VDV Die Verkehrsunternehmen

VDV-Schrift 301-2-0

01/2023

IBIS-IP Beschreibung der Dienste / Service description

DIENST- / SERVICE-DeviceManagementService
V2.4

Gesamtbearbeitung
Ausschuss für Telematik und Informationssysteme (ATI)

Gefördert durch:

Bundesministerium für Wirtschaft und Energie

aufgrund eines Beschlusses des Deutschen Bundestages

Das dieser VDV-Schrift zugrundeliegende Vorhaben IP-KOM-ÖV wurde mit Mitteln des Bundesministeriums für Wirtschaft und Energie unter dem Förderkennzeichen 19P10003 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren.

IBIS-IP Beschreibung der Dienste / Service description

DIENST- / SERVICE-DeviceManagementService V2.4

Sachbearbeitung
Unterausschuss für Telematik
(UA Telematik)

Autorenverzeichnis
Dipl.-Ing. Dirk Weißer, VDV, Köln
Dr. Torsten Franke, IVU, Aachen
Dipl.-Ing. Berthold Radermacher, VDV, Köln
Dipl.-Ing. René Fischli, Trapeze, Neuhausen
Dr. Bernd Schubert, iris-GmbH, Berlin

Vorwort

Im Forschungsprojekt „Internet Protokoll basierte Kommunikationsdienste im ÖV - IP-KOM-ÖV“, gefördert vom Bundesministerium für Wirtschaft und Energie BMWi, wurde das Grundkonzept von IBIS-IP für die Fahrgastinformation entwickelt.

In der VDV-301-2 werden die technischen Grundlagen wie auch die Basisdienste, welche die Grundlagen eines IBIS-IP-Systems bilden, beschrieben.

In dieser VDV-Schrift sind der HtmlDisplayService und seine spezifischen Datenstrukturen beschrieben.

Diese VDV-Schrift wird zweisprachig veröffentlicht. Dabei ist zu beachten, dass Erläuterungen in Deutsch und Englisch verfasst sind, während die technischen Operationen und zugehörigen Datenstrukturen, die sich an Softwareentwickler richten, nur in Englisch beschrieben sind.

Foreword

In the research project "Internet Protocol based communication services in public transport - IP-KOM-ÖV", funded by the Federal Ministry of Economics and Energy BMWi, the basic concept of IBIS-IP for passenger information was developed.

The technical basics as well as the basic services of the IBIS-IP systems are described in the VDV-301-2.

This VDV document describes the HtmlDisplayService and its specific data structures.

The HTMLDisplayService provides a URL to a web server for multifunction screens. When the screen is accessed via the URL, it receives the content to be displayed via HTML.

This VDV publication is published in two languages. It should be noted that explanations are written in German and English, while the technical operations and data structures related to software developers are described in English only.

Inhaltsverzeichnis / Content

Abkürzungen / Abbreviations

Die bereits in der VDV 301-1 definierten Abkürzungen werden an dieser Stelle nicht wiederholt.

The abbreviations already defined in VDV 301-1 are not repeated here.

1 Dienst DeviceManagementService

Aufgaben des Dienstes und die Nutzung

Der Dienst `DeviceManagementService` ist der Dienst, welcher die Aufgaben der Fachkomponente Geräte-Management innerhalb von IBIS-IP umsetzt.

Der Dienst ist für alle Daten als http-Dienst spezifiziert.

Dieser Dienst ist auf jedem an IBIS-IP teilnehmenden Gerät einmal vorhanden und wird mit dem Ausführen der Applikation auf dem Gerät automatisch gestartet.

1 DeviceManagementService

Tasks of the Service and its Usage

The `DeviceManagementService` is a service which is used to maintain and manage the plugged devices.

A DeviceManagementService has to run on each device participating in the IBIS-IP network.

The `DeviceManagementService` is designed as an http-service.

2 Operationen des DeviceManagementService

Die vom `DeviceManagementService` angebotenen Operationen lassen sich unterteilen in Operationen die das Gerät betreffen

— Statische Informationen über das Gerät abfragen

— Konfiguration des Geräts ändern

— Konfiguration des Geräts abfragen

— Gerät restarten

— Status des Geräts abfragen

— Fehlermeldungen abfragen

— Updates des Geräts anstoßen und überwachen

sowie Operationen, die die Dienste auf dem Gerät betreffen

— Statische Informationen über die Dienste

— Dienste starten/stoppen/restarten

— Zustand der Dienste abfragen

See Tabelle/Table 1

2.1 Aktualisierung von Firmware und Konfiguration

Eine Aktualisierung wird mit der Operationen InstallUpdate eingeleitet. Mit Hilfe dieser Operation kann ein Update-Controller einem IBIS-IP-Gerät eine Update-Datei übergeben, die dieses dann installiert. Der Speicherort der Update-Datei wird optional als URI übergeben. Falls kein Speicherort übergeben wird, muss das Gerät selbstständig in der Lage sein, die Update-Datei an einem zuvor konfigurierten Speicherort abzuholen. Das Gerät lädt die Datei herunter und prüft anschließend, ob diese sich für eine Aktualisierung eignet. Wenn das nicht der Fall ist, wird die Aktualisierung zurückgewiesen. Die Operation RetrieveUpdateState kann eingesetzt werden, um bei einer laufenden Aktualisierung deren aktuellen Status abzufragen. Nach dem Abschluss einer Aktualisierung ist diese mit der Operation FinalizeUpdate zu finalisieren.

Das IBIS-IP-Gerät speichert persistent eine Update-History. Diese kann mit Hilfe der Operation GetUpdateHistory abgefragt werden.

