# **Presence Agent Web Services Integration**

Version 11.1 Date: 11/2018





# **Contents**

1. Introduction	9
2. Fundamentals	
Architecture	
Installation	
3. Examples	13
Introduction	13
Example 1	13
Example 2	21
Updating User Interface	27
4. Reference Guide	
Enumerations	30
TWSAgentState	
TWSCallState	
TWSContactDataVisibility	
TWSContactType	
TWSCTISkillStatus	
TWSCustomDataVisibility	
TWSEventCode	
TWSImageFormat	
TWSImageType	
TWSIntegrationType	34
TWSInternetSessionType	34
TWSMailStatus	35
TWSMailStatusFilter	36
TWSMailType	36
TWSOutboundType	37
TWSPhoneType	37
TWSPredictionType	37
TWSRecordState	38
TWSScriptLanguage	38
TWSServerState	39
TWSServerType	39
TWSServiceState	39
TWSServiceType	40
TWSSoundType	40
TWSTemplateFieldType	41
Events	41
AGM_ACTION_CHANGE	
AGM_ACW_EVENT	
AGM_ALTERNATIVEPHONES_CHANGE	42
AGM_CALL_ALERT	42
AGM_CAPTURED_CHANGE	42
AGM_CAPTUREDDATE_CHANGE	43

AGM_COMMENTS_CHANGE	. 43
AGM_CONTACTCALL_CHANGE	. 43
AGM_CONTACTDATA1_CHANGE	. 44
AGM_CONTACTDATA2_CHANGE	. 44
AGM_CONTACTDATA3_CHANGE	. 44
AGM_CONTACTNAME_CHANGE	. 45
AGM_CUSTOMBUTTONS_CHANGE	. 45
AGM_CUSTOMDATA1_CHANGE	. 45
AGM_CUSTOMDATA2_CHANGE	. 46
AGM_CUSTOMDATA3_CHANGE	. 46
AGM_END_CALL_ALERT	. 46
AGM_ENDCODE_CHANGE	. 46
AGM_EXTERNAL_STOP_SESSION	. 47
AGM_FATAL_ERROR	. 47
AGM_FINALIZE_CONTACT	. 47
AGM_FINDMAIL_EVENT	. 48
AGM_INBOUND_CALL	. 49
AGM_INTERNET_CALL	. 49
AGM_INTERNET_MULTICHAT_CONNECT	. 49
AGM_INTERNET_MULTICHAT_DISCONNECT	. 50
AGM_INTERNET_OPENURL	. 50
AGM_INTERNET_SERVICE_CONNECT	. 50
AGM_INTERNET_SERVICE_DISCONNECT	. 51
AGM_LAUNCH_URL	
AGM_LINES_CHANGE	. 51
AGM_LOGIN_EVENT	. 52
AGM_LOGOUT_EVENT	. 52
AGM_MAILIN_DECODED	. 52
AGM_MAILINATTOPEN_EVENT	. 52
AGM_MAILINATTSAVE_EVENT	. 53
AGM_MAILOUT_CHANGE	. 53
AGM_MAILOUTBCC_CHANGE	. 54
AGM_MAILOUTBODY_CHANGE	. 54
AGM_MAILOUTCC_CHANGE	. 54
AGM_MAILOUTCOMPOSE_EVENT	. 54
AGM_MAILOUTSUBJECT_CHANGE	. 55
AGM_MAILOUTTO_CHANGE	. 55
AGM_MAXACW_EVENT	. 55
AGM_MESSAGEALERT_EVENT	. 56
AGM_MESSAGESTATE_EVENT	. 56
AGM_MINACW_EVENT	. 57
AGM_OUTBOUND_CALL	. 57
AGM_PHONE_CHANGE	. 58
AGM_RECORDING_PAUSE	
AGM_RECORDING_RESUME	. 58
AGM_RECORDING_START	. 59
AGM_RECORDING_STOP	. 59
AGM_REQUEST_FINALIZE_CONTACT	. 59
AGM_SAVEPARTIALMAIL_CHANGE	
AGM_SCHEDULEDDATE_CHANGE	. 60
AGM_SERVER_STATE_CHANGE	. 60
AGM_SERVICE_CHANGE	

	AGM_SERVICE_CONNECT	
	AGM_SERVICE_DISCONNECT	. 61
	AGM_SKILLS_CHANGE	. 62
	AGM_SOUND_EVENT	. 62
	AGM_STATE_CHANGE	. 62
	AGM_STOP_SESSION	
	AGM_TRANSFER_COMPLETED	. 63
	AGM_TRANSFER_RECEIVED	. 63
Гу	pes	. 64
Ī	TWSAgentEvent	
	TWSAgentEventField	. 64
	TWSAgentServiceInformation	. 64
	TWSAlternativePhoneDescription	. 65
	TWSCall	. 65
	TWSCallDataInformation	. 66
	TWSContact	. 66
	TWSContactData	. 67
	TWSCustomButton	. 68
	TWSCustomData	. 68
	TWSImage	. 69
	TWSInboundContact	. 69
	TWSInboundService	. 70
	TWSInternetContact	. 71
	TWSMail	. 71
	TWSMailAddress	
	TWSMailAddressBook	
	TWSMailAddressGroup	
	TWSMailAttachment	
	TWSMailboxToMove	
	TWSMailContact	
	TWSMailGroup	
	TWSMailHistory	
	TWSMailIdError	
	TWSMailSuspended	
	TWSMultiChatService	
	TWSNoticeHeader	
	TWSOutboundContact	
	TWSOutboundContactHistory	
	TWSOutboundPhone	
	TWSOutboundService	
	TWSOrd of fine dT suf	
	TWSPredefinedTextTreeItem	
	TWSQLofe	
	TWSQInfo	
	TWSQInfo2	
	TWSQInfo3TWSQInfoInbound	
	TWSQInfoMail	
	TWSQInfoOutbound	
	TWSReason	. 00 88
	T VV II VEGOVIII	OO

	TWSServer	89
	TWSServerList	
	TWSService	90
	TWSSessionInfo	92
	TWSSkill	93
	TWSSkillActionResult	
	TWSSoftPhoneInfo	
	TWSSurvey	94
	TWSTemplateField	
	TWSTemplateTreeItem	95
	TWSTimeZone	95
	TWSTransferPhoneInfo	95
	TWSTransferServiceInfo	96
	TWSWaitingCallAlert	96
Li	sts	07
	TWSAgentEventFieldList	-
	TWSAgentEventList	
	TWSAgentServiceInformationList	
	TWSAlternativePhoneDescriptionList	
	TWSAttachmentIdList	
	TWSCallDataInformationList	
	TWSCustomButtonList	
	TWSImageList	
	TWSLinesList	
	TWSMailAddressBookList	
	TWSMailAddressGroupList	
	TWSMailAddressList	
	TWSMailAttachmentList	
	TWSMailboxToMoveList	
	TWSMailGroupList	
	TWSMailHistoryList	
	TWSMailIdErrorList	100
	TWSMailIdList	100
	TWSMailSuspendedList	100
	TWSMultiChatServiceInfoList	101
	TWSNoticeHeaderList	101
	TWSOutboundContactHistoryList	101
	TWSOutboundPhoneList	101
	TWSPredefinedTextTreeItemList	101
	TWSQCodeList	102
	TWSReasonList	102
	TWSServiceInfoList	102
	TWSSkillActionResultList	102
	TWSSkillExtensionList	102
	TWSSkillList	
	TWSSurveyList	103
	TWSTemplateFieldList	
	TWSTemplateTreeItemList	103
	TWSTimeZoneList	
	TWSTransferPhoneList	
	TWSTransforSorvical ist	104

TWSWaitingCallAlertList	104
Methods	104
AddCallData	104
AddOutgoingMailAttachment	105
AnswerCall	105
ChangeAgentPassword	106
CheckTimeZonePrefix	106
CheckTimeZoneSchedule	107
CheckTimeZoneService	108
ClearAllCalls	108
ClearCall	109
CloseContact	109
ConferenceCall	109
DeleteCallData	110
DeleteOutgoingMailAttachment	110
DropCall	111
Echo	111
ExecuteCustomButton	112
ExecuteScriptCode	112
FinalizeContact	113
FinalizeContact2	113
FinalizeContact3	113
FinalizeUnansweredMails	114
GetAgentCount	114
GetAgentName	115
GetAgentState	115
GetAgentStation	116
GetAllCallData	116
GetAlternativePhoneDescription	117
GetAlternativePhones	117
GetAppData	118
GetCallData	118
GetCanChangePassword	119
GetContactData1	119
GetContactData2	120
GetContactData3	120
GetCurrentContact	121
GetCurrentInboundContact	121
GetCurrentInboundService	122
GetCurrentInternetContact	122
GetCurrentMailContact	123
GetCurrentOutboundContact	123
GetCurrentOutboundService	124
GetCurrentService	124
GetCurrentServiceLogo	125
GetCustomButtons	
GetCustomButtons2	125
GetCustomData1	126
GetCustomData2	
GetCustomData3	
GetEvent	127

GetIncomingMailAttachment	
GetIncomingMailAttachmentAsMail	129
GetInvalidQCodes	129
GetLines	130
GetLogoutReasons	130
GetMailAddressBooks	131
GetMailboxesToMove	131
GetMailHistoryAttachment	132
GetMailHistoryAttachmentAsMail	133
GetMailHistory	134
GetMailHistoryLog	134
GetMailsSuspended	135
GetManagedSkills	136
GetNoticeBody	136
GetNoticesHeaders	137
GetOutboundContactHistory	137
GetOutgoingMail	138
GetPartialMailAttachment	
GetPhoneDescriptions	139
GetPredefinedText	139
GetPredefinedTextTreeItems	140
GetQCode	140
GetQInfo	141
GetQInfo2	141
GetQInfo3	141
GetScheduledQCodes	142
GetServerStatus	142
GetServerTime	143
GetServiceAgents	143
GetServiceCallSound	144
GetServiceName	144
GetServiceQCodes	145
GetSessionCallSound	146
GetSessionInfo	146
GetSoftPhoneBooks	147
GetStopReasons	147
GetTemplate	148
GetTemplateAttachment	148
GetTemplateTreeItems	149
GetTimeZones	149
GetUnreachableQCodes	150
GetWaitingCallMessageAlert	150
HoldCall	151
InitializeOutgoingMail	151
InsertOutboundRecord	152
InsertOutboundRecord2	152
IsValidQCode	156
LoadMailOutInfo	157
Login	157
Logout	158
MakeCall	159
MakeConsultationCall	150

MakeContactCall	160
MakeDirectAgentCall	160
MakeDirectAgentConsultationCall	161
MakeServiceCall	162
NoticesAcknowledge	162
PauseSkills	163
RecordPause	163
RecordStart	
RecordStop	
RequestOutboundACDCall	165
RequestOutboundACDCall2	165
ResumeSkills	166
RetrieveCall	166
RetrieveMailSuspended	167
SaveOutgoingMailAsPartial	167
SelectLine	168
SendDTMFTones	
SendOutgoingMail	169
SendOutgoingMailHistory	169
SetAlternativePhones	170
SetAppData	170
SetContactData1	
SetContactData2	172
SetContactData3	
SetCustomData1	
SetCustomData2	
SetCustomData3	174
SetNoticeRead	
SetOutgoingMail	
SetQCode	
SetQInfo	
SetQInfo2	
SetQInfo3	
SetSourceId	
SingleStepTransferCall	
SingleStepTransferDirectAgentCall	
StartSession	
StopSession	179
Sum	
TransferCall	
TransferSurveyCall	
UnblockWaitForEvent	
UnloadOutboundRecord	
WaitForEvent	182
5. Appendices	183
Troubleshooting	183
Error Codes	186

# Chapter

# Introduction



This manual is meant for technical personnel who have experience or, at least, a basic knowledge of programming fundamentals. It is also assumed that the user is familiar with the Presence Suite as well as the Presence Agent application and its functionality.

The Web Services allow for the creation of tailor-made agent toolbars that support both inbound and outbound voice services. This is particularly useful for those integrations where the existing agent toolbar or the other third-party integration options available (ActiveX, HTTP, ScriptCode) are not suitable or powerful enough to fully satisfy the customer's needs.

This document sets out to provide the necessary basis for developing an integrated application using the Presence Agent Web Services, while providing both a useful reference guide and examples for the development stage of such application.

Any programming language supporting this type of technology (such as C#, Visual Basic, Java, Delphi or JavaScript) may be used to develop integrated applications via the Presence Agent Web Services. Likewise, other computing systems may be integrated, either by adding the Web Services to the system -if applicable-, or by using "bridge" applications developed in any of the above-mentioned languages.

This document is divided into four chapters, which describe the implementation and development test processes of an application to be integrated with the Presence Agent Web Services.

*Fundamentals.* Explains the basics that must be known before proceeding with any integration process with the Web Services supplied by the Presence Agent Web Services.

*Examples.* This chapter is meant to be a practical guide to designing an application to be integrated with the Presence Agent Web Services. It describes both the process of adding the Web Services to the development tool and the creation of an application from scratch. The examples in this chapter have been developed in the C# and JavaScript/HTML programming languages.

*Reference Guide.* This chapter describes all enumerations, events, types, lists and methods provided by the Presence Agent Web Services. For each of these items, further information is provided, e.g. the fields, values or required parameters, the return values, or the C# syntax.

*Troubleshooting.* This chapter is very useful in the stage of development and test of the created application, since the most common problems occurring when integrating an application with the Presence Agent Web Services are listed here. The troubleshooting method for each problem is described in a quick, easy way.

# Chapter

# **Fundamentals**

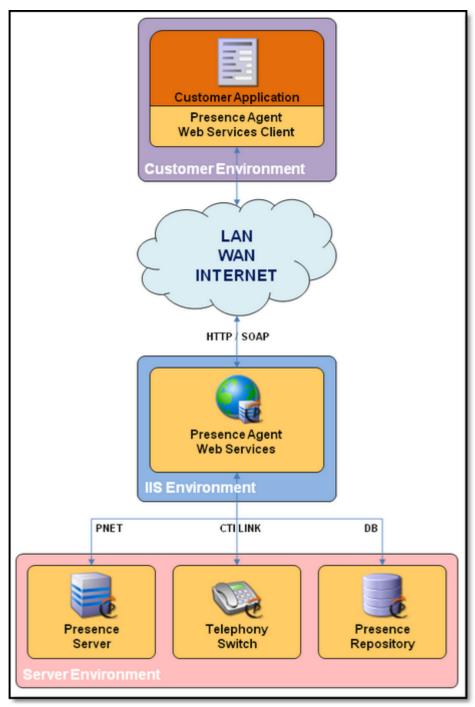
2

## **Architecture**

The Presence system allows you to integrate third-party applications to carry out actions related to the agent's activity that have been processed using the provided Web Services. These Web Services, called Presence Agent Web Services, can be included in a third-party application that will be able to invoke methods and receive events to perform these actions.

As shown in the figure below, the Client object establishes communication between the third-party application and the Web Services. In general terms, the customer application uses HTTP / SOAP protocol to communicate with the Presence Agent Web Services. This communication can be carried out using a local network within a company environment, through global networks or directly through the Internet.

At the same time, the Presence Agent Web Services communicate directly with the Presence Suite, Presence Server, the chosen telephony solution (OpenGate / Avaya) and the Presence database repository. The Web Services manage the information in each system in order to handle an ACD contact, generate a call, store changes of a handled contact to the repository, etc.

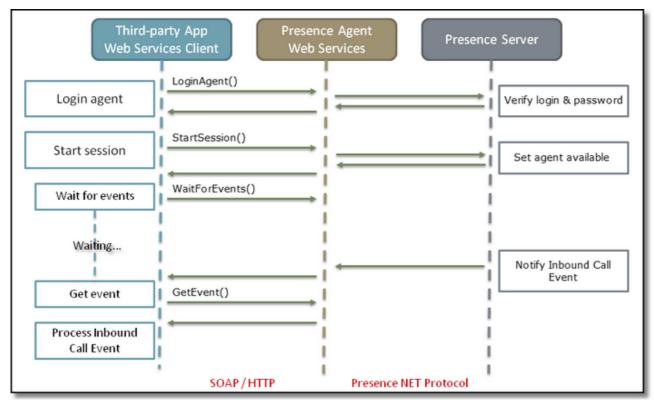


Integration architecture with the Presence Agent Web Services

This type of integration allows you to process events received through the Presence Agent Web Services. These events are the same that those received by Presence Agent which affect the interface, contact management, CTI events, etc.

Events allow you to act in a timely manner without having to constantly monitor a particular state. For example, these events notify you when an ACD contact should be handled, a call has been hung up, etc.

The following sequence diagram shows the way these events are managed. In this example, after the agent logs in and starts the session, an inbound call event is received.



Event sequence diagram sample

## Installation

Presence Agent Web Services are automatically installed when executing the Presence Web Agent setup program. For more information, refer to the <u>Presence Suite, Clients, Web applications, Presence Web Agent</u> section in the <u>Presence Installation Guides.</u>

# Chapter **Examples** 3

#### Introduction

This chapter describes the steps to create two integrated applications with the Presence Agent Web Services —one in C# language and the other one in JavaScript/HTML language. The examples below are intended to be used as a reference guide when developing a new application to be integrated with the Presence Agent Web Services, providing the basic programming structures using Web services.

Basic programming skills on C# or JavaScript/HTML languages are required. The C# sample has been developed in Visual Studio 2008.

