

TR-064 Support – X_AVM-DE_OnTel

Supported by AVM

Author: AVM GmbH

Date: 2020-06-12

1 urn:X_AVM-DE_OnTel-com:serviceId:X_AVM-DE_OnTel1

The service allows to configure contact features. The following features are covered by this service:

- online telephone books,
- telephone books,
- DECT handsets,

the call list.

1.1 History

Date	Version	Changes
2011-02-24	11	Shot usage description for GetPhonebook and SetPhonebookEntry.
2011-10-14	12	XML description for call list and phonebook
2011-10-20	13	Call list URL parameter days.
2011-10-28	14	Add Port tag to call list XML file content.
2012-08-21	15	Call list contents.
2012-09-24	16	Call list: rejected calls.
2012-10-09	17	Call List Content, URLs: parameter used twice in one URL Description to differ between voice calls, fax calls and tam calls.
2012-11-15	18	Unique ID and timestamp in call list content. New parameters for call list URL.
2012-11-26	19	Detailed description for timestamp and ID in Table 69: Call list content description and Table 18: URL parameters for CallListURL .
2013-04-04	20	Description for new parameter for phonebook URL in chapter 2.11 GetPhonebook . Description for XML content of phonebook in chapter 5.1 Phonebook Content .
2013-04-15	21	Phonebook XML content: new tag uniqueid
2014-05-27	22	Add number used for call in addition to the name for this number.
2015-08-21	23	New functions for call deflection
2015-08-28	24	Add enum parameter for the call deflection
2015-10-01	25	Corrected description for eLongDelayed
2015-10-29	26	TypeEnum: Value fon changed to fon1, fon2, fon3, fon4
2015-11-20	27	Del action GetMessageStatus
2016-03-08	28	Change Action GetCallList
2016-09-27	29	Added eOff in ModeEnum List.
2018-06-05	30	New actions: GetPhonebookEntryUID, SetPhonebookEntryUID, DeletePhonebookEntryUID Added return codes tables.
2018-07-17	31	Minor spelling fix
2019-01-10	32	Changed action description for DeletePhonebook
2019-07-02	33	Added actions: GetCallBarringEntry GetCallBarringEntryByNum GetCallBarringList SetCallBarringEntry DeleteCallBarringEntryUID
2020-06-12	34	Additional information for calllist types added.

Content

1	<i>urn:X_AVM-DE_OnTel-com:serviceId:X_AVM-DE_OnTel1</i>	1
1.1	History.....	2
2	Action List.....	4
2.1	GetInfo.....	4
2.2	SetEnable.....	4
2.3	SetConfig.....	5
2.4	GetInfoByIndex.....	5
2.5	SetEnableByIndex.....	5
2.6	SetConfigByIndex.....	6
2.7	GetNumberOfEntries.....	6
2.8	DeleteByIndex.....	7
2.9	GetCallList.....	7
2.10	GetPhonebookList.....	8
2.11	GetPhonebook.....	8
2.12	AddPhonebook.....	9
2.13	DeletePhonebook.....	9
2.14	GetPhonebookEntry.....	9
2.15	GetPhonebookEntryUID.....	10
2.16	SetPhonebookEntry.....	10
2.17	SetPhonebookEntryUID.....	11
2.18	DeletePhonebookEntry.....	11
2.19	DeletePhonebookEntryUID.....	12
2.20	GetCallBarringEntry.....	12
2.21	GetCallBarringEntryByNum.....	12
2.22	GetCallBarringList.....	13
2.23	SetCallBarringEntry.....	13
2.24	DeleteCallBarringEntryUID.....	14
2.25	GetDECTHandsetList.....	14
2.26	GetDECTHandsetInfo.....	14
2.27	SetDECTHandsetPhonebook.....	15
2.28	GetNumberOfDeflections.....	15
2.29	GetDeflection.....	15
2.30	GetDeflections.....	16
2.31	SetDeflectionEnable.....	16
3	Parameters.....	17
3.1	ModeEnum.....	17
3.2	TypeEnum.....	17
4	Service States Table.....	18
5	XML Document Contents.....	19
5.1	Phonebook Content.....	19
5.1.1.1	URLs.....	20
5.2	Call List Content.....	21
5.2.1.1	URLs.....	22
5.2.1.2	Types.....	22
5.3	DeflectionList Example.....	23

2 Action List

This chapter contains the supported actions of the service X_AVM-DE_OnTel which are listed incl. arguments.

