



HiPath TAPI170 DeveloperGuide

We are now **Unify**.

This document references our previous company name; all other content is still valid and correct. For further information, please visit unify.com



HiPath TAPI 170 V2

Description

TAPI application developer's guide

Version 1.4

Table of Contents

1	<i>Survey of TAPI SP Functions</i>	7
1.1	Basic Services Functions	7
1.2	Supplementary Line Services Functions	8
1.3	Extended Line Services Functions	9
1.4	Supplementary Phone Services Functions	10
1.5	Extended Phone Services Functions	11
1.6	Provider Functions	12
2	<i>Survey of TAPI SP Messages</i>	13
2.1	Line Device Messages	13
2.2	Phone Device Messages	15
3	<i>Survey of TAPI SP Structures</i>	16
3.1	LINEADDRESSCAPS	16
3.2	LINEADDRESSCAPS for Extended Services Line	20
3.3	LINEADDRESSSTATUS	22
3.4	LINEADDRESSSTATUS for Extended Services Line	23
3.5	LINECALLINFO	24
3.6	LINECALLPARAMS	27
3.7	LINECALLSTATUS	29
3.8	LINECALLTREATMENTENTRY	31
3.9	LINEDEVCAPS	32
3.10	LINEDEVCAPS for Extended Services Line	34
3.11	LINEDEVSTATUS	36
3.12	LINEDEVSTATUS for Extended Services Line	37
3.13	LINEDIALPARAMS	38
3.14	LINEEXTENSIONID	38
3.15	LINEFORWARD	39
3.16	LINEFORWARDLIST	39
3.17	LINEGENERATETONE	40

3.18	LINEMEDIACONTROLCALLSTATE	40
3.19	LINEMEDIACONTROLDIGIT	40
3.20	LINEMEDIACONTROLMEDIA	40
3.21	LINEMEDIACONTROLTONE	40
3.22	LINEMONITORTONE	40
3.23	LINETERMCAPS	40
3.24	PHONEBUTTONINFO	41
3.25	PHONECAPS	42
3.26	PHONECAPS for Extended Services Phone	46
3.27	PHONEEXTENSIONID	48
3.28	PHONESTATUS	49
3.29	PHONESTATUS for Extended Services Phone	51
3.30	VARSTRING	52
3.31	TUISPICREATEDIALOGINSTANCEPARAMS	52
4	Key-Features	53
5	TAPI Extended Services	57
5.1	TSP-internal Extended Services	57
5.1.1	Siemens-specific	57
5.1.1.1	Device-Specific Line Functions	57
5.1.1.1.1	lineDevSpecific	57
5.1.1.1.2	lineDevSpecificFeature	58
5.1.1.2	Device-Specific Data Structures and Constants	59
5.1.1.2.1	SIEMENS_LINE_DEV_SPECIFIC	59
6	TAPI ACD Proxy Request Handler	60
6.1	How-To handle HiPath 3000/5000 UCD agents / groups with HiPath TAPI 170 ACD-ProxyRequestHandler (ACD-PRH, APH)	60
6.1.1	Prerequisites	60
6.1.2	General	60
6.1.3	Login an HiPath UCD agent	60
6.1.4	Change a HiPath UCD agent state	60
6.1.5	Handling of HiPath UCD groups	60
6.2	ACD-ProxyRequestHandler Functions for TAPI 2.0 / 2.1	62
6.2.1	lineAgentSpecific	62
6.2.2	lineGetAgentActivityList	62
6.2.3	lineGetAgentCaps	63
6.2.4	lineGetAgentGroupList	64
6.2.5	lineGetAgentStatus	65
6.2.6	LineSetAgentActivity	66
6.2.7	lineSetAgentGroup	66
6.2.8	lineSetAgentState	66

6.3	ACD-ProxyRequestHandler Functions for TAPI 2.2 and 3.x	68
7	<i>DevSpecific structures</i>	69
7.1	struct CAPS_DEVSPECIFIC	69
7.2	struct STATUS_DEVSPECIFIC	69
8	<i>Neutral Lines</i>	70
8.1	Procedures for Neutral Line Devices	70
8.1.1	Initial Situation	70
8.1.2	Dynamical Assignment of a Callnumber	70
8.1.3	Callnumber has been assigned	71
8.1.4	Next Dynamical Assignment of a Callnumber	72
8.1.5	Next Callnumber has been assigned	72
8.1.6	Cancelling of a Callnummer Assignment	73
8.1.7	Callnummer Assignment was cancelled	74
8.2	Handling of Neutral Lines	75
8.2.1	Free neutral Line	75
8.2.1.1	Device Classes	75
8.2.1.2	Capabilities	75
8.2.1.3	Assignment	75
8.2.2	Assigned neutral Lines	77
8.2.2.1	Device Classes	77
8.2.2.2	Cancelling of an assignment	77
8.2.2.3	Capabilities	77
9	<i>References</i>	78
10	<i>NewsGroups</i>	80
11	<i>Internet-Links</i>	81

List of Tables

Table 1: Basic Services Functions	7
Table 2: Supplementary Line Services Functions	8
Table 3: Extended Line Services Functions	9
Table 4: Supplementary Phone Services Functions	10
Table 5: Extended Phone Services Functions	11
Table 6: Provider Functions	12
Table 7: Line Device Messages	14
Table 8: Phone Device Messages	15
Table 9: LINEADDRESSCAPS	19
Table 10: LINEADDRESSCAPS for Extended Services Line	21
Table 11: LINEADDRESSSTATUS	22
Table 12: LINEADDRESSSTATUS for Extended Services Line	23
Table 13: LINECALLINFO	26
Table 14: LINECALLPARAMS	28
Table 15: LINECALLSTATUS	30
Table 16: LINECALLTREATMENTENTRY	31
Table 17: LINEDEVCAPS	33
Table 18: LINEDEVCAPS for Extended Services Line	35
Table 19: LINEDEVSTATUS	36
Table 20: LINEDEVSTATUS for Extended Services Line	37
Table 21: LINEDIALPARAMS	38
Table 22: LINEEXTENSIONID	38
Table 23: LINEFORWARD	39
Table 24: LINEFORWARDLIST	39
Table 25: PHONEBUTTONINFO	41
Table 26: PHONECAPS	45
Table 27: PHONECAPS for Extended Services Phone	47
Table 28: PHONEEXTENSIONID	48
Table 29: PHONESTATUS	50
Table 30: PHONESTATUS for Extended Services Phone	51
Table 31: VARSTRING	52
Table 32: Key-Features	56
Table 33: ACD-PRH Functions for TAPI 2.0 / 2.1	62
Table 34: HostControlledFlag and OverflowFlag	67
Table 35: AgentStates for Agent- and UCD-Devices	67
Table 36: Getting / Setting AgentStates	67
Table 37: ACD-PRH Functions for TAPI 2.2 and TAPI 3.x	68
Table 38: CAPS_DEVSPECIFIC	69

Table 39: STATUS_DEVSPECIFIC _____	69
Table 40: STATUS_DEVSPECIFIC.bInitialQueryCompleted _____	69
Table 41: Device Classes for a free neutral Line _____	75
Table 42: Capabilities for a free neutral Line _____	75
Table 43: Assignment for a free neutral Line _____	76
Table 44: Device Classes for an assigned neutral Line _____	77
Table 45: Cancelling of an assignment for an assigned neutral Line _____	77
Table 46: Capabilities for an assigned neutral Line _____	77

List of Figures

Figure 1: Initial Situation (neutral Lines) _____	70
Figure 2: Dynamical Assignment of a Callnumber (neutral Lines) _____	71
Figure 3: Callnumber has been assigned (neutral Lines) _____	71
Figure 4: Next Dynamical Assignment of a Callnumber (neutral Lines) _____	72
Figure 5: Next Callnumber has been assigned (neutral Lines) _____	73
Figure 6: Cancelling of a Callnummer Assignment (neutral Lines) _____	73
Figure 7: Callnummer Assignment was cancelled (neutral Lines) _____	74
Figure 8: DEVICECONFIGEDIT for lineGetDevConfig / lineSetDevConfig _____	76

1 Survey of TAPI SP Functions

1.1 Basic Services Functions

TSPI functions	Supported	Comment
TSPI_lineAnswer	yes	
TSPI_lineClose	yes	
TSPI_lineCloseCall	yes	
TSPI_lineConditionalMediaDetection	yes	
TSPI_lineConfigDialog	yes	
TSPI_lineConfigDialogEdit	no	obsolete, TAPI 1.4, 16 bit
TSPI_lineDial	yes	
TSPI_lineDrop	yes	
TSPI_lineDropNoOwner	no	obsolete, TAPI 1.4, 16 bit
TSPI_lineDropOnClose	no	obsolete, TAPI 1.4, 16 bit
TSPI_lineGetAddressCaps	yes	
TSPI_lineGetAddressID	yes	
TSPI_lineGetAddressStatus	yes	
TSPI_lineGetCallAddressID	yes	
TSPI_lineGetCallInfo	yes	
TSPI_lineGetCallStatus	yes	
TSPI_lineGetDevCaps	yes	
TSPI_lineGetDevConfig	yes	neutral-line
TSPI_lineGetIcon	yes	
TSPI_lineGetID	yes	
TSPI_lineGetLineDevStatus	yes	
TSPI_lineGetNumAddressIDs	yes	
TSPI_lineMakeCall	yes	
TSPI_lineNegotiateTSPIVersion	yes	TAPI 2.0 / 2.1 / 2.2
TSPI_lineOpen	yes	
TSPI_lineSetAppSpecific	yes	
TSPI_lineSetDefaultMediaDetection	yes	
TSPI_lineSetStatusMessages	yes	
TUISPI_lineConfigDialog	yes	
TUISPI_lineConfigDialogEdit	no	

Table 1: Basic Services Functions

1.2 Supplementary Line Services Functions

TSPI function	Supported	Comment
TSPI_lineAccept	no	
TSPI_lineAddToConference	yes	
TSPI_lineBlindTransfer	yes	
TSPI_lineCompleteCall	no	
TSPI_lineCompleteTransfer	yes	
TSPI_lineForward	yes	
TSPI_lineGatherDigits	no	
TSPI_lineGenerateDigits	yes	
TSPI_lineGenerateTone	no	
TSPI_lineHold	yes ¹	
TSPI_lineMonitorDigits	no	
TSPI_lineMonitorMedia	no	
TSPI_lineMonitorsTones	no	
TSPI_linePark	no	
TSPI_linePickup	yes	
TSPI_linePrepareAddToConference	yes	
TSPI_lineRedirect	yes	
TSPI_lineReleaseUserUserInfo	no	
TSPI_lineRemoveFromConference	yes	
TSPI_lineSendUserUserInfo	no	
TSPI_lineSetCallData	yes	
TSPI_lineSetCallParams	no	
TSPI_lineSetCallQualityOfService	no	
TSPI_lineSetCallTreatment	yes	
TSPI_lineSetCurrentLocation	no	obsolete, TAPI 1.4, 16 bit
TSPI_lineSetDevConfig	yes	neutral-line
TSPI_lineSetLineDevStatus	yes	
TSPI_lineSetMediaControl	no	
TSPI_lineSetMediaMode	no	
TSPI_lineSetTerminal	no	
TSPI_lineSetupConference	yes	
TSPI_lineSetupTransfer	yes	
TSPI_lineSwapHold	yes	
TSPI_lineUncompleteCall	no	
TSPI_lineUnhold	yes	
TSPI_lineUnpark	no	

Table 2: Supplementary Line Services Functions

¹ Hicom 150 E Office and HiPath 3000 support only hold / unhold of external call.

1.3 Extended Line Services Functions

TSPI function	Supported	Comment
TSPI_lineDevSpecific	yes	
TSPI_lineDevSpecificFeature	yes	
TSPI_lineGetExtensionID	yes	
TSPI_lineNegotiateExtVersion	yes	0x00018000 / 0x00020000 / 0x00030000 / 0x00040000
TSPI_lineSelectExtVersion	yes	0x00018000 / 0x00020000 / 0x00030000 / 0x00040000

Table 3: Extended Line Services Functions

1.4 Supplementary Phone Services Functions

TSPI function	Supported	Comment
TSPI_phoneClose	yes	
TSPI_phoneConfigDialog	yes	
TSPI_phoneGetButtonInfo	yes	
TSPI_phoneGetData	yes	
TSPI_phoneGetDevCaps	yes	
TSPI_phoneGetDisplay	yes	
TSPI_phoneGetGain	yes	
TSPI_phoneGetHookSwitch	yes	
TSPI_phoneGetIcon	yes	
TSPI_phoneGetID	yes	
TSPI_phoneGetLamp	no	
TSPI_phoneGetRing	no	
TSPI_phoneGetStatus	yes	
TSPI_phoneGetVolume	yes	
TSPI_phoneNegotiateTSPIVersion	yes	TSPI 2.0 / 2.1 / 2.2
TSPI_phoneOpen	yes	
TSPI_phoneSetButtonInfo	no	
TSPI_phoneSetData	yes	
TSPI_phoneSetDisplay	yes	
TSPI_phoneSetGain	yes	
TSPI_phoneSetHookSwitch	yes	
TSPI_phoneSetLamp	yes	
TSPI_phoneSetRing	no	
TSPI_phoneSetStatusMessages	yes	
TSPI_phoneSetVolume	yes	
TUISPI_phoneConfigDialog	yes	

Table 4: Supplementary Phone Services Functions

1.5 Extended Phone Services Functions

TSPI function	Supported	Comment
TSPI_phoneDevSpecific	yes	
TSPI_phoneGetExtensionID	yes	
TSPI_phoneNegotiateExtVersion	yes	0x00020000 / 0x00030000 / 0x00040000
TSPI_phoneSelectExtVersion	yes	0x00020000 / 0x00030000 / 0x00040000

Table 5: Extended Phone Services Functions

1.6 Provider Functions

TSPI function	Supported	Comment
TSPI_providerConfig	yes	
TSPI_providerCreateLineDevice	no	
TSPI_providerCreatePhoneDevice	no	
TSPI_providerEnumDevices	yes	
TSPI_providerFreeDialogInstance	no	
TSPI_providerGenericDialogData	yes	
TSPI_providerInit	yes	TSPI 2.0 / 2.1 / 2.2
TSPI_providerInstall	yes	
TSPI_providerRemove	yes	
TSPI_providerShutdown	yes	
TSPI_providerUIIdentify	yes	
TUISPI_providerConfig	yes	
TUISPI_providerGenericDialog	no	
TUISPI_providerGenericDialogData	no	
TUISPI_providerInstall	yes	
TUISPI_providerRemove	yes	