Abhängigkeiten von anderen Aktualisierungen können nicht verwaltet werden. Wenn unterschiedliche Komponenten eines IBIS-IP-Geräts zur selben Zeit passende Aktualisierungen erhalten müssen, sind diese in einer einzigen Update-Datei zusammenzufassen.

Der Update-Controller unterstützt keine Aktualisierungspläne.

Die richtige Zuordnung einer Update-Datei zu einem Gerät ist nicht die Aufgabe des Update-Controllers. Dafür kann ein Update-Manager eingesetzt werden, der den Update-Controller bei Bedarf aufruft. Der Update-Controller gibt dann das Ergebnis einer Aktualisierung an den Update-Manager weiter, welcher es entsprechend auswertet.

Der Update-Controller muss eine Software sein und ist für diese Aufgaben zuständig:

- Steuerung des Update-Prozesses beginnend mit der Operation InstallUpdate und endend mit der Operation FinalizeUpdate
- Übergabe des Ergebnisses eines Updates an den Update-Manager

Der Update-Manager kann eine Software sein oder eine Person. Er ist für diese Aufgaben zuständig:

- Auswahl der Update-Datei
- Planung von Aktualisierungen

Der Update-Controller startet das Gerät neu, wenn es den Aktualisierungsstatus *DeviceRestartRequired* zurückmeldet. Mindestens nach dem Abschluss einer Aktualisierung ist das betreffende Gerät neu zu starten. Neustarts im Verlauf des Update-Prozesses müssen wie erforderlich ausgeführt werden. Der initiale Aktualisierungsstatus nach einem solchen Neustart muss *UpdateRunning* sein.

Da während des Update-Vorgangs das Gerät unter Umständen das Gerät nicht in gewohnter Form zur Verfügung steht, sind entsprechend die Geräte-Status-Meldungen anzupassen.

2.1.1 Update-Prozess

1. Der Update-Controller führt die Operation *InstallUpdate* aus und setzt seine Tätigkeit fort, wenn das Gerät die Aktualisierung durch Rückgabe des Werts *UpdateAccepted* akzeptiert.
2. Mit Hilfe der Operation *RetrieveUpdateState* kann periodisch der Aktualisierungsstatus abgefragt werden. Das maximale Abfrageintervall beträgt 5s. Der minimale Wert ist 1s.
3. Das Gerät wird mittels der Operation *DeviceManagementService.RestartDevice* neu gestartet, wenn es den Aktualisierungsstatus *DeviceRestartRequired* zurückgibt.
4. Der Update-Controller wartet, während das Gerät herunterfährt. Der Standardwert der Wartezeit ist 10s.
5. Der Update-Controller prüft, ob das Gerät wieder mit dem IBIS-IP-Netzwerk verbunden ist und kehrt danach zur Abfrage des Aktualisierungsstatus zurück. Der Standardwert der Wartezeit auf das Wiederverbinden ist 5 Minuten.
6. Die Aktualisierung wird durch die Operation *FinalizeUpdate* finalisiert, wenn das Gerät einen anderen Aktualisierungsstatus als *UpdateRunning* oder *DeviceRestartRequired* zurückgibt. Die Aktualisierung ist zu wiederholen, wenn der finale Aktualisierungsstatus *DownloadUpdateFileFailed* ist.
7. Der Update-Controller kann das Ergebnis der Aktualisierung an den Update-Manager übermitteln.

Alle weiteren Informationen sind in Kapiteln 2.2ff in englischer Sprache beschrieben.

2 Operations of DeviceManagementService

The `DeviceManagementService` Operations are defined in tasks dealing on the one side with the device functions like

— Request static information about the device

— Change the device configuration

— Request the device configuration

— Restart the device

— Request the current device status

— Request error codes from the device

— Initiate and monitor updates of the device

and on the other side with functions for Services on the device like

— Static information about the services

— Request the state of the services

Operation	**Request**	**Verwendeter Datentyp, Datenstruktur**
GetDeviceInformation	Req.	-
	Resp.	DeviceManagementService. GetDeviceInformationResponseStructure
SubscribeDeviceInformation	Req.	SubscribeRequestStructure
	Resp.	SubscribeResponseStructure
UnsubscribeDeviceInformation	Req.	UnsubscribeRequestStructure
	Resp.	UnsubscribeResponseStructure
GetDeviceConfiguration	Req.	-
	Resp.	DeviceManagementService. GetDeviceConfigurationResponseStructure
GetDeviceStatus	Req.	-
	Resp.	DeviceManagementService. GetDeviceStatusResponseStructure
SubscribeDeviceStatus	Req.	SubscribeRequestStructure
	Resp.	SubscribeResponseStructure
UnsubscribeDeviceStatus	Req.	UnsubscribeRequestStructure
	Resp.	UnsubscribeResponseStructure
GetDeviceErrorMessages	Req.	-
	Resp.	DeviceManagementService. GetDeviceErrorMessagesResponseStructure
SubscribeDeviceErrorMessages	Req.	SubscribeRequestStructure
	Resp.	SubscribeResponseStructure
UnsubscribeDeviceErrorMessages	Req.	UnsubscribeRequestStructure
	Resp.	UnsubscribeResponseStructure
RestartDevice[1]	Req.	-
	Resp.	DataAcceptedResponseStructure