2.1 GetInfo

This action is obsolete and shall not be used any more.

Argument name	Direction	Related state variable	Remarks
NewEnable	out	Enable	
NewStatus	out	Status	
NewLastConnect	out	LastConnect	
NewUrl	out	Url	
NewServiceId	out	ServiceId	
NewUsername	out	Username	
NewName	out	Name	

Table 1: Argument list of action GetInfo

Return code	Description	Related argument
402	Invalid arguments	Any
713	Invalid array index	

Table 2: Return codes of action GetInfo

2.2 SetEnable

The action is used to trigger the telephone book synchronization manually. The synchronization starts if switching from false to true. After enabling, the synchronization is automatically started periodically once within 24 hours.

This action is obsolete and shall not be used any more.

Argument name	Direction	Related state variable	Remarks
NewEnable	in	Enable	

Table 3: Argument list of action SetEnable

Return code	Description	Related argument
402	Invalid arguments	Any
713	Invalid array index	

Table 4: Return codes of action SetEnable

2.3 SetConfig

This action is obsolete and shall not be used any more.

Argument name	Direction	Related state variable	Remarks
NewEnable	in	Enable	
NewUrl	in	Url	
NewServiceId	in	ServiceId	
NewUsername	in	Username	
NewPassword	in	Password	
NewName	in	Name	Telephone book name

Table 5: Argument list of action SetConfig

Return code	Description	Related argument
713	Invalid array index	Any input parameter

Table 6: Return codes of action SetConfig

2.4 GetInfoByIndex

Argument name	Direction	Related state variable	Remarks
NewIndex	in	Index	
NewEnable	out	Enable	
NewStatus	out	Status	
NewLastConnect	out	LastConnect	
NewUrl	out	Url	
NewServiceId	out	ServiceId	
NewUsername	out	Username	
NewName	out	Name	

Table 7: Argument list of action GetInfoByIndex

Return code	Description	Related argument
402	Invalid arguments	Any
713	Invalid array index	Index

Table 8: Return codes of action GetInfoByIndex

2.5 SetEnableByIndex

The action is used to trigger the telephone book synchronization manually. The synchronization starts if switching from false to true. After enabling, the synchronization is automatically started periodically once within 24 hours.

All accounts are triggered to check for updates on COMS by invoking this action. If the revision has not increased, no synchronization will be made.

Argument name	Direction	Related state variable	Remarks
NewIndex	in	Index	
NewEnable	in	Enable	

Table 9: Argument list of action SetEnableByIndex

Return code	Description	Related argument
402	Invalid arguments	Any
713	Invalid array index	Index

Table 10: Return codes of action SetEnableByIndex

2.6 SetConfigByIndex

If the given index addresses an existing account the configuration is changed. If the index addresses a new account and the index is `OntelNumberOfEntries + 1` then a new account is generated.

Argument name	Direction	Related state variable	Remarks
NewIndex	in	Index	
NewEnable	in	Enable	
NewUrl	in	Url	
NewServiceId	in	ServiceId	
NewUsername	in	Username	
NewPassword	in	Password	
NewName	in	Name	Telephone book name

Table 11: Argument list of action SetConfigByIndex

Return code	Description	Related argument
713	Invalid array index	Any input parameter

Table 12: Return codes of action SetConfigByIndex

2.7 GetNumberOfEntries

Argument name	Direction	Related state variable	Remarks
NewOntelNumberOfEntries	out	OntelNumberOfEntries	

Table 13: Argument list of action GetNumberOfEntries

Return code	Description	Related argument
820	Internal error	

Table 14: Return codes of action GetNumberOfEntries

2.8 DeleteByIndex

Argument name	Direction	Related state variable	Remarks
NewIndex	in	Index	

Table 15: Argument list of action DeleteByIndex

Return code	Description	Related argument
402	Invalid arguments	Any
713	Invalid array index	Index

Table 16: Return codes of action DeleteByIndex

2.9 GetCallList

The URL can be extended to limit the number of entries in the XML call list file.

