

TR-064 Support – WLANConfiguration

Supported by AVM

Author: AVM GmbH

Date: 2022-10-13

1 urn:WLANConfiguration-com:serviceId:WLANConfiguration1

For details please refer the TR-064 document at

<https://www.broadband-forum.org/technical/trlist.php>.

For the support of device with more than one access point more than one WLANConfiguration services are listed in the TR-064 service description. If the device supports WLAN one service is listed (service #1). If the device additionally supports a second physical access point support 2.4 GHz and 5 GHz, one more service is listed (service #2). If a third physical access point is supported, an additional service is listed (service #3). If the device supports a logical Access Point for guests, one more service is listed (service #2, #3 or #4).

1.1 History

Date	Version	Changes
2019-09-25	31	Added value "ax" to variable Standard
2019-09-26	32	Added value "WPA3" to variable BeaconType
2019-11-13	33	Remove WPS PIN method
2020-01-09	34	Added new Action X_AVM-DE_GetWLANConnectionInfo and new state variables X_AVM-DE_SpeedRX, X_AVM-DE_SpeedMax, X_AVM-DE_SpeedRXMax
2020-03-17	35	Changed allowed values for Variable "BeaconType": - Added new value "OWETrans" (only accepted on hardware which supports OWE) - remove value "WPA" and "WPA3"
2020-10-13	36	Changed allowed values for Variable "Status": - Added new value "UP" and "Disabled" Changed allowed values for Variable "BeaconType": - Added new value "Basic", "WPA", "WPA3" and "OWE"
2020-11-02	37	Action added: X_AVM-DE_SetWPSEnable
2021-08-05	38	Added parameter and state variable X_AVM-DE_FrequencyBand for action GetInfo, GetChannelInfo and X_AVM-DE_GetWLANExtInfo
2021-10-06	39	Added parameter and state variable X_AVM-DE_ChannelWidth for actions GetSpecificAssociatedDeviceInfo, GetGenericAssociatedDeviceInfo and GetSpecificAssociatedDeviceInfoByIP Added X_AVM-DE-ChannelWidth to DeviceList given through X_AVM-DE_GetWLANDeviceListPath
2021-10-19	40	Added state variable X_AVM-DE_ChannelWidth for action X_AVM-DE_GetWLANConnectionInfo
2021-11-15	41	Added state variable X_AVM-DE_AutoChannelEnabled for actions GetChannelInfo and X_AVM-DE_GetWLANConnectionInfo Added state variable X_AVM-DE_FrequencyBand for action X_AVM-DE_GetWLANConnectionInfo
2022-02-15	42	Added required rights for actions
2022-08-29	43	Added state variable X_AVM-DE_WLANGlobalEnable for action GetInfo Added description for GetInfo and X_AVM-DE_SetWLANGlobalEnable
2022-10-13	44	Mark action SetDefaultWEPKeyIndex and GetDefaultWEPKeyIndex as removed

Table of Content

1	urn:WLANConfiguration-com:serviceId:WLANConfiguration1.....	1
1.1	History.....	2
2	Action List.....	4
2.1	SetEnable.....	4
2.2	GetInfo.....	4
2.3	SetConfig.....	5
2.4	SetSecurityKeys.....	5
2.5	GetSecurityKeys.....	5
2.6	SetDefaultWEPKeyIndex.....	6
2.7	GetDefaultWEPKeyIndex.....	6
2.8	SetBasBeaconSecurityProperties.....	6
2.9	GetBasBeaconSecurityProperties.....	6
2.10	GetBSSID.....	6
2.11	GetSSID.....	6
2.12	SetSSID.....	7
2.13	GetBeaconType.....	7
2.14	SetBeaconType.....	7
2.15	GetChannelInfo.....	7
2.16	SetChannel.....	7
2.17	GetBeaconAdvertisement.....	7
2.18	SetBeaconAdvertisement.....	7
2.19	GetTotalAssociations.....	8
2.20	GetGenericAssociatedDeviceInfo.....	8
2.21	GetSpecificAssociatedDeviceInfo.....	8
2.22	X_AVM-DE_GetSpecificAssociatedDeviceInfoByIp.....	8
2.23	X_AVM-DE_GetWLANDeviceListPath.....	9
2.24	X_AVM-DE_SetStickSurfEnable.....	9
2.25	X_AVM-DE_GetIPTVOptimized.....	9
2.26	X_AVM-DE_SetIPTVOptimized.....	10
2.27	GetStatistics.....	10
2.28	GetPacketStatistics.....	10
2.29	X_AVM-DE_GetNightControl.....	10
2.30	X_SetHighFrequencyBand.....	10
2.31	X_AVM-DE_GetWLANHybridMode.....	11
2.32	X_AVM-DE_SetWLANHybridMode.....	11
2.33	X_AVM-DE_GetWLANExtInfo.....	12
2.34	X_AVM-DE_SetWLANGlobalEnable.....	12
2.35	X_AVM-DE_GetWPSInfo.....	12
2.36	X_AVM-DE_SetWPSConfig.....	13
2.37	X_AVM-DE_SetWPSEnable.....	13
2.38	X_AVM-DE_GetWLANConnectionInfo.....	13
3	Service States Table.....	15

