSULT	**10.1.16**	**X**	
SPS_BTRANSFER	**10.1.17**	**X**	
SPS_TRANSFER	**10.1.18**	**X**	
SPS_CONFERENCE	**10.1.19**	**X**	
SPS_RECONNECT	**10.1.20**	**X**	
SPS_HOLD	**10.1.21**	**X**	
SPS_UNHOLD	**10.1.22**	**X**	
SPS_ALTERNATE	**10.1.23**	**X**	
SPS_DROPPARTY	**10.1.24**	**X**	
SPS_DROPCALL	**10.1.25**	**X**	
SPS_DROPALL	**10.1.26**	**X**	
SPS_NEW_CALL	**10.2.2**		**X**
SPS_CALL_ENDED	**10.2.3**		**X**
SPS_GETCALLINFO	**10.4.1**	**X**	
SPS_SETCALLINFO	**10.4.2**	**X**	
SPS_APPENDCALLINFO	**10.4.3**	**X**	
SPS_DELETECALLINFO	**10.4.4**	**X**	
SMON_TRACE_SET	**10.6.1**	**X**	
SMON_TRACE_UPLOAD	**10.6.2**	**X**	
SMON_TRACE_EXPLAIN	**10.6.3**	**X**	
SMON_COMP_STATE_RETRIEVE	**10.6.4**	**X**	

8.2 Call Center telephony

The gateway supports all telephony functions listed under 8.1 Basic telephony and additionally the following call center specific functions:

Function	Described in chapter	Implemented in gateway	Implemented in R/3
SPS_AGENTLOGIN	**10.1.5**	**X**	
SPS_AGENTLOGOUT	**10.1.6**	**X**	
SPS_SETWORKMODE	**10.1.7**	**X**	
SPS_GETWORKMODE	**10.1.8**	**X**	
SPS_GETQUEUES	**10.1.10**	**X**	

8.3 Call Center telephony and Predictive dialing

The gateway supports all telephony and call center functions listed under 8.2 Call Center telephony and the predictive dialing interface, which includes the following functions:

Function	Described in chapter	Implemented in gateway	Implemented in R/3
SPS_PDLISTTRANSFER	**10.3.1**	**X**	
SPS_PD_STATUS	**10.3.2**		**X**
SPS_MODIFY_PDCALL	**10.3.3**	**X**	
SPS_STATCAMPAIGN	**10.3.4**	**X**	
SPS_DELETE_PDCALL	**10.3.5**	**X**	
SPS_CAMPAIGNS_GET	**10.3.6**	**X**	
SPS_ASSIGNED_CAMPAIGNS_GET	**10.3.7**	**X**	

8.4 Optional functions

The function SPS_GET_LINES_PER_SERVER is optional in each of the categories. The gateway only has to support it, when the output of this function is needed to provide the functionality (see 4.1 Registration)
The function SPS_GENERIC may be used to support gateway specific features. It is therefore up to the gateway whether or not it supports this function.

Function	Described in chapter	Implemented in gateway	Implemented in R/3
SPS_GET_LINES_PER_SERVER	**10.2.1**		**X**
SPS_GENERIC	**10.1.27**	**X**	

9 SAPphone data definitions: structures, types, constants

9.1 Structures

9.1.1 SPH_CINFO

Key-value-pair for additional information about one call.

Field	Data element	Description
INFOKEY	SP_INFOKEY	Keyname of key value pair
VALUE	SP_INFOVAL	Value of key value pair

9.1.2 SPH_CSTATE

Structure for information about one call. This structure is used to pass information about all calls currently connected to one extension, with one entry describing one call.

Field	Data element	Description
HANDLE	SP_HANDLE	Call handle
STATE	SP_STATE	Status of the call
PARTY	SP_TELNO	Remote party telephone number
PTYPE	SP_PTYPE	Type of remote party
DNIS	SP_TELNO	Local party telephone number

9.1.3 SPHOPTIONS

Structure to control the behaviour of SPS_NEW_CALL The structure is flexible to keep the interface stable even if new control parameters are needed. For use of the option fields, see function interface description.

Field	Data element	Description
OPTION1	SP_OPTION	control parameter
OPTION2	SP_OPTION	control parameter
OPTION3	SP_OPTION	control parameter
OPTION4	SP_OPTION	control parameter
OPTION5	SP_OPTION	control parameter

9.1.4 SPH_LINES

Structure to export a list of extensions out of R/3.

Field	Data element	Description
TELNO	SP_TELNO	Extension assigned to a specific telephony server

9.1.5 SPH_IOBJ

Key-value-pair structure for call-attached data info object

Field	Data element	Description
KEYNAME	SP_IOKNAM	Key name of call-attached data
VALUE	SP_IOKVAL	Values of call-attached data

9.1.6 SPH_IOCONT

Structure for call-attached data

Field	Data element	Description
OBJNAME	SP_IONAM	Object
INSTANCE	SP_IONUM	Instance of object
KEYNAME	SP_IOKNAM	Key name
VALUE	SP_IOKVAL	Key value

9.1.7 SPH_IODESC

Information object descriptor

Field	Data element	Description
KEYNAME	SP_IOKNAM	Key name
VTYPE	SP_IOKVAL	Type of value

9.1.8 SPH_PDCALL (PD call / list entry)

This structure specifies one PD list entry and is used for the list transfer and for the modification of planned calls in the dialer.

Field	Type/Data element	Description
OBJTYPE	SWO_OBJTYP (CHAR 10)	This triple determines the unique R/3 key for a planned call (R/3)
OBJKEY	SWO_TYPEID (CHAR 70)	
LOGSYS	LOGSYS (CHAR 10)	
CALLID	SP_CALLID (CHAR 30)	Unique dialer key for the corresponding planned call in the dialer, please refer to paragraph 5.4 Relation between Planned Calls in R/3 and in the Dialer
ADDRNUMBER	AD_ADDRNUM (CHAR 10)	Data of the R/3 central address management
PERSNUMBER	AD_PERSNUM (CHAR 10)	
ADDR_TYPE	AD_ADRTYPE (CHAR 1)	
TELNO1	SP_TELNO (CHAR 30)	Telephone number
STARTTIME1	SP_STIME (CHAR 6)	Determine time range to call the customer (for TELNO1)
ENDTIME1	SP_ETIME (CHAR 6)	
TIMEZONE1	TZNZONE (CHAR 6)	Necessary for the correct interpretation of STARTTIME1 and ENDTIME1
TELNO2	SP_TELNO (CHAR 30)	1st alternative telephone number, optional
STARTTIME2	SP_STIME (CHAR 6)	Determine time range to call the customer (for TELNO2)
ENDTIME2	SP_ETIME (CHAR 6)	
TIMEZONE2	TZNZONE (CHAR 6)	Necessary for the correct interpretation of STARTTIME2 and ENDTIME2
TELNO3	SP_TELNO (CHAR 30)	2nd alternative telephone number, optional
STARTTIME3	SP_STIME (CHAR 6)	Determine time range to call the customer (for TELNO3)
ENDTIME3	SP_ETIME (CHAR 6)	
TIMEZONE3	TZNZONE (CHAR 6)	Necessary for the correct interpretation of STARTTIME3 and ENDTIME3
STARTDATE	SP_SDATE (CHAR 8)	Determine date range to call the customer
ENDDATE	SP_EDATE (CHAR 8)	
FINISHED	CHAR1 (CHAR 1)	Flag, values: ‘Y’ or ‘N’, default: ‘N’
PDCSTATE	CHAR3 (CHAR 3)	Specifies call status

Annotations:

- STARTTIMEx, ENDTIMEx and TIMEZONEx determine the time range from the customer or business point of view. The dialer has to convert this according to its own time zone. Default value: SPACE; then the global call center values are used for the fields STARTTIMEx and ENDTIMEx.
- The flag FINISHED is used to reset a planned call (dialer) to “unfinished”. Example scenario: Although the telephone number was correct, the wrong person was reached by the dialer.
  U This flag is necessary, because usually the dialer marks a PD list entry as finished as soon as a person has been reached.
- PDCSTATE specifies the call status in more detail. This field is used by R/3 to store its call status.

Incompatible change between SAPphone interface versions 4.00A and 5.00A
In version 4.00A the field “FINISHED” has the two possible values SPACE and ‘X’. This has been changed for the delta mechanism in function SPS_MODIFY_PDCALL in 5.00A.

9.1.9 SPH_PDCHIS (PD call history)

Field	Type/Data element	Description
OBJTYPE	SWO_OBJTYP (CHAR 10)	This triple determines the unique R/3 key for a planned call (R/3)
OBJKEY	SWO_TYPEID (CHAR 70)	
LOGSYS	LOGSYS (CHAR 10)	
NO_TRIES	NUMC3 (NUMC 3)	Total number of tries made for this PD list entry
HW_DISP	SP_HW_DISP (CHAR 15)	Hardware disposition (possible values include: FAX, BUSY, NO_ANSWER, ANSWER_MACHINE, SUCCESS, WRONG_PARTY)

TELNO	SP_TELNO (CHAR 30)	Successful or last attempted telephone number that was dialed
AGENT_NAME	SP_CCUSER (CHAR 15)	Agent who handled the call
CAMPAIGNID	SP_QUEUE (CHAR 30)	Campaign the call is assigned to
CALL_TIME	SP_CTIME (CHAR 6)	Time the call was made
CALL_DATE	SP_CDATE (CHAR 8)	Date the call was made
DIAL_TIME	SP_DTIME (CHAR 6)	Time of the call
FINISHED	SP_BOOLEAN (CHAR 1)	Flag, values: ‘Y’ or ‘N’, default: ‘N’
PDCSTATE	CHAR3 (CHAR 3)	Specifies call status

Incompatible change between SAPphone interface version 4.00A and 5.00A

- In version 4.00A the field “FINISHED” has the two possible values SPACE and ‘X’. This has been changed for the delta mechanism in function SPS_MODIFY_PDCALL in 5.00A.