E.g. max=42 would limit to 42 calls in the list.

If the parameter is not set or the value is 0 all calls will be inserted into the call list file.

The URL can be extended to fetch a limited number of entries using the parameter days.

E.g. days=7 would fetch the calls from now until 7 days in the past.

If the parameter is not set or the value is 0 all calls will be inserted into the call list file.

The parameter NewCallListURL is empty, if the feature (CallList) is disabled. If the feature is not supported an internal error (820) is returned. In the other case the URL is returned.

Argument name	Direction	Related state variable	Remarks
NewCallListURL	out	CallListURL	

Table 17: Argument list of action GetCallList

The following URL parameters are supported.

Parameter name	Type	Remarks
days	number	number of days to look back for calls e.g. 1: calls from today and yesterday, 7: calls from the complete last week, default 999
id	number	calls since this unique ID
max	number	maximum number of entries in call list, default 999
sid	hex-string	Session ID for authentication
timestamp	number	value from timestamp tag, to get only entries that are newer (timestamp is resetted by a factory reset)
tr064sid	string	Session ID for authentication (obsolete)
type	string	optional parameter for type of output file: xml (default) or csv

Table 18: URL parameters for CallListURL

The parameters timestamp and id have to be used in combination. If only one of both is used, the feature is not supported.

Return code	Description	Related argument
820	Internal error	

Table 19: Return codes of action GetCallList

2.10 GetPhonebookList

Argument name	Direction	Related state variable	Remarks
NewPhonebookList	out	PhonebookList	

Table 20: Argument list of action GetPhonebookList

Return code	Description	Related argument
402	Invalid arguments	Any
820	Internal Error	

Table 21: Return codes of action GetPhonebookList

2.11 GetPhonebook

The value of PhonebookExtraID may be an empty string.

Argument name	Direction	Related state variable	Remarks
NewPhonebookID	in	PhonebookID	
NewPhonebookURL	out	PhonebookURL	
NewPhonebookName	out	PhonebookName	
NewPhonebookExtraID	out	PhonebookExtraID	New since 2011-01-31

Table 22: Argument list of action GetPhonebook

The following URL parameters are supported.

Parameter name	Type	Remarks
pbid	number	Phonebook ID
max	number	maximum number of entries in call list, default 999
sid	hex-string	Session ID for authentication
timestamp	number	value from timestamp tag, to get the phonebook content only if last modification was made after this timestamp
tr064sid	string	Session ID for authentication (obsolete)

Table 23: URL parameters for PhonebookURL

Return code	Description	Related argument
402	Invalid arguments	Any
713	Invalid array index	PhonebookID
820	Internal Error	

Table 24: Return codes of action GetPhonebook

2.12 AddPhonebook

Argument name	Direction	Related state variable	Remarks
NewPhonebookName	in	PhonebookName	
NewPhonebookExtraID	in	PhonebookExtraID	Optional

Table 25: Argument list of action AddPhonebook

Return code	Description	Related argument
402	Invalid arguments	Any
820	Internal Error	

Table 26: Return codes of action AddPhonebook

2.13 DeletePhonebook

The default phonebook (PhonebookID = 0) is not deletable, but therefore, each entry will be deleted and the phonebook will be empty afterwards.