Table 6: Provider Functions

2 Survey of TAPI SP Messages

2.1 Line Device Messages

TSPI Message	Supported	Supported dwParam1 Values
LINE_ADDRESSSTATE	yes	LINEADDRESSSTATE_OTHER ² LINEADDRESSSTATE_INUSEONE LINEADDRESSSTATE_NUMCALLS LINEADDRESSSTATE_FORWARD LINEADDRESSSTATE_CAPSCHANGE ³
LINE_CALLDEVSPECIFIC	no	
LINE_CALLDEVSPECIFICFEATURE	no	
LINE_CALLINFO	yes	LINECALLINFOSTATE_APPSPECIFIC LINECALLINFOSTATE_CALLERID LINECALLINFOSTATE_CALLEDID LINECALLINFOSTATE_CONNECTEDID, LINECALLINFOSTATE_REDIRECTINGID LINECALLINFOSTATE_REDIRECTIONID LINECALLINFOSTATE_CALLID LINECALLINFOSTATE_RELATEDCALLID LINECALLINFOSTATE_ORIGIN LINECALLINFOSTATE_REASON LINECALLINFOSTATE_CHARGINGINFO LINECALLINFOSTATE_OTHER ⁴ LINECALLINFOSTATE_CALLDATA LINECALLINFOSTATE_TRUNK LINECALLINFOSTATE_TREATMENT
LINE_CALLSTATE	yes	LINECALLSTATE_IDLE LINECALLSTATE_DIALTONE LINECALLSTATE_DIALING LINECALLSTATE_PROCEEDING LINECALLSTATE_SPECIALINFO LINECALLSTATE_BUSY ⁵ LINECALLSTATE_RINGBACK LINECALLSTATE_CONFERENCED LINECALLSTATE_ONHOLDPENDCONF LINECALLSTATE_ONHOLDPENDTRANSFER LINECALLSTATE_DISCONNECTED LINECALLSTATE_CONNECTED LINECALLSTATE_OFFERING LINECALLSTATE_ONHOLD LINECALLSTATE_UNKNOWN

² dwAddressFeatures have changed

³ dwAddressCapFlags have changed due to detection of line device being a trunk or UCD group

⁴ dwCallParamFlags or DialParams have changed

LINE_CLOSE	yes	n.a.
LINE_CREATE	yes	
LINE_CREATEDIALOGINSTANCE	no	
LINE_DEVSPECIFIC	yes	
LINE_DEVSPECIFICFEATURE	yes	
LINE_GATHERDIGITS	no	
LINE_GENERATE	yes	LINEGENERATETERM_DONE LINEGENERATETERM_CANCEL
LINE_LINEDEVSTATE	yes	LINEDEVSTATE_OTHER ⁶ LINEDEVSTATE_CONNECTED LINEDEVSTATE_DISCONNECTED LINEDEVSTATE_INSERVICE LINEDEVSTATE_OUTOFSERVICE LINEDEVSTATE_MSGWAITON LINEDEVSTATE_MSGWAITOFF LINEDEVSTATE_RINGING LINEDEVSTATE_REINIT LINEDEVSTATE_NUMCALLS
LINE_MONITORDIGITS	no	
LINE_MONITORMEDIA	no	
LINE_MONITORTONE	no	
LINE_NEWCALL	yes	n.a.
LINE_REMOVE	yes	
LINE_SENDDIALOGINSTANCEDATA	no	

Table 7. Line Device Messages

⁵ If „Busy“ is signalled for an internal call in Hicom 150 E Office, there will be usually an automatic delayed (5 seconds) camp on. So the call state of this call handle transitions from busy to ringback.

⁶ dwLineFeatures have changed

2.2 Phone Device Messages

TSPI Message	Supported	Supported dwParam1 Values
PHONE_BUTTON	yes	
PHONE_CLOSE	yes	n.a.
PHONE_CREATE	yes	
PHONE_DEVSPECIFIC	yes	
PHONE_REMOVE	yes	
PHONE_STATE	yes	PHONESTATE_CAPSCHANGE PHONESTATE_CONNECTED PHONESTATE_DISCONNECTED PHONESTATE_DISPLAY PHONESTATE_LAMP PHONESTATE_HANDSETGAIN PHONESTATE_HANDSETHOOKSWITCH PHONESTATE_HANDSETVOLUME PHONESTATE_SPEAKERGAIN PHONESTATE_SPEAKERHOOKSWITCH PHONESTATE_SPEAKERVOLUME PHONESTATE_HEADSETGAIN PHONESTATE_HEADSETHOOKSWITCH PHONESTATE_HEADSETVOLUME PHONESTATE_REINIT

Table 8: Phone Device Messages

3 Survey of TAPI SP Structures

3.1 LINEADDRESSCAPS

Field	Supported Values
dwTotalSize	yes (filled in by TAPI)
dwNeededSize	yes
dwUsedSize	yes
dwLineDeviceID	yes
dwAddressSize	yes
dwAddressOffset	
dwDevSpecificSize	CAPS_DEVSPECIFIC ⁷
dwDevSpecificOffset	
dwAddressSharing	LINEADDRESSSHARING_PRIVATE
dwAddressStates	LINEADDRESSSTATE_OTHER LINEADDRESSSTATE_INUSEONE LINEADDRESSSTATE_NUMCALLS LINEADDRESSSTATE_FORWARD LINEADDRESSSTATE_CAPSCHANGE ⁸
dwCallInfoStates	LINECALLINFOSTATE_APPSPECIFIC LINECALLINFOSTATE_CALLERID LINECALLINFOSTATE_CALLEDID LINECALLINFOSTATE_CONNECTEDID, LINECALLINFOSTATE_REDIRECTINGID LINECALLINFOSTATE_REDIRECTIONID LINECALLINFOSTATE_CALLID LINECALLINFOSTATE_RELATEDCALLID LINECALLINFOSTATE_ORIGIN LINECALLINFOSTATE_REASON LINECALLINFOSTATE_CHARGINGINFO LINECALLINFOSTATE_OTHER LINECALLINFOSTATE_CALLDATA LINECALLINFOSTATE_TRUNK LINECALLINFOSTATE_TREATMENT
dwCallerIDFlags	LINECALLPARTYID_UNAVAIL LINECALLPARTYID_UNKNOWN LINECALLPARTYID_ADDRESS
dwCalledIDFlags	LINECALLPARTYID_UNAVAIL LINECALLPARTYID_UNKNOWN LINECALLPARTYID_ADDRESS

⁷ see 7.1 struct CAPS_DEVSPECIFIC for details

⁸ dwAddressCapFlags have changed due to detection of line device being a trunk or UCD group

dwConnectedIDFlags	LINECALLPARTYID_UNAVAIL LINECALLPARTYID_UNKNOWN LINECALLPARTYID_ADDRESS
dwRediricetionIDFlags	LINECALLPARTYID_UNAVAIL LINECALLPARTYID_UNKNOWN LINECALLPARTYID_ADDRESS
dwRedirectingIDFlags	LINECALLPARTYID_UNAVAIL LINECALLPARTYID_UNKNOWN LINECALLPARTYID_ADDRESS
dwCallStates	LINECALLSTATE_IDLE LINECALLSTATE_DIALTONE LINECALLSTATE_DIALING LINECALLSTATE_PROCEEDING LINECALLSTATE_SPECIALINFO LINECALLSTATE_BUSY ⁹ LINECALLSTATE_RINGBACK LINECALLSTATE_CONFERENCED LINECALLSTATE_ONHOLDPENDCONF LINECALLSTATE_ONHOLDPENDTRANSFER LINECALLSTATE_DISCONNECTED LINECALLSTATE_CONNECTED LINECALLSTATE_OFFERING LINECALLSTATE_ONHOLD LINECALLSTATE_UNKNOWN
dwDialToneModes	LINEDIALTONMODE_UNAVAIL
dwBusyModes	LINEBUSYMODE_UNAVAIL
dwSpecialInfo	LINESPECIALINFO_UNAVAIL
dwDisconnectModes	LINEDISCONNECTMODE_NORMAL LINEDISCONNECTMODE_FORWARDED LINEDISCONNECTMODE_PICKUP
dwMaxNumActiveCalls	150
DwMaxNumOnHoldCalls	150
dwMaxNumOnHoldPendingCalls	1
dwMaxNumConference	5
dwMaxNumTransConf	5
dwAddressCapFlags	LINEADDRCAPFLAGS_DIALED LINEADDRCAPFLAGS_ORIGOFFHOOK LINEADDRCAPFLAGS_AUTORECONNECT LINEADDRCAPFLAGS_FWDSTATUSVALID LINEADDRCAPFLAGS_PICKUPCALLWAIT LINEADDRCAPFLAGS_QUEUE ¹⁰ LINEADDRCAPFLAGS_ROUTEPOINT LINEADDRCAPFLAGS_NOINTERNALCALLS ¹¹

⁹ If „Busy“ is signalled for an internal call in Hicom 150 E Office, there will be usually an automatic delayed (5 seconds) camp on. So the call state of this call handle transitions from busy to ringback.

¹⁰ only if line device is an UCD group; note: this flag will only be available after lineOpen

¹¹ only if line device is a trunk; note: this flag will only be available after lineOpen

dwCallFeatures	LINECALLFEATURE_ANSWER LINECALLFEATURE_DIAL LINECALLFEATURE_DROP LINECALLFEATURE_GENERATEDIGITS LINECALLFEATURE_REDIRECT LINECALLFEATURE_BLINDTRANSFER LINECALLFEATURE_SETUPTRANSFER LINECALLFEATURE_SWAPHOLD LINECALLFEATURE_COMPLETETRANSF LINECALLFEATURE_SETUPCONF LINECALLFEATURE_PREPAREADDCONF LINECALLFEATURE_ADDTOCONF LINECALLFEATURE_REMOVEFROMCONF LINECALLFEATURE_HOLD LINECALLFEATURE_UNHOLD LINECALLFEATURE_SETCALLDATA LINECALLFEATURE_SETTREATMENT
dwRemoveFromConfCaps	LINEREMOVEFROMCONF_ANY
dwRemoveFromConfState	LINECALLSTATE_IDLE
dwTransferModes	LINETRANSFERMODE_TRANSFER LINETRANSFERMODE_CONFERENCE
dwParkModes	n.s.
dwForwardModes	LINEFORWARDMODE_UNCONDEXTERNAL LINEFORWARDMODE_UNCONDINTERNAL LINEFORWARDMODE_UNCOND
dwMaxForwardEntries	1
dwMaxSpecificEntries	0
dwMinFwdNumRings	0
dwMaxFwdNumRings	0
dwMaxCallCompletions	n.s.
dwCallCompletionConds	n.s.
dwCallCompletionModes	n.s.
dwCompletionMsgTextEntrySize	n.s.
dwCompletionMsgTextSize	n.s.
deCompletionMsgTextOffset	
dwAddressFeatures	LINEADDRFEATURE_MAKECALL LINEADDRFEATURE_PICKUP LINEADDRFEATURE_PICKUPWAITING LINEADDRFEATURE_PICKUPDIRECT LINEADDRFEATURE_PICKUPGROUP LINEADDRFEATURE_FORWARD LINEADDRFEATURE_FORWARDFWD LINEADDRFEATURE_FORWARDDND
dwPredictiveAutoTransferStates	n.s.
dwNumCallTreatments	1 ¹² .. (1+n ¹³)
dwCallTreatmentListSize	yes ¹⁴
dwCallTreatmentListOffset	

¹² Internal music on hold (MOH) is always available

¹³ service provider-specific values for external MOH / recorded announcements

dwDeviceClassesSize	tapi/line
dwDeviceClassesOffset	tapi/phone tapi/line/neutral/free tapi/phone/neutral/free tapi/line/neutral/assigned tapi/phone/neutral/assigned
dwMaxCallDataSize	65536
dwCallFeatures2	LINECALLFEATURE2_TRANSFERCONF LINECALLFEATURE2_TRANSFERNORM LINECALLFEATURE2_ONESTEPTRANSFER
dwMaxNoAnswerTimeout	n.s.
dwConnectedModes	LINECONNECTEDMODE_ACTIVEHELD LINECONNECTEDMODE_CONFIRMED
dwOfferingModes	n.s.
dwAvailableMediaModes	LINEMEDIAMODE_INTERACTIVEVOICE

Table 9: LINEADDRESSCAPS

¹⁴ LINECALLTREATMENT_MUSIC, „ [int. MOH](#) “ for internal music on hold; 0x80000000 .., for external MOH / recorded announcements: „xxxxxx“ (xxxxxx is the extension number of device) if indices are not used; „[Rcaldx##](#)“ ([## is the index of the announcement](#)) if [indices are used](#)

3.2 LINEADDRESSCAPS for Extended Services Line

Field	Supported Values
dwTotalSize	yes (filled in by TAPI)
dwNeededSize	yes
dwUsedSize	yes
dwLineDeviceID	yes
dwAddressSize	yes
dwAddressOffset	
dwDevSpecificSize	CAPS_DEVSPECIFIC ¹⁵
dwDevSpecificOffset	
dwAddressSharing	n.s.
dwAddressStates	LINEADDRESSSTATE_INUSEZERO
dwCallInfoStates	n.s.
dwCallerIDFlags	n.s.
dwCalledIDFlags	n.s.
dwConnectedIDFlags	n.s.
dwRediricetionIDFlags	n.s.
dwRedirectingIDFlags	n.s.
dwCallStates	n.s.
dwDialToneModes	n.s.
dwBusyModes	n.s.
dwSpecialInfo	n.s.
dwDisconnectModes	n.s.
dwMaxNumActiveCalls	n.s.
dwMaxNumOnHoldCalls	n.s.
dwMaxNumOnHoldPendingCalls	n.s.
dwMaxNumConference	n.s.
dwMaxNumTransConf	n.s.
dwAddressCapFlags	LINEADDRCAPFLAGS_NOEXTERNALCALLS LINEADDRCAPFLAGS_NOINTERNALCALLS
dwCallFeatures	n.s.
dwRemoveFromConfCaps	n.s.
dwRemoveFromConfState	n.s.
dwTransferModes	n.s.
dwParkModes	n.s.
dwForwardModes	n.s.
dwMaxForwardEntries	n.s.
dwMaxSpecificEntries	n.s.
dwMinFwdNumRings	n.s.
dwMaxFwdNumRings	n.s.
dwMaxCallCompletions	n.s.
dwCallCompletionConds	n.s.