9.1.10 SMON_COMPO

Information about subcomponent: name, state and trace availability

Field	Data element	Description
NAME	SMON_CO_NA	Name of component
STATE	SMON_CO_ST	Current state of component
TRACE	SMON_TR_ST	Current state of trace
TEXT	SMON_TR_LI	Further explanation of component's state

9.1.11 SMON_TRACE

Structure of one entry in the uploaded trace

Field	Data element	Description
DATE	CHAR8	Date of trace entry (in format YYYYMMDD)
TIME	CHAR6	Time of trace entry (in format HHMMSS)
TYPE	SMON_TR_TY	Type of trace message
ID	SP_SYSRC	ID of trace entry, identifying a situation traced (e.g. same error situations have the same trace ID).
TEXT	SMON_TR_LI	Trace text line: 1000 characters in ASCII character set. Use special character '#' to indicate a new line.

9.1.12 SMON_HELP

Structure of one entry in the uploaded help file

Field	Data element	Description
TEXT	SMON_HL_LI	Help text line in ASCII character set. Use special character '#' to indicate a new line.

9.1.13 SPH_LINEIP

Structure to return the list of lines including users currently logged on and IP-address of workcenter

Field	Data element	Description
TELNO	SP_TELNO	Extension assigned to a specific telephony server
USER	SYUNAME	R/3 user ID of user logged on to the workstation the extension is assigned to.
IP_ADDRESS	SP_TERM	IP-address of workcenter the extension is assigned to.

9.1.14 SPH_QUEUES

Structure for a list of queue names.

Field	Data element	Description
QUEUE	SP_QUEUE	ACD queue/campaign

9.1.15 SPH_CAMPGS

Structure for a list of campaign names.

Field	Data element	Description
CAMPAIGN	SP_QUEUE	ACD queue/campaign

9.2 Data elements/Types

Data element	Type	Description	Values
SMON_CO_NA*	CHAR50	Name of component	
SMON_CO_ST*	NUMC3	Status of component	See chapter 9.3.6
SMON_HL_LI*	CHAR 255	Help text line	ASCII character set, '#' to indicate a new line
SMON_TR_LI*	CHAR1000	Trace text line	ASCII character set, '#' to indicate a new line
SMON_TR_ST*	CHAR1	Status of trace	See chapter 9.3.7
SMON_TR_TY*	CHAR1	Type of trace message	See chapter 9.3.8
SP_BOOLEAN	CHAR1	Flag	X = on, true, space = off, false
SP_CALLID*	CHAR30	unique dialer key for a PD list entry	
SP_CCPASS*	CHAR10	Call center password	
SP_CCUSER*	CHAR15	Call center agent ID	
SP_CDATE*	CHAR8	Date the PD call was made,	in format YYYYMMDD
SP_CTIME*	CHAR6	Time the PD call was made,	in format HHMMSS
SP_DFLTKEY	CHAR10	Unique number (out of a number group) of a deflect number	
SP_DTIME*	CHAR6	Length of the PD call,	in format HHMMSS
SP_EDATE*	CHAR8	Ending date of a PD call,	in format YYYYMMDD
SP_ETIME*	CHAR6	Ending time of a PD call,	in format HHMMSS
SP_HANDLE	CHAR32	Call handle/call-ID/connID	
SP_HW_DISP*	CHAR15	Hardware disposition (fax/busy/answermach.,..)	
SP_INFO1	CHAR50	Error description	
SP_INFOKEY*	CHAR15	Call information keyname	
SP_INFOVAL*	CHAR30	Call information value	
SP_IOKNAM*	CHAR32	Information object keyname	See chapter 6.3
SP_IOKVAL*	CHAR255	Information object value	
SP_IONAM*	CHAR32	Information object name	See chapter 6.3
SP_IONUM*	NUMC4	Information object Instance	
SP_OPTION	CHAR10	Control parameter for multiple purposes	
SP_PTYPE*	CHAR1	Party type	See chapter 9.3.5
SP_QTYPE	CHAR1	Type of queue list	See chapter 9.3.9
SP_QUEUE*	CHAR30	ACD queue/campaign	
SP_REASON*	NUM2	Reason for inbound call	See chapter 9.3.2
SP_RETCODE*	CHAR4	Returncode from RFC server	See chapter 11
SP_SDATE*	CHAR8	Starting date of a PD call,	in format YYYYMMDD
SP_SERV_ID	CHAR6	ID of telephony server representing external CTI system	
SP_STATE*	NUM2	Connection state of call	See chapter 9.3.4
SP_STIME*	CHAR6	Starting time of a PD call,	in format HHMMSS
SP_SYSRC	CHAR10	Returncode from external subsystem	
SP_TELNO	CHAR30	Directory number/extension	
SP_TERM	CHAR36	IP address of SAPGUI	
SP_VAL_55	CHAR55	String for returning name, company etc. of remote party	
SP_VERSION	CHAR8	Version number of SAPphone and external CTI system	See chapter 1.4
SP_WRKMODE*	NUM2	Agent workmode	See chapter 9.3.1
SX_NODE_ID	CHAR6	Name of gateway as configured in R/3	
SYUNAME	CHAR12	Logon user ID of R/3 user	

* Denotes new data element

9.3 Constants

9.3.1 Agent workmodes

SPH_WM_READY	01
SPH_WM_NOT_READY	02
SPH_WM_WORK_READY	03
SPH_WM_WORK_NOT_READY	04

9.3.2 Reasons for return of SPS_WAITFORCALL

SPH_REASON_INBOUND_CALL	01
SPH_REASON_WAIT_CANCELLED	02
SPH_REASON_ALREADY_WAITING	03
SPH_REASON_CALLS_AT_EXT	04
SPH_REASON_USER_DEFINED	05
SPH_REASON_PD_CALL	06
SPH_REASON_ERROR	99

9.3.3 Follow_up actions for consult calls

SPH_CONSULT_AS_CONFERENCE	'C'
SPH_CONSULT_AS_TRANSFER	'T'
SPH_CONSULT_AS_UNKNOWN	'U'

9.3.4 Call connection states

SPH_CSTATE_NULL	00
SPH_CSTATE_INITIATED	01
SPH_CSTATE_ALERTING	02
SPH_CSTATE_CONNECTED	03
SPH_CSTATE_HELD	04
SPH_CSTATE_QUEUED	05
SPH_CSTATE_FAILED	06
SPH_CSTATE_OFFERED	07

9.3.5 Call state party types

SPH_PTYPE_INTERNAL	‘I’
SPH_PTYPE_EXTERNAL	‘E’

9.3.6 Component states

SMON_CO_ST_OK	0
SMON_CO_ST_OFF	1
SMON_CO_ST_WARNING	2
SMON_CO_ST_FAILURE	3
SMON_CO_ST_UNKNOWN	4

9.3.7 Trace states

SMON_TR_ST_NOTAVAIL	0
SMON_TR_ST_OFF	1
SMON_TR_ST_ON	2

9.3.8 Trace message types

SMON_TR_TY_INFO	'I'
SMON_TR_TY_WARNING	'W'
SMON_TR_TY_ERROR	'E'

9.3.9 Queue list types

SPH_QUEUES_LOGGED_IN	‘0’
SPH_QUEUES_AUTHORISED	‘1’

10 SAPphone RFC function definitions

The following chapter lists the interface definitions of all SAPphone RFC functions. The parameter attribute 'Import' or 'Export' is described from the perspective of the implementer of the function (the gateway for RFC server functions, R/3 for RFC client functions). E.g. when the function is implemented in the gateway and called by R/3, an import parameter is filled by R/3 when calling the function. The export parameter is returned to R/3.

The parameter RETURNCODE (an export parameter of all of the functions called by R/3) is used to return an error code generated in the telephony gateway. Each function has a list of valid returncodes for error situations. For a complete list of all error codes and guidelines on how to use them see chap. 11. Although the returncode is generated in the telephony gateway, the error may have occurred in the underlying CTI system. To distinguish between errors originating in the telephony gatway and errors originating in the CTI system, the returncode has the following structure: AXXX.
A must be set to the value '0' if the error originated in the telephony gateway, or to the value '1' if the error originated in the underlying CTI system. XXX must have one of the values defined for each function.

Even in the event of an error, all export parameters (if possible) must be filled. The values may be used for user information.

10.1 Call Control Functions

The following functions are called by the R/3 System and are implemented within the external software.

10.1.1 XCHGVERSION

Purpose: to exchange the version numbers of SAPphone and the telephony gateway.
The telephony gateway should in every case return its own version number. This version number is stored in R/3. If the version is not compatible with SAPphone, a warning is displayed to the user. Every call control function implemented in R/3 requires a version. If the telephony gateway has a lower version number than this required version, the function will not be executed.

	Name	**Type**	**Comment**
IMPORT	FB_VERSION	SP_VERSION	R/3 SAPphone version
EXPORT	WS_VERSION	SP_VERSION	telephony gateway version
	RETURNCODE	SP_RETCODE	Returncode of the telephony gateway
	SYSTEM_RC_X	SP_SYSRC	Returncode of the CTI system
	ERR_TEXT	SP_INFO1	Error description

Valid error codes:
000 No error
001 Function not supported
002 Function could not be executed
005 System error
006 Resources not available
009 Gateway and SAPphone not compatible

10.1.2 XCHGPARAMS

Purpose: to exchange parameters that have to be maintained both in R/3 and in the telephony gateway.
The name, by which the telephony gateway is identified within SAPphone, is exported to the telephony gateway. This name must be returned to SAPphone when the beginning or the end of an arriving call is reported using SPS_NEW_CALL or SPS_CALL_ENDED, so the name should be stored somewhere in the telephony gateway.

