Mitel MiContact Center Enterprise

AGENT APPLICATION INTEGRATION DESCRIPTION

Release 9.2

NOTICE

TRADEMARKS

MiContact Center Agent Application Integration Description
Release 9.2 – November 2016

INTRODUCTION

This document contains information for integrating into the MiContact Center Agent application. MiContact Center Agent supports a Component Object Model (COM) interface and a .NET interface.

Note that the integration interface is only supported for agents assigned with the Use Integration Interface user privilege.

INTEGRATION OBJECT

The AgentClient object is implemented in a DLL named AgentIntegration.dll. The object provides an interface for client applications to control and automate MiContact Center Agent, and it also has events to inform all connected clients about changed data in MiContact Center Agent or the current status of the object.

The object can be added to a project for connection via COM or it can be added to a Microsoft .NET project as a reference for integration via .NET. For integration using .NET, the interfaces below should be accessed using the .NET counterpart without the I prefix, that is, AgentClient, UserInformation, Call, CallInformation, Email, and SMS).

Test client applications are available on the installation media, in the Tools\AgentIntegration folder.

SCRIPTING FROM WEB TAB

A subset of the integration interface is also available to JavaScript code running in a web page displayed within a tab of the Agent application.

The corresponding methods are identified in the **JS** column of the following interface tables.

When invoked from JavaScript, the method names are changed from PascalCase to camelCase, so for instance GetUserInformation() becomes getUserInformation().

A test page is available on the installation media, in the Tools\AgentIntegration\JavaScriptTest folder.

LOGGING REQUESTS AND EVENTS

It is possible to log all requests received and events sent through the integration interface. Set the Diagnostic Log Level on Agent to be 5 or higher, and all requests and events will be logged to the local Agent.log file on the client machine.

INTERFACES

This chapter describes the settings for different interfaces in MiContact Center Agent.

IAGENTCLIENT

The IAgentClient interface supports the following requests:

METHOD	INPUT PARAMETERS	JS	DESCRIPTION
Initialization Commands			
bool Initialize()	None	No	Establishes a connection to a running Agent application. If the connection is established, true is returned.
bool Uninitialize()	None	No	Removes the connection to the Agent application.
bool CanInitialize()	None	No	Returns true if MiContact Center Agent is currently running and available to be connected to by the interface.
Startup/Shutdown Commands			
bool IsBSARunning()	None	No	Returns true if MiContact Center Agent is currently running.
bool LaunchBSA()	None	No	Starts the MiContact Center Agent application.
bool LaunchBSAWithCredentials()	Logon ID – The logon ID of the user. Password – The password of the user. Extension – The extension number to be used (optional). Extension Password – The password associated with the extension, if required (optional). Site Name – The name of the OAS site to be connected to (optional). Web Server – The machine where the MiContact Center Logon web service is installed (optional). Broker Server – The machine where the MiContact Center Broker service is running (optional). Broker Port – The port number	No	Starts the MiContact Center Agent application with the parameters provided. Returns true if the application is able to be launched.

METHOD	INPUT PARAMETERS	JS	DESCRIPTION
	to be used to connect to the Broker service (optional).		
IUserInformation GetUserInformation()	None	Yes	Returns an IUserInformation interface with information about the current logged on user.
bool IsRegistered()	None	No	Returns true if the application is registered with the MiContact Center Agent request interface and able to make requests and receive events.
void CloseApplication()	None	No	Shuts down the MiContact Center Agent application.
Call Control Commands			
Note: Call Control commands will return True if the command is successfully sent to the Agent application. The call events will indicate if the operation was successfully executed.			
bool MakeCall()	Number – Represents the number to be dialed. Name – Represents the name of the party to be dialed (optional). Use Number Translation – Boolean value indicating whether the configured number translation rules should be applied before making the call.	Yes	Initiates a call to the number indicated.
bool AnswerActiveCall()	None	Yes	Answers a currently ringing call.
bool AnswerCall()	Call ID – Call ID of the call to be answered.	Yes	Answers the call indicated by the Call ID parameter, if it is currently ringing.
bool HangupActiveCall()	None	Yes	Disconnects an active call.
bool HangupCall()	Call ID – Call ID of the call to be disconnected.	Yes	Disconnects the call indicated by the Call ID parameter.
bool HoldActiveCall()	None	Yes	Holds an active call.
bool RetrieveActiveCall()	None	Yes	Retrieves the currently held call.
bool RetrieveCall()	Call ID – Call ID of the call to be retrieved.	Yes	Retrieves the call indicated by the Call ID parameter.