¹⁵ see 7.1 struct CAPS_DEVSPECIFIC for details

dwCallCompletionModes	n.s.
dwCompletionMsgTextEntrySize	n.s.
dwCompletionMsgTextSize	n.s.
deCompletionMsgTextOffset	
dwAddressFeatures	n.s.
dwPredictiveAutoTransferStates	n.s.
dwNumCallTreatments	n.s.
dwCallTreatmentListSize	n.s.
dwCallTreatmentListOffset	
dwDeviceClassesSize	tapi/line
dwDeviceClassesOffset	tapi/phone
dwMaxCallDataSize	n.s.
dwCallFeatures2	n.s.
dwMaxNoAnswerTimeout	n.s.
dwConnectedModes	n.s.
dwOfferingModes	n.s.
dwAvailableMediaModes	n.s.

Table 10: LINEADDRESSCAPS for Extended Services Line

3.3 LINEADDRESSSTATUS

Field	Supported values
dwTotalSize	yes (filled in by TAPI)
dwNeededSize	yes
dwUsedSize	yes
dwNumInUse	1
dwNumActiveCalls	0 .. 150
dwNumOnHoldCalls	0 .. 150
dwNumOnHoldPendCalls	0 .. 1
dwAddressFeatures	LINEADDRFEATURE_MAKECALL LINEADDRFEATURE_PICKUP LINEADDRFEATURE_PICKUPWAITING LINEADDRFEATURE_PICKUPGROUP LINEADDRFEATURE_PICKUPDIRECT LINEADDRFEATURE_FORWARD LINEADDRFEATURE_FORWARDFWD LINEADDRFEATURE_FORWARDDND
dwNumRingsNoAnswer	n.s.
dwForwardNumEntries	0 .. 1
dwForwardSize	yes
dwForwardOffset	
dwTerminalModeSize	n.s.
dwTerminalModeOffset	
dwDevSpecificSize	STATUS_DEVSPECIFIC ¹⁶
dwDevSpecificOffset	

Table 11: LINEADDRESSSTATUS

¹⁶ see 0 Table 38: CAPS_DEVSPECIFIC
struct STATUS_DEVSPECIFIC for details

3.4 LINEADDRESSSTATUS for Extended Services Line

Field	Supported values
dwTotalSize	yes (filled in by TAPI)
dwNeededSize	yes
dwUsedSize	yes
dwNumInUse	0
dwNumActiveCalls	n.s.
dwNumOnHoldCalls	n.s.
dwNumOnHoldPendCalls	n.s.
dwAddressFeatures	n.s.
dwNumRingsNoAnswer	n.s.
dwForwardNumEntries	n.s.
dwForwardSize	n.s.
dwForwardOffset	
dwTerminalModeSize	n.s.
dwTerminalModeOffset	
dwDevSpecificSize	n.s.
dwDevSpecificOffset	

Table 12: LINEADDRESSSTATUS for Extended Services Line

3.5 LINECALLINFO

Field	Supported values
dwTotalSize	yes (filled in by TAPI)
dwNeededSize	yes
dwUsedSize	yes
hLine	yes (filled in by TAPI)
dwLineDeviceID	yes
dwAddressID	0
dwBearerMode	LINEBEAREMODE_VOICE
dwRate	0
dwMediaMode	LINEMEDIAMODE_INTERACTIVEVOICE
dwAppSpecific	yes
dwCallID	yes
dwReleatedCallID	yes
dwCallParamFlags	LINECALLPARAMFLAGS_IDLE LINECALLPARAMFLAGS_ORIGOFFHOOK LINECALLPARAMFLAGS_BLOCKID LINECALLPARAMFLAGS_ONESTEPTRANSFER
dwCallStates	LINECALLSTATE_IDLE LINECALLSTATE_DIALTONE LINECALLSTATE_DIALING LINECALLSTATE_PROCEEDING LINECALLSTATE_SPECIALINFO LINECALLSTATE_BUSY ¹⁷ LINECALLSTATE_RINGBACK LINECALLSTATE_CONFERENCED LINECALLSTATE_ONHOLDPENDCONF LINECALLSTATE_ONHOLDPENDTRANSFER LINECALLSTATE_DISCONNECTED LINECALLSTATE_CONNECTED LINECALLSTATE_OFFERING LINECALLSTATE_ONHOLD LINECALLSTATE_UNKNOWN
dwMonitorDigitModes	yes (filled in by TAPI)
dwMonitorMediaModes	yes (filled in by TAPI)
DialParams	yes
dwOrigin	LINECALLORIGIN_UNKNOWN LINECALLORIGIN_EXTERNAL LINECALLORIGIN_INTERNAL LINECALLORIGIN_INBOUND LINECALLORIGIN_OUTBOUND LINECALLORIGIN_CONFERENCE

¹⁷ If „Busy“ is signalled for an internal call in Hicom 150 E Office, there will be usually an automatic delayed (5 seconds) camp on. So the call state of this call handle transitions from busy to ringback.

dwReason	LINECALLREASON_DIRECT LINECALLREASON_REDIRECT LINECALLREASON_PICKUP LINECALLREASON_TRANSFER LINECALLREASON_REMINDER LINECALLREASON_CALLCOMPLETION LINECALLREASON_UNPARK LINECALLREASON_FWDUNCOND LINECALLREASON_CAMPEDON
dwCompletionID	n.s.
dwNumOwners	yes (filled in by TAPI)
dwNumMonitors	yes (filled in by TAPI)
dwCountryCode	n.s.
dwTrunk	yes ¹⁸
dwCallerIDFlags	LINECALLPARTYID_UNAVAIL LINECALLPARTYID_UNKNOWN LINECALLPARTYID_ADDRESS
dwCallerIDSize	yes
dwCallerIDOffset	
dwCallerIDNameSize	yes
dwCallerIDNameOffset	
dwCalledIDFlags	LINECALLPARTYID_UNAVAIL LINECALLPARTYID_UNKNOWN LINECALLPARTYID_ADDRESS
dwCalledIDSize	yes
dwCalledIDOffset	
dwCalledIDNameSize	n.s.
dwCalledIDNameOffset	
dwConnectedIDFlags	LINECALLPARTYID_UNAVAIL LINECALLPARTYID_UNKNOWN LINECALLPARTYID_ADDRESS
dwConnectedIDSize	yes
dwConnectedIDOffset	
dwConnectedIDNameSize	n.s.
dwConnectedIDNameOffset	
dwRedirectionIDFlags	LINECALLPARTYID_UNAVAIL LINECALLPARTYID_UNKNOWN LINECALLPARTYID_ADDRESS
dwRedirectionIDSize	yes
dwRedirectionIDOffset	
dwRedirectionIDNameSize	n.s.
dwRedirectionNameOffset	
dwRedirectingIDFlags	LINECALLPARTYID_UNAVAIL LINECALLPARTYID_UNKNOWN LINECALLPARTYID_ADDRESS
dwRedirectingIDSize	yes

¹⁸ dwTrunk may be converted to char szTrunkCode[6+1] by means of
_ultoa(dwTrunk, szTrunkCode, 10);

dwRedirectingIDOffset	
dwRedirectingIDNameSize	n.s.
dwRedirectingIDNameOffset	
dwAppNameSize	yes (filled in by TAPI)
dwAppNameOffset	
dwDisplayableAddressSize	yes (filled in by TAPI)
dwDisplayableAddressOffset	
dwCalledPartySize	yes (filled in by TAPI)
dwCalledPartyOffset	
dwCommentSize	yes (filled in by TAPI)
dwCommentOffset	
dwDisplaySize	n.s.
dwDisplayOffset	
dwUserUserInfoSize	n.s.
dwUserUserInfoOffset	
dwHighLevelCompSize	n.s.
dwHighLevelCompOffset	
dwLowLevelCompSize	n.s.
dwLowLevelCompOffset	
dwChargingInfoSize	yes
dwChargingInfoOffset	
dwTerminalModesSize	n.s.
dwTerminalModesOffset	
dwSpecificSize	n.s.
dwSpecificOffset	
dwCallTreatment	0 LINECALLTREATMENT_MUSIC ¹⁹ other ²⁰
dwCallDataSize	yes
dwCallDataOffset	
dwSendingFlowspecSize	n.s.
dwSendingFlowspecOffset	
dwReceivingFlowspecSize	n.s.
dwReceivingFlowspecOffset	

Table 13: LINECALLINFO

¹⁹ Internal music on hold (MOH)

²⁰ service provider-specific values for external MOH / recorded announcements: 0x80000000 ..

3.6 LINECALLPARAMS

Field	Supported values
dwTotalSize	yes (filled in by TAPI)
dwBearerMode	LINEBEARERMODE_VOICE
dwMinRate	0
dwMaxRate	0
dwMediaMode	LINEMEDIAMODE_INTERACTIVEVOICE
dwCallParamsFlags	LINECALLPARAMFLAGS_IDLE LINECALLPARAMFLAGS_ORIGOFFHOOK LINECALLPARAMFLAGS_BLOCKID LINECALLPARAMFLAGS_ONESTEPTRANSFER
dwAddressMode	LINEADDRESSMODE_ADDRESSID
dwAddressID	0
DialParams	yes
dwOrigAddressSize	n.s.
dwOrigAddressOffset	
dwDisplayableAddressSize	yes (filled in by TAPI)
dwDisplayableAddressOffset	
dwCalledPartySize	yes (filled in by TAPI)
dwCalledPartyOffset	
dwCommentSize	yes (filled in by TAPI)
dwCommentOffset	
dwUserUserInfoSize	n.s.
dwUserUserInfoOffset	
dwHighLevelCompSize	n.s.
dwHighLevelCompOffset	
dwLowLevelCompSize	n.s.
dwLowLevelCompOffset	
dwSpecificSize	n.s.
dwSpecificOffset	
dwPredictiveAutoTransferStates	n.s.
dwTargetAddressSize	n.s.
dwTargetAddressOffset	
dwSendingFlowspecSize	n.s.
dwSendingFlowspecOffset	
dwReceivingFlowspecSize	n.s.
dwReceivingFlowspecOffset	
dwDeviceClassSize	n.s.
dwDeviceClassOffset	
dwDeviceConfigSize	n.s.
dwDeviceConfigOffset	
dwCallDataSize	n.s.
dwCallDataOffset	
dwNoAnswerTimeout	n.s.
dwCallingPartyIDSize	n.s.

dwCallingPartyIDOffset	
dwAddressType	n.s.

Table 14: LINECALLPARAMS

3.7 LINECALLSTATUS

Field	Supported values
dwTotalSize	yes (filled in by TAPI)
dwNeededSize	yes
dwUsedSize	yes
dwCallState	LINECALLSTATE_IDLE LINECALLSTATE_DIALTONE LINECALLSTATE_DIALING LINECALLSTATE_PROCEEDING LINECALLSTATE_SPECIALINFO LINECALLSTATE_BUSY ²¹ LINECALLSTATE_RINGBACK LINECALLSTATE_CONFERENCED LINECALLSTATE_ONHOLDPENDCONF LINECALLSTATE_ONHOLDPENDTRANSFER LINECALLSTATE_DISCONNECTED LINECALLSTATE_CONNECTED LINECALLSTATE_OFFERING LINECALLSTATE_ONHOLD LINECALLSTATE_UNKNOWN
dwCallStateMode	LINEDIALTONMODE_UNAVAIL LINEBUSYMODE_UNAVAIL LINESPECIALINFO_UNAVAIL LINEDISCONNECTMODE_NORMAL LINEDISCONNECTMODE_FORWARDED LINEDISCONNECTMODE_PICKUP LINECONNECTEDMODE_ACTIVEHELD LINECONNECTEDMODE_CONFIRMED
dwCallPrivilege	yes (filled in by TAPI)

²¹ If „Busy“ is signalled for an internal call in Hicom 150 E Office, there will be usually an automatic delayed (5 seconds) camp on. So the call state of this call handle transitions from busy to ringback.

dwCallFeatures	LINECALLFEATURE_ANSWER LINECALLFEATURE_DIAL LINECALLFEATURE_DROP LINECALLFEATURE_GENERATEDIGITS LINECALLFEATURE_REDIRECT LINECALLFEATURE_BLINDTRANSFER LINECALLFEATURE_SETUPTRANSFER LINECALLFEATURE_SWAPHOLD LINECALLFEATURE_COMPLETETRANSF LINECALLFEATURE_SETUPCONF LINECALLFEATURE_PREPAREADDCONF LINECALLFEATURE_ADDTOCONF LINECALLFEATURE_REMOVEFROMCONF LINECALLFEATURE_HOLD LINECALLFEATURE_UNHOLD LINECALLFEATURE_SETCALLDATA LINECALLFEATURE_SETTREATMENT
dwDevSpecificSize	n.s.
dwDevSpecificOffset	
dwCallFeatures2	LINECALLFEATURE2_TRANSFERNORM LINECALLFEATURE2_TRANSFERCONF LINECALLFEATURE2_ONESTEPTRANSFER
tStateEntryTime	yes

Table 15: LINECALLSTATUS

3.8 LINECALLTREATMENTENTRY

Field	Supported values
dwCallTreatmentID	LINECALLTREATMENT_MUSIC ²² other ²³
dwCallTreatmentNameSize	yes ²⁴
dwCallTreatmentNameOffset	

Table 16: LINECALLTREATMENTENTRY

²² Internal music on hold (MOH)

²³ service provider-specific values for external MOH / recorded announcements: 0x80000000 ..

