

Chapter 1

Look Inside

Introduction

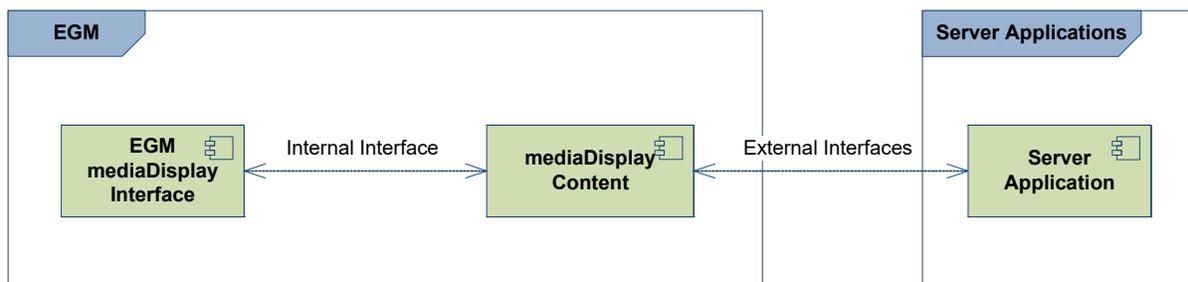
1.1 Introduction to the EGM mediaDisplay Interface

The EGM mediaDisplay Interface serves as a standard interface for content running in the media display device to the EGM to:

- Show the media display device.
- Hide the media display device.
- Toggle call attendant functionality.
- Receive events from the EGM.

The first two points serve as a mechanism for local access. They allow media display content to request that the EGM make local changes in state without being required to perform a round trip to server applications that would use G2S commands. The ability to toggle the call attendant functionality is used so that the mechanical button can be moved on screen to the media display content. The final benefit is that it helps to facilitate quick media display content reaction to events emitted by the EGM, thereby preventing any network latency issues from causing delays in any initial responses to key events that need to be displayed in the media display content. The following diagram provides an illustration of this interface:

Figure 1.1 Overall Diagram



The EGM mediaDisplay Interface is not intended to replace the external interface that media display content can have with server applications. It merely offers a local method of limited scope, which provides low-latency delivery of important data, and an efficient local mechanism to open and close the media display device.

Extensions to the EGM mediaDisplay Interface should keep these principles in mind. Extensions should not supersede the device ownership model that is inherent within G2S. Content may act as a G2S guest and receive information generated by the EGM and reported through the G2S protocol. However, content should not act as a G2S owner and exercise command and control authority over the EGM through the EGM mediaDisplay Interface except (1) to take actions related to the management of the mediaDisplay window in which the content is executing and (2) to take actions related to other devices that support the player user interface - for example, hiding a window, showing a window, toggling the call attendant button, clearing an ID from an ID reader, requesting that the EGM initiate a WAT transaction, etc. Furthermore, except when host-to-content messaging is being used, the EGM mediaDisplay Interface should not be used as a method for conveying host-originated information to the content. Host-originated information should be delivered through an external interface between the content and the server application.

1.1.1 Raising Window Priority

Extension in v3.0: RMD

The EGM Media Display Interface can also be used by content to request that an overlay window be raised to the topmost position. See [Section 6.22, raiseMediaDisplay Command](#) for more details.

Chapter 6

Look Inside

mdCabinet Functional

Group

6.1 Introduction

The mdCabinet functional group of commands offers media display content the ability to:

- Show the media display device.
- Hide the media display device.
- Toggle call attendant functionality.
- Get card status.

The sections that follow provide further details on the command set provided.