METHOD	INPUT PARAMETERS	JS	DESCRIPTION
bool TransferCall()	Call ID – Call ID of the held call.	Yes	Transfers the call indicated by the Call ID to the currently active call.
bool ConferenceActiveCall()	None	Yes	Creates a conference with the currently held call and the currently active call.
bool EnterDtmfActiveCall()	Digits – The digits to be entered.	Yes	Sends the digit string provided as DTMF signals for the currently active call.
bool DeflectActiveCall()	Destination – Number to which the call will be deflected	Yes	Deflects the currently active call to the destination provided.
bool DeflectCall()	Call ID – Call ID of the call to be deflected. Destination – Number to which the call will be deflected	Yes	Deflects the call indicated by the Call ID to the destination provided.
bool CanMakeCall()	None	Yes	Returns true if MiContact Center Agent is in the correct state to initiate an outgoing call.
bool CanConference()	None	Yes	Returns true if MiContact Center Agent is in the correct state to create a conference.
bool CanTransfer()	None	Yes	Returns true if MiContact Center Agent is in the correct state to transfer a held call to the currently active call.
bool SetPrivateData()	Call ID – Call ID of the call for which private data will be set. Data - Data to be set for the call.	Yes	Returns true if the request to set the data is able to be successfully sent to the Call Control Service. Subsequent call events will contain the private data, if successful.
bool StartRecording()	Agent ID – Currently not used. All Calls – Boolean value indicating whether all calls should be recorded, or just the currently active/next active call.	Yes	Returns true if the request to start recording is successfully processed by MiContact Center Agent.
bool StopRecording()	Agent ID – Currently not used.	Yes	Returns true if the request to stop recording is successfully processed by MiContact Center Agent.
bool DeflectCallToServiceGroup()	Call ID – Call ID of the call to be deflected. Service Group ID – ID of the	Yes	Deflects the call indicated by the Call ID to the group provided. The group must

METHOD	INPUT PARAMETERS	JS	DESCRIPTION
	service group to which the call will be deflected		handle the same media type.
bool SetIVRData()	Call ID – Call ID of the call for which IVR data will be set. Label – Indicates the IVR label to be set. This can indicate IVR1, IVR2 or IVR3 or an existing IVR label to update an existing field, or a new label to add a new IVR data field. For campaign calls, the Label can indicate the name of an existing campaign customer field allowed to be updated. Data – Indicates the IVR data to be set.	Yes	Sets IVR or campaign data for the voice call or media session.
Agent Commands			
bool CancelClerical()	Call ID – Call ID of the call in clerical state	Yes	Cancels clerical state for the call indicated by the Call ID provided.
bool ExtendClerical()	Call ID – Call ID of the call in clerical state	Yes	Extends clerical state for the call indicated by the Call ID provided.
bool RejectServiceCall()	Call ID – Call ID of the service group call currently ringing	Yes	Rejects the incoming service group call indicated by the Call ID provided. The call will be returned to the service group queue.
bool SetReadyStatus()	Status – Boolean value indicating that the agent is Ready (true) or Not Ready (false) Reason ID – Record ID of the Not Ready Reason, or 0 if not used or not applicable. Reason Name – Name of the Not Ready Reason; null or empty string if not used or not applicable.	Yes	Changes the Ready/Not Ready status of MiContact Center Agent for voice calls. To set the Not Ready Reason, provide either the Reason ID or the Reason Name field. If both are provided, the Reason ID will take precedence. This command can also be used to change the Not Ready Reason for an agent already in Not Ready state by calling SetReadyStatus(false) with the new Not Ready Reason.
bool SetEmailReadyStatus()	Status – Boolean value indicating that the agent is Ready (true) or Not Ready (false)	Yes	Changes the Ready/Not Ready status of MiContact Center Agent for E-mail and SMS messages.
bool SetOpenMediaReadyStatus()	Status – Boolean value indicating that the agent is	Yes	Changes the Ready/Not Ready status of MiContact