Argument name	Direction	Related state variable	Remarks
NewPhonebookID	in	PhonebookID	
NewPhonebookExtraID	in	PhonebookExtraID	Optional

Table 27: Argument list of action DeletePhonebook

Return code	Description	Related argument
402	Invalid arguments	Any
713	Invalid array index	Any input parameter
820	Internal Error	

Table 28: Return codes of action DeletePhonebook

2.14 GetPhonebookEntry

Get a single telephone book entry from the specified book.

Argument name	Direction	Related state variable	Remarks
NewPhonebookID	in	PhonebookID	
NewPhonebookEntryID	in	PhonebookEntryID	
NewPhonebookEntryData	out	PhonebookEntryData	

Table 29: Argument list of action GetPhonebookEntry

Return code	Description	Related argument
402	Invalid arguments	Any
600	Argument invalid	PhonebookID
713	Invalid array index	Any input parameter
820	Internal Error	

Table 30: Return codes of action GetPhonebookEntry

2.15 GetPhonebookEntryUID

Get a single telephone book entry from the specified book using the unique ID from the entry.

Argument name	Direction	Related state variable	Remarks
NewPhonebookID	in	PhonebookID	
NewPhonebookEntryUniqueID	in	PhonebookEntryUniqueID	
NewPhonebookEntryData	out	PhonebookEntryData	

Table 31: Argument list of action GetPhonebookEntryUID

Return code	Description	Related argument
402	Invalid arguments	Any
600	Argument invalid	PhonebookID
713	Invalid array index	Any input parameter
820	Internal Error	

Table 32: Return codes of action GetPhonebookEntryUID

2.16 SetPhonebookEntry

Add a new or change an existing entry in a telephone book.

Add new entry:

- set phonebook ID and an empty value for PhonebookEntryID and XML entry data structure (without the unique ID tag)

Change existing entry:

- set phonebook ID an entry ID and XML entry data (without the unique ID tag)
- set phonebook ID and an empty value for PhonebookEntryID and XML entry data structure with the unique ID tag (e.g. <uniqueid>28</uniqueid>)

Changes to online phonebooks are not allowed.

Argument name	Direction	Related state variable	Remarks
NewPhonebookID	in	PhonebookID	
NewPhonebookEntryID	in	PhonebookEntryID	
NewPhonebookEntryData	in	PhonebookEntryData	

Table 33: Argument list of action SetPhonebookEntry

Return code	Description	Related argument
402	Invalid arguments	Any
600	Argument invalid	PhonebookID
713	Invalid array index	PhonebookID, PhonebookEntryID
820	Internal Error	

Table 34: Return codes of action SetPhonebookEntry

2.17 SetPhonebookEntryUID

Add a new or change an existing entry in a telephone book using the unique ID of the entry.

Add new entry:

- set phonebook ID and XML entry data structure (without the unique ID tag)

Change existing entry:

- set phonebook ID and XML entry data structure with the unique ID tag (e.g. <uniqueid>28</uniqueid>)

The action returns the unique ID of the new or changed entry.

Argument name	Direction	Related state variable	Remarks
NewPhonebookID	in	PhonebookID	
NewPhonebookEntryData	in	PhonebookEntryData	
NewPhonebookEntryUniqueID	out	PhonebookEntryUniqueID	

Table 35: Argument list of action SetPhonebookEntryUID

Return code	Description	Related argument
402	Invalid arguments	Any
600	Argument invalid	PhonebookID
713	Invalid array index	PhonebookID
820	Internal Error	

Table 36: Return codes of action SetPhonebookEntryUID

2.18 DeletePhonebookEntry

Delete an existing telephone book entry.

Changes to online phonebooks are not allowed.

Argument name	Direction	Related state variable	Remarks
NewPhonebookID	in	PhonebookID	
NewPhonebookEntryID	in	PhonebookEntryID	

Table 37: Argument list of action DeletePhonebookEntry

Return code	Description	Related argument
402	Invalid arguments	Any
600	Argument invalid	PhonebookID
713	Invalid array index	Any input parameter
820	Internal Error	

Table 38: Return codes of action DeletePhonebookEntry

2.19 DeletePhonebookEntryUID

Delete an existing telephone book entry using the unique ID from the entry. Changes to online phonebooks are not allowed.