Operation	Request	Verwendeter Datentyp, Datenstruktur
GetServiceInformation	Req.	-
	Resp.	DeviceManagementService. GetServiceInformationResponseStructure
SubscribeServiceInformation	Req.	SubscribeRequestStructure
	Resp.	SubscribeResponseStructure
UnsubscribeServiceInformation	Req.	UnsubscribeRequestStructure
	Resp.	UnsubscribeResponseStructure
GetServiceStatus	Req.	-
	Resp.	DeviceManagementService. GetServiceStatusResponseStructure
SubscribeServiceStatus	Req.	SubscribeRequestStructure
	Resp.	SubscribeResponseStructure
UnsubscribeServiceStatus	Req.	UnsubscribeRequestStructure
	Resp.	UnsubscribeResponseStructure
GetAllSubdeviceInformation	Req.	-
	Resp.	DeviceManagementService. GetAllSubdeviceInformationResponseStructure
SubscribeAllSubdeviceInformation	Req.	SubscribeRequestStructure
	Resp.	SubscribeResponseStructure
UnsubscribeAllSubdeviceInformation	Req.	UnsubscribeRequestStructure
	Resp.	UnsubscribeResponseStructure
GetDeviceStatusInformation	Req.	-
	Resp.	DeviceManagementService. GetDeviceStatusInformationResponseStructure
SubscribeDeviceStatusInformation	Req.	SubscribeRequestStructure
	Resp.	SubscribeResponseStructure
UnsubscribeDeviceStatusInformation	Req.	UnsubscribeRequestStructure
	Resp.	UnsubscribeResponseStructure
GetAllSubdeviceStatusInformation	Req.	-
	Resp.	DeviceManagementService. GetAllSubdeviceStatusInformationResponseStructure
Subscribe AllSubdeviceStatusInformation	Req.	SubscribeRequestStructure
	Resp.	SubscribeResponseStructure
Unsubscribe AllSubdeviceStatusInformation	Req.	UnsubscribeRequestStructure
	Resp.	UnsubscribeResponseStructure
GetAllSubdeviceErrorMessages	Req.	-
	Resp.	DeviceManagementService. GetAllSubdeviceErrorMessagesResponseStructure
SubscribeAllSubdeviceErrorMessages	Req.	SubscribeRequestStructure
	Resp.	SubscribeResponseStructure
Unsubscribe AllSubdeviceErrorMessages	Req.	UnsubscribeRequestStructure
	Resp.	UnsubscribeResponseStructure
InstallUpdate[1]	Req.	DeviceManagementService. InstallUpdateRequestStructure
	Resp.	DeviceManagementService. InstallUpdateResponseStructure
RetrieveUpdateState[1]	Req.	DeviceManagementService. RetrieveUpdateStateRequestStructure
	Resp.	DeviceManagementService. RetrieveUpdateStateResponseStructure

Operation	Request	Verwendeter Datentyp, Datenstruktur
GetUpdateHistory[1]	Req.	-
	Resp.	DeviceManagementService. GetUpdateHistoryResponseStructure
FinalizeUpdate[1]	Req.	DeviceManagementService. FinalizeUpdateRequestStructure
	Resp.	DeviceManagementService. FinalizeUpdateResponseStructure
FinalizeAllPendingUpdates[1]	Req.	-
	Resp.	DataAcceptedResponseStructure

Tabelle/Table 1 Beschreibung von Operationen des DeviceManagementService /
Description of Operationen des DeviceManagementService

1) Operation is optional

2.1 Update of Firmware and Configuration

Update is initiated by operation InstallUpdate. Using this operation an update controller provides an update file for an IBIS-IP device which can be used by the device for installation. Location of the update file is given by an optional URI. If no storage location is provided, the device has to be able by itself to retrieve the update file from a pre-configured storage location. Device downloads update file and checks it afterwards. Update is rejected if update file is not suitable. Operation RetrieveUpdateState can be used during a running update to get its current state. After completion of update it has to be finalized using operation FinalizeUpdate.

Update history is persistently stored by IBIS-IP device. It can be queried using operation GetUpdateHistory.

Management of dependencies to other updates is not covered. If several components of an IBIS-IP device must receive different updates at the same time, these updates must be included in a single update file.

Update controller does not support update schedules.

Right assignment of update file to the device to be updated is not the task of the update controller. For this an update manager can be used, which calls the update controller as needed. In such a case the update controller reports the update result to the update manager which handles the result appropriately.

The update controller has to be a program. These tasks are assigned to it:

- Controlling of update process beginning with operation InstallUpdate and ending with operation FinalizeUpdate
- Reporting update result to update manager

The update manager may be a program or a person. These tasks are assigned to it:

- Selection of update file
- Scheduling of updates

The update controller restarts device if it returns update status *DeviceRestartRequired*. At least at the end of an update a restart of the updated device is required. Intermediate restarts shall be done as needed. Initial update status after an intermediate restart shall be *UpdateRunning*.

With having the device during the update process not available in the usual way, the device status messages have to be adapted accordingly.

2.1.1 Update Process

1. Update controller performs operation *InstallUpdate* and continues if the device accepts the update by returning *UpdateAccepted*.
2. Query update state can be used periodically using operation *RetrieveUpdateState*. Maximum query interval is 5s. Minimum query interval is 1s.
3. Restart device using operation *DeviceManagementService*.*RestartDevice* if the device returns update status *DeviceRestartRequired*.
4. Wait while the device shuts down. The default value of the shut down wait time is 10s.
5. Check if the device is reconnected to the IBIS-IP network and go to update state query afterwards. The default value of the reconnect wait time is 5 minutes.
6. Finalize the update using operation *FinalizeUpdate* if the device returns any other update status than *UpdateRunning* or *DeviceRestartRequired*. Repeat the update if the final update status is *DownloadUpdateFileFailed*.
7. The update result can be reported to the update manager.