METHOD	INPUT PARAMETERS	JS	DESCRIPTION
	Ready (true) or Not Ready (false)		Center Agent for open media sessions.
bool SetChatReadyStatus()	Status – Boolean value indicating that the agent is Ready (true) or Not Ready (false)	Yes	Changes the Ready/Not Ready status of MiContact Center Agent for chat sessions.
bool SetCampaignCallStatus()	Call ID – Call ID of the campaign call. Status – Status of the campaign call. May be standard status value of: 1 = Not Yet Called 2 = Busy 4 = No Answer 8 = Callback Later 16 = Completed 32 = Wrong Number 64 = Do Not Call Again Or may be customer value greater than 64. ScheduleCallback – Boolean value indicating if the call should be scheduled for a later time. Status must be Callback Later if this is supplied. CallbackTime – Date and time to schedule the callback.	No	Set the call status for a campaign call.
INotReadyReasons[] GetNotReadyReasons()	None	Yes	Returns an array of INotReadyReason objects with one object for each defined Not Ready Reason in the MiCC Enterprise system.
Session Retrieval Commands			
ICall[] GetCalls()	None	Yes	Returns an array of ICall objects, with one object for each active voice call on the Agent application.
IEmail[] GetEmails()	None	Yes	Returns an array of IEmail objects, with one object for each active e-mail message on the Agent application.
ISMS[] GetSMS()	None	Yes	Returns an array of ISMS objects, with one object for each active SMS message on the Agent application.
IOpenMedia[] GetOpenMedia()	None	Yes	Returns an array of IOpenMedia objects, with one object for each active open media session on the Agent application.

METHOD	INPUT PARAMETERS	JS	DESCRIPTION
IChat[] GetChats()	None	Yes	Returns an array of IChat objects, with one object for each active chat session on the Agent application.
Dispatch			
bool SubscribeDispatchPersonalCalls()	ServiceGroupID – ID of the service group or 0 for all service groups. Subscribe – Boolean value indicating whether to subscribe or unsubscribe.	No	Subscribes or unsubscribes for dispatch personal call events. Returns true if successful.
bool SubscribeDispatchCommonHoldCalls()	Subscribe – Boolean value indicating whether to subscribe or unsubscribe.	No	Subscribes or unsubscribes for dispatch common hold call events. Returns true if successful.
bool SubscribeDispatchVoiceCalls()	ServiceGroupID – ID of the service group or 0 for all service groups. Subscribe – Boolean value indicating whether to subscribe or unsubscribe.	No	Subscribes or unsubscribes for dispatch voice call events. Returns true if successful.
bool SubscribeDispatchEmailsCalls()	ServiceGroupID – ID of the service group or 0 for all service groups. Subscribe – Boolean value indicating whether to subscribe or unsubscribe.	No	Subscribes or unsubscribes for dispatch e-mail events. Returns true if successful.
bool SubscribeDispatchSMSsCalls()	ServiceGroupID – ID of the service group or 0 for all service groups. Subscribe – Boolean value indicating whether to subscribe or unsubscribe.	No	Subscribes or unsubscribes for dispatch SMS events. Returns true if successful.
bool SubscribeDispatchCampaignCalls()	CampaignID – ID of the campaign or 0 for all campaigns. Subscribe – Boolean value indicating whether to subscribe or unsubscribe.	No	Subscribes or unsubscribes for dispatch campaign call events. Returns true if successful.
bool RetrieveDispatchPersonalCall()	CallID – Call ID of the call to retrieve.	No	Retrieves the dispatch call. Returns true if successful.
bool RetrieveDispatchCommonHoldCall()	CallID – Call ID of the call to retrieve.	No	Retrieves the dispatch call. Returns true if successful.
bool RetrieveDispatchVoiceCall()	CallID – Call ID of the call to retrieve.	No	Retrieves the dispatch call. Returns true if successful.
bool RetrieveDispatchEmail()	CallID – Call ID of the call to	No	Retrieves the dispatch call.