Argument name	Direction	Related state variable	Remarks
NewPhonebookID	in	PhonebookID	
NewPhonebookEntryUniqueID	in	PhonebookEntryUniqueID	

Table 39: Argument list of action DeletePhonebookEntryUID

Return code	Description	Related argument
402	Invalid arguments	Any
600	Argument invalid	PhonebookID
713	Invalid array index	Any input parameter
820	Internal Error	

Table 40: Return codes of action DeletePhonebookEntryUID

2.20 GetCallBarringEntry

Returns a call barring entry by its PhonebookEntryID of the specific call barring phonebook.

Argument name	Direction	Related state variable	Remarks
NewPhonebookEntryID	in	PhonebookEntryID	
NewPhonebookEntryData	out	PhonebookEntryData	

Table 41: Argument list of action GetCallBarringEntry

Return code	Description	Related argument
402	Invalid arguments	Any
713	Invalid array index	NewPhonebookID
820	Internal Error	

Table 42: Return codes of action GetCallBarringEntry

2.21 GetCallBarringEntryByNum

Returns a call barring entry by its number. If the number exists in the internal phonebook but not in the specific call barring phonebook, error code 714 is returned.

Argument name	Direction	Related state variable	Remarks
NewNumber	in	Number	
NewPhonebookEntryData	out	PhonebookEntryData	

Table 43: Argument list of action GetCallBarringEntryByNum

Return code	Description	Related argument
402	Invalid arguments	Any
713	Invalid array index	NewNumber
714	No such array entry in array	NewNumber
820	Internal Error	

Table 44: Return codes of action *GetCallBarringEntryByNum*

2.22 GetCallBarringList

Returns a url which leads to an xml formatted file which contains all entries of the call barring phonebook.

Argument name	Direction	Related state variable	Remarks
NewPhonebookURL	out	PhonebookURL	

Table 45: Argument list of action *GetCallBarringList*

Return code	Description	Related argument
402	Invalid arguments	Any
820	Internal Error	

Table 46: Return codes of action *GetCallBarringList*

2.23 SetCallBarringEntry

Add a phonebook entry to the specific call barring phonebook. When no uniqueid is given a new entry is created. Even when an entry with the given number is already existing. When a uniqueid is set which already exist, this entry will be overwritten. When a uniqueid is given which does not exist, a new entry is created and the new uniqueid is returned in argument NewPhonebookEntryUniqueID.

Argument name	Direction	Related state variable	Remarks
NewPhonebookEntryData	in	PhonebookEntryData	
NewPhonebookEntryUniqueID	out	PhonebookEntryUniqueID	

Table 47: Argument list of action *SetCallBarringEntry*

Return code	Description	Related argument
402	Invalid arguments	Any
820	Internal Error	

Table 48: Return codes of action *SetCallBarringEntry*

2.24 DeleteCallBarringEntryUID

Delete an entry of the call barring phonebook by its uniqueid.

Argument name	Direction	Related state variable	Remarks
NewPhonebookEntryUniqueID	in	PhonebookEntryUniqueID	

Table 49: Argument list of action DeleteCallBarringEntryUID

Return code	Description	Related argument
402	Invalid arguments	Any
820	Internal Error	

Table 50: Return codes of action DeleteCallBarringEntryUID

2.25 GetDECTHandsetList

Argument name	Direction	Related state variable	Remarks
NewDectIDList	out	DectIDList	

Table 51: Argument list of action GetDECTHandsetList

Return code	Description	Related argument
501	Action failed	

Table 52: Return codes of action GetDECTHandsetList

2.26 GetDECTHandsetInfo

Argument name	Direction	Related state variable	Remarks
NewDectID	in	DectID	
NewHandsetName	out	HandsetName	
NewPhonebookID	out	PhonebookID	

