



A MITEL
PRODUCT
GUIDE

MiVoice MX-ONE

MX-ONE API, CSTA III - INTERFACE DESCRIPTION

Release 7.7
3/15519-ANF 901 43 Uen N

May 2024

Notices

The information contained in this document is believed to be accurate in all respects but is not warranted by **Mitel Networks Corporation (MITEL[®])**. The information is subject to change without notice and should not be construed in any way as a commitment by Mitel or any of its affiliates or subsidiaries. Mitel and its affiliates and subsidiaries assume no responsibility for any errors or omissions in this document. Revisions of this document or new editions of it may be issued to incorporate such changes. No part of this document can be reproduced or transmitted in any form or by any means - electronic or mechanical - for any purpose without written permission from Mitel Networks Corporation.

Trademarks

The trademarks, service marks, logos and graphics (collectively "Trademarks") appearing on Mitel's Internet sites or in its publications are registered and unregistered trademarks of Mitel Networks Corporation (MNC), its affiliates, parents, or subsidiaries (collectively "Mitel") or others. Use of the Trademarks is prohibited without the express consent from Mitel. Please contact our legal department at legal@mitel.com for additional information. For a list of the worldwide Mitel Networks Corporation registered trademarks, please refer to the website: <http://www.mitel.com/trademarks>.

[®], [™] Trademark of Mitel Networks Corporation

© Copyright 2024, Mitel Networks Corporation

All rights reserved

Contents

1 Introduction.....	1
1.1 References.....	1
2 Configuration.....	2
2.1 CSTA Settings.....	2
2.2 CSTA_Authentication Settings.....	3
2.3 Redundancy.....	4
3 Identifier Definitions.....	5
3.1 Device ID.....	5
3.1.1 Dialable Digits.....	5
3.1.2 Switching Function Representation.....	5
3.1.3 Device Number.....	6
3.1.4 URI Representation.....	6
3.2 CallID.....	6
3.3 ConnectionID.....	6
3.4 Monitor Cross Reference ID.....	6
4 MX-ONE Definitions.....	7
4.1 Named Device Types.....	7
4.2 Type of Device.....	7
4.3 CSTA SubType of Device.....	8
4.4 CSTA Phone SubType of Device.....	8
4.5 Long and Short Number.....	8
5 Supported CSTA Services and Events.....	9
5.1 Capability Exchange Services.....	9
5.1.1 Services.....	9
5.2 System Services.....	14
5.2.1 Services.....	14
5.3 Monitoring Services.....	16
5.3.1 Services.....	16
5.4 Snapshot Services.....	19
5.4.1 Services.....	19
5.5 Call Control Services and Events.....	20
5.5.1 Services.....	20
5.5.2 Events.....	31
5.6 Call Associated Features.....	50
5.6.1 Services.....	50
5.6.2 Events.....	51

5.7 Physical Device Features.....	52
5.7.1 Services.....	52
5.7.2 Events.....	54
5.8 Logical Device Features.....	54
5.8.1 Services.....	54
5.8.2 Events.....	60
5.9 Device Maintenance Events.....	64
5.9.1 Events.....	64
5.10 Vendor Specific Extensions Services & Events.....	66
5.10.1 Services.....	66
5.10.2 Events.....	68

6 Application Session Services..... 69

6.1 Services.....	69
6.1.1 Start Application Session Service.....	69
6.2 Stop Application Session Service.....	71
6.2.1 Service Request.....	71
6.3 Reset Application Session Timer Service.....	71
6.3.1 Service Request.....	72
6.4 Application Session Terminated Service.....	72
6.4.1 Service Request.....	73

7 Multi-terminal..... 74

7.1 Learning About Registered Terminals.....	74
7.2 Monitoring.....	75
7.2.1 Monitoring the Number.....	75
7.2.2 Monitoring the Number with Switching Function Representation.....	76
7.2.3 Monitoring the Terminal.....	77
7.3 Changes in Registered Terminals.....	77
7.3.1 Out of Service Event.....	77
7.3.2 Back in Service Event.....	78
7.3.3 Device Capabilities Changed Event.....	78
7.4 Services.....	79

8 Vendor Specific Extensions..... 80

8.1 Private Data on CSTA Services and Events.....	80
8.1.1 Services.....	80
8.1.2 Get Logical Device Information Response.....	81
8.1.3 Monitor Start Response Information.....	81
8.1.4 Events.....	81
8.2 Back in Service Information.....	81
8.3 Escape Services and Private Event.....	81
8.3.1 Services.....	81
8.3.2 Events.....	83
8.4 Private Data in CSTA Session Application Services.....	86
8.4.1 Services Containing Private Data.....	86

9 Supported Services by Device..... 87

10 Limitations.....	91
----------------------------	-----------

11 Appendix A, XML Schema File MX-ONE Private Data.....	92
--	-----------

11.1 MX-ONE-CSTA-PRIVATE-DATA.XSD.....	92
--	----

This chapter contains the following sections:

- [References](#)

This document describes services and events supported by the CSTA Phase III interface in MiVoice MX-ONE. This document is not an exhaustive description of the CSTA protocol and should be treated as additional to the ECMA269 document.

MX-ONE supports the XML protocol for the services of ECMA269 as defined in ECMA323. The CSTA XML documents can be transported over a TCP or TLS session to the MX-ONE, or over a SIP session (uaCSTA – user agent CSTA) as explained in TR/87.

Application Session Services (ECMA-354) are supported.

MX-ONE supports monitoring of calls, devices and trunks (external lines) however there is no requirement to start a monitor before a service request can be made.

All types of external lines can be monitored but only the snapshot device service is supported and then only on CAS (TLP30), ISDN, H323, and SIP trunks.

1.1 References

1. ECMA-269 Services for CSTA Phase III
2. ECMA-323 XML Protocol for CSTA Phase III
3. TR/82 Scenarios for Computer Supported Telecommunications Applications (CSTA) Phase III
4. TR/87 Using CSTA for SIP Phone User Agents (uaCSTA)
5. ECMA-354 Application Session Services (Authentication)

This chapter contains the following sections:

- [CSTA Settings](#)
- [CSTA_Authentication Settings](#)
- [Redundancy](#)

Each service node in a MX-ONE system can be configured to have a two CSTA interface, one with TLS active and one without TLS.

Each interface is independent, there is no load balancing or redundancy between different interfaces.

When the interface is configured the port to use can be chosen and there are several options that control some parts of the behavior. Each client connected to the configured interface inherits these configuration options.

Refer to the *MX-ONE CPI* for more information on how to configure the MX-ONE.

2.1 CSTA Settings

Parameters that is shared by all clients, set with *csta* command. The following parameters are shared with all application connection to MX-ONE.

- **Heartbeat Support**
 - If this is enabled a periodic systemStatus is sent to the client, if no systemStatusResponse is received then the connection is closed.
- **Application Session Services (Authentication)**
 - The criteria for the authentication of CSTA applications must be configured, i.e. which parameters to check. Application Identity, password, and duration timer are included.
- **Bypass of personal number**
 - When a deflect is requested should personal number of the deflect-to device be bypassed. This option is possible to control via private data in the deflect request which takes higher priority than the configuration option.
- **Permit diversion after deflection**
 - When a deflect is requested should diversion at the deflect-to device allowed. This option is possible to control via private data in the deflect request which takes higher priority than the configuration option.
- **Replace dialed number with deflected-to number**
 - When a deflect is requested should the dialed number be replaced. This can be used to control which voicemail box the caller will enter. This option is possible to control via private data in the deflect request which takes higher priority than the configuration option.
- **Connection view type**
 - Local or Fixed connection view. This controls how conference and transfer events are reported, see [1].
- **Send encryption keys in private event**

- In the private event reporting media streams encryption keys should be included. The encryption keys can be used by voice recorders to decode captured media streams.
- **Type of interface**
 - Is this an ECMA323 or uaCSTA interface. Both types can be configured on the same service node.
- **Use TLS**
 - Should TLS (Transport Layer Security) be used on the connection. Currently only supported for an ECMA323 interface.
- **Diversion category override**
 - When SetForwarding is ordered from the client should the category of the extension that forwarding is being ordered for be checked. This allows, for example, an InAttend operator to set forwarding on an extension that would be no allowed to do it themselves.
- **Application authentication**
 - If CSTA application is required to send application ID and Password when connecting to the system.

Note:

If set to No, the application can still supply application ID and Password. It is just not mandatory to do so. It is recommended that security is activated for the system.

2.2 CSTA_Authentication Settings

These settings include parameters that are shared by all clients using default user; or by application clients set with `csta_authentication` command.

These parameters are relevant when an application uses the Application Session Services to connect to MX-ONE. The parameters define characteristics that will be valid for all registered users.

A default user is always available for application that does not use Application Session Services to connect to MX-ONE. All default users share the same default user characteristics.

When you key in the `csta_authentication -c` command, a private event is sent with a current value of the TSS bool to the application.

The parameters are:

- **Send name in call controlling event**
 - Name is sent in 'switching function representation', N<DN>NM in the deviceIdentifier field.
 - Number presentation restriction override category
 - No number restriction override
- **Override external A-Party number restriction**
 - Override A-Party number restriction.
 - Override all number restriction.

- **Terminal Selection Service, TSS, allowed to handle multiplicity**

- True, TSS can handle multiplicity.
- False, TSS is not allowed to handle multiplicity.

- **Terminal Selection Service, TSS, convert number**

This parameter defines how numbers are presented to application, for application display purpose.

- True, TSS request must be sent with number in international number type, +xxxx. Event and responses will be converted to international number type +xxxx.
- False, TSS request must be sent with number in private unknown number type. Number must match what is received from the public destination and include all digits needed to route back to that destination.

- **Simplified TSS, only possible to call to remote device**

This parameter defines how remote extension shall function together with TSS service 'user defined number'.

- Yes, TSS 'user define remote' service, remote extension will work as 'ECF', that is call to remote extension will be distributed to remote device. Call from device will not work as remote extension.
- No, TSS 'user define remote' service will create a true remote extension, calls to and from remote device is possible.

2.3 Redundancy

The application can build redundancy by connecting to two (or more) Service Nodes and start monitoring needed devices on both connections. MX-ONE will send event on both (all) connections. The application can ignore the duplicated events from the extra links.

This chapter contains the following sections:

- [Device ID](#)
- [CallID](#)
- [ConnectionID](#)
- [Monitor Cross Reference ID](#)

3.1 Device ID

MX-ONE uses device identifier formats defined in section 10 of [2].

3.1.1 Dialable Digits

The dialable digits format is used to identify phone numbers. MX-ONE supports the digits 0-9, “*”, “#” and “+”.

To aid in readability, the digit string may contain the following visual separators: “-”, “(”, and “)”.

3.1.2 Switching Function Representation

The Switching function representation format is used to identify internal extensions when multi-terminal devices are in use. This allows for identification of the specific device. With the optional parameter, it is also used when session shall report users name.

This is a string containing between 2 and 64 characters.

MX-ONE uses the following format: N<DN!SA>NM

Where

- **N** identifies that Device Identifier uses the Switching Function Representation format.
- **< >** are angle brackets encompassing the number and sub-address.
- **DN** is the extension number of the user.
- **!** is a separator between number and sub-address.
- **SA** is the sub-address which is used to identify the specific terminal. This is an integer between 1 and 9. When sub address is no available is 'SA' omitted.
- **NM** represents the name of the person associated with the device. The name string may contain any character. Configuration determines if name string is sent in call controlling events. In some responses or other event, NM represents the device name. See more in [Learning About Registered Terminals](#) on page 74.

3.1.3 Device Number

MX-ONE uses this format in external calls to identify the used trunk line to represent trunks. The identifier is the trunk line identity which is a 10-digit string consisting of route, server and trunk line number. Route 63, Server 1, trunk line number 9 will look like this 0630010009.

3.1.4 URI Representation

MX-ONE supports URI representation using the tel scheme. It is only used when a monitor start request is received in this format from the client. The first monitor start request received controls whether URI representation is used for that session.

3.2 CallID

MX-ONE uses a 24-digit string as CallID.

3.3 ConnectionID

The ConnectionID is comprised of the CallID and deviceID described in previous sections.

3.4 Monitor Cross Reference ID

The monitor cross reference identifier (monitorCrossRefID) is used to uniquely identify the monitor. It is included in the in the positive response to a Monitor Start request and included as a parameter in all events for that monitor. MX-ONE will use a numeric value in the range 0 – 4294967295.

Monitor Stop releases the monitor cross reference identifier, after a short time out, can same monitor cross reference identifier be reused for next Monitor Start.

This chapter contains the following sections:

- [Named Device Types](#)
- [Type of Device](#)
- [CSTA SubType of Device](#)
- [CSTA Phone SubType of Device](#)
- [Long and Short Number](#)

4.1 Named Device Types

MX-ONE uses the following values as defined in [1]:

- Line
- Station
- Other

4.2 Type of Device

This parameter is included within private data in some responses and events to more clearly specify the type of device.

The possible values are:

- acdGroup
- analogExtension
- ctiGroup
- dectExtension
- digitalExtensionAdditionalDirectoryNumber
- digitalExtension
- genericExtension
- ipExtension
- multiTerminalExtension
- pbxGroup

Note:

The list above affects external applications, so change and update it with care. (Coordinate with application responsible).

4.3 CSTA SubType of Device

This parameter is included within private data in some responses and events to specify the level of support for CSTA services that can be expected.

The possible values are:

- noSupportForCstaCallControl
- multipleLineAccess
- singleLineAccess
- virtualSipExtension

4.4 CSTA Phone SubType of Device

This parameter is included within private data in some response and events to specify terminal type of the generic extension. When terminal report the vendor id, '6867i' will that be presented.

The other possible values are:

- H323Extension
- RemoteExtension
- Sip-cordlessPhone
- Sip-remotePhone
- Sip-softPhone
- Sip-videoPhone

4.5 Long and Short Number

When using customer groups and long and short number, Monitor Start always done on the long number, complete and unique number. SIP terminals can log on with either long or short number. Recording equipment need to know the *log on number* to be able to record the conversation. Therefore, the log on number is sent as private data in event.

Log on number information is sent in [Monitor Start Response](#), [Get Logical Device Information Response](#), and in [Back in Service Event](#).

Supported CSTA Services and Events

5

This chapter contains the following sections:

- [Capability Exchange Services](#)
- [System Services](#)
- [Monitoring Services](#)
- [Snapshot Services](#)
- [Call Control Services and Events](#)
- [Call Associated Features](#)
- [Physical Device Features](#)
- [Logical Device Features](#)
- [Device Maintenance Events](#)
- [Vendor Specific Extensions Services & Events](#)

The used parameters and expected values in the services and events supported by MX-ONE are described here. For further detail regarding each section see [1].

Not all services are supported for all device types.

5.1 Capability Exchange Services

5.1.1 Services

5.1.1.1 Get CSTA Features

The Get CSTA Features service obtains the list of supported services and events.

Service Request

No parameters are expected.

Positive Acknowledgment

Table 1: Positive acknowledgment

Parameter	Type	Description
supportedServices	Bitmap	List of the supported services.
supportedEvents	Bitmap	List of the supported events.

Negative Acknowledgment

Standard error codes as described in [1].

5.1.1.2 Get Logical Device Information

The Get Logical Device Information service is used to obtain the current set of characteristics/capabilities associated with the logical element of a given device.

Service Request

Table 2: Service request

Parameter	Type	Description
Device	DeviceID	The device being queried.