2.2 Data Structure of Operation GetDeviceInformation

The operation GetDeviceInformation returns static device information like serial number, manufacturer etc.

2.2.1 Request

Because of being a ***`GetDeviceInformation`*** operation, there is no request structure for this operation.

2.2.2 Response

DeviceManagementService.GetDeviceInformationResponse				*+Structure*	**Request structure with non configuration able device information**
				choice	One of the choices below
	a	***DeviceManagementService.GetDeviceInformationResponseData***	**–1:1**	*+DeviceManagementService.GetDeviceInformationResponse Data*	Detailed request structure with the non configurable device parameters (cf. table below)
	b	***OperationErrorMessage***		*IBIS-IP.string*	Error message

Table 2 Description of DeviceManagementService.GetDeviceInformationResponse

DeviceManagementService.GetDeviceInformationResponseData			*+Structure*	**Detailed response structure with static device settings**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	DeviceInformation	**1:1**	*+DeviceInformation*	Detailed response structure (cf. VDV 301-2-1)

Table 3 Description of DeviceManagementService.GetDeviceInformationResponseData

The data structures described in VDV 301-2-1 are used to set up subscriptions.

2.3 Data Structure of Operation SubscribeDeviceInformation

The data structures described in VDV 301-2-1 are used to set up subscriptions.

2.4 Data Structure of Operation UnsubscribeDeviceInformation

For the subscription the data structures from VDV 301-2-1 are used.

2.5 Data Structure of Operation GetDeviceConfiguration

The **`GetDeviceConfiguration`** operation enables to set the single variable parameter of a device. This parameter is the device –ID, which is used as a reference to the plugged position inside the vehicle

2.5.1 Request

Because of being a ***`GetDeviceConfiguration`*** operation, there is no request structure for this operation.

2.5.2 Response

DeviceManagementService.GetDeviceConfigurationResponse				*+Structure*	**Response structure with device configuration data content**
				choice	One of the structures below
	a	***DeviceManagementService.GetDeviceConfigurationResponseData***	**–1:1**	*+DeviceManagementService.GetDeviceConfigurationResponseData*	Detailed response structure with the device configuration (cf. below)
	b	***OperationErrorMessage***		*IBIS-IP.string*	Error message

Table 4 Description of DeviceManagementService.GetDeviceConfigurationResponse

DeviceManagementService.GetDeviceConfigurationResponseData			*+Structure*	**Detailed response structure with the device configuration**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	DeviceID	**1:1**	*IBIS-IP.int*	Device-ID (device plug-in position)

Table 5 Description of DeviceManagementService.GetDeviceConfigurationResponseData

2.6 Data Structure of Operation GetDeviceStatus

2.6.1 Request

Because of being a ***GetDeviceStatus*** operation, there is no request structure for this operation.

2.6.2 Response

DeviceManagementService.GetDeviceStatusResponse				*+Structure*	**Response structure with the device status**
				choice	One of the structures below
	a	***DeviceManagementService.GetDeviceStatusResponseData***	**–1:1**	*+DeviceManagementService.GetDeviceStatusResponseData*	Detailed response structure with the device status information (cf. below)
	b	***OperationErrorMessage***		*IBIS-IP.string*	Error message

Table 6 Description of DeviceManagementService.GetDeviceStatusResponse

DeviceManagementService.GetDeviceStatusResponseData			*+Structure*	**Detailed response structure with the device status information**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	DeviceState	**1:1**	*DeviceStateEnumeration*	Device status (cf. VDV 301-2-1)

Table 7 Description of DeviceManagementService.GetDeviceStatusResponseData

2.7 Data Structure of Operation SubscribeDeviceStatus

The data structures described in VDV 301-2-1 are used to set up subscriptions.

2.8 Data Structure of Operation UnsubscribeDeviceStatus

The data structures described in VDV 301-2-1 are used to set up subscriptions.

2.9 Data Structure of Operation GetDeviceErrorMessages

2.9.1 Request

Because of being a ***GetDeviceErrorMessage*** operation, there is no request structure for this operation.

2.9.2 Response

DeviceManagementService.GetDeviceErrorMessagesResponse				*+Structure*	**Response structure for device error messages**
				choice	One of the choices below
	a	***DeviceManagementService.GetDeviceErrorMessagesResponseData***	**–1:1**	*+DeviceManagementService.GetDeviceErrorMessagesResponseData*	Detailed response structure for device errors
	b	***OperationErrorMessage***		*IBIS-IP.string*	Error message

Table 8 Description of DeviceManagementService.GetDeviceErrorMessagesResponse

DeviceManagementService.GetDeviceErrorMessagesResponseData			*+Structure*	**Detailed response structure for device errors**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	ErrorMessage	0:*	*+ Message*	Error message, if available a minimum number of 10 error messages seems useful

Table 9 Description of DeviceManagementService.GetDeviceErrorMessagesResponseData

2.10 Data Structure of Operation SubscribeDeviceErrorMessages

The data structures described in VDV 301-2-1 are used to set up subscriptions.

2.11 Data Structure of Operation UnsubscribeDeviceErrorMessages

The data structures described in VDV 301-2-1 are used to set up subscriptions.

2.12 Data Structure of Operation RestartDevice

2.12.1 Request

With the request **`RestartDevice`** is no data transmitted.

2.12.2 Response

For the acknowledge of the request the DataAcceptedResponseStructure (cf. VDV 301-2-1) is used.

2.13 Data Structure of Operation GetServiceInformation

2.13.1 Request

Because of being a ***`GetServiceInformation`*** operation, there is no request structure for this operation.