	Name	**Type**	**Comment**
IMPORT	SERVER_NAME	SP_SERV_ID	Server name as maintained within R/3
EXPORT	RETURNCODE	SP_RETCODE	Returncode of the telephony gateway
	ERR_TEXT	SP_INFO1	Error description

Valid error codes:
000 No error
001 Function not supported
002 Function could not be executed
005 System error
006 Resources not available

10.1.3 SPS_REGISTER

Purpose: to register an extension at a telephony gateway, thereby making it known to the telephony gateway.
This function must not be made a prerequisite for offering CTI to the user.
Another possibility to get a list of all extensions, that are configured within SAPphone and therefore are likely to require CTI, is to call SPS_GET_LINES_PER_SERVER.

	Name	**Type**	**Comment**
IMPORT	EXT	SP_TELNO	Own extension
	SAPGUI_ADDRESS	SP_TERM	IP-Address of terminal
EXPORT	RETURNCODE	SP_RETCODE	Returncode of the telephony gateway
	SYSTEM_RC_X	SP_SYSRC	Returncode of the CTI system
	ERR_TEXT	SP_INFO1	Error description

Valid error codes:
000 No error
001 Function not supported
002 Function could not be executed
003 Authorization error
004 Connection error
005 System error
006 Resources not available
010 Max. number of registered extensions reached
011 Extension already registered with different IP-address
012 Network error for IP-address
016 Extension is not known

10.1.4 SPS_DEREGISTER

Purpose: to take an extension out of the list of extensions registered and therefore known to the telephony gateway.

	Name	**Type**	**Comment**
IMPORT	EXT	SP_TELNO	Own extension
EXPORT	RETURNCODE	SP_RETCODE	Returncode of the telephony gateway
	SYSTEM_RC_X	SP_SYSRC	Returncode of the CTI system
	ERR_TEXT	SP_INFO1	Error description

Valid error codes:
000 No error
001 Function not supported
002 Function could not be executed
003 Authorization error
004 Connection error
005 System error
006 Resources not available
014 Deregistration currently not possible due to active call
015 Deregistration currently not possible due to user still logged into queue
016 Extension is not known

10.1.5 SPS_AGENTLOGIN

Purpose: to log a call center agent into a queue to receive calls that are in this queue.

	Name	Type	Comment
IMPORT	EXT	SP_TELNO	Own extension
	QUEUE	SP_QUEUE	Queue to log into (blank = switch selected)
	WORKMODE	SP_WRKMODE	Initial agent workmode (0=none specified, see 9.3 Constants - work mode constants and 10.1.7 for more information)
	CCUSER	SP_CCUSER	Call center user ID
	CCPASS	SP_CCPASS	Call center password
EXPORT	RETURNCODE	SP_RETCODE	Returncode of the telephony gateway
	SYSTEM_RC_X	SP_SYSRC	Returncode of the CTI system
	ERR_TEXT	SP_INFO1	Error description

Valid error codes:

000 No error
001 Function not supported
002 Function could not be executed
003 Authorization error
004 Connection error
005 System error
006 Resources not available
013 Extension is not yet registered
016 Extension is not known
017 Queue is unknown
018 Queue is not available
019 Max. number of users logged in reached
020 Agents call center user-ID is not valid
021 Agents call center user-ID is locked
022 Agents call center password is not valid
026 Workmode is not supported
027 Agent can temporarily not be set into workmode

10.1.6 SPS_AGENTLOGOUT

Pupose: to take a call center agent out of a queue, so that calls in this queue no longer get routed to the agent.

	Name	Type	Comment
IMPORT	EXT	SP_TELNO	Own extension
	QUEUE	SP_QUEUE	Queue to log out of (blank = all)
	CCUSER	SP_CCUSER	Call center user ID
EXPORT	RETURNCODE	SP_RETCODE	Returncode of the telephony gateway
	SYSTEM_RC_X	SP_SYSRC	Returncode of the CTI system
	ERR_TEXT	SP_INFO1	Error description

Valid error codes:

000 No error
001 Function not supported
002 Function could not be executed
003 Authorization error
004 Connection error
005 System error
006 Resources not available
013 Extension is not yet registered
016 Extension is not known
017 Queue is unknown
018 Queue is not available
020 Agents call center user-ID is not valid
021 Agents call center user-ID is locked
024 Logging out currently not possible due to active call

10.1.7 SPS_SETWORKMODE

Purpose: to set the current status of an agent.
This as well as the status of the extension (busy, ready) controls whether calls in a queue are transferred to the agent or not. Only if the extension and the agent are both in status ready, the agent can receive arriving calls. There are 4 defined work modes for ready, not ready, work ready and work not ready (the last two can be called during a phone call and set the work mode after the call is finished). Other work modes are possible, e.g. to provide switch specific functions in R/3 (e.g. setting the phone busy / not busy)). The values of these work modes must be communicated between the gateway and the application (Customer Interaction Center – CIC). SAPphone only passes the values.

	Name	**Type**	**Comment**
IMPORT	EXT	SP_TELNO	Own extension
	WORKMODE	SP_WRKMODE	Agent work mode (for values: see 9.3 Constants - work mode constants)
EXPORT	RETURNCODE	SP_RETCODE	Returncode of the telephony gateway
	SYSTEM_RC_X	SP_SYSRC	Returncode of the CTI system
	ERR_TEXT	SP_INFO1	Error description

Valid error codes:
000 No error
001 Function not supported
002 Function could not be executed
003 Authorization error
004 Connection error
005 System error
006 Resources not available
013 Extension is not yet registered
016 Extension is not known
023 Agent is not logged into a queue
026 Workmode is not supported
027 Agent can temporarily not be set into workmode

10.1.8 SPS_GETWORKMODE

Purpose: to get the current workmode for an extension.

	Name	**Type**	**Comment**
IMPORT	EXT	SP_TELNO	Own extension
EXPORT	WORKMODE	SP_WRKMODE	Agent work mode (for values: see 9.3 Constants - work mode constants)
	RETURNCODE	SP_RETCODE	Returncode of the telephony gateway
	SYSTEM_RC_X	SP_SYSRC	Returncode of the CTI system
	ERR_TEXT	SP_INFO1	Error description

Valid error codes:
000 No error
001 Function not supported
002 Function could not be executed
003 Authorization error
004 Connection error
005 System error
006 Resources not available
013 Extension is not yet registered
016 Extension is not known
023 Agent is not logged into a queue

10.1.9 SPS_GETCALLSTATE

Purpose: to return a list of all calls currently active for the extension (e.g. on hold, consult etc).
For each call information about remote party and status is given.

	Name	**Type**	**Comment**
IMPORT	EXT	SP_TELNO	Own extension
EXPORT	RETURNCODE	SP_RETCODE	Returncode of the telephony gateway
	SYSTEM_RC_X	SP_SYSRC	Returncode of the CTI system
	ERR_TEXT	SP_INFO1	Error description
EXPORT TABLES	EXTCALLS	SPH_CSTATE	List of active calls and parties at own extension

Valid error codes:
000 No error
001 Function not supported
002 Function could not be executed
003 Authorization error
004 Connection error
005 System error
006 Resources not available
013 Extension is not yet registered
016 Extension is not known

10.1.10 SPS_GETQUEUES

Purpose: to return a list of queues.
Two types of lists can be requested. The import parameter TYPE indicates what type of list is requested:
'0' all queues the specified extension is currently logged into
'1' all queues the specified extension has the authority to log into
Please note: In version 5.00A a NULL value may also be passed to the telephony gateway as value of TYPE instead of '0'.

	Name	**Type**	**Comment**
IMPORT	EXT	SP_TELNO	Own extension
	TYPE	SP_QTYPE	Type of list to be returned
EXPORT	RETURNCODE	SP_RETCODE	Returncode of the telephony gateway
	SYSTEM_RC_X	SP_SYSRC	Returncode of the CTI system
	ERR_TEXT	SP_INFO1	Error description
EXPORT TABLES	QUEUES	SPH_QUEUE	ACD Queue/Campaign

Valid error codes:
000 No error
001 Function not supported
002 Function could not be executed
003 Authorization error
004 Connection error
005 System error
006 Resources not available
013 Extension is not yet registered
016 Extension is not known
023 Agent is not logged into a queue

10.1.11 SPS_WAITFORCALL

Purpose: to put the user in wait mode
When the user calls this function, the function should wait and only return, when one of the following events occurs:

1. an incoming call arrives at this extension (→ reason code 01)
2. the wait mode is cancelled using SPS_CANCELWAIT (→ reason code 02)
3. there are currently active calls at the extenstion (→ reason code 04)
4. there is a user-defined reason to return (→ reason code 05)

	Name	Type	Comment
IMPORT	EXT	SP_TELNO	Own extension
	WORKMODE	SP_WRKMODE	Work mode to set before waiting for call (0=no work mode change, for other values: see 9.3 Constants– work mode constants and following note)
EXPORT	HANDLE	SP_HANDLE	Alerting call
	REASON	SP_REASON	Reason for returning the function
	RETURNCODE	SP_RETCODE	Returncode of the telephony gateway
	SYSTEM_RC_X	SP_SYSRC	Returncode of the CTI system
	ERR_TEXT	SP_INFO1	Error description
EXPORT TABLES	EXTCALLS	SPH_CSTATE	List of active calls for own extension
	CALLDATA	SPH_IOCONT	Call-attached data

Please note: The following work modes are valid for the parameter WORKMODE:

No change (00)	No work mode change
Ready (01)	The agent is set to workmode Ready before being put in waitmode.
Work ready (03)	The agent is set to workmode Ready before being put in waitmode and set to workmode Ready after the next call
Work not ready (04)	The agent is set to workmode Ready before being put in waitmode and set to workmode Not ready after the next call

The workmode 'Not ready (02)' is not a valid value of the parameter WORKMODE for this function. Other workmodes cannot be set using this function. If other, customer defined workmodes exist in an installation, they must be set using function SPS_SETWORKMODE (chapter 10.1.7).