2 Action List

This chapter contains the supported actions of the service WLAN configuration which are listed incl. arguments.

2.1 SetEnable

Required rights: App

Argument name	Direction	Related state variable	Remarks
NewEnable	in	Enable	

Table 1: Argument list of action SetEnable

2.2 GetInfo

Returns some basic status information about the request WLAN Access Point.

Required rights: App or Phone

Argument name	Direction	Related state variable	Remarks
NewEnable	out	Enable	
NewStatus	out	Status	
NewMaxBitRate	out	MaxBitRate	Not supported. Returns default string.
NewChannel	out	Channel	
NewSSID	out	SSID	
NewBeaconType	out	BeaconType	
NewX_AVM-DE_PossibleBeaconTypes	out	X_AVM-DE_PossibleBeaconTypes	
NewMACAddressControlEnabled	out	MACAddressControlEnabled	
NewStandard	out	Standard	Only the highest of the active modes is returned.
NewBSSID	out	BSSID	
NewBasicEncryptionModes	out	BasicEncryptionModes	
NewBasicAuthenticationMode	out	BasicAuthenticationMode	Returns fixed string "none".
NewMaxCharsSSID	out	MaxCharsSSID	
NewMinCharsSSID	out	MinCharsSSID	
NewAllowedCharsSSID	out	AllowedCharsSSID	
NewMinCharsPSK	out	MinCharsPSK	
NewMaxCharsPSK	out	MaxCharsPSK	
NewAllowedCharsPSK	out	AllowedCharsPSK	
NewX_AVM-DE_FrequencyBand	out	X_AVM-DE_FrequencyBand	
NewX_AVM-DE_WLANGlobalEnable	out	X_AVM-DE_WLANGlobalEnable	

Table 2: Argument list of action GetInfo

2.3 SetConfig

Argument name	Direction	Related state variable	Remarks
NewMaxBitRate	in	MaxBitRate	
NewChannel	in	Channel	
NewSSID	in	SSID	
NewBeaconType	in	BeaconType	Determines WLAN-encryption to be used.
NewMacAddressControlEnabled	in	MacAddressControlEnabled	
NewBasicEncryptionModes	in	BasicEncryptionModes	
NewBasicAuthenticationMode	in	BasicAuthenticationMode	

Table 3: Argument list of action SetConfig

2.4 SetSecurityKeys

Since WEP is deprecated, the WEPKey variables are ignored.

Required rights: App

Argument name	Direction	Related state variable	Remarks
NewWEPKey0	in	WEPKey0	
NewWEPKey1	in	WEPKey1	
NewWEPKey2	in	WEPKey2	
NewWEPKey3	in	WEPKey3	
NewPreSharedKey	in	PreSharedKey	
NewKeyPassphrase	in	KeyPassphrase	

Table 4: Argument list of action SetSecurityKeys

2.5 GetSecurityKeys

Since WEP is deprecated, the WEPKey variables are always empty.

Required rights: App

Argument name	Direction	Related state variable	Remarks
NewWEPKey0	out	WEPKey0	
NewWEPKey1	out	WEPKey1	
NewWEPKey2	out	WEPKey2	
NewWEPKey3	out	WEPKey3	
NewPreSharedKey	out	PreSharedKey	
NewKeyPassphrase	out	KeyPassphrase	

Table 5: Argument list of action GetSecurityKeys

2.6 SetDefaultWEPKeyIndex (removed)

Since WEP is deprecated, this action always returns 401 (invalid action).