2.13.2 Response

DeviceManagementService.GetServiceInformationResponse				*+Structure*	**Response structure with information about the available services at the device**
				choice	One of the choices below
	a	***DeviceManagementService.GetServiceInformationResponseData***	**–1:1**	*+DeviceManagementService.GetServiceInformationResponseData*	Detailed response structure with information about the available services (cf. below)
	b	***OperationErrorMessage***		*IBIS-IP.string*	Error message

Table 10 Description of DeviceManagementService.GetServiceInformationResponse

DeviceManagementService.GetServiceInformationResponseData			*+Structure*	**Detailed response structure with information about the available services**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	ServiceInformationList	**1:1**	*+ServiceInformationList*	List of available services (cf. VDV 301-2-1)

Table 11 Description of DeviceManagementService.GetServiceInformationResponseData

2.14 Data Structure of Operation SubscribeServiceInformation

For this subscription the data structures from VDV 301-2-1 are used.

2.15 Data Structure of Operation UnsubscribeServiceInformation

To terminate this subscription the structures of VDV 301-2-1 are used.

2.16 Data Structure of Operation GetServiceStatus

2.16.1 Request

Because of being a ***`GetServiceStatus`*** operation, there is no request structure for this operation.

2.16.2 Response

DeviceManagementService.GetServiceStatusResponse				*+Structure*	**Response structure with status information about the services located at the device**
				choice	One of the choices below
	a	***DeviceManagementService.GetServiceStatusResponseData***	**–1:1**	*+DeviceManagementService.GetServiceStatusResponseData*	Detailed response structure with the status of services running on the device (cf. below)
	b	***OperationErrorMessage***		*IBIS-IP.string*	Error message

Table 12 Description of DeviceManagementService.GetServiceStatusResponse

DeviceManagementService.GetServiceStatusResponseData			*+Structure*	**Detailed response structure with the services located at the device**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	ServiceSpecificationWithStateList	**1:1**	*+ServiceSpecificationWithStateList*	Service list including the status (cf. VDV 301-2-1)

Table 13 Description of DeviceManagementService.GetServiceStatusResponseData

2.17 Data Structure of Operation SubscribeServiceStatus

For this subscription the data structures from chapters VDV 301-2-1 are used.

2.18 Data Structure of Operation UnsubscribeServiceStatus

To terminate this subscription the structures of chapters VDV 301-2-1 are used.

2.19 Data Structures of Operation GetAllSubdeviceInformation

2.19.1 Request

Because of being a **`Get`** operation, there is no request structure for operation ***`GetAllSubdeviceInformation`***.

2.19.2 Response

DeviceManagementService. GetAllSubdeviceInformationResponse				*+Structure*	**Response structure of operation *GetAllSubdeviceInformation***
				choice	One of the structures below
	a	***DeviceManagementService.GetAllSubdeviceInformationResponseData***	**–1:1**	*+DeviceManagementService.GetAllSubdeviceInformationResponseData*	Detailed response structure for static parameters of subdevices (cf. table below)
	b	***OperationErrorMessage***		*IBIS-IP.string*	Error message indicating cause of failure of operation

Table 14 Description of DeviceManagementService.GetAllSubdeviceInformationResponse

DeviceManagementService. GetAllSubdeviceInformationResponseData			*+Structure*	**Detailed response structure for static parameters of subdevices**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	SubdeviceInformationList	**1:***	*+SubdeviceInformation*	List of static parameters of subdevices (cf. table below)

Table 15 Description of DeviceManagementService.GetAllSubdeviceInformationResponseData

DeviceManagementService. SubdeviceInformation			*+Structure*	**Static parameters of subdevices**
	SubdeviceName	**1:1**	*IBIS-IP.string*	Name of subdevice
	DeviceInformation	**1:1**	*+DeviceInformation*	Static device parameters (cf. VDV 301-2-1)

Table 16 Description of DeviceManagementService.SubdeviceInformationStructure

2.20 Data Structures of Operation SubscribeAllSubdeviceInformation

Data structures described in document VDV 301-2-1 are used to establish subscription.

2.21 Data Structures of Operation UnsubscribeAllSubdeviceInformation

Data structures described in document VDV 301-2-1 are used to terminate subscription.

2.22 Data Structures of Operation GetDeviceStatusInformation

2.22.1 Request

Because of being a **`Get`** operation, there is no request structure for operation ***`GetDeviceStatusInformation`***.

2.22.2 Response

DeviceManagementService. GetDeviceStatusInformationResponse				*+Structure*	**Response structure of operation *GetDeviceStatusInformation***
				choice	One of the structures below
	a	***DeviceManagement Service.DeviceStat usInformationResp onseData***	**–1:1**	*+DeviceMana gementServic e.GetDeviceSt atusInformatio nResponseDa ta*	Detailed response structure for detailed device status (cf. table below)
	b	***OperationErrorMes sage***		*IBIS-IP.string*	Error message indicating cause of failure of operation

Table 17 Description of DeviceManagementService.GetDeviceStatusInformationResponse

DeviceManagementService. GetDeviceStatusInformationResponseData			*+Structure*	**Detailed response structure for detailed device status**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	DeviceStatusInformati on	**1:1**	*+DeviceStatus Information*	Detailed device status (cf. table below)

Table 18 Description of DeviceManagementService.GetDeviceStatusInformationResponseData

DeviceManagementService. DeviceStatusInformation			*+Structure*	**Detailed device status**
	DeviceState	**1:1**	*DeviceStateE numeration*	Common device state (cf. VDV 301-2-1)
	DeviceStatusList	**0:***	*+DeviceStatus*	List of statuses (cf. table below)

Table 19 Description of DeviceManagementService.DeviceStatusInformationStructure

DeviceManagementService.DeviceStatus			*+Structure*	**Status**
	DeviceStatusName	**1:1**	*IBIS-IP.string*	Name of status
	DeviceStatusFlag	**1:1**	*IBIS-IP. boolean*	Activation of status
	DeviceStatusImpact	0:1	*DeviceStateEnumeration*	Impact of the status to the DeviceState.
	eDeviceStatusPriority	0:1	*IBIS-IP.int*	Priority defines how most important status is. 1 means most important, higher values are less important. The priority should only be used once.