Valid error codes:

000 No error
001 Function not supported
002 Function could not be executed
003 Authorization error
004 Connection error
005 System error
006 Resources not available
013 Extension is not yet registered
016 Extension is not known
026 Workmode is not supported
028 Waitmode terminated because gateway is shut down
029 Agent could not be set to waitmode because agent is already waiting

10.1.12 SPS_CANCELWAIT

Purpose: to cancel the wait mode
If this function is called and the specified extenstion is in wait mode, the function SPS_WAITFORCALL, that was called from this extension, has to return immediately with reason code 2 (cancelled). (Of course the function SPS_CANCELWAIT also has to return immediately).

	Name	**Type**	**Comment**
IMPORT	EXT	SP_TELNO	Own extension
	WORKMODE	SP_WRKMODE	Workmode to set after cancelling wait (0=no workmode change, for other values: see 9.3 Constants - work mode constants)
EXPORT	RETURNCODE	SP_RETCODE	Returncode of the telephony gateway
	SYSTEM_RC_X	SP_SYSRC	Returncode of the CTI system
	ERR_TEXT	SP_INFO1	Error description

Valid error codes:
000 No error
001 Function not supported
002 Function could not be executed
003 Authorization error
004 Connection error
005 System error
006 Resources not available
013 Extension is not yet registered
016 Extension is not known
030 Agent is not in waitmode (cancellation not possible)

10.1.13 SPS_ANSWER

Purpose: to connect an arriving call.
The user calls this function to connect the call as he would when picking up the receiver.

	Name	**Type**	**Comment**
IMPORT	EXT	SP_TELNO	Own extension
	HANDLE	SP_HANDLE	Alerting call (blank=let system select call)
EXPORT	ANS_HANDLE	SP_HANDLE	Answered call
	RETURNCODE	SP_RETCODE	Returncode of the telephony gateway
	SYSTEM_RC_X	SP_SYSRC	Returncode of the CTI system
	ERR_TEXT	SP_INFO1	Error description
EXPORT TABLES	EXTCALLS	SPH_CSTATE	List of all active calls for own extension

Valid error codes:
000 No error
001 Function not supported
002 Function could not be executed
003 Authorization error
004 Connection error
005 System error
006 Resources not available
007 Invalid callstate
008 Not authorized to access call handle
013 Extension is not yet registered
016 Extension is not known
031 Call handle not valid
032 No call available

10.1.14 SPS_DEFLECT

Purpose: to transfer an arriving call to another extension without having answered the call first.

	Name	**Type**	**Comment**
IMPORT	EXT	SP_TELNO	Own extension
	DESTINATION	SP_TELNO	Deflect destination
	HANDLE	SP_HANDLE	Call to deflect (blank=let system select alerting call)
EXPORT	RETURNCODE	SP_RETCODE	Returncode of the telephony gateway
	SYSTEM_RC_X	SP_SYSRC	Returncode of the CTI system
	ERR_TEXT	SP_INFO1	Error description
EXPORT TABLES	EXTCALLS	SPH_CSTATE	List of active calls for own extension

Valid error codes:
000 No error
001 Function not supported
002 Function could not be executed
003 Authorization error
004 Connection error
005 System error
006 Resources not available
007 Invalid callstate
008 Not authorized to access call handle
013 Extension is not yet registered
016 Extension is not known
031 Call handle not valid
032 No call available
033 Deflect number not valid

10.1.15 SPS_MAKECALL

Purpose: to initiate an outbound call
This function replaces the function OUTGOING_CALL in former SAPphone interface versions, which is still called, when the telephony gateway has a version number 3.x.

	Name	**Type**	**Comment**
IMPORT	OWNLINEID	SP_TELNO	Own extension
	NUMBER	SP_TELNO	Number to call
EXPORT	HANDLE	SP_HANDLE	Initiated call
	NUMBER_CALLED	SP_TELNO	Number called by telephony gateway
	RETURNCODE	SP_RETCODE	Returncode of the telephony gateway
	SYSTEM_RC_X	SP_SYSRC	Returncode of the CTI system
	ERR_TEXT	SP_INFO1	Error description
EXPORT TABLES	EXTCALLS	SPH_CSTATE	List of active calls for own extension
IMPORT TABLES	CALLDATA	SPH_IOCONT	Call-attached data

The number to be dialed is either in canonical format (+country (area) number) or a dialstring including digits to get outside lines (depending on the settings within SAPphone).
The function returns immediately after the command to initiate a call is sent to the telephone system. If the command could be sent successfully, the function should return with returncode '0000' and return a valid handle. If the command could not be sent successfully, the returncode must be set.
Wether the connection could actually be established or not (destination busy, not answering) does not matter for the returncode of this function.

Valid error codes:
000 No error
001 Function not supported
002 Function could not be executed
003 Authorization error
004 Connection error
005 System error
006 Resources not available
007 Invalid callstate
008 Not authorized to access call handle
013 Extension is not yet registered
016 Extension is not known
034 No line available
035 Destination is not valid
039 Warning: Call attached data could not be transferred to new call

10.1.16 SPS_CONSULT

Purpose: to place a call on hold and initiate a consult call to the destination specified.
The function must perform both actions: place the call identified by REF_HANDLE on hold and initiate a new outbound call to DESTINATION. Also, the call-attached data belonging to the referenced call on hold must be transferred to the new call, so that the destination receiving the consult call will see the call-attached data.
NEXT_STEP describes what the result of the consult is intended to be: a conference or a transfer. When the user chooses 'CONSULT' or when he gives no information at all, the parameter is set to UNKNOWN. The gateway must then decide itself how to react.
If the gateway and the switch do not need to know beforehand what the result of consult will be, it should ignore this parameter. The user is always free to change their mind and try another action.

	Name	**Type**	**Comment**
IMPORT	EXT	SP_TELNO	Own extension
	DESTINATION	SP_TELNO	Number to call for consultation
	REF_HANDLE	SP_HANDLE	Referenced call to place on hold
	NEXT_STEP	CHAR1	Follow-up action after consult (C=Conference, T=Transfer, U=Unknown)
EXPORT	NEW_HANDLE	SP_HANDLE	Handle of consultation call
	HELD_HANDLE	SP_HANDLE	Handle of original, now held call
	RETURNCODE	SP_RETCODE	Returncode of the telephony gateway
	SYSTEM_RC_X	SP_SYSRC	Returncode of the CTI system
	ERR_TEXT	SP_INFO1	Error description
EXPORT TABLES	EXTCALLS	SPH_CSTATE	List of active calls for own extension
IMPORT TABLES	CALLDATA	SPH_IOCONT	Call-attached data (in addition to the data attached to the referenced call)

Valid error codes:
000 No error
001 Function not supported
002 Function could not be executed
003 Authorization error
004 Connection error
005 System error
006 Resources not available
007 Invalid callstate
008 Not authorized to access call handle
013 Extension is not yet registered
016 Extension is not known
031 Call handle not valid
032 No call available
034 No line available
035 Destination is not valid
036 Consult: Active call cannot be put on hold
038 Consult: New call cannot be initiated
039 Warning: Call attached data could not be transferred to new call

10.1.17 SPS_BTRANSFER

Purpose: to tranfer a call to a new destination without contacting the new destination first.

	Name	Type	Comment
IMPORT	EXT	SP_TELNO	Own extension
	HANDLE	SP_HANDLE	Call to transfer
	DESTINATION	SP_TELNO	Destination to transfer call to
EXPORT	RETURNCODE	SP_RETCODE	Returncode of the telephony gateway
	SYSTEM_RC_X	SP_SYSRC	Returncode of the CTI system
	ERR_TEXT	SP_INFO1	Error description
EXPORT TABLES	EXTCALLS	SPH_CSTATE	List of all active calls for own extension, after current transfer.

Valid error codes:
000 No error
001 Function not supported
002 Function could not be executed
003 Authorization error
004 Connection error
005 System error
006 Resources not available
007 Invalid callstate
008 Not authorized to access call handle
013 Extension is not yet registered
016 Extension is not known
031 Call handle not valid
035 Destination is not valid

10.1.18 SPS_TRANSFER

Purpose: to tranfer a call to a new destination after contacting this new destination
This function is called e.g. after SPS_CONSULT, when one call is already on hold and the connection to the new destination is already established. It completes the transfer.

	Name	Type	Comment
IMPORT	EXT	SP_TELNO	Own extension
	HELD_HANDLE	SP_HANDLE	Held handle to transfer
	DEST_HANDLE	SP_HANDLE	Active (!) call handle to which held call is transferred.
EXPORT	RETURNCODE	SP_RETCODE	Returncode of the telephony gateway
	SYSTEM_RC_X	SP_SYSRC	Returncode of the CTI system
	ERR_TEXT	SP_INFO1	Error description
EXPORT TABLES	EXTCALLS	SPH_CSTATE	Info on all open calls for own extension, after current transfer.