Positive Acknowledgment

Table 3: Positive acknowledgment

Parameter	Type	Description
deviceCategory	Enumerated	Group when ACD, CTI or PBX group, Station otherwise.
groupDeviceAttributes	Bitmap	Only one bit is set. Parameter is sent when deviceCategory is Group. For ACD and CTI groups ACD attribute is true. For PBX groups Hunt attribute is true.
namedDeviceTypes	Enumerated	See Named Device Types .
hasPhysicalElement	Boolean	For non-generic extensions always true. For generic extensions set to true when at least one terminal is logged on. False in all other cases.
acdModels	Bitmap	Visible ACD-related Devices is true, Non-Visible ACD-related Devices is false.
appearanceAddressable	Boolean	False – appearances are not addressable

appearanceType	Enumerated	Selected-Standard
otherPhysicalDeviceList	List of DeviceIDs	<p>Present when device being queried is a multi-terminal device. Contains a list of the currently logged on terminals presented in Switching Function Representation.</p> <p>NM in this context contains model name of device.</p> <p>Device list is sorted. Desk phone first, SIP second. Making SIP desk phone on top.</p>
transAndConfSetup	Bitmap	All bits are set to true.
privateData	CSTAPrivateData	<p>This includes:</p> <ul style="list-style-type: none"> • typeOfDevice • cstaSubTypeOfDevice • deviceModelName • ipAddress • logOnNumber <p>For more information, see the Type of Device section.</p>

Negative Acknowledgment

Standard error codes as described in [1].

5.1.1.3 Get Physical Device Information

The Get Physical Device Information service is used to obtain the current set of characteristics/capabilities associated with the physical element of a given device.

Service Request

Table 4: Service request

Parameter	Type	Description
Device	DeviceID	The device being queried.

Positive Acknowledgment

Table 5: Positive acknowledgment

Parameter	Type	Description
deviceCategory	Enumerated	Station
namedDeviceTypes	Enumerated	See Named Device Types .
hasLogicalElement	Boolean	True
deviceModelName	Characters	Where the model of physical device is known it is provided here. For example if the device is a Mitel SIP phone the model number is provided. See CSTA Phone SubType of Device .

Negative Acknowledgment

Standard error codes as described in [1].

Only supported on devices that have a physical element. If the device does not have a physical element e.g. a generic extension with no logged-on terminals, or a group the error requestIncompatibleWithDevice is used.

5.1.1.4 Get Switching Function Devices

The Get Switching Function Devices service is used to fetch the device categories for the given device.

Service Request

Table 6: Service request

Parameter	Type	Description
Device	DeviceID	The device being queried – this parameter is mandatory in MX-ONE.

Positive Acknowledgment

This service follows the multi-step Acknowledgment model. A positive response means one of more Switching Function Devices services will follow.

Table 7: Positive acknowledgment

Parameter	Type	Description
-----------	------	-------------

serviceCrossRefID	ServiceCrossRefID	The identifier used to associate this request to subsequent Switching Function Devices services.
-------------------	-------------------	--

Negative Acknowledgment

Standard error codes as described in [1].

5.1.1.5 Switching Function Devices

The Switching Function Devices service is used to provide a list of devices as a result of the Get Switching Function Devices service.

Table 8: Service request

Parameter	Type	Description
serviceCrossRefID	ServiceCrossRefID	Identifier used to associate this request with the Get Switching Function Devices request.
segmentID	Value	MX-ONE currently sends all devices in one segment, so value is 1.
lastSegment	Boolean	MX-ONE currently sends all devices in one segment, so always true.
deviceList	List of Structures	List of device identifiers for the queried device. Device list is sorted. Desk phone first, SIP second. Making SIP desk phone on top. User name is replaced by deviceModelName in this context.

Positive Acknowledgment

There is no positive acknowledgment associated with this service request.

Negative Acknowledgment

This request is sent by MX-ONE and therefore MX-ONE does not send any acknowledgments.

5.1.1.6 Get Switching Function Capabilities

The Get Switching Function Capabilities service is used by the computing function to obtain the current set of capabilities for the entire switching function.

Note:

Positive acknowledgment does not include all mandatory parameters. Only, stated parameters are sent.

Service Request

No parameters are expected.

Positive Acknowledgment

Table 9: Positive acknowledgment

Parameter	Type	Description
switchingSubDomainName	Character (64)	<p>Specifies the name of switching subdomain which distinguishes it from other switching subdomains.</p> <p>Format: station id:MX-ONE version</p> <p>Example, Master SN:7.3.1.0.13</p> <p>Read it like Version: 7.3 SP1 HF0 RC13</p>
manufacturerName	Character (64)	Specifies the name of the manufacturer of the switching sub-domain.

Negative Acknowledgment

Standard error codes as described in [1].

5.2 System Services

5.2.1 Services

5.2.1.1 Request System Status

The Request System Status service is used to obtain (i.e. query) the system status of its peer function. MX-ONE does not send this request but supports receiving it.

Service Request

No parameters are expected.

Positive Acknowledgment

Table 10: Positive acknowledgment

Parameter	Type	Description
systemStatus	Enumerated	-
	Normal	Operating under normal conditions.
	partiallyDisabled	Server/LIM(s) isolated, blocked.
	overloadImminen	Load regulation reports Yellow level. Some traffic is blocked.
	overloadReached	Load regulation reports Red level. All traffic is blocked.

Negative Acknowledgment

Standard error codes as described in [1].

5.2.1.2 System Status

The System Status service is used to report own status to its peer function. MX-ONE uses this service as a heartbeat mechanism if set by configuration.

Service Request

Table 11: Service request

Parameter	Enumerated	-
systemStatus	Normal	Operating under normal conditions.
	partiallyDisabled	Server/LIM(s) isolated, blocked.

	overloadImminen	Load regulation reports Yellow level. Some traffic is blocked.
	overloadReached	Load regulation reports Red level. All traffic is blocked.

Positive Acknowledgment

No parameters are sent or expected. If heartbeat mechanism is set by configuration and no response is received. Session will be closed. Application deemed not available.

Negative Acknowledgment

Standard error codes as described in [1].

5.3 Monitoring Services

5.3.1 Services

5.3.1.1 Change Monitor Filter

The Change Monitor Filter service is used to modify the events that are filtered out (not sent) for an existing monitor.

Service Request

Table 12: Service request

Parameter	Type	Description
crossRefIdentifier	MonitorCrossRefID	The monitor for which to change the filter.
requestedFilterList	MonitorFilter	The monitor filter to be used.

Positive Acknowledgment

The positive acknowledgment does not contain any parameters.

Negative Acknowledgment

Standard error codes as described in [1].

5.3.1.2 Monitor Start

The Monitor Start service initiates event reports (otherwise known as events) for a call, device, or for one or more calls involving a device.

The server starts a monitor, allocates a Monitor Cross Reference Identifier that uniquely identifies the monitor, and then positively acknowledges the request.

MX-ONE supports monitoring of calls, devices and trunks (external lines); however, there is no requirement to start a monitor before a service request can be made.

These event reports stop when the monitor is removed. The monitor can be removed by a service request ([Monitor Stop](#)) or it can automatically be removed by MX-ONE.

Once the monitor is terminated, the monitor cross reference ID is no longer valid.

Note:

That MX-ONE will always respond with monitorExistingCalls set to false. Indicates event reports will not be provided for calls that are at the device at the time of the acknowledgment. And, the event will come at the next action. To get current status, use Snapshot device request.

Service Request

Table 13: Service request

Parameter	Type	Description
monitorObject	Choice Structure	Specifies the call or device to be monitored. call (ConnectionID) device (DeviceID)
monitorType	Enumerated	Specifies the requested monitor (call or device)
requestedFilterList	MonitorFilter	The monitor filter to be used.
privateData	CSTAPrivateData	Device in switching function representation. See Vendor Specific Extensions .

Positive Acknowledgment

Table 14: Positive acknowledgment

Parameter	Type	Description
crossRefIdentifier	MonitorCrossRefID	The cross-reference identifier for this monitor. This value is unique within this session.
monitorExistingCalls	Boolean	MX-ONE will always send false. Indicates event reports will not be provided for calls that are at the device at the time of the acknowledgment.
privateData	CSTAPrivateData	Type of device, log on number if valid for this type of device, type of phone subtype. Included for device monitors that are not on trunk devices. See Vendor Specific Extensions .

Negative Acknowledgment

Standard error codes as described in [1].

5.3.1.3 Monitor Stop

The Monitor Stop service is used to cancel a previously initiated Monitor Start service.

MX-ONE uses this service to indicate it can no longer provide information for monitors on calls that have ended (call monitoring) and if an extension with a device monitor is removed by configuration.

Service Request

Table 15: Service request

Parameter	Type	Description
crossRefIdentifier	MonitorCrossRefID	The cross reference identifier for the monitor to be removed.

Positive Acknowledgment

The positive acknowledgment does not contain any parameters.

Negative Acknowledgment

Standard error codes as described in [1].

5.4 Snapshot Services

5.4.1 Services

5.4.1.1 Snapshot Call

The Snapshot Call service provides information about the devices participating in a specified call. The information provided includes device identifiers, their connections in the call, and local connection states of the devices in the call as well as call related information.

Service Request

Table 16: Service request

Parameter	Type	Description
snapshotObject	ConnectionID	The call to be snapshot. CallID only connectionID is allowed.

Positive Acknowledgment

Table 17: Positive acknowledgment

Parameter	Type	Description
snapshotData	List of Structures	Information for each endpoint in the call.
callingDevice	CallingDeviceID	Specifies the calling device, not valid in traffic cases with multiple parties, conference/intrusion/... or after such events.
calledDevice	CalledDeviceID	Specifies the called device, not valid in traffic cases with multiple parties, conference/intrusion/... or after such events.
correlatorData	CorrelatorData	The correlator data associated with the call.

Negative Acknowledgment

Standard error codes as described in [1].

5.4.1.2 Snapshot Device

The Snapshot Device service provides information about calls associated with a given device. The information provided identifies each call the device is participating in and the local connection state of the device in that call.

A group with no logged-on members, or no members and queue full is reported as blocked. A group with available members or queue position is reported with state null.

Service Request

Table 18: Service request

Parameter	Type	Description
snapshotObject	DeviceID	The device to be snapshot.
privateData	CSTAPrivateData	Device in switching function representation. See Vendor Specific Extensions .

Positive Acknowledgment

Table 19: Positive acknowledgment

Parameter	Type	Description
snapshotData	List of Structures	Information for each call at a device.

Negative Acknowledgment

Standard error codes as described in [1].

5.5 Call Control Services and Events

5.5.1 Services

5.5.1.1 Accept Call

The Accept Call service causes an offered call to transit from the offered mode to the Ringing or Entering Distribution mode of the alerting state.

Service Request

Table 20: Service request

Parameter	Type	Description
callToBeAccepted	ConnectionID	The connection to be accepted.

Positive Acknowledgment

The positive acknowledgment does not contain any parameters.

Negative Acknowledgment

Standard error codes as described in [1].

5.5.1.2 Alternate Call

The Alternate Call service places an existing active call on hold and then retrieves a previously held call. This service is also used to place an active call on hold and then connect to an alerting or queued call at the same device (i.e., to answer a call-waiting call).

Service Request

Table 21: Service request

Parameter	Type	Description
heldCall	ConnectionID	The held connection at the alternating device.
activeCall	ConnectionID	The active connection at the alternating device.

Positive Acknowledgment

The positive acknowledgment does not contain any parameters.

Negative Acknowledgment

Standard error codes as described in [1].

5.5.1.3 Answer Call

The Answer Call service connects an alerting, queued, or initiated call.

The service is only supported for devices where it is possible for the MX-ONE to either answer the call or order the terminal to answer the call.

Service Request

Table 22: Service request

Parameter	Type	Description
callToBeAnswered	ConnectionID	The connection to be answered.

Positive Acknowledgment

The positive acknowledgment does not contain any parameters.

Negative Acknowledgment

Standard error codes as described in [1].

5.5.1.4 Call Back Call-Related

The Call Back Call-Related service allows a client to request that the calling device retry the call to the called device when the called device is in an appropriate state to accept the call.

Service Request

Table 23: Service request

Parameter	Type	Description
callbackConnection	ConnectionID	The connection at the calling device

Positive Acknowledgment

The positive acknowledgment does not contain any parameters.

Negative Acknowledgment

Standard error codes as described in [1].

5.5.1.5 Camp on Call

The Camp on Call service allows the client to queue a call for a device (that typically is busy) until that device becomes available.

Service Request

Table 24: Service request

Parameter	Type	Description
camponConnection	ConnectionID	The connection at the calling device

Positive Acknowledgment

The positive acknowledgment does not contain any parameters.

Negative Acknowledgment

Standard error codes as described in [1].

5.5.1.6 Clear Call

The Clear Call service releases all devices from an existing call. In the case of a conference call, this results in all devices in the conference call being released from the call.

MX-ONE does not support Clear Call in failed state.

Service Request

Table 25: Service request

Parameter	Type	Description
callToBeCleared	ConnectionID	The connection to be cleared

Positive Acknowledgment

The positive acknowledgment does not contain any parameters.

Negative Acknowledgment

Standard error codes as described in [1].

5.5.1.7 Clear Connection

The Clear Connection service releases a specific device from a call. In the case of a two-party call, this may result in the call being torn down. In the case of a conference call, this results in the specific party being removed from the conference.

The Connection ID provided in the request is released.

Service Request

Table 26: Service request

Parameter	Type	Description
connectionToBeCleared	ConnectionID	The connection to be cleared

Positive Acknowledgment

The positive acknowledgment does not contain any parameters.

Negative Acknowledgment

Standard error codes as described in [1].

5.5.1.8 Conference Call

The Conference Call service provides a conference of an existing held call and another active call at a conferencing device.

The two calls are merged into a single call and the two connections at the conferencing device are resolved into a single connection. The Connection IDs formerly associated with the conferenced connections are released and a new Connection ID for the resulting connection is created.

Service Request

Table 27: Service request

Parameter	Type	Description
heldCall	ConnectionID	The held connection.
activeCall	ConnectionID	The active connection.

Positive Acknowledgment

Table 28: Positive acknowledgment

Parameter	Type	Description
conferenceCall	ConnectionID	The resulting connection to the new call. This is the CallID of the conference and the DeviceID of the conferencing device.

Negative Acknowledgment

Standard error codes as described in [1].

5.5.1.9 Consultation Call

The Consultation Call service places an existing active call at a device on hold and initiates a new call from the same device.

Service Request

Table 29: Service request

Parameter	Type	Description
existingCall	ConnectionID	The active connection.
consultedDevice	DeviceID	The device to be consulted.
privateData	CSTAPrivateData	Bypass active diversion. See Vendor Specific Extensions .

Positive Acknowledgment

Table 30: Positive acknowledgment

Parameter	Type	Description
initiatedCall	ConnectionID	The initial connection to the new call. This is the CallID of the new call and the DeviceID of the consulting device.

Negative Acknowledgment

Standard error codes as described in [1].

5.5.1.10 Deflect Call

The Deflect Call service allows the client to divert a call to another destination.

MX-ONE does not support deflect to multiple devices.

Service Request

Table 31: Service request

Parameter	Type	Description
callToBeDiverted	ConnectionID	The connection to be diverted
newDestination	DeviceID	The device to which the call is to be diverted.
privateData	CSTAPrivateData	How to handle possible queue position, maintain queue position, ringtone. See Vendor Specific Extensions .

Positive Acknowledgment

The positive acknowledgment does not contain any parameters.

Negative Acknowledgment

Standard error codes as described in [1].

5.5.1.11 Dial Digits

The Dial Digits service allows the computing function to perform a dialing sequence that is associated with a call that has already been initiated (i.e., has manually gone off-hook).

Service Request

Table 32: Service request

Parameter	Type	Description
diallingConnection	ConnectionID	The connection, which is dialing the digits.
diallingSequence	DeviceID	The string of digits to be dialled. MX-ONE only supports the diallable digits DeviceID format.

Positive Acknowledgment

The positive acknowledgment does not contain any parameters.

Negative Acknowledgment

Standard error codes as described in [1].

5.5.1.12 Directed Pickup Call

The Directed Pickup Call service moves a specified call and connects it at a new specified destination.

This results in the connection being diverted to a new destination inside the MX-ONE.

The service is only supported for devices where it is possible for the MX-ONE to either answer the call or order the terminal to answer the call.