²⁴ „int. MOH“ for internal music on hold;
for external MOH / recorded announcements: „xxxxxx“ (xxxxxx is the extension number of device) if indices are not used; „Rcaldx##“ (## is the index of the announcement) if indices are used

3.9 LINEDEVCAPS

Field	Supported values
dwTotalSize	yes (filled in by TAPI)
dwNeededSize	yes
dwUsedSize	yes
dwProviderInfoSize	yes
dwProviderInfoOffset	
dwSwitchInfoSize	yes
dwSwitchInfoOffset	
dwPermanentLineID	yes
dwLineNameSize	yes
dwLineNameOffset	
dwStringFormat	STRINGFORMAT_ASCII
dwAddressModes	LINEADDRESSMODE_ADDRESSID
dwNumAddresses	1
dwBearerModes	LINEBEARERMODE_VOICE
dwMaxRate	0
dwMediaModes	LINEMEDIAMODE_INTERACTIVEVOICE
dwGenerateToneModes	n.s.
dwGenerateToneMaxNumFreq	n.s.
dwGenerateDigitModes	LINEDIGITMODE_DTMF
dwMonitorToneMaxNumFreq	n.s.
dwMonitorToneMaxNumEntries	n.s.
dwMonitorDigitModes	n.s.
dwGatherDigitsMinTimeout	n.s.
dwGatherDigitsMaxTimeout	n.s.
dwMedCtlDigitMaxListSize	n.s.
dwMedCtlMediaMaxListSize	n.s.
dwMedCtlToneMaxListSize	n.s.
dwMedCtlCallStateMaxListSize	n.s.
dwDevCapFlags	0
dwMaxNumActiveCalls	1
dwAnswerMode	LINEANSWERMODE_HOLD
dwRingModes	1
dwLineStates	LINEDEVSTATE_OTHER LINEDEVSTATE_CONNECTED LINEDEVSTATE_DISCONNECTED LINEDEVSTATE_INSERVICE LINEDEVSTATE_OUTOFSERVICE LINEDEVSTATE_MSGWAITON LINEDEVSTATE_MSGWAITOFF LINEDEVSTATE_RINGING LINEDEVSTATE_REINIT LINEDEVSTATE_NUMCALLS
dwUIIAcceptSize	n.s.

dwUIAnswerSize	n.s.
dwUIMakeCallSize	n.s.
dwUIDropSize	n.s.
dwUISendUserUserInfoSize	n.s.
dwUICallInfoSize	n.s.
MinDialParams	yes
MaxDialParams	yes
DefaultDialParams	yes
dwNumTerminals	0
dwTerminalCapSize	n.s.
dwTerminalCapsOffset	
dwTerminalTextEntrySize	n.s.
dwTerminalTextSize	n.s.
dwTerminalTextOffset	
dwDevSpecificSize	CAPS_DEVSPECIFIC ²⁵
dwDevSpecificOffset	
dwLineFeatures	LINEFEATURE_MAKECALL LINEFEATURE_FORWARD LINEFEATURE_FORWARDFWD LINEFEATURE_FORWARDDND LINEFEATURE_SETDEVSTATUS
dwSettableDevStatus	LINEDEVSTATUSFLAGS_MSGWAIT
dwDeviceClassesSize	tapi/line
dwDeviceClassesOffset	tapi/phone tapi/line/neutral/free tapi/phone/neutral/free tapi/line/neutral/assigned tapi/phone/neutral/assigned

Table 17: LINEDEVCAPS

²⁵ see 7.1 struct CAPS_DEVSPECIFIC for details

3.10 LINEDEVCAPS for Extended Services Line

Field	Supported values
dwTotalSize	yes (filled in by TAPI)
dwNeededSize	yes
dwUsedSize	yes
dwProviderInfoSize	yes
dwProviderInfoOffset	
dwSwitchInfoSize	yes
dwSwitchInfoOffset	
dwPermanentLineID	yes
dwLineNameSize	yes
dwLineNameOffset	
dwStringFormat	STRINGFORMAT_ASCII
dwAddressModes	LINEADDRESSMODE_ADDRESSID
dwNumAddresses	n.s.
dwBearerModes	n.s.
dwMaxRate	n.s.
dwMediaModes	n.s.
dwGenerateToneModes	n.s.
dwGenerateToneMaxNumFreq	n.s.
dwGenerateDigitModes	LINEDIGITMODE_DTMF
dwMonitorToneMaxNumFreq	n.s.
dwMonitorToneMaxNumEntries	n.s.
dwMonitorDigitModes	n.s.
dwGatherDigitsMinTimeout	n.s.
dwGatherDigitsMaxTimeout	n.s.
dwMedCtlDigitMaxListSize	n.s.
dwMedCtlMediaMaxListSize	n.s.
dwMedCtlToneMaxListSize	n.s.
dwMedCtlCallStateMaxListSize	n.s.
dwDevCapFlags	0
dwMaxNumActiveCalls	n.s.
dwAnswerMode	n.s.
dwRingModes	n.s.
dwLineStates	LINEDEVSTATE_CONNECTED LINEDEVSTATE_DISCONNECTED LINEDEVSTATE_INSERVICE LINEDEVSTATE_OUTOFSERVICE LINEDEVSTATE_REINIT
dwUIAcceptSize	n.s.
dwUIAnswerSize	n.s.
dwUIMakeCallSize	n.s.
dwUIDropSize	n.s.
dwUISendUserUserInfoSize	n.s.
dwUICallInfoSize	n.s.

MinDialParams	n.s.
MaxDialParams	n.s.
DefaultDialParams	n.s.
dwNumTerminals	n.s.
dwTerminalCapSize	n.s.
dwTerminalCapsOffset	
dwTerminalTextEntrySize	n.s.
dwTerminalTextSize	n.s.
dwTerminalTextOffset	
dwDevSpecificSize	CAPS_DEVSPECIFIC ²⁶
dwDevSpecificOffset	
dwLineFeatures	LINEFEATURE_DEVSPECIFIC LINEFEATURE_DEVSPECIFICFEAT
dwSettableDevStatus	n.s.
dwDeviceClassesSize	tapi/line
dwDeviceClassesOffset	tapi/phone

Table 18: LINEDEVCAPS for Extended Services Line

²⁶ see 7.1 struct CAPS_DEVSPECIFIC for details

3.11 LINEDEVSTATUS

Field	Supported values
dwTotalSize	yes (filled in by TAPI)
dwNeededSize	yes
dwUsedSize	yes
dwNumOpens	yes (filled in by TAPI)
dwOpenMediaModes	yes (filled in by TAPI)
dwNumActiveCalls	0 .. 150
dwNumOnHoldCalls	0 .. 150
dwNumOnHoldPendCalls	0 .. 1
dwLineFeatures	LINEFEATURE_MAKECALL LINEFEATURE_FORWARD LINEFEATURE_FORWARDFWD LINEFEATURE_FORWARDDND LINEFEATURE_SETDEVSTATUS
dwNumCallCompletions	n.s.
dwRingMode	n.s.
dwSignalLevel	n.s.
dwBatteryLevel	n.s.
dwRoamMode	LINEROAMMODE_UNAVAIL
dwDevStatusFlags	LINEDEVSTATUSFLAGS_CONNECTED LINEDEVSTATUSFLAGS_INSERTSERVICE LINEDEVSTATUSFLAGS_MSGWAIT
dwTerminalModesSize	n.s.
dwTerminalModesOffset	
dwDevSpecificSize	STATUS_DEVSPECIFIC ²⁷
dwDevSpecificOffset	
dwAvailableMediaModes	LINEMEDIAMODE_INTERACTIVEVOICE
dwAppInfoSize	yes (filled in by TAPI)
dwAppInfoOffset	

Table 19: LINEDEVSTATUS

²⁷ see 0 Table 38: CAPS_DEVSPECIFIC

struct STATUS_DEVSPECIFIC for details

3.12 LINEDEVSTATUS for Extended Services Line

Field	Supported values
dwTotalSize	yes (filled in by TAPI)
dwNeededSize	yes
dwUsedSize	yes
dwNumOpens	yes (filled in by TAPI)
dwOpenMediaModes	yes (filled in by TAPI)
dwNumActiveCalls	n.s.
dwNumOnHoldCalls	n.s.
dwNumOnHoldPendCalls	n.s.
dwLineFeatures	LINEFEATURE_DEVSPECIFIC LINEFEATURE_DEVSPECIFICFEAT
dwNumCallCompletions	n.s.
dwRingMode	n.s.
dwSignalLevel	n.s.
dwBatteryLevel	n.s.
dwRoamMode	n.s.
dwDevStatusFlags	LINEDEVSTATUSFLAGS_CONNECTED LINEDEVSTATUSFLAGS_INSERTSERVICE
dwTerminalModesSize	n.s.
dwTerminalModesOffset	
dwDevSpecificSize	n.s.
dwDevSpecificOffset	
dwAvailableMediaModes	n.s.
dwAppInfoSize	yes (filled in by TAPI)
dwAppInfoOffset	

Table 20: LINEDEVSTATUS for Extended Services Line

3.13 LINEDIALPARAMS

Field	Supported values
dwDialPause	0, 2000, 255000 ²⁸
dwDialSpeed	50, 95, 255
dwDigitDuration	50, 95, 255
dwWaitForDialtone	1000, 60000, 255000

Table 21: LINEDIALPARAMS

3.14 LINEEXTENSIONID

Field	Supported values
dwExtensionID0	0x3AB39627
dwExtensionID1	0x9FA111D3
dwExtensionID2	0xBD250050
dwExtensionID3	0x046786EB

Table 22: LINEEXTENSIONID

²⁸ min., default, max.

3.15 LINEFORWARD

Field	Supported values
dwForwardMode	LINEFORWARDMODE_UNCOND LINEFORWARDMODE_UNCONDINTERNAL LINEFORWARDMODE_UNCONDEXTERNAL
dwCallerAddressSize	n.s.
dwCallerAddressOffset	
dwDestCountryCode	n.s.
dwDestAddressSize	yes
dwDestAddressOffset	

Table 23: LINEFORWARD

3.16 LINEFORWARDLIST

Field	Supported values
dwTotalSize	yes
dwNumEntries	1
ForwardList	yes

Table 24: LINEFORWARDLIST

3.17 LINEGENERATETONE

Not supported.

3.18 LINEMEDIACONTROLCALLSTATE

Not supported.

3.19 LINEMEDIACONTROLDIGIT

Not supported.

3.20 LINEMEDIACONTROLMEDIA

Not supported.

3.21 LINEMEDIACONTROLTONE

Not supported.

3.22 LINEMONITORTONE

Not supported.

3.23 LINETERMCAPS

Not supported.

3.24 PHONEBUTTONINFO

Field	Supported values
dwTotalSize	yes (filled in by TAPI)
dwNeededSize	yes
dwUsedSize	yes
dwButtonMode	PHONEBUTTONMODE_KEYPAD PHONEBUTTONMODE_LOCAL ²⁹ PHONEBUTTONMODE_DISPLAY ³⁰ PHONEBUTTONMODE_FEATURE PHONEBUTTONMODE_CALL PHONEBUTTONMODE_DUMMY
dwButtonFunction	PHONEBUTTONFUNCTION_UNKNOWN PHONEBUTTONFUNCTION_VOLUMEUP PHONEBUTTONFUNCTION_VOLUMEDOWN 0xC0000000 + n ³¹
dwButtonTextSize	yes ³²
dwButtonTextOffset	
dwDevSpecificSize	n.s.
dwDevSpecificOffset	
dwButtonState	PHONEBUTTONSTATE_UNAVAIL

Table 25: PHONEBUTTONINFO

²⁹ buttons for volume control

³⁰ buttons for menu control

³¹ n = buttonFunction; for corresponding feature refer to section 4 Key-Features

³² for PHONEBUTTONMODE_KEYPAD "0", "1", .. "9", "*" or "#" will be returned, for PHONEBUTTONMODE_LOCAL "+" or "-" will be returned, for PHONEBUTTONMODE_DISPLAY "/", "<" or ">" will be returned, for PHONEBUTTONMODE_FEATURE only "" will be returned

3.25 PHONECAPS

Field	Supported values
dwTotalSize	yes (filled in by TAPI)
dwNeededSize	yes
dwUsedSize	yes
dwProviderInfoSize	yes
dwProviderInfoOffset	
dwPhoneInfoSize	n.s.
dwPhoneInfoOffset	
dwPermanentPhoneID	yes
dwPhoneNameSize	yes
dwPhoneNameOffset	
dwStringFormat	STRINGFORMAT_ASCII
dwPhoneStates	PHONESTATE_CAPSCHANGE PHONESTATE_CONNECTED PHONESTATE_DISCONNECTED PHONESTATE_DISPLAY PHONESTATE_LAMP PHONESTATE_HANDSETGAIN PHONESTATE_HANDSETHOOKSWITCH PHONESTATE_HANDSETVOLUME PHONESTATE_SPEAKERGAIN PHONESTATE_SPEAKERHOOKSWITCH PHONESTATE_SPEAKERVOLUME PHONESTATE_HEADSETGAIN PHONESTATE_HEADSETHOOKSWITCH PHONESTATE_HEADSETVOLUME PHONESTATE_REINIT
dwHookSwitchDevs	PHONEHOOKSWITCHDEV_HANDSET PHONEHOOKSWITCHDEV_HEADSET PHONEHOOKSWITCHDEV_SPEAKER
dwHandsetHookSwitchModes	PHONEHOOKSWITCHMODE_ONHOOK PHONEHOOKSWITCHMODE_SPEAKER PHONEHOOKSWITCHMODE_MICSPEAKER PHONEHOOKSWITCHMODE_UNKNOWN
dwSpeakerHookSwitchModes	PHONEHOOKSWITCHMODE_ONHOOK PHONEHOOKSWITCHMODE_SPEAKER PHONEHOOKSWITCHMODE_MIC PHONEHOOKSWITCHMODE_MICSPEAKER PHONEHOOKSWITCHMODE_UNKNOWN
dwHeadsetHookSwitchModes	PHONEHOOKSWITCHMODE_ONHOOK PHONEHOOKSWITCHMODE_SPEAKER PHONEHOOKSWITCHMODE_MICSPEAKER PHONEHOOKSWITCHMODE_UNKNOWN

dwVolumeFlags	PHONEHOOKSWITCHDEV_HANDSET PHONEHOOKSWITCHDEV_SPEAKER PHONEHOOKSWITCHDEV_HEADSET
dwGainFlags	PHONEHOOKSWITCHDEV_HANDSET PHONEHOOKSWITCHDEV_SPEAKER PHONEHOOKSWITCHDEV_HEADSET
dwDisplayNumRows	1
dwDisplayNumColumns	24
dwNumRingModes	n.s.
dwNumButtonLamps	94 ³³
dwButtonModesSize	376
dwButtonModesOffset	PHONEBUTTONMODE_KEYPAD PHONEBUTTONMODE_LOCAL ³⁴ PHONEBUTTONMODE_DISPLAY ³⁵ PHONEBUTTONMODE_FEATURE PHONEBUTTONMODE_CALL PHONEBUTTONMODE_DUMMY
dwButtonFunctionsSize	376
dwButtonFunctionsOffset	PHONEBUTTONFUNCTION_UNKNOWN PHONEBUTTONFUNCTION_VOLUMEUP PHONEBUTTONFUNCTION_VOLUMEDOWN 0xC0000000 + n ³⁶
dwLampModesSize	376
dwLampModesOffset	PHONELAMPMODE_DUMMY PHONELAMPMODE_FLUTTER ³⁷ PHONELAMPMODE_OFF PHONELAMPMODE_STEADY PHONELAMPMODE_WINK ³⁸
dwNumSetData	8
dwSetDataSize	0x00000100 ³⁹
dwSetDataOffset	
dwNumGetData	8
dwGetDataSize	0x00000100
dwGetDataOffset	
dwDevSpecificSize	CAPS_DEVSPECIFIC ⁴⁰