Argument name	Direction	Related state variable	Remarks
NewDefaultWEPKeyIndex	in	WEPKeyIndex	

Table 6: Argument list of action SetDefaultWEPKeyIndex

2.7 GetDefaultWEPKeyIndex (removed)

Since WEP is deprecated, this action always returns 401 (invalid action).

Argument name	Direction	Related state variable	Remarks
NewDefaultWEPKeyIndex	out	WEPKeyIndex	

Table 7: Argument list of action GetDefaultWEPKeyIndex

2.8 SetBasBeaconSecurityProperties

The action is deprecated, this action always returns 401 (invalid action).

Argument name	Direction	Related state variable	Remarks
NewBasicEncryptionModes	in	BasicEncryptionModes	
NewBasicAuthenticationMode	in	BasicAuthenticationMode	

Table 8: Argument list of action SetBasBeaconSecurityProperties

2.9 GetBasBeaconSecurityProperties

Argument name	Direction	Related state variable	Remarks
NewBasicEncryptionModes	out	BasicEncryptionModes	
NewBasicAuthenticationMode	out	BasicAuthenticationMode	

Table 9: Argument list of action GetBasBeaconSecurityProperties

2.10 GetBSSID

Argument name	Direction	Related state variable	Remarks
NewBSSID	out	BSSID	

Table 10: Argument list of action GetBSSID

2.11 GetSSID

Required rights: App

Argument name	Direction	Related state variable	Remarks
NewSSID	out	SSID	

Table 11: Argument list of action GetSSID

2.12 SetSSID

Argument name	Direction	Related state variable	Remarks
NewSSID	in	SSID	

Table 12: Argument list of action SetSSID

2.13 GetBeaconType

Required rights: App

Argument name	Direction	Related state variable	Remarks
NewBeaconType	out	BeaconType	
NewX_AVM-DE_PossibleBeaconTypes	out	X_AVM-DE_PossibleBeaconTypes	

Table 13: Argument list of action GetBeaconType

2.14 SetBeaconType

Argument name	Direction	Related state variable	Remarks
NewBeaconType	in	BeaconType	

Table 14: Argument list of action SetBeaconType

2.15 GetChannelInfo

Argument name	Direction	Related state variable	Remarks
NewChannel	out	Channel	
NewPossibleChannels	out	PossibleChannels	
NewX_AVM-DEAuto_ChannelEnabled	out	X_AVM-DE_AutoChannelEnabled	
NewX_AVM-DE_FrequencyBand	out	X_AVM-DE_FrequencyBand	

Table 15: Argument list of action GetChannelInfo

2.16 SetChannel

Argument name	Direction	Related state variable	Remarks
NewChannel	in	Channel	

Table 16: Argument list of action SetChannel

2.17 GetBeaconAdvertisement

Argument name	Direction	Related state variable	Remarks
NewBeaconAdvertisementEnabled	out	BeaconAdvertisementEnabled	

Table 17: Argument list of action GetBeaconAdvertisement

2.18 SetBeaconAdvertisement

Argument name	Direction	Related state variable	Remarks
NewBeaconAdvertisementEnabled	in	BeaconAdvertisementEnabled	

Table 18: Argument list of action SetBeaconAdvertisement

2.19 GetTotalAssociations

Required rights: App or Phone

Argument name	Direction	Related state variable	Remarks
NewTotalAssociations	out	TotalAssociations	

Table 19: Argument list of action GetTotalAssociations

2.20 GetGenericAssociatedDeviceInfo

Required rights: App or Phone

Argument name	Direction	Related state variable	Remarks
NewAssociatedDeviceIndex	in	TotalAssociations	
NewAssociatedDeviceMACAddress	out	AssociatedDeviceMACAddress	
NewAssociatedDeviceIPAddress	out	AssociatedDeviceIPAddress	
NewAssociatedDeviceAuthState	out	AssociatedDeviceAuthState	
NewX_AVM-DE_Speed	out	X_AVM-DE_Speed	22.01.10
NewX_AVM-DE_SignalStrength	out	X_AVM-DE_SignalStrength	22.01.10
NewX_AVM-DE_ChannelWidth	out	X_AVM-DE_ChannelWidth	