Table 53: Argument list of action GetDECTHandsetInfo

Return code	Description	Related argument
402	Invalid arguments	Any
501	Action failed	

Table 54: Return codes of action GetDECTHandsetInfo

2.27 SetDECTHandsetPhonebook

Argument name	Direction	Related state variable	Remarks
NewDectID	in	DectID	
NewPhonebookID	in	PhonebookID	

Table 55: Argument list of action SetDECTHandsetPhonebook

Return code	Description	Related argument
402	Invalid arguments	Any
501	Action failed	

Table 56: Return codes of action SetDECTHandsetPhonebook

2.28 GetNumberOfDeflections

Get the number of deflection entries.

Argument name	Direction	Related state variable	Remarks
NewNumberOfDeflections	out	NumberOfDeflections	

Table 57: Argument list of action GetNumberOfDeflections

Return code	Description	Related argument
402	Invalid arguments	Any
820	Internal Error	Output parameter

Table 58: Return codes of action GetNumberOfDeflections

2.29 GetDeflection

Get the parameter for a deflection entry.

DeflectionID is in the range of 0 .. NumberOfDeflections-1.

Argument name	Direction	Related state variable	Remarks
NewDeflectionId	in	DeflectionId	
NewEnable	out	Enable	
NewType	out	Type	
NewNumber	out	Number	
NewDeflectionToNumber	out	DeflectionToNumber	
NewMode	out	Mode	
NewOutgoing	out	Outgoing	
NewPhonebookID	out	PhonebookID	Only valid if Type==fromPB

Table 59: Argument list of action GetDeflection

Return code	Description	Related argument
402	Invalid arguments	Any
713	Invalid array index	Any input parameter
820	Internal Error	Output parameter

Table 60: Return codes of action *GetDeflection*

2.30 *GetDeflections*

Returns a list of deflections.

Argument name	Direction	Related state variable	Remarks
NewDeflectionList	out	DeflectionList	

Table 61: Argument list of action *GetDeflections*

Return code	Description	Related argument
402	Invalid arguments	Any
820	Internal Error	Any

Table 62: Return codes of action *GetDeflections*

2.31 *SetDeflectionEnable*

Enable or disable a deflection.

DeflectionID is in the range of 0 .. NumberOfDeflections-1.

Argument name	Direction	Related state variable	Remarks
NewDeflectionId	in	DeflectionId	
NewEnable	in	Enable	

Table 63: Argument list of action *SetDeflectionEnable*

Return code	Description	Related argument
402	Invalid arguments	Any input parameter
713	Invalid array index	Any input parameter
820	Internal Error	Any input parameter

Table 64: Return codes of action *SetDeflectionEnable*

3 Parameters

3.1 ModeEnum

Value	Description
eBellBlockade	Deflect if a bell blockade is activ
eBusy	Busy
eDelayed	Deflect with a delay
eDelayedOrBusy	Deflect if busy or with a delay
eDirectCall	Direct call
eImmediately	Deflect immediately
eLongDelayed	Deflect with a long delay
eNoSignal	Do not signal this call
eOff	Deflect disabled
eParallelCall	Parallel call
eShortDelayed	Deflect with a short delay
eUnknown	Mode unknown
eVIP	VIP

Table 65: Parameter type ModeEnum

3.2 TypeEnum

Value	Description
fon1	Phone port 1 is selected
fon2	Phone port 2 is selected
fon3	Phone port 3 is selected
fon4	Phone port 4 is selected
fromAll	From all
fromAnonymous	From a anonymous call
fromNotVIP	Call not from a VIP
fromNumber	Specific Number
fromPB	The caller is in the phonebook
fromVIP	Call from a VIP
toAny	To Any
toMSN	To MSN
toPOTS	To POTS
toVoIP	To VoIP
unknown	Type unknown