6.2 Request-Response Pairs

The following tables organize the commands contained within the mdCabinet functional group into request-response pairs:

Table 6.1 Commands Originated By mediaDisplay

| Request | Response |
|-------------------------------|-------------------------|
| getCallAttendantState | callAttendantStatus |
| setCallAttendantState | callAttendantStatus |
| getDeviceVisibleState | deviceVisibleStatus |
| setDeviceVisibleState | deviceVisibleStatus |
| getCardState | cardStatus |
| <i>Extension in 1b1: HCI</i> | |
| contentToHostMessage | contentToHostMessageAck |
| <i>Extension in 1b1: PLC</i> | |
| getCabinetStatus | cabinetStatus |
| <i>Extension in 1b1: CPC</i> | |
| setCardRemoved | cardStatus |
| <i>Extension in v3.0: PSD</i> | |
| getPlayerSessionData | playerSessionDataAck |
| <i>Extension in v3.0: MCS</i> | |
| getCardStateList | cardStatusList |
| <i>Extension in v3.0: RMD</i> | |
| raiseMediaDisplay | deviceVisibleStatus |

6.2.1 Commands Originated by EGM

Extension in 1b1: HCI

Table 6.2 Commands Originated By EGM

| Request | Response |
|----------------------|-------------------------|
| hostToContentMessage | hostToContentMessageAck |

6.3 getCallAttendantState Command

6.3.1 Command Description

The `getCallAttendantState` command can be sent by the media display content to the EGM in order to determine the status of the call attendant button. The `callAttendantStatus` command is sent in response.

6.3.2 Attribute and Element Details

| | |
|------------------------------|--------------------------------------|
| getCallAttendantState | |
| type | <code>c_getCallAttendantState</code> |

The `getCallAttendantState` command has no attributes or sub-elements.

6.4 setCallAttendantState Command

6.4.1 Command Description

The `setCallAttendantState` command can be sent by the media display content to the EGM in order to set the state of the EGM’s call attendant button. The `callAttendantStatus` command is sent in response.

If the EGM cannot change the state to the requested value, the current state is returned in the `callAttendantStatus`.

6.4.2 Attribute and Element Details

| | |
|------------------------------|-------------------------|
| setCallAttendantState | |
| type | c_setCallAttendantState |

Table 6.3 setCallAttendantState Attributes

| Attribute | Restrictions | Description |
|-----------|---|--|
| enable | type: <code>xs:boolean</code> use: <code>required</code> | Sets the state of the EGM’s call attendant button. A value of <code>true</code> means the button has been pressed. |

6.5 callAttendantStatus Command

6.5.1 Command Description

The `callAttendantStatus` command is sent by the EGM to the media display content in response to the `setCallAttendantState` command or the `getCallAttendantState` command. The `callAttendantStatus` command lets the media display content know if the call attendant button is currently set to active.

6.5.2 Attribute and Element Details

| | |
|----------------------------|------------------------------------|
| callAttendantStatus | |
| type | <code>c_callAttendantStatus</code> |

Table 6.4 callAttendantStatus Attributes

| Attribute | Restrictions | Description |
|----------------------------------|--|--|
| <code>callAttendantActive</code> | type: <code>xs:boolean</code> use: required | Represents the state of the EGM's call attendant button. A value of <code>true</code> means the button has been pressed. |

6.6 getDeviceVisibleState Command

6.6.1 Command Description

The `getDeviceVisibleState` command can be sent by the media display content to the EGM in order to determine if the media display device is being shown or hidden. The `deviceVisibleStatus` command is sent by the EGM in response to the `getDeviceVisibleState` command.

6.6.2 Attribute and Element Details

| | |
|------------------------------|--------------------------------------|
| getDeviceVisibleState | |
| type | <code>c_getDeviceVisibleState</code> |

The `getDeviceVisibleState` command has no attributes or sub-elements.

6.7 deviceVisibleStatus Command

6.7.1 Command Description

The `deviceVisibleStatus` command is sent by the EGM to the media display content in response to a `getDeviceVisibleState` or `setDeviceVisibleState` command. The `deviceVisibleStatus` command indicates the state of the media display device (shown or hidden).

6.7.1.1 Raising Overlay Window Priority

Extension in v3.0: RMD

The `deviceVisibleStatus` command also indicates whether the media display device is currently the topmost window and whether the overlay window is currently being shown as a modal window. See the G2S Message Protocol for more details about topmost and modal windows as well as the `modalDisplay`, `topMostWindow`, and `gamePlaySuspended` attributes.