Table 20: Argument list of action GetGenericAssociatedDeviceInfo

2.21 GetSpecificAssociatedDeviceInfo

Required rights: App or Phone

Argument name	Direction	Related state variable	Remarks
NewAssociatedDeviceMACAddress	in	AssociatedDeviceMACAddress	
NewAssociatedDeviceIPAddress	out	AssociatedDeviceIPAddress	
NewAssociatedDeviceAuthState	out	AssociatedDeviceAuthState	
NewX_AVM-DE_Speed	out	X_AVM-DE_Speed	19.09.11
NewX_AVM-DE_SignalStrength	out	X_AVM-DE_SignalStrength	19.09.11
NewX_AVM-DE_ChannelWidth	out	X_AVM-DE_ChannelWidth	

Table 21: Argument list of action GetSpecificAssociatedDeviceInfo

2.22 X_AVM-DE_GetSpecificAssociatedDeviceInfoByIp

Required rights: App or Phone

Argument name	Direction	Related state variable	Remarks
NewAssociatedDeviceIPAddress	in	AssociatedDeviceIPAddress	
NewAssociatedDeviceMACAddress	out	AssociatedDeviceMACAddress	
NewAssociatedDeviceAuthState	out	AssociatedDeviceAuthState	
NewX_AVM-DE_Speed	out	X_AVM-DE_Speed	
NewX_AVM-DE_SignalStrength	out	X_AVM-DE_SignalStrength	
NewX_AVM-DE_ChannelWidth	out	X_AVM-DE_ChannelWidth	

Table 22: Argument list of action GetSpecificAssociatedDeviceInfoByIp

Return code	Description	Related argument
714	NoSuchEntryInArray	AssociatedDeviceIPAddress
402	Invalid Args	AssociatedDeviceIPAddress

Table 23: Return codes of action GetSpecificAssociatedDeviceInfoByIp

2.23 X_AVM-DE_GetWLANDeviceListPath

Returns an URL path to fetch a WIFI device list of all connected devices as an XML document.

Required rights: App or Phone

Argument name	Direction	Related state variable	Remarks
NewX_AVM-DE_WLANDeviceListPath	out	X_AVM-DE_WLANDeviceListPath	

Table 24: Argument list of action X_AVM-DE_GetWLANDeviceListPath

The following listing is an XML example of the generated list:

```
<?xml version="1.0" encoding="UTF-8"?>
<List>
  <TotalAssociations>13</TotalAssociations>
  <Item>
    <AssociatedDeviceIndex>1</AssociatedDeviceIndex>
    <AssociatedDeviceMACAddress>aa:bb:cc:11:11:11</AssociatedDeviceMACAddress>
    <AssociatedDeviceIPAddress>192.168.192.11</AssociatedDeviceIPAddress>
    <AssociatedDeviceAuthState>1</AssociatedDeviceAuthState>
    <X_AVM-DE_Speed>866</X_AVM-DE_Speed>
    <X_AVM-DE_SignalStrength>98</X_AVM-DE_SignalStrength>
    <AssociatedDeviceChannel>1</AssociatedDeviceChannel>
    <X_AVM-DE_ChannelWidth>0</X_AVM-DE_ChannelWidth>
    <AssociatedDeviceGuest>0</AssociatedDeviceGuest>
  </Item>
  <Item>
    ...
  </Item>
</List>
```

Devices with AssociatedDeviceChannel set to 0 are not listed in this list, because these are not connected to any access point of this CPE.

The “TotalAssociations” list element shows the number of all connected WIFI devices

which are listed in the file independent from the current access point or WLANConfiguration service.