Table 66: Parameter type TypeEnum

4 Service States Table

Variable name	Allowed values (* == default)	Data type
CallListURL		String
DectID		String
DectIDList	Comma separated list of DectID	String
DeflectionId		ui2
DeflectionList	XML struct, example in 5.3.DeflectionList Example	String
DeflectionToNumber		String
Enable		Bool
Fax		ui2
Foncalls		ui2
HandsetName		String
Index	1 ... 10	ui2
LastConnect		Date
Mode		ModeEnum
Name		String
Number		String
NumberOfDeflections		ui2
OntelNumberOfEntries	0 ... 10	ui2
Outgoing	MSN Index "0"- "9", "SIPx (SIP99 für GSM)", "POTS", "" (default)	String
Password		String
PhonebookEntryData	XML document with a single entry.	String
PhonebookEntryID	Number for a single entry in a phonebook.	ui4
PhonebookEntryUniqueID	Unique identifier (number) for a single entry in a phonebook.	ui4
PhonebookExtraID	Optional parameter to make a phonebook unique.	String
PhonebookID		ui4
PhonebookList	Comma separated list of PhonebookID	String
PhonebookName		String
PhonebookURL		String
ServiceId		String
Status		String
Type		TypeEnum
Url		String
Username		String

Table 67: Variable list

5 XML Document Contents

5.1 Phonebook Content

The following shows an example XML content for a telephone book.

```
<?xml version="1.0"?>
<phonebooks>
  <phonebook owner="0" name="Telefonbuch">
    <!-- number of contacts 17 -->
    <contact>
      <!-- numbers 1 -->
      <category/>
      <person>
        <realName>Alle (Rundruf)</realName>
      </person>
      <telephony>
        <services>
          <!-- emails 0 -->
        </services>
        <number type="intern" quickdial="" vanity="" prio="0">**9</number>
      </telephony>
      <uniqueid>1232</uniqueid>
    </contact>
    <contact>
      <!-- numbers 1 -->
      <category/>
      <person>
        <realName>FRITZ!App Fon Desire HD</realName>
      </person>
      <telephony>
        <services>
          <!-- emails 0 -->
        </services>
        <number type="intern" quickdial="" vanity="" prio="0">**621</number>
      </telephony>
      <uniqueid>1235</uniqueid>
    </contact>
    <contact>
      <!-- numbers 1 -->
      <category/>
      <person>
        <realName>SRE</realName>
        <imageUrl>/download.lua?path=/var/media/ftp/USB/FRITZ/fonpix/123.jpg</imageUrl>
      </person>
      <telephony>
        <services>
          <!-- emails 0 -->
        </services>
        <number type="work" quickdial="1" vanity="" prio="0" >030399760</number>
      </telephony>
      <uniqueid>1234</uniqueid>
    </contact>
  </phonebook>
</phonebooks>
```

Tag	Type	Description
phonebooks	Complex	Root element
phonebook	Complex	
timestamp	Integer	Last modification timestamp.
contact	Complex	
person	Complex	
realName	String	Name of Contact
imageURL	String	HTTP URL to image for this contact
telephony	Complex	
services	Complex	
email	String	Email address
number	String	Phone number
uniqueid	Integer	Unique ID for a single contact (new since 2013-04-20)

Table 68: Phonebook content description

5.1.1.1 URLs

A telephone book may contain URLs with an image for the contact. The content can be downloaded using the protocol, hostname and port with the image URL.

An example is described here:

Protocol: https

Hostname: fritz.box

Port: 49443

image URL:

/download.lua?path=/var/media/ftp/JetFlash-Transcend4GB-01/FRITZ/fonpix/1316705057-0.jpg

The combination of

Protocoll + :// + Hostname + : + Port + image URL

will be the complete URL

https://fritz.box:49443/download.lua?path=/var/media/ftp/JetFlash-Transcend4GB-01/FRITZ/fonpix/1316705057-0.jpg

Please note, that this URL might require authentication.

5.2 Call List Content

The following shows an example XML content for a call list.

To differ between voice calls, fax calls and TAM calls use the Port value.