METHOD	INPUT PARAMETERS	JS	DESCRIPTION
	retrieve.		Returns true if successful.
bool RetrieveDispatchSMS()	CallID – Call ID of the call to retrieve.	No	Retrieves the dispatch call. Returns true if successful.
bool RetrieveDispatchCampaignCall()	CustomerID – ID of the customer to retrieve.	No	Retrieves the dispatch call. Returns true if successful.
ReserveDispatchPersonalCall()	CallID – Call ID of the call to reserve or unreserved Reserve – Boolean value indicating whether to reserve or unreserve.	No	Reserves or unreserves the dispatch call. Returns true if successful.
ReserveDispatchCommonHoldCall()	CallID – Call ID of the call to reserve or unreserved Reserve – Boolean value indicating whether to reserve or unreserve.	No	Reserves or unreserves the dispatch call. Returns true if successful.
ReserveDispatchVoiceCall()	CallID – Call ID of the call to reserve or unreserved Reserve – Boolean value indicating whether to reserve or unreserve.	No	Reserves or unreserves the dispatch call. Returns true if successful.
ReserveDispatchEmail()	CallID – Call ID of the call to reserve or unreserved Reserve – Boolean value indicating whether to reserve or unreserve.	No	Reserves or unreserves the dispatch call. Returns true if successful. This call is reserved for future use. Reserving dispatch e-mails is not currently supported.
ReserveDispatchSMS()	CallID – Call ID of the call to reserve or unreserved Reserve – Boolean value indicating whether to reserve or unreserve.	No	Reserves or unreserves the dispatch call. Returns true if successful. This call is reserved for future use. Reserving dispatch SMSs is not currently supported.
ReserveDispatchCampaignCall()	CustomerID –ID of the customer to reserve or unreserved Reserve – Boolean value indicating whether to reserve or unreserve.	No	Reserves or unreserves the dispatch call. Returns true if successful.
bool EnableDispatchEvents()	Enable – Boolean value indicating whether to enable or disable events.	No	Enables or disables dispatch call events. When events are enabled, events will be sent for all existing calls. Returns true if successful.
Tab control			
Note: These methods are used to control (only) web tabs, i.e. those containing a web view displaying a URL: personal tabs, default URL, Agent Group tabs and Agent Action.			

METHOD	INPUT PARAMETERS	JS	DESCRIPTION
bool SetTabHeaderSymbol()	tabIndex – 0-based position index of the tab in the Agent application symbol – Unicode string to be shown in the tab header htmlColor – Color of the symbol, in HTML format (examples: red, #00FFFF)	Yes	Displays a symbol (or any string value) in a given color next to the name (title) of a tab. Returns true if successful.
bool SelectTab()	tabIndex – 0-based position index of the tab in the Agent application	Yes	Selects a particular tab, making it the currently active / visible one. Returns true if successful.
ITab[] GetAllTabs()	None	Yes	Returns an array of ITab objects, with one object for each visible web tab in the Agent application.
ITab GetTabByName()	name – Name of the tab	Yes	Performs a case-sensitive search for the complete given name in all visible tabs headers. Returns an ITab object, or null if no match is found.
bool TabReload()	tabIndex – 0-based position index of the tab in the Agent application	Yes	Reloads the current URL in the specified tab. Returns true if successful.
bool TabNavigate()	tabIndex – 0-based position index of the tab in the Agent application url – URL of the web page to load in the tab	Yes	Displays a new URL in a specific tab. Returns true if successful.