Service Request

Table 33: Service request

Parameter	Type	Description
callToBePickedUp	ConnectionID	The connection to be picked up.
requestingDevice	DeviceID	The device picking up the call.

Positive Acknowledgment

The positive acknowledgment does not contain any parameters.

Negative Acknowledgment

Standard error codes as described in [1].

5.5.1.13 Group Pickup Call

The Group Pickup Call service request picks up a call that is a member of a pickup group in MX-ONE. This results in the connection being diverted to a new destination inside the MX-ONE.

The difference between this service and the Directed Pickup Call service is that Directed Pickup Call service specifies the actual connection to be picked up whereby the Group Pickup Call service does not.

The service is only supported for devices where it is possible for the MX-ONE to either answer the call or order the terminal to answer the call.

Device with a parked call can not request Group Pickup Call.

Service Request

Table 34: Service request

Parameter	type	description
newDestination	DeviceID	The device to pick up the call.
pickGroup	DeviceID	Specifies the pick group.

Positive Acknowledgment

The positive acknowledgment does not contain any parameters.

Negative Acknowledgment

Standard error codes as described in [1].

5.5.1.14 Hold Call

The Hold Call service places a connected connection on hold.

Service Request

Table 35: Service request

Parameter	Type	Description
callToBeHeld	ConnectionID	The connection to be held.

Positive Acknowledgment

The positive acknowledgment does not contain any parameters.

Negative Acknowledgment

Standard error codes as described in [1].

5.5.1.15 Intrude Call

The Intrude Call service adds the calling device to a call at a busy called device. The result will be that the calling device is actively participating in the called device's existing call.

The Connection IDs formerly associated with the calls are released and a new Connection ID for the resulting connection is created. An intrude conference is created.

Service Request

Table 36: Service request

Parameter	Type	Description
intrude	ConnectionID	The connection of the calling device.

Positive Acknowledgment

Table 37: Positive acknowledgment

Parameter	type	description
conferenceCall	ConnectionID	The CallID of the resulting call and the DeviceID of the intruding device.

Negative Acknowledgment

Standard error codes as described in [1].

5.5.1.16 Make Call

The Make Call service allows the computing function to set up a call between a calling device and a called device.

Service Request

Table 38: Service request

Parameter	Type	Description
callingDevice	DeviceID	The calling device.

calledDirectoryNumber	DeviceID	The called device.
privateData	CSTAPrivateData	Bypass active diversion. See Vendor Specific Extensions .

Positive Acknowledgment

Table 39: Positive acknowledgment

Parameter	Type	Description
callingDevice	ConnectionID	The connection to the new call.

Negative Acknowledgment

Standard error codes as described in [1].

5.5.1.17 Reconnect Call

The Reconnect Call service will clear a specified connection at the reconnecting device and retrieve a specified held connection at the same device.

Service Request

Table 40: Service request

Parameter	Type	Description
activeCall	ConnectionID	The connection to be cleared.
heldCall	ConnectionID	The held connection which will be retrieved.

Positive Acknowledgment

The positive acknowledgment does not contain any parameters.

Negative Acknowledgment

Standard error codes as described in [1].

5.5.1.18 Retrieve Call

The Retrieve Call service connects a specified held connection.

Service Request

Table 41: Service request

Parameter	Type	Description
callToBeRetrieved	ConnectionID	The connection to be retrieved.

Positive Acknowledgment

The positive acknowledgment does not contain any parameters.

Negative Acknowledgment

Standard error codes as described in [1].

5.5.1.19 Single Step Transfer Call

The Single Step Transfer Call service transfers an existing connection at a device to another device.

This device doing the transfer does not have to place the existing call on hold before issuing the Single Step Transfer Call service.

Service Request

Table 42: Service request

Parameter	Type	Description
activeCall	ConnectionID	The connection to be transferred.
transferredTo	DeviceID	The device to which the call is to be transferred to.

Positive Acknowledgment

Table 43: Positive acknowledgment

Parameter	Type	Description
transferredCall	ConnectionID	The connectionID of the transferred to device in the resulting call.

Negative Acknowledgment

Standard error codes as described in [1].

5.5.1.20 Transfer Call

The Transfer Call service transfers a call held at a device to an active call at the same device. The held and active calls at the transferring device shall be merged into a new call. Also, the Connections of the held and active calls at the transferring device shall become Null and their ConnectionIDs shall be released (i.e., the transferring device is no longer involved with the call).

Transfer calls to an intruded conference or a conference will not result in a transferred event. It will cause a conferenced event with the transferred party as an added party.

Clear Connection will remove the transferring party from the conference.

Service Request

Table 44: Service request

Parameter	Type	Description
heldCall	ConnectionID	The held connection.
activeCall	ConnectionID	The active connection.

Positive Acknowledgment

Table 45: Positive acknowledge

Parameter	Type	Description
transferredCall	ConnectionID	The connectionID of the transferred to device in the resulting call.

Negative Acknowledgment

Standard error codes as described in [1].

5.5.2 Events

There is a configurable option to add username to the device Id. When name is added, device id is sent in Switching Function Representation. Name is sent in type '*SubjectDeviceID*', '*CallingDeviceID*' and '*CalledDeviceID*'.

Name will only be added when number is available and is not restricted. When name is not available, no name or text added; number is sent as normal. When name is restricted 'Anonymous' will be sent as name.

Configurable option to set number presentation restriction override. When device has number restriction set can this category override this restriction and show the number anyway.

5.5.2.1 Call Cleared

This event indicates that a call is cleared and no longer exists. The Call Cleared event is only provided for calls that are being call-type monitored.

Event Parameters

Table 46: Event parameters

Parameter	Type	Description
monitorCrossRefID	MonitorCrossRefID	Associates the event to existing monitor.
clearedCall	ConnectionID	The connectionID of the cleared call. This is a CallID only connectionID.
correlatorData	CorrelatorData	The correlator data associated with the call.
Cause	EventCause	The reason for the event.

Event Causes

The cause will always be Normal Clearing.

5.5.2.2 Conferenced

The Conferenced event indicates that the conferencing device has conferenced itself or another device with an existing call and that no devices have been removed from the resulting call.

Transfer calls to an intruded conference or a conference will not result in a transferred event. It will cause a conferenced event with the transferred party as an added party.

Clear Connection will remove the transferring party from the conference.

Event Parameters

Table 47: Event parameters

Parameter	Type	Description
monitorCrossRefID	MonitorCrossRefID	Associates the event to existing monitor.

primaryOldCall	ConnectionID	The connectionID of the primary call.
secondaryOldCall	ConnectionID	The connectionID of the secondary call. Always present when fixed view is used. If local view is used included when there are two known calls at the monitored device before the conference is created.
conferencingDevice	SubjectDeviceID	The conferencing device.
addedParty	SubjectDeviceID	The added device.
conferenceConnections	ConnectionList	Information about each device/ ConnectionID in the resulting conference call.
localConnectionInfo	LocalConnectionState	Local connection state of the device associated with the monitor cross reference ID. Only included for device monitors.
correlatorData	CorrelatorData	The correlator data associated with the call.
Cause	EventCause	The reason for the event.

Event Causes

The default cause is Conference; other possible value is:

- Override

5.5.2.3 Connection Cleared

The Connection Cleared event indicates that a single device has disconnected or dropped out of a call.

Event Parameters

Table 48: Event parameters

Parameter	Type	Description
-----------	------	-------------

monitorCrossRefID	MonitorCrossRefID	Associates the event to existing monitor.
droppedConnection	ConnectionID	The connectionID of the device that was dropped from the call.
releasingDevice	SubjectDeviceID	The device that dropped from the call.
localConnectionInfo	LocalConnectionState	Local connection state of the device associated with the monitor cross reference ID. For the clearing device null. For other devices, unaffected. Only included for device monitors.
correlatorData	CorrelatorData	The correlator data associated with the call.
Cause	EventCause	The reason for the event.

Event Causes

The default cause is Normal Clearing; other possible values are:

- Call Back
- Multiple Alerting
- Call Not Answered

MX-ONE will send multiple alerting in event when user has feature forking active on the number. Regardless if there are more terminals active or not.

5.5.2.4 Delivered

The Delivered event indicates that a call is being presented to a device in either the Ringing or Entering Distribution modes of the alerting state.

Event Parameters

Table 49: Event parameters

Parameter	Type	Description
-----------	------	-------------

monitorCrossRefID	MonitorCrossRefID	Associates the event to existing monitor.
Connection	ConnectionID	The connectionID of the device that is alerting.
alertingDevice	SubjectDeviceID	The device that is alerting.
callingDevice	CallingDeviceID	The calling device.
calledDevice	CalledDeviceID	The originally called device.
lastRedirectionDevice	RedirectionDeviceID	The previously known redirected from device.
localConnectionInfo	LocalConnectionState	Local connection state of the device associated with the monitor cross reference ID. For the alerting device Alerting. For other devices, unaffected. Only included for device monitors.
correlatorData	CorrelatorData	The correlator data associated with the call.
cause	EventCause	The reason for the event.
associatedCallingDevice	AssociatedCallingDeviceID	The trunk identity if the call is an incoming external call.
associatedCalledDevice	AssociatedCalledDeviceID	The trunk identity if the call is an outgoing external call.

Event Causes

The default cause is New Call; other possible values are:

- Call Back

- Call Forward—Immediate
- Call Forward—Busy
- Call Forward—No Answer
- Distributed
- Multiple Alerting
- Recall
- Redirected
- Single Step Transfer

MX-ONE will send multiple alerting in event when user has feature forking active on the number. Regardless if there are more terminals active or not.

5.5.2.5 Diverted

The Diverted event indicates that a call has been diverted from a device and that the call is no longer present at the device.

Common situations that generate this event include:

- A call leaves a device due to diversion on no answer.
- A call leaves an ACD/Hunt group device to be delivered to a member.
- A divert (Deflect, Pick, etc.) is successfully invoked.

Event Parameters

Table 50: Event parameters

Parameter	Type	Description
monitorCrossRefID	MonitorCrossRefID	Associates the event to existing monitor.
connection	ConnectionID	The connection that was diverted.
divertingDevice	SubjectDeviceID	The device from which the call was diverted.
newDestination	SubjectDeviceID	The device to which the call was diverted.
callingDevice	CallingDeviceID	The device remaining in the call with the new destination device.
calledDevice	CalledDeviceID	The originally called device.

lastRedirectionDevice	RedirectionDeviceID	The previously known redirected from device.
localConnectionInfo	LocalConnectionState	Local connection state of the device associated with the monitor cross reference ID. For the diverting device null. For other devices, unaffected. Only included for device monitors.
correlatorData	CorrelatorData	The correlator data associated with the call.
Cause	EventCause	The reason for the event.
associatedCallingDevice	AssociatedCallingDeviceID	The trunk identity if the call is an incoming external call.
associatedCalledDevice	AssociatedCalledDeviceID	The trunk identity if the call is an outgoing external call.

Event Causes

The default cause is Normal; other possible values are:

- Call Pickup
- Call Forward—Immediate
- Call Forward—Busy
- Call Forward—No Answer
- Distributed
- Redirected
- Single Step Transfer

5.5.2.6 Established

The Established event indicates that a call has been answered at a device or that a call has been connected to a device.

Event Parameters

Table 51: Event Parameters

Parameter	Type	Description
monitorCrossRefID	MonitorCrossRefID	Associates the event to existing monitor.
establishedConnection	ConnectionID	The connection that was connected.
answeringDevice	SubjectDeviceID	The device the connected to the call.
callingDevice	CallingDeviceID	The calling device.
calledDevice	CalledDeviceID	The originally called device.
lastRedirectionDevice	RedirectionDeviceID	The previously known redirected from device.
localConnectionInfo	LocalConnectionState	Local connection state of the device associated with the monitor cross reference ID. For the answering device connected. For other devices, unaffected (never null). Only included for device monitors.
correlatorData	CorrelatorData	The correlator data associated with the call.
cause	EventCause	The reason for the event.
associatedCallingDevice	AssociatedCallingDeviceID	The trunk identity if the call is an incoming external call.
associatedCalledDevice	AssociatedCalledDeviceID	The trunk identity if the call is an outgoing external call.

Event Causes

The default cause is Normal; other possible values are:

- Call Pickup
- Call Forward—Immediate
- Call Forward—Busy
- Call Forward—No Answer
- Distributed
- New Call

5.5.2.7 Failed

The Failed event indicates that a call cannot be completed, or a connection has entered the Fail state.

Failed event with cause busy will be sent to both calling and called device. Called busy device is the failing device. A connection cleared event is sent immediately for the failing device.

Event Parameters

Table 52: Event parameters

Parameter	Type	Description
monitorCrossRefID	MonitorCrossRefID	Associates the event to existing monitor.
failedConnection	ConnectionID	The connection that failed.
failingDevice	SubjectDeviceID	The device that failed.
callingDevice	CallingDeviceID	The calling device.
calledDevice	CalledDeviceID	The originally called device.
lastRedirectionDevice	RedirectionDeviceID	The previously known redirected from device.

localConnectionInfo	LocalConnectionState	Local connection state of the device associated with the monitor cross reference ID. For the failing device fail. For other devices, unaffected. Only included for device monitors.
correlatorData	CorrelatorData	The correlator data associated with the call.
cause	EventCause	The reason for the event.
associatedCallingDevice	AssociatedCallingDeviceID	The trunk identity if the call is an incoming external call.
associatedCalledDevice	AssociatedCalledDeviceID	The trunk identity if the call is an outgoing external call.

Event Causes

The default cause is Normal; other possible values are:

- Blocked
- Busy
- Do Not Disturb
- Network Congestion
- Not Supported Bearer Service
- Invalid Account Code
- Destination Out of Order
- Lockout
- Call Not Answered
- Invalid Number Format
- Number Unallocated

5.5.2.8 Held

The Held event indicates that a call has been placed on hold.

Event Parameters

Table 53: Event parameters

Parameter	Type	Description
monitorCrossRefID	MonitorCrossRefID	Associates the event to existing monitor.
heldConnection	ConnectionID	The connection at which the hold was activated.
holdingDevice	SubjectDeviceID	The device at which the hold was activated.
localConnectionInfo	LocalConnectionState	Local connection state of the device associated with the monitor cross reference ID. For the holding device Hold. For other devices, unaffected. Only included for device monitors.
correlatorData	CorrelatorData	The correlator data associated with the call.
cause	EventCause	The reason for the event.

Event Causes

The cause will always be Normal.

5.5.2.9 Network Reached

The Network Reached event indicates that a call has reached a Network Interface Device (e.g. trunk).

Event Parameters

Table 54: Event parameters

Parameter	Type	Description
-----------	------	-------------

monitorCrossRefID	MonitorCrossRefID	Associates the event to existing monitor.
outboundConnection	ConnectionID	The connection associated with the outgoing call.
networkInterfaceUsed	SubjectDeviceID	The selected network device (trunk).
callingDevice	CallingDeviceID	The calling device.
calledDevice	CalledDeviceID	The originally called device.
lastRedirectionDevice	RedirectionDeviceID	The previously known redirected from device.
localConnectionInfo	LocalConnectionState	Local connection state of the device associated with the monitor cross reference ID. For the networking interface device Connected. For other devices, unaffected. Only included for device monitors.
correlatorData	CorrelatorData	The correlator data associated with the call.
cause	EventCause	The reason for the event.
associatedCallingDevice	AssociatedCallingDeviceID	The trunk identity if the call is an incoming external call.

Event Causes

The cause will always be Normal.

5.5.2.10 Offered

The Offered event indicates that the connection is in the Offered mode of the Alerting state. This indicates that a call is in a pre-delivery state.

In this pre-delivery state, the opportunity exists to issue one of a set of supported services (e.g. Accept Call, Clear Connection (“reject”), Deflect Call) to accept or reject the call.