Valid error codes:
000 No error
001 Function not supported
002 Function could not be executed
003 Authorization error
004 Connection error
005 System error
006 Resources not available
008 Not authorized to access call handle
013 Extension is not yet registered
016 Extension is not known
042 Active party is not valid (call cannot be transferred, conferenced or alternated)
043 Held party is not valid (call cannot be transferred, conferenced or alternated)
052 Held call has invalid callstate
053 Active call has invalid callstate

10.1.19 SPS_CONFERENCE

Purpose: to initiate a conference call between at least 3 parties
This function is called e.g. after SPS_CONSULT, when one call (could itself be a conference call) is already on hold and the connection to the new destination is already established. It initiates the conference. The conference handle that is returned as NEW_HANDLE should be the same handle as HELD_HANDLE (see chapter 4.4)

	Name	**Type**	**Comment**
IMPORT	EXT	SP_TELNO	Own extension
	HELD_HANDLE	SP_HANDLE	Call to which the new call is added
	DEST_HANDLE	SP_HANDLE	Call which is added to conference
EXPORT	NEW_HANDLE	SP_HANDLE	Conference call
	RETURNCODE	SP_RETCODE	Returncode of the telephony gateway
	SYSTEM_RC_X	SP_SYSRC	Returncode of the CTI system
	ERR_TEXT	SP_INFO1	Error description
EXPORT TABLES	EXTCALLS	SPH_CSTATE	List of all open calls at own extension

Valid error codes:
000 No error
001 Function not supported
002 Function could not be executed
003 Authorization error
004 Connection error
005 System error
006 Resources not available
008 Not authorized to access call handle
013 Extension is not yet registered
016 Extension is not known
040 Consult call initiated as transfer, conference not possible
042 Active party is not valid (call cannot be transferred, conferenced or alternated)
043 Held party is not valid (call cannot be transferred, conferenced or alternated)
052 Held call has invalid callstate
053 Active call has invalid callstate

10.1.20 SPS_RECONNECT

Purpose: to drop an active call and return to a held call.
This function could e.g. be called after SPS_CONSULT, when one call is on hold, another one is active. This function has to perform both actions: dropping the active call and reconnecting to the held call.

	Name	**Type**	**Comment**
IMPORT	EXT	SP_TELNO	Own extension
	HELD_HANDLE	SP_HANDLE	Held call to be reconnected
	DROP_HANDLE	SP_HANDLE	Active call to drop
EXPORT	RETURNCODE	SP_RETCODE	Returncode of the telephony gateway
	SYSTEM_RC_X	SP_SYSRC	Returncode of the CTI system
	ERR_TEXT	SP_INFO1	Error description
EXPORT TABLES	EXTCALLS	SPH_CSTATE	List of all open calls at own extension

Valid error codes:
000 No error
001 Function not supported
002 Function could not be executed
003 Authorization error
004 Connection error
005 System error
006 Resources not available
008 Not authorized to access call handle
013 Extension is not yet registered
016 Extension is not known
037 Reconnect: Held call cannot be retrieved from hold
041 No call available for reconnecting
044 Reconnect: Active call cannot be dropped
052 Held call has invalid callstate
053 Active call has invalid callstate

10.1.21 SPS_HOLD

Purpose: to place a call on hold.

	Name	Type	Comment
IMPORT	EXT	SP_TELNO	Own extension
	HANDLE	SP_HANDLE	Active call to be put on hold
EXPORT	RETURNCODE	SP_RETCODE	Returncode of the telephony gateway
	SYSTEM_RC_X	SP_SYSRC	Returncode of the CTI system
	ERR_TEXT	SP_INFO1	Error description
EXPORT TABLES	EXTCALLS	SPH_CSTATE	List of all open calls at own extension

Valid error codes:
000 No error
001 Function not supported
002 Function could not be executed
003 Authorization error
004 Connection error
005 System error
006 Resources not available
007 Invalid callstate
008 Not authorized to access call handle
013 Extension is not yet registered
016 Extension is not known
031 Call handle not valid

10.1.22 SPS_UNHOLD

Purpose: to release a call from hold.

	Name	Type	Comment
IMPORT	EXT	SP_TELNO	Own extension
	HANDLE	SP_HANDLE	Held call to activate
EXPORT	RETURNCODE	SP_RETCODE	Returncode of the telephony gateway
	SYSTEM_RC_X	SP_SYSRC	Returncode of the CTI system
	ERR_TEXT	SP_INFO1	Error description
EXPORT TABLES	EXTCALLS	SPH_CSTATE	List of all open calls at own extension

Valid error codes:
000 No error
001 Function not supported
002 Function could not be executed
003 Authorization error
004 Connection error
005 System error
006 Resources not available
007 Invalid callstate
008 Not authorized to access call handle
013 Extension is not yet registered
016 Extension is not known
031 Call handle not valid

10.1.23 SPS_ALTERNATE

Purpose: to switch between two calls, one on hold, one active.
This function is called e.g. after SPS_CONSULT, when one call is already on hold and another active call is available.

	Name	Type	Comment
IMPORT	EXT	SP_TELNO	Own extension
	ACTIVE_HANDLE	SP_HANDLE	Active call to be put on hold
	HELD_HANDLE	SP_HANDLE	Held call to be activated
EXPORT	RETURNCODE	SP_RETCODE	Returncode of the telephony gateway
	SYSTEM_RC_X	SP_SYSRC	Returncode of the CTI system
	ERR_TEXT	SP_INFO1	Error description
EXPORT TABLES	EXTCALLS	SPH_CSTATE	Info on all calls at own extension

Valid error codes:
000 No error
001 Function not supported
002 Function could not be executed
003 Authorization error
004 Connection error
005 System error
006 Resources not available
008 Not authorized to access call handle
013 Extension is not yet registered
016 Extension is not known
042 Active party is not valid (call cannot be transferred, conferenced or alternated)
043 Held party is not valid (call cannot be transferred, conferenced or alternated)
052 Held call has invalid callstate
053 Active call has invalid callstate

10.1.24 SPS_DROPPARTY

Purpose: to drop one party out of a conference call
This function is called, when the user wants to drop one party out of a conference with the other parties remaining in the conference. The party to be dropped can be the user himself or another party, if the switch and the CTI system allow the dropping of other parties (e.g. with call center supervisor rights).

	Name	Type	Comment
IMPORT	EXT	SP_TELNO	Own extension
	HANDLE	SP_HANDLE	Call from which party is dropped
	PARTY	SP_TELNO	Party to drop from call
EXPORT	RETURNCODE	SP_RETCODE	Returncode of the telephony gateway
	SYSTEM_RC_X	SP_SYSRC	Returncode of the CTI system
	ERR_TEXT	SP_INFO1	Error description
EXPORT TABLES	EXTCALLS	SPH_CSTATE	List of all open calls at own extension

Valid error codes:
000 No error
001 Function not supported
002 Function could not be executed
003 Authorization error
004 Connection error
005 System error
006 Resources not available
007 Invalid callstate
008 Not authorized to access call handle
013 Extension is not yet registered
016 Extension is not known
025 Party is not connected to call
031 Call handle not valid

10.1.25 SPS_DROPCALL

Purpose: to drop a call
This function is used to drop a call between two parties.

	Name	Type	Comment
IMPORT	EXT	SP_TELNO	Own extension
	HANDLE	SP_HANDLE	Call to drop
EXPORT	RETURNCODE	SP_RETCODE	Returncode of the telephony gateway
	SYSTEM_RC_X	SP_SYSRC	Returncode of the CTI system
	ERR_TEXT	SP_INFO1	Error description
EXPORT TABLES	EXTCALLS	SPH_CSTATE	List of all open calls at own extension

Valid error codes:
000 No error
001 Function not supported
002 Function could not be executed
003 Authorization error
004 Connection error
005 System error
006 Resources not available
007 Invalid callstate
008 Not authorized to access call handle
013 Extension is not yet registered
016 Extension is not known
031 Call handle not valid

10.1.26 SPS_DROPALL

Purpose: to drop all calls for the extension specified.
This function has the same effect as a SPS_DROPCALL on every call of the extension. The telephony gateway has to know by itself what calls to drop. If one or more calls could not be dropped, error code '045' must be returned and the parameter EXTCALLS must contain the calls that are still active (and could not be dropped).

	Name	Type	Comment
IMPORT	EXT	SP_TELNO	Own extension
EXPORT	RETURNCODE	SP_RETCODE	Returncode of the telephony gateway
	SYSTEM_RC_X	SP_SYSRC	Returncode of the CTI system
	ERR_TEXT	SP_INFO1	Error description
EXPORT TABLES	EXTCALLS	SPH_CSTATE	Info on all calls for own extension after drop. Empty if dropall was successful.

Valid error codes:
000 No error
001 Function not supported
002 Function could not be executed
003 Authorization error
004 Connection error
005 System error
006 Resources not available
013 Extension is not yet registered
016 Extension is not known
032 No call available
045 Not all calls can be dropped

10.1.27 SPS_GENERIC

Purpose: to pass data from R/3 to the gateway for generic purposes

	Name	Type	Comment
EXPORT	RETURNCODE	SP_RETCODE	Returncode of the telephony gateway
	SYSTEM_RC_X	SP_SYSRC	Returncode of the CTI system
	ERR_TEXT	SP_INFO1	Error description
IMPORT TABLES	PRIVATEDATA	SPH_IOBJ	Key-value-pairs

Note: This function is not part of the interface version 4.00A, it will never be called from an R/3 rel. 4.5A.

Valid error codes:
000 No error
001 Function not supported
002 Function could not be executed
003 Authorization error
004 Connection error
005 System error
006 Resources not available

10.2 Reporting functions

The following functions are implemented within R/3 and can be called by the telephony gateway.

10.2.1 SPS_GET_LINES_PER_SERVER

A list of all extensions that are assigned to the specified telephony gateway is returned. The information comes from the SAPphone administration tables. Purpose: To limit the number of surveyed extensions to those who are using the SAPphone functionality. This function replaces the function module LIST_INCOMING_ACTIVE_LINES, which is still supported. However, from Rel. 4.5A on the function SPS_GET_LINES_PER_SERVER should be called.