³³ The keypad buttons '0', through '9', '*', and '#' are assigned the identifiers 0 through 12 (s. /1/), ID13 refers to „+“ key, ID14 to „-“, ID15 to „<“, ID16 to „>“, ID17 to „OK“, IDs18..29 refer to the 12 feature keys of the base unit and the IDs 30..93 refer to the feature keys on the 4 possible extension units (these feature keys are the only ones with associated lamps, all others have PHONELAMPMODE_DUMMY set)

³⁴ buttons for volume control

³⁵ buttons for menu control

³⁶ n = buttonFunction; for corresponding feature refer to section 4 Key-Features

³⁷ LED cadence 5 (250 / 250 ms)

³⁸ LED cadence 6 (750 / 750 ms)

³⁹ per DataID

dwDevSpecificOffset	
dwDeviceClassesSize	tapi/line
dwDeviceClassesOffset	tapi/phone tapi/line/neutral/free tapi/phone/neutral/free tapi/line/neutral/assigned tapi/phone/neutral/assigned
dwPhoneFeatures	PHONEFEATURE_GETBUTTONINFO PHONEFEATURE_GETDATA PHONEFEATURE_GETDISPLAY PHONEFEATURE_GETGAINHANDSET PHONEFEATURE_GETGAINSPEAKER PHONEFEATURE_GETGAINHEADSET PHONEFEATURE_GETHOOKSWITCHHANDSET PHONEFEATURE_GETHOOKSWITCHSPEAKER PHONEFEATURE_GETHOOKSWITCHHEADSET PHONEFEATURE_GETVOLUMEHANDSET PHONEFEATURE_GETVOLUMESPEAKER PHONEFEATURE_GETVOLUMEHEADSET PHONEFEATURE_SETDATA PHONEFEATURE_SETDISPLAY PHONEFEATURE_SETLAMP PHONEFEATURE_SETGAINHANDSET PHONEFEATURE_SETGAINSPEAKER PHONEFEATURE_SETGAINHEADSET PHONEFEATURE_SETHOOKSWITCHHANDSET PHONEFEATURE_SETHOOKSWITCHSPEAKER PHONEFEATURE_SETHOOKSWITCHHEADSET PHONEFEATURE_SETVOLUMEHANDSET PHONEFEATURE_SETVOLUMESPEAKER PHONEFEATURE_SETVOLUMEHEADSET
dwSettableHandsetHookSwitchModes	PHONEHOOKSWITCHMODE_SPEAKER PHONEHOOKSWITCHMODE_MICSPEAKER
dwSettableSpeakerHookSwitchModes	PHONEHOOKSWITCHMODE_SPEAKER PHONEHOOKSWITCHMODE_MIC PHONEHOOKSWITCHMODE_MICSPEAKER PHONEHOOKSWITCHMODE_ONHOOK
dwSettableHeadsetHookSwitchModes	PHONEHOOKSWITCHMODE_SPEAKER PHONEHOOKSWITCHMODE_MICSPEAKER
dwMonitoredHandsetHookSwitchModes	PHONEHOOKSWITCHMODE_SPEAKER PHONEHOOKSWITCHMODE_MICSPEAKER PHONEHOOKSWITCHMODE_ONHOOK PHONEHOOKSWITCHMODE_UNKNOWN
dwMonitoredSpeakerHookSwitchModes	PHONEHOOKSWITCHMODE_SPEAKER PHONEHOOKSWITCHMODE_MIC PHONEHOOKSWITCHMODE_MICSPEAKER PHONEHOOKSWITCHMODE_ONHOOK PHONEHOOKSWITCHMODE_UNKNOWN

⁴⁰ see 7.1 struct CAPS_DEVSPECIFIC for details

dwMonitoredHeadsetHookSwitchModes	PHONEHOOKSWITCHMODE_SPEAKER PHONEHOOKSWITCHMODE_MICSPEAKER PHONEHOOKSWITCHMODE_ONHOOK PHONEHOOKSWITCHMODE_UNKNOWN
--	--

Table 26: PHONECAPS

3.26 PHONECAPS for Extended Services Phone

Field	Supported values
dwTotalSize	yes (filled in by TAPI)
dwNeededSize	yes
dwUsedSize	yes
dwProviderInfoSize	yes
dwProviderInfoOffset	
dwPhoneInfoSize	n.s.
dwPhoneInfoOffset	
dwPermanentPhoneID	yes
dwPhoneNameSize	yes
dwPhoneNameOffset	
dwStringFormat	STRINGFORMAT_ASCII
dwPhoneStates	PHONESTATE_CONNECTED PHONESTATE_DISCONNECTED PHONESTATE_REINIT
dwHookSwitchDevs	n.s.
dwHandsetHookSwitchModes	n.s.
dwSpeakerHookSwitchModes	n.s.
dwHeadsetHookSwitchModes	n.s.
dwVolumeFlags	n.s.
dwGainFlags	n.s.
dwDisplayNumRows	n.s.
dwDisplayNumColumns	n.s.
dwNumRingModes	n.s.
dwNumButtonLamps	n.s.
dwButtonModesSize	n.s.
dwButtonModesOffset	n.s.
dwButtonFunctionsSize	n.s.
dwButtonFunctionsOffset	n.s.
dwLampModesSize	n.s.
dwLampModesOffset	n.s.
dwNumSetData	n.s.
dwSetDataSize	n.s.
dwSetDataOffset	
dwNumGetData	n.s.
dwGetDataSize	n.s.
dwGetDataOffset	
dwDevSpecificSize	CAPS_DEVSPECIFIC ⁴¹
dwDevSpecificOffset	
dwDeviceClassesSize	tapi/line
dwDeviceClassesOffset	tapi/phone

⁴¹ see 7.1 struct CAPS_DEVSPECIFIC for details

dwPhoneFeatures	n.s.
dwSettableHandsetHookSwitchModes	n.s.
dwSettableSpeakerHookSwitchModes	n.s.
dwSettableHeadsetHookSwitchModes	n.s.
dwMonitoredHandsetHookSwitchModes	n.s.
dwMonitoredSpeakerHookSwitchModes	n.s.
dwMonitoredHeadsetHookSwitchModes	n.s.

Table 27: PHONECAPS for Extended Services Phone

3.27 PHONEEXTENSIONID

Field	Supported values
dwExtensionID0	0x3AB39627
dwExtensionID1	0x9FA111D3
dwExtensionID2	0xBD250050
dwExtensionID3	0x046786EB

Table 28: PHONEEXTENSIONID

3.28 PHONESTATUS

Field	Supported values
dwTotalSize	yes (filled in by TAPI)
dwNeededSize	yes
dwUsedSize	yes
dwStatusFlags	PHONESTATUSFLAGS_CONNECTED
dwNumOwners	yes (filled in by TAPI)
dwNumMonitors	yes (filled in by TAPI)
dwRingMode	n.s.
dwRingVolume	n.s.
dwHandsetHookSwitchMode	PHONEHOOKSWITCHMODE_ONHOOK PHONEHOOKSWITCHMODE_SPEAKER PHONEHOOKSWITCHMODE_MICSPEAKER PHONEHOOKSWITCHMODE_UNKNOWN
dwHandsetVolume	yes ⁴²
dwHandsetGain	0x00000000 0x00007FFF ⁴³
dwSpeakerHookSwitchMode	PHONEHOOKSWITCHMODE_ONHOOK PHONEHOOKSWITCHMODE_SPEAKER PHONEHOOKSWITCHMODE_MIC PHONEHOOKSWITCHMODE_MICSPEAKER PHONEHOOKSWITCHMODE_UNKNOWN
dwSpeakerVolume	yes
dwSpeakerGain	yes
dwHeadsetHookSwitchMode	PHONEHOOKSWITCHMODE_ONHOOK PHONEHOOKSWITCHMODE_SPEAKER PHONEHOOKSWITCHMODE_MICSPEAKER PHONEHOOKSWITCHMODE_UNKNOWN
dwHeadsetVolume	yes
dwHeadsetGain	yes
dwDisplaySize	1 * 24 + 1 ⁴⁴
dwDisplayOffset	yes
dwLampModesSize	PHONELAMPMODE_DUMMY
dwLampModesOffset	PHONELAMPMODE_FLUTTER PHONELAMPMODE_OFF PHONELAMPMODE_STEADY PHONELAMPMODE_WINK
dwOwnerNameSize	yes (filled in by TAPI)
dwOwnerNameOffset	
dwDevSpecificSize	n.s.
dwDevSpecificOffset	

⁴² range: 0x00000000 .. 0x0000FFFF

⁴³ default mid-range value (TSP can't determine the device' gain)

⁴⁴ number of ASCII characters in bytes: rows * cols + string terminator

<p>dwPhoneFeatures</p>	<p>PHONEFEATURE_GETBUTTONINFO PHONEFEATURE_GETDATA PHONEFEATURE_GETDISPLAY PHONEFEATURE_GETGAINHANDSET PHONEFEATURE_GETGAINSPEAKER PHONEFEATURE_GETGAINHEADSET PHONEFEATURE_GETHOOKSWITCHHANDSET PHONEFEATURE_GETHOOKSWITCHSPEAKER PHONEFEATURE_GETHOOKSWITCHHEADSET PHONEFEATURE_GETVOLUMEHANDSET PHONEFEATURE_GETVOLUMESPEAKER PHONEFEATURE_GETVOLUMEHEADSET PHONEFEATURE_SETDATA PHONEFEATURE_SETDISPLAY PHONEFEATURE_SETLAMP PHONEFEATURE_SETGAINHANDSET PHONEFEATURE_SETGAINSPEAKER PHONEFEATURE_SETGAINHEADSET PHONEFEATURE_SETHOOKSWITCHHANDSET PHONEFEATURE_SETHOOKSWITCHSPEAKER PHONEFEATURE_SETHOOKSWITCHHEADSET PHONEFEATURE_SETVOLUMEHANDSET PHONEFEATURE_SETVOLUMESPEAKER PHONEFEATURE_SETVOLUMEHEADSET</p>
-------------------------------	---

Table 29: PHONESTATUS

3.29 PHONESTATUS for Extended Services Phone

Field	Supported values
dwTotalSize	yes (filled in by TAPI)
dwNeededSize	yes
dwUsedSize	yes
dwStatusFlags	PHONESTATUSFLAGS_CONNECTED
dwNumOwners	yes (filled in by TAPI)
dwNumMonitors	yes (filled in by TAPI)
dwRingMode	n.s.
dwRingVolume	n.s.
dwHandsetHookSwitchMode	n.s.
dwHandsetVolume	n.s.
dwHandsetGain	n.s.
dwSpeakerHookSwitchMode	n.s.
dwSpeakerVolume	n.s.
dwSpeakerGain	n.s.
dwHeadsetHookSwitchMode	n.s.
dwHeadsetVolume	n.s.
dwHeadsetGain	n.s.
dwDisplaySize	n.s.
dwDisplayOffset	n.s.
dwLampModesSize	n.s.
dwLampModesOffset	n.s.
dwOwnerNameSize	yes (filled in by TAPI)
dwOwnerNameOffset	n.s.
dwDevSpecificSize	n.s.
dwDevSpecificOffset	n.s.
dwPhoneFeatures	n.s.

Table 30: PHONESTATUS for Extended Services Phone

3.30 VARSTRING

Field	Supported values
dwTotalSize	yes
dwNeededSize	yes
dwUsedSize	yes
dwStringFormat	STRINGFORMAT_ASCII STRINGFORMAT_BINARY
dwStringSize	yes
dwStringOffset	

Table 31: VARSTRING

3.31 TUISPICREATEDIALOGINSTANCEPARAMS

Not supported.

4 Key-Features

buttonFunction	Feature
000	Free
001	Internal station
002	S - service
003	Repeat last number
004	Open listening/hands free talking
005	DTMF sending
006	Door opener (relay)
007	Toggle
008	Call forward, no destination
009	Conference
010	Speaker mute
011	Code lock
012	Release
013	Night service
014	Override
015	Waiting calls
016	Call key
017	Voice calling
018	Account code
019	Speed dial individual
020	Speed dial central/individual
021	Int. calls diverted, int. dest.
022	Int. calls diverted, ext. dest.
023	Ext. calls diverted, int. dest.
024	Ext. calls diverted, ext. dest.
025	All calls diverted, int. dest.
026	All calls diverted, ext. dest.
027	Do Not Disturb
028	Direct Station Select
029	Message leave
030	Send textmessage
031	Send textmessage, Text 0
032	Send textmessage, Text 1
033	Send textmessage, Text 2
034	Send textmessage, Text 3
035	Send textmessage, Text 4
036	Send textmessage, Text 5

037	Send textmessage, Text 6
038	Send textmessage, Text 7
039	Send textmessage, Text 8
040	Send textmessage, Text 9
041	Callback
042	Data station
043	Seizure
044	Actors
045	Trunk
046	Call pickup
047	Park
048	Flash to analogue trunk line
049	Retrieve call charge data
050	Call provider (part.)
051	Call provider (compl.)
052	Call pickup group
053	CFW for ext line
054	Answer camp on
055	Direct answer
056	Babyphone
057	Incomming call assignment
058	Call forward unconditional
059	Call forward on no reply
060	Call forward busy
061	Message waiting
062	Entry to speed dial central
063	Incomming call log
064	Telephonebook
065	Start/Stop linehunting
066	Catch call
067	Associated service
068	Associated dialing
069	Silence
070	Silent call waiting
071	Calling line identification restriction
072	Fax received
073	Dooropen via DTMF
074	Common Hold
075	Retrieve hold call
076	MUSAP-key
077	Seizure key
078	Release of trunk

079	Code-lock central
080	Message waiting receive
081	Telephone Data Service
082	Alarm clock
083	Charge call
084	Paging
085	PMEETME
086	Send textmessage, ext.,
087	Send textmessage, ext., Text 0
088	Send textmessage, ext., Text 1
089	Send textmessage, ext., Text 2
090	Send textmessage, ext., Text 3
091	Send textmessage, ext., Text 4
092	Send textmessage, ext., Text 5
093	Send textmessage, ext., Text 6
094	Send textmessage, ext., Text 7
095	Send textmessage, ext., Text 8
096	Send textmessage, ext., Text 9
097	Call transfer
098	ACD Log On
099	ACD available
100	ACD wrap-up
101	ACD show-calls
102	ACD Night service internal
103	ACD Night service external
104	Internal callback
105	Telephone Data Service
106	Common call key
107	Call Waiting Terminating
108	Clearkey
109	Boss Ringing
110	Boss 2 Ringing
111	Call transfer of undialed trunks
112	PAULMSN
113	MSN-related CFW, busy
114	MSN- related CFW, no answer
115	MSN- related CFW, all
116	Deactivate Trace
117	Shift key
118	Mulap
119	Mulap Ringassignment
120	Mulap call forward, no destination

121	Mulap int. calls diverted, int. dest.
122	Mulap int. calls diverted, ext. dest.
123	Mulap ext. calls diverted, int. dest.
124	Mulap ext. calls diverted, ext. dest.
125	Mulap all calls diverted, int. dest.
126	Mulap all calls diverted, ext. dest.
127	Mobile Pin
140	Carrier Select Override

Table 32: Key-Features

5 TAPI Extended Services

5.1 TSP-internal Extended Services

5.1.1 Siemens-specific

5.1.1.1 Device-Specific Line Functions

5.1.1.1.1 lineDevSpecific

- The **dwAddressID** parameter is not used. Only 0 is accepted.
- The **hCall** parameter is not used. Only NULL is accepted.
- The parameter **lpParams** should point to a memory area whose contents are defined according to the data structure **SIEMENS_LINE_DEV_SPECIFIC**. (Refer to “SIEMENS_LINE_DEV_SPECIFIC” in chapter 5.1.1.2.1).
- The **dwSize** parameter must be the size of the **SIEMENS_LINE_DEV_SPECIFIC** structure.