# **Example 1**

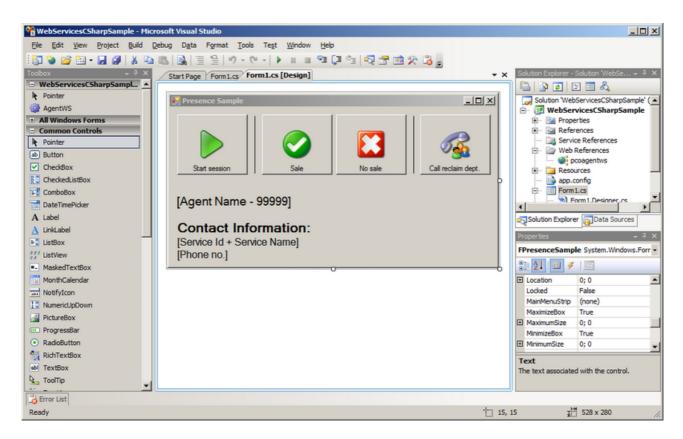
Sample application in C#

This example shows how to call Web service methods in C# language in order to perform certain actions, like starting/stopping the agent's session, qualifying and closing the contact, and calling a reclaim department. It can also be used to retrieve the Presence Agent Web Services events.

The Form1.cs file, which is located by default in the .

\inetpub\wwwroot\agentws\Samples\CSharp path of the server where Presence Web Agent has been installed, contains the full source code for this sample application.

The main form consists of a set of buttons that will execute the actions of the example and a set of labels to retrieve agent and contact information:



In this form, each button is assigned a specific agent action:

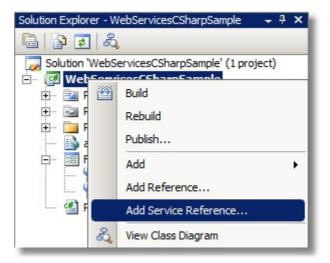
- Start/Stop session: This button implements the agent session control. It starts the agent session if the agent is stopped or it makes a request to stop the session if the agent is started. To make these requests, simply call the <u>StartSession</u> and <u>StopSession</u> Web service methods.
- Sale: This button is enabled when a contact event is received (inbound or outbound). It
  automatically hangs up the call and qualifies the contact as a positive contact (sale). The

  <u>ClearCall</u> method will be used to hang up the call, the <u>SetQCode</u> method will be used to
  select a qualification code for the contact (in this example, code 100 indicates that a sale
  has been achieved) and the <u>CloseContact</u> method will be used to finish the current
  contact.
- No sale: This button works in the same way as the Sale button, except that the selected qualification code is different (in this case code 101 means that a sale has not been achieved).
- Call reclaim dept.: This button is used to make a call to the reclaim department if the call from the contact is a complaint. You just need a <u>MakeCall</u> method to call the phone number of the reclaim department (in this case, 50123).

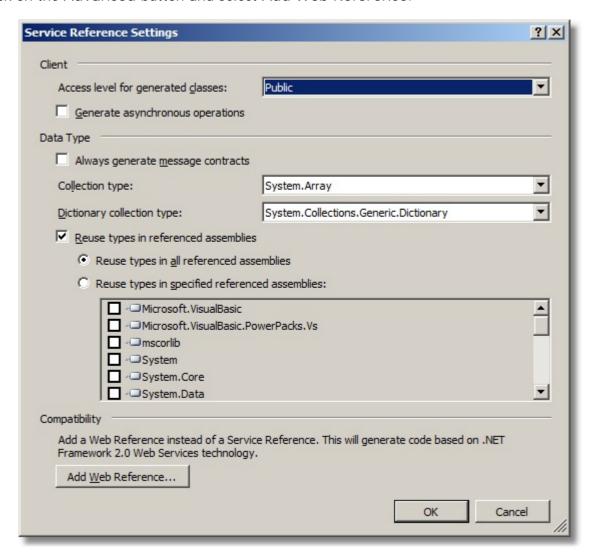
In this example, the agent login and logout operations will be executed automatically when starting and exiting the application.

Adding the Web Reference to the Presence Agent Web Services

To add the Web Reference to your solution, select Add Service Reference by right-clicking on the project name in the Solution Explorer.

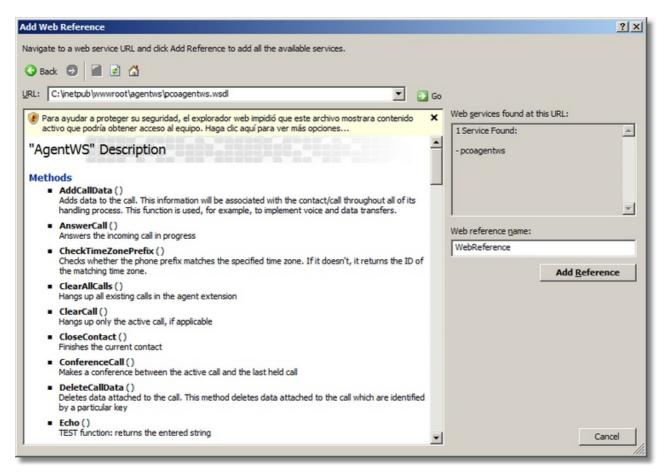


Click on the Advanced button and select Add Web Reference.



In the URL field, the path where the pcoagentws.wsdl file is located must be specified. This file can be copied locally from the .\inetpub\wwwroot\agentws path of the server where Presence Web Agent has been installed, or referred by the

http://<WebAgentServerDomain>/agentws/pcoagentws.wsdl URL. Next, rename the Web Reference to pcoagentws and click on the Add Reference button.



The Web Reference should now appear in your Solution Explorer.

NOTE: Before you can execute the application, you need to specify the location where the web services will be executed. To do this, click the Project menu item of Microsoft Visual Studio and then select the project properties. The Settings tab of the project will show a key named WebServicesCSharpSample\_pcoagentws\_AgentWS, and you must change the Value field to specify the location of the web services (http://<WebAgentServerDomain>/agentws/pcoagentws.dll/SOAP?service=AgentWS).

#### Functional code

Once the Web Reference has been added, a client instance of the Presence Agent Web Services can already be created. In order to do it, you must declare a global variable in the application as follows:

```
// Agent Web Services
static AgentWS AWS = new pcoagentws.AgentWS();
```

When starting the application, call the *Initialize* function which performs the agent connection to the Presence system, as well as initializing some application parameters. To perform the agent connection, this function calls the *Login* Web service method:

```
// Logs the agent in
private int Initialize()
{
```

```
// Disables main bar buttons
   ContactButtonsState(false);
   // Creates and initializes CultureInfo,
   // needed to specify Web Scripting locale settings (not used in this example)
   CultureInfo myClintl = new CultureInfo("en-US", false);
   LCID = myClintl.LCID;
   // Gets local IP address
   IP = GetLocalIP();
   // Logs in the user agent (Login) to Presence Server
   int Error = AWS.Login(Login, ref SessionId, Password, Station, true, "", IP,
      Language, LCID);
   return Error;
}
private void FPresenceSample_Load(object sender, EventArgs e)
   Initialize();
}
```

When closing the application, call the *Deinitialize* function in order to perform the agent disconnection from the Presence system. In order to do it, use the *Logout* Web service method:

```
// Logs the agent out
private int Deinitialize()
{
   int Error = AWS.Logout(SessionId, 0);
   return Error;
}

private void FPresenceSample_FormClosed(object sender, FormClosedEventArgs e)
{
   Deinitialize();
}
```

Next, the code to start and stop the agent session is shown. In order to perform these actions, use the <u>StartSession</u> and <u>StopSession</u> Web service methods:

```
// Starts the agent session
private Boolean StartSession()
{
   return (AWS.StartSession(SessionId) == NO_ERROR);
}

// Stops the agent session
private Boolean StopSession()
{
   return (AWS.StopSession(SessionId, 0) == NO_ERROR);
}
```

Refer to the source code of the example for information about actions other than those explained above.

Once the agent actions have been implemented, you must access the events that the Web Services report. To do this, you must generate an execution thread that is separate from the main execution of the application to capture these events. Making a separate execution thread will ensure that the main execution is not hindered thus preventing blocks when processing these events.

This separate thread will only make a request to the <u>GetEvent</u> method that returns the pending events to be received. In order not to have to continuously repeat this request, the <u>WaitForEvent</u> method is provided. This method remains blocked until an event is received or a timeout is reached. This is the reason why this logic must be implemented using a separate thread from the main execution, otherwise it would block the application. When the <u>WaitForEvent</u> method is unblocked, you must make a request to the <u>GetEvent</u> method to retrieve the events that have been reported.

In the example 3 events are controlled: AGM\_INBOUND\_CALL and AGM\_OUTBOUND\_CALL, which are responsible for reporting that a contact has been received (either inbound or outbound, respectively), in which case the basic information about the contact will be displayed and the contact qualification buttons will be enabled; and AGM\_FINALIZE\_CONTACT, which is responsible for reporting that the contact has been finished and the qualification buttons will be disabled:

```
// Agent Web Services event handling thread
Thread EventThread;
// WaitForEvent thread
public class WaitForEvent
   public FPresenceSample PresenceSample;
   // This method will be called when the thread is started
   public void WaitForEventExecute()
      TWSAgentEvent[] EventList = new TWSAgentEvent[0];
      while (true)
      {
         // Waits for Presence Agent Web Services events
         AWS.WaitForEvent(SessionId);
         // Gets pending events and processes them
         AWS.GetEvent(SessionId, out EventList);
         for (int i = 0; i < EventList.Length; i++)</pre>
            switch (EventList[i].EventType)
               case TWSEventCode.AGM_INBOUND_CALL:
               case TWSEventCode.AGM_OUTBOUND_CALL:
                  // New contact event received, enables main bar contact buttons
                  object[] pListTrue = { true };
                  PresenceSample.Invoke(new ContactButtonsStateCallback(
                     PresenceSample.ContactButtonsState), pListTrue);
                  PresenceSample.Invoke(new ContactInformationCallback(
                     PresenceSample.ContactInformation));
                  break:
               case TWSEventCode.AGM_FINALIZE_CONTACT:
                  // Contact finalization event received,
                  // disables main bar contact buttons
                  object[] pListFalse = { false };
                  PresenceSample.Invoke(new ContactButtonsStateCallback(
                     PresenceSample.ContactButtonsState), pListFalse);
                  break;
            }
         }
      }
   }
}
// Changes main bar button state
private void ContactButtonsState(Boolean State)
   btSale.Enabled = State;
```

```
btNoSale.Enabled = State;
  btClaims.Enabled = State;
}
// Delegate to change button state from thread
public delegate void ContactButtonsStateCallback(Boolean State);
// Changes the main form contact information
private void ContactInformation()
   string ServiceName;
   GetServiceName(out ServiceName);
  lService.Text = "Service: " + ServiceName;
  string Phone;
   GetContactPhone(out Phone);
   lPhone.Text = "Phone no.: " + Phone;
}
// Delegate to change the main form contact information from thread
public delegate void ContactInformationCallback();
```

Once the thread to receive events has been defined, it must be created and executed when starting the application. To create and execute the thread, the required code must be added to the *Initialize* function:

```
// Logs the agent in
private int Initialize()
   // Disables main bar buttons
   ContactButtonsState(false);
   // Creates and initializes CultureInfo,
   // needed to specify Web Scripting locale settings (not used in this example)
   CultureInfo myClintl = new CultureInfo("en-US", false);
   LCID = myClintl.LCID;
   // Gets local IP address
   IP = GetLocalIP();
   // Logs in the user agent (Login) to Presence Server
   int Error = AWS.Login(Login, ref SessionId, Password, Station, true, "", IP,
      Language, LCID);
   // Creates event handling thread
   WaitForEvent Events = new WaitForEvent();
   // Assigns this main form to the event handling thread
   Events.PresenceSample = this;
   \//\ Creates the thread object, passing in the WaitForEventExecute method
   // via a ThreadStart delegate. This does not start the thread
   EventThread = new Thread(new ThreadStart(Events.WaitForEventExecute));
   // Starts the thread
   EventThread.Start();
   return Error;
}
```

The thread must also be stopped when exiting the application. To do this, the required code must be added to the *Deinitialize* function:

```
// Logs the agent out
private int Deinitialize()
{
  int Error = AWS.Logout(SessionId, 0);
```

```
// Requests that EventThread be stopped
EventThread.Abort();

// Waits until EventThread finishes. Join also has overloads
// that take a millisecond interval or a TimeSpan object
EventThread.Join();

return Error;
}
```

How to read event fields

Some of the events received through the Presence Agent Web Services include fields with additional information about the event. In order to read these fields and their values, the <code>GetEventFields</code> function is provided, which can be found in the <code>PresenceAgentLib.cs</code> file located in the <code>.\inetpub\wwwroot\agentws\Resources\CSharp</code> path of the server where the Presence Web Agent has been installed.

Find below an example of use of this method in order to retrieve event fields and field values. Refer to the specific event in the reference guide for information about the fields that can be retrieved for each event.

```
// WaitForEvent thread
public class WaitForEvent
   public FPresenceSample PresenceSample;
   // This method will be called when the thread is started
   public void WaitForEventExecute()
      TWSAgentEvent[] EventList = new TWSAgentEvent[0];
      Dictionary<string, string> Fields;
      while (true)
         // Waits for Presence Agent Web Services events
         AWS.WaitForEvent(SessionId);
         // Gets pending events and process them
         AWS.GetEvent(SessionId, out EventList);
         for (int i = 0; i < EventList.Length; i++)</pre>
            // Method to retrieve the event field list and field values
            Fields = PresenceUtils.GetEventFields(EventList[i]);
            switch (EventList[i].EventType)
               case TWSEventCode.AGM_CONTACTNAME_CHANGE:
                  // Event received when the name of the contact changes
                  string Name = Fields["Name"];
                  break;
               case TWSEventCode.AGM_SCHEDULEDDATE_CHANGE:
                  // Event received when the contact scheduled date changes
                  DateTime Date = DateTime.Parse(Fields["Date"]);
                  break;
               case TWSEventCode.AGM_MESSAGEALERT_EVENT:
                  // Event received to notify changes in the notices,
                  // the recordings or the waiting calls
                  Byte AlertType = Byte.Parse(Fields["AlertType"]);
                  switch (AlertType)
                     case 0: // Notices
                        Boolean NewNotice = Boolean.Parse(Fields["NewNotice"]);
                        break;
```

The type of these field values is Variant; therefore they must be converted to their specific type. Refer to the specific event in the reference guide to identify the field type and see the following table for the conversion function to be used:

Field ty	ype	Conversion function
WideString	No	conversion required
Integer	Int	t32.Parse / Int32.TryParse
DateTime	Da	teTime.Parse / DateTime.TryParse
Boolean	Во	olean.Parse / Boolean.TryParse
Byte	Ву	te.Parse / Byte.TryParse

# **Example 2**

Sample application in JavaScript

With the aim of facilitating the use of the Presence Agent Web Services in web applications, Presence Technology supplies a JavaScript library that wraps the Web service calls and helps managing events. This library is located by default in the .

\inetpub\wwwroot\agentws\Resources\JavaScript path of the server where Presence Web Agent has been installed, and includes the following files:

- PresenceAgentSDK.js: contains the core code to call the Web Services.
- PresenceAgentIntf.js: contains all JavaScript functions and objects available to access the Web Services.
- PresenceAgentLib.js: contains the service initialization and the functions to manage Presence Agent Web Services events.

These files must always be included in any project or web application that will use JavaScript functions to invoke the Presence Agent Web Services. In order to add them, the following code must be added to the HEAD tag of the HTML document:

Invoking Web service methods

This section describes how to call Presence Agent Web Services methods using the JavaScript library. For each Web service method, there is an associated JavaScript function with the same name. For example, for the <u>Login</u> Web service method, the following JavaScript function is provided:

```
Service.Login(AgentLogin, SessionId, Password, Station, ClearCalls, AgentVersion, LocalIP, Language, LCIDScripting, CallbackResult, CallbackError);
```

As shown, the *Login* JavaScript function has all the input parameters of the *Login* method, plus two additional parameters: *CallbackResult* and *CallbackError*. These parameters are always included in all JavaScript functions associated with a Web service method. In the *CallbackResult* parameter, a JavaScript function must be passed. This function will be called by the JavaScript library in order to return the call result code, as well as the output parameters that the called method may have. For the *Login* method, apart from the call result code, an output parameter with the agent session identifier is returned:

```
function LoginCallback(result, SessionId) {
// Customer code goes here
}
```

Besides, in the *CallbackError* parameter a JavaScript function must also be passed. This function will be called by the JavaScript library if an error occurs when calling the Web service method. There will always be only one input parameter specifying the error that occurred:

```
function showError(msg) {
// Customer code goes here
}
```

#### Example

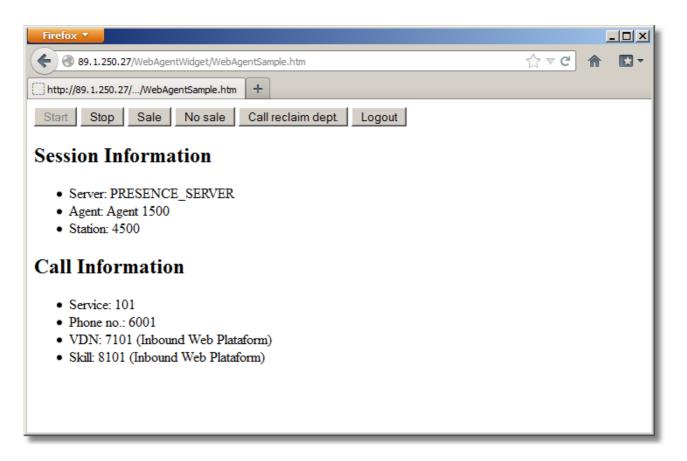
This example shows how to call Web service methods in JavaScript language in order to perform certain actions, like starting/stopping the agent's session, qualifying and closing the contact, and calling a reclaim department. They can also be used to retrieve Presence Agent Web Services events.