	Name	**Type**	**Comment**
IMPORT	SERVER	SP_SERV_ID	Server-ID as maintained within SAPphone
EXPORT TABLES	LINES	SPH_LINES	List of extensions
	LINES_WITH_IP	SPH_LINEIP	List of extensions including user and workcenter IP address (only when user is logged on to R/3)

The tables parameter LINES_WITH_IP is supported from interface version 5.01ASP onwards only.

10.2.2 SPS_NEW_CALL

This function replaces the function module INCOMING_CALL, which is still supported. However, as of Rel. 4.5A the function SPS_NEW_CALL should be used instead, if a new call – inbound call or outbound PD call – is to be reported at all (alternatively, the functions SPS_WAITFORCALL or SPS_GETCALLSTATE can be used by the R/3 system). SPS_NEW_CALL should only be called to report arriving calls, if the user did not call SPS_WAITFORCALL before (is in waitmode).

	Name	**Type**	**Comment**
IMPORT	REMOTE_PARTY	SP_TELNO	Remote party (in case of incoming call: calling number (ANI), in case of outgoing call: called number)
	DNIS	SP_TELNO	Dialed number (e.g. service number)
	OWN_EXTENSION	SP_TELNO	Own extension
	SERVER_NAME	SP_SERV_ID	Own telephony gateway (ID as maintained within SAPphone)
	HANDLE	SP_HANDLE	Call handle
	OPTIONS	SPH_OPTIONS	Flags to control the function. If first character of OPTION1 is set to ‚X‘, the function only returns the found caller data without popup.
	PROTOCOL	SP_BOOLEAN	For internal use only
EXPORT	NAME	SP_VAL_55	Name of caller
	COMPANY	SP_VAL_55	Company of caller
	CITY	SP_VAL_55	City of caller
	COUNTRY	SP_VAL_55	Country of caller
IMPORT TABLES	EXTCALLS	SPH_CSTATE	
	CALLDATA	SPH_IOCONT	Call-attached data
EXCEPTIONS	ERROR_NUMBER_DIALED		The combination of own extension and server name is not valid within SAPphone
	ERROR_EXPRESS_SEND		The express popup to inform the user could not be sent.

10.2.3 SPS_CALL_ENDED

If a call could not be connected, and if the calling number is available and a caller can be identified, this function sends a message to the user. If no user can be determined dynamically (if no user is logged on to the receiving workcenter), the user responsible for the workcenter is informed. The user can switch this function off in the user settings. This function replaces the function module INCOMING_CALL_ENDED, which is still supported. However from Rel. 4.5A on, SPS_CALL_ENDED should be called.

	Name	**Type**	**Comment**
IMPORT	OWN_EXTENSION	SP_TELNO	Own extension
	SERVER_NAME	SP_SERV_ID	Own telephony gateway
	HANDLE	SP_HANDLE	Call handle
	FLAG_CONNECTION	SP_BOOLEAN	Connection between caller and called user was established.
	REMOTE_PARTY	SP_TELNO	Remote party: phone number of caller.

10.3 Functions concerning predictive dialing / power dialing

Please also refer to chapter 5 Predictive Dialing / Power Dialing.

Note: In release 4.5B, there are incompatible changes to release 4.5A.

10.3.1 SPS_PDLISTTRANSFER

Purpose: Transfers a PD list to the dialer. This function is called by R/3 and implemented in the dialer (see also paragraph 5.5).

	Name	**Type**	**Comment**
IMPORT	CAMPAIGNID	SP_QUEUE	campaign to which the PD list is assigned, mandatory
	CALLBACK_FCT	RS38L-NAME (CHAR 30)	*No standard functionality! *** R/3 function which has to be called to report CTI events to R/3; optional
	LOGSYS	LOGSYS (CHAR 10)	*No standard functionality! *** determines R/3 target system for CTI events; optional
	TRACEID	SWT_HANDLE (NUMC 12)	needed for minuting list transfer
Import/Export TABLES	PDLIST	SPH_PDCALL	PD list *
EXPORT	RETURNCODE	SP_RETCODE	Returncode of the telephony gateway
	SYSTEM_RC_X	SP_SYSRC	Returncode of the CTI system
	ERR_TEXT	SP_INFO1	Error description

** Note: The import parameters CALLBACK_FCT and LOGSYS are *NOT* supported in a standard R/3 release (version 4.5B or 4.6A). They are only included in this function module for usage in customer modifications!

Valid error codes:

000 No error
001 Function not supported
002 Function could not be executed
003 Authorization error
004 Connection error
005 System error
006 Resources not available
046 Function could not be executed completely *
047 No entries could be created
048 Campaign ID is unknown
049 Campaign ID is not available

* In the case of the error code 046 (Function could not be executed completely), the dialer is expected (if possible) to return the PDLIST table with all entries. This can be done either immediately or later (via the function SPS_PD_STATUS). The field PDCSTATE indicates whether an entry has been handled correctly or not. If no error has occurred, this field is left blank, in the event of an error the following PDCSTATEs should be set:

PDCSTATE	Description
CID	CallID could not be created for this entry (field CALLID is left blank)
TNF	Telephone Number Format incorrect
TZF	Time Zone Format incorrect
DAT	DATe range exceeds date range of the corresponding campaign
UDF	UnDeFined: combination of the above listed errors or any other error

Release 4.5A only:

IMPORT	LISTID	SP_QUEUE	*Not used*
	THEMEID	SP_QUEUE	*Not used*
	PACKAGEID	CHAR3 (CHAR 3)	*Not used*

10.3.2 SPS_PD_STATUS

This function is implemented in R/3 and is called by the SAPphone interface itself (dialer keys are passed back to R/3 immediately after list transfer) or by the dialer (dialer keys are passed back to R/3 later). Please also refer to paragraphs 5.4 and 5.5.

Purpose:

- This function maintains the mapping table, i.e. creates mapping relations between the R/3 key (triple: OBJTYPE, OBJKEY, LOGSYS) for a planned call (R/3) and the dialer key (CALLID) for a planned call (dialer).
- A corresponding trace entry (protocol) is written (by using TRACEID and CAMPAIGNID).

	Name	**Type**	**Comment**
IMPORT	CAMPAIGNID	SP_QUEUE	campaign to which the list is assigned, mandatory
	TRACEID	SWT_HANDLE (NUMC 12)	needed for protocolling of list transfer, mandatory
Import TABLES	PDLIST	SPH_PDCALL	PD list*

* In the case of errors, the error type has to be specified in the field PDCSTATE for each entry. Please see annotations to error code 046 in function SPS_PDLISTTRANSFER.

10.3.3 SPS_MODIFY_PDCALL

Purpose: Modifies an individual planned call in the dialer. This function is called by R/3 and implemented in the dialer. The dialer key and the complete structure of a PD list entry is transferred to the dialer. The dialer is expected to update the corresponding PD list entry .

Delta mechanism:

The dialer is expected to modify only these fields of the structure PDCALL which are not initial (initial means blank for character fields, ‚000000‘ for time fields and ‚00000000‘ for date fields).

	Name	**Type**	**Comment**
IMPORT	CALLID	SP_CALLID	unique call center key for an individual PD call
	PDCALL	SPH_PDCALL	PD list entry
EXPORT	ERR_DETAILED	CHAR3	detailed error information in case of error code 046
	RETURNCODE	SP_RETCODE	Returncode of the telephony gateway
	SYSTEM_RC_X	SP_SYSRC	Returncode of the CTI system
	ERR_TEXT	SP_INFO1	Error description

Valid error codes:
000 No error
001 Function not supported
002 Function could not be executed
003 Authorization error
004 Connection error
005 System error
006 Resources not available
046 Function could not be executed completely *
050 PD CALLID is unknown
051 Modifying not possible, PD call is in wrong state

* In the case of error code 046, the detailed error type has to be specified in the export parameter ERR_DETAILED. Please see annotations to error code 046 in function SPS_PDLISTTRANSFER. Please note, that instead of the field PDCSTATE the export parameter ERR_DETAILED has to be used.

Incompatible changes between R/3 release 4.5A and 4.5B:

- In release 4.5A the whole structure PDCALL has to be used for modifying a planned call in the dialer. A delta mechanism is not available. The dialer is expected to modify the planned call in the dialer by using the complete structure.
- In 4.5B: new export parameter ERR_DETAILED.

10.3.4 SPS_STATCAMPAIGN

Purpose: Retrieves statistical data from the dialer. This function is called by R/3 and implemented in the dialer.

	Name	Type	Comment
IMPORT	CAMPAIGNID	SP_QUEUE	Specifies a campaign, optional
	AGENT_NAME	SP_CCUSER	Specifies an agent, optional, default: SPACE. By specifying an agent the field CAMPAIGNID can be left blank and the dialer is expected to identify the campaigns this agent is assigned to.
	STARTDATE	SP_SDATE	Specifies date range used for statistics, optional, default: SPACE
	ENDDATE	SP_EDATE	
EXPORT TABLE	PDSTATISTC	SPH_PDCHIS	statistical data list
EXPORT	RETURNCODE	SP_RETCODE	Returncode of the telephony gateway
	SYSTEM_RC_X	SP_SYSRC	Returncode of the CTI system
	ERR_TEXT	SP_INFO1	Error description

Valid error codes:
000 No error
001 Function not supported
002 Function could not be executed
003 Authorization error
004 Connection error
005 System error
006 Resources not available
020 Agents call center user-ID is not valid
048 Campaign ID is unknown
049 Campaign ID is not available

10.3.5 SPS_DELETE_PDCALL

Purpose: Deletes one planned call in the dialer. This function is called by R/3 and implemented in the dialer.