5.1.1.1.1.1 SIEMENSLINE_DIAL_DIGITS

- This function allows the application to send a **dialable** string to the PBX, independent of the phone’s state or the state of any line addresses or calls on the phone.
 - The **dwStructType** field in the data structure must be the value **SIEMENS_LINE_DIAL_DIGITS**.
 - The **szDialableString[]** field in the data structure is a null-terminated ASCII string of dialable digits (* # and 0 through 9). The maximum number of digits is 100.
 - Invalid characters will be ignored.

5.1.1.1.1.2 DEVSPECIFIC_STRUCT_FIRSTDESTINATION

- This function allows the application to set a dialable string which is used to identify the first destination the call has been.
 - The **dwStructType** field in the data structure must be the value **DEVSPECIFIC_STRUCT_FIRSTDESTINATION**.
 - The **FirstDestination.szCallNumber** field in the data structure is a null-terminated ASCII string of dialable digits (* # and 0 through 9). The maximum number of digits is 100.

5.1.1.1.1.3 DEVSPECIFIC_STRUCT_ACDAUTOFLAG

- This function is reserved for internal use and is not released.

5.1.1.1.2 lineDevSpecificFeature

- The TSP invokes a device-specific feature by essentially pressing a button for the feature. The success of the feature invocation is highly context-specific.
- The service provider does not provide an explicit indication of feature success or failure. The application must know how to interpret success or failure of the feature based on asynchronous information from the phone, for example: LED indications, or state changes.
- The application sends the physical button number on the phone (pointed to by lpParams) and leaves dwFeature empty (PHONEBUTTONFUNCTION_UNKNOWN). The TSP invokes the button..
- The **dwSize** parameter must be set correctly when **lpParams** is not NULL.

5.1.1.2 Device-Specific Data Structures and Constants

The following chapter describes the device-specific data structures and constants created by Siemens for the telephony service provider.

5.1.1.2.1 SIEMENS_LINE_DEV_SPECIFIC

The **SIEMENS_LINE_DEV_SPECIFIC** structure is passed to the **lineDevSpecific** function via parameter **lpParams**. It identifies the extended feature that is being requested and any other information that is required for that extended feature.

```
typedef struct _PARAM_DEVSPECIFIC
{
    DWORD dwStructType;
    union
    {
        struct
        {
            char szCallNumber [TAPI_MAX_DEST_ADDRESS_SIZE];
        } FirstDestination;
        struct
        {
            DWORD dwSetACDAutoFlag;
        } ACDAutoFlag;
        BYTE szDialableString [SIEMENS_MAX_DIALABLE_STRING_BYTES + 1];
    };
} SIEMENS_LINE_DEV_SPECIFIC, FAR *LPSIEMENS_LINE_DEV_SPECIFIC;
```

Constants

dwStructType

This field identifies the extended line device feature that is being requested. The values supported for this field are:

SIEMENS_LINE_DIAL_DIGITS	0x00000001
// Send a dialable string to the phone.	
DEVSPECIFIC_STRUCT_ACDAUTOFLAG	0xFCFCFCFC
// for internal use	
DEVSPECIFIC_STRUCT_FIRSTDESTINATION	0xFDFDFDFD
// set first destination	

6 TAPI ACD Proxy Request Handler

6.1 How-To handle HiPath 3000/5000 UCD agents / groups with HiPath TAPI 170 ACD-ProxyRequestHandler (ACD-PRH, APH)

6.1.1 Prerequisites

- ACD3Service must be running (recommended service start mode 'automatic')
- check ACD checkboxes in HT170 TSP-UI lines tab for HiPath UCD agents and HiPath UCD groups

6.1.2 General

- *lineOpen* TAPI line devices for HiPath UCD agents and HiPath UCD groups
- call *lineGetAgentCaps* and *lineGetAgentStatus* as you need
- *lineGetAgentGroupList* returns a list with all HiPath UCD groups configured as TAPI line devices
- *lineGetAgentActivity* returns a list with all currently unused HiPath UCD AgentIDs (100 through 249) as dwActivityID

6.1.3 Login an HiPath UCD agent

- 1) *lineSetAgentActivity* with dwActivityID = <a currently unused HiPath UCD AgentID>
 - 2) *lineSetAgentState* with dwAgentState = LINEAGENTSTATE_READY
- This sequence is mandatory.

6.1.4 Change a HiPath UCD agent state

- *lineSetAgentState* with dwAgentState = LINEAGENTSTATE_READY, LINEAGENTSTATE_NOTREADY, LINEAGENTSTATE_WORKINGAFTERCALL, or LINEAGENTSTATE_LOGGEDOFF

The latter state change logs the HiPath UCD agent off and requires a re-login for further usage.

6.1.5 Handling of HiPath UCD groups

- only *lineSetAgentState* with LINEAGENTSTATE_READY, LINEAGENTSTATE_NOTREADY, or LINEAGENTSTATE_LOGGEDOFF is supported for HiPath UCD groups

- `LINEAGENTSTATE_LOGGEDOFF` is the default and means call distribution is done automatically by the PBX
- `LINEAGENTSTATE_READY` means the call distribution is host controlled, i.e. the TAPI app need to *lineRedirect* the incoming calls to the agents
- `LINEAGENTSTATE_NOTREADY` means overload-rejection, i.e. all incoming UCD calls get busy signaling

6.2 ACD-ProxyRequestHandler Functions for TAPI 2.0 / 2.1

Function	Supported
<i>lineAgentSpecific</i>	no
<i>lineGetAgentActivityList</i>	yes
<i>lineGetAgentCaps</i>	yes
<i>lineGetAgentGroupList</i>	yes
<i>lineGetAgentStatus</i>	yes
<i>lineSetAgentActivity</i>	yes
<i>lineSetAgentGroup</i>	no
<i>lineSetAgentState</i>	yes

Table 33: ACD-PRH Functions for TAPI 2.0 / 2.1

The following sections describe the Parameter usage for the supported ACD-PRH functions.

6.2.1 lineAgentSpecific

not supported

6.2.2 lineGetAgentActivityList

<code>hLine</code>	Handle to the open line device. (AgentDevice)
<code>dwAddressID</code>	not used, always 0
<code>lpAgentActivityList</code>	Pointer to a variably sized structure <code>LINEAGENTACTIVITYLIST</code> , containing <code>dwNumEntries</code> of <code>LINEAGENTACTIVITYENTRY</code>

LINEAGENTACTIVITYENTRY :

<code>dwID</code>	ID is the numerical representation of a <code>HiPathAgentID</code> *)
<code>dwNameSize</code>	not used, always 0
<code>dwNameOffset</code>	not used, always 0

- *) **(HiPathAgentID = „100“ <-> dwId=dwActivityID = 0x64)**
 The values may range from 100 up to 249. The value 0 is used for deletion.
 The `dwID` reflects the value that has been set via *lineSetAgentActivity*.

6.2.3 lineGetAgentCaps

hLineApp	The handle to the application's registration with TAPI.	
dwDeviceID	ID of the line device.	(AgentDevice, UCDDevice)
dwAddressID	not used, always 0	
dwAppAPIVersion	0x00020001	
lpAgentCaps	A pointer to a variably sized structure of type LINEAGENTCAPS.	

LINEAGENTCAPS:

dwAgentHandlerInfoSize	„ACD3Service“	
dwAgentHandlerInfoOffset		
dwCapsVersion	0x00020001	
dwFeatures	LINEAGENTFEATURE_SETAGENTGROUP LINEAGENTFEATURE_SETAGENTSTATE LINEAGENTFEATURE_SETAGENTACTIVITY LINEAGENTFEATURE_AGENTSPECIFIC LINEAGENTFEATURE_GETAGENTACTIVITYLIST LINEAGENTFEATURE_GETAGENTGROUP The feature set given above is only valid for AgentDevice . For UCDDevice only LINEAGENTFEATURE_SETAGENTSTATE is supported.	
dwStates	see Table 35: AgentStates for Agent- and UCD-Devices see Table 36: Getting / Setting AgentStates	
dwNextStates	not used, always 0	
dwMaxNumGroupEntries	not used, always 0	
dwAgentStatusMessages	LINEAGENTSTATUS_GROUP LINEAGENTSTATUS_STATE LINEAGENTSTATUS_NEXTSTATE LINEAGENTSTATUS_ACTIVITY *) LINEAGENTSTATUS_ACTIVITYLIST LINEAGENTSTATUS_GROUPLIST LINEAGENTSTATUS_CAPSCHANG LINEAGENTSTATUS_VALIDSTATES LINEAGENTSTATUS_VALIDNEXTSTATES	

*) only **AgentDevice**

dwNumAgentExtensionIDs	0
dwAgentExtensionIDListSize	0
dwAgentExtensionIDListOffset	0

6.2.4 lineGetAgentGroupList

The *lineGetAgentGroupList* function obtains the identities of all agent groups configured in HiPath TAPI 170.

hLine	Handle to the open line device. (AgentDevice)
dwAddressID	not used, always 0
lpAgentGroupList	A pointer to a variably sized structure of type <code>LINEAGENTGROUPLIST</code> . Upon successful completion of the request, this structure is filled with a list of the agent groups into which the agent can log in at this time.

6.2.5 lineGetAgentStatus

hLine	Handle to the open line device. (AgentDevice, UCDDevice)
dwAddressID	not used, always 0
lpAgentStatus	Pointer to a variably sized structure <code>LINEAGENTSTATUS</code> , containing <code>dwNumEntries</code> of <code>LINEAGENTGROUPEENTRY</code>

LINEAGENTSTATUS:

dwNumEntries	0 or 1
dwState	see Table 35: AgentStates for Agent- and UCD-Devices see Table 36: Getting / Setting AgentStates
dwNextState	not used, always 0
dwActivityID	ActivityID is the numerical representation of a HiPathAgentID *)
dwAgentFeatures	<code>LINEAGENTFEATURE_SETAGENTGROUP</code> <code>LINEAGENTFEATURE_SETAGENTSTATE</code> <code>LINEAGENTFEATURE_SETAGENTACTIVITY</code> <code>LINEAGENTFEATURE_AGENTSPECIFIC</code> <code>LINEAGENTFEATURE_GETAGENTACTIVITYLIST</code> <code>LINEAGENTFEATURE_GETAGENTGROUP</code> The feature set given above is only valid for AgentDevice . For UCDDevice only <code>LINEAGENTFEATURE_SETAGENTSTATE</code> is supported.
dwValidStates	see Table 35: AgentStates for Agent- and UCD-Devices see Table 36: Getting / Setting AgentStates
dwValidNextStates	not used, always 0

- *) **(HiPathAgentID = „100“ <-> dwActivityID = 0x64)**
The values may range from 100 up to 249. The value 0 is used for deletion. The `dwActivityID` reflects the value that has been set via `lineSetAgentActivity`.

6.2.6 lineSetAgentActivity

lineSetAgentActivity assigns an ActivityID (HicomAgentID) to a TAPI line device. The same ActivityID can only be active at one device at a time.

hLine	Handle to the open line device.	(AgentDevice)
dwAddressID	not used, always 0	
dwActivityID	ActivityID is the numerical representation of a HiPathAgentID	

(HiPathAgentID = „100“ <-> dwActivityID = 0x64)

The values may range from 100 up to 249. The value 0 is used for deletion.

6.2.7 lineSetAgentGroup

not supported

6.2.8 lineSetAgentState

hLine	Handle to the open line device. (AgentDevice, UCDDDevice)
dwAddressID	not used, always 0
dwAgentState	see Table 35: AgentStates for Agent- and UCD-Devices see Table 36: Getting / Setting AgentStates
dwNextAgentState	not used, always 0

Device = AgentDevice:

This function is used to set the AgentState of an agent. It can only be executed if an ActivityID has been assigned previously to this AgentDevice.

Device = UCDDDevice:

This function is used to control the HostControlledFlag and OverflowFlag of an UCDDDevice (see Table 34: HostControlledFlag and OverflowFlag). ActivityID is not relevant for UCDDDevices.

