|  |
| --- |
| Application Programming Interface Document |
| Media Sever API Specification |
| Version 1.0 |

|  |
| --- |
| **Document Information** |

|  |
| --- |
| **Document Sign Off** |

|  |  |
| --- | --- |
| **Project Manager**  (Solution Architecture & QA) | Mr. INAMULLAH |
| **Development Lead**  (Media Server Project) | Mr. Waqqas Jabbar |
| **Documentation Team** | Technical Writing Department |

|  |  |
| --- | --- |
| **Document Information** | |
|  | |
| **Version #** | 1.0 |
| **Revision Date** | April 22, 2008. |
| **Prepared By** | Shafaq Irshad. |

|  |
| --- |
| **History** |

**Document Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Revision** | **Author** | **Description** |
|  | | | |
| April 22, 2008 |  | Shafaq Irshad | The document contains information of various Media Server APIs |
|  |  |  |  |

**Document Purpose:**

The document is designed to give comprehensive and quick information about Media Server APIs. It further contains detailed structure explanation of Media Server APIs.

**Contents**

[2. Initialization & Configuration APIs 7](#_Toc196520068)

[2.1.1. InitMediaServer 7](#_Toc196520069)

[3. SIP Call Back APIs 8](#_Toc196520070)

[3.1.1. HandleSIPRequest 8](#_Toc196520071)

[3.1.2. HandleSIPResponse 8](#_Toc196520072)

[4. Conference Processing APIs 9](#_Toc196520073)

[4.1.1. HandleConferenceRequest 9](#_Toc196520074)

[4.1.2. HandleConferenceResponse 9](#_Toc196520075)

[4.1.3. CreateConference 10](#_Toc196520077)

[4.1.4. RemoveConference 10](#_Toc196520078)

[4.1.5. AddParticipant 11](#_Toc196520079)

[4.1.6. RemoveParticipant 11](#_Toc196520080)

[4. Non Conference Processing API 12](#_Toc196520081)

[4.2.1. HandleNonConferenceRequest 12](#_Toc196520082)

[4.2.2. HandleNonConferenceResponse 12](#_Toc196520083)

[6. User Call Leg API 15](#_Toc196520084)

[6.1.1. EventProcessor 15](#_Toc196520085)

[6.1.2. SendInfo 15](#_Toc196520086)

[9. MSCML Processing API 18](#_Toc196520087)

[9.1.1. EventProcessor 18](#_Toc196520088)

[9.1.2. StartTimer 18](#_Toc196520090)

[9.1.3. StopTimer 19](#_Toc196520091)

[9.1.4. SendMSCMLResponse 19](#_Toc196520092)

[9.1.5. Play 19](#_Toc196520093)

[9.1.6. Collect 20](#_Toc196520094)

[9.1.7. Record 20](#_Toc196520095)

[9.1.8. Stop 21](#_Toc196520096)

[10. Media Processing API 23](#_Toc196520097)

[10.1.1. AllocateMedia 23](#_Toc196520098)

[10.1.2. ConfigureMedia 23](#_Toc196520100)

[10.1.3.UnConfigureMedia 24](#_Toc196520102)

[10.1.4. UnallocateMedia 24](#_Toc196520104)

1. MsRetCode

*Typedef enum MsRetCode*

*{*

*MSE\_Success,*

*MSE\_InvalidValue*

*MSE\_InvalidID*

*MSE\_NotFound*

*MSE\_AlreadyExsists*

*MSE\_InsufficientResources,*

*} MsRetCode\_t*

1. MSInitData

*MsInitData*

*{*

String strNonConfUri

SDP\* pLocalCap

SIP\_Config\* pSipConfig

Framework\_Config\* pFrameworkConfig

*} MsInitData\_t;*

**Structure Explanation**

**Fields:**

StrNonConfigUri

URI for non – conference processing.

pLocalCap

Handle of Local Media Capabilities.

pSipConfig

Pointer to SIP Stack Configuration.

pFrameworkCongi

Pointer to Framwork Configurations.