	Name	Type	Comment
IMPORT	CALLID	SP_CALLID	unique call center key for an individual PD call
EXPORT	RETURNCODE	SP_RETCODE	Returncode of the telephony gateway
	SYSTEM_RC_X	SP_SYSRC	Returncode of the CTI system
	ERR_TEXT	SP_INFO1	Error description

Valid error codes:
000 No error
001 Function not supported
002 Function could not be executed
003 Authorization error
004 Connection error
005 System error
006 Resources not available
050 PD CALLID is unknown
054 Deleting not possible, PD call is in wrong state

10.3.6 SPS_CAMPAIGNS_GET

Purpose: Determines all R/3 campaigns that are completely or only partly active in the specified date range. This function is called by R/3 and implemented in the dialer.

	Name	Type	Comment
IMPORT	STARTDATE	SP_SDATE	Specifies date range optional, default: SPACE
	ENDDATE	SP_EDATE	
EXPORT TABLE	CAMPAIGNS	SPH_CAMPGS	list of campaigns
EXPORT	RETURNCODE	SP_RETCODE	Returncode of the telephony gateway
	SYSTEM_RC_X	SP_SYSRC	Returncode of the CTI system
	ERR_TEXT	SP_INFO1	Error description

Valid error codes:
000 No error
001 Function not supported
002 Function could not be executed
003 Authorization error
004 Connection error
005 System error
006 Resources not available

10.3.7 SPS_ASSIGNED_CAMPAIGNS_GET

Purpose: Determines all R/3 campaigns into which the planned call (R/3) has been downloaded and in which the corresponding planned call (dialer) still exist. This function is called by R/3 and implemented in the dialer.

	Name	Type	Comment
IMPORT	CALLID	SP_CALLID	unique call center key for an individual PD call
EXPORT TABLE	CAMPAIGNS	SPH_CAMPGS	list of assigned campaigns
EXPORT	RETURNCODE	SP_RETCODE	Returncode of the telephony gateway
	SYSTEM_RC_X	SP_SYSRC	Returncode of the CTI system
	ERR_TEXT	SP_INFO1	Error description

Valid error codes:
000 No error
001 Function not supported
002 Function could not be executed
003 Authorization error
004 Connection error
005 System error
006 Resources not available
050 PD CALLID is unknown

10.4 Functions for call-attached data from 4.5B onwards

For a detailed description of how to use the functions listed in this paragraph, please refer to the chapter "Call attached data". The functions described in this chapter are called by R/3 and must be implemented in the telephony gateway.

10.4.1 SPS_GETCALLINFO

Purpose: to return information about a specific call such as call-attached data and additional call info.

	Name	**Type**	**Comment**
IMPORT	EXT	SP_TELNO	Own extension
	HANDLE	SP_HANDLE	ID of call whose attached data are read
EXPORT	RETURNCODE	SP_RETCODE	Returncode of the telephony gateway
	SYSTEM_RC_X	SP_SYSRC	Returncode of the CTI system
	ERR_TEXT	SP_INFO1	Error description
EXPORT TABLES	CALLDATA	SPH_IOCONT	Call-attached data

In addition to the data attached to the call, the tables parameter CALLDATA must contain the following information:

Objectname	**Instance**	**Keyname**	**Type of value**	**Description**
CINFO	0001	DATE	CHAR8 (YYYYMMDD)	Date of call
		TIME	CHAR6 (HHMMSS)	Time of call (starttime)
		LOCALPARTY	CHAR30	Local party (own extension)
		REMOTEPARTY	CHAR30	Remote party (calling number in case of incoming calls)
		DNIS	CHAR30	Originally dialed number in case of incoming calls, e.g. service number

This information must be provided by the gateway. It should only be passed to R/3 with this function and not every time that call-attached data is reported to R/3 (e.g. with SPS_NEW_CALL, SPS_WAITFORCALL)

Valid error codes:
000 No error
001 Function not supported
002 Function could not be executed
003 Authorization error
004 Connection error
005 System error
006 Resources not available
007 Invalid callstate
008 Not authorized to access call handle
031 Call handle not valid
046 Function could not be executed completely

10.4.2 SPS_SETCALLINFO

Purpose: to set call-attached data for a specific call. The list of data specified in the interface replaces all existing call-attached data of the call.

	Name	**Type**	**Comment**
IMPORT	HANDLE	SP_HANDLE	ID of call whose attached data are set
EXPORT	RETURNCODE	SP_RETCODE	Returncode of the telephony gateway
	SYSTEM_RC_X	SP_SYSRC	Returncode of the CTI system
	ERR_TEXT	SP_INFO1	Error description
IMPORT/EXPORT TABLES	CALLDATA	SPH_IOCONT	Call-attached data

When the function is called, the tables parameter CALLDATA contains the data that should be set. If all lines could successfully be attached to the call, the returncode must be set to '000'. The tables parameter must then return the complete new list of call-attached data. If one or more lines of the table could not be attached, the returncode must be set to '046' and the tables parameter must return the lines that could not be attached.

Valid error codes:
000 No error
001 Function not supported
002 Function could not be executed
003 Authorisation error
004 Connection error
005 System error
006 Resources not available
007 Invalid callstate
008 Not authorized to access call handle
031 Call handle not valid
046 Function could not be executed completely

10.4.3 SPS_APPENDCALLINFO

Purpose: to add data to the list of existing call-attached data of a specific call

	Name	**Type**	**Comment**
IMPORT	HANDLE	SP_HANDLE	ID of call whose attached data are added
EXPORT	RETURNCODE	SP_RETCODE	Returncode of the telephony gateway
	SYSTEM_RC_X	SP_SYSRC	Returncode of the CTI system
	ERR_TEXT	SP_INFO1	Error description
IMPORT/EXPORT TABLES	CALLDATA	SPH_IOCONT	Call-attached data

Only one object instance at a time can be appended to the list of existing call-attached data by this function. When the function is called, the tables parameter CALLDATA contains the data that should be appended. The object instance number is empty. The gateway must determine the next number based on the object instances already attached to the call. Important: call-attached data can also be set using SPS_SETCALLINFO. In this case, the calling application is responsible for setting the object instance number. When determining the next object instance number, the gateway must take all existing object instances into account. If the object instance could successfully be attached to the call, the returncode must be set to '000'. The tables parameter then must return the complete new list of call-attached data. If one or more lines of the table could not be appended, the returncode must be set to '046' and the tables parameter must return the lines that could not be appended.

Valid error codes:
000 No error
001 Function not supported
002 Function could not be executed
003 Authorization error
004 Connection error
005 System error
006 Resources not available
007 Invalid callstate
008 Not authorized to access call handle
031 Call handle not valid
046 Function could not be executed completely

10.4.4 SPS_DELETECALLINFO

Purpose: to delete data of the list of existing call-attached data of a specific call

	Name	Type	Comment
IMPORT	HANDLE	SP_HANDLE	ID of call whose attached data are deleted
EXPORT	RETURNCODE	SP_RETCODE	Returncode of the telephony gateway
	SYSTEM_RC_X	SP_SYSRC	Returncode of the CTI system
	ERR_TEXT	SP_INFO1	Error description
IMPORT/EXPORT TABLES	CALLDATA	SPH_IOCONT	Call-attached data

When the function is called, the tables parameter CALLDATA contains the data that should be deleted. If all lines could successfully be deleted, the returncode must be set to '000'. The tables parameter then must return the complete new list of call-attached data. If one or more lines could not be deleted, the returncode must be set to '046' and the tables parameter must return the lines that could not be deleted.

To delete all entries belonging to one object instance, the object name and instance number must be specified. The wildcard symbol '*' must be entered in all other fields (the wildcard symbol for the instance number is '9999'). Example:
The following data is attached to the call 12345:

Objectname	Instance	Keyname	Value
CIC_CLIPBOARD	0001	REFOBJTYPE	SCUSTOMER
CIC_CLIPBOARD	0001	OBJKEY	00000256
CIC_CLIPBOARD	0002	REFOBJTYPE	SFLIGHT
CIC_CLIPBOARD	0002	OBJKEY	LH4875

To delete only the object key of the second object instance, SPS_DELETECALLINFO would be called with the tables parameter CALLDATA as follows:

CIC_CLIPBOARD	0002	OBJKEY	LH4875

or

CIC_CLIPBOARD	0002	OBJKEY	*

To delete the second objectinstance completely, SPS_DELETECALLINFO is called with the tables parameter CALLDATA as follows:

CIC_CLIPBOARD	0002	REFOBJTYPE	SFLIGHT
CIC_CLIPBOARD	0002	OBJKEY	LH4875

or

CIC_CLIPBOARD	0002	*	*

To delete all clipboard data, SPS_DELETECALLINFO is called with the tables parameter CALLDATA looking as follows:

CIC_CLIPBOARD	0001	REFOBJTYPE	SCUSTOMER
CIC_CLIPBOARD	0001	OBJKEY	00000256
CIC_CLIPBOARD	0002	REFOBJTYPE	SFLIGHT
CIC_CLIPBOARD	0002	OBJKEY	LH4875

or

CIC_CLIPBOARD	9999	*	*

To delete all call-attached data of a call, SPS_DELETEINFO is called with the tables parameter CALLDATA looking as follows:

*	9999	*	*

Valid error codes:
000 No error
001 Function not supported
002 Function could not be executed
003 Authorization error
004 Connection error
005 System error
006 Resources not available
007 Invalid callstate
008 Not authorized to access call handle
031 Call handle not valid
046 Function could not be executed completely

10.5 Functions for call-attached data in interface version 4.00A

These functions are used for attaching data to a call in Rel. 4.5A only. The are implemented in the telephony gateway and are called by R/3.