Table 20 Description of DeviceManagementService.DeviceStatusStructure

2.23 Data Structures of Operation SubscribeDeviceStatusInformation

Data structures described in document VDV 301-2-1 are used to establish subscription.

2.24 Data Structures of Operation UnsubscribeDeviceStatusInformation

Data structures described in document VDV 301-2-1 are used to terminate subscription.

2.25 Data Structures of Operation GetAllSubdeviceStatusInformation

2.25.1 Request

Because of being a **`Get`** operation, there is no request structure for operation ***`GetAllSubdeviceStatusInformation`***.

2.25.2 Response

DeviceManagementService. GetAllSubdeviceStatusInformation Response				*+Structure*	**Response structure of operation *GetAllSubdeviceStatusInformation***
				choice	One of the structures below
	a	***DeviceManagementService.GetAllSubdeviceStatusInformationResponseData***	**–1:1**	*+DeviceManagementService.GetAllSubdeviceStatusInformationResponseData*	Detailed response structure for detailed statuses of subdevices (cf. table below)
	b	***OperationErrorMessage***		*IBIS-IP.string*	Error message indicating cause of failure of operation

Table 21 Description of DeviceManagementService.GetAllSubdeviceStatusInformationResponse

DeviceManagementService. GetAllSubdeviceStatusInformation ResponseData			*+Structure*	**Detailed response structure for detailed statuses of subdevices**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	SubdeviceStatusInformationList	**1:***	*+SubdeviceStatusInformation*	List of detailed statuses of subdevices (cf. table below)

Table 22 Description of DeviceManagementService.GetAllSubdeviceStatusInformationResponseData

DeviceManagementService. SubdeviceStatusInformation			*+Structure*	**Detailed status of subdevice**
	SubdeviceName	**1:1**	*IBIS-IP.string*	Name of subdevice
	DeviceStatusInformation	**1:1**	*+DeviceStatus Information*	Detailed device status (cf. Table 19)

Table 23 Description of DeviceManagementService.SubdeviceStatusInformationStructure

2.26 Data Structures of Operation SubscribeAllSubdeviceStatusInformation

Data structures described in document VDV 301-2-1 are used to establish subscription.

2.27 Data Structures of Operation UnsubscribeAllSubdeviceStatusInformation

Data structures described in document VDV 301-2-1 are used to terminate subscription.

2.28 Data Structures of Operation GetAllSubdeviceErrorMessages

2.28.1 Request

Because of being a **`Get`** operation, there is no request structure for operation ***`GetAllSubdeviceErrorMessages`***.

2.28.2 Response

DeviceManagementService. GetAllSubdeviceErrorMessagesResponse				*+Structure*	**Response structure of operation GetAllSubdeviceErrorMessages**
				choice	One of the structures below
	a	***DeviceManagement Service.GetAllSubdeviceErrorMessagesResponseData***	**–1:1**	*+DeviceManagementService.GetAllSubdeviceErrorMessagesResponseData*	Detailed response structure for error messages of subdevices (cf. table below)
	b	***OperationErrorMessage***		*IBIS-IP.string*	Error message indicating cause of failure of operation

Table 24 Description of DeviceManagementService.GetAllSubdeviceErrorMessagesResponse

DeviceManagementService. GetAllSubdeviceErrorMessages ResponseData			*+Structure*	**Detailed response structure for error messages of subdevices**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	SubdeviceErrorMessagesList	**1:***	*+SubdeviceErrorMessages*	List of error messages of subdevices (cf. table below)

Tabelle 25 Description of DeviceManagementService.GetAllSubdeviceErrorMessagesResponseData

DeviceManagementService. SubdeviceErrorMessages			*+Structure*	**Error messages of subdevice**
	SubdeviceName	**1:1**	*IBIS-IP.string*	Name of subdevice
	ErrorMessage	0:*	*+Message*	Error messages and warnings only (no status information, cf. VDV 301-2-1), if available a minimum number of 10 error messages seems useful

Table 26 Description of DeviceManagementService.SubdeviceErrorMessagesStructure

2.29 Data Structures of Operation SubscribeAllSubdeviceErrorMessages

Data structures described in document VDV 301-2-1 are used to establish subscription.

2.30 Data Structures of Operation UnsubscribeAllSubdeviceErrorMessages

Data structures described in document VDV 301-2-1 are used to terminate subscription.

2.31 Data Structures of Operation InstallUpdate

This operation asks a peripheral device to install a new update. In case there are other necessary dependencies or update conditions these must be included in the update package itself. There must be only one active update per device at any time. If a device receives a new *InstallUpdate* request while a previous update job is not finished yet the new request must not be accepted.