6.7.2 Attribute and Element Details

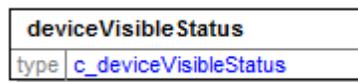


Table 6.5 deviceVisibleStatus Attributes

| Attribute | Restrictions | Description |
|---------------------------------|--|--|
| <code>deviceVisibleState</code> | type: <code>xs:boolean</code> use: optional default: true | Represents the state of the media display content's media display window. A value of <code>true</code> indicates the media display window is visible, while a value of <code>false</code> indicates the media display window is not visible. |
| <i>Extension in v3.0: RMD</i> | | |
| <code>modalDisplay</code> | type: <code>xs:boolean</code> use: optional default: false | Indicates whether the overlay window is a modal window; MUST be set to false for scaled windows. |
| <code>topMostWindow</code> | type: <code>xs:boolean</code> use: optional default: false | Indicates whether the overlay window is currently being displayed as the topmost window; MUST be set to false for scaled windows. |
| <code>gamePlaySuspended</code> | type: <code>xs:boolean</code> use: optional default: false | Indicates whether the overlay window currently has game play suspended; MUST be set to false for scaled windows. |

6.8 setDeviceVisibleState Command

6.8.1 Command Description

The `setDeviceVisibleState` command is sent by media display content to the EGM to show or hide the media display window. The `deviceVisibleStatus` is sent by the EGM in response. If the EGM cannot set the visible state to the requested value, the current state is returned immediately in the `deviceVisibleStatus` command.

NOTE:

When the `deviceVisibleState` changes, the `contentStatus.deviceVisibleState` attribute of the corresponding `mediaDisplay` device **MUST** be updated and the appropriate G2S event **MUST** be generated. `IGT_MDE106 Media Display Shown` is generated when the device is shown and `IGT_MDE107 Media Display Hidden` is generated when the device is hidden.

6.8.1.1 Raising Window Priority

Extension in v3.0: RMD

When the `setDeviceVisibleState` command is sent by media display content to the EGM to request that an overlay window be shown, the content can also request that the media display window be shown as modal and control other characteristics of the window. See the G2S Message Protocol for more details about `topmost` and `modal` windows as well as the `modalWindow`, `modalTimeout`, `suspendGamePlay`, `suspendTimeout`, `suspendCondition`, and `autoHideOnPlay` attributes.

6.8.2 Attribute and Element Details

| | |
|------------------------------|--------------------------------------|
| setDeviceVisibleState | |
| type | <code>c_setDeviceVisibleState</code> |

Table 6.6 setDeviceVisibleState Attributes

| Attribute | Restrictions | Description |
|---------------------------------|---|---|
| <code>deviceVisibleState</code> | type: <code>xs:boolean</code> use: optional default: <code>true</code> | Indicates the state of the media display content's media display window. A value of <code>true</code> indicates the EGM needs to show the media display window, while a value of <code>false</code> indicates the EGM needs to hide the media display window. |
| <i>Extension in v3.0: RMD</i> | | |
| <code>displayCondition</code> | type: <code>t_displayConditions</code> use: optional default: <code>G2S_gameIdle</code> | Indicates when the window should be displayed; MUST be ignored for scaled windows. |
| <code>modalWindow</code> | type: <code>xs:boolean</code> use: optional default: <code>false</code> | Indicates whether the EGM must display the overlay window as modal; MUST be ignored for scaled windows. |

Table 6.6 setDeviceVisibleState Attributes

| Attribute | Restrictions | Description |
|-----------------|---|---|
| modalTimeout | type: <code>t_milliseconds</code> use: optional default: 10000 minIncl: 1000 | When the modalWindow attribute is set to true, indicates the minimum amount of time the overlay window must be displayed before it can be overlaid by another window; MUST be ignored for scaled windows. |
| suspendGamePlay | type: <code>xs:boolean</code> use: optional default: false | Indicates whether the EGM must suspend game play while the window is being displayed; MUST be ignored for scaled windows. |
| suspendTimeout | type: <code>t_milliseconds</code> use: optional default: 10000 minIncl: 1000 | When the suspendGamePlay attribute is set to true, indicates the maximum amount of time that game play must be suspended; MUST be ignored for scaled windows. |
| autoHideOnPlay | type: <code>xs:boolean</code> use: optional default: true | When modalWindow is set to false, indicates whether the window should be displayed or hidden during game play; MUST be ignored for scaled windows. |