Event Parameters

Table 55: Event parameters

Parameter	Type	Description
monitorCrossRefID	MonitorCrossRefID	Associates the event to existing monitor.
offeredConnection	ConnectionID	The connection that is alerting.
offeredDevice	SubjectDeviceID	The device that is alerting.
callingDevice	CallingDeviceID	The calling device.
calledDevice	CalledDeviceID	The originally called device.
lastRedirectionDevice	RedirectionDeviceID	The previously known redirected from device.
localConnectionInfo	LocalConnectionState	Local connection state of the device associated with the monitor cross reference ID. For the alerting device Alerting. For other devices, unaffected. Only included for device monitors.
correlatorData	CorrelatorData	The correlator data associated with the call.
cause	EventCause	The reason for the event.

associatedCallingDevice	AssociatedCallingDeviceID	The trunk identity if the call is an incoming external call.
associatedCalledDevice	AssociatedCalledDeviceID	The trunk identity if the call is an outgoing external call.

Event Causes

The default cause is Normal; other possible values are:

- Call Forward—Immediate
- Call Forward—Busy
- Call Forward—No Answer
- New Call
- Recall
- Redirected
- Single Step Transfer

5.5.2.11 Originated

The Originated event indicates that a call is being attempted from a device.

Event Parameters

Table 56: Event parameters

Parameter	Type	Description
monitorCrossRefID	MonitorCrossRefID	Associates the event to existing monitor.
originatedConnection	ConnectionID	The connection at which the call originated.
callingDevice	CallingDeviceID	The calling device.
calledDevice	CalledDeviceID	The originally called device.
originatingDevice	DeviceID	The originating device.

localConnectionInfo	LocalConnectionState	Local connection state of the device associated with the monitor cross reference ID. For the device making the call Connected. Only included for device monitors.
correlatorData	CorrelatorData	The correlator data associated with the call.
cause	EventCause	The reason for the event.
associatedCallingDevice	AssociatedCallingDeviceID	The trunk identity if the call is an incoming external call.
associatedCalledDevice	AssociatedCalledDeviceID	The trunk identity if the call is an outgoing external call.

Event Causes

The default cause is New Call; other possible value is:

- Call Back

5.5.2.12 Queued

The Queued event indicates that a call has been queued.

Event Parameters

Table 57: Event parameters

Parameter	Type	Description
monitorCrossRefID	MonitorCrossRefID	Associates the event to existing monitor.
queuedconnection	ConnectionID	The queued connection.
queue	SubjectDeviceID	The queue device.

callingDevice	CallingDeviceID	The calling device.
calledDevice	CalledDeviceID	The originally called device.
lastRedirectionDevice	RedirectionDeviceID	The previously known redirected from device.
localConnectionInfo	LocalConnectionState	Local connection state of the device associated with the monitor cross reference ID. For the queue device Queued. For other devices, unaffected. Only included for device monitors.
correlatorData	CorrelatorData	The correlator data associated with the call.
cause	EventCause	The reason for the event.
associatedCallingDevice	AssociatedCallingDeviceID	The trunk identity if the call is an incoming external call.
associatedCalledDevice	AssociatedCalledDeviceID	The trunk identity if the call is an outgoing external call.

Event Causes

The default cause is Normal; other possible values are:

- Call Forward—Immediate
- Call Forward—Busy
- Call Forward—No Answer
- No Available Agents
- Redirected

5.5.2.13 Retrieved

The Retrieved event indicates that a previously held call has been retrieved.

Event Parameters

Table 58: Event parameters

Parameter	Type	Description
monitorCrossRefID	MonitorCrossRefID	Associates the event to existing monitor.
retrievedConnection	ConnectionID	The connection at which hold was deactivated.
retrievingDevice	SubjectDeviceID	The device at which the hold was activated.
localConnectionInfo	LocalConnectionState	Local connection state of the device associated with the monitor cross reference ID. For the retrieving device Connected. For other devices, unaffected. Only included for device monitors.
correlatorData	CorrelatorData	The correlator data associated with the call.
cause	EventCause	The reason for the event.

Event Causes

The cause will always be Normal.

5.5.2.14 Service Initiated

The Service Initiated event indicates that a telephony service has been initiated at a monitored device. MX-ONE typically generates this event when “dial-tone” is being provided.

Event Parameters

Table 59: Event parameters

Parameter	Type	Description
-----------	------	-------------

monitorCrossRefID	MonitorCrossRefID	Associates the event to existing monitor.
initiatedConnection	ConnectionID	The connection at which the service was initiated.
initiatingDevice	SubjectDeviceID	The initiating device.
localConnectionInfo	LocalConnectionState	Local connection state of the device associated with the monitor cross reference ID. Only included for device monitors.
correlatorData	CorrelatorData	The correlator data associated with the call.
cause	EventCause	The reason for the event.
associatedCallingDevice	AssociatedCallingDeviceID	The trunk identity if the call is an incoming external call.

Event Causes

The default cause is New Call; other possible value is:

- Call Back

5.5.2.15 Transferred

The Transferred event indicates that an existing call has been transferred to another device and the transferring device has been dropped from the call.

Transfer calls to an intruded conference or a conference will not result in a transferred event. It will cause a conferenced event with the transferred party as an added party.

Clear Connection will remove the transferring party from the conference.

Event Parameters

Table 60: Event parameters

Parameter	Type	Description
-----------	------	-------------

monitorCrossRefID	MonitorCrossRefID	Associates the event to existing monitor.
primaryOldCall	ConnectionID	The connectionID of the primary call.
secondaryOldCall	ConnectionID	The connectionID of the secondary call. Always present when fixed-view is used. If local-view is used included when there are two known calls at the monitored device before the transfer.
transferringDevice	SubjectDeviceID	The device that transfers the call.
transferredToDevice	SubjectDeviceID	Transferred to device.
transferredConnections	ConnectionList	Information about each device/ ConnectionID in the resulting transferred call.
localConnectionInfo	LocalConnectionState	Local connection state of the device associated with the monitor cross reference ID. Only included for device monitors.
correlatorData	CorrelatorData	The correlator data associated with the call.
cause	EventCause	The reason for the event.

Event Causes

The default cause is Normal; other possible values are:

- Transfer
- Network Signal
- Path Replacement

5.6 Call Associated Features

5.6.1 Services

5.6.1.1 Associate Data

The Associate Data service associate information (correlator data, account code or authorization code) with a specified call.

MX-ONE only supports adding one piece of information in each request.

MX-ONE does not support removal of correlator data, but existing correlator data can be overwritten.

MX-ONE only support digits '0, 1, 2, 3, 4, 5, 6, 7, 8, 9' in accountCode and authCode. In correlatorData, these characters '0, 1, 2, 3, 4, 5, 6, 7, 8, 9, *, #, A, B' supported.

Service Request

Table 61: Service request

Parameter	Type	Description
existingCall	ConnectionID	The call with which to associate the information.
accountCode	AccountInfo	The account code to associate with the call.
authCode	AuthCode	The authorization code to allow the call.
correlatorData	CorrelatorData	The correlator data to associate with the call.

Positive Acknowledgment

The positive acknowledgment does not contain any parameters.

Negative Acknowledgment

Standard error codes as described in [1].

5.6.1.2 Generate Digits

The Generate Digits service causes a series of digits to be sent on behalf of a connection in a call.

Service Request

Table 62: Service request

Parameter	Type	Description
connectionToSendDigits	ConnectionID	The connection of the device which is generating the digits.
charactersToSend	Characters (64)	The characters to send. Allowed digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, *, #.

Positive Acknowledgment

The positive acknowledgment does not contain any parameters.

Negative Acknowledgment

Standard error codes as described in [1].

5.6.2 Events

5.6.2.1 Call Information

The Call Information event indicates that call associated information has been updated for a call.

MX-ONE sends this event when account code or correlator data has been changed.

Event Parameters

Table 63: Event parameters

Parameter	Type	Description
monitorCrossRefID	MonitorCrossRefID	Associates the event to existing monitor.
connection	ConnectionID	The connection of the device responsible for associating the information with the call.

device	SubjectDeviceID	The device responsible for associating the information with the call.
accountInfo	AccountInfo	The account code associated with the call.
correlatorData	CorrelatorData	The correlator data associated with the call.

5.6.2.2 Digits Generated

The Digits Generated event indicates that DTMF digits have been generated at a device.

Event Parameters

Table 64: Event parameters

Parameter	Type	Description
monitorCrossRefID	MonitorCrossRefID	Associates the event to existing monitor.
connection	ConnectionID	The connection of the device.
digitGeneratedList	Characters(64)	The digits generated.

5.7 Physical Device Features

5.7.1 Services

5.7.1.1 Get Message Waiting Indicator

The Get Message Waiting Indicator service provides the message waiting feature status at a specified device.

Service Request

Table 65: Service request

Parameter	Type	Description
-----------	------	-------------

device	DeviceID	The device.
--------	----------	-------------

Positive Acknowledgment

Table 66: Positive acknowledgment

Parameter	Type	Description
messageWaitingOn	Boolean	False – message waiting off. True – message waiting on.

Negative Acknowledgment

Standard error codes as described in [1].

5.7.1.2 Set Message Waiting Indicator

The Set Message Waiting Indicator service allows a computing function to control the status of the message waiting feature at a specified device. The message waiting feature is typically used to notify a user (typically via a dedicated lamp on a phone device) when messages are available.

Service Request

Table 67: Service request

Parameter	Type	Description
device	DeviceID	The device.
messageWaitingOn	Boolean	False – message waiting off. True – message waiting on.
deviceForMessage	DeviceID	The device where the message is waiting.

Positive Acknowledgment

The positive acknowledgment does not contain any parameters.

Negative Acknowledgment

Standard error codes as described in [1].

5.7.2 Events

5.7.2.1 Message Waiting

The Message Waiting event indicates that the message waiting status has been changed for a device.

Event Parameters

Table 68: Event parameters

Parameter	Type	Description
monitorCrossRefID	MonitorCrossRefID	Associates the event to existing monitor.
targetDevice	SubjectDeviceID	The device where the message waiting status has changed.
messageWaitingOn	Boolean	False – message waiting off. True – message waiting on.

5.8 Logical Device Features

5.8.1 Services

5.8.1.1 Get Agent State

The Get Agent State service provides the agent state at a specified device. All parameters are mandatory in the request.

MX-ONE supports Agent Log On to a PBX Hunt group (Explicit/One Step Model).

Service Request

Table 69: Service request

Parameter Name	Type	Description
device	DeviceID	Group member
acdGroup	DeviceID	Group number

Positive Acknowledgment

Table 70: Positive acknowledgment

Parameter Name	Type	Description
agentStateList	List of structures	<p>This parameter specifies a list of agent identifiers, and hunt groups for a given device.</p> <ul style="list-style-type: none"> agentID - Indicates the deviceID. loggedOnState - Indicates the logged on state of the agent. agentInfo List of Structures - A specific device may be associated with one or more agent states. The following components are associated with each device state: <ul style="list-style-type: none"> acdGroup agentState

Supported agent states are Agent Busy, Agent Not Ready, Null, and Agent Ready.

Negative Acknowledgment

Standard error codes as described in [1].

5.8.1.2 Cancel Call Back

The Cancel Call Back service allows the computing function to cancel a previous (or all) Call Back feature at a device.

Service Request

Table 71: Service request

Parameter	Type	Description
originatingDevice	DeviceID	The device which initiated the call back.
targetDevice	DeviceID	The deviceID of the target.

Positive Acknowledgment

The positive acknowledgment does not contain any parameters.

5.8.1.3 Get Do Not Disturb

The Get Do Not Disturb service provides the do not disturb feature status at a specified device.

Service Request

Table 72: Service request

Parameter	Type	Description
device	DeviceID	The device.

Positive Acknowledgment

Table 73: Positive acknowledgment

Parameter	Type	Description
doNotDisturbOn	Boolean	False – Do not disturb is not active. True – Do not disturb is active.

Negative Acknowledgment

Standard error codes as described in [1].

5.8.1.4 Get Forwarding

The Get Forwarding service provides the forwarding feature status at a specified device. The status returned may consist of one or more forwarding types that are active at the specified device based on user defined conditions. Private data describes how the service will be presented on own idle display and on calling party display.

Possible values are:

- noNotification. No forwarding information is visible on display.
- notificationWithServiceOnly. Information that the device is forwarded.
- notificationWithServiceAndToUserNumber. Information that the call is forwarded with forward to number.

Service Request

Table 74: Service request

Parameter	Type	Description
device	DeviceID	The device.

Positive Acknowledgment

Service restriction information is sent in the response, ICS information may also be included when it is valid.

Table 75: Positive acknowledgment

Parameter	Type	Description
forwardList	List of Structures	One structure per forwarding type. MX-ONE supports forwardImmediate, forwardNoAns and forwardBusy.
privateData	CSTAPrivateData	Service restriction information and when valid ICS information. See Vendor Specific Extensions .

Negative Acknowledgment

Standard error codes as described in [1].

5.8.1.5 Set Agent State

The Set Agent State service requests a new agent state for a specified device. In the case where a Hunt group member is involved with a call, the transition to the requested state may or may not occur until the current connection transitions to the null state. All parameters are mandatory in the request.

Service Request

Table 76: Service request

Parameter Name	Type	Description
device	DeviceID	Group member
requestedAgentState	Enumerated	Only logged on and logged off is supported.
group	DeviceID	Group number

Positive Acknowledgment

The positive acknowledgment does not contain any parameters.

Negative Acknowledgment

Standard error codes as described in [1].

5.8.1.6 Set Do Not Disturb

The Set Do Not Disturb service controls the DND feature at a specified device. Active DND is stored in semi-permanent data.

Service Request

Table 77: Service request

Parameter	Type	Description
device	DeviceID	The device.
doNotDisturbOn	Boolean	False – Do not disturb is not active. True – Do not disturb is active.

Positive Acknowledgment

The positive acknowledgment does not contain any parameters.

Negative Acknowledgment

Standard error codes as described in [1].

5.8.1.7 Set Forwarding

Active Forwarding data is stored in semi-permanent data.

The Set Forwarding service allows control of the forwarding feature for a specified device. This service allows only one user-specified setting (forwarding type/forward-destination combination) to be changed per service invocation.

MX-ONE only supports the forwardDN parameter when the service is forwardImmediate, forwardBusy, and forwardNoAns.

MX-ONE supports ICS diversion, set as forwardImmediate with private data, forwardDn is not supported. Diversion destination is configured in MX-ONE. A CSTA application can override the device COS with another COS set on the CSTA link. For more information, see the Diversion category override section under [CSTA Settings](#).

A CSTA application can override the device COS with another COS set on the CSTA link, see [CSTA Settings](#) on page 2 - **Diversion category override**.

A service restriction category can be set at activation of each forwarding type. This is set in private data and possible values are:

- noNotification - No forwarding information is visible on display.
- notificationWithServiceOnly - Information that the device is forwarded.
- notificationWithServiceAndToUserNumber - Information that the call is forwarded with forward to number (**Default Value**).

When Set Forwarding is used to deactivate forwarding all MX-ONE immediate diversions are removed, including Follow-me, ECF and ICS. By using private data in Set Forwarding request, the user can remove FME/ECF or ICS.

Service Request

Parameter	Type	Description
device	DeviceID	The device.
forwardingType	Enumerated	MX-ONE supports forwardImmediate, forwardNoAns and forwardBusy.
activateForward	Boolean	False – Deactivate forwarding. True – Activate forwarding.
forwardDN	DeviceID	The device to which calls should be forwarded.
privateData	CSTAPrivateData	Set request – Service restriction category. Remove request – Immediate forwarding type to be removed. See Vendor Specific Extensions .

Positive Acknowledgment

The positive acknowledgment does not contain any parameters.

Negative Acknowledgment

Standard error codes as described in [1].

5.8.2 Events

5.8.2.1 Agent Busy

The Agent Busy event indicates that an agent has entered the Busy state. In this state the agent is involved with an existing Hunt group call at a device, even if that call is on hold at the device. Calls directly to group members will not cause this transition.