IUSERINFORMATION

The IUserInformation interface is returned by the GetUserInformation request. It supports the following properties:

PROPERTY	DESCRIPTION
string ExtensionNumber	Indicates the extension number used by the Agent application.
int LoggedOn	Indicates 0 if the client is logged off, or 1 if the client is logged on.
string LoginID	Indicates the logon ID used by the Agent application.
string UserName	Indicates the name of the user currently logged on to the Agent application.
bool ReadyStatus	Indicates the current Ready/Not Ready status for Voice sessions.

bool EmailReadyStatus	Indicates the current Ready/Not Ready status for E-mail and SMS messages of the Agent application.
bool OpenMediaReadyStatus	Indicates the current Ready/Not Ready status for Open Media messages of the Agent application.
int OASID	The OAS ID of the OAS server that the Agent application is connected to.
string OASServerName	The name of the OAS server that the Agent application is connected to.
bool ChatReadyStatus	Indicates the current Ready/Not Ready status for chat sessions.
int RecordID	The record ID of this agent in the SQL database.
int VoiceNotReadyReasonID	The record ID of the Not Ready Reason entered for voice sessions. Set to 0 if no reason was entered, or the agent is not currently in Not Ready state for voice sessions.
string VoiceNotReadyReasonString	The string value of the Not Ready Reason entered for voice sessions. Set to an empty string if no reason was entered, or the agent is not currently in Not Ready state for voice sessions.

ICALL

The ICall interface is returned by the GetCalls request for each call on the MiContact Center Agent. It supports the following properties:

PROPERTY	DESCRIPTION
string CallerName	Name of the opposite party, if available.
ICallInformation CallInfo	ICallInformation interface with additional information about the call.
ICampaignCallInformation CampaignCallInfo	ICampaignCallInformation interface with additional information about the call if it is a campaign call.
bool CanAnswer	Indicates true if the call is in a state where it can be answered.
bool CanDeflect	Indicates true if the call is in a state where it can be deflected.
bool CanHangup	Indicates true if the call is in a state where it can be disconnected.
bool CanHold	Indicates true if the call is in a state where it can be placed on hold.
bool CanRetrieve	Indicates true if the call is in a state where it can be retrieved.
bool CanReject	Indicates true if the call is in a state where it can be rejected.
Date ConnectedTime	Amount of time in seconds that the call has been in Talking state.
ICallInformation ConsultingCallInfo	If the call is a service group call transferred by another MiContact Center agent, indicates additional information about the call.
Date EndTime	Date and time that the call ended.

PROPERTY	DESCRIPTION
int ID	Indicates the call ID.
bool Incoming	Indicates true if the call is an incoming call, or false if it was initiated by the Agent application.
String PhoneNumber	Number of the opposite party.

CallRecordingState RecordingState	Current recording state of the call, if it is being recorded.
	Stopped = 0
	Paused = 1
	Dialing = 2

Date StartTime	Date and time that the call began.

SessionState State	State of the call, as follows: Failed = -1
	Undefined = 0
	Idle = 1
	Dialing = 2
	Offered = 3
	Calling = 4
	Connected = 6
	Held = 8
	On Hold = 9
	Conference = 10
	Clearing = 11
	Clerical = 99
	Force Clearing = 100

String StateDescription	String representation of the current call state.
string PrivateData	String representation of the private data associated with the call.
string CalledNumber	Phone number that was originally called.

CallType Type	Type of the call.
	Unknown = 0
	Incoming = 1
	Outgoing = 2
	Consultation = 3
	Callback = 4

ICALLINFORMATION

The ICallInformation interface is returned in the ICall interface for service group calls. It supports the following properties:

PROPERTY	DESCRIPTION

PROPERTY	DESCRIPTION
string CallingNumber	Number of the opposite party.
string GroupName	Name of the service group associated with the e-mail.
int QueueDuration	Duration of time in seconds that the call waited in the service group for an available agent.
string[] IVRData	Array of strings with the IVR Data attached to the call.
string[] IVRLabels	Array of strings with the IVR Labels for the IVR Data attached to the call.