TAPI	HostControlledFlag	OverflowFlag
LINEAGENTSTATE_LOGGEDOFF	FALSE	FALSE
LINEAGENTSTATE_NOTREADY	TRUE	TRUE
LINEAGENTSTATE_READY	TRUE	FALSE

Table 34: HostControlledFlag and OverflowFlag

TAPI	CSTA	AgentDevice	UCDDevice
LINEAGENTSTATE_LOGGEDOFF	X	X	X
LINEAGENTSTATE_NOTREADY	X	X	X
LINEAGENTSTATE_READY	X	X	X
LINEAGENTSTATE_BUSYACD	X	X	
LINEAGENTSTATE_BUSYINCOMING			
LINEAGENTSTATE_BUSYOUTBOUND			
LINEAGENTSTATE_BUSYOTHER			
LINEAGENTSTATE_WORKINGAFTERCALL	X	X	
LINEAGENTSTATE_UNKNOWN		X	X
LINEAGENTSTATE_UNAVAIL			

Table 35: AgentStates for Agent- and UCD-Devices

Events	Query	lineSet Agent State	lineGet Agent Status	TAPI
X	X	X	X	LINEAGENTSTATE_LOGGEDOFF
X	X	X	X	LINEAGENTSTATE_NOTREADY
X	X	X	X	LINEAGENTSTATE_READY
X			X	LINEAGENTSTATE_BUSYACD
				LINEAGENTSTATE_BUSYINCOMING
				LINEAGENTSTATE_BUSYOUTBOUND
				LINEAGENTSTATE_BUSYOTHER
X	X	X	X	LINEAGENTSTATE_WORKINGAFTERCALL
			X	LINEAGENTSTATE_UNKNOWN
				LINEAGENTSTATE_UNAVAIL

Table 36: Getting / Setting AgentStates

6.3 ACD-ProxyRequestHandler Functions for TAPI 2.2 and 3.x

TAPI	Supported
<i>lineCreateAgent</i>	no
<i>lineCreateAgentSession</i>	no
<i>lineGetAgentInfo</i>	no
<i>lineGetAgentSessionInfo</i>	no
<i>lineGetAgentSessionList</i>	no
<i>lineGetGroupList</i>	no
<i>lineGetProxyStatus</i>	no
<i>lineGetQueueInfo</i>	no
<i>lineGetQueueList</i>	no
<i>lineSetAgentMeasurementPeriod</i>	no
<i>lineSetAgentStateEx</i>	no
<i>lineSetAgentSessionState</i>	no
<i>lineSetQueueMeasurementPeriod</i>	no

Table 37: ACD-PRH Functions for TAPI 2.2 and TAPI 3.x

The TAPI 2.2 and TAPI 3.x agent functionality (incl. ITTAPICallCenter interface) is not supported by ACD-PRH.

7 DevSpecific structures

7.1 struct CAPS_DEVSPECIFIC

This struct is **only supported for HiPath AllServe and HiPath 3000/5000 RSM.**

```
// structure of dev specific for get caps
typedef struct _CAPS_DEVSPECIFIC
{
    DWORD        dwSwitchID;
    long         nVNodeID;
    char         szName [TAPIMAXDESTADDRESSSIZE];
    char         szIPAdress [TAPIMAXDESTADDRESSSIZE];
    char         szNationalCode [TAPIMAXDESTADDRESSSIZE];
    char         szLocalAreaCode [TAPIMAXDESTADDRESSSIZE];
    char         szPBXNumber [TAPIMAXDESTADDRESSSIZE];
    char         szCallNumber [TAPIMAXDESTADDRESSSIZE];
    char         szDuwaNumber [TAPIMAXDESTADDRESSSIZE];
    BOOL        bLCRFlag;
    char         szCurrency [TAPIMAXDESTADDRESSSIZE];
} CAPS_DEVSPECIFIC;
```

Table 38: CAPS_DEVSPECIFIC

7.2 struct STATUS_DEVSPECIFIC

This struct is **not supported for Extended Services Lines.**

```
// structure of dev specific for get status
typedef struct _STATUS_DEVSPECIFIC
{
    BOOL        bInitialQueryCompleted;
} STATUS_DEVSPECIFIC;
```

Table 39: STATUS_DEVSPECIFIC

The meaning of `bInitialQueryCompleted` depends upon the TAPI struct it is returned within:

LINEADDRESSSTATUS	Forwarding / DND status
LINEDEVSTATUS	MsgWaiting status

Table 40: STATUS_DEVSPECIFIC.bInitialQueryCompleted

8 Neutral Lines

The concept of neutral lines represents a new feature for the dynamical assignment of devices as opposed to the statical assignment via User Interface (TSP configuration).

8.1 Procedures for Neutral Line Devices

In the following figures the procedures for neutral line devices will be shown by means of a sample configuration.

8.1.1 Initial Situation

The basic configuration of the communication system consists of 7 station devices with the callnumbers 200,201,202,300,301,355,356. Only 3 of them (200,201,202) are set up in the TSP configuration (statically assigned) to be controlled via TAPI.

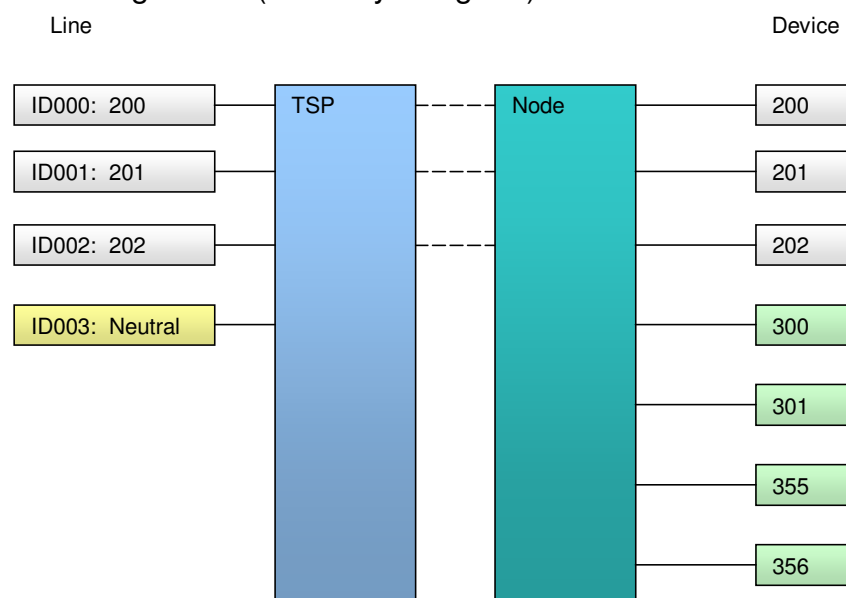


Figure 1: Initial Situation (neutral Lines)

8.1.2 Dynamical Assignment of a Callnumber

If it becomes necessary for a TAPI application to control a device that is not configured within the TSP, then it is possible to assign a callnumber (355) to the existing neutral line (ID003).

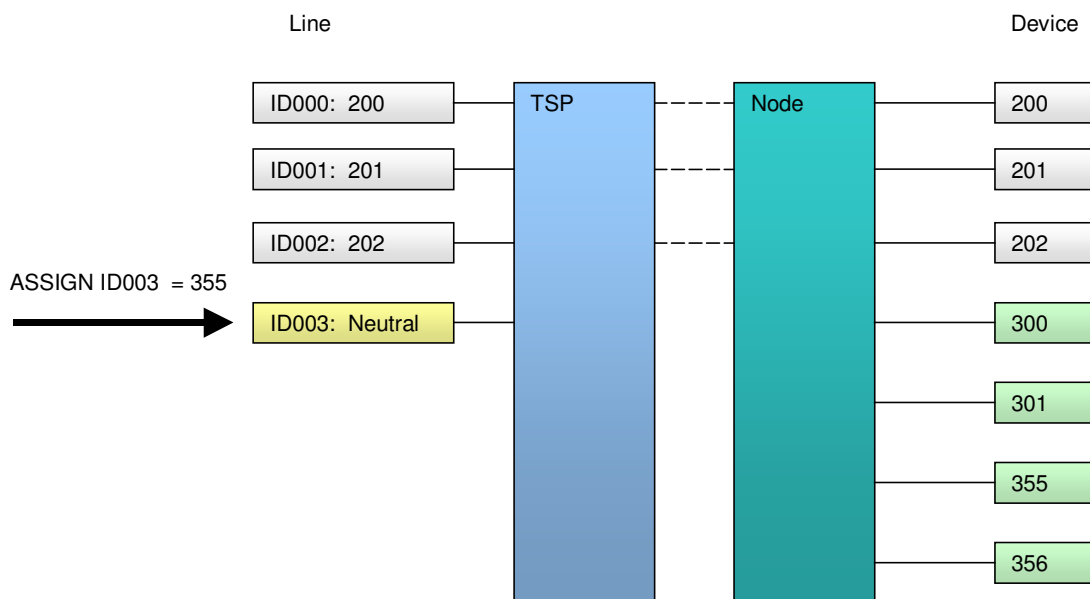


Figure 2: Dynamical Assignment of a Callnumber (neutral Lines)

8.1.3 Callnumber has been assigned

After the dynamical assignment of callnumber 355 to the neutral line (ID003) all concerned TAPI applications will be notified of this change. Additionally there will be a new neutral line (ID004) created and all concerned TAPI applications will be notified of this creation. This newly created line is now the proxy for the next dynamical assignment.

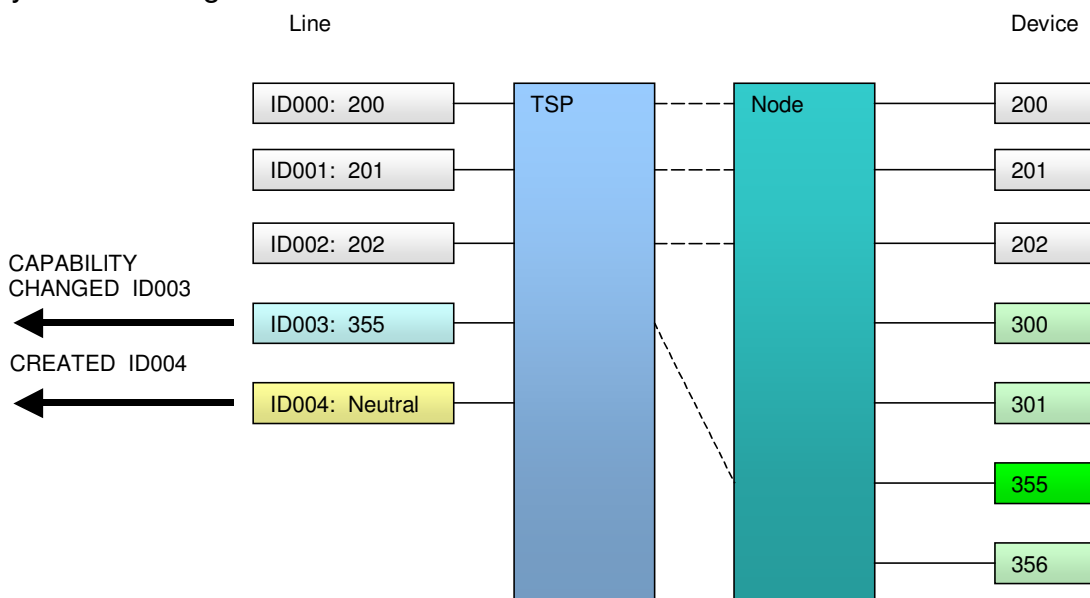


Figure 3: Callnumber has been assigned (neutral Lines)

8.1.4 Next Dynamical Assignment of a Callnumber

Now the next dynamical assignment of a callnumber (356) to the current neutral line (ID004) takes place.

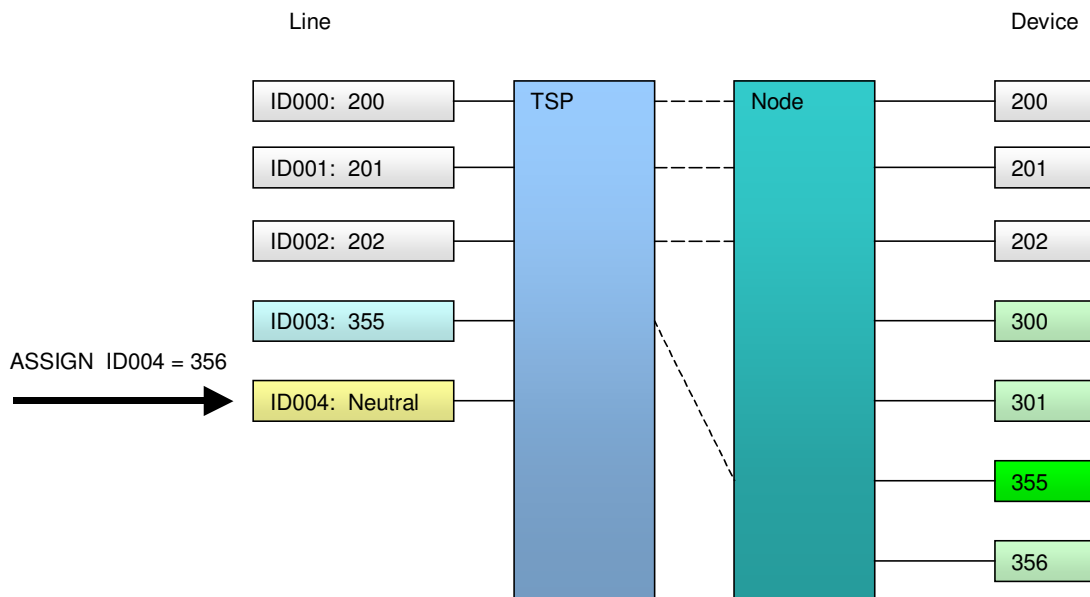


Figure 4: Next Dynamical Assignment of a Callnumber (neutral Lines)

8.1.5 Next Callnumber has been assigned

After this new dynamical assignment of a callnumber (356) to the neutral line (ID004) all concerned TAPI application will be notified of this change. Additionally again there will be a new neutral line (ID005) created and all concerned TAPI application will be notified of this creation. This newly created line is now the proxy for the next dynamical assignment.

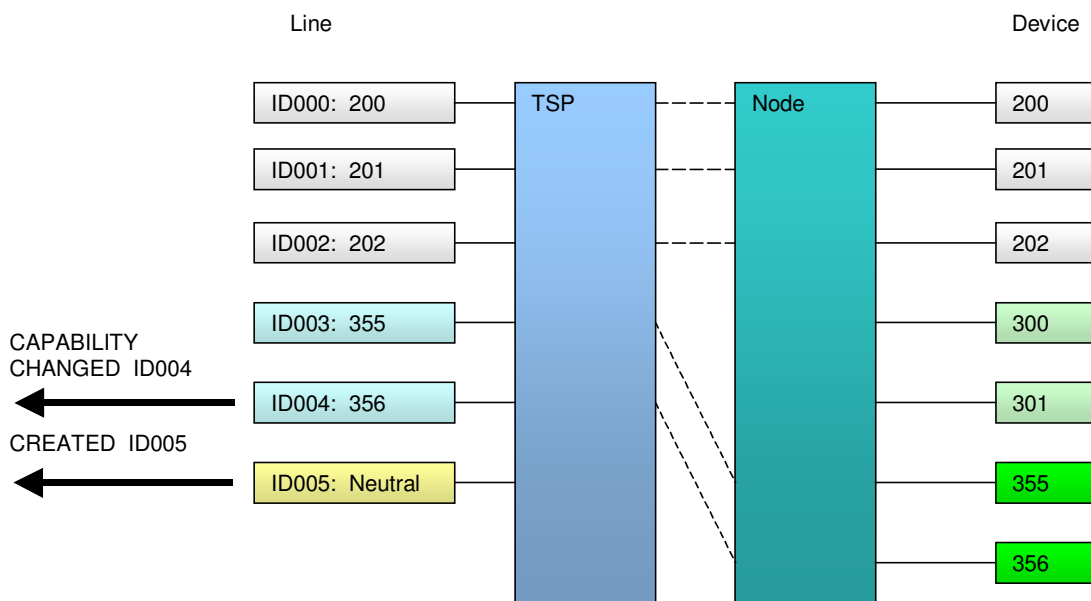


Figure 5: Next Callnumber has been assigned (neutral Lines)

8.1.6 Cancelling of a Callnummer Assignment

If an assigned line is no longer needed, it can be released by the TAPI application. In the sample it is shown for line ID 004 with the assigned callnumber 356.