6.9 getCardState Command

6.9.1 Command Description

The `getCardState` command is sent by the media display content to the EGM to determine whether or not a card is in the ID reader which is associated with the currently active player session. If there is no active player session, the status of the ID reader associated with the last active player session is reported. If there have been no player sessions, the status of any active ID reader may be reported.

6.9.2 Attribute and Element Details

| | |
|---------------------------|-----------------------------|
| <code>getCardState</code> | |
| type | <code>c_getCardState</code> |

The `getCardState` command has no attributes or sub-elements.

6.10 cardStatus Command

6.10.1 Command Description

The `cardStatus` command is sent by the EGM to the media display content in response to a `getCardState` command. This informs the media display content if a card is currently in the card reader.

6.10.1.1 setCardRemoved

Extension in 1b1: CPC

The `cardStatus` command is also sent by the EGM to the media display content in response to a `setCardRemoved` command.

6.10.2 Attribute and Element Details

| |
|---------------------------------|
| cardStatus |
| type: <code>c_cardStatus</code> |

Table 6.7 cardStatus Attributes

| Attribute | Restrictions | Description |
|------------------------------|---|---|
| <code>cardIn</code> | type: <code>xs:boolean</code> use: optional default: <code>false</code> | Indicates whether an ID is present. A value of <code>true</code> indicates that an ID is present. |
| <code>idReaderType</code> | type: <code>t_idReaderTypes</code> use: required | Type of ID reader. |
| <code>idNumber</code> | type: <code>t_idNumber</code> use: optional default: <code><empty></code> | ID number. If no ID present then set to <code><empty></code> . |
| <code>idValidExpired</code> | type: <code>xs:boolean</code> use: optional default: <code>true</code> | Indicates whether the validation has expired due to inactivity; if no ID is present then set to <code>false</code> . |
| <i>Extension in 1b1: CPC</i> | | |
| <code>idReaderId</code> | type: <code>t_deviceId</code> use: optional default: <code>0</code> | The <code>idReader</code> device identifier; a value of <code>0</code> (zero) indicates that the <code>idReader</code> device is unknown; a value of <code>0</code> (zero) MUST NOT be used if this extension is supported. |

6.11 setCardRemoved Command

Extension in 1b1: CPC

6.11.1 Command Description

The `setCardRemoved` command is sent by the media display content to the EGM to clear the current ID information from the specified `idReader` device. Upon processing the command, the EGM MUST behave as if the ID had been removed from the specified `idReader` device, generating G2S event `G2S_IDE103 ID Cleared From Reader`. If no ID is present at the specified `idReader` device, the EGM MUST simply generate the `cardStatus` response reporting the current status of the specified `idReader` device to the content.

The `cardStatus` command is sent in response to the `setCardRemoved` command.

This command is intended to be used when the content includes an option for the player to end the current player session. Typically, this feature would be used when the player's ID is not present for the entire player session and, thus, the player needs a way to indicate that the player session should be terminated. For example, when an EGM-controlled swipe-style ID reader is used, the player cannot indicate that the player session should be terminated by removing the ID and, thus, the content may include an option that allows the player to terminate the player session. However, the use of this feature is not limited to EGM-controlled swipe-style ID readers. It can be used with any style of ID reader.