ICAMPAIGNCALLINFORMATION

The ICampaignCallInformation interface is returned in the ICall interface for campaign calls. It supports the following properties:

PROPERTY	DESCRIPTION
string DialedNumber	Number originally called.
int CampaignID	Identifier of the campaign.
string CampaignName	Name of the campaign.
int CustomerID	Identifier of the customer.
string CustomerName	Name of the customer.
string Comment	Comment added about the customer call.
int CallStatus	Status of the customer call. 1 = Not Yet Called 2 = Busy 4 = No Answer 8 = Callback Later 16 = Completed 32 = Wrong Number 64 = Do Not Call Again
ICampaignCustomrField[] Fields	Array of campaign customer data fields.

IEMAIL

The IEmail interface is returned by the GetEmails request for each e-mail currently handled by the Agent application. It supports the following properties:

PROPERTY	DESCRIPTION
string BodyText	Body text of the e-mail.

PROPERTY	DESCRIPTION
DateTime ConnectedTime	Amount of time in seconds that the e-mail has been in Handling state.
DateTime EndTime	Date and time that the e-mail ended.
string GroupName	Name of the service group associated with the e-mail.
int ID	Indicates the unique ID of the e-mail.
string SenderEmailAddress	E-mail address of the e-mail originator.
DateTime StartTime	Date and time that the e-mail arrived at the Agent application.
SessionState State	State of the call, as follows: Undefined = 0 Email/SMS Offered = 101 Email/SMS Handling = 102
string StateDescription	String representation of the current call state.
string Subject	Subject of the e-mail.

ISMS

The ISMS interface is returned by the GetSMS request for each SMS currently handled by the Agent application. It supports the following properties:

PROPERTY	DESCRIPTION
DateTime ConnectedTime	Amount of time in seconds that the SMS has been in Handling state.
DateTime EndTime	Date and time that the SMS ended.
string GroupName	Name of the service group associated with the SMS.
int ID	Indicates the unique ID of the SMS.
string Message	Indicates the content of the SMS.
string SenderNumber	Number of the SMS originator.
DateTime StartTime	Date and time that the SMS arrived at the Agent application.
SessionState State	State of the call, as follows: Undefined = 0 Email/SMS Offered = 101 Email/SMS Handling = 102
string StateDescription	String representation of the current call state.

IOPENMEDIA

The IOpenMedia interface is returned by the GetOpenMedia request for each open media session currently handled by the Agent application. It supports the following properties:

PROPERTY	DESCRIPTION
int ID	Indicates the unique ID of the open media session.

SessionState State	State of the open media session, as follows:
	Failed = -1
	Undefined = 0
	Idle = 1
	Dialing = 2
	Offered = 3
	Calling = 4
	Connected = 6
	Held = 8
	On Hold = 9
	Conference = 10
	Clearing = 11
	Clerical = 99
	Force Clearing = 100

string StateDescription	String representation of the current open media session state.
ICallInformation CallInfo	ICallInformation interface with additional information about the open media session.
DateTime StartTime	Date and time that the open media session arrived at the Agent application.
DateTime ConnectedTime	Date and time that the open media session was first connected.
DateTime EndTime	Date and time that the open media session ended.

ICHAT

The IChat interface is returned by the GetChats request for each chat session currently handled by the Agent application. It supports the following properties:

PROPERTY	DESCRIPTION
int ID	Indicates the unique ID of the chat session.
string GroupName	Name of the chat service group.
string CustomerName	Name of the customer associated with the chat.
string CustomerEmailAddress	E-mail address of the customer, if available.

SessionState State	State of the chat session, as follows:
	Failed = -1
	Undefined = 0
	Idle = 1
	Calling = 2
	Offered = 3
	Connected = 6
	Held = 8
	On Hold = 9
	Conference = 10
	Clearing = 11
	Clerical = 99
	Force Clearing = 100

string StateDescription	String representation of the current chat session state.
DateTime StartTime	Date and time that the chat session arrived at the Agent application.
DateTime ConnectedTime	Date and time that the chatsession was first connected.
DateTime EndTime	Date and time that the chat session ended.