Return code	Description	Related argument
501	Action Failed	
402	Invalid Args	

Table 25: Return codes of action X_AVM-DE_GetWLANDeviceListPath

2.24 X_AVM-DE_SetStickSurfEnable

Argument name	Direction	Related state variable	Remarks
NewStickSurfEnable	in	StickSurfEnable	

Table 26: Argument list of action X_AVM-DE_SetStickSurfEnable

2.25 X_AVM-DE_GetIPTVOptimized

Argument name	Direction	Related state variable	Remarks
NewX_AVM-DE_IPTVoptimize	out	X_AVM-DE_IPTVoptimize	

Table 27: Argument list of action X_AVM-DE_GetIPTVOptimized

2.26 X_AVM-DE_SetIPTVOptimized

Argument name	Direction	Related state variable	Remarks
NewX_AVM-DE_IPTVoptimize	in	X_AVM-DE_IPTVoptimize	

Table 28: Argument list of action X_AVM-DE_SetIPTVOptimized

2.27 GetStatistics

Argument name	Direction	Related state variable	Remarks
NewTotalPacketsSent	out	TotalPacketsSent	
NewTotalPacketsReceived	out	TotalPacketsReceived	

Table 29: Argument list of action GetStatistics

2.28 GetPacketStatistics

Argument name	Direction	Related state variable	Remarks
NewTotalPacketsSent	out	TotalPacketsSent	
NewTotalPacketsReceived	out	TotalPacketsReceived	

Table 30: Argument list of action GetPacketStatistics

2.29 X_AVM-DE_GetNightControl

Required rights: None

Argument name	Direction	Related state variable	Remarks
NewNightControl	out	NightControl	
NewNightTimeControlNoForcedOff	out	NightTimeControlNoForcedOff	

Table 31: Argument list of action X_AVM-DE_GetNightControl

2.30 X_SetHighFrequencyBand

The action is listed in the WLAN configuration service but only supported on certain OEM devices. Please see WLAN configuration service 2 on FRITZ!Box 7390 for reading and writing 5 GHz WLAN settings.

Argument name	Direction	Related state variable	Remarks
NewEnableHighFrequency	in	EnableHighFrequency	

Table 32: Argument list of action X_SetHighFrequencyBand

2.31 X_AVM-DE_GetWLANHybridMode

This action delivers informations about the WLAN-Mode access.

Argument name	Direction	Related state variable	Remarks
NewEnable	out	Enable	
NewBeaconType	out	BeaconType	
NewKeyPassphrase	out	KeyPassphrase	
NewSSID	out	SSID	
NewBSSID	out	BSSID	
NewTrafficMode	out	TrafficMode	
NewManualSpeed	out	ManualSpeed	
NewMaxSpeedDS	out	MaxSpeedDS	
NewMaxSpeedUS	out	MaxSpeedUS	

Table 33: Argument list of action X_AVM-DE_GetWLANHybridMode

2.32 X_AVM-DE_SetWLANHybridMode

This action delivers informations about the WLAN-Mode access.

Argument name	Direction	Related state variable	Remarks
NewEnable	in	Enable	
NewBeaconType	in	BeaconType	
NewKeyPassphrase	in	KeyPassphrase	
NewSSID	in	SSID	
NewBSSID	in	BSSID	

Argument name	Direction	Related state variable	Remarks
NewTrafficMode	in	TrafficMode	
NewManualSpeed	in	ManualSpeed	
NewMaxSpeedDS	in	MaxSpeedDS	
NewMaxSpeedUS	in	MaxSpeedUS	

Table 34: Argument list of action X_AVM-DE_SetWLANHybridMode

2.33 X_AVM-DE_GetWLANExtInfo

This action delivers informations about the WLAN-Guest access.

Required rights: App

Argument name	Direction	Related state variable	Remarks
NewX_AVM-DE_APEEnabled	out	X_AVM-DE_APEEnabled	
NewX_AVM-DE_APTYPE	out	X_AVM-DE_APTYPE	
NewX_AVM-DE_FrequencyBand	out	X_AVM-DE_FrequencyBand	
NewX_AVM-DE_TimeoutActive	out	X_AVM-DE_TimeoutActive	
NewX_AVM-DE_Timeout	out	X_AVM-DE_Timeout	
NewX_AVM-DE_TimeRemain	out	X_AVM-DE_TimeRemain	
NewX_AVM-DE_NoForcedOff	out	X_AVM-DE_NoForcedOff	
NewX_AVM-DE_UserIsolation	out	X_AVM-DE_UserIsolation	
NewX_AVM-DE_EncryptionMode	out	X_AVM-DE_EncryptionMode	
NewX_AVM-DE_LastChangedStamp	out	X_AVM-DE_LastChangedStamp	

Table 35: Argument list of action X_AVM-DE_GetWLANExtInfo

2.34 X_AVM-DE_SetWLANGlobalEnable

Enables “1” or disables “0” WLAN as if pushing the “WLAN” button on the CPE device. Only the last time active Access Points will be enabled. The global active status can be checked with GetInfo X_AVM-DE_WLANGlobalEnable.