E.g. if port equals 5 it is a fax call. If port equals 6 or port in in the rage of 40 to 49 it is a TAM call.

```
<?xml version="1.0"?>
<root>
  <timestamp>123456</timestamp>
  <Call>
    <Id>123</Id>
    <Type>3</Type>
    <Called>0123456789</Called>
    <Caller>SIP: 98765</Caller>
    <CallerNumber>98765</CallerNumber>
    <Name>Max Mustermann</Name>
    <Numbertype/>
    <Device>Mobilteil 1</Device>
    <Port>10</Port>
    <Date>23.09.11 08:13</Date>
    <Duration>0:01</Duration>
    <Count/>
    <Path/>
  </Call>
  <Call>
    <Id>122</Id>
    <Type>1</Type>
    <Caller>012456789</Caller>
    <Called>SIP: 56789</Called>
    <CalledNumber>98765</CalledNumber>
    <Name>Max Mustermann</Name>
    <Numbertype/>
    <Device>Anrufbeantworter 1</Device>
    <Port>40</Port>
    <Date>22.09.11 14:19</Date>
    <Duration>0:01</Duration>
    <Count/>
    <Path>/download.lua?path=/var/media/ftp/USB/FRITZ/voicebox/rec/rec.0.000</Path>
  </Call>
</root>
```

Tag	Type	Description
timestamp	Integer	Timestamp of call list creation (unique ID per call list).
Call	Complex	
Id	Integer	Unique ID per call.
Type	Integer	Type of call. For more information see Table 70: Different call types within the callist and their meaning.
Called	String	Number or name of called party
Caller	String	Number of calling party
CallerNumber/ CalledNumber	String	Number of called/ called party (outgoing/ incoming call)
Name	String	Name of called/ called party (outgoing/ incoming call)
Numbertype	String	pots, isdn, sip, umts, "
Device	String	Name of used telephone port.
Port	String	Number of telephone port.
Date	Date-String	31.07.12 12:03
Duration	String	hh:mm (minutes rounded up)
Path	String	URL path to TAM or FAX file.

Table 69: Call list content description

5.2.1.1 URLs

A call list may contain URLs for telephone answering machine messages or fax messages. The content can be downloaded using the protocol, hostname and port with the path URL. An example is described here:

Protocol: https

Hostname: fritz.box

Port: 49443

path URL: /download.lua?path=/var/media/ftp/USB/FRITZ/voicebox/rec/rec.0.000

The combination of

Protocoll + :// + Hostname + : + Port + path URL

will be the complete URL

https://fritz.box:49443/download.lua?path=/var/media/ftp/USB/FRITZ/voicebox/rec/rec.0.000

Please note, that this URL might require authentication.

5.2.1.2 Types

Type value	Meaning	Description
1	incoming	Call answered by phone or answering machine.
2	missed	Incoming call was not answered by internal phone or answering machine.
3	outgoing	Finished call to external number.
9	active incoming	Phone or answering machine has answered the incoming call and the call isn't over yet.
10	rejected incoming	The incoming call was refused eg. by call barring.
11	active outgoing	Call to external number isn't over yet.

Table 70: Different call types within the callist and their meaning.

5.3 DeflectionList Example

```
<List>
  <Item>
    <DeflectionId>0</DeflectionId>
    <Enable>1</Enable>
    <Type>fromNumber</Type>
    <Number>12345</Number>
    <DeflectionToNumber>987456</DeflectionToNumber>
    <Mode>eImmediately</Mode>
    <Outgoing>0</Outgoing>
    <PhonebookID></PhonebookID>
  </Item>
  <Item>
    <DeflectionId>1</DeflectionId>
    <Enable>1</Enable>
    <Type>fromAll</Type>
    <Number></Number>
    <DeflectionToNumber>1234679</DeflectionToNumber>
    <Mode>eImmediately</Mode>
    <Outgoing></Outgoing>
    <PhonebookID></PhonebookID></>
  </Item>
</List>
```