

TR-064 Support – X_AVM-DE_OnTel

Supported by AVM

Author: AVM GmbH

Date: 2016-09-27

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 35: Call list content description and Table 10: 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.

1.2 Content

Content

1.1 History	1
1.2 Content	2
2 Action List	3
2.1 GetInfo	3
2.2 SetEnable	3
2.3 SetConfig	3
2.4 GetInfoByIndex	3
2.5 SetEnableByIndex	4
2.6 SetConfigByIndex	4
2.7 GetNumberOfEntries	4
2.8 DeleteByIndex	5
2.9 GetCallList	5
2.10 GetPhonebookList	6
2.11 GetPhonebook	6
2.12 AddPhonebook	6
2.13 DeletePhonebook	6
2.14 GetPhonebookEntry	7
2.15 SetPhonebookEntry	7
2.16 DeletePhonebookEntry	7
2.17 GetDECTHandsetList	7
2.18 GetDECTHandsetInfo	7
2.19 SetDECTHandsetPhonebook	7
2.20 GetNumberOfDeflections.....	8
2.21 GetDeflection	8
2.22 GetDeflections.....	9
2.23 SetDeflectionEnable.....	9
3 Parameters	10
3.1 ModeEnum.....	10
3.2 TypeEnum.....	10
4 Service States Table	11
5 XML Document Contents	12
5.1 Phonebook Content	12
5.1.1.1 URLs	13
5.2 Call List Content	13
5.2.1.1 URLs	15
5.3 DeflectionList Example	16

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

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 2: Argument list 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 3: Argument list of action SetConfig

2.4 GetInfoByIndex

Argument name	Direction	Related state variable	Remarks
NewIndex	in	Index	

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 4: Argument list 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 5: Argument list 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 6: Argument list of action *SetConfigByIndex*

2.7 GetNumberOfEntries

Argument name	Direction	Related state variable	Remarks
NewOntelNumberOfEntries	out	OntelNumberOfEntries	

Table 7: Argument list of action *GetNumberOfEntries*

2.8 DeleteByIndex

Argument name	Direction	Related state variable	Remarks
NewIndex	in	Index	

Table 8: Argument list 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 9: 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 10: 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 11: Return codes of action GetCallList

2.10 GetPhonebookList

Argument name	Direction	Related state variable	Remarks
NewPhonebookList	out	PhonebookList	

Table 12: Argument list 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 13: 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 14: URL parameters for PhonebookURL

2.12 AddPhonebook

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

Table 15: Argument list of action AddPhonebook

2.13 DeletePhonebook

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

Table 16: Argument list of action DeletePhonebook

2.14 GetPhonebookEntry

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

Table 17: Argument list of action GetPhonebookEntry

2.15 SetPhonebookEntry

Add new entries with "" as value for PhonebookEntryID.

Change existing entries with a value used for PhonebookEntryID with GetPhonebookEntry. The variable PhonebookEntryData may contain a unique ID.

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

Table 18: Argument list of action SetPhonebookEntry

2.16 DeletePhonebookEntry

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

Table 19: Argument list of action DeletePhonebookEntry

2.17 GetDECTHandsetList

Argument name	Direction	Related state variable	Remarks
NewDectIDList	out	DectIDList	

Table 20: Argument list of action GetDECTHandsetList

2.18 GetDECTHandsetInfo

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

Table 21: Argument list of action GetDECTHandsetInfo

2.19 SetDECTHandsetPhonebook

Argument name	Direction	Related state variable	Remarks
NewDectID	in	DectID	

Argument name	Direction	Related state variable	Remarks
NewPhonebookID	in	PhonebookID	

Table 22: Argument list of action *SetDECTHandsetPhonebook*

2.20 GetNumberOfDeflections

Get the number of deflection entries.

Argument name	Direction	Related state variable	Remarks
NewNumberOfDeflections	out	NumberOfDeflections	

Table 23: Argument list of action *GetNumberOfDeflections*

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

Table 24: Return codes of action *GetNumberOfDeflections*

2.21 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 25: 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 26: Return codes of action *GetDeflection*

2.22 GetDeflections

Returns a list of deflections.

Argument name	Direction	Related state variable	Remarks
NewDeflectionList	out	DeflectionList	

Table 27: Argument list of action GetDeflections

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

Table 28: Return codes of action GetDeflections

2.23 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 29: 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 30: Return codes of action SetDeflectionEnable

3 Parameters

3.1 ModeEnum

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

Table 31: Parameter type ModeEnum

3.2 TypeEnum

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

Table 32: 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
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 33: 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/JetFlash-Transcend4GB-
01/FRITZ/fonpix/1316705057-0.jpg</imageUrl>
      </person>
      <telephony>
        <services>
          <!-- emails 0 -->
        </services>
        <number type="work" quickdial="1" vanity="" prio="0"
>03039976794</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 34: 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://frtiz.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 the range 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>
  </Call>
</root>
```

```
<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/JetFlash-Transcend4GB-
01/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	1 incoming, 2 missed, 3 outgoing, 9 active incoming, 10 rejected incoming, 11 active outgoing
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 35: 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/JetFlash-Transcend4GB-01/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/JetFlash-Transcend4GB-01/FRITZ/voicebox/rec/rec.0.000`

Please note, that this URL might require authentication.

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>
```