Required rights: App

Argument name	Direction	Related state variable	Remarks
NewX_AVM-DE_WLANGlobalEnable	in	X_AVM-DE_WLANGlobalEnable	

Table 36: Argument list of action X_AVM-DE_SetWLANGlobalEnable

Code	Description	Related argument
402	Invalid Arguments	
801	String argument too short	
802	String argument too long	
803	Argument contains invalid characters	
820	Internal Error	

Table 37: Return codes of action X_AVM-DE_SetWLANGlobalEnable

2.35 X_AVM-DE_GetWPSInfo

Required rights: App

Argument name	Direction	Related state variable	Remarks
NewX_AVM-DE_WPSMode	out	X_AVM-DE_WPSMode	
NewX_AVM-DE_WPSStatus	out	X_AVM-DE_WPSStatus	

Table 38: Argument list of action X_AVM-DE_GetWPSInfo

2.36 X_AVM-DE_SetWPSConfig

WPS is supported with Push Button Configuration.

WPS is only supported for either access points or guest access points.

WPS is not supported for both types of access points in parallel.

Required rights: App

Argument name	Direction	Related state variable	Remarks
NewX_AVM-DE_WPSMode	in	X_AVM-DE_WPSMode	
NewX_AVM-DE_WPSStatus	out	X_AVM-DE_WPSStatus	

Table 39: Argument list of action X_AVM-DE_SetWPSConfig

WPS Mode	NewX_AVM-DE_WPSMode
Push Button Configuration	pbcc
Stop running WPS session	stop
WPS disabled or unknown WPS mode	other

Table 40: Example WPS configuration

2.37 X_AVM-DE_SetWPSEnable

Enables “1” or disables “0” WPS.

If X_AVM-DE_WPSEnable is “1”, you may start the WPS mechanism with 2.36 X_AVM-DE_SetWPSConfig .

Required rights: App

Argument name	Direction	Related state variable	Remarks
NewX_AVM-DE_WPSEnable	in	X_AVM-DE_WPSEnable	

Table 41: Argument list of action X_AVM-DE_SetWPSEnable

2.38 X_AVM-DE_GetWLANConnectionInfo

Get information about the Wi-Fi connection of the requesting Wi-Fi host device. If the device is not directly connected to any of the access points, the status code 714 (no such entry in array) will be returned.

Required rights: None

Argument name	Direction	Related state variable	Remarks
NewAssociatedDeviceMACAddress	out	AssociatedDeviceMACAddress	
NewSSID	out	SSID	
NewBSSID	out	BSSID	
NewBeaconType	out	BeaconType	
NewChannel	out	Channel	
NewStandard	out	Standard	
NewX_AVM-DE_AutoChannelEnabled	out	X_AVM-DE_AutoChannelEnabled	
NewX_AVM-DE_ChannelWidth	out	X_AVM-DE_ChannelWidth	
NewX_AVM-DE_FrequencyBand	out	X_AVM-DE_FrequencyBand	
NewX_AVM-DE_SignalStrength	out	X_AVM-DE_SignalStrength	
NewX_AVM-DE_Speed	out	X_AVM-DE_Speed	
NewX_AVM-DE_SpeedRX	out	X_AVM-DE_SpeedRX	
NewX_AVM-DE_SpeedMax	out	X_AVM-DE_SpeedMax	
NewX_AVM-DE_SpeedRXMax	out	X_AVM-DE_SpeedRXMax	

Table 42: Argument list of action X_AVM-DE_GetWLANConnectionInfo

Code	Description	Related argument
402	Invalid Arguments	
714	No such entry in array	
820	Internal Error	