Event Parameters

Table 78: Event parameters

Parameter	Type	Description
monitorCrossRefID	MonitorCrossRefID	Associates the event to an established monitor.
agentDevice	SubjectDeviceID	Indicates the device at which the agent entered the Agent Busy state
acdGroup	DeviceID	Indicates the hunt group
cause	EventCause	Indicates a reason for the event.

5.8.2.2 Agent Logged Off

The Agent Logged Off event indicates that an agent has logged off a hunt group.

Event Parameters

Table 79: Event parameters

Parameter	Type	Description
monitorCrossRefID	MonitorCrossRefID	Associates the event to an established monitor.
agentDevice	SubjectDeviceID	Indicates the device that has logged off from the group.
acdGroup	DeviceID	Indicates the hunt group that the device has logged off from

cause	EventCause	Indicates the reason for the event.
-------	------------	-------------------------------------

5.8.2.3 Agent Logged On

The Agent Logged On event indicates that an agent has logged on an ACD group or a Hunt group.

Event Parameters

Table 80: Event parameters

Parameter	Type	Description
monitorCrossRefID	MonitorCrossRefID	Associates the event to an established monitor.
agentDevice	SubjectDeviceID	Indicates the device that has logged on to the group
acdGroup	DeviceID	Indicates the hunt group that the device has logged on to
cause	EventCause	Indicates the reason for the event.

5.8.2.4 Agent Not Ready

The Agent Not Ready event indicates that an agent has entered the Agent Not Ready state. In this state, an agent is logged on at a particular device to a hunt group but is not prepared to handle calls that the group distributes. While in this state, an agent may receive calls that are not group calls.

Typical examples of when this event may be generated are:

- Busy with a call to another hunt group
- Call directly from/to the device

Event Parameters

Table 81: Event parameters

Parameter	Type	Description
monitorCrossRefID	MonitorCrossRefID	Associates the event to an established monitor.

agentDevice	SubjectDeviceID	Indicates the device that has entered the Agent Not Ready state.
acdGroup	DeviceID	Indicates the hunt group that the device has logged on to.
cause	EventCause	Indicates the reason for the event.

Supported event cause are normal and forced transition.

5.8.2.5 Agent Ready

The Agent Ready event indicates that an agent has entered the Agent Ready state. In this state, an agent is logged on at a particular device to a hunt group and is prepared to handle calls that the group distributes. While in this state, an agent may receive calls that are not group calls.

Event Parameters

Table 82: Event parameters

Parameter	Type	Description
monitorCrossRefID	MonitorCrossRefID	Associates the event to an established monitor.
agentDevice	SubjectDeviceID	Indicates the device that has entered the Agent Ready state.
acdGroup	DeviceID	Indicates the hunt group that the device has logged on to.
cause	EventCause	Indicates the reason for the event.

5.8.2.6 Do Not Disturb

The Do Not Disturb event indicates that the do not disturb feature has been changed for a device.

Event Parameters

Table 83: Event parameters

Parameter	Type	Description
-----------	------	-------------

monitorCrossRefID	MonitorCrossRefID	Associates the event to existing monitor.
device	SubjectDeviceID	The device where Do Not Disturb (DND) was changed.
doNotDisturbOn	Boolean	False – Do not disturb is not active. True – Do not disturb is active.

5.8.2.7 Forwarding

The Forwarding event indicates that the forwarding feature has been changed for a device.

Event Parameters

Service restriction information is sent in the response, ICS information may be included when it is valid.

Table 84: Event parameters

Parameter	Type	Description
monitorCrossRefID	MonitorCrossRefID	Associates the event to existing monitor.
device	SubjectDeviceID	The device where forwarding status was changed.
forwardingType	Enumerated	The type of forwarding. MX-ONE supports forwardImmediate, forwardNoAns and forwardBusy.
forwardStatus	Boolean	False – The forwarding type is not active. True – The forwarding type is active.
forwardTo	DeviceID	The device to which calls are forwarded.

privateData	CSTAPrivateData	Service restriction information and when valid ICS information. See Vendor Specific Extensions .
-------------	-----------------	---

5.9 Device Maintenance Events

5.9.1 Events

5.9.1.1 Back in Service

The back in service event indicates that the device has been returned to service and is operating normally.

Event Parameters

Table 85: Event parameters

Parameter	Type	Description
monitorCrossRefID	MonitorCrossRefID	Associates the event to existing monitor.
device	SubjectDeviceID	The device that is back in service.
privateData	CSTAPrivateData	Includes: <ul style="list-style-type: none"> • typeOfDevice • cstaSubTypeOfDevice • deviceModelName • ipAddress • logOnNumber See Vendor Specific Extensions .

5.9.1.2 Device Capabilities Changed

The Device Capabilities Changed event indicates that device level information has changed.

MX-ONE uses this event to indicate when a monitor is started on a multi-terminal device and the number of logged on terminals changes.

Event Parameters

Table 86: Event parameters

Parameter	Type	Description
monitorCrossRefID	MonitorCrossRefID	Associates the event to existing monitor.
device	SubjectDeviceID	The device whose information has changed.
cause	EventCause	The reason for the event.
privateData	CSTAPrivateData	Device list. See Vendor Specific Extensions . Device list is sorted. Desk phone first, SIP second. Making SIP desk phone on top. User name is replaced by deviceModelName in this context.

Event Causes

The cause will always be Normal.

5.9.1.3 Out of Service

The Out of Service event indicates that the device has been taken out of service and can no longer accept calls and some categories of CSTA service requests (Call Control services, for example).

Event Parameters

Table 87: Event parameters

Parameter	Type	Description
monitorCrossRefID	MonitorCrossRefID	Associates the event to existing monitor.
device	SubjectDeviceID	The device that is out of service.
cause	EventCause	The reason for the event.

Event Causes

The cause will always be Normal.

5.9.1.4 Partially in Service

The Partially in service event indicates that the device is not fully back in service but not completely out of service. The event is used for groups only. ACD and Hunt group, when all members are busy, but there is at least one available queue position. CTI group, when there is at least one available queue position.

Event Parameters

Table 88: Event parameters

Parameter	Type	Description
monitorCrossRefID	MonitorCrossRefID	Associates the event to existing monitor.
device	SubjectDeviceID	The device (group) that is partially back in service.
cause	EventCause	The reason for the event.

Event Causes

The cause will always be Normal.

5.10 Vendor Specific Extensions Services & Events

5.10.1 Services

5.10.1.1 Escape

The Escape service is used by an implementation to send a non-standardized feature.

The Escape Service is used to change and fetch the active personal number list (all supported extension types) and free on second (digital extensions only).

It allows user to control and fetch status of the flexible public CLI service (generic extension only). Allows to set if it exists a common public number, number set with `extension_option` command, group number, or that user number which must be restricted as users calling line identity when calling to a public destination.

It allows to control and fetch status of terminal selection service (forked terminals only). Allows selecting which forked device shall alert and enables selecting the remote extension's remote number from a list of predefined numbers. Selecting one user-defined remote number. Selecting the remote number is possible when a remote extension exists). The selected remote number is stored as semi-permanent data.

Get Services

- Get flexible public calling line identity
- Get personal number, list number.
- Get terminal selection service data.

Set Services

- Set flexible public calling line identity
- Set personal number, list number.
- Set terminal selection data

Service Request

Table 89: Service request

Parameter	Type	Description
privateData	CSTAPrivateData	See Vendor Specific Extensions .

Positive Acknowledgment

Table 90: Positive acknowledgment

Parameter	Type	Description
privateData	CSTAPrivateData	See Vendor Specific Extensions .

All get service is returning data in the positive response. Set user defined number request send data in the positive response, number added in international format or unknown private depending on configuration.

Negative Acknowledgment

Standard error codes as described in [1].

Possible, Flexible CLI set request, error message:

- OperationErrors, generic
- OperationErrors, invalidDeviceID
- OperationErrors, atLeastOneConditionalParameterNotProvided

Possible, terminal selection service set request, error message:

- OperationErrors, generic

- OperationErrors, serviceNotSupported. Rejected due to configuration
- OperationErrors, invalidDeviceID
- OperationErrors, valueOutOfRange. Number already exists

5.10.2 Events

5.10.2.1 Private Event

The Private Event is used by an implementation to send unsolicited, non-standardized event information using the CSTA protocol.

The private event is used to report changes in active personal number or free on second (digital extensions only).

The private event is used to report changes in the flexible public calling line identity.

The private event is used to report media stream information for IP extension and SIP trunks. The information can also include encryption keys to enable decryption of a media stream if required.

The private event is used to report changes in terminal selection service.

The Private events are sent to all devices in a call pickup group that has activated PGM functionality, (pickup group monitoring).

Event Parameters

Table 91: Event parameters

Parameter	Type	Description
monitorCrossRefID	MonitorCrossRefID	Associates the event to existing monitor.
PrivateData	CSTAPrivateData	See Vendor Specific Extensions .

Application Session Services

6

This chapter contains the following sections:

- [Services](#)
- [Stop Application Session Service](#)
- [Reset Application Session Timer Service](#)
- [Application Session Terminated Service](#)

Application Session services (according to ECMA-354) are used to authenticate, i.e. validate the permission of the CSTA application to connect to the ASP 113's CSTA Phase 3 interface.

The Application Session Services are supported via the XML protocol.

6.1 Services

6.1.1 Start Application Session Service

The Start Application Session service is used to initiate an application session between an application and a server. A globally unique identifier, called a sessionID, is returned in the positive service response that identifies the application session. Once an application session is established, the server must maintain information associated with the application session. When the application session is terminated, the application context information is cleared.

The application session exists until:

- It is stopped by using a Stop Application Session service.
- The session is abnormally terminated by the server as indicated by the Application Session Terminated service (e.g., due to the session timer expiry).

6.1.1.1 Service Request

Table 92: Service request

Parameter	Type	Description
applicationInfo	Sequence	ApplicationID, identifies the application.
applicationSpecificInfo	Sequence	Contains the SessionLoginInfo.
sessionLoginInformation	Character string	Password. See Vendor Specific Extensions .
requestedProtocolVersion	List of character strings	Application protocol version(s) that the application wishes to use for the application association.

requestedSessionDuration (Optional)	Integer value	Length of time in seconds that the application session should be maintained.
-------------------------------------	---------------	--

The sessionDuration timer can be periodically refreshed via the Reset Application Session Timer service. If the requestedSessionDuration is not provided the server will choose a default sessionDuration value, set with command. At timer expire of the session duration, timer is the session terminated.

6.1.1.2 Positive Acknowledgment

Table 93: Positive acknowledgment

Parameter	Type	Description
sessionId	Character string	Unique session identifier.
actualProtocolVersion	Character string	Session protocol version used by the application.
actualSessionDuration	Integer value	Length of time (in seconds) that the application session will be maintained by the server.

6.1.1.3 Negative Acknowledgment

Table 94: Negative acknowledge

Parameter	Type	Description
errorCode	String	Standard error codes as described in [5].

Figure 1: Start Application Session example

```
<?xml version="1.0" encoding="utf-8"?>
<StartApplicationSession xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://www.ecma-international.org/standards/ecma-354/appl_session">
  <applicationInfo>
    <applicationID>MiCollab001</applicationID>
    <applicationSpecificInfo>
      <SessionLoginInfo xmlns:xsd="http://www.w3.org/2001/XMLSchema"
        xmlns="http://www.aastra.com/sn/schemas/csta">
        <password>TJRvgo07sDxlvmt</password>
      </SessionLoginInfo>
    </applicationSpecificInfo>
  </applicationInfo>
  <requestedProtocolVersions>
    <protocolVersion>http://www.ecma-international.org/standards/ecma-323/csta/ed6</protocolVersion>
  </requestedProtocolVersions>
  <requestedSessionDuration>300</requestedSessionDuration>
</StartApplicationSession>
```

6.2 Stop Application Session Service

If the server does not recognize the `sessionID` in the service request, it shall provide a negative service response with the error code `invalidSessionID`.

6.2.1 Service Request

The Stop Application Session service is used by the application to terminate an existing application session.

Table 95: Service request

Parameter	Type	Description
<code>sessionID</code>	Character string	Specifies the globally unique identifier associated with the application session that is being stopped.
<code>sessionEndReason</code>	Character string	Specifies the reason that the application session is being stopped.

6.2.1.1 Positive Acknowledgment

Table 96: Positive acknowledgment

Parameter	Type	Description
None	--	Does not contain any parameter.

6.2.1.2 Negative Acknowledgment

Table 97: Negative acknowledgment

Parameter	Type	Description
<code>errorCode</code>	Character string	Standard error codes as described in [5].

6.3 Reset Application Session Timer Service

The Reset Application Session Timer service is used by the application to reset the duration that an existing application session should be maintained. That is ask session server to maintain the session little longer. If the requestedSessionDuration is not provided, the server will use a default sessionDuration value, set with command.

6.3.1 Service Request

The Reset Application Session Timer Service is used by the application to extend the duration the session shall be kept active.

Table 98: Service request

Parameter	Type	Description
sessionID	Character string	Unique identifier associated with the application session that is being reset.
requestedSessionDuration (Optional)	Integer value	The length of time (in seconds) that the application requests that the server maintain the application session.
sessionLogInfo	Character string	Password. See Vendor Specific Extensions .

6.3.1.1 Positive Acknowledgment

Table 99: Positive acknowledgment

Parameter	Type	Description
actualSessionDuration	Integer value	Length of time (in seconds) that the application session will be maintained by the server.

6.3.1.2 Negative Acknowledgment

Table 100: Negative acknowledgment

Parameter	Type	Description
errorCode	Character string	Standard error codes as specified in [5]

6.4 Application Session Terminated Service

The Application Session Terminated service is sent by a server when the server has terminated an application session.

When the server terminates an application session, it clears all the application context information associated with the application session.

6.4.1 Service Request

Table 101: Service request

Parameter	Type	Description
sessionID	Character string	Unique identifier associated with the application session that is being terminated.
sessionTermReason	Character string	Reason that the application session has been terminated, as described in [5].

6.4.1.1 Service Acknowledgment

There is no acknowledge to this request.

This chapter contains the following sections:

- [Learning About Registered Terminals](#)
- [Monitoring](#)
- [Changes in Registered Terminals](#)
- [Services](#)

For MX-ONE multi-terminal extensions most services require use of the switching function representation device identifier format which specifies the used terminal, rather than just the number. There is minimal support for using the number in the dialable digits format.

7.1 Learning About Registered Terminals

The registered terminals of a multi-terminal user are not static, this can change at any time so there are several methods to learn about the current registered terminals, and to know when anything changes.

To learn about whether this is a multi-terminal extension there are two methods. By starting a monitor as the MonitorStartResponse contains the type of device in private data or by using the GetLogicalDeviceInformation request. The GetLogicalDeviceInformationResponse contains the type of device in private data.

Once this is known, the application has two methods to know about the currently registered terminals for that user, and the sub-address for each one. By using the GetSwitchingFunctionDevices request and waiting for the asynchronous SwitchingFunctionDevices event which contains a device list. By using the GetLogicalDeviceInformation request, issued for a multi-terminal extension and there are registered terminals, then the GetLogicalDeviceInformationResponse will have the hasPhysicalElement set to true and otherPhysicalDeviceList will contain a list of the terminals in Switching Function Representation format like described earlier but with some additional information.

In GetLogicalDeviceInformationResponse and in otherPhysicalDeviceList uses the format:**N<DN!SA/EXT>NM**

Where

- **N** identifies that Device Identifier uses the Switching Function Representation format.
- **< >** are angle brackets encompassing the number and sub-address.
- **DN** is the extension number of the user.
- **!** is a separator between number and sub-address.
- **SA** is the sub-address which is used to identify the specific terminal. This is an integer between 1 and 9.
- **/** represents the start of a physical element extension. This is only present if the physical identifier is present.
- **EXT** is the physical identifier (MAC address) of the device if this is available.
- **NM** is a text label indicating the device model name; for example, 6863i, Dect, Remote, and so on.