2.31.1 Request

DeviceManagementService. InstallUpdateRequest			*+Structure*	**Request structure of DeviceManagementService**
	UpdateID	0:1	*IBIS-IP. NMTOKEN*	Unique id generated by the controller to identify an update job
	UpdateTimestamp	0:1	*IBIS-IP. dateTime*	Timestamp used for GetUpdateStates and RetrieveUpdateState responses and for logging
	UpdateURL	0:1	*IBIS-IP.anyURI*	URL from which the device shall download the update file
	UpdateFileChecksum	0:1	*Checksum Structure*	Optional checksum of update file
	UpdateFileSize	0:1	*IBIS-IP. unsignedLong*	Optional size of update file

Table 27 Description of DeviceManagementService.InstallUpdateRequestStructure

The UpdateURL shall be valid until the update process is finished. End of update is signalled by operation FinalizeUpdate.

Minimum requirement for any device is to enable transfer of update file with a suitable protocol.

2.31.2 Response

DeviceManagementService. InstallUpdateResponse				*+Structure*	**Response structure of DeviceManagementService**
				choice	One of the structures below
	a	***UpdateAccept***	**–1:1**	*UpdateAcceptEnumeration*	Enumeration (cf. Table 29 below)
	b	***OperationErrorMessage***		*IBIS-IP.string*	Error message indicating cause of failure of operation

Table 28 Description of DeviceManagementService.InstallUpdateResponseStructure

Enumeration	Defined Values	Description
UpdateAcceptEnumeration	UpdateAccepted URLTypeUnknown NoUpdatesAllowed ToBePostponed	Update will be performed URL type has been rejected, e.g. FTP may not supported by some devices No updates are possible Update has to be postponed

Table 29 Description of UpdateAcceptEnumeration

On accepting an update the device may change to a special update mode. Some desired device functionality may not be available during the update. After accepting an update the device shall handle the corresponding *UpdateID* as a known request parameter. The initial update status shall be *UpdateRunning*.

2.32 Data Structures of Operation RetrieveUpdateState

This operation is used to retrieve the update state of active update job. The job is specified by an *UpdateID*.

2.32.1 Request

DeviceManagementService. RetrieveUpdateStateRequest			*+Structure*	**Request structure of DeviceManagementService**
	UpdateID	**1:1**	*IBIS-IP. NMTOKEN*	Unique id generated by the controller to identify an update job

Table 30 Description of DeviceManagementService.RetrieveUpdateStateRequestStructure

2.32.2 Response

If the requested UpdateID is unknown, an operation error message is returned.

DeviceManagementService. RetrieveUpdateStateResponse				*+Structure*	**Response structure of DeviceManagementService**
				Choice	One of the structures below
	a	***UpdateStateData***	**–1:1**	*+DeviceManagementService. UpdateStateData*	Detailed response structure (cf. Table 32 below)
	b	***OperationErrorMessage***		*IBIS-IP.string*	Error message indicating cause of failure of operation

Table 31 Description of DeviceManagementService.RetrieveUpdateStateResponseStructure

DeviceManagementService. UpdateStateData			*+Structure*	**Response structure of DeviceManagementService**
	UpdateID	**1:1**	*IBIS-IP. NMTOKEN*	Unique id generated by the controller to identify an update job
	UpdateTimestamp	**1:1**	*IBIS-IP. dateTime*	Timestamp given by operation InstallUpdate
	UpdateStatus	**1:1**	*UpdateStatusEnumeration*	Current status of update (cf. Table 33 below)
	UpdateDetails	**0:1**	*IBIS-IP.string*	Optional device specific update log

Table 32 Description of DeviceManagementService.UpdateStateDataStructure

Enumeration	**Defined Values**	**Description**
UpdateStatusEnumeration	UpdateRunning	Update in progress
	DeviceRestartRequired	Device has to be restarted by operation RestartDevice
	DownloadUpdateFileFailed	Specified update file could not be downloaded from URL
	UpdateFileCorrupted	Specified update file is not usable
	UpdateNotNecessary	State of device firmware already as required
	InstallationFailed	Update failed e.g. due to memory write error
	InstallationSuccessful	Update successfully completed

Table 33 Description of UpdateStatusEnumeration

2.33 Data Structures of Operation GetUpdateHistory

This operation is used to get the update history of a device. It needs no request data.

2.33.1 Request

Because of being a **`Get`** operation, there is no request structure for operation ***`GetUpdateHistory`***.

2.33.2 Response

DeviceManagementService. GetUpdateHistoryResponse				*+Structure*	**Response structure of DeviceManagementService**
				Choice	One of the structures below
	a	***UpdateHistory***	**–1:1**	*+DeviceManagementService. UpdateHistory*	Detailed response structure (cf. table below)
	b	***OperationErrorMessage***		*IBIS-IP.string*	Error message indicating cause of failure of operation

Table 34 Description of DeviceManagementService.GetUpdateHistoryResponseStructure

DeviceManagementService. UpdateHistory			*+Structure*	**Response structure of DeviceManagementService**
	UpdateHistoryEntry	**0:***	*+DeviceManagementService.UpdateHistoryEntry*	List of updates (empty if the device was not updated yet) Cf. table below

Table 35 Description of DeviceManagementService.UpdateHistoryStructure

Minimum requirement for any device allowing updates is an update history depth of 1, i.e. history shall contain at least the last update performed (regardless of its result), if there was any.