6.11.2 Attribute and Element Detail

| |
|----------------------------|
| setCardRemoved |
| type: CPC:c_setCardRemoved |

Table 6.8 setCardRemoved Attributes

| Attribute | Restrictions | Description |
|-------------------------|--|--|
| <code>idReaderId</code> | type: <code>t_deviceId</code> use: required | The <code>idReader</code> device identifier. |

6.12 hostToContentMessage Command

Extension in 1b1: HCI

6.12.1 Command Description

The `hostToContentMessage` command is used by the EGM to deliver instructions or other data received from the host to the active content. The EGM MUST, if possible, send this command to the content after receiving a G2S `hostToContentMessage` command from the G2S `mediaDisplay` host. The command MUST only be generated after receiving a G2S `hostToContentMessage` command from the G2S host and MUST only be sent to the active content associated with the G2S `mediaDisplay` device that received the G2S `hostToContentMessage` command. The command contains, as XML binary data, the instructions or other data to be delivered to the active content. It is up to the content provider to define the proper formatting so that the active content can parse the instructions or data. The EGM is not expected to retry this command; it is up to the content provider to define any application-level retry requirements for the host. The media display content generates a `hostToContentMessageAck` command in response to this command.

6.12.2 Attribute and Element Detail

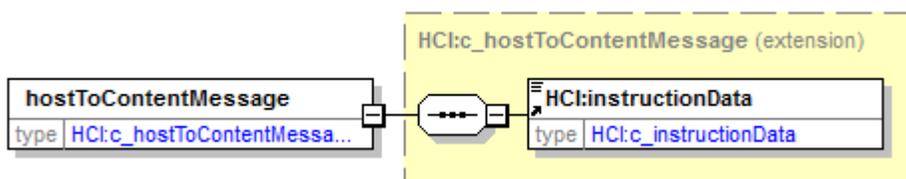


Table 6.9 hostToContentMessage Sub-Elements

| Element | Restrictions | Description |
|-----------------|--|--|
| instructionData | type: <code>xs:base64Binary</code> minOcc: 1 maxOcc: 1 | Contains instructions or other data for the content. |

6.13 hostToContentMessageAck Command

Extension in 1b1: HCI

6.13.1 Command Description

The `hostToContentMessageAck` is sent by the media display content to the EGM to acknowledge the receipt of instructions or other data from the host. The `hostToContentMessageAck` command is generated in response to a `hostToContentMessage` command.

6.13.2 Attribute and Element Detail

| | |
|--------------------------------|------------------------------|
| hostToContentMessageAck | |
| type | HCI:c_hostToContentMessag... |

The `hostToContentMessageAck` command has no attributes or sub-elements.

6.14 contentToHostMessage Command

Extension in 1b1: HCI

6.14.1 Command Description

The `contentToHostMessage` command is used by the media display content to send instructions or other data to the host. The EGM MUST, if possible, generate a G2S `contentToHostMessage` command for the G2S `mediaDisplay` host from the G2S `mediaDisplay` device in which the content is running. The command contains, as XML binary data, the instructions or other data to be delivered to the host. The EGM is not expected to retry the G2S `contentToHostMessage` command; it is up to the content provider to define any application-level retry requirements for the content. A `contentToHostMessageAck` command is generated in response to the `contentToHostMessage` command.

6.14.2 Attribute and Element Detail

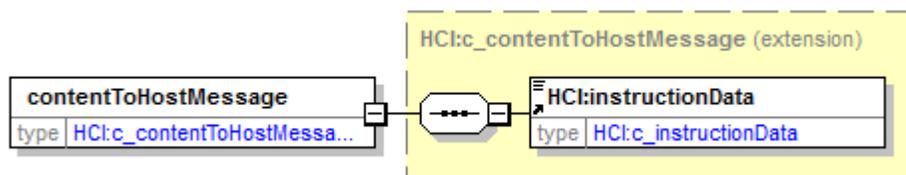


Table 6.10 contentToHostMessage Sub-Elements

| Element | Restrictions | Description |
|-----------------|--|---|
| instructionData | type: <code>xs:base64Binary</code> minOcc: 1 maxOcc: 1 | Contains instructions or other data for the host. |

6.15 contentToHostMessageAck Command

Extension in 1b1: HCI

6.15.1 Command Description

This command is used by the EGM to acknowledge the receipt of instructions or other data from the content. The EGM may send the acknowledgement before generating the G2S `contentToHostMessage` command. The `contentToHostMessageAck` command is generated in response to a `contentToHostMessage` command.