It is possible with a single GetLogicalDeviceInformation request to learn whether this is a multi-terminal extension and get a list of the registered terminals.

Figure 2: Multi-terminal

```
<?xml version="1.0" encoding="utf-8"?>
<GetLogicalDeviceInformationResponse xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed5">
  <deviceCategory>station</deviceCategory>
  <namedDeviceTypes>station</namedDeviceTypes>
  <hasPhysicalElement>true</hasPhysicalElement>
  <acdModels>
    <visibleACDRelatedDevices>true</visibleACDRelatedDevices>
    <nonVisibleACDRelatedDevices>false</nonVisibleACDRelatedDevices>
  </acdModels>
  <appearanceAddressable>false</appearanceAddressable>
  <appearanceType>selectedStandard</appearanceType>
  <otherPhysicalDeviceList>
    <deviceID>N&lt;10420!1&gt;Remote</deviceID>
    <deviceID>N&lt;10420!2&gt;DBC42502</deviceID>
    <deviceID>N&lt;10420!3/00085D3F2343&gt;6865i</deviceID>
  </otherPhysicalDeviceList>
  <transAndConfSetup>
    <consultationCall>true</consultationCall>
    <holdCallMakeCall>true</holdCallMakeCall>
    <alternateCall>true</alternateCall>
    <twoCallsInHold>true</twoCallsInHold>
    <twoCallsInConnected>true</twoCallsInConnected>
  </transAndConfSetup>
  <extensions>
    <privateData>
      <private>
        <getLogicalDeviceInformationResponsePrivateData xmlns="http://www.aastra.com/sn/schemas/csta">
          <typeOfDevice>multiTerminalExtension</typeOfDevice>
        </getLogicalDeviceInformationResponsePrivateData>
      </private>
    </privateData>
  </extensions>
</GetLogicalDeviceInformationResponse>
```

7.2 Monitoring

7.2.1 Monitoring the Number

This method is not recommended.

This is when just the number of the extension is monitored. In this case events for all registered terminals will be sent on the same cross reference identifier with no way for an application to know which terminal is reporting which event.

For example, when someone calls to a multi-terminal extension the application will receive up to 4 identical delivered events.

7.2.2 Monitoring the Number with Switching Function Representation

In this case the monitor is started using just the number of the extension but in the monitor start request private data is included that indicates the switching function representation identifier format should be used.

Figure 3: Monitor start

```
<?xml version="1.0" encoding="utf-8"?>
<MonitorStart xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed5">
  <monitorObject>
    <deviceObject>10420</deviceObject>
  </monitorObject>
  <monitorType>device</monitorType>
  <extensions>
    <privateData>
      <private>
        <monitorStartPrivateData xmlns="http://www.aastra.com/sn/schemas/csta">
          <multiTerminalUseSFReprDeviceID>true</multiTerminalUseSFReprDeviceID>
          </monitorStartPrivateData>
        </private>
      </privateData>
    </extensions>
  </MonitorStart>
```

All events for the multi-terminal extension will be sent on the same cross reference identifier but the application will be able to identify the specific terminal the event is being reported for as the device identifier will include the sub-address.

Figure 4: Events

```

<?xml version="1.0" encoding="utf-8"?>
<DeliveredEvent xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed5">
  <monitorCrossRefID>2</monitorCrossRefID>
  <connection>
    <callID>000001502033823010012747</callID>
    <deviceID>N&lt;67923!2&gt;</deviceID>
  </connection>
  <alertingDevice>
    <deviceIdentifier>N&lt;67923!2&gt;</deviceIdentifier>
  </alertingDevice>
  <callingDevice>
    <deviceIdentifier typeOfNumber="dialingNumber">+19725550199</deviceIdentifier>
  </callingDevice>
  <calledDevice>
    <deviceIdentifier typeOfNumber="dialingNumber">67923</deviceIdentifier>
  </calledDevice>
  <lastRedirectionDevice>
    <notSpecified/>
  </lastRedirectionDevice>
  <localConnectionInfo>alerting</localConnectionInfo>
  <cause>multipleAlerting</cause>
  <associatedCallingDevice>
    <deviceIdentifier typeOfNumber="deviceNumber">0010010001</deviceIdentifier>
  </associatedCallingDevice>
</DeliveredEvent>

```

7.2.3 Monitoring the Terminal

The application can just monitor a specific sub-address (terminal) for the user by using the switching function representation device identifier in the start monitor request. When this is done, only events for the specific terminal will be sent on the cross-reference identifier. When using customer data and have long and short number, this method is the only way to get the log on number.

When terminal logs off, monitor will get an out of service event. When it logs on again, it might get another sub-address, where no event is sent as sub-address and not monitored.

7.3 Changes in Registered Terminals

7.3.1 Out of Service Event

When a terminal logs off and the monitor is for the specific sub-address (see [Monitoring the Terminal](#) on page 77) then you will receive this event. If the number is monitored this event is only sent when the last terminal logs off.

Note:

If you are trying to re-establish CSTA connection on the ASC side after CSTA service restart on MX-ONE, you also need to restart CSTA service on the ASC server.

7.3.2 Back in Service Event

When a terminal log in a back-in-service event will be received. If a monitor is started for that specific sub-address, then the event will include data about that terminal will be received in private data. If the monitor is on just the number, then only the type of device (multiTerminalExtension) is included.

If a terminal logs off and logs back in a new sub-address will be assigned. This may or may not be the same as the device had previously.

7.3.3 Device Capabilities Changed Event

When a multi-terminal extension is monitored just by using the number (see [Monitoring the Number](#) on page 75 or [Monitoring the Number with Switching Function Representation](#) on page 76) then this event is sent whenever there is a change in the registered terminals and contains a list of the registered terminals.

Figure 5: Device capability change event

```
<?xml version="1.0" encoding="utf-8"?>
<DeviceCapsChangedEvent xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed5" >
  <monitorCrossRefID>0</monitorCrossRefID>
  <device>
    <deviceIdentifier typeOfNumber="dialingNumber">10420</deviceIdentifier>
  </device>
  <cause>normal</cause>
  <extensions>
    <privateData>
      <private>
        <deviceCapsChangedEventPrivateData xmlns="http://www.aastra.com/sn/schemas/csta">
          <deviceList>
            <deviceID>N&lt;10420!2&gt;Remote</deviceID>
            <deviceID>N&lt;10420!3/00085D3F2343&gt;6865i</deviceID>
          </deviceList>
        </deviceCapsChangedEventPrivateData>
      </private>
    </privateData>
  </extensions>
</DeviceCapsChangedEvent>
```

7.4 Services

Most services for multi-terminal extensions require use of the switching function representation device identifier format including the sub-address with a few exceptions, mainly for services that are handled in the MX-ONE at a user level.

- Set/Get Forwarding
- Set/Get Do Not Disturb
- Set/Get Message Waiting
- Set Display
- Snapshot Device – when executed on the user is will report all calls on all devices
- Monitor Device – this is explained above
- Get Switching Function Devices
- Get Logical Device Information
- Get Physical Device Information
- Set/Get private Active Personal Number List
- Set/Get private Terminal Selection Service
- Set/Get Flexible Public CLI
- Set/Get Agent State

This chapter contains the following sections:

- [Private Data on CSTA Services and Events](#)
- [Back in Service Information](#)
- [Escape Services and Private Event](#)
- [Private Data in CSTA Session Application Services](#)

Private data is used by MX-ONE in several places to allow extra information to be conveyed about the requested service, or the event being reported.

MX-ONE also uses Escape services and Private event to support services and events not defined by [1].

The schema used by MX-ONE is defined in an xsd schema, see [Appendix A](#).

8.1 Private Data on CSTA Services and Events

8.1.1 Services

The following services requests or responses include private data.

- Consultation Call
- Deflect Call
- Get Forwarding Response
- Get Logical Device Information Response
- Get Physical Device Information
- Make Call
- Monitor Start
- Monitor Start Response
- Set Forwarding
- Snapshot Device
- Switching Function Devices

8.1.1.1 Deflect Information

Deflect with maintain queue/maintain queue with ringtone is only allowed for ACD/CTI groups and in following cases.

- Deflect to IVR, voice mail equipment, connected over public trunk
- Deflect to free extension
- Deflect to group with available members
- Deflect to free forked extension

- Services that can be executed on a call deflected with maintain queue/maintain queue with ringtone .
- Individual or group call pickup

8.1.2 Get Logical Device Information Response

For all types of devices, *type of device* and *csta subtype of device* are sent as private data. For logged on IP/SIP extension, its IP address is included in the private data. For SIP extension, that is logged on, is the *terminal log on number* included in the private data.

IP address and terminal log on number is used by recording applications.

8.1.3 Monitor Start Response Information

For all types of devices, *type of device* and *csta subtype of device* are sent as private data. For generic extension is also phone subtype sent. For logged on IP/SIP extension, its IP address is included in the private data. For SIP extension, that is logged on, is the *terminal log on number* included in the private data.

IP address and terminal log on number is used by recording applications.

8.1.4 Events

The following events include private data.

- Back in Service
- Device Capabilities Changed
- Forwarding

8.2 Back in Service Information

For all types of devices is *type of device* and *csta subtype of device* sent as private data. For logged on IP/SIP extension, its IP address is included in the private data. For SIP extension, that is logged on, is the *terminal log on number* included in the private data.

IP address and terminal log on number is used by recording applications.

8.3 Escape Services and Private Event

8.3.1 Services

The Escape Service is used to change and fetch the active personal number list (all supported extension types) and free on second (digital extensions only).

The Escape Services can be used to change and get the terminal selection service, TSS, data (forked user only).

Escape Service Flexible Public CLI can set the user calling line identity to another number than own number (generic extension only). The number can be a defined common public number, a number defined with `extension_option` command, a CTI group number, a group hunt number user is a member of, or user number can be set to restricted.

An example of a request to set a group hunt number as the users flexible public calling line identity.

Figure 6: Group hunt number flexible public calling line identity

```
<?xml version="1.0" encoding="utf-8"?>
<Escape xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed6">
  <privateData>
    <private>
      <escapePrivateData xmlns:xsd="http://www.w3.org/2001/XMLSchema"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns="http://www.aastra.com/sn/schemas/csta">
        <setCallingLineIdentityData>
          <deviceId>10400</deviceId>
          <selectCallingLineIdentityActive>true</selectCallingLineIdentityActive>
          <selectCallingLineIdentityOrder>group</selectCallingLineIdentityOrder>
          <groupNumber typeOfNumber="dialingNumber" mediaClass="notKnown">890</groupNumber>
        </setCallingLineIdentityData>
      </escapePrivateData>
    </private>
  </privateData>
</Escape>
```

An example of a response to get request users flexible public calling line identity.

Figure 7: Users flexible public calling line identity

```
<?xml version="1.0" encoding="utf-8"?>
<EscapeResponse xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed6"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <extensions>
    <privateData>
      <private>
        <escapeResponsePrivateData xmlns="http://www.aastra.com/sn/schemas/csta">
          <getCallingLineIdentityDataResponse>
            <overridingNumberAndName>
              <active>false</active>
              <deviceId>
                <notKnown/>
              </deviceId>
            </overridingNumberAndName>
            <selectedPublicCallingLineIdentity>
              <active>true</active>
              <deviceId>
                <deviceIdIdentifier>N&lt;886&gt;HuntGroup MX-ONE</deviceIdIdentifier>
              </deviceId>
            </selectedPublicCallingLineIdentity>
            <publicCallingLineIdentityList>
              <commonPublicNumber typeOfNumber="dialingNumber">10000</commonPublicNumber>
              <groupHuntNumberList>
                <deviceIdIdentifier>N&lt;886&gt;HuntGroup MX-ONE</deviceIdIdentifier>
                <deviceIdIdentifier>N&lt;889&gt;Test department MX-ONE</deviceIdIdentifier>
                <deviceIdIdentifier>N&lt;887&gt;CascadeGroup MX-ONE</deviceIdIdentifier>
              </groupHuntNumberList>
            </publicCallingLineIdentityList>
          </getCallingLineIdentityDataResponse>
        </escapeResponsePrivateData>
      </private>
    </privateData>
  </extensions>
</EscapeResponse>
```

Escape service TSS request can change (how a forked user's terminals shall alert) only one specific terminal or all terminals shall alert.

Escape service TSS request can select the remote number that shall alert from a list (for forked remote extension only). The selected remote number is stored as semi-permanent data. The list contains predefined remote numbers in MX-ONE including one user-defined remote number, user-defined number is only included when simplified TSS is set to 'NO'. The configuration can set whether the number will be presented in international number type (+xxxx), or in private unknown number type in events and responses.

A user can use the escape service TSS request to set and remove one remote number. This will be added first in the list of selectable numbers. Added user defined number must be entered in private unknown number type, destination code + public number.

TSS multiplicity function can be enabled and disabled by configuration.

An example of a request to select that one terminal shall alert and that a specified remote terminal shall be called.

Figure 8: Private request to set terminal selection service

```
<Escape xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed6">
  <privateData>
    <private>
      <escapePrivateData xmlns="http://www.aastra.com/sn/schemas/csta">
        <setTerminalSelectionServiceData>
          <deviceID>68002</deviceID>
          <selectRemoteExtensionAction>changeToPredefinedNumber</selectRemoteExtensionAction>
          <selectedRemoteExtension typeOfNumber="dialingNumber" mediaClass="notKnown">+460407301</selectedRemoteExtension>
          <deviceAlertList>
            <deviceID mediaClass="notKnown">N<6800211</deviceID>
          </deviceAlertList>
        </setTerminalSelectionServiceData>
      </escapePrivateData>
    </private>
  </privateData>
</Escape>
```

8.3.2 Events

The private event is used to report changes in active personal number or free on second (digital extensions only).

The private event report changes to the users flexible public CLI data.

An example of a flexible public CLI event.

Figure 9: Flexible public CLI event


```

<?xml version="1.0" encoding="utf-8"?>
<PrivateEvent xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed6"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <monitorCrossRefID>4</monitorCrossRefID>
  <privateData>
    <private>
      <privateEventPrivateData xmlns="http://www.aastra.com/sn/schemas/csta">
        <callingLineIdentityEventData>
          <overridingNumberAndName>
            <active>false</active>
            <deviceID>
              <notKnown/>
            </deviceID>
          </overridingNumberAndName>
          <selectedPublicCallingLineIdentity>
            <active>true</active>
            <deviceID>
              <deviceIdentifier>N<886>HuntGroup MX-ONE</deviceIdentifier>
            </deviceID>
          </selectedPublicCallingLineIdentity>
        </callingLineIdentityEventData>
      </privateEventPrivateData>
    </private>
  </privateData>
</PrivateEvent>

```

It is also used to report media stream information for IP extension and SIP trunks. The information can also include encryption keys to enable decryption of a media stream if required.

An example of a media stream event is below.

Figure 10: Private event

```

<?xml version="1.0" encoding="utf-8"?>
<PrivateEvent xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed5">
  <monitorCrossRefID>0</monitorCrossRefID>
  <privateData>
    <private>
      <privateEventPrivateData xmlns="http://www.aastra.com/sn/schemas/csta">
        <mediaStreamInformationData>
          <callID>000000000011115080010003</callID>
          <cryptoSuite>AES_CM_128_HMAC_SHA1_80</cryptoSuite>
          <direction>sendrecv</direction>
          <mediaType>audio</mediaType>
          <thisEndPoint>
            <mediaAddress>
              <ipV4Address>10.16.15.123</ipV4Address>
              <port>3000</port>
            </mediaAddress>
            <masterKey>YGBFVWN9fEA/TCI4a0VcQDVTWlwsaCooWURJe1Rz</masterKey>
          </thisEndPoint>
          <otherEndPoint>
            <mediaAddress>
              <ipV4Address>10.16.15.120</ipV4Address>
              <port>3000</port>
            </mediaAddress>
            <masterKey>TiRybyE3fCVfRnF9N1k+Qmg8VEohU2ZDNGE7a3FG</masterKey>
          </otherEndPoint>
          <sipCallId>c77fff08e94920a8</sipCallId>
        </mediaStreamInformationData>
      </privateEventPrivateData>
    </private>
  </privateData>
</PrivateEvent>

```

The private event reports the new status in Terminal Selection Service data.