DeviceManagementService. UpdateHistoryEntry			*+Structure*	**Response structure of DeviceManagementService**
	UpdateID	**1:1**	*IBIS-IP. NMTOKEN*	Unique id generated by the controller to identify an update job
	UpdateTimestamp	**1:1**	*IBIS-IP. dateTime*	Timestamp given by operation InstallUpdate
	UpdateURL	**1:1**	*IBIS-IP.anyURI*	URL from which the device downloaded the update file
	UpdateStatus	**1:1**	*UpdateStatusEnumeration*	Status of update (cf. Table 33 above) Typically final status (InstallationSuccessfull or InstallationFailed)
	DataVersionList	**0:1**	*+DataVersionList*	Optional list of updated components
	UpdateDetails	**0:1**	*IBIS-IP.string*	Optional device specific update log

Table 36 Description of DeviceManagementService.UpdateHistoryEntryStructure

2.34 Data Structures of Operation FinalizeUpdate

Any accepted update job has to be finished by performing the operation *FinalizeUpdate*. The job is specified by an *UpdateID*.

2.34.1 Request

DeviceManagementService. FinalizeUpdateRequest			*+Structure*	**Request structure of DeviceManagementService**
	UpdateID	**1:1**	*IBIS-IP. NMTOKEN*	Unique id generated by the controller to identify an update job

Table 37 Description of DeviceManagementService.FinalizeUpdateRequestStructure

2.34.2 Response

If the requested *UpdateID* is unknown an operation error message is returned.

DeviceManagementService. FinalizeUpdateResponse				*+Structure*	**Response structure of DeviceManagementService**
				Choice	One of the structures below
	a	***UpdateStatus***	**–1:1**	*UpdateStatus Enumeration*	Status of update on its finalization
	b	***OperationErrorMessage***		*IBIS-IP.string*	Error message indicating cause of failure of operation

Table 38 Description of DeviceManagementService.FinalizeUpdateResponseStructure

If the device receives a valid *FinalizeUpdateRequest* it returns to normal operation and the requested *UpdateID* is no longer valid.

Operation *FinalizeUpdate* has to be done in these cases:

1. Device returns any other update status than *UpdateRunning* or *DeviceRestartRequired*.
2. Update has to be cancelled because the update controller is shut down.

2.35 Data Structures of Operation FinalizeAllPendingUpdates

All pending update jobs stored in update history are finalized.

2.35.1 Request

There are no data used by the request.

2.35.2 Response

Common DataAcceptedResponseStructure is used.

Operation FinalizeAllPendingUpdates has to be done if operation InstallUpdate returns NoUpdatesAllowed.

2 Versionshistorie / Version History

Version 2.2

2.1.1 Funktionale Erweiterungen
Functional Upgrade

- Operationen zur Aktivierung/Deaktivierung von Geräten und Diensten entfernt
  Operation for activation/deactivation of devices and services deleted
- Optionale Methode in Tabelle 1 markiert
  marked methodes in table 1 as optional

2.1.2 Technische Ergänzungen/Korrekturen
Technical Upgrade/Corrections

- Keine/none

Version 2.4

2.1.3 Funktionale Erweiterungen
Functional Upgrade

- Keine/none

2.1.4 Technische Ergänzungen/Korrekturen
Technical Upgrade/Corrections

- Anzahl an Fehlermeldungen in GetDeviceErrorMessageResponse und GetAllSubdeviceErrorMessageResponse von verpflichtend 10 auf optional reduziert / number of error messages in GetDeviceErrorMessageResponse and GetAllSubdeviceErrorMessageResponse reduced from mandatory 10 to optional
- die DeviceManagementService.DeviceStatusStructure an das XML angepasst / adapted the DeviceManagementService.DeviceStatusStructure to the XML-file
- Datenstrukturen in InstallUpdate angepasst, so dass Geräte von vordefinierten Orten Update-Pakete installieren können / adapted data structures in InstallUpdate, so that devices can install update packages from pre-defined storage locations

Tabellenverzeichnis

Regelwerke – Normen und Empfehlungen / References

(1) CEN/TS 13149-7 Öffentlicher Verkehr - Planungs- und Steuerungssysteme für Straßenfahrzeuge - Teil 7: IP-basierende Vernetzung in einem Fahrzeug, Netzwerk- und Systemarchitektur (CEN/TS 13149-7:2015)

(2) CEN/TS 13149-8 Öffentlicher Verkehr - Planungs- und Steuerungssysteme für Straßenfahrzeuge - Teil 8: Physikalische Schicht für IP-Kommunikation; Englische Fassung CEN/TS 13149-8:2013

(3) VDV 301-2-0 IBIS-IP Beschreibung der Dienste / Service description Basisdienste / Base Services DeviceManagementService, SystemManagementService, SystemDocumentationService V2.0, 02/2018

(4) VDV 301-2-1 IBIS-IP Beschreibung der Dienste / Service description Gemeinsame Datenstrukturen und Aufzählungstypen/ Common data structures and enumerations, 05/2017

(5) VDV 301-1 IBIS-IP- Teil 1: Systemarchitectur / System architecture V1.0, 01/2014

Die IBIS-IP XSD-Dateien stehen unter www.vdv.de/ip-kom-oev.aspx zum Download bereit.

The IBIS-IP XSD files are available for download at www.vdv.de/ip-kom-oev.aspx.

Impressum

Verband Deutscher Verkehrsunternehmen e. V. (VDV)
Kamekestraße 37-39 · 50672 Köln
T 0221 57979-0 · F 0221 57979-8000
info@vdv.de · www.vdv.de

Ansprechpartner

Dipl.-Ing. Dirk Weißer
T 0221 57979-176
F 0221 57979-8176
weisser@vdv.de

Verband Deutscher Verkehrsunternehmen e. V. (VDV)
Kamekestraße 37-39 · 50672 Köln
T 0221 57979-0 · F 0221 57979-8000
info@vdv.de · www.vdv.de