The following files, which are located by default in the .

\inetpub\wwwroot\agentws\Samples\JavaScript path of the server where Presence Web Agent has been installed, contain the full source code for this sample application:

- WebAgentSample.html: contains the web front end of the application.
- WebAgentSample.js: contains the JavaScript functions used for the application.

The HTML page consists of a set of buttons that will execute the actions of the example, and two sections that show information related to the agent session and the call that the agent is handling:



In this web page, each button is assigned a specific agent action:

- Start session: This button implements the agent session control. It starts the agent session if the agent is stopped. To make this request simply call the <u>StartSession</u> Web service method.
- Stop session: This button implements the agent session control. It makes a request to stop the session if the agent is started. To make this request simply call the <u>StopSession</u> Web service method.
- Sale: This button is enabled when a contact event is received (inbound or outbound). It
  automatically hangs up the call and qualifies the contact as a positive contact (sale). The
   <u>ClearCal/</u> method will be used to hang up the call, the <u>SetQCode</u> method will be used to
   select a qualification code for the contact (in this example, code 100 indicates that a sale
   has been achieved) and the <u>CloseContact</u> method will be used to finish the current
   contact
- No sale: This button works in the same way as the Sale button, except that the selected qualification code is different (in this case, code 101 means that a sale has not been achieved).
- Call reclaim dept.: This button is used to make a call to the reclaim department if the call from the contact is a complaint. You just need a <u>MakeCall</u> method to call the phone number of the reclaim department (in this case, 50123).
- Logout: This button sends a logout request to the Web Services.

NOTE: For more information, refer to the source code of the example, located by default in the .\inetpub\wwwroot\agentws\Samples\JavaScript path of the server where Presence Web Agent has been installed.

Functional code

The WebAgentSample.js file contains the JavaScript code used in this example. In this file you can find the *Initialize* function, which is called then the HTML page is loaded:

```
<body onload="Initialize()">
```

First, this function creates a client instance of the Presence Agent Web Services and assigns it to a variable. For that purpose the *StartPresenceAgentServices* function is used:

```
StartPresenceAgentServices(URLWEBSERVICES, SYNCMODE)
```

This function has two parameters:

- The URLWEBSERVICES parameter indicates the URL where the Web Services are located and its default value would be <a href="http://<ServerIP>/agentws/pcoagentws.dll/JSON">http://<ServerIP>/agentws/pcoagentws.dll/JSON</a>, where <a href="https://serverIP> is the IP of the server where the Presence Agent Web Services have been installed.">https://serverIP></a>
- The SYNCMODE parameter indicates whether the JavaScript functions associated with Web service methods will be executed in synchronous mode or not. If this parameter is set to TRUE, the JavaScript function associated with a method will remain "blocked" while waiting for an answer from the Web Services. After getting the answer, it will execute the specified callback function and return from the function. If the parameter is set to FALSE, the JavaScript function will not wait for an answer but return after making the call to the method, thus allowing the web application to continue its execution. Once the Web service answer is received, the specified callback function will be executed.

Once the client instance of the Presence Agent Web Services has been created, the <u>Login</u> method is called in order to connect the agent to the Presence system. The <u>LoginCallback</u> function is specified as the callback function to retrieve the call result. The full code of the *Initialize* and *LoginCallback* functions is as follows:

```
function Initialize() {
   // Creates Presence Agent Web Services
   PresenceAgentService = StartPresenceAgentServices
      ("http://<ServerIP>/agentws/pcoagentws.dll/JSON", false);
   // Some GUI initializations
   ContactButtonsState(false);
   ButtonEnabled('webAgentToolbarStop', false);
   // Sets global variable
   SessionId = '';
   // Logs in the user agent (Login) to Presence Server
   PresenceAgentService.Login(LoginId, SessionId, Password, StationId,
      true, '', IP, Language, LCID, LoginCallback, showError);
}
function LoginCallback(result, SessionIdResult) {
   if (result == NO_ERROR) {
      SessionId = SessionIdResult;
      // Runs ProcessEvents method if there are events
      StartCheckEvent(ProcessEvents);
      GetSessionInfo();
   } else showDialog('Error', 'Error in session login:' + result);
}
```

Next you can see the sample code associated with the Call reclaim dept. button, where the <u>MakeCall</u> Web service method is used to make a phone call to the extension 4011:

HTML code:

```
<button id="webAgentToolbarClaims" onclick="MakeCall ('4011','');">
```

}

The deinitialization process is done here to click on the Logout button, basically performing the agent <u>Logout</u> process. The <u>StopCheckEvent</u> function is used to stop receiving events:

At this point, the basic logic and the way to make requests to the Presence Agent Web Services have already been defined. For more information, refer to the source code of the example.

Next you can see how to receive events from the Web Services. In order to do it, simply define a JavaScript function to be called by the JavaScript library every time that an event is received from the Web Services. This function has only one parameter called *Events* that contains the list of received events to be processed. The sample code of this function is as follows:

```
function ProcessEvents(Events) {
   try {
      var numEvents;
      try {
         numEvents = Events.items.length;
      } catch (err) {
         // Avoids failure if no events
         numEvents = 0;
      }
      var i = 0;
      for (i = 0; i < numEvents; i++) {
         try {
            var Event = Events.items[i];
            var EventType = Event.EventType.value.value;
            switch (EventType) {
               case "CALL_ALERT":
                  AnswerCall();
                  break;
               case "AGM_INBOUND_CALL":
               case "AGM_OUTBOUND_CALL":
```

```
callAlert();
                  break;
               case "AGM_FINALIZE_CONTACT":
                  CloseContactAcepted(null);
                  break;
               case "AGM_STOP_SESSION":
                  alert("Your request to stop the session was ACCEPTED!");
                  ButtonEnabled('webAgentToolbarStop', false);
                  ButtonEnabled('webAgentToolbarStart', true);
                  break:
            }
         }
         catch (err) {
            alert('error process events:' + EventType);
   } finally {
}
```

In the example 5 events are controlled: CALL\_ALERT to pick up the phone for an incoming call; AGM\_I NBOUND\_CALL and AGM\_OUTBOUND\_CALL to report that a contact has been received (either inbound or outbound, respectively), in which case the basic information about the contact will be displayed and the contact qualification buttons will be enabled; AGM\_FINALIZE\_CONTACT to report that the contact has been finished and disable the qualification buttons; and AGM\_STOP\_SESSION to indicate that the agent state has changed to 'stopped'.

After defining the *ProcessEvents* function, it must be assigned as the callback function to receive events and start the notification process, for which the *StartCheckEvent* function of the JavaScript library should be used. In the example this function is called when the agent connection process is completed, i.e. in the callback function that has been assigned to the *Login* method call result:

```
function LoginCallback(result, SessionIdResult) {
   if (result == NO_ERROR) {
       SessionId = SessionIdResult;

       // Runs ProcessEvents function if there are events
       StartCheckEvent(ProcessEvents);
       GetSessionInfo();

   } else showDialog('Error', 'Error in session login' + result);
}
```

To stop receiving events, the *StopCheckEvent* must be called. In the example this function is called when the agent disconnection is completed, i.e. in the callback function that has been assigned to the *Logout* method call result:

```
function LogoutCallback(result) {
   if (result == 0) {
      // Stops receiving events
      StopCheckEvent();
      window.close();
   }
   else showDialog('Error', 'Error in session logout:' + result);
}
```

How to read event fields

Some of the events received through the Presence Agent Web Services include fields with additional information about the event. In order to read these fields and their values, the

GetEventFields function is provided, which can be found in the PresenceAgentLib.js file located in the .\inetpub\wwwroot\agentws\Resources\JavaScript path of the server where the Presence Web Agent has been installed.

Find below an example of use of this method in order to retrieve event fields and field values. Refer to the specific event in the reference guide for information about the fields that can be retrieved for each event.

```
function ProcessEvents(Events) {
   try {
      var numEvents;
      try {
         numEvents = Events.items.length;
      } catch (err) {
         // Avoids failure if no events
         numEvents = 0;
      }
      var i = 0;
      for (i = 0; i < numEvents; <math>i++) {
         try {
            var Event = Events.items[i];
            var EventType = Event.EventType.value.value;
            var Fields = GetEventFields(Event);
            switch (EventType) {
               case "AGM_CONTACTNAME_CHANGE":
                  // Event received when the name of the contact changes
                  var Name = Fields.Name;
                  break;
               case "AGM_SCHEDULEDDATE_CHANGE":
                  // Event received when the contact scheduled date changes
                  var Date = Fields.Date;
                  break;
               case "AGM_MESSAGEALERT_EVENT":
                  // Event received to notify changes in the notices,
                  // the recordings or the waiting calls
                  switch (Fields.AlertType) {
                     case 0: // Notices
                        var NewNotices = Fields.NewNotice;
                        break;
                     case 1: // Recordings
                        var RecordingId = Fields.RecordingId;
                        var RecordType = Fields.RecordType;
                        var RecordStatus = Fields.RecordStatus;
                        break:
                  break;
            }
         }
         catch (err) {
            alert('error process events:' + EventType);
   } finally {
}
```

# **Updating User Interface**

The examples in previous sections show how the interface changes as contact information events are received. Consequently these events program which actions should be allowed according to the application needs. While this choice is fully reliable, the following option may

as well be used in order to control whether or not those actions are allowed in case that a more advanced, alternative Presence Agent bar is chosen to be implemented.

Each time the availability of an action changes an event notification is sent, including information on the availability of all possible actions. This event is called <u>AGM\_ACTION\_CHANGE</u> and it includes an <u>ActionMask</u> field.

When this event is received, logical AND operations can be used to establish which actions are allowed and which are not allowed in that moment. For each action, a constant is available (refer to the table of constants below); by performing a logical AND between the constant and the *ActionMask* field value you can define whether the action is allowed or not. If the result of this operation is higher than 0 the action is allowed, whereas if the value is 0 the action is not allowed.

#### Table of constants

ACT\_TBV\_REC

Action name	Description
ACT_START	The agent session can be started
ACT_STOP	The agent session can be stopped
ACT_QUALIFY	The contact can be qualified
ACT_INFO_CONTACT	The contact information is available
ACT_SOFTPHONE	The phone books assigned to the agent and the services available for placing direct calls can be used
ACT_END_CONTACT	The contact can be finished
ACT_ALTERNATIVE	The alternative phone numbers can be used
ACT_CONTACTCALL	The call to the contact in preview mode can be generated
ACT_HISTORY	The history of the outbound record is available
ACT_SPH_ANSWERCALL	The active call line can be answered
ACT_SPH_HOLDCALL	The active call line can be held
ACT_SPH_RETRIEVECALL	The held call line can be retrieved
ACT_SPH_CLEARCALL	The active call line can be cleared
ACT_SPH_TRANSFERCALL	The active call line can be transferred
ACT_SPH_CONFERCALL	The active call line can be conferenced
ACT_SPH_MAKECALL	A new call line can be generated
ACT_SPH_CLEARALLCALL	All call lines can be hung up

Enghouse Interactive Presence Suite

The recording control actions (starting, resuming, pausing and stopping recording) are enabled

ACT\_REC\_START

The recording of the active call can be started/resumed

ACT\_REC\_PAUSE

The recording of the active call can be paused

ACT\_REC\_STOP

The recording of the active call can be stopped

Refer to the following files for the definition of these constants: PresenceAgentLib.js in the . \inetpub\wwwroot\agentws\Resources\JavaScript path of the server where Presence Web Agent has been installed (for JavaScript); or PresenceAgentLib.cs file in the . \inetpub\wwwroot\agentws\Resources\CSharp path of the server where Presence Web Agent has been installed (for C#).

#### Examples of use

• C#:

```
// Updates actions using the ActionMask field received
// in the AGM_ACTION_CHANGE event
internal void UpdateActions(int ActionMask)
{
    Boolean StartSessionAllowed, StopSessionAllowed, QualifyContactAllowed;

    StartSessionAllowed = ((ActionMask & AGFWEventsCode.ACT_START) > 0);
    StopSessionAllowed = ((ActionMask & AGFWEventsCode.ACT_STOP) > 0);
    QualifyContactAllowed = ((ActionMask & AGFWEventsCode.ACT_QUALIFY) > 0);
}
```

• JavaScript:

```
// Updates actions using the ActionMask field received
// in the AGM_ACTION_CHANGE event
function UpdateActions(ActionMask) {
  var StartSessionAllowed;
  var StopSessionAllowed;
  var QualifyContactAllowed;

  StartSessionAllowed = (ActionMask & ACT_START);
  StopSessionAllowed = (ActionMask & ACT_STOP);
  QualifyContactAllowed = (ActionMask & ACT_QUALIFY);
}
```

# Chapter

# Reference Guide



## **Enumerations**

# **TWSAgentState**

#### Description

Enumeration of the states in which the agent may be.

**Values** 

Value Description

STOPPED The agent is logged in to the system but the work

session has not been started yet or has been

stopped

STARTED The agent has just started the work session (this

state will be immediately replaced by AVAILABLE)

ATTENDING The agent is servicing a contact

ACW The agent is in after-call work, i.e. the call has been

hung up but the agent is performing certain tasks

before closing the contact handling

AVAILABLE The agent is ready to receive a contact after

starting a work session or closing a contact

handling

RINGING The agent is receiving a contact

## **TWSCallState**

#### Description

Enumeration of the states in which a softphone call line may be.

**Values** 

Value Description

NONE The line does not have any call

ORIGINATED The call has just been dialed out

*RING* The call is ringing

ACTIVE The call has been established

HELD The call has been held

RETRIEVED The call was held and has been retrieved

CLEARED The call has been finished

# TWSContactDataVisibility

#### Description

Enumeration of the access levels of agents to custom data values associated with contacts.

#### **Values**

Value Description

NOT\_SHOW Agents are not able to view nor modify the contact

custom data value

SHOW Agents are able to view the contact custom data

value but cannot modify it

SHOW\_EDIT Agents are able to view and modify the contact

custom data value

# TWSContactType

#### Description

Enumeration of contact types. It is determined by the type of service that the contact belongs to.

#### Values

Value Description

NONE No contact is being handled

PREVIEW The contact belongs to an outbound service in

preview mode

PROGRESSIVE The contact belongs to an outbound service in

progressive mode

PREDICTIVE The contact belongs to an outbound service in

predictive mode

MANUAL Deprecated

AUTOMATIC Deprecated

INBOUND The contact belongs to an inbound service

MAIL The contact is a mail contact

//NTERNET The contact is an Internet contact

## **TWSCTISkillStatus**

#### Description

Enumeration of the states in which the skills assigned to the agent login may be.

#### **Values**

Value Description

ENABLED The skill is enabled, therefore the agent can receive

contacts that are queued to that skill

PAUSED The skill has been paused by the agent, therefore

the agent cannot receive contacts that are queued

to that skill at the moment

# TWSCustomDataVisibility

#### Description

Enumeration of the access levels of agents to custom data values associated with outbound records.

#### **Values**

Value Description

NOT\_SHOW Agents are not able to view nor modify the record

custom data value

SHOW Agents are able to view the record custom data

value but cannot modify it

SHOW\_EDIT Agents are able to view and modify the record

custom data value

## **TWSEventCode**

#### Description

Enumeration of event codes. These events are used to report changes in the agent session.

#### Values

Refer to Events.

## **TWSImageFormat**

#### Description

Enumeration of image file formats for custom button icons.

#### **Values**

Value Description

/CO Microsoft Windows image file format for icons

PNG Portable Network Graphics

# TWSImageType

#### Description

Enumeration of types of image formats.

**Values** 

Value Description

BMP Windows Bitmap

JPG Joint Photographic Experts Group

GIF Graphics Interchange Format

PNG Portable Network Graphics

## TWSIntegrationType

#### Description

Enumeration of types of service integration with a third-party agent application.

#### **Values**

Value Description

GUID Deprecated

OCX Deprecated

NONE There is no integration with any kind of application

PROGRAMID Deprecated

ACTIVEX The integration is performed by using the Presence

Agent ActiveX component

SCRIPT The service is integrated using a script designed

with Presence Scripting. Only enabled if the corresponding Presence Scripting license is

available

URL The integration is performed by opening in the

default agent web browser the URL defined for the available agent session events (e.g. when a call is

answered, a contact is closed, etc.)

# TWSInternetSessionType

#### Description

Enumeration of Internet session types.

#### **Values**

Value Description

CALLBACK The Internet session is a callback session

CHAT The Internet session is a chat session

VIDEO The Internet session is a video session

## **TWSMailStatus**

#### Description

Enumeration of the states in which the incoming and outgoing e-mails may be.

#### **Values**

Value Description

///\_UNANSWERED The incoming e-mail has not been handled by

any agent yet

///\_PROGRESS The incoming e-mail is currently being handled

by an agent

IN\_COMPLETED The incoming e-mail handling has been

completed

///\_SUSPENDED The incoming e-mail has been suspended by an

agent

//N\_DELETED The incoming e-mail has been deleted

///\_WAITING\_QUEUE The incoming e-mail has been queued in the skill

and is pending to be handled by an agent

///\_SUSPENDED\_///\_PROCESS The suspended incoming e-mail is being

retrieved by an agent

OUT\_UNSENT The outgoing e-mail has not been sent yet

OUT\_PARTIAL The outgoing e-mail is a partial reply to a

suspended incoming e-mail

OUT\_SENT The outgoing e-mail has been sent

OUT\_ERRONEOUS The outgoing e-mail could not be sent due to an

error

OUT\_DELETED The outgoing e-mail has been deleted

#### **TWSMailStatusFilter**

#### Description

Enumeration of the state filters that may be used to search in the mail history.