1. Initialization & Configuration APIs

2.1.1. InitMediaServer

*MsRetCode InitMediaServer(MsInitData\* pInitData );*

**Purpose**

This function initializes the Media Server.

**Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | In/out | Description |
| pInitData | MsInitData\* | In | Handle to initialization data of Media Server. |

## Return value

This function will return MsRetCode, which will contain one of the possible values as defined in MsRetCode Enumeration.

1. Conference Structure

*MsConference*

*{*

*String StrConfId;*

*MsCallLeg\* pCtrlLeg;*

*MsCallLegList\* pParticipantList;*

*unsigned int uiCurrNumPatipant;*

*unsigned int uiMaxAllowedPartcipant;*

*} MsConference\_t;*

**Structure Explanation**

**Fields:**

StrConfId

This string is a conference identifier of this conference.

pCtrlLeg

Control Leg associated with this Conference.

pParticipantList

List of participants in the conference.

uiCurrNumPartipant

This field identifies current number of participants in the conference.

uiMaxAllowedParticipant

This field identifies maximum allowed participants in the conference.

1. SIP Call Back APIs

3.1.1. HandleSIPRequest

*MsRetCode HandlesSIPRequest ( SipCallLeg\* pCallLeg ,SipMsg\* pSipMsg );*

**Purpose**

The purpose of this API is to handle SIP requests received by Media Server.

**Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | In/out | Description |
| pCallLeg | SipCallLeg\* | In | Handle of Call Leg on which request is received. |
| pSipMsg | SipMsg\* | In | Handle of SIP Message that is received. |

## Return value

This function will return MsRetCode, which will contain one of the possible values as defined in MsRetCode Enumeration.

3.1.2. HandleSIPResponse

*MsRetcode HandleSIPResponse(SipCallLeg\* pCallLeg, SipMsg\* pSipMsg );*

**Purpose**

This function is used handle SIP responses received by Media Server.

**Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | In/out | Description |
| pCallLeg | SipCallLeg\* | In | Handle of Call Leg on which request is received. |
| pSipMsg | SipMsg\* | In | Handle of SIP Message that is received. |

## Return value

This function will return MsRetCode, which will contain one of the possible values as defined in MsRetCode Enumeration.

1. Conference Processing APIs

4.1.1. HandleConferenceRequest

*MsRetCode HandleConferenceRequest(SipCallLeg\* pCallLeg , SipMsg\* pSipMsg );*

**Purpose**

This function handles SIP Request received on a conference URI.

**Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | In/out | Description |
| pCallLeg | SipCallLeg\* | In | Handle of Call Leg on which request is received. |
| pSipMsg | SipMsg\* | In | Handle of SIP Message that is received. |

## Return value

This function will return MsRetCode, which will contain one of the possible values as defined in MsRetCode Structure.

4.1.2. HandleConferenceResponse

*MsRetCode HandleConferenceReponse ( SipCallLeg\* pCallLeg , SipMsg\* pSipMsg);*

**Purpose**

This function handles the conference response.

**Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | In/out | Description |
| pCallLeg | SipCallLeg\* | In | Handle of Call Leg on which request is received. |
| pSipMsg | SipMsg\* | In | Handle of SIP Message that is received. |

## Return value

This function will return MsRetCode, which will contain one of the possible values as defined in MsRetCode Structure.

4.1.3. CreateConference

*MsRetCode CreateConference (string StrConfId, stConfigureConf\* pConfigConf);*

**Purpose**

This function is used to create a new conference.

**Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | In/out | Description |
| StrConfId | string | In | Unique ID from the conference from Request URI. |
| pConfigConf | stConfigureConf \* | In | Parameter for configuring conference. |

## Return value

This function will return MsRetCode, which will contain one of the possible values as defined in MsRetCode Enumeration.

4.1.4. DestroyConference

*MsRetCode DestroyConference (string StrConfId );*

**Purpose**

It removes the conference.

**Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | In/out | Description |
| StrConfId | string | In | Unique ID from the conference from Request URI. |

## Return value

This function will return MsRetCode, which will contain one of the possible values as defined in MsRetCode Enumeration.

.

4.1.5. AddParticipant

*MsRetCode AddPartcipant (string StrConfId, SipCallLeg\* pCallLeg);*

**Purpose**

This function adds the new participant in the conference.

**Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | In/out | Description |
| StrConfId | string | In | Unique Id for the Conference from Request URI. |
| pCallLeg | SipCallLeg\* | In | Call Leg handle to add in a conference. |

## Return value

This function will return MsRetCode, which will contain one of the possible values as defined in MsRetCode Structure.

4.1.6. RemoveParticipant

*MsRetCode RemovePartcipant (string StrConfId, SipCallLeg\* pCallLeg);*

**Purpose**

This function removes participant from the conference.

**Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | In/out | Description |
| StrConfId | string | In | Unique Id for the Conference from Request URI. |
| pCallLeg | SipCallLeg\* | In | Call Leg handle to add in a conference. |

## Return value

This function will return MsRetCode, which will contain one of the possible values as defined in MsRetCode Structure.

**4.2. Non Conference Processing API**

4.2.1. HandleNonConferenceRequest

*MsRetCode HandleNonConferenceRequest (SipCallLeg\* pCallLeg, SipMsg\* pSipMsg);*

**Purpose**

This function handles non- conference based requests.

**Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | In/out | Description |
| pCallLeg | SipCallLeg\* | In | Handle of Call Leg on which request is received. |
| pSipMsg | SipMsg\* | In | Handle of SIP Message that is received. |

## Return value

This function will return MsRetCode, which will contain one of the possible values as defined in MsRetCode Structure.

4.2.2. HandleNonConferenceResponse

*MsRetCode HandleNonConferenceResponse (SipCallLeg\* pCallLeg, SipMsg\* pSipMsg);*

**Purpose**

This function handles response of non – conference based requests.

**Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | In/out | Description |
| pSipCallLeg | SipCallLeg\* | In | Handle of Call Leg on which call is received. |
| pSipMsg | SipMsg\* | In | Handle of SIP Message that is received. |

## Return value

This function will return MsRetCode, which will contain one of the possible values as defined in MsRetCode Structure.

1. MS Call Leg

*MsCallLeg*

*{*

*CallLegState eState;*

*MsMcml\* pMscml*

*SipCallLeg\* pSipCallLeg*

*CallLegType\* etype*

*MsCallLegMedia\* pMedia*

*} MsCallleg\_t;*

**Structure Explanation**

**Fields:**

eState

This represents the current state of a call.

pMscml

Handle of MSCML object associated with this leg.

pSipCallLeg

Handle of SIP Call Leg.

eType

Type of Call Leg.

PMedia

Handle of Media Objects associated with this Leg.

1. **MsCallLegMode**

*Typedef enum MsCallLegMode*

*{*

*MsAPI-SendOnly,*

*MsAPI-ReceiveOnly,*

*MsAPI-SendRecvOnly,*

*MsAPI-Inactive,*

*} MsCallLegMode\_t*

1. **CallLegType**

*Typedef enum CallLegType*

*{*

*MsAPI-ConferenceControlLeg,*

*MsAPI-UserLeg,*

*} ConferenceControlLeg\_t*

1. MS Call Leg Event

*MsCallLegEvent*

*{*

*MsCallLegEventType eType;*

*MsCallLegEventData\* pdata*

*} MsCallLegEvent\_t;*

**Structure Explanation**

**Fields:**

eType

This represents the type of event

pdata

This represents the data corresponding to type of events

1. **User Call Leg API**

6.1.1. EventProcessor

*MsRetCode EventProcessor(MsCallLegEvent\* pMsClEvent);*

**Purpose**

Processes incoming events for Call Leg according to its state machine

**Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | In/out | Description |
| pMsClEvent | MsCallLegEvent\* | In | Pointer to event structure |

## Return value

This function will return MsRetCode, which will contain one of the possible values as defined in MsRetCode Enumeration.