6.15.2 Attribute and Element Detail

| | |
|--------------------------------|------------------------------|
| contentToHostMessageAck | |
| type | HCI:c_contentToHostMessag... |

The `contentToHostMessageAck` command has no attributes or sub-elements.

6.16 getCabinetStatus Command

Extension in 1b1: PLC

6.16.1 Command Description

The `getCabinetStatus` command is sent by the media display content to the EGM to determine the current state of the EGM including the current locale identifier in use at the EGM for displaying information to the player. A `cabinetStatus` command is generated in response to the `getCabinetStatus` command.

6.16.2 Attribute and Element Detail

| |
|---|
| getCabinetStatus |
| type PLC:c_getCabinetStatus |

The `getCabinetStatus` command has no attributes or sub-elements.

6.17 cabinetStatus Command

Extension in 1b1: PLC

6.17.1 Command Description

The `cabinetStatus` command is sent by the EGM to the media display content in response to a `getCabinetStatus` command. The command informs the media display content about the current state of the EGM including the current locale identifier in use.

6.17.2 Attribute and Element Detail

| |
|--|
| cabinetStatus |
| type c_cabinetStatus |

See [Section 5.14.4](#), [Table 5.22](#), for information sent in this command.

6.18 getPlayerSessionData Command

Extension in v3.0: PSD

6.18.1 Command Description

The `getPlayerSessionData` command is used to retrieve player session information from the EGM. The EGM MUST only send back player session information from the G2S player class that has the `playerLog.sessionState = G2S_sessionOpen`. A `playerSessionDataAck` command is generated in response to the `getPlayerSessionData` command.

If there are no open player sessions, the EGM MUST return error code 501 `No Open Player Sessions`.

6.18.2 Attribute and Element Detail

| |
|---|
| getPlayerSessionData |
| type PSD:c_getPlayerSessionData |

The `getPlayerSessionData` command has no attributes or sub-elements.

6.19 playerSessionDataAck Command

Extension in v3.0: PSD

6.19.1 Command Description

The `playerSessionDataAck` command is used by the EGM to send the open player session information to the content.

6.19.2 Attribute and Element Detail

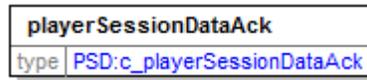


Table 6.11 playerSessionAck Attributes

| Attributes | Restrictions | Description |
|----------------------------|---|--|
| <code>idReaderId</code> | type: <code>t_deviceId</code> use: required | ID Reader device identifier used to initiate the player session. |
| <code>idNumber</code> | type: <code>t_idNumber</code> use: required | ID Number. |
| <code>startDateTime</code> | type: <code>xs:dateTime</code> use: required | Date and time that the session was started. |

6.20 getCardStateList Command

Extension in v3.0: MCS

6.20.1 Command Description

The `getCardStateList` command is sent by the `mediaDisplay` content to the EGM to request the list of all the active ID reader devices on the EGM and the current status of those devices. A `cardStatusList` command is generated in response to the `getCardStateList` command.

6.20.2 Attributes and Element Details

| |
|---|
| getCardStateList |
| type MCS:c_getCardStateList |

The `getCardStateList` command has no attributes or sub-elements.

6.21 cardStatusList Command

Extension in v3.0: MCS

6.21.1 Command Description

The `cardStatusList` command is sent by the EGM to the `mediaDisplay` content in response to a `getCardStateList` command. The `cardStatusList` command reports the list of all active ID Reader devices on the EGM, and the current status of those ID Reader devices.