#### **Values**

Value Description

ALL Any state

///\_UNANSWERED Incoming e-mails that have not been answered by

any agent yet

///\_PROGRESS Incoming e-mails that are currently being handled

by an agent

IV\_COMPLETED
Incoming e-mails for which the handling process

has been completed

Incoming e-mails that have been suspended by an

agent

OUT\_UNSENT Outgoing e-mails that have not been sent yet

OUT\_SENT Outgoing e-mails that have been sent

## **TWSMailType**

#### Description

Enumeration that indicates whether the e-mail is incoming or outgoing.

#### **Values**

Value Description

INCOMING Incoming e-mail

OUTGOING Outgoing e-mail

# TWSOutboundType

### Description

Enumeration of outbound service types.

#### **Values**

Value Description

PREDICTIVE The outbound service type is predictive

AUTOMATIC Deprecated

MANUAL Deprecated

PREVIEW The outbound service type is preview

PROGRESSIVE The outbound service type is progressive

# TWSPhoneType

#### Description

Enumeration of phone number/extension types included in a phone book.

### Values

Value Description

//NTERNAL Internal extension (no prefixes added)

EXTERNAL External phone number (service and switch prefixes

will be added to the phone number)

# TWSPredictionType

#### Description

Enumeration of prediction types for an outbound service in predictive mode.

### Values

Value Description

TIME A new predictive call is generated when the

specified time has elapsed after a call is received by

an agent

APPLICATION Deprecated

PERCENTAGE Deprecated

AUTOMATIC The predictive algorithm determines when a new

predictive call must be generated

## **TWSRecordState**

#### Description

Enumeration of the states in which a call recording may be.

#### **Values**

Value Description

NONE The call is not being recorded

STARTED The call is being recorded

PAUSED The call recording is paused

## TWSScriptLanguage

#### Description

Enumeration of the different programming languages that can be used to define the script codes.

#### Values

Value Description

PASCAL The programming language is PascalScript

BASIC The programming language is VBScript

## **TWSServerState**

### Description

Enumeration of the states in which a server of the Presence Suite may be.

#### **Values**

Value Description

SS\_UNLICENSED There are no licenses for this server

SS DOWN The server is down

SS\_UP The server is up

# TWSServerType

## Description

Enumeration of the Presence Suite server types.

#### **Values**

Value Description

MAIN\_SERVER The server is Presence Server

MESSAGING The server is Presence Mail Interactions Server

RECORDING The server is Presence Recording Server

//NTERNET The server is Presence Web Interactions Server

## **TWSServiceState**

## Description

Enumeration of the states in which the services connected to the agent session may be.

#### **Values**

Value Description

SS\_STARTED The service is started and the agent can handle

contacts from this service

SS\_PAUSED The service has been paused to perform

administrative tasks, therefore no new outbound calls are generated for this service. It only applies

to outbound services

SS\_STOPPED The service is stopped and the agent cannot receive

contacts from this service

SS\_EMPTY The service is started but there are no inbound

records to call at the moment, therefore no new outbound calls are generated for this service. It only

applies to outbound services

SS\_HOURS The service is out of calling hours, therefore no new

outbound calls are generated for this service. It only

applies to outbound services

SS\_PHANTOM There are no free phantom extensions to generate a

preview call for this service. It only applies to outbound services configured in preview mode

# TWSServiceType

## Description

Enumeration of possible types for a service.

#### **Values**

Value Description

OUTBOUND Outbound service

///BOUND Inbound service

## TWSSoundType

### Description

Enumeration that indicates when a sound must be played in the agent's PC.

### **Values**

Value Description

DELIVERED The sound must be played when receiving the

contact

ESTABLISHED The sound must be played when answering the

contact call

## TWSTemplateFieldType

### Description

Enumeration of customized field types for mail templates.

#### **Values**

Value Description

TEXT Text field

NUMERIC Numeric field

DATE Date field

## **Events**

# AGM\_ACTION\_CHANGE

#### Description

Received when the availability of an agent action changes.

### **Fields**

Field Data Type Description

ActionMask Integer Bitmask that indicates the availability of the actions.

Each bit corresponds to an Allowed (1)/Not allowed (0) state of an action. Refer to *Examples, Updating User* 

*Interface* for more information.

The *DisableAgentActions* bitmask of the *TWSSessionInfo* type is applied to this field

## AGM\_ACW\_EVENT

## Description

Received when the agent switches to after-call work state.

#### Fields

No fields

# AGM\_ALTERNATIVEPHONES\_CHANGE

### Description

Received when the alternative phone list changes for the current outbound contact. The <u>GetAlternativePhones</u> method can be called in order to retrieve the alternative phone list for the outbound contact.

#### **Fields**

No fields

## AGM\_CALL\_ALERT

#### Description

Received when the agent gets a contact from a service and the phone starts ringing.

#### **Fields**

Field Data Type Description

ShowCallData Boolean Indicates if the Show attached call data option is

checked in the inbound service that the contact

belongs to

## AGM\_CAPTURED\_CHANGE

#### Description

Received when an outbound contact is captured or the capture is removed.

#### **Fields**

Field Data Type Description

Captured Boolean Indicates if the contact is captured

## AGM\_CAPTUREDDATE\_CHANGE

#### Description

Received when the date and time until which the phone number is captured by the agent changes. Only for inbound contacts.

#### **Fields**

Field Data Type Description

Date Time Date and time until which the capture is applicable.

When the set date and time is reached, the phone

number will be released from the capture

# AGM\_COMMENTS\_CHANGE

### Description

Received when the comments for the current outbound contact change.

#### **Fields**

Field Data Type Description

Comments WideString New comments for the outbound contact

## AGM\_CONTACTCALL\_CHANGE

#### Description

Received when the call switches from active to held or vice versa.

#### **Fields**

Field Data Type Description

IsHold Boolean Indicates if the call is held

StateTimeStamp DateTime Date and time when the state was changed

# AGM\_CONTACTDATA1\_CHANGE

#### Description

Received when the first custom data value assigned to the current contact changes.

#### **Fields**

Field Data Type Description

ContactData WideString New value for the contact's first custom data

# AGM\_CONTACTDATA2\_CHANGE

### Description

Received when the second custom data value assigned to the current contact changes.

#### **Fields**

Field Data Type Description

ContactData WideString New value for the contact's second custom data

# AGM\_CONTACTDATA3\_CHANGE

### Description

Received when the third custom data value assigned to the current contact changes.

#### **Fields**

Field Data Type Description

ContactData WideString New value for the contact's third custom data

# AGM\_CONTACTNAME\_CHANGE

#### Description

Received when the contact name of the current outbound contact changes.

#### **Fields**

Field Data Type Description

Name WideString New contact name for the outbound contact

## AGM\_CUSTOMBUTTONS\_CHANGE

#### Description

Received when the custom button list changes. The <u>GetCustomButtons</u> method can be called in order to retrieve the custom buttons that can be used by the agent.

#### **Fields**

No fields

# AGM\_CUSTOMDATA1\_CHANGE

#### Description

Received when the first custom data value of the outbound record associated with the current contact changes. Only for outbound contacts.

#### **Fields**

Field Data Type Description

CustomData WideString New value for the record's first custom data

## AGM\_CUSTOMDATA2\_CHANGE

### Description

Received when the second custom data value of the outbound record associated with the current contact changes. Only for outbound contacts.

#### **Fields**

Field Data Type Description

CustomData WideString New value for the record's second custom data

## AGM\_CUSTOMDATA3\_CHANGE

#### Description

Received when the third custom data value of the outbound record associated with the current contact changes. Only for outbound contacts.

#### **Fields**

Field Data Type Description

CustomData WideString New value for the record's third custom data

## AGM\_END\_CALL\_ALERT

#### Description

Received when the call from a service is hung up while it is still ringing, i.e. before it can be answered by the agent.

### **Fields**

No fields

# AGM\_ENDCODE\_CHANGE

#### Description

Received when the qualification code for the current contact changes.

#### **Fields**

Field Data Type Description

EndCode Integer New qualification code for the current contact

## AGM\_EXTERNAL\_STOP\_SESSION

### Description

Received when an external application requests that the agent work session be stopped.

#### **Fields**

No fields

## AGM\_FATAL\_ERROR

### Description

Received when an error that requires the agent session to be restarted occurs.

#### **Fields**

Field Data Type Description

FatalError Integer Error code that identifies the failure. Refer to Error

*Codes* for a description of the most common error

codes

## AGM\_FINALIZE\_CONTACT

### Description

Received when the current contact is properly finished by the agent.

#### **Fields**

No fields

# AGM\_FINDMAIL\_EVENT

### Description

Received when a request is made to perform a search within the mail history. The <u>GetMailHistoryLog</u> method can be called in order to get the mail history from the filtering fields specified in the event.

### **Fields**

Field	Data Type	Description
MailboxType	Integer	Specifies the mailbox for the search. The different values available are as follows:  0: Inbox 1: Outbox
Mailld	Integer	Mail identifier to be used as a filter to locate a particular e-mail. If the value is set to 0, the search is filtered by sender (incoming mail) or recipient (outgoing mail), time range and state. Otherwise, those filtering fields are ignored
Sender	WideString	Full or partial e-mail address of the sender (incoming mail) or the recipient (outgoing mail) to be used as a filter. Only if the <i>Mailld</i> parameter is set to 0
Status	Integer	Mail status to filter results. The values differ depending on whether the search is performed within the inbox or the outbox.
		Inbox status values are: 0: (All) 1: Unanswered 2: In progress 3: Completed 4: Suspended
		Outbox status values are: 0: (All) 1: Unsent 2: Sent
		Only if the Mailld parameter is set to 0
Days	Integer	Days ago for the search: 1, 2, 7, 15. Only if the <i>Mailld</i> parameter is set to 0. If the value is set to 0, the <i>StartDate</i> and <i>EndDate</i> fields will be used
StartDate	DateTiime	Date that marks the start of the time period within which you wish to perform the search for e-mails. Only

if the *Days* parameter is set to 0 (i.e., if you need to set a period other than the last 1, 2, 7 or 15 days)

**EndDate** 

DateTime

Date that marks the end of the time period within which you wish to perform the search for e-mails. The range between the start date and the end date cannot exceed 60 days. Only if the *Days* parameter is set to 0 (i.e., if you need to set a period other than the last 1, 2, 7 or 15 days)

## AGM\_INBOUND\_CALL

### Description

Received when the agent answers an inbound call.

#### **Fields**

No fields

## AGM INTERNET CALL

#### Description

Received when the agent starts handling a chat, callback, or video session.

#### **Fields**

No fields

# AGM\_INTERNET\_MULTICHAT\_CONNECT

#### Description

Received when the first Internet service configured in the concurrent chat mode is connected to the agent session. Indicates that the list of concurrent chat Internet services must be shown.

#### Fields

No fields

# AGM\_INTERNET\_MULTICHAT\_DISCONNECT

#### Description

Received when the last Internet service configured in the concurrent chat mode is disconnected from the agent session. Indicates that the list of concurrent chat Internet services must be hidden.

#### **Fields**

No fields

# AGM\_INTERNET\_OPENURL

#### Description

Received when the concurrent chat window must be launched.

#### **Fields**

Field Data Type Description

InternetURL String URL that must be opened in the new window

# AGM\_INTERNET\_SERVICE\_CONNECT

## Description

Received when a new Internet service configured in the concurrent chat mode is connected to the agent session.

#### **Fields**

Field Data Type Description

ServiceId Integer Identifier of the concurrent chat Internet service to be

connected to the agent session

ServiceName String Name of the concurrent chat Internet service to be

connected to the agent session

# AGM\_INTERNET\_SERVICE\_DISCONNECT

#### Description

Received when an Internet service configured in the concurrent chat mode is disconnected from the agent session. The <u>GetSessionInfo</u> method can be called in order to retrieve the current list of connected services.

#### **Fields**

Field Data Type Description

ServiceId Integer Identifier of the concurrent chat Internet service to be

disconnected from the agent session

## AGM\_LAUNCH\_URL

### Description

Received when an URL must be opened in the web browser. Used only if the service that the current contact belongs to has URL integration.

#### **Fields**

Field Data Type Description

URL to be opened in the web browser

## AGM\_LINES\_CHANGE

#### Description

Received when the state of a call line changes. The <u>GetLines</u> method can be called in order to retrieve all call line states.

#### **Fields**

No fields

# AGM\_LOGIN\_EVENT

### Description

Received when the agent logs in to the Presence system.

#### Fields

No fields

## AGM\_LOGOUT\_EVENT

#### Description

Received when the agent logs out from the Presence system.

#### **Fields**

No fields

# AGM\_MAILIN\_DECODED

### Description

Received when the current mail contact is decoded and therefore the <u>GetCurrentMailContact</u> method can be called in order to retrieve information about that contact.

#### **Fields**

No fields

# AGM\_MAILINATTOPEN\_EVENT

#### Description

Received when a request is made to open an incoming e-mail attachment. The <u>GetIncomingMailAttachment</u> method can be called in order to get the content of the attachment.

**Fields** 

Field Data Type Description

AttachmentId Integer Identifier of the incoming e-mail attachment to be

opened. The incoming e-mail attachment identifier refers to the position (index) held by the attachment in

the e-mail

## AGM\_MAILINATTSAVE\_EVENT

### Description

Received when a request is made to save an incoming e-mail attachment. The <u>GetIncomingMailAttachment</u> method can be called in order to get the content of the attachment.

#### **Fields**

Field Data Type Description

AttachmentId Integer Identifier of the incoming e-mail attachment to be

saved. The incoming e-mail attachment identifier refers to the position (index) held by the attachment in the e-

mail

Path WideString Full path of the directory where the attachment is to be

saved

# AGM\_MAILOUT\_CHANGE

#### Description

Received when the content (header or body) of the outgoing e-mail changes. The <u>GetOutgoingMail</u> method can be called in order to retrieve the outgoing e-mail data.

#### **Fields**

Field	Data Type	Description
То	WideString	Recipient(s) of the outgoing e-mail
CC	WideString	Carbon copy recipient(s) of the outgoing e-mail
BCC	WideString	Blind carbon copy recipient(s) of the outgoing e-mail
Subject	WideString	Subject of the outgoing e-mail

## AGM\_MAILOUTBCC\_CHANGE

## Description

Received when the blind carbon copy recipient(s) of the outgoing e-mail change.

#### **Fields**

Field Data Type Description

BCC WideString Blind carbon copy recipient(s) of the outgoing e-mail

## AGM\_MAILOUTBODY\_CHANGE

#### Description

Received when the body of the outgoing e-mail changes. The <u>GetOutgoingMail</u> method can be called in order to get the current content of the outgoing e-mail.

#### **Fields**

No fields

# AGM\_MAILOUTCC\_CHANGE

#### Description

Received when the carbon copy recipient(s) of the outgoing e-mail change.

#### **Fields**

Field Data Type Description

CC WideString Carbon copy recipient(s) of the outgoing e-mail

# AGM\_MAILOUTCOMPOSE\_EVENT

### Description

Received when a request is made to generate a new blank outgoing e-mail.

#### **Fields**

No fields

# AGM\_MAILOUTSUBJECT\_CHANGE

#### Description

Received when the subject of the outgoing e-mail changes.

#### **Fields**

Field Data Type Description

Subject WideString Subject of the outgoing e-mail

## AGM\_MAILOUTTO\_CHANGE

#### Description

Received when the recipient(s) of the outgoing e-mail change.

#### **Fields**

Field Data Type Description

To WideString Recipient(s) of the outgoing e-mail

# AGM\_MAXACW\_EVENT

### Description

Received when the maximum time in after-call work set for the service that the current contact belongs to is reached.

#### **Fields**

No fields

# AGM\_MESSAGEALERT\_EVENT

## Description

Received to notify changes in the notices, the recordings or the waiting calls in the services connected to the agent session.

#### **Fields**

Field	Data Type	Description
AlertType	Byte	Type of element that has changed: 0: Notices 1: Recordings 2: Waiting calls. Refer to the GetWaitingCallMessageAlert method to get the queued calls for the service connected to the agent session
NewNotice	Boolean	Indicates if there are new notices. Only if the <i>AlertType</i> field is set to 0
RecordingId	Integer	Identifier of the recording that has changed. Only if the AlertType field is set to 1
RecordType	Byte	Indicates how the recording has been started:  0: By a recording plan  1: By the agent  Only if the <i>AlertType</i> field is set to 1
RecordStatus	Byte	Current state of the recording:  0: There is no recording  1: The recording is in progress  2: The recording is paused  3: The recording has been resumed  4: The recording has been stopped  5: The recording is being stopped  Only if the AlertType field is set to 1

## AGM\_MESSAGESTATE\_EVENT

### Description

Received to notify the action that is currently being performed.

#### **Fields**

Field Data Type Description

0: Starting

- 1: Requesting to stop
- 2: Stopping session
- 3: Stop accepted
- 4: Waiting for user action
- 5: Returning outbound contact
- 6: Returning inbound contact
- 7: Requesting outbound record
- 8: Deprecated
- 9: Recording started
- 10: Recording start failed
- 11: Recording stopped
- 12: Recording paused
- 13: Recording resumed
- 14: Recording in progress
- 99: No action is being performed

## AGM\_MINACW\_EVENT

#### Description

Received when the minimum time in after-call work set for the service that the current contact belongs to is reached.

#### **Fields**

No fields

# AGM\_OUTBOUND\_CALL

### Description

Received when the agent answers an outbound call.

#### **Fields**

Field Data Type Description

ServiceCallerId WideString Phone number that will be displayed at the

destination of calls made from the service. Only if the Enable outgoing calls identification option

is checked in the service

ServiceCallerName WideString Descriptive text that will be displayed at the

destination of calls made from the service. Only if the installed platform is OpenGate and the Enable outgoing calls identification option is checked in

the service

ContactCallerId WideString Phone number that will be displayed at the

destination of calls made to the record. Only if the Enable outgoing calls identification option is

checked in the service.