An example of a Terminal Selection Service event.

Figure 11: Private event terminal selection service

```
<PrivateEvent xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed6"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <monitorCrossRefID>0</monitorCrossRefID>
  <privateData>
    <private>
      <privateEventPrivateData>
        <terminalSelectionServiceEventData>
          <remoteExtensionsList>
            <deviceIdentifier typeOfNumber="dialingNumber">+460401101</deviceIdentifier>
            <deviceIdentifier typeOfNumber="dialingNumber">+460401201</deviceIdentifier>
            <deviceIdentifier typeOfNumber="dialingNumber">+460407301</deviceIdentifier>
          </remoteExtensionsList>
          <selectedRemoteExtension>
            <deviceIdentifier typeOfNumber="dialingNumber">+460401101</deviceIdentifier>
          </selectedRemoteExtension>
          <AlertingDevice>
            <deviceID>N&lt;68002!2&gt;DBC42502#alert</deviceID>
            <deviceID>N&lt;68002!1&gt;RemoteExtension#alert</deviceID>
            <deviceID>N&lt;68002!3/00085D32E2F4&gt;6737i#alert</deviceID>
          </AlertingDevice>
        </terminalSelectionServiceEventData>
      </privateEventPrivateData>
    </private>
  </privateData>
</PrivateEvent>
```

It can report changes in status of a call pickup group. Event is only sent to members that have a PGM, Pickup Group Monitor, and key programmed at the device.

An example of a Pickup Group Monitoring event is shown below notifying the user that 10701 is calling 10402.

Figure 12: Private event call to call pickup group

```
<PrivateEvent xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed6"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <monitorCrossRefID>4</monitorCrossRefID>
  <privateData>
    <private>
      <privateEventPrivateData>
        <groupCallPickupDelivered>
          <connection>
            <callID>000000673007115541230017</callID>
            <deviceID>10401</deviceID>
          </connection>
          <alertingDevice>
            <deviceIdentifier>N&lt;10402&gt;Easy M4E8</deviceIdentifier>
          </alertingDevice>
          <callingDevice>
            <deviceIdentifier>N&lt;10410&gt;GW Panther</deviceIdentifier>
          </callingDevice>
          <calledDevice>
            <deviceIdentifier>N&lt;10402&gt;Easy M4E8</deviceIdentifier>
          </calledDevice>
          <notificationDelayTime>0</notificationDelayTime>
          <notificationVisibleTime>30</notificationVisibleTime>
          <displayOption>0</displayOption>
        </groupCallPickupDelivered>
      </privateEventPrivateData>
    </private>
  </privateData>
</PrivateEvent>
```

8.4 Private Data in CSTA Session Application Services

8.4.1 Services Containing Private Data

The following service requests or responses include private (proprietary) data:

- Start application session
- Reset application session timer

Supported Services by Device

9

This table shows which of the CSTA services described in this document are supported by the various MX-ONE devices that can be controlled by CSTA. Any CSTA services not referenced should be considered as not supported.

Table 102: Supported services by device

Service	Mitel SIP Desk Phone	OTHER SIP	Group	H323	RXN	CXN	DTS	ATS	Trunk
Get Logical Device Information	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Get Physical Device Information	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Get Switching Function Devices	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Change Monitor Filter	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Monitor Start	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Monitor Stop	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Snapshot Device	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	1)
Accept Call	2)	2)	No	2)	2)	2)	2)	2)	No
Alternate Call	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	No
Answer Call	Yes	3)	No	Yes	No	No	Yes	No	No

Call Back Call-Related	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	No
Camp On Call	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	No
Clear Connection	Yes	Yes	4)	Yes	Yes	Yes	Yes	Yes	No
Conference Call	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	No
Consultation Call	Yes	3)	No	Yes	Yes	Yes	Yes	Yes	No
Deflect Call	Yes	Yes	4)	Yes	Yes	Yes	Yes	Yes	No
Dial Digits	Yes	No	No	No	No	No	No	Yes	No
Directed Pickup Call	Yes	No	No	Yes	6)	Yes	Yes	6)	No
Group Pickup Call	Yes	No	No	Yes	6)	Yes	Yes	6)	No
Hold Call	Yes	3)	No	Yes	Yes	Yes	Yes	Yes	No
Intrude Call	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	No
Make Call	Yes	3)	No	Yes	Yes	Yes	Yes	Yes	No
Reconnect Call	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	No
Retrieve Call	Yes	3)	No	Yes	Yes	Yes	Yes	Yes	No
Single Step Transfer Call	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	No
Transfer Call	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	No

Associate Data	Yes	Yes	4)	Yes	Yes	Yes	Yes	Yes	No
Generate Digits	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	No
Get Message Waiting Indicator	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	No
Set Message Waiting Indicator	Yes	7)	No	Yes	7)	7)	7)	7)	No
Set Display	Yes	7)	No	7)	8)	7)	7)	8)	No
Cancel Call Back	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	No
Get Do Not Disturb	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	No
Get Forwarding	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	No
Set Do Not Disturb	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	No
Set Forwarding	Yes	Yes	9)	Yes	Yes	Yes	Yes	Yes	No
Escape	5)	5)	No	5)	5)	5)	5)	5)	No

1) Not supported on all trunk types.

2) In conjunction with configuration on the extension.

3) Functionality for SIP endpoint varies depending on the terminal. Where possible the service is supported.

4) Only supported on ACD/CTI groups.

5) See description of service for supported devices.

It allows user to control and fetch status of the flexible public CLI service. Allows to set if it exists a common public number, number set with `extension_option` command, group number, or that user number which must be restricted as users calling line identity when calling to an external part (only supported by generic extension).

6) Extension must be in register state (off hook).

7) Service will be accepted, characteristics of the connected terminal control whether there is any indication on the device.

8) Service will be accepted, nothing will be shown on the device.

9) It is 'Yes' in case of 'follow me and external follow me'; 'No' in case of message diversion, direct diversion, diversion on busy, and for diversion on no reply.

There are some configurations which are not supported in the CSTA Phase III interface. Monitoring, or the requested service may not be rejected however the event reporting or service will not be handled correctly.

1. Digital extensions represented on an MDN key on another device.
2. Extensions with shared call appearance (SCA).
3. Customer group, multi-tenant configurations.
4. MX-ONE operators and call origin groups are not supported.
5. There is an extra limitation for digital extensions with additional directory number keys (ADN). Monitoring of these devices is supported but not if transfer or conference is executed with one call on the ODN and the second on the ADN.
6. The Session Application Services do not support allowlist or denylist of application IP addresses.

Appendix A, XML Schema File MX-ONE Private Data

11

This chapter contains the following sections:

- [MX-ONE-CSTA-PRIVATE-DATA.XSD](#)

11.1 MX-ONE-CSTA-PRIVATE-DATA.XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.aastra.com/sn/schemas/csta"
  elementFormDefault="qualified" version="3.3.5">
  <?ezd-metadata
    name="uuid" value="b3f0b661-47af-4fe0-997e-5edf35cbb30a"?>
  <?ezd-metadata
    name="validation-err-msg" value=""?>
  <?ezd-metadata name="char-count"
    value="3694"?>
  <?ezd-metadata name="lock-owner" value=""?>
  <?ezd-metadata
    name="normalized-checksum" value="04629595f88e0587a3ac2f49b357478e"?
  <?ezd-metadata name="is-valid" value=""?>
  <?ezd-metadata name="__source-
    resource-uuid" value="7d1f8720-e6a3-11ee-b869-0242da05c2d4"?
  <?ezd-metadata name="dita-class" value=""?>
  <?ezd-metadata
    name="title" value=""?>
  <?ezd-metadata name="__document_owner"
    value="ajithkumar.emanuvel@mitel.com"?>
  <?ezd-metadata name="num-open-
    comments" value="0"?>
  <?ezd-metadata name="__last_modified_revision"
    value="-1"?>
  <?ezd-metadata name="__document_links_last_processed_rev"
    value="-1"?>
  <?ezd-metadata name="__has_broken_links" value="false"?
  <?ezd-metadata name="dita-domains" value=""?>
  <?ezd-metadata
    name="checksum" value="e70f77991010e7496a5422del6e72c41"?>
  <?ezd-metadata
    name="__checksum-dirty" value="false"?>
  <?ezd-metadata name="content-
    type" value=""?>
  <?ezd-metadata name="word-count" value="7"?>
  <?ezd-
    metadata name="__root-resource-uuid" value="7d1f8720-e6a3-11ee-
    b869-0242da05c2d4"?>
  <?ezd-metadata name="document-valid-md-field"
    value=""?>
  <?ezd-metadata name="__last_modified_by" value="SYSTEM"?
  <?ezd-metadata name="status" value="in_progress"?>
  <?ezd-metadata
    name="uuid" value="56578716-b167-4fbe-aa89-8149d33c7d47"?>
  <?ezd-metadata
    name="owner" value="ajithkumar.emanuvel@mitel.com"?>
  <?ezd-metadata
    name="created" value="1716794853202"?>
  <?ezd-metadata name="last-modified"
    value="1716794853387"?>
  <xs:annotation>
    <xs:documentation>Private data schema for MX-ONE</xs:documentation>
  </xs:annotation>
  <xs:complexType name="crypto">
    <xs:sequence>
      <xs:element name="key" type="mx1:encryptionKey"/>
      <xs:element name="lifeTime" type="xs:positiveInteger" minOccurs="0"/>
      <xs:element name="mki" type="xs:positiveInteger" minOccurs="0"/>
      <xs:element name="mkiLength" type="xs:positiveInteger" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="cryptoSuite">
    <xs:restriction base="xs:string">
      <xs:enumeration value="NO_CRYPT0"/>
      <xs:enumeration value="AES_CM_128_HMAC_SHA1_80"/>
      <xs:enumeration value="AES_CM_128_HMAC_SHA1_32"/>
      <xs:enumeration value="F8_128_HMAC_SHA1_80"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

```

    <xs:enumeration value="AES_CM_128"/>
    <xs:enumeration value="AES_192_CM_HMAC_SHA1_80"/>
    <xs:enumeration value="AES_192_CM_HMAC_SHA1_32"/>
    <xs:enumeration value="AES_256_CM_HMAC_SHA1_80"/>
    <xs:enumeration value="AES_256_CM_HMAC_SHA1_32"/>
    <xs:enumeration value="AEAD_AES_128_GCM"/>
    <xs:enumeration value="AEAD_AES_256_GCM"/>
    <xs:enumeration value="AEAD_AES_128_GCM_8"/>
    <xs:enumeration value="AEAD_AES_256_GCM_8"/>
    <xs:enumeration value="AEAD_AES_128_GCM_12"/>
    <xs:enumeration value="AEAD_AES_256_GCM_12"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="deflectSubService">
  <xs:restriction base="xs:string">
    <!-- noSubService is the default value if subservice is not received
-->
    <xs:enumeration value="noSubService"/>
    <xs:enumeration value="maintainedQueue"/>
    <xs:enumeration value="maintainedQueueWithRingtone"/>
    <xs:enumeration value="reactivateQueue"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="direction">
  <xs:restriction base="xs:string">
    <xs:enumeration value="sendrecv"/>
    <xs:enumeration value="sendonly"/>
    <xs:enumeration value="recvonly"/>
    <xs:enumeration value="inactive"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="encryptionKey">
  <xs:restriction base="xs:base64Binary"/>
</xs:simpleType>
<xs:complexType name="encryptionKeyData">
  <xs:sequence>
    <xs:element name="encryptedCall" type="xs:boolean"/>
    <xs:element name="callID" type="xs:string"/>
    <xs:element name="localKey" type="mx1:encryptionKey" minOccurs="0"/>
    <xs:element name="remoteKey" type="mx1:encryptionKey" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="endPoint">
  <xs:sequence>
    <xs:element name="mediaAddress" type="mx1:ipAddress" minOccurs="0"/>
    <xs:element name="port" type="mx1:port" minOccurs="0"/>
    <xs:element name="crypto" type="mx1:crypto" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="freeOnSecondStatus">
  <xs:sequence>
    <xs:element name="freeOnSecondActive" type="xs:boolean"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="getCallingLineIdentityDataRequest">
  <xs:sequence>
    <xs:element name="deviceID" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="getFreeOnSecond">
  <xs:sequence>
    <xs:element name="deviceID" type="xs:string"/>
  </xs:sequence>
</xs:complexType>

```

```

<xs:complexType name="getTerminalSelectionServiceData">
  <xs:sequence>
    <xs:element name="deviceID" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="getPersonalNumber">
  <xs:sequence>
    <xs:element name="deviceID" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:simpleType name="notificationDelayTime">
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="60"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="notificationVisibleTime">
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="180"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="showCallingNumber">
  <xs:restriction base="xs:string">
    <xs:enumeration value="showNone"/>
    <xs:enumeration value="showCallingNumber"/>
    <xs:enumeration value="showCalledNumber"/>
    <xs:enumeration value="showAll"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="groupCallPickupClearData">
  <xs:annotation>
    <xs:documentation/>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="droppedConnection" type="csta:ConnectionID"/>
    <xs:element name="releasingDevice" type="csta:DeviceID"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="groupCallPickupDeliveredData">
  <xs:annotation>
    <xs:documentation/>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="connection" type="csta:ConnectionID"/>
    <xs:element name="alertingDevice" type="csta:DeviceID"/>
    <xs:element name="callingDevice" type="csta:DeviceID"/>
    <xs:element name="calledDevice" type="csta:DeviceID"/>
    <xs:element name="notificationDelayTime"
type="mx1:notificationDelayTime"/>
    <xs:element name="notificationVisibleTime"
type="mx1:notificationVisibleTime"/>
    <xs:element name="showCallingNumber" type="mx1:showCallingNumber"/>
  </xs:sequence>
</xs:complexType>
<xs:simpleType name="icsDiversionReasonCode">
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="9"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="diversionTypes">
  <xs:restriction base="xs:string">
    <xs:enumeration value="allImmediate"/>

```

```

        <xs:enumeration value="icsImmediateDiversion"/>
        <xs:enumeration value="diversionImmediate"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="serviceRestriction">
    <xs:restriction base="xs:string">
        <xs:enumeration value="noNotification"/>
        <xs:enumeration value="notificationWithServiceOnly"/>
        <xs:enumeration value="notificationWithServiceAndToUserNumber"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="ipV4Address">
    <xs:restriction base="xs:string">
        <xs:pattern value="((25[0-5]|2[0-4][0-9]|1[0-9][0-9]|1[0-9][0-9]|
[0-9])\.){3}(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|1[0-9][0-9]|[0-9])"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="ipV6Address">
    <xs:restriction base="xs:string">
        <xs:pattern value="([A-Fa-f0-9]{0,4}:){2,7}[A-Fa-f0-9]{1,4}"/>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="ipAddress">
    <xs:choice>
        <xs:element name="ipV4Address" type="mxl:ipV4Address"/>
        <xs:element name="ipV6Address" type="mxl:ipV6Address"/>
    </xs:choice>
</xs:complexType>
<xs:complexType name="mediaStreamInformationData">
    <xs:sequence>
        <xs:element name="callID" type="xs:string"/>
        <xs:element name="cryptoSuite" type="mxl:cryptoSuite"/>
        <xs:element name="direction" type="mxl:direction"/>
        <xs:element name="mediaType" type="mxl:mediaType" minOccurs="0"/>
        <xs:element name="thisEndPoint" type="mxl:endPoint" minOccurs="0"/>
        <xs:element name="otherEndPoint" type="mxl:endPoint" minOccurs="0"/>
        <xs:element name="sipCallId" type="xs:string" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
<xs:simpleType name="mediaType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="audio"/>
        <xs:enumeration value="video"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="personalNumberList">
    <xs:restriction base="xs:integer">
        <xs:minInclusive value="1"/>
        <xs:maxInclusive value="10"/>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="personalNumberStatus">
    <xs:sequence>
        <xs:element name="personalNumberActive" type="xs:boolean"/>
        <xs:element name="personalNumberList" type="mxl:personalNumberList"
minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
<xs:simpleType name="port">
    <xs:restriction base="xs:unsignedShort"/>
</xs:simpleType>
<xs:complexType name="returnDateOrTime">
    <xs:choice>
        <xs:element name="returnDate" type="mxl:returnDate"/>