6.21.2 Attributes and Element Details

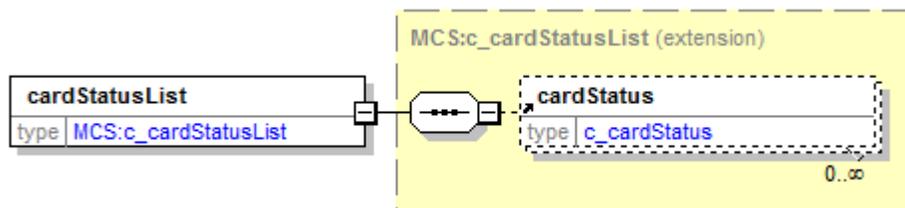


Table 6.12 cardStatusList Sub-Elements

| Element | Restrictions | Description |
|------------|------------------------|--|
| cardStatus | minOcc: 0 maxOcc: ∞ | Identifies the active <code>idReader</code> devices on the EGM and the current status of those devices. See Element details in Table 6.7 . |

6.22 raiseMediaDisplay Command

Extension in v3.0: RMD

6.22.1 Command Description

The `raiseMediaDisplay` command can be sent by media display content to the EGM to raise the overlay window to be the topmost position. The `deviceVisibleStatus` command is sent by the EGM in response.

If content attempts to raise an overlay window and the `deviceVisibleState` attribute is not set to `IGT_shown`, the EGM MUST respond with error code 511 `Content Must Be Shown Before Raised`. The window MUST remain hidden with the `deviceVisibleState` set to `IGT_hidden`.

The command can only be used with overlay windows. If the content sends the `raiseMediaDisplay` command to a window that is not an overlay window, the EGM MUST respond with error code 512 `Window Is Not An Overlay Window`.

If a host attempts to raise an overlay window, but there is currently an overlay window being shown as modal, the EGM MUST respond with error code 513 `Modal Overlay Window Active`.

See the G2S Message Protocol for more details about how the `topMostWindow` and `modalDisplay` attributes of the `deviceVisibleStatus` response should be managed.

6.22.2 Attribute and Element Detail

| |
|--------------------------------|
| raiseMediaDisplay |
| type RMD:c_raiseMediaDisplay |

The `raiseMediaDisplay` command has no attributes or sub-elements.

6.23 mdCabinet Error Codes

The mdCabinet functional group contains the following error codes.

Table 6.13 mdCabinet Group-Level Error Codes

| Error Code | Description |
|-------------------------------|--------------------------------------|
| <i>Extension in v3.0: PSD</i> | |
| 501 | No Open Player Sessions |
| <i>Extension in v3.0: RMD</i> | |
| 511 | Content Must Be Shown Before Raised. |
| 512 | Window Is Not An Overlay Window |
| 513 | Modal Overlay Window Active |

6.24 Data Types

This section identifies the data types specific to the mdCabinet functional group.

Table 6.14 mdEventHandler Data Types

| Data Type | Restrictions | Description |
|-------------------------------|---|--|
| t_idNumber | type: <code>xs:string</code> minLen: 0 maxLen: 32 | Identification number; player or employee. |
| t_idReaderTypes | type: <code>t_uniqueIdentifier64</code> | Identification reader types. See G2S protocol for reserved enumerations. |
| t_egmStates | type: <code>xs:string</code> enumerations: See Section 6.24.1 | Current state of the EGM. |
| <i>Extension in v3.0: RMD</i> | | |
| t_displayConditions | type: <code>t_uniqueIdentifier64</code> | Display conditions for an overlay window; see the G2S protocol for enumeration values. |
| t_milliseconds | type: <code>xs:int</code> minIncl: 0 | Milliseconds |

6.24.1 egmStates Data Type Enumerations

Table 6.15 egmStates Data Type Enumeration

| Enumeration | Description |
|-----------------------|--|
| G2S_transportDisabled | EGM disabled by the transport layer. |
| G2S_operatorDisabled | EGM disabled via operator menu. |
| G2S_hostDisabled | EGM is disabled due to host. |
| G2S_egmDisabled | EGM disabled due to device failure or door open. |
| G2S_enabled | EGM enabled and available for play. |
| G2S_operatorMode | Operator menu active. |
| G2S_demoMode | Demo mode activated. |
| G2S_auditMode | Meters/audit mode active. |
| G2S_operatorLocked | EGM locked via operator menu. |
| G2S_egmLocked | EGM locked due to device action. |
| G2S_hostLocked | EGM locked due to host action. |