If this field contains an empty string, the ServiceCallerId field is used for the record

ContactCallerName WideString Descriptive text that will be displayed at the

destination of calls made to the record. Only if the installed platform is OpenGate and the Enable outgoing calls identification option is checked in

the service.

If this field contains an empty string, the ServiceCallerName field is used for the record

## AGM\_PHONE\_CHANGE

#### Description

Received when the phone number of the current outbound contact changes.

#### **Fields**

Field Data Type Description

Phone WideString New phone number for the outbound contact

# AGM\_RECORDING\_PAUSE

#### Description

Received when the recording of the active call is paused.

#### **Fields**

No fields

# AGM\_RECORDING\_RESUME

#### Description

Received when the recording of the active call is resumed.

#### **Fields**

No fields

# AGM\_RECORDING\_START

#### Description

Received when the recording of the active call is started. If the recording cannot be started, the error will be reported in the *RecordResult* field.

#### **Fields**

Field Data Type Description

Recorded Integer Identifier for the recording

RecordResult Integer Indicates if the recording has been successfully started.

When the returned value is <0, the recording could not be started. If the value is 0, the recording has been successfully started. Refer to *Error Codes* for a

description of the most common error codes

## AGM\_RECORDING\_STOP

### Description

Received when the recording of the active call is stopped.

#### **Fields**

No fields

## AGM\_REQUEST\_FINALIZE\_CONTACT

#### Description

Received when the agent is requested to finish the contact (e.g. because the maximum time in after-call work has been reached).

#### **Fields**

No fields

# AGM\_SAVEPARTIALMAIL\_CHANGE

#### Description

Received when the contact qualification option to save the outgoing e-mail if the mail contact is suspended changes. The <u>SaveOutgoingMailAsPartial</u> method can be called in order to save the outgoing e-mail.

#### **Fields**

Field Data Type Description

SavePartialMail Boolean Indicates if the contact qualification option to save the

outgoing e-mail is enabled

## AGM\_SCHEDULEDDATE\_CHANGE

#### Description

Received when the scheduled date for the current outbound contact changes.

#### **Fields**

Field Data Type Description

Date Time New scheduled date for the outbound contact

## AGM\_SERVER\_STATE\_CHANGE

#### Description

Received when the state of a server of the Presence Suite changes. The <u>GetServerStatus</u> method can be called in order to retrieve all server states.

#### **Fields**

No fields

# AGM\_SERVICE\_CHANGE

### Description

Received when the state of a service connected to the agent session changes. The <u>GetSessionInfo</u> method can be called in order to retrieve the current state of the services.

#### **Fields**

No fields

## AGM\_SERVICE\_CONNECT

### Description

Received when a new service is connected to the agent session. If this connection cannot be performed, the error will be reported in the *ErrorCode* field.

#### **Fields**

Field	Data Type	Description
ServiceId	Integer	Identifier of the service to be connected to the agent session
ErrorCode	Integer	Indicates if the connection has been successfully performed. When the returned value is <0, the service could not be connected. If the value is 0, the service has been successfully connected. Refer to <i>Error Codes</i> for a description of the most common error codes

# AGM\_SERVICE\_DISCONNECT

#### Description

Received when a service is disconnected from the agent session. The <u>GetSessionInfo</u> method can be called in order to retrieve the current list of connected services.

### **Fields**

No fields

## AGM\_SKILLS\_CHANGE

#### Description

Received when the list of skills controlled by the agent login changes. The <u>GetManagedSkills</u> method can be called in order to retrieve the list of skills under agent control. Only for the OpenGate platform.

#### **Fields**

No fields

## AGM\_SOUND\_EVENT

### Description

Received when a sound must be played in the agent's PC.

#### **Fields**

Field Data Type Description

SoundType Byte Type of sound that must be played:

0: No sound must be played. If a sound is being played

in a loop, it should be stopped

1: System sound (refer to the GetSessionCallSound

method)

2: Service sound when receiving a call (refer to the

<u>GetServiceCallSound</u> method)

3: Service sound when answering a call (refer to the

GetServiceCallSound method)

# AGM\_STATE\_CHANGE

### Description

Received when the agent state changes.

#### **Fields**

Field Data Type Description

NewState Byte Current agent state:

0: Stopped1: Started

2: Servicing contact

3: After-call work

4: Available

5: Ringing

TimeStamp DateTime Date and time when the state was changed

## AGM\_STOP\_SESSION

### Description

Received when the agent's request to stop the work session is accepted.

#### **Fields**

No fields

# AGM\_TRANSFER\_COMPLETED

### Description

Received when a call transfer process initiated by the agent is completed. Only if a consultation call is made.

#### **Fields**

No fields

## AGM\_TRANSFER\_RECEIVED

### Description

Received when the agent is handling a service call that is transferred from another agent and the transfer process is completed.

#### **Fields**

No fields

# **Types**

# **TWSAgentEvent**

#### Description

Information about an agent event.

#### **Fields**

Field Data Type Description

EventCode TWSEventCode Event code

EventFields <u>TWSAgentEventFieldList</u> Event field list

# TWSAgentEventField

#### Description

Information about an event field and its value. Refer to *Events* to see the list of fields reported by each event.

#### **Fields**

Field Data Type Description

Field WideString Field name

Value\_ Variant Field value. The data type depends on the field

content (refer to *Events* to see the list of fields and

their data types)

# TWSAgentServiceInformation

## Description

Information about an agent that is working in a specific service.

#### **Fields**

Field Data Type Description

AgentLogin Int64 Agent login

AgentName	WideString	Agent name
Station	Int64	Station number where the agent is connected
IpAddress	WideString	IP address of the machine where the agent is working
ConnectionTime	Integer	Time that the agent has been connected to the service
WorkingTime	Integer	Time that the agent has been working in the service
Status	Integer	Current state of the agent
StatusTime	Integer	Time that the agent has been in the current state

# TWSAlternativePhoneDescription

## Description

Information about an alternative phone number description.

#### **Fields**

Field	Data Type	Description
Code	Integer	Description code
Description	WideString	Alternative phone number description

## **TWSCall**

## Description

Information about a call.

## Fields

Field	Data Type	Description
CallID	Integer	Call identifier

CallingDevice WideString Device that made the call

CalledDevice WideString Device that was dialed to make the call

Phone Description Wide String Additional information about the call, usually

the customer's phone number. If the calling device is a phantom extension, a description of the contact type will be shown, e.g. *Preview call* 

CallState TWSCallState Call state

Selected Boolean Indicates if the call is active

InConference Boolean Indicates if the call is in a conference

Start TimeStamp Date Time Date and time when the call was started

StateTimeStamp DateTime Date and time when the call switched to the

current state

RecordStatus <u>TWSRecordState</u> State of the call recording

## **TWSCallDataInformation**

#### Description

Information about a variable attached to a contact.

#### **Fields**

Field Data Type Description

Key WideString Name/Key of the variable

Value\_ WideString Variable value

## **TWSContact**

#### Description

General information about a contact. For specific contact type information, refer to <u>TWSInboundContact</u>, <u>TWSInternetContact</u>, <u>TWSMailContact</u> or <u>TWSOutboundContact</u>.

### **Fields**

Field Data Type Description

ServiceId Integer Identifier of the service that the contact belongs

to

ContactType <u>TWSContactType</u> Contact type

CallState	<u>TWSCallState</u>	Current state of the call associated with the contact
CallId	Integer	Identifier of the call associated with the contact
UCID	WideString	Universal Call Identifier of the call associated with the contact
Skill	Int64	Extension number of the skill in which the call has been queued
SkillDesc	WideString	Name of the skill in which the call has been queued
VDN	Int64	VDN (Avaya platform) or service extension (OpenGate platform) used by the ACD system for routing the call
VDNDesc	WideString	Name of the VDN (Avaya platform) or service extension (OpenGate platform) used by the ACD system for routing the call
ContactId	Integer	Contact identifier
Phone	WideString	Contact phone number
Captured	Boolean	Indicates if the contact is captured
CapturedLogin	Int64	Login that has captured the contact. Only if the Captured field is set to TRUE
IsManualACDCall	Boolean	Indicates if the current contact is a manual outbound ACD call. Only if the Enable support for manual outbound ACD calls option is checked in the outbound service
EnableBCC	Boolean	Indicates if sending e-mails to blind carbon copy recipients (BCC) is allowed for the mailbox that the contact belongs to. Only if the contact is a mail contact

# TWSContactData

## Description

Information about contact custom data.

## Fields

Field Data Type Description

ShowContactData1 TWSContactDataVisibility Access level of agents to the first contact custom data value ShowContactData2 TWSContactDataVisibility Access level of agents to the second contact custom data value ShowContactData3 TWSContactDataVisibility Access level of agents to the third contact custom data value ContactDataDesc1 WideString Description of the first contact custom

data value

ContactDataDesc2 WideString Description of the second contact custom

data value

Description of the third contact custom ContactDataDesc3 WideString

data value

## **TWSCustomButton**

#### Description

Information about a custom button. These buttons allow you to execute custom script code.

### **Fields**

Field	Data Type	Description
Id	WideString	Identifier of the custom button
GroupId	Integer	Identifier of the group which the custom button belongs to
ServiceId	Integer	Identifier of the service in which the custom button could be used
Description	WideString	Custom button description
Icon	Binary	Icon that is used for the custom button
AlwaysEnabled	Boolean	Indicates if the custom button will remain enabled at all times or only when handling a service call

## **TWSCustomData**

## Description

Information about record custom data. Only for outbound services.

#### **Fields**

Field	Data Type	Description
ShowCustomData1	TWSCustomDataVisibility	Access level of agents to the first record custom data value
ShowCustomData2	TWSCustomDataVisibility	Access level of agents to the second record custom data value
ShowCustomData3	TWSCustomDataVisibility	Access level of agents to the third record custom data value
CustomDataDesc1	WideString	Description of the first record custom data value
CustomDataDesc2	WideString	Description of the second record custom data value
CustomDataDesc3	WideString	Description of the third record custom data value

# TWSImage

## Description

Information about an image.

#### Fields

Field	Data Type	Description
Name	WideString	Image name
Data	Binary	Image content

# TWSInboundContact

## Description

Information about an inbound contact. It inherits from <u>TWSContact</u> including all its fields.

#### **Fields**

Field	Data Type	Description
DNIS	WideString	Dialed Number Identification Service. This is the phone number that the inbound contact dialed
CollectDigits	WideString	DTMF digits collected while the call is handled by the switch
CollectVDN	Int64	VDN (Avaya platform) or service extension (OpenGate platform) where the <i>CollectDigits</i> field value was collected
Surveys	<u>TWSSurveyList</u>	List of surveys that the contact can be transferred to
ContactData1	WideString	First custom data value assigned to the inbound contact
ContactData2	WideString	Second custom data value assigned to the inbound contact
ContactData3	WideString	Third custom data value assigned to the inbound contact

# **TWSInboundService**

## Description

Information about an inbound service connected to the agent session. It inherits from *TWSService* including all its fields.

### Fields

Field	Data Type	Description
AutomaticCaptureEnabled	Boolean	Indicates if the contact phone number is automatically captured for the agent that is handling the contact. Only if the <i>CaptureEnabled</i> field is set to TRUE
EnableContactData	Boolean	Indicates if contact custom data are enabled
ContactData	<u>TWSContactData</u>	Description of contact custom data and access level of agents to them

# TWSInternetContact

## Description

Information about an Internet (chat/callback/video) contact. It inherits from <u>TWSContact</u> including all its fields.

### **Fields**

Field	Data Type	Description
InternetSessionType	<u>TWSInternetSessionType</u>	Internet session type
CallBackPhone	WideString	Phone number provided by the customer for the callback session. Only if the <i>InternetSessionType</i> field is set to <i>CALLBACK</i>
ExecuteURL	WideString	URL to open the agent chat window

## **TWSMail**

## Description

Information about an e-mail or a mail template.

### **Fields**

Field	Data Type	Description
1d	Integer	E-mail or template identifier
MailboxId	Integer	Identifier of the mailbox that the e-mail or template belongs to
Subject	WideString	E-mail or template subject
From	WideString	E-mail or template sender
MailTo	WideString	E-mail or template recipient(s)
Reply	WideString	E-mail address to send the reply to
CC	WideString	E-mail or template carbon copy recipient(s)
BCC	WideString	E-mail or template blind carbon copy recipient(s)
Date	WideString	Date in text format as displayed in the e- mail or template

HTMLBody WideString E-mail or template body in HTML format

PlainTextBody WideString E-mail or template body in plain text format

Attachments <u>TWSMailAttachmentList</u> List of e-mail or template attachments

InplaceImages <u>TWSImageList</u> List of images embedded in the e-mail or

template body

# **TWSMailAddress**

#### Description

Information about an e-mail address.

#### **Fields**

Field	Data Type	Description
Id	Integer	E-mail address identifier
Name	WideString	Name of the e-mail address holder
Organization	WideString	Organization of the e-mail address holder
Title	WideString	Position of the e-mail address holder
WorkPhone	WideString	Work phone number of the e-mail address holder
HomePhone	WideString	Home phone number of the e-mail address holder
CellPhone	WideString	Cell phone number of the e-mail address holder
Fax	WideString	Fax number of the e-mail address holder
MailAddress	WideString	E-mail address
PostalAddress	WideString	Home address of the e-mail address holder

## **TWSMailAddressBook**

## Description

Information about an address book.

### **Fields**

Field	Data Type	Description
Id	Integer	Identifier of the address book
Name	WideString	Name of the address book
Addresses	TWSMailAddressList	List of e-mail addresses included in the address book

Groups <u>TWSMailGroupList</u> List of address groups included in the address

book

# TWSMailAddressGroup

### Description

Information about the e-mail addresses of an address group. Both addresses added from the address book and addresses specific to the address group are included.

#### **Fields**

Field Data Type Description

Name WideString Name of the e-mail address holder

MailAddress WideString E-mail address

## **TWSMailAttachment**

### Description

Information about an e-mail or template attachment.

#### **Fields**

Field Data Type Description

Integer Identifier of the e-mail or template that the attachment

belongs to

AttachmentId Integer Identifier of the attachment for the Presence system

FileName WideString Name of the attachment

FileSize WideString Size of the attachment

# TWSMailboxToMove

### Description

Basic information about a mailbox that the incoming mail can be moved to.

### Fields

Field	Data Type	Description
Id	Integer	Identifier of the mailbox that the incoming mail can be moved to
Name	WideString	Mailbox name
Enabled	Boolean	Indicates if the mailbox is enabled (TRUE) or disabled (FALSE)

# **TWSMailContact**

### Description

Information about a mail contact. It inherits from <u>TWSContact</u> including all its fields.

### **Fields**

Field	Data Type	Description
Mailld	Integer	Mail identifier
Mailboxld	Integer	Identifier of the mailbox that the e-mail belongs to
MailboxName	WideString	Name of the mailbox that the e-mail belongs to
MailboxAddress	WideString	E-mail address associated to the mailbox POP3 account. It matches the address that the users or clients send the e-mails to
IncomingMail	<u>TWSMail</u>	Information about the incoming e-mail received from the contact that is being handled
HasPartialMail	Boolean	Indicates if the incoming e-mail is associated to a partial reply

PartialMail	<u>TWSMail</u>	Information about the partial reply. Only if the HasPartialMail field is set to TRUE
MailboxCaptureEnabled	Boolean	Indicates if the mail capturing of incoming e- mails by an agent is enabled for the mailbox that the e-mail belongs to
CapturedDateTime	DateTime	Date and time until which the e-mail address is captured for the agent. When the set date and time is reached, the e-mail address will be released from the capture. Only if the <i>Captured</i> (refer to <i>TWSContact</i> ) and <i>MailboxCaptureEnabled</i> fields are set to TRUE
AutomaticCapture	Boolean	Indicates if the contact e-mail address is automatically captured for the agent that is handling the contact. Only if the <i>MailboxCaptureEnabled</i> field is set to TRUE
MailboxSplitCompose	Boolean	Indicates if the incoming e-mail should be displayed separately from the reply (TRUE) or in the same window (FALSE)

# TWSMailGroup

## Description

Information about an address group.

## Fields

Field	Data Type	Description
Id	Integer	Address group identifier
Name	WideString	Address group name
Addresses	TWSMailAddressGroupList	List of e-mail addresses included in the address group

# **TWSMailHistory**

## Description

Information about an e-mail retrieved through the <u>GetMailHistoryLog</u> method.

### Fields

Field	Data Type	Description
MailboxId	Integer	Identifier of the mailbox that the e-mail belongs to
Mailld	Integer	Mail identifier
MailHistoryId	Integer	Identifier assigned to the e-mail within the retrieved mail history
Status	<u>TWSMailStatus</u>	E-mail current status
Subject	WideString	E-mail subject
Date	DateTime	Date and time when Presence Mail Interactions Server downloaded the incoming e-mail or sent the outgoing e-mail
MailboxName	WideString	Name of the mailbox that the e-mail belongs to
MailboxAddress	WideString	E-mail address associated to the mailbox POP3 account. It matches the address that the users or clients send the e-mails to
EnableBCC	Boolean	Indicates if sending e-mails to blind carbon copy recipients (BCC) is allowed for the mailbox that the e-mail belongs to

# **TWSMailIdError**

## Description

Information about an e-mail that has generated an error when processed in a request.

## Fields

Field	Data Type	Description
Mailld	Integer	Identifier of the e-mail that generated the error
ErrorCode	Integer	Error code that identifies the failure. Refer to <u>Error</u> <u>Codes</u> for a description of the most common error codes

# TWSMailSuspended

### Description

Basic information about an incoming e-mail that has been suspended by an agent.