6.1.2. SendInfo

*MsRetCode SendInfo (MsMscml\* pMscmlMessage);*

**Purpose**

Send an INFO message on this Call Leg

**Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | In/out | Description |
| pMscmlMessage | MsMscml\* | In | MSCML message to send with the message |

## Return value

This function will return MsRetCode, which will contain one of the possible values as defined in MsRetCode Structure.

1. MS MSCML Event

*MsMscmlEvent*

*{*

*MsMscmlEventType eType;*

*MsMsmlEventData\* pdata*

*} MsMscmlEventType\_t;*

**Structure Explanation**

**Fields:**

eType

This represents the type of event

pdata

This represents the data corresponding to type of events.

1. MS Call Leg Event

*MsCallLegEvent*

*{*

*MsCallLegEventType eType;*

*MsCallLegEventData\* pdata*

*} MsCallLegEvent\_t;*

**Structure Explanation**

**Fields:**

eType

This represents the type of event

pdata

This represents the data corresponding to type of events

1. MS MSCML

*MsMscml*

*{*

*MsMscmlStMachine pStatMachine;*

*MsMscmlMessage\* pMscmlReq,*

*} MsCallLegEvent\_t;*

**Structure Explanation**

**Fields:**

eStateMachine

This is handle of state machine.

pMscmlReq

Handle of MSCML Request.

1. **MsMscmlEventType**

*Typedef enum MsMscmlEventType*

*{*

*MS\_MSCML\_DTMF\_KEY,*

*MS\_MSCML\_RECORDING\_STARTED,*

*MS\_MSCML\_RECORDING\_STOPPED,*

*MS\_MSCML\_RECORDING\_ERROR,*

*MS\_MSCML\_PLAYING\_STARTED,*

*MS\_MSCML\_PLAYING\_STOPPED*

*MS\_MSCML\_PLAYING\_ERROR,*

*MS\_MSCML\_TIMER\_EXPIRED,*

*MS\_MSCML\_REQUEST,*

*}MsMscmlEventType\_t*

1. **MsCallLegEventType**

*typedef enum MsCallLegEventType*

*{*

*MS\_CL\_SIP\_REQUEST*

*MS\_CL\_SIP\_RESPONSE*

*MS\_CL\_DTMF\_KEY*

*MS\_CL\_MSCML\_RESPONSE*

*} MsCallLegEventType\_t*

**9.1. MSCML Processing API**

9.1.1. EventProcessor

*MsRetCode EventProcessor(MsMscmlEvent\* pMscmlEvent);*

**Purpose**

Processes incoming events according to its state machine.

**Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | In/out | Description |
| pMscmlEvent | MsMscmlEvent\* | In | Handle to MSMCL event. |

## Return value

This function will return MsRetCode, which will contain one of the possible values as defined in MsRetCode Enumeration

9.1.2. StartTimer

*MsRetCode StartTimer(unsigned int iIntervalMillisec, void\* pTimerData, void\* pTimerId);*

**Purpose**

This function starts a timer.

**Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | In/out | Description |
| iIntervalMillisec | unsigned int | In | Interval of timer in milliseconds |
| pTimerData | void\* | In | Handle to data to return after timer is expired |
| pTimerId | void\* | Out | Handle to timer identifier |

## Return value

This function will return MsRetCode, which will contain one of the possible values as defined in MsRetCode Enumeration

9.1.3. StopTimer

*MsRetCode StopTimer( void\* pTimerId);*

**Purpose**

This function stops a timer.

**Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | In/out | Description |
| pTimerId | void\* | In | Handle to timer identifier |

## Return value

This function will return MsRetCode, which will contain one of the possible values as defined in MsRetCode Enumeration.

9.1.4. SendMSCMLResponse

*MsRetCode SendMSCMLResponse (MsMscmlMessage\* pMscmlResponse);*

**Purpose**

This function sends MSCML response.

**Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | In/out | Description |
| pMscmlResponse | MsMscmlMessage\* | Out | Handle of MSMCL response. |

## Return value

This function will return MsRetCode, which will contain one of the possible values as defined in MsRetCode Enumeration.

9.1.5. Play

*MsRetCode Play(pPlayNode\* pPlayNode);*

**Purpose**

This function plays a Call Leg as specified in <play>, <playrecord> and <playcollect> tag.

**Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | In/out | Description |
| pPlayNode | pPlayNode\* | In | Pointer to play node in MSCML message. |

## Return value

This function will return MsRetCode, which will contain one of the possible values as defined in MsRetCode Enumeration.

.

9.1.6. Collect

*MsRetCode Collect(pCollectNode\* pCollectNode);*

**Purpose**

This function collects digits on a call leg as specified in <playcollect> tag.

**Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | In/out | Description |
| pCollectNode | CollectNode\* | In | Pointer to Collect Node in MSCML message. |

## Return value

This function will return MsRetCode, which will contain one of the possible values as defined in MsRetCode Structure.

9.1.7. Record

*MsRetCode Record(pRecordNode\* pRecordNode);*

**Purpose**

This function records media on a Call Leg as specified in <playrecord> tag.

**Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | In/out | Description |
| pRecordNode | RecordNode\* | In | Pointer to Record Node in MSCML message. |

## Return value

This function will return MsRetCode, which will contain one of the possible values as defined in MsRetCode Enumeration.

9.1.8. Stop

*MsRetCode Stop(pStopNode\* pStopNode);*

**Purpose**

This function stops an operation on a call leg.

**Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | In/out | Description |
| pStopNode | StopNode\* | In | Pointer to StopNode in MSCML message. |

## Return value

This function will return MsRetCode, which will contain one of the possible values as defined in MsRetCode Enumeration.

1. MS Call Leg Media

*MsCallLegMedia*

*{*

*CallLegMode eMode;*

*MsSdp\* pSdp*

*} MsCallLegMedia\_t;*

**Structure Explanation**

**Fields:**

eMode

This represents the current mode of the call leg.

pSdp

Handle of SDP objects associated with this leg.

1. **CallLegType**

*Typedef enum CallLegType*

*{*

*CLT\_User,*

*CLT\_Type,*

*} CallLegType\_t*

1. **CallLegMode**

*Typedef enum CallLegMode*

*{*

*CLM\_Sendonly*

*CLM\_Recvonly*

*CLM\_Sendrecv*

*CLM\_Inactive*

*} CallLegMode\_t*

**10.1 Media Processing API**

10.1.1. AllocateMedia

*MsRetCode AllocateMedia(MsCallLegMedia\* pMedia);*

**Purpose**

Allocate resources for a media channel.

**Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | In/out | Description |
| pMedia | MsCallLegMedia\* | Out | Handle of Media object |

## Return value

This function will return MsRetCode, which will contain one of the possible values as defined in MsRetCode Enumeration.

10.1.2. ConfigureMedia

*MsRetCode ConfigureMedia(SDP\* pSdp);*

**Purpose**

Configure a media channel with negotiated media parameters

**Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | In/out | Description |
| pSdp | SDP\* | In | Negotiated SDP on the media channel |

## Return value

This function will return MsRetCode, which will contain one of the possible values as defined in MsRetCode Enumeration.

10.1.3.UnConfigureMedia

*MsRetCode UnConfigureMedia(MsCallLegMedia\* pMedia);*

**Purpose**

Remove the configuration

**Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | In/out | Description |
| pMedia | MsCallLegMedia\* | In | Handle of Media object to operate on |

## Return value

This function will return MsRetCode, which will contain one of the possible values as defined in MsRetCode Enumeration.

10.1.4. UnallocateMedia

*MsRetCode UnallocateMedia(MsCallLegMedia\* pMedia);*

**Purpose**

Unallocate resources for a media channel

**Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | In/out | Description |
| pMedia | MsCallLegMedia\* | In | Handle of Media object to operate on |

## Return value

This function will return MsRetCode, which will contain one of the possible values as defined in MsRetCode Enumeration.