10.5.1 SPS_CREATEINFO

	Name	Type	Comment
IMPORT	HANDLE	SP_HANDLE	Call that the attached data belongs to
	OBJECT_NAME	SP_IONAM	Name of the object that is to be created
EXPORT	OBJECT_NUMBER	SP_IONUM	Instance number assigned by telephony gateway
	RETURNCODE	SP_RETCODE	Returncode of the telephony gateway
	SYSTEM_RC_X	SP_SYSRC	Returncode of the CTI system
	ERR_TEXT	SP_INFO1	Error description
IMPORT TABLES	IOBJECT_DESC	SPH_IODESC	List of keynames and typedescription of value field

This function returns a new object instance number for a given object name. This is the only function that returns that number.

Valid error codes:
000 No error
001 Function not supported
002 Function could not be executed
003 Authorization error
004 Connection error
005 System error
006 Resources not available
007 Invalid callstate
008 Not authorized to access call handle
031 Call handle not valid

10.5.2 SPS_SETINFO

	Name	Type	Comment
IMPORT	HANDLE	SP_HANDLE	Call that the attached data belongs to
	OBJECT_NAME	SP_IONAM	Name of object whose key-value-pairs are set
	OBJECT_NUMBER	SP_IONUM	Number of object whose key-value-pairs are set
EXPORT	RETURNCODE	SP_RETCODE	Returncode of the telephony gateway
	SYSTEM_RC_X	SP_SYSRC	Returncode of the CTI system
	ERR_TEXT	SP_INFO1	Error description
IMPORT TABLES	IOBJECT	SPH_IOBJ	List of key-value-pairs

This function sets the key-value-pairs for an object instance. The instance number (OBJECT_NUMBER) must be requested from the gateway in advance using SPS_CREATEINFO.

Valid error codes:
000 No error
001 Function not supported
002 Function could not be executed
003 Authorization error
004 Connection error
005 System error
006 Resources not available
007 Invalid callstate
008 Not authorized to access call handle
031 Call handle not valid

10.6 Functions for monitoring external components from rel. 4.5B onwards

These functions are implemented in the telephony gateway and are called by R/3.

10.6.1 SMON_TRACE_SET

Purpose: To switch between the trace level ON and OFF

	Name	Type	Comment
IMPORT	ON_OFF	SMON_TR_ST	Trace level ('X' = ON, space = OFF)
	COMPONENT	SMON_CO_NA	Name of component, for which the status should be set
EXCEPTION	EXEC_FAILED		Function could not be executed

10.6.2 SMON_TRACE_UPLOAD

Purpose: To upload the trace text.

	Name	Type	Comment
IMPORT	FROM_DATE	CHAR8	Select trace entries from this date onwards (in format YYYYMMDD)
	FROM_TIME	CHAR6	Select trace entries from this time ownards (in format HHMMSS)
	UNTIL_DATE	CHAR8	Select trace entries until this date (in format YYYYMMDD)
	UNTIL_TIME	CHAR6	Select trace entries until this time (in format HHMMSS)
	TYPE	SMON_TR_TY	Type of selected messages ('E' = errors, 'W' = warnings and errors, 'I' = errors, warning and information (default))
	COMPONENT	SMON_CO_NA	Name of component, for which the trace is to be uploaded
	MAX_LINES	INT4	Max. number of lines that are to be uploaded
EXPORT	MAX_LINES_EXCEEDED	CHAR1	More than the specified number of lines are available = 'X'
EXPORT TABLES	TRACE	SMON_TRACE	Trace entries
EXCEPTION	EXEC_FAILED		Function could not be executed
	INVALID_PARAMETER		The parameter combination is not valid: e.g. one of the dates is not in the correct format, from date is higher than until date, message type is unknown
	COMPONENT_UNKNOWN		The component for which the trace is to be uploaded is unknown to the gateway

10.6.3 SMON_TRACE_EXPLAIN

Purpose: To provide more information and problem-solving hints for each returncode and trace id.

	Name	Type	Comment
IMPORT	MSG_ID	SP_SYSRC	Gateway specific returncode
EXPORT TABLES	HELP	SMON_HELP	Help text
EXCEPTION	EXEC_FAILED		Function could not be executed

10.6.4 SMON_COMP_STATE_RETRIEVE

Purpose: To obtain a list of all subcomponents of a gateway including their state and the availability and level of trace information.

	Name	Type	Comment
EXPORT TABLES	COMPONENT_LIST	SMON_COMPO	List of components
EXCEPTION	EXEC_FAILED		Function could not be executed

11 Error codes

To provide the user with as much information as possible in the event of an error, (almost) all of the functions called by R/3 have 3 export parameters: RETURNCODE, SYSTEM_RC_X and ERR_TEXT.

RETURNCODE: The returncode is used to return an error code generated in the telephony gateway. This chapter contains a list of valid error codes. Although the returncode is generated in the telephony gateway, the error may have occurred in the underlying CTI system. To distinguish between errors originating in the telephony gatway and errors originating in the CTI system, the returncode has the following structure: AXXX. A must be set to the value '0' if the error originated in the telephony gateway, or to the value '1' if the error originated in the underlying CTI system. XXX must have one of the values defined in this chapter.

SYSTEM_RC_X: This is the returncode of the underlying CTI system. It should provide further information in combination with the CTI system documentation.
ERR_TEXT: This 50-character short text should be set by the telephony gateway to provide additional information.

The list of valid error codes can be divided in two parts: specific errors and general errors. The general error codes should only be used, if none of the specific error codes applies to the situation. Whenever a general error code is used, it is highly recommended that you give values to SYSTEM_RC_X and ERR_TEXT to provide the user with more specific information.

General error codes

001	Function not supported	Either the telephony gateway or the CTI system does not support the desired action (e.g. not implemented)
002	Function could not be executed	The function could not be executed. SYSTEM_RC_X and ERR_TEXT should contain further information.
003	Authorisation error	There is an authorization error. SYSTEM_RC_X and ERR_TEXT should contain further information. This error code should only be used, if the situation can be solved by changing the user or extension rights. If the error cannot be corrected by configuration, this error code should not be used.
004	Connection error	Connection to one of the subsystems is not available. SYSTEM_RC_X and ERR_TEXT should contain further information
005	System error	General system error. SYSTEM_RC_X and ERR_TEXT should contain further information
006	Resources not available	One of the needed resources is not available. SYSTEM_RC_X and ERR_TEXT should contain further information
007	Invalid callstate	Desired action is not allowed on calls in that state (state information is displayed to the user and is taken from the tables parameter EXTCALLS of the particular function).
008	Not authorized to access call handle	The call itself is still valid, but the extension is not allowed to perform any action with this call (e.g. no longer connected to the call).

Specific error codes

000	No error
009	Telephony gateway and SAPphone not compatible
010	Max. number of registered extensions reached
011	Extenstion already registered with different IP-address
012	Network error for IP-address
013	Extension is not yet registered
014	Deregistration currently not possible due to active call
015	Deregistration currently not possible because user still logged into queue
016	Extension is not known
017	Queue is unknown

018	Queue is not available
019	Max. number of users logged in reached
020	Agents call center user-ID is not valid
021	Agents call center user-ID is locked
022	Agents call center password is not valid
023	Agent is not logged into a queue
024	Logging out currently not possible due to active call
025	Party is not connected to call
026	Work mode is not supported
027	Agent can temporarily not be set into work mode
028	Wait mode terminated because gateway is shut down
029	Agent could not be set to waitmode because agent is already waiting
030	Agent is not in wait mode (cancellation not possible)
031	Call handle not valid
032	No call available
033	Deflect number not valid
034	No line available
035	Destination is not valid
036	Consult: Active call cannot be put on hold
037	Reconnect: Held call cannot be retrieved from hold
038	Consult: New call cannot be initiated
039	Warning: Call attached data could not be transferred to new call
040	Consult call initiated as transfer, conference not possible
041	No call available for reconnecting
042	Active party is not valid (call cannot be transferred, conferenced or alternated)
043	Held party is not valid (call cannot be transferred, conferenced or alternated)
044	Reconnect: Active call cannot be dropped
045	Not all calls can be dropped
046	Function could not be executed completely
047	No entries could be created
048	Campaign ID is unknown
049	Campaign ID is not available
050	PD CALLID is unknown
051	Modifying not possible, PD call is in wrong state
052	Held call has invalid callstate
053	Active call has invalid callstate
054	Deleting not possible, PD call is in wrong state

12 Delta between interface versions

12.1 4.00A → 5.00A

New defined error codes

Functions no longer supported:
- SPS_CREATEINFO
- SPS_SETINFO

New functions for call-attached data:
- SPS_SETCALLINFO
- SPS_DELETECALLINFO
- SPS_APPENDCALLINFO

New functions for monitoring of external component:
- SMON_TRACE_SET
- SMON_TRACE_UPLOAD
- SMON_TRACE_EXPLAIN
- SMON_COMP_STATE_RETRIEVE

New functions for predictive dialing / power dialing
- SPS_DELETE_PDCALL
- SPS_CAMPAIGNS_GET
- SPS_ASSIGNED_CAMPAIGNES_GET

Changes in functions:

SPS_PDLISTTRANSFER:	parameters deleted:	LISTID
		THEMEID
		PACKAGEID
	New parameters:	LOGSYS
SPS_MODIFY_PDCALL:	New parameters:	ERR_DETAILED
	New delta mechanism	
SPS_GETCALLINFO:	parameters deleted:	CALLINFO (information instead in CALLDATA)

Valid values have changed:
- in structure SPH_PDCALL for the field ´FINISHED´ from ´ ´ and ´X´ to ´N´ and ´Y´.
- in structure SPH_PDCHIS for the field ´FINISHED´ from ´ ´ and ´X´ to ´N´ and ´Y´.

12.2 5.00A → 5.01ASP

New parameter TYPE in function SPS_GETQUEUES
New parameter LINES_WITH_IP in function SPS_GET_LINES_PER_SERVER

13 Miscellaneous

The TAPI functions supported by SAPphone are listed in the SAPphone server documentation.