#### **Fields**

Field	Data Type	Description
-------	-----------	-------------

MailId Integer Mail identifier

Date Integer Date and time when Presence Mail Interactions Server

downloaded the e-mail

Subject WideString E-mail subject

From WideString E-mail sender

## **TWSMultiChatService**

### Description

General information about a concurrent chat Internet service that is connected to the agent session.

### **Fields**

Field	Data Type	Description
-------	-----------	-------------

Integer Identifier of the concurrent chat

Internet service

Name WideString Name of the concurrent chat

Internet service

## **TWSNoticeHeader**

#### Description

Information about a notice. The notice body is not included.

### Fields

Field Data Type Description

NoticeId Integer Notice identifier

NoticeDate DateTime Date when the notice was created

Subject	WideString	Subject of the notice

User WideString Identifier of the user that created the notice

IsRead Boolean Indicates if the notice has been read by the agent

# TWSOutboundContact

## Description

Information about an outbound contact. It inherits from <u>TWSContact</u> including all its fields.

### **Fields**

Field	Data Type	Description
ID	Integer	Outbound record identifier
Name	WideString	Contact name for the outbound record
SourceID	Integer	This field provides a reference to the ID assigned to the customer in the system integrated with the Presence Suite
LastAgent	Int64	Login of the last agent who handled the outbound record (0 if never handled)
LastQCodeId	Integer	Last qualification code for the outbound record (0 if never handled)
LastQCodeDescription	WideString	Description of the last qualification code for the outbound record (blank if never handled)
CapturingAgent	Int64	If the field is 0, the record has not been captured by any agent. Otherwise, it's the login of the agent who currently keeps the outbound record captured
Comments	WideString	Comments for the outbound record
<i>TimeZoneName</i>	WideString	Description of the time zone that the outbound record belongs to. Only if the Enable support for time zones option is checked in the service
TimeZoneId	WideString	Identifier of the time zone that the outbound record belongs to. Only if the Enable support for time zones option is checked in the service

LastHandlingDate	DateTime	Date and time of the last time that the outbound record was handled, regardless of whether it was handled by an agent or by the system
DefaultScheduledDateTime	DateTime	Default date and time to schedule the outbound record, which is calculated taking into account the values defined in the Default schedule option of the service settings
CallerId	WideString	Phone number that will be displayed at the destination of calls made to the record. Only if the Enable outgoing calls identification option is checked in the service.  If this field contains an empty string, the service outgoing calls identification data are used for the record
CallerName	WideString	Descriptive text that will be displayed at the destination of calls made to the record. Only if the installed platform is OpenGate and the Enable outgoing calls identification option is checked in the service. If this field contains an empty string, the service outgoing calls identification data are used for the record
Surveys	<u>TWSSurveyList</u>	List of surveys that the contact can be transferred to
CustomData1	WideString	First custom data value of the outbound record associated with the contact
CustomData2	WideString	Second custom data value of the outbound record associated with the contact
CustomData3	WideString	Third custom data value of the outbound record associated with the contact
ContactData1	WideString	First custom data value assigned to the outbound contact
ContactData2	WideString	Second custom data value assigned to the outbound contact
ContactData3	WideString	Third custom data value assigned to the outbound contact

# TWSOutboundContactHistory

## Description

Information about the history of contacts previously made to the outbound record currently being handled.

### **Fields**

Field	Data Type	Description
Id	Integer	Outbound contact identifier
RDate	DateTime	Date when the outbound contact was handled. This is the date when the contact handling was finished
ServiceId	Integer	Identifier for the outbound service that the outbound contact belongs to
SourceId	Integer	This field provides a reference to the ID assigned to the customer in the system integrated with the Presence Suite
<i>QCode</i>	Integer	Qualification code that was used to qualify the outbound contact
QCodeDescription	WideString	Description of the qualification code that was used to qualify the outbound contact
ScheduledDateTime	DateTime	Date and time that the outbound record was scheduled to when the contact was handled
AgentLogin	Int64	Login that handled the outbound contact
Phone	WideString	Phone number of the outbound record that was called when the contact was handled
LoginName	WideString	Name of the agent that handled the outbound contact

# **TWSOutboundPhone**

#### Description

Information about an alternative outbound phone number.

### **Fields**

Field Data Type Description

Number	WideString	Alternative phone number
Description	Integer	Description code of the alternative phone number
Status	Integer	State of the alternative phone number: 0: Valid phone number 1: Do-Not-Call phone number (System) 2: Phone number does not exist (System) 51: Do-Not-Call phone number (Agent) 52: Phone number does not exist (Agent) 53: Wrong phone number (Agent)
TimeZoneId	WideString	Identifier for the time zone of the alternative phone number. Only if the Enable support for time zones option is checked in the service

# **TWSOutboundService**

## Description

Information about an outbound service connected to the agent session. It inherits from <u>TWSService</u> including all its fields.

## Fields

Field	Data Type	Description
Outbound Type	<u>TWSOutboundType</u>	Outbound type of the service. Please note that this field indicates the outbound mode in which the service is running at the moment. Thus, if the service type is predictive, in certain circumstances this field may return the progressive type, for example, if the abandonment control is enabled and the set limit is exceeded, or in particular cases related to the operation of the automatic predictive type (for further information, refer to the <i>Outbound services</i> , <i>Configuration, ACD</i> section of the <i>Presence Administrator Manual</i> )
PredictionType	TWSPredictionType	Prediction type. Only if the OutboundType field is set to PREDICTIVE

CallHoursEnabled	Boolean	Indicates if the option to allow the agent to detail specific calling hours for an outbound record is enabled
CallHours	WideString	Calling hours set for the outbound service. Calls can be made exclusively within this timeframe

EnableManualACD

Boolean

Indicates if the support for manual outbound ACD calls is enabled. Only if the

OutboundType field is set to PREVIEW and the

CaptureEnabled field is set to

TRUE

EnableAgentScheduleManualACD Boolean Indicates if the option to allow the agent to schedule manual outbound ACD calls is enabled.

Only if the OutboundType field is

set to *PREVIEW*, and the *CaptureEnabled* and

EnableManualACD fields are set

to TRUE

EnableAgentEditAltPhones Boolean Indicates if the option to allow

the agent to edit the scheduled

phone number, and the

alternative phone numbers with its descriptions is enabled. Only if the *AlternativePhonesEnabled* 

is set to TRUE

EnableAgentEditScheduledPhone Boolean Indicates if the option to allow

the agent to edit the phone number and time zone (only if the Enable support for time zones option is checked) when scheduling an outbound record is

enabled

AlternativePhonesEnabled Boolean Indicates if the alternative phone

numbers option to allow the use

of several contact phone numbers for each outbound

record is enabled

AlternativePhonesChangeState Boolean Indicates if the option to allow

the agent to change the state of each of the available phone

numbers is enabled

TimeZoneCheckNPATZ Boolean Indicates if the option to check

the time zones (using the area codes) input by the agent is

ena	hl	ed	$\bigcirc$ r	ıΙν	if	the	
cria	$\sim$ 1	cu.	$\sim$ 1	11 V		uic	

TimeZoneEnabled field is set to

TRUE

TimeZoneEnabled Boolean Indicates if the support for time

zones is enabled

TimeZoneId WideString Identifier of the time zone to be

used as default for the service.
Only if the *TimeZoneEnabled* 

field is set to TRUE

EnableCallerId Boolean Indicates if the support for

outgoing calls identification is

enabled

EnableCustomData Boolean Indicates if record custom data

are enabled

data and access level of agents

to them

EnableContactData Boolean Indicates if contact custom data

are enabled

data and access level of agents

to them

# TWSOutgoingMail

### Description

Information about an outgoing e-mail (reply).

### **Fields**

Field	Data Type	Description
Mailld	Integer	Mail identifier
MailboxId	Integer	Identifier of the mailbox that the e-mail belongs to
Subject	WideString	E-mail subject
From	WideString	E-mail sender
MailTo	WideString	E-mail recipient(s)
Reply	WideString	E-mail address that the recipient should send the reply to

CC	WideString	E-mail carbon copy recipient(s)
BCC	WideString	E-mail blind carbon copy recipient(s)

Date WideString Date in text format

HTMLBody WideString E-mail body in HTML format

PlainTextBody WideString E-mail body in plain text format

InplaceImages <u>TWSImageList</u> List of images embedded in the e-mail

## **TWSPredefinedText**

### Description

Body of a predefined text.

#### **Fields**

Field	Data Type	Description
-------	-----------	-------------

HTMLBody WideString Predefined text body in HTML format

## TWSPredefinedTextTreeItem

#### Description

Information about a predefined text group or a predefined text.

### **Fields**

Field	Data Type	Description

Integer Identifier of the group or predefined text

Name WideString Name of the group or predefined text

ParentId Integer Identifier of the group if the element belongs to a

group. If the field value is 0, it does not belong to any

group

Indicates if the element is a group (TRUE) or a

predefined text (FALSE)

# **TWSQCode**

## Description

Information about a qualification code group or a qualification code.

### Fields

Field	Data Type	Description
1d	Integer	Identifier of the group or qualification code. Only if the <i>IsGroup</i> field is set to TRUE or the <i>Code</i> field is a service qualification code (equal to or higher than 100)
Code	Integer	Qualification code
Description	WideString	Description of the group or qualification code
IsGroup	Boolean	Indicates if the element is a group (TRUE) or a qualification code (FALSE)
ParentId	Integer	Identifier of the group if the element belongs to a group. If the field value is 0, it does not belong to any group. Only if the <i>IsGroup</i> field is set to TRUE or the <i>Code</i> field is a service qualification code (equal to or higher than 100)

# **TWSQInfo**

Deprecated. Refer to <u>TWSOInfo3</u>.

# TWSQInfo2

Deprecated. Refer to <u>TWSOInfo3</u>.

# TWSQInfo3

## Description

Information about the qualification of the current contact.

## Fields

Field	Data Type	Description
<i>OCode</i>	Integer	Qualification code, which may be specific for a service (equal to or higher than 100) or a common qualification code (lower than 100). For a list of common qualification codes, refer to <i>Common service options, Qualification codes</i> in the <i>Presence Administrator Manual</i>
сТуре	<u>TWSContactType</u>	Contact type determined by the type of service that the contact belongs to
InfoInbound	<u>TWSQInfoInbound</u>	Information about the qualification of the inbound contact
InfoOutbound	TWSQInfoOutbound	Information about the qualification of the outbound contact
InfoMail	<u>TWSQInfoMail</u>	Information about the qualification of the mail contact
CustomData1	WideString	First custom data value of the outbound record associated with the contact. Only for outbound contacts
CustomData2	WideString	Second custom data value of the outbound record associated with the contact. Only for outbound contacts
CustomData3	WideString	Third custom data value of the outbound record associated with the contact. Only for outbound contacts
ContactData1	WideString	First custom data value assigned to the outbound or inbound contact
ContactData2	WideString	Second custom data value assigned to the outbound or inbound contact
ContactData3	WideString	Third custom data value assigned to the outbound or inbound contact

# TWSQInfoInbound

## Description

Information about the qualification of an inbound contact.

### Fields

Field	Data Type	Description
ContactCaptured	Boolean	Indicates if the agent has captured the phone number for the inbound contact. Only if the Enable call capturing option is checked in the service
CapturedDateTime	DateTime	Date and time until which the capture is applicable. When the set date and time is reached, the phone number will be released from the capture
DNCPhone	WideString	Do-not-call phone number. Only if the Allow agent to qualify Do-Not-Call phone numbers option is checked and the qualification code for the contact is no. 11 (Do not call again)

# **TWSQInfoMail**

## Description

Information about the qualification of a mail contact.

## Fields

Field	Data Type	Description
MailCaptured	Boolean	Indicates if the agent has captured the e-mail address for the mail contact. Only if the Enable mail capturing option is checked in the mailbox
<i>MailCaptureDateTime</i>	DateTime	Date and time until which the capture is applicable. When the set date and time is reached, the e-mail address will be released from the capture. Only if the Enable mail capturing option is checked in the mailbox, the Automatically capture mails for X minutes option is unchecked, and the value of the <i>MailCaptured</i> field is set to TRUE
MoveToMailboxId	Integer	Identifier of the mailbox to move the incoming e- mail to. Only if the qualification code for the

contact is 29 (Moved to another mailbox). If the value is 0, the incoming e-mail will not be moved. Refer to the <u>GetMailboxesToMove</u> method to retrieve the mailboxes that the incoming e-mail can be moved to

## **TWSQInfoOutbound**

### Description

Information about the qualification of an outbound contact.

#### **Fields**

Field	Data Type	Description
Phone	WideString	Contact phone number for the outbound record
Name	WideString	Contact name for the outbound record
Comments	WideString	Comments for the outbound record
ScheduleDate	DateTime	Date and time that the outbound record has been scheduled to
Captured	Boolean	Indicates if the outbound record is captured. Only if the Enable call capturing option is checked in the service
TimeZoneId	WideString	Identifier of the time zone that the outbound record belongs to. Only if the Enable support for time zones option is checked in the service
CallingHours	WideString	Specific calling hours for the outbound record. Only if the Enable customer calling hours option is checked in the service

# **TWSReason**

## Description

Information about a logout/stop reason.

### **Fields**

Field Data Type Description

Code	Integer	Logout/stop reason code
Description	WideString	Logout/stop reason description
CMSCode	Integer	Code that will be stored in the Avaya CMS when an agent selects the logout/stop reason. Only for Avaya switches

## **TWSServer**

## Description

Information about the state of a server of the Presence Suite.

### Fields

Field	Data Type	Description
Name	WideString	Product server name, i.e. <i>Presence Server</i> , <i>Presence Mail Interactions Server</i> , <i>Presence Web</i> <i>Interactions Server</i> or <i>Presence Recording Server</i>
sType	<u>TWSServerType</u>	Presence Suite server type
State	<u>TWSServerState</u>	Server state
<i>Ip</i>	WideString	IP address of the machine where the server is running
Port	Integer	Port used by the client applications to communicate with the server

# **TWSServerList**

## Description

Information about the state of Presence Mail Interactions Server, Presence Web Interactions Server and Presence Recording Server.

### Fields

Field	Data Type	Description
Messaging	<u>TWSServer</u>	State of Presence Mail Interactions Server
Internet	<u>TWSServer</u>	State of Presence Web Interactions Server

Recording Server State of Presence Recording Server

# **TWSService**

## Description

General information about a service connected to the agent session. For specific service type information, refer to <u>TWSInboundService</u> or <u>TWSOutboundService</u>.

### **Fields**

Field	Data Type	Description
ServiceId	Integer	Identifier of the service
Name	WideString	Name of the service
State	TWSServiceState	State of the service
ServiceType	<u>TWSServiceType</u>	Indicates if the service is an inbound service or an outbound service
MinACWTime	Integer	Minimum compulsory amount of time (in seconds) an agent has to spend in after-call work
MaxACWTime	Integer	Maximum time (in seconds) an agent can spend in after-call work
MaxACWQCode	Integer	Qualification code used to qualify a contact which has been finished automatically when the maximum time set for after-call work has been reached
UseMaxACWQCode	Boolean	Indicates if the value selected as qualification code in the <i>MaxACWQCode</i> field must be used only in case that the agent did not qualify the contact before the maximum time set for after-call work was reached
HasLogo	Boolean	Indicates if the service has a logo
CaptureEnabled	Boolean	Indicates if the call capturing feature is enabled for the service
IntegrationType	<u>TWSIntegrationType</u>	Type of service integration with a third-party application

EstablishedURL	WideString	URL that must be opened in the web browser when the agent receives a contact. Only if the <i>IntegrationType</i> field is set to <i>URL</i>
ClosedURL	WideString	URL that must be opened in the web browser when the agent closes a contact. Only if the <i>IntegrationType</i> field is set to <i>URL</i>
SendMailEnabled	Boolean	Indicates if the option to send e- mails using a mailbox when handling a contact is enabled
PlaySoundOnDeliveredCall	Boolean	Indicates if a warning sound must be played in the agent's PC when receiving a service contact
PlaySoundOnEstablishedCa.	// Boolean	Indicates if a warning sound must be played in the agent's PC when answering a service contact
ScriptId	Integer	Code of the script integrated with the service. Only if the <i>IntegrationType</i> field is set to <i>SCRIPT</i>
ShowContactInfo	Integer	Indicates if the contact information window will be shown in Presence Agent: 0: Do not show 1: Show 2: Show on top
EnableCompleteMail	Boolean	Indicates if the option to allow the agent to complete unanswered emails from the mail history is enabled. Only if the <i>EnableMailHistory</i> field is set to TRUE
EnableReplyMail	Boolean	Indicates if the option to allow the agent to reply/forward e-mails from the mail history is enabled. Only if the <i>EnableMailHistory</i> field is set to TRUE
EnableMailHistory	Boolean	Indicates if the option to allow the search of the mail history for any type of contact is enabled
EnableManualACD	Boolean	Indicates if the support for manual outbound ACD calls is enabled
EnableInsertOutboundReco.	rd'Boolean	Indicates that the service allows the insertion of outbound records. Refer to <a href="mailto:lnsertOutboundRecord2">lnsertOutboundRecord2</a>

## **TWSSessionInfo**

### Description

Information about the agent session.

#### **Fields**

Enghouse Interactive

Field Data Type Description

AgentName WideString Name of the agent

ServerName WideString Name of the Presence Server

StationNumber Int64 Station number where the

agent is connected

the agent session

PlaySoundOnDeliveredCall Boolean Indicates if a warning sound

must be played in the agent's

PC when receiving a call

LoopSoundOnDeliveredCall Boolean Indicates if it is necessary to

repeat the warning sound until the agent answers the

call

MultiChatServiceInfoList TWSMultiChatServiceInfoList List of concurrent chat

Internet services that are connected to the agent

session

DisableAgentActions Int64 Bitmask that indicates which

actions are available in the agent toolbar. Each bit corresponds to a Disabled (1)/Enabled (0) state of an

action.