```

```

    <xs:element name="returnTime" type="mx1:returnTime"/>
  </xs:choice>
</xs:complexType>
<xs:simpleType name="returnDate">
  <xs:annotation>
    <xs:documentation>MMDD</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value="\d{4}"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="returnTime">
  <xs:annotation>
    <xs:documentation>HHMM</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value="\d{4}"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="setFreeOnSecond">
  <xs:sequence>
    <xs:element name="deviceId" type="xs:string"/>
    <xs:element name="freeOnSecondStatus" type="mx1:freeOnSecondStatus"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="groupHuntNumberList">
  <xs:annotation>
    <xs:documentation/>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="deviceIdentifier" type="csta:DeviceID" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="extensionOptionNumberList">
  <xs:annotation>
    <xs:documentation/>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="deviceIdentifier" type="csta:DeviceID" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="overridingNumberAndName">
  <xs:sequence>
    <xs:element name="active" type="xs:boolean"/>
    <xs:element name="deviceId" type="xs:string" minOccurs="1"
maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="publicCallingLineIdentityList">
  <xs:sequence>
    <xs:element name="commonPublicNumber" type="csta:DeviceID"
minOccurs="0" maxOccurs="1"/>
    <xs:element name="extensionOptionNumberList"
type="mx1:extensionOptionNumberList" minOccurs="0" maxOccurs="1"/>
    <xs:element name="groupHuntNumberList" type="mx1:groupHuntNumberList"
minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="selectedPublicCallingLineIdentity">
  <xs:sequence>
    <xs:element name="active" type="xs:boolean"/>

```

```

        <xs:element name="selectedPublicCallingLineIdentity"
type="csta:DeviceID" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="callingLineIdentityData">
    <xs:annotation>
        <xs:documentation/>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="overridingNumberAndName"
type="mx1:overridingNumberAndName" minOccurs="1" maxOccurs="1"/>
        <xs:element name="selectedPublicCallingLineIdentity"
type="mx1:selectedPublicCallingLineIdentity" minOccurs="1" maxOccurs="1"/>
        <xs:element name="publicCallingLineIdentityList"
type="mx1:publicCallingLineIdentityList" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
</xs:complexType>
<xs:simpleType name="selectCallingLineIdentityOrder">
    <xs:restriction base="xs:string">
        <xs:enumeration value="commonPublic"/>
        <xs:enumeration value="extensionOption"/>
        <xs:enumeration value="group"/>
        <xs:enumeration value="restricted"/>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="setCallingLineIdentityData">
    <xs:annotation>
        <xs:documentation/>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="deviceID" type="xs:string" minOccurs="1"
maxOccurs="1"/>
        <xs:element name="selectCallingLineIdentityActive" type="xs:boolean"
minOccurs="1" maxOccurs="1"/>
        <xs:element name="selectCallingLineIdentityOrder"
type="mx1:selectCallingLineIdentityOrder" minOccurs="1" maxOccurs="1"/>
        <xs:element name="groupNumber" type="csta:DeviceID" minOccurs="0"
maxOccurs="1"/>
    </xs:sequence>
</xs:complexType>
<xs:simpleType name="selectRemoteExtensionAction">
    <xs:restriction base="xs:string">
        <xs:enumeration value="noChangeToSelectedRemoteExtension"/>
        <xs:enumeration value="changeToPredefinedNumber"/>
        <xs:enumeration value="changeToUserDefinedNumber"/>
        <xs:enumeration value="removeUserDefinedNumber"/>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="deviceAlertList">
    <xs:sequence>
        <xs:element name="deviceID" type="csta:DeviceID" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="remoteExtensionsList">
    <xs:annotation>
        <xs:documentation/>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="deviceIdentifier" type="csta:DeviceID" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="setTerminalSelectionServiceData">

```

```

    <xs:annotation>
      <xs:documentation/>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="deviceID" type="xs:string" minOccurs="1"
maxOccurs="1"/>
      <xs:element name="selectRemoteExtensionAction"
type="mx1:selectRemoteExtensionAction" minOccurs="1" maxOccurs="1"/>
      <xs:element name="selectedRemoteExtension" type="csta:DeviceID"
minOccurs="1" maxOccurs="1"/>
      <xs:element name="deviceAlertList" type="mx1:deviceAlertList"
minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="terminalSelectionServiceData">
    <xs:annotation>
      <xs:documentation/>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="remoteExtensionsList"
type="mx1:remoteExtensionsList" minOccurs="1" maxOccurs="1"/>
      <xs:element name="selectedRemoteExtension" type="csta:DeviceID"
minOccurs="1" maxOccurs="1"/>
      <xs:element name="AlertingDevice" type="mx1:deviceAlertList"
minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="tssSetUserDefinedNumberResponseData">
    <xs:annotation>
      <xs:documentation/>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="definedUserNumber" type="csta:DeviceID"
minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="setPersonalNumber">
    <xs:sequence>
      <xs:element name="deviceID" type="xs:string"/>
      <xs:element name="personalNumberStatus"
type="mx1:personalNumberStatus"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="sessionCategoryEventData">
    <xs:sequence>
      <xs:element name="allowMultiplicityInTSS" type="xs:boolean"/>
      <xs:element name="numberConversionInTSS" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="typeOfDevice">
    <xs:restriction base="xs:string">
      <xs:enumeration value="acdGroup"/>
      <xs:enumeration value="analogExtension"/>
      <xs:enumeration value="callOriginGroup"/>
      <xs:enumeration value="ctiGroup"/>
      <xs:enumeration value="dectExtension"/>
      <xs:enumeration value="digitalExtensionAdditionalDirectoryNumber"/>
      <xs:enumeration value="digitalExtension"/>
      <xs:enumeration value="genericExtension"/>
      <xs:enumeration value="ipExtension"/>
      <xs:enumeration value="multiTerminalExtension"/>
      <xs:enumeration value="operator"/>
      <xs:enumeration value="pbxGroup"/>
      <xs:enumeration value="remoteExtension"/>
    </xs:restriction>
  </xs:simpleType>

```

```

    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="cstaSubTypeOfDevice">
    <xs:restriction base="xs:string">
      <xs:enumeration value="noSupportForCstaCallControl"/>
      <xs:enumeration value="multipleLineAccess"/>
      <xs:enumeration value="singleLineAccess"/>
      <xs:enumeration value="virtualSipExtension"/>
    </xs:restriction>
  </xs:simpleType>
  <!-- Service requests and responses -->
  <xs:element name="consultationCallPrivateData">
    <xs:complexType>
      <xs:annotation>
        <xs:documentation/>
      </xs:annotation>
      <xs:sequence>
        <xs:element name="bypassDiversion" type="xs:boolean"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="deflectCallPrivateData">
    <xs:complexType>
      <xs:annotation>
        <xs:documentation/>
      </xs:annotation>
      <xs:sequence>
        <xs:element name="deflectSubService" type="mx1:deflectSubService"
minOccurs="0"/>
        <xs:element name="allowDiversionDuringDeflect" type="xs:boolean"
minOccurs="0"/>
        <xs:element name="replaceDialledNumber" type="xs:boolean"
minOccurs="0"/>
        <xs:element name="bypassPersonalNumber" type="xs:boolean"
minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="escapePrivateData">
    <xs:complexType>
      <xs:annotation>
        <xs:documentation/>
      </xs:annotation>
      <xs:choice>
        <xs:element name="getCallingLineIdentityData"
type="mx1:getCallingLineIdentityDataRequest"/>
        <xs:element name="getFreeOnSecond" type="mx1:getFreeOnSecond"/>
        <xs:element name="getPersonalNumber" type="mx1:getPersonalNumber"/>
      </xs:choice>
    </xs:complexType>
  </xs:element>
  <xs:element name="getTerminalSelectionServiceData"
type="mx1:getTerminalSelectionServiceData"/>
  <xs:element name="setCallingLineIdentityData"
type="mx1:setCallingLineIdentityData"/>
  <xs:element name="setFreeOnSecond" type="mx1:setFreeOnSecond"/>
  <xs:element name="setPersonalNumber" type="mx1:setPersonalNumber"/>
  <xs:element name="setTerminalSelectionServiceData"
type="mx1:setTerminalSelectionServiceData"/>
  </xs:choice>
</xs:complexType>
</xs:element>
<xs:element name="escapeResponsePrivateData">
  <xs:complexType>
    <xs:annotation>

```



```

    <xs:documentation/>
  </xs:annotation>
  <xs:choice>
    <xs:element name="getCallingLineIdentityDataResponse"
type="mx1:callingLineIdentityData"/>
    <xs:element name="getFreeOnSecondResponse"
type="mx1:freeOnSecondStatus"/>
    <xs:element name="getTerminalSelectionServiceDataResponse"
type="mx1:terminalSelectionServiceData"/>
    <xs:element name="getPersonalNumberResponse"
type="mx1:personalNumberStatus"/>
    <xs:element name="setTerminalSelectionServiceDataResponse"
type="mx1:tssSetUserDefinedNumberResponseData"/>
  </xs:choice>
</xs:complexType>
</xs:element>
<xs:element name="getForwardingResponsePrivateData">
  <xs:complexType>
    <xs:annotation>
      <xs:documentation/>
    </xs:annotation>
    <xs:choice>
      <xs:sequence>
        <xs:element name="icsDiversionReasonCode"
type="mx1:icsDiversionReasonCode"/>
        <xs:element name="icsDiversionReasonString" type="xs:string"
minOccurs="0"/>
        <xs:element name="returnDateOrTime" type="mx1:returnDateOrTime"
minOccurs="0"/>
        <xs:element name="serviceRestrictionNoAnswer"
type="mx1:serviceRestriction"/>
        <xs:element name="serviceRestrictionOnBusy"
type="mx1:serviceRestriction"/>
      </xs:sequence>
      <xs:sequence>
        <xs:element name="serviceRestrictionImmediate"
type="mx1:serviceRestriction"/>
        <xs:element name="serviceRestrictionNoAnswer"
type="mx1:serviceRestriction"/>
        <xs:element name="serviceRestrictionOnBusy"
type="mx1:serviceRestriction"/>
      </xs:sequence>
    </xs:choice>
  </xs:complexType>
</xs:element>
<xs:element name="getLogicalDeviceInformationResponsePrivateData">
  <xs:complexType>
    <xs:annotation>
      <xs:documentation/>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="typeOfDevice" type="mx1:typeOfDevice"/>
      <xs:element name="cstaSubTypeOfDevice"
type="mx1:cstaSubTypeOfDevice" minOccurs="0"/>
      <xs:element name="deviceModelName" type="xs:string" minOccurs="0"/>
      <xs:element name="ipAddress" type="mx1:ipAddress" minOccurs="0"/>
      <xs:element name="logOnNumber" type="csta:SubjectDeviceID"
minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="makeCallPrivateData">
  <xs:complexType>
    <xs:annotation>

```

```

        <xs:documentation/>
      </xs:annotation>
    <xs:sequence>
      <xs:element name="bypassDiversion" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="monitorStartPrivateData">
  <xs:complexType>
    <xs:annotation>
      <xs:documentation/>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="multiTerminalUseSFReprDeviceID" type="xs:boolean"
minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="monitorStartResponsePrivateData">
  <xs:complexType>
    <xs:annotation>
      <xs:documentation/>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="typeOfDevice" type="mx1:typeOfDevice"/>
      <xs:element name="cstaSubTypeOfDevice"
type="mx1:cstaSubTypeOfDevice" minOccurs="0"/>
      <xs:element name="deviceModelName" type="xs:string" minOccurs="0"/>
      <xs:element name="ipAddress" type="mx1:ipAddress" minOccurs="0"/>
      <xs:element name="logOnNumber" type="csta:SubjectDeviceID"
minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="setForwardingPrivateData">
  <xs:complexType>
    <xs:annotation>
      <xs:documentation/>
    </xs:annotation>
    <xs:choice>
      <xs:sequence>
        <xs:element name="icsDiversionReasonCode"
type="mx1:icsDiversionReasonCode"/>
        <xs:element name="returnDateOrTime" type="mx1:returnDateOrTime"
minOccurs="0"/>
      </xs:sequence>
      <xs:element name="diversionType" type="mx1:diversionTypes"/>
      <xs:element name="serviceRestriction" type="mx1:serviceRestriction"/
>
    </xs:choice>
  </xs:complexType>
</xs:element>
<xs:element name="snapshotDevicePrivateData">
  <xs:complexType>
    <xs:annotation>
      <xs:documentation/>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="multiTerminalUseSFReprDeviceID" type="xs:boolean"
minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!-- Events -->

```

```

<xs:element name="backInServiceEventPrivateData">
  <xs:complexType>
    <xs:annotation>
      <xs:documentation/>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="typeOfDevice" type="mx1:typeOfDevice"/>
      <xs:element name="cstaSubTypeOfDevice"
type="mx1:cstaSubTypeOfDevice" minOccurs="0"/>
      <xs:element name="deviceModelName" type="xs:string" minOccurs="0"/>
      <xs:element name="ipAddress" type="mx1:ipAddress" minOccurs="0"/>
      <xs:element name="logOnNumber" type="csta:SubjectDeviceID"
minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="deviceCapsChangedEventPrivateData">
  <xs:complexType>
    <xs:annotation>
      <xs:documentation/>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="deviceList" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="deviceID" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="forwardingEventPrivateData">
  <xs:complexType>
    <xs:annotation>
      <xs:documentation/>
    </xs:annotation>
    <xs:choice>
      <xs:sequence>
        <xs:element name="icsDiversionReasonCode"
type="mx1:icsDiversionReasonCode"/>
        <xs:element name="icsDiversionReasonString" type="xs:string"
minOccurs="0"/>
        <xs:element name="returnDateOrTime" type="mx1:returnDateOrTime"
minOccurs="0"/>
      </xs:sequence>
      <xs:sequence>
        <xs:element name="serviceRestriction"
type="mx1:serviceRestriction"/>
      </xs:sequence>
    </xs:choice>
  </xs:complexType>
</xs:element>
<xs:element name="privateEventPrivateData">
  <xs:complexType>
    <xs:annotation>
      <xs:documentation/>
    </xs:annotation>
    <xs:sequence>
      <xs:choice>
        <xs:element name="callingLineIdentityEventData"
type="mx1:callingLineIdentityData"/>

```

```

        <xs:element name="encryptionKeyData" type="mx1:encryptionKeyData"/
>
        <xs:element name="freeOnSecondData" type="mx1:freeOnSecondStatus"/
>
        <xs:element name="groupCallPickupClearData"
type="mx1:groupCallPickupClearData"/>
        <xs:element name="groupCallPickupDeliveredData"
type="mx1:groupCallPickupDeliveredData"/>
        <xs:element name="mediaStreamInformationData"
type="mx1:mediaStreamInformationData"/>
        <xs:element name="personalNumberData"
type="mx1:personalNumberStatus"/>
        <xs:element name="sessionCategoryEventData"
type="mx1:sessionCategoryEventData"/>
        <xs:element name="terminalSelectionServiceEventData"
type="mx1:terminalSelectionServiceData"/>
        </xs:choice>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="SessionLoginInfo">
    <xs:complexType>
        <xs:annotation>
            <xs:documentation/>
        </xs:annotation>
        <xs:sequence>
            <xs:element name="password" type="xs:string" minOccurs="1"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:schema>

```