IDISPATCHPERSONALCALL

The IDispatchPersonalCall interface is supplied during the OnDispatchPersonalCallEvent. It supports the following properties:

PROPERTY	DESCRIPTION
int ID	Indicates the unique ID of the dispatch call.
string Number	Phone number of the calling party.
int PreferredAgentID	ID of original receiving agent.
string PreferredAgentName	Name of the original receiving agent.
int ServiceGroupID	ID of the service group associated with the dispatch call.
DateTime QueueStartTime	Date and time that the dispatch call entered the queue.
bool GeneralCall	True if the dispatch is generally available to all agents. The call will be specific to the original receiving agent until the configured timeout occurs. Then it will be generally available.
string IVR	IVR data associated with the dispatch call.

IDISPATCHCOMMONHOLDCALL

The IDispatchCommonHoldCall interface is supplied during the OnDispatchCommonHoldCallEvent. It supports the following properties:

PROPERTY	DESCRIPTION
int ID	Indicates the unique ID of the dispatch call.
string Number	Phone number of the calling party.
int ReservedAgentID	ID of agent that has the call reserved or 0 if unreserved.
string ReservedAgentName	Name of the agent that has the call reserved or empty if unreserved.
DateTime QueueStartTime	Date and time that the dispatch call entered the queue.

IDISPATCHVOICECALL

The IDispatchVoiceCall interface is supplied during the OnDispatchVoiceCallEvent. It supports the following properties:

PROPERTY	DESCRIPTION
int ID	Indicates the unique ID of the dispatch call.
int ReservedAgentID	ID of agent that has the call reserved or 0 if unreserved.
string ReservedAgentName	Name of the agent that has the call reserved or empty if unreserved.
int ServiceGroupID	ID of the service group associated with the dispatch call.
DateTime QueueStartTime	Date and time that the dispatch call entered the queue.
string Number	Phone number of the calling party.
string CalledNumber	Phone number that was originally called.
string IVR	IVR data associated with the dispatch call.

IDISPATCHEMAIL

The IDispatchEmail interface is supplied during the OnDispatchEmailEvent. It supports the following properties:

PROPERTY	DESCRIPTION

int ID	Indicates the unique ID of the dispatch call.
int ReservedAgentID	ID of agent that has the call reserved or 0 if unreserved.
string ReservedAgentName	Name of the agent that has the call reserved or empty if unreserved.
int ServiceGroupID	ID of the service group associated with the dispatch call.
DateTime QueueStartTime	Date and time that the dispatch call entered the queue.
string EmailAddress	E-mail address of the sender.

IDISPATCHSMS

The IDispatchSMS interface is supplied during the OnDispatchSMSEvent. It supports the following properties:

PROPERTY	**DESCRIPTION**
int ID	Indicates the unique ID of the dispatch call.
int ReservedAgentID	ID of agent that has the call reserved or 0 if unreserved.
string ReservedAgentName	Name of the agent that has the call reserved or empty if unreserved.
int ServiceGroupID	ID of the service group associated with the dispatch call.
DateTime QueueStartTime	Date and time that the dispatch call entered the queue.
string SMSAddress	SMS address of the sender.

IDISPATCHCAMPAIGNCALL

The IDispatchCampaignCall interface is supplied during the OnDispatchCampaignCallEvent. It supports the following properties:

PROPERTY	**DESCRIPTION**
int ID	Indicates the unique ID of the customer.
string Name	Name of the customer.
string Number	Phone number of the customer.
int CampaignID	ID of the campaign.
int ServiceGroupID	ID of the service group associated with the dispatch call.
Int ReservedAgentID	ID of agent that has the call reserved or 0 if unreserved.
String ReservedAgentName	Name of the agent that has the call reserved or empty if unreserved.

PROPERTY	DESCRIPTION
int PreferredAgentID	ID of the agent preferred agent or 0 if none.
string PreferredAgentName	Name of the preferred agent or empty if none.
ICampaignCustomerField[] Fields	Array of fields associated with the customer.