It is applied to the ActionMask field of the

AGM ACTION CHANGE event

EnableManualACD Boolean Indicates if the agent has the

option to make manual outbound ACD calls enabled. Only if support for manual outbound ACD calls has been enabled in a service that the

Presence Suite

agent is working in

## **TWSSkill**

### Description

Information about a skill.

#### **Fields**

Field Data Type Description

Extension Int64 Skill identifier

Name WideString Skill name

Status <u>TWSCT/Ski//Status</u> Skill status

## **TWSSkillActionResult**

#### Description

Information about the result of an action (pause/resume) performed on a skill.

### **Fields**

Field Data Type Description

Skill Int64 Skill identifier

Result Integer Indicates if the action has been successfully

performed. When the returned value is <0, the action could not be performed. If the value is 0, the action has been successfully performed.

## **TWSSoftPhoneInfo**

### Description

Information about the entries of all phone books assigned to the agent and the services that are available for placing direct calls to agents.

### **Fields**

Field Data Type Description

TransferPhoneBookList <u>TWSTransferPhoneList</u> Entries of all phone books assigned

to the agent

TransferServiceBookList <u>TWSTransferServiceList</u> Services that are available for

placing direct calls to agents

# **TWSSurvey**

### Description

Information about a survey.

### **Fields**

Field Data Type Description

Name WideString Survey name

VDN Int64 Survey VDN/service extension to transfer the contacts

to

# **TWSTemplateField**

### Description

Information about a custom field for a mail template.

#### **Fields**

Field Data Type Description

Name WideString Field name

FieldType <u>TWSTemplateFieldType</u> Field type

DefaultValue WideString Default value for the field

Required Boolean Indicates if it is compulsory to specify a

value for this field when using the template

StartTag WideString HTML text to indicate where the field begins

in the mail template content

EndTag WideString HTML text to indicate where the field ends in the mail template content

# TWSTemplateTreeItem

## Description

Information about a template group or a template.

#### Fields

Field	Data Type	Description
Id	Integer	Identifier of the group or template
Name	WideString	Name of the group or template
ParentId	Integer	Identifier of the group if the element belongs to a group. If the field value is 0, it does not belong to any group
IsGroup	Boolean	Indicates if the element is a group (TRUE) or a template (FALSE)

Description

# **TWSTimeZone**

## Description

Information about a time zone defined in the Presence system.

Data Type

### **Fields**

Field

	zata i jpo	2 000. ipt.o
Id	WideString	Time zone identifier
Name	WideString	Time zone description

# TWSTransferPhoneInfo

## Description

Information about a phone number or extension included in the phone books.

### **Fields**

Field Data Type Description

Phone WideString Phone number or extension

PhoneType <u>TWSPhoneType</u> Indicates if the specified phone number is an

external phone number or an internal

extension

Description WideString Description of the phone number or extension

Phonebook Group Identifier of the phone book that the phone

number or extension belongs to

## **TWSTransferServiceInfo**

### Description

Information about a service that is available for placing direct calls to agents.

#### **Fields**

Field Data Type Description

Integer Service identifier

Name WideString Name of the service

# TWSWaitingCallAlert

### Description

Information about the calls in queue (waiting) for a service.

#### **Fields**

Field Data Type Description

ServiceId Integer Identifier of the service

ServiceName WideString Name of the service

InQueueCalls Integer Number of queued calls in the service

MaxTime Integer Amount of time (in seconds) that the oldest call has

been

waiting in the service queue

# Lists

# TWSAgentEventFieldList

Description

List of objects of type TWSAgentEventField.

# TWSAgentEventList

Description

List of objects of type TWSAgentEvent.

# TWSAgentServiceInformationList

Description

List of objects of type **TWSAgentServiceInformation**.

# TWSAlternativePhoneDescriptionList

Description

List of objects of type <u>TWSAlternativePhoneDescription</u>.

# TWSAttachmentIdList

## Description

List of objects of type Integer to store an attachment identifier.

# **TWSCallDataInformationList**

## Description

List of objects of type TWSCallDataInformation.

# TWSCustomButtonList

### Description

List of objects of type **TWSCustomButton**.

# **TWSI**mageList

### Description

List of objects of type *TWSImage*.

## **TWSLinesList**

### Description

List of objects of type TWSCall.

# TWSMailAddressBookList

## Description

List of objects of type <u>TWSMailAddressBook</u>.

# TWSMailAddressGroupList

## Description

List of objects of type TWSMailAddressGroup.

# **TWSMailAddressList**

### Description

List of objects of type TWSMailAddress.

# TWSMailAttachmentList

### Description

List of objects of type <u>TWSMailAttachment</u>.

# TWSMailboxToMoveList

### Description

List of objects of type TWSMailboxToMove.

# TWSMailGroupList

## Description

List of objects of type TWSMailGroup.

# TWSMailHistoryList

## Description

List of objects of type *TWSMailHistory*.

# **TWSMailIdErrorList**

### Description

List of objects of type TWSMailIdError.

# **TWSMailIdList**

### Description

List of objects of type Integer to store a mail identifier.

# TWSMailSuspendedList

### Description

List of objects of type TWSMailSuspended.

# TWSMultiChatServiceInfoList

## Description

List of objects of type TWSMultiChatService.

# TWSNoticeHeaderList

## Description

List of objects of type *TWSNoticeHeader*.

# TWSOut bound Contact History List

### Description

List of objects of type TWSOutboundContactHistory.

# TWSOutboundPhoneList

### Description

List of objects of type **TWSOutboundPhone**.

# TWSPredefinedTextTreeItemList

### Description

List of objects of type <u>TWSPredefinedTextTreeItem</u>.

# **TWSQCodeList**

### Description

List of objects of type TWSQCode.

# **TWSReasonList**

### Description

List of objects of type **TWSReason**.

## **TWSServiceInfoList**

### Description

List of objects of type <u>TWSService</u>. For optimization purposes, please note that the full content of the <u>TWSService</u> structure is not sent. The following are the fields delivered in this event: <u>ServiceId</u>, <u>EnableManualACD</u>, <u>EnableInsertOutboundRecord</u> and <u>IntegrationType</u>.

## TWSSkillActionResultList

### Description

List of objects of type TWSSkillActionResult.

## TWSSkillExtensionList

### Description

List of objects of type Int64 to store a skill identifier.

# **TWSSkillList**

### Description

List of objects of type TWSSkill.

# TWSSurveyList

### Description

List of objects of type **TWSSurvey**.

# TWSTemplateFieldList

### Description

List of objects of type *TWSTemplateField*.

# TWSTemplate Tree Item List

### Description

List of objects of type **TWSTemplateTreeItem**.

# **TWSTimeZoneList**

### Description

List of objects of type **TWSTimeZone**.

# **TWSTransferPhoneList**

## Description

List of objects of type TWSTransferPhoneInfo.

# **TWSTransferServiceList**

## Description

List of objects of type TWSTransferServiceInfo.

# TWSWaitingCallAlertList

### Description

List of objects of type TWSWaitingCallAlert.

# **Methods**

## AddCallData

### Description

Adds the data of a variable to the call. This information will be associated with the contact throughout all of its handling process. This method is used, for example, to implement voice and data transfers.

### **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
In	Key	WideString	Name of the variable to identify the data that will be added. The maximum length value for this parameter is 30 characters

In	KeyValue	WideString	Value that will be added. The maximum
			length value for this parameter is 250
			characters

### Return value

Integer. When the returned value is <0, the call data could not be added. If the value is 0, the call data have been successfully added.

### C# syntax

public Integer AgentWS.AddCallData(WideString SessionId , WideString Key ,
WideString KeyValue);

# AddOutgoingMailAttachment

### Description

Adds an attachment to the outgoing e-mail (reply) that is currently being generated.

### **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
In	FileName	WideString	Name of the attachment to be added
In	FileData	Binary	Attachment content
Out	AttachmentId	Integer	Attachment identifier for the Presence system. This identifier must be used to retrieve information about the attachment

### Return value

Integer. When the returned value is <0, the attachment could not be added. If the value is 0, the attachment has been successfully added.

### C# syntax

public Integer AgentWS.AddOutgoingMailAttachment(WideString SessionId ,
WideString FileName , Binary FileData , out Int32 AttachmentId);

# **AnswerCall**

#### Description

Answers the incoming call in progress.

#### **Parameters**

Flag	Parameter	Type	Description

In SessionId WideString Session control identifier

### Return value

Integer. When the returned value is <0, the call could not be answered. If the value is 0, the call has been successfully answered.

### C# syntax

public Integer AgentWS.AnswerCall(WideString SessionId);

# ChangeAgentPassword

#### Description

Replaces the password associated with the agent login.

#### **Parameters**

Flag	Parameter	Type	Description
In	SessionId	WideString	Session control identifier
In	OldPassword	WideString	Former password used by the agent to log in to the Presence system
In	NewPassword	WideString	New password for the agent login
In	CheckNewPasswo rd	WideString	New password confirmation, which must match the value entered in the <i>NewPassword</i> parameter

#### Return value

Integer. When the returned value is <0, the password could not be changed. If the value is 0, the password has been successfully changed.

#### C# syntax

public Integer AgentWS.ChangeAgentPassword(WideString SessionId ,
WideString OldPassword , WideString NewPassword , WideString
CheckNewPassword);

# CheckTimeZonePrefix

### Description

Checks whether the phone number matches the specified time zone using the area codes defined in the system. If it doesn't, it returns the identifier of the matching time zone.

#### **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
In	Phone	WideString	Phone number to be checked
In	TimeZoneId	WideString	Identifier of the time zone to be matched with the phone number
Out	ValidTimeZone	Boolean	Indicates if the phone number matches the specified <i>TimeZoneId</i>
Out	ValidTimeZoneId	WideString	Matching <i>TimeZoneId</i> for the specified phone number. Only if the <i>ValidTimeZone</i> parameter is FALSE

#### Return value

Integer. When the returned value is <0, the time zone could not be checked. If the value is 0, the time zone has been successfully checked.

#### C# syntax

public Integer AgentWS.CheckTimeZonePrefix(WideString SessionId ,
WideString Phone , WideString TimeZoneId , out Boolean ValidTimeZone , out
WideString ValidTimeZoneId);

# CheckTimeZoneSchedule

### Description

Checks whether the date and time that the current outbound contact has been scheduled to is within the service time zone calling hours.

#### **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
In	TimeZoneId	WideString	Identifier of the time zone used to check the scheduled date
Out	Valid	Boolean	Indicates if the date and time is within the service time zone calling hours

### Return value

Integer. When the returned value is <0, the scheduled date could not be checked. If the value is 0, the scheduled date has been successfully checked.

C# syntax

public Integer AgentWS.CheckTimeZoneSchedule(WideString SessionId ,
WideString TimeZoneId , out Boolean Valid);

## CheckTimeZoneService

### Description

Checks whether the time zone has been added to the outbound service that the current contact belongs to.

### **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
In	TimeZoneId	WideString	Identifier of the time zone to be checked
Out	Valid	Boolean	Indicates if the time zone is added to the service

### Return value

Integer. When the returned value is <0, the time zone could not be checked. If the value is 0, the time zone has been successfully checked.

### C# syntax

public Integer AgentWS.CheckTimeZoneService(WideString SessionId ,
WideString TimeZoneId , out Boolean Valid);

## ClearAllCalls

### Description

Hangs up all existing calls in the agent extension.

#### **Parameters**

Flag Parameter Type Description

In SessionId WideString Session control identifier

### Return value

Integer. When the returned value is <0, the calls could not be hung up. If the value is 0, the calls have been successfully hung up.

### C# syntax

public Integer AgentWS.ClearAllCalls(WideString SessionId);

# ClearCall

## Description

Hangs up only the active call, if applicable.

### **Parameters**

Flag Parameter Type Description

In SessionId WideString Session control identifier

### Return value

Integer. When the returned value is <0, the call could not be hung up. If the value is 0, the call has been successfully hung up.

# C# syntax

public Integer AgentWS.ClearCall(WideString SessionId);

# CloseContact

## Description

Finishes the current contact.

### **Parameters**

Flag Parameter Type Description

In SessionId WideString Session control identifier

#### Return value

Integer. When the returned value is <0, the contact could not be finished. If the value is 0, the contact has been successfully finished.

## C# syntax

public Integer AgentWS.CloseContact(WideString SessionId);

# ConferenceCall

## Description

Creates a conference call with the active call and the call that was placed on hold when generating the active call.

### **Parameters**

Flag Parameter Type Description

In SessionId WideString Session control identifier

### Return value

Integer. When the returned value is <0, the conference call could not be established. If the value is 0, the conference call has been successfully established.

#### C# syntax

public Integer AgentWS.ConferenceCall(WideString SessionId);

# DeleteCallData

#### Description

Deletes data attached to the call. The data to be deleted are identified by a particular key.

### **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
In	Key	WideString	Name of the variable which identifies the data to be deleted. The maximum length value for this parameter is 30 characters

## Return value

Integer. When the returned value is <0, the call data could not be deleted. If the value is 0, the call data have been successfully deleted.

### C# syntax

public Integer AgentWS.DeleteCallData(WideString SessionId , WideString
Key);

# DeleteOutgoingMailAttachment

## Description

Deletes an attachment from the outgoing e-mail (reply) that is currently being generated using the attachment identifier for the Presence system.

## **Parameters**

n
)

In SessionId WideString Session control identifier

In AttachmentId Integer Identifier of the attachment to be deleted

from the outgoing e-mail

### Return value

Integer. When the returned value is <0, the attachment could not be deleted. If the value is 0, the attachment has been successfully deleted.

### C# syntax

public Integer AgentWS.DeleteOutgoingMailAttachment(WideString SessionId ,
Int32 AttachmentId);

# DropCall

## Description

Hangs up the call of the latest party to join the active conference call. This method is only available on Avaya PBXs.

## **Parameters**

No parameters

# Return value

Integer. When the returned value is <0, the latest party who joined the active conference call could not be disconnected. If the value is 0, the latest party has been successfully disconnected.

# C# syntax

public Integer AgentWS.DropCall();

# **Fcho**

#### Description

Test method: Returns the entered string.

## **Parameters**

Flag Parameter Type Description

In Value WideString Parameter string

# Return value

# WideString

## C# syntax

public WideString AgentWS.Echo(WideString Value);

# ExecuteCustomButton

## Description

Executes the assigned programming script from a specific custom button.

### **Parameters**

Flag	Parameter	Type	Description
In	SessionId	WideString	Session control identifier
In	CustomButtonId	WideString	Custom button identifier

#### Return value

Integer. When the returned value is <0, the programming script could not be executed. If the value is 0, the programming script has been successfully executed.

#### C# syntax

public Integer AgentWS.ExecuteCustomButton(WideString SessionId ,
WideString CustomButtonId);

# ExecuteScriptCode

### Description

Executes the specified programming script code.

## **Parameters**

Flag	Parameter	Type	Description
In	SessionId	WideString	Session control identifier
In	ScriptCode	WideString	Programming script code to be executed
In	Language	TWSScriptLanguage	Specifies the programming language used to define the script code

# Return value

Integer. When the returned value is <0, the programming script code could not be executed. If the value is 0, the programming script code has been successfully executed.

# C# syntax

public Integer AgentWS.ExecuteScriptCode(WideString SessionId , WideString ScriptCode , TWSScriptLanguage Language);

# **FinalizeContact**

Deprecated. Refer to FinalizeContact3.

# FinalizeContact2

Deprecated. Refer to FinalizeContact3.

# FinalizeContact3

# Description

Finishes the current contact.

## **Parameters**

Flag	Parameter	Type	Description
In	SessionId	WideString	Session control identifier
In	QInfo	TWSQInfo3	Contact qualification information
In	FinalizedByMaxACW	Boolean	Indicates if the contact is finished by maximum after-call work time

### Return value

Integer. When the returned value is <0, the contact could not be finished. If the value is 0, the contact has been successfully finished.

# C# syntax

public Integer AgentWS.FinalizeContact3(WideString SessionId , TWSQInfo3
QInfo , Boolean FinalizeByMaxACW);

# FinalizeUnansweredMails

## Description

Finalizes unanswered incoming e-mails (i.e., changes the state to 'completed') without the need for an agent to handle them.

#### **Parameters**

Flag	Parameter	Type	Description
In	SessionId	WideString	Session control identifier
In	MailIds	<u>TWSMailIdList</u>	List of identifiers of the e-mails to be finalized
Out	MailldsError	<u>TWSMailIdErrorList</u>	List of identifiers of the e-mails that could not be finalized. For each mail identifier, the reason (error code) why the e-mail could not be finalized is also listed. Refer to <i>Error Codes</i> for a description of the most common error codes

## Return value

Integer. When the returned value is <0, the list of incoming e-mails to be finalized could not be processed. If the value is 0, the list of incoming e-mails to be finalized has been successfully processed.

# C# syntax

public Integer AgentWS.FinalizeUnansweredMails(WideString SessionId ,
TWSMailIdList MailIds , out TWSMailIdErrorList MailIdsError);

# GetAgentCount

# Description

Retrieves the total number of agents that are connected to the Web Services.

#### **Parameters**

Flag	Parameter	Lype	Description
------	-----------	------	-------------

Out AgentCount Integer Number of agents connected to the Web

Services

### Return value

Integer. When the returned value is <0, the number of agents could not be retrieved. If the value is 0, the number of agents has been successfully retrieved.

### C# syntax

public Integer AgentWS.GetAgentCount(out Int32 AgentCount);

# GetAgentName

### Description

Retrieves the name of the agent.