Table 43: Return codes of action X_AVM-DE_GetWLANConnectionInfo

3 Service States Table

Variable name	Allowed values (* == default)	Send Events	Data type
AllowedChars KeyPassphrase	01234567890ABCDEFGHIJKLMNPOQRSTUVWXYZ_!\"#\$%&'()*+,-./:;<=>?@[\\]^_`{ }~(*)	no	String
AllowedCharsPSK	01234567890ABCDEFabcdef (*)	no	String
AllowedCharsSSID	01234567890ABCDEFGHIJKLMNPOQRSTUVWXYZ_!\"#\$%&'()*+,-./:;<=>?@[\\]^_`{ }~(*)	no	String
AssociatedDeviceAuthState	0 (*)	no	Boolean
AssociatedDeviceChannel		no	ui4
AssociatedDeviceGuest	0 (*)	no	Boolean
AssociatedDeviceIPAddress		no	String
AssociatedDeviceMACAddress	Lower case MAC address	no	String
X_AVM-DE_AutoChannelEnabled		no	Boolean
BasicAuthenticationMode	None	no	String
BasicEncryptionModes	None	no	String
BeaconAdvertisementEnabled		no	Boolean
BeaconType	None, Basic, WPA, 11i, WPAand11i, WPA3, 11iandWPA3, OWE, OWETrans	no	String
BSSID		no	String
Channel	0 means auto channel	no	ui1
Enable		no	Boolean
EnableHighFrequency		no	Boolean
KeyPassphrase		no	String
MACAddressControlEnabled		no	Boolean
ManualSpeed		no	Boolean
MaxBitRate	Auto	no	String
MaxCharsKeyPassphrase	63 (*)	no	ui1
MaxCharsPSK	64 (*)	no	ui1
MaxCharsSSID	32 (*)	no	ui1
MaxSpeedDS		no	ui4
MaxSpeedUS		no	ui4
MinCharsKeyPassphrase	8 (*)	no	ui1
MinCharsPSK	64 (*)	no	ui1
MinCharsSSID	3 (*)	no	ui1
NightControl	XML-String	no	String
NightTimeControlNoForcedOff		no	Boolean
PossibleChannels	depends on currently used standard	no	String

Variable name	Allowed values (* == default)	Send Events	Data type
PreSharedKey		no	String
SSID		no	String
Standard	b, g, n, ac, ax, ""	no	String
Status	Up, Disabled	no	String
StickSurfEnable	0 (*)	no	Boolean
TrafficMode		no	String
TotalAssociations	0 (*)	no	ui2
TotalPacketsReceived		no	ui4
TotalPacketsSent		no	ui4
WEPKey0	obsolete	no	String
WEPKey1	obsolete	no	String
WEPKey2	obsolete	no	String
WEPKey3	obsolete	no	String
WEPKeyIndex	obsolete	no	ui1
X_AVM-DE_APEnabled		yes	String
X_AVM-DE_APTYPE	normal, guest	no	String
X_AVM-DE_ChannelWidth	0 (*), 20, 40, 80, 160, 320	no	ui2
X_AVM-DE_EncryptionMode		no	String
X_AVM-DE_FrequencyBand	unknown (*), 2400, 5000, 6000	no	String
X_AVM-DE_IPTVoptimize	0 (*)	no	Boolean
X_AVM-DE_LastChangedStamp	0 (*)	no	ui4
X_AVM-DE_NoForcedOff		no	String
X_AVM-DE_PossibleBeaconTypes	Comma separated string of possible BeaconTypes. e.g. "WPA,None,WPAand11i"	no	String
X_AVM-DE_SignalStrength	0 (*)	no	ui1
X_AVM-DE_Speed	0 (*)	no	ui2
X_AVM-DE_SpeedMax	0 (*)	no	ui4
X_AVM-DE_SpeedRX	0 (*)	no	ui4
X_AVM-DE_SpeedRXMax	0 (*)	no	ui4
X_AVM-DE_Timeout		no	String
X_AVM-DE_TimeoutActive		no	String
X_AVM-DE_TimeRemain		no	String
X_AVM-DE_UserIsolation		no	String
X_AVM-DE_WLANGlobalEnable	0	no	Boolean
X_AVM-DE_WPSAPPIN	8 digits	no	String
X_AVM-DE_WPSCClientPIN	8 digits	no	String

Variable name	Allowed values (* == default)	Send Events	Data type
X_AVM-DE_WPSEnable	0 (*)	no	Boolean
X_AVM-DE_WPSMode	pbc, stop, other	no	String
X_AVM-DE_WPSStatus	off, inactive, active, success, err_common, err_timeout, err_reconfig, err_internal, err_abort	no	String

Table 44: Variable list