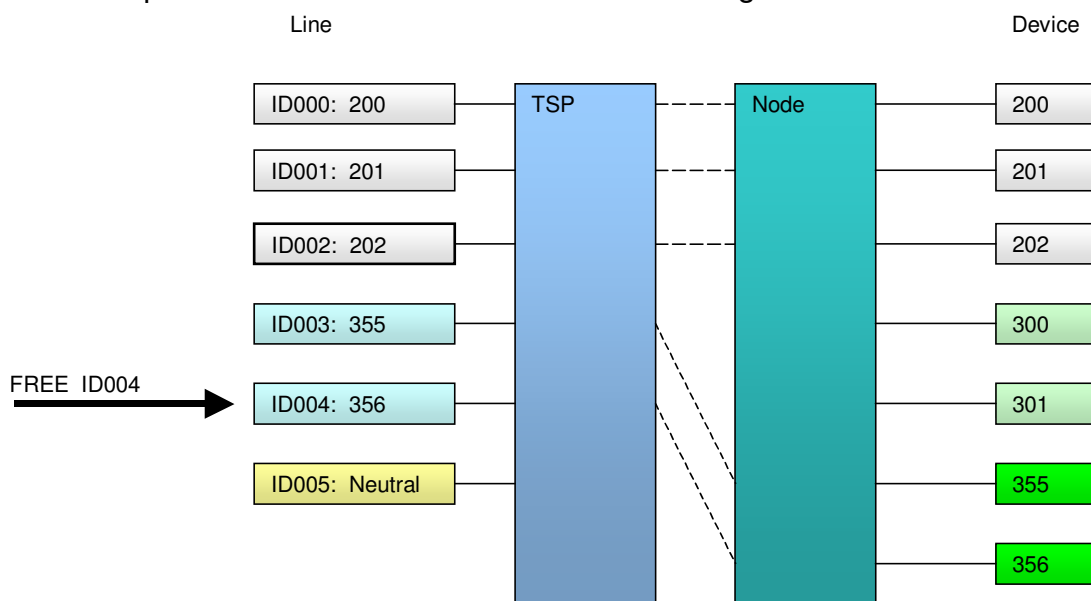


Figure 6: Cancelling of a Callnummer Assignment (neutral Lines)

8.1.7 Callnummer Assignment was cancelled

As soon as the callnummer assignment for a dynamically assigned line (ID004) was cancelled, this leads to the removal of this line. All concerned TAPI application will be notified of this removal.

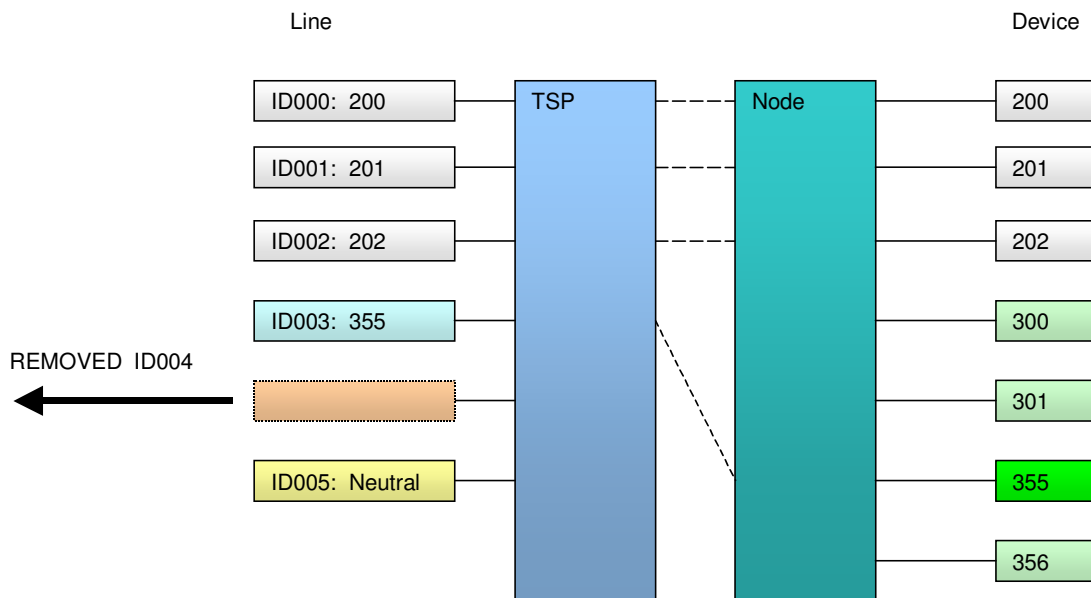


Figure 7: Callnummer Assignment was cancelled (neutral Lines)

8.2 Handling of Neutral Lines

8.2.1 Free neutral Line

There exists **exactly one** free neutral line.

When a callnumber is assigned to it, a **new free neutral line will be created**.

A free neutral line consists of a TAPI line device and a corresponding TAPI phone device.

8.2.1.1 Device Classes

To mark a free neutral line there exists two new device classes:

- "tapi/line/neutral/free"
- "tapi/phone/neutral/free"

Table 41: Device Classes for a free neutral Line

The new device classes are set in `LINEDEVCAPS`, `LINEADDRESSCAPS` and `PHONECAPS`.

8.2.1.2 Capabilities

- **Callnumber** "---" (`LINEADDRESSCAPS.Address`)
- **Name** "neutral-line" (`LINEDEVCAPS.LineName` and `PHONECAPS.PhoneName`)
- Additionally to the existing **DeviceClasses** "tapi/line" and "tapi/phone" the new device classes (s. section 8.2.1.1) are available

Table 42: Capabilities for a free neutral Line

8.2.1.3 Assignment

- *lineOpen*
 - will be rejected with `LINEERR_OPERATIONUNAVAIL`, if a callnumber assignment has not taken place so far
- *phoneOpen*
 - will be rejected with `PHONEERR_OPERATIONUNAVAIL`, if a callnumber assignment has not taken place so far
- *lineGetDevConfig*
 - for all allowed device classes `struct DEVICECONFIGEDIT` (s. Figure 8) in a `VARSTRING` is returned
- *lineSetDevConfig*
 - device data (callnumber and device name) to be assigned is expected as `struct DEVICECONFIGEDIT` (s. Figure 8) in `lpDeviceConfig` with `dwSize = sizeof(DEVICECONFIGEDIT)` for the new device classes (s. section 8.2.1.1)

- LINE_CREATE / PHONE_CREATE for new free neutral Line
- LINE_CREATE / PHONE_CREATE for new Extended Services Line, if node was not in scope before (HiPath 3000/5000 V4 only)

Table 43: Assignment for a free neutral Line

Callnumber and name of a device are members of the `struct DEVICECONFIGEDIT`:

```
typedef struct _DEVICECONFIGEDIT
{
    char        szStationID    [HICOMMAXIDSIZE];
    char        szStationName  [HICOMMAXNAME_SIZE];
}    DEVICECONFIGEDIT;
```

with

```
#define HICOMMAXIDSIZE        13    // new since TSP V3.2.0.670
#define HICOMMAXNAME_SIZE    41
```

Starting with TSP Version 3.2.0.670 the width of callnumbers has been extended from 6 to 12 characters (`HICOMMAXIDSIZE-1`). Since this changes the **sizeof(DEVICECONFIGEDIT)**, an additional `struct DEVICECONFIGEDIT_OLD` has been introduced for backward compatibility:

```
typedef struct _DEVICECONFIGEDIT_OLD
{
    char        szStationID    [HICOMMAXIDSIZE_OLD];
    char        szStationName  [HICOMMAXNAME_SIZE];
}    DEVICECONFIGEDIT_OLD;
```

with

```
#define HICOMMAXIDSIZE_OLD    7
#define HICOMMAXNAME_SIZE    41
```

The compatibility mechanism works as follows:

- *lineGetDevConfig* is returning `struct DEVICECONFIGEDIT_OLD` if the callnumber width is up to 6 chars, else it is returning `struct DEVICECONFIGEDIT`
- *lineSetDevConfig* is accepting `struct DEVICECONFIGEDIT_OLD` and `struct DEVICECONFIGEDIT`. Of course with `struct DEVICECONFIGEDIT_OLD` the callnumber width is restricted to 6 chars

Figure 8: DEVICECONFIGEDIT for *lineGetDevConfig* / *lineSetDevConfig*

8.2.2 Assigned neutral Lines

There may exist **any number** (0 .. "infinity") of assigned neutral lines.

If a callnumber assignment is cancelled, the **assigned neutral line will be removed**.

8.2.2.1 Device Classes

To mark an assigned neutral line there exists two new device classes:

- "tapi/line/neutral/assigned"
- "tapi/phone/neutral/assigned"

Table 44: Device Classes for an assigned neutral Line

The new device classes are set in `LINEDEVCAPS`, `LINEADDRESSCAPS` and `PHONECAPS`.

8.2.2.2 Cancelling of an assignment

- *lineGetDevConfig*
 - for all allowed device classes `struct DEVICECONFIGEDIT` (s. Figure 8) in a `VARSTRING` is returned
- *lineSetDevConfig*
 - `dwSize = 0` and `lpDeviceConfig = NULL` are expected for the new device classes (s. section 8.2.2.1)
 - `PHONE_STATE (PHONESTATE_REMOVED) / PHONE_REMOVE / LINE_LINEDEVSTATE (LINEDEVSTATE_REMOVED) / LINE_REMOVE` for assigned neutral Line

Table 45: Cancelling of an assignment for an assigned neutral Line

8.2.2.3 Capabilities

- assigned callnumber (`LINEADDRESSCAPS.Address`)
- assigned name (`LINEDEVCAPS.LineName` and `PHONECAPS.PhoneName`)
- Additionally to the existing **DeviceClasses** "tapi/line" and "tapi/phone" the new device classes (s. section 8.2.2.1) are available

Table 46: Capabilities for an assigned neutral Line

9 References

- /1/ TAPI 3.1 / TAPI 2.2 / TSPI 2.1
MSDN-Library October 2001
Microsoft
- /2/ TAPI 2.1 SDK
June '97
Microsoft
- /3/ Windows Telephony Programming : A Developer's Guide to TAPI
First printing, July 1998
Chris Sells
Addison-Wesley, ISBN 0-201-63450-3
- /4/ Computer Telephony Demystified
Putting CTI, Media Services, and IP Telephony to Work
First printing, 2001
Michael Bayer
McGraw-Hill, ISBN 0-07-135987-7
- /5/ ASN.1
Abstract Syntax Notation One
3. Auflage, 1998
Walter Gora
FOSSIL-Verlag, ISBN 3-931959-18-X
- /6/ CSTA Interface Manual
CSTA-Link for HiPath 3000/5000 and HiPath 2000 and HiPath OpenOffice EE
Siemens Enterprise Communications GmbH & Co. KG, SEN ESY SME MP 51
http://wiki.siemens-enterprise.com/index.php/HiPath_3000_open_interfaces
- /7/ Standard ECMA - 269
Services for Computer Supported Telecommunications Applications (CSTA)
Phase III
4th Edition - June 2000
ECMA

/8/ Standard ECMA - 285
Protocol for Computer Supported Telecommunications Applications (CSTA)
Phase III
December 1998
ECMA

10 NewsGroups

/9/ <news:\\msnews.microsoft.com\microsoft.public.win32.programmer.tapi>

11 Internet-Links

- /10/ Andreas Marschall
TAPI-FAQ, TAPI-Tools
[http://www.i-b-a-m.de/Andreas Marschall's TAPI and TSPI FAQ.htm](http://www.i-b-a-m.de/Andreas_Marschall's_TAPI_and_TSPI_FAQ.htm)
[http://www.i-b-a-m.de/Andreas Marschall's Toto Tools.htm](http://www.i-b-a-m.de/Andreas_Marschall's_Toto_Tools.htm)
- /11/ Matthias Moetje
TAPI-Wiki
<http://www.tapi.info/>
- /12/ Grant Schenck
TAPI-FAQ, Debug-Tools
<http://grantschenck.tripod.com/>
- /13/ Bruce Pennypacker
TAPI-FAQ
<http://tapifaq.pennypacker.org/>
- /14/ Michael Dunn
TAPI-FAQ, TAPI-Tools
<http://www.rainyjay.com/tapi/tapi.htm>
- /15/ Chris Sells
TAPI-Tools
[http://www.sellsbrothers.com/tools/#TAPI Explorer](http://www.sellsbrothers.com/tools/#TAPI_Explorer)
<http://mysite.verizon.net/robert.bamberg/csells.htm>
- /16/ JulMar Technology
TAPI-Tools
<http://www.julmar.com/tapi>
- /17/ Sysinternals Freeware (Mark Russinovich & Bryce Cogswell)
Debug-Tools
<http://www.sysinternals.com/>
- /18/ Microsoft
TAPI-Tools
<ftp://ftp.microsoft.com/developr/tapi/>

About Unify

Unify is one of the world's leading communications software and services firms, providing integrated communications solutions for approximately 75 percent of the Fortune Global 500. Our solutions unify multiple networks, devices and applications into one easy-to-use platform that allows teams to engage in rich and meaningful conversations. The result is a transformation of how the enterprise communicates and collaborates that amplifies collective effort, energizes the business, and enhances business performance. Unify has a strong heritage of product reliability, innovation, open standards and security.

Unify.com

Copyright © Unify Software and Solutions GmbH & Co. KG 2015
Mies-van-der-Rohe-Str. 6, 80807 Munich/Germany
All rights reserved.

The information provided in this document contains merely general descriptions or characteristics of performance which in case of actual use do not always apply as described or which may change as a result of further development of the products. An obligation to provide the respective characteristics shall only exist if expressly agreed in the terms of contract.

Availability and technical specifications are subject to change without notice.

Unify, OpenScape, OpenStage and HiPath are registered trademarks of Unify Software and Solutions GmbH & Co. KG. All other company, brand, product and service names are trademarks or registered trademarks of their respective holders.

UNIFY Harmonize
your enterprise