ICAMPAIGNCUSTOMERFIELD

The ICampaignCustomerField interface supports the following properties:

PROPERTY	DESCRIPTION
int Label	Name of the field.
string Data	Field data.

INOTREADYREASON

The INotReadyReason interface supports the following properties:

PROPERTY	DESCRIPTION
int ID	ID of the Not Ready Reason
string Name	Name of the Not Ready Reason

ITAB

The ITab interface is returned by the GetTabByName and GetAllTabs requests for all web tabs currently visible in the Agent application. It supports the following properties:

PROPERTY	DESCRIPTION
int Index	0-based position index of the tab in the Agent application
string Name	Name of the tab
string Symbol	Symbol (or string value) displayed next to the name (title) of the tab.
bool Selected	Indicates whether the tab is currently selected or not.

EVENTS

The AgentClient object provides the following events as information on the Agent application is updated. For the .NET implementation, these are implemented as Event Handlers.

EVENT	PARAMETERS	DESCRIPTION
OnApplicationClosed	None	This event is generated when the Agent application shuts down.
OnApplicationReadyForInitialization	None	This event is generated when the Agent application is running and Initialize() can be called to connect to the Agent application.
OnCallEvent	ICall- ICall interface with additional information about the call.	This event is generated when the state or another attribute of a voice call on the Agent application changes.
OnEmailEvent	IEmail - IEmail interface with additional information about the e- mail.	This event is generated when the state or another attribute of an e-mail on the Agent application changes.
OnSMSEvent	ISMS - ISMS interface with additional information about the call.	This event is generated when the state or another attribute of an SMS on the Agent application changes.
OnOpenMediaEvent	IOpenMedia – IopenMedia interface with additional information about the open media session.	This event is generated when the state or another attribute of an open media session on the Agent application changes.
OnRegistrationStatus	Registered – Boolean value indicating whether the agent’s extension is registered with the switching interface.	This event is generated when the registration status of the SIP extension used by the Agent application changes.
OnUserLogin	None	This event is generated when the user logs on to the Agent application.
OnDispatchPersonalCallEvent	IDispatchPersonalCall – IDispatchPersonalCall interface with additional information about the dispatch call. UpdateType – Type of update: 0 = Add, 1 = Update, 2 = Delete	This event is generated when a dispatch personal call is added, modified or deleted.
OnDispatchCommonHoldCallEvent	IDispatchCommonHoldCall – IDispatchCommonHoldCall interface with additional information about the dispatch call. UpdateType – Type of update: 0 = Add, 1 = Update, 2 = Delete	This event is generated when a dispatch common hold call is added, modified or deleted.
OnDispatchVoiceCallE	IDispatchVoiceCall –	This event is generated when a dispatch voice call is

EVENT	PARAMETERS	DESCRIPTION
vent	IDispatchVoiceCall interface with additional information about the dispatch call. UpdateType – Type of update: 0 = Add, 1 = Update, 2 = Delete	added, modified or deleted.
OnDispatchEmailEvent	IDispatchEmail – IDispatchEmail interface with additional information about the dispatch call. UpdateType – Type of update: 0 = Add, 1 = Update, 2 = Delete	This event is generated when a dispatch e-mail is added, modified or deleted.
OnDispatchSMSEvent	IDispatchSMS – IDispatchSMS interface with additional information about the dispatch call. UpdateType – Type of update: 0 = Add, 1 = Update, 2 = Delete	This event is generated when a dispatch SMS is added, modified or deleted.
OnDispatchCampaign CallEvent	IDispatchCampaignCall – IDispatchCampaignCall interface with additional information about the dispatch call. UpdateType – Type of update: 0 = Add, 1 = Update, 2 = Delete	This event is generated when a dispatch campaign call is added, modified or deleted.
OnUserStatusUpdate	IUserInformation – IuserInformation interface with the current status of the agent.	This event is generated when an agent’s ready/not ready status changes for any logged on media types.
OnChatEvent	IChat – IChat interface with additional information about the chat session.	This event is generated when the state or another attribute of a chat session on the Agent application changes.

Mitel®
Powering connections

mitel.com