#### **Parameters**

Flag	Parameter	Type	Description

In SessionId WideString Session control identifier

Out AgentName WideString Name of the agent

## Return value

Integer. When the returned value is <0, the name of the agent could not be retrieved. If the value is 0, the name of the agent has been successfully retrieved.

### C# syntax

public Integer AgentWS.GetAgentName(WideString SessionId , out WideString AgentName);

# GetAgentState

### Description

Retrieves the current agent state.

# **Parameters**

Flag Parameter Type Description

In SessionId WideString Session control identifier

Out State <u>TWSAgentState</u> Current agent state

### Return value

Integer. When the returned value is <0, the agent state could not be retrieved. If the value is 0, the agent state has been successfully retrieved.

# C# syntax

public Integer AgentWS.GetAgentState(WideString SessionId , out TWSAgentState State);

# GetAgentStation

# Description

Retrieves the station number to which the agent is connected.

#### **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
Out	StationNumber	Int64	Station number to which the agent is connected

### Return value

Integer. When the returned value is <0, the station number could not be retrieved. If the value is 0, the station number has been successfully retrieved.

# C# syntax

public Integer AgentWS.GetAgentStation(WideString SessionId , out Int64 StationNumber);

# GetAllCallData

#### Description

Retrieves the values of all the variables attached to the current contact.

#### **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
Out	CallDataList	TWSCallDataInformationList	List of all variables attached to the current contact

#### Return value

Integer. When the returned value is <0, the call data could not be retrieved. If the value is 0, the call data have been successfully retrieved.

# C# syntax

public Integer AgentWS.GetAllCallData(WideString SessionId , out TWSCallDataInformationList CallDataList);

# GetAlternativePhoneDescription

## Description

Retrieves the description of an alternative phone number with its code.

### **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
In	PhoneDescriptionCode	Integer	Code of the alternative phone number description
Out	PhoneDescription	WideString	Alternative phone number description

### Return value

Integer. When the returned value is <0, the description not be retrieved. If the value is 0, the description has been successfully retrieved.

### C# syntax

public Integer AgentWS.GetAlternativePhoneDescription(WideString SessionId
, Int32 PhoneDescriptionCode , out WideString PhoneDescription);

# GetAlternativePhones

### Description

Retrieves the alternative phone numbers of the current contact.

### Parameters

Flag	Parameter	Type	Description
In	SessionId	WideString	Session control identifier
Out	Phones	<u>TWSOutboundPhoneList</u>	List of alternative phone numbers of the current contact

#### Return value

Integer. When the returned value is <0, the alternative phone numbers could not be retrieved. If the value is 0, the alternative phone numbers have been successfully retrieved.

## C# syntax

public Integer AgentWS.GetAlternativePhones(WideString SessionId , out TWSOutboundPhoneList Phones);

# GetAppData

### Description

Retrieves additional information (such as business data) that is attached to the contact information and stored in the APPDATA field of the PCO\_INBOUNDLOG table (refer to the *Presence Repository* manual). For inbound services only.

#### **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
Out	AppData	WideString	Additional information attached to the contact information. Supports up to 30 characters

### Return value

Integer. When the returned value is <0, the information could not be retrieved. If the value is 0, the information has been successfully retrieved.

# C# syntax

public Integer AgentWS.GetAppData(WideString SessionId , out WideString AppData);

# GetCallData

# Description

Retrieves the value of a variable attached to the contact.

# **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
In	Key	WideString	Name of the attached variable to be retrieved. The maximum length value for this parameter is 30 characters
Out	KeyValue	WideString	Value of the retrieved variable. The maximum length value for this parameter is

## 250 characters

### Return value

Integer. When the returned value is <0, the call data could not be retrieved. If the value is 0, the call data have been successfully retrieved.

### C# syntax

public Integer AgentWS.GetCallData(WideString SessionId , WideString Key ,
out WideString KeyValue);

# GetCanChangePassword

## Description

Indicates if the agent can change the password associated with his/her agent login.

### **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
Out	Value	Boolean	Indicates if the agent can change the password

### Return value

Integer. When the returned value is <0, the method could not be executed. If the value is 0, the *Value* parameter is returned.

### C# syntax

 $\verb|public Integer AgentWS.GetCanChangePassword(WideString SessionId , \verb|out| Boolean Value)|; \\$ 

# GetContactData1

# Description

Retrieves the first custom data value assigned to the outbound or inbound contact that is being handled.

# **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
Out	ContactData	WideString	First custom data value assigned to the contact

#### Return value

Integer. When the returned value is <0, the custom data value could not be retrieved. If the value is 0, the custom data value has been successfully retrieved.

### C# syntax

public Integer AgentWS.GetContactData1(WideString SessionId , out WideString ContactData);

# GetContactData2

## Description

Retrieves the second custom data value assigned to the outbound or inbound contact that is being handled.

#### **Parameters**

Flag	Parameter	Type	Description
In	SessionId	WideString	Session control identifier
Out	ContactData	WideString	Second custom data value assigned to the contact

## Return value

Integer. When the returned value is <0, the custom data value could not be retrieved. If the value is 0, the custom data value has been successfully retrieved.

## C# syntax

public Integer AgentWS.GetContactData2(WideString SessionId , out WideString ContactData);

# GetContactData3

### Description

Retrieves the third custom data value assigned to the outbound or inbound contact that is being handled.

#### **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
Out	ContactData	WideString	Third custom data value assigned to the contact

#### Return value

Integer. When the returned value is <0, the custom data value could not be retrieved. If the value is 0, the custom data value has been successfully retrieved.

#### C# syntax

public Integer AgentWS.GetContactData3(WideString SessionId , out WideString ContactData);

# GetCurrentContact

## Description

Retrieves the general information about the current contact.

#### **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
Out	ServiceContact	TWSContact	Current contact general information

## Return value

Integer. When the returned value is <0, the contact information could not be retrieved. If the value is 0, the contact information has been successfully retrieved.

### C# syntax

public Integer AgentWS.GetCurrentContact(WideString SessionId , out TWSContact ServiceContact);

# GetCurrentInboundContact

# Description

Retrieves information about the current inbound contact.

# **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
Out	InboundContact	<u>TWSInboundContact</u>	Current inbound contact information

### Return value

Integer. When the returned value is <0, the contact information could not be retrieved. If the value is 0, the contact information has been successfully retrieved.

### C# syntax

public Integer AgentWS.GetCurrentInboundContact(WideString SessionId , out TWSInboundContact InboundContact);

# GetCurrentInboundService

### Description

Retrieves information about the inbound service that the current contact belongs to.

#### **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
Out	CurrentService	TWSInboundService	Inbound service information

#### Return value

Integer. When the returned value is <0, the service information could not be retrieved. If the value is 0, the service information has been successfully retrieved.

# C# syntax

public Integer AgentWS.GetCurrentInboundService(WideString SessionId , out TWSInboundService CurrentService);

# GetCurrentInternetContact

# Description

Retrieves information about the current Internet contact.

### **Parameters**

Flag	Parameter	Type	Description
In	SessionId	WideString	Session control identifier
Out	InternetContact	TWSInternetContact	Internet contact information

#### Return value

Integer. When the returned value is <0, the contact information could not be retrieved. If the value is 0, the contact information has been successfully retrieved.

# C# syntax

public Integer AgentWS.GetCurrentInternetContact(WideString SessionId ,
out TWSInternetContact InternetContact);

# GetCurrentMailContact

## Description

Retrieves information about the current mail contact. This method can only be called after receiving the <u>AGM\_MA/L/N\_DECODED</u> event, which indicates that the current mail contact has been decoded. contact.

#### **Parameters**

Flag	Parameter	Type	Description

In SessionId WideString Session control identifier

Out MailContact TWSMailContact Current mail contact information

# Return value

Integer. When the returned value is <0, the mail contact information could not be retrieved. If the value is 0, the mail contact information has been successfully retrieved.

#### C# svntax

public Integer AgentWS.GetCurrentMailContact(WideString SessionId , out TWSMailContact MailContact);

# GetCurrentOutboundContact

### Description

Retrieves information about the current outbound contact.

### **Parameters**

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

In SessionId WideString Session control identifier

Out OutboundContact <u>TWSOutboundContact</u> Current outbound contact

information

#### Return value

Integer. When the returned value is <0, the contact information could not be retrieved. If the value is 0, the contact information has been successfully retrieved.

C# syntax

public Integer AgentWS.GetCurrentOutboundContact(WideString SessionId ,
out TWSOutboundContact OutboundContact);

# GetCurrentOutboundService

## Description

Retrieves information about the outbound service that the current contact belongs to.

### **Parameters**

Flag F	Parameter	Type	Description
--------	-----------	------	-------------

In SessionId WideString Session control identifier

Out CurrentService TWSOutboundService Outbound service information

### Return value

Integer. When the returned value is <0, the service information could not be retrieved. If the value is 0, the service information has been successfully retrieved.

#### C# syntax

public Integer AgentWS.GetCurrentOutboundService(WideString SessionId ,
out TWSOutboundService CurrentService);

# GetCurrentService

### Description

Retrieves general information about the service that the current contact belongs to.

# **Parameters**

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

In SessionId WideString Session control identifier

Out *CurrentService* <u>TWSService</u> Service general information

## Return value

Integer. When the returned value is <0, the service information could not be retrieved. If the value is 0, the service information has been successfully retrieved.

# C# syntax

public Integer AgentWS.GetCurrentService(WideString SessionId , out TWSService CurrentService);

# GetCurrentServiceLogo

## Description

Retrieves the logo for the service that the current contact belongs to.

### **Parameters**

Flag	Parameter	Type	Description
In	SessionId	WideString	Session control identifier
Out	LogoLength	Integer	Length of the logo in bytes
Out	<i>LogoType</i>	<u>TWSImageType</u>	Image format for the logo
Out	LogoData	Binary	Image content

### Return value

Integer. When the returned value is <0, the logo could not be retrieved. If the value is 0, the logo has been successfully retrieved.

## C# syntax

public Integer AgentWS.GetCurrentServiceLogo(WideString SessionId , out Int32 LogoLength , out TWSImageType LogoType , out Binary LogoData);

# GetCustomButtons

Deprecated. Refer to <u>GetCustomButtons2</u>.

# GetCustomButtons2

## Description

Retrieves information about the custom buttons that can be used by the agent.

### **Parameters**

Flag	Parameter	Туре	Description
------	-----------	------	-------------

In SessionId WideString Session control identifier

In Format <u>TWS/mageFormat</u> Image file format for the icon

Out CustomButtons <u>TWSCustomButtonList</u> List of custom buttons that can

be used by the agent

#### Return value

Integer. When the returned value is <0, the custom buttons could not be retrieved. If the value is 0, the custom buttons have been successfully retrieved.

C# syntax

```
public Integer AgentWS.GetCustomButtons2(WideString SessionId ,
TWSImageFormat Format , out TWSCustomButtonList CustomButtons);
```

# GetCustomData1

# Description

Retrieves the first custom data value of the outbound record associated to the contact that is being handled. For outbound services only.

### **Parameters**

Flag Parameter Type Description

In SessionId WideString Session control identifier

Out CustomData WideString First custom data value assigned to the

outbound record

## Return value

Integer. When the returned value is <0, the custom data value could not be retrieved. If the value is 0, the custom data value has been successfully retrieved.

# C# syntax

```
public Integer AgentWS.GetCustomData1(WideString SessionId , out
WideString CustomData);
```

# GetCustomData2

## Description

Retrieves the second custom data value of the outbound record associated to the contact that is being handled. For outbound services only.

#### **Parameters**

Flag Parameter Type Description

In	SessionId	WideString	Session control identifier
----	-----------	------------	----------------------------

Out CustomData WideString Second custom data value assigned to the

outbound record

#### Return value

Integer. When the returned value is <0, the custom data value could not be retrieved. If the value is 0, the custom data value has been successfully retrieved.

#### C# syntax

public Integer AgentWS.GetCustomData2(WideString SessionId , out WideString CustomData);

# GetCustomData3

# Description

Retrieves the third custom data value of the outbound record associated to the contact that is being handled. For outbound services only.

### **Parameters**

Flag	Parameter	Type	Description
In	SessionId	WideString	Session control identifier
Out	CustomData	WideString	Third custom data value assigned to the outbound record

## Return value

Integer. When the returned value is <0, the custom data value could not be retrieved. If the value is 0, the custom data value has been successfully retrieved.

# C# syntax

public Integer AgentWS.GetCustomData3(WideString SessionId , out WideString CustomData);

# **GetEvent**

## Description

Retrieves the list of events that have not been processed yet. After retrieving the events, it deletes given events from the server.

#### **Parameters**

Flag Parameter Type Description

In SessionId WideString Session control identifier

Out *Events <u>TWSAgentEventList</u>* List of pending events

### Return value

Integer. When the returned value is <0, the events could not be retrieved. If the value is 0, the events have been successfully retrieved.

# C# syntax

public Integer AgentWS.GetEvent(WideString SessionId , out TWSAgentEventList Events);

# GetIncomingMailAttachment

### Description

Retrieves an attachment from the incoming e-mail that is currently being handled.

## **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
In	AttachmentPathIds	s <u>TWSAttachmentIdList</u>	Path of attachment identifiers. When the attachment has been added directly to the e-mail, only the attachment identifier is needed. However, the attachment to be retrieved may be included within another attachment in mail format (.eml extension). In such case you must specify the path of identifiers needed to browse through the different e-mails that have been added as attachments to the e-mail

# Return value

Out FileData

Integer. When the returned value is <0, the attachment could not be retrieved. If the value is 0, the attachment has been successfully retrieved.

Attachment content

Binary

#### C# syntax

public Integer AgentWS.GetIncomingMailAttachment(WideString SessionId ,
TWSAttachmentIdList AttachmentPathIds , out Binaty FileData);

# GetIncomingMailAttachmentAsMail

# Description

Retrieves an attached e-mail (i.e., an attachment with .eml extension) in mail format (*TWSMail*) from the incoming e-mail that is currently being handled.

### **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
In	AttachmentPathIds	s <u>TWSAttachmentIdList</u>	Path of attachment identifiers. When the attachment has been added directly to the e-mail, only the attachment identifier is needed. However, the attachment to be retrieved may be included within another attachment in mail format (.eml extension). In such case you must specify the path of identifiers needed to browse through the different e-mails that have been added as attachments to the e-mail
Out	Mail	<u>TWSMail</u>	Attachment content in mail

### Return value

Integer. When the returned value is <0, the attachment could not be retrieved. If the value is 0, the attachment has been successfully retrieved.

format

## C# syntax

public Integer AgentWS.GetIncomingMailAttachmentAsMail(WideString
SessionId , TWSAttachmentIdList AttachmentPathIds , out TWSMail Mail);

# GetInvalidQCodes

### Description

Retrieves the list of invalid qualification codes for a specific service.

# **Parameters**

Flag Parameter Type Description

In Session/d WideString Session control identifier

In	ServiceId	Integer	Service identifier
----	-----------	---------	--------------------

Out *QCodes TWSQCodeList* List of invalid qualification codes

#### Return value

Integer. When the returned value is <0, the invalid qualification codes could not be retrieved. If the value is 0, the invalid qualification codes have been successfully retrieved.

## C# syntax

public Integer AgentWS.GetInvalidQCodes(WideString SessionId , Int32 ServiceId , out TWSQCodeList QCodes);

# **GetLines**

# Description

Retrieves the information about all the call lines for the agent extension.

#### **Parameters**

Flag	Parameter	Type	Description

In SessionId WideString Session control identifier

Out *Lines* <u>TWSLinesList</u> List of call lines in the agent extension

## Return value

Integer. When the returned value is <0, the call lines could not be retrieved. If the value is 0, the call lines have been successfully retrieved.

#### C# syntax

public Integer AgentWS.GetLines(WideString SessionId , out TWSLinesList Lines);

# GetLogoutReasons

### Description

Retrieves the logout reasons that can be selected by the agent when closing the application.

#### **Parameters**

Flag Parameter Type Description

In SessionId WideString Session control identifier

Out *Reasons* <u>TWSReasonList</u> List of logout reasons

#### Return value

Integer. When the returned value is <0, the logout reasons could not be retrieved. If the value is 0, the logout reasons have been successfully retrieved.

#### C# syntax

public Integer AgentWS.GetLogoutReasons(WideString SessionId , out TWSReasonList Reasons);

# GetMailAddressBooks

## Description

Retrieves the list of address books to which the agent has access to handle the current mail contact, including all e-mail addresses and address groups belonging to it.

### **Parameters**

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

In SessionId WideString Session control identifier

Out Books <u>TWSMailAddressBookList</u> List of address books

#### Return value

Integer. When the returned value is <0, the list of address books could not be retrieved. If the value is 0, the list of address books has been successfully retrieved.

### C# syntax

public Integer AgentWS.GetMailAddressBooks(WideString SessionId , out TWSMailAddressBookList Books);

# GetMailboxesToMove

### Description

Retrieves the list of mailboxes to which the incoming e-mail that is being handled by the agent can be moved.

### **Parameters**

Flag Parameter Type Description

In SessionId WideString Session control identifier

Out *Mailboxes* <u>TWSMailboxToMoveList</u> List of mailboxes to which the incoming e-mail can be moved

### Return value

Integer. When the returned value is <0, the list of mailboxes could not be retrieved. If the value is 0, the list of mailboxes has been successfully retrieved.

### C# syntax

public Integer AgentWS.GetMailboxesToMove(WideString SessionId , out TWSMailboxToMoveList Mailboxes);

# GetMailHistoryAttachment

# Description

Retrieves an attachment in binary format from an e-mail found through a search within the mail history.

### **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
In	MailType	<u>TWSMailType</u>	Indicates if the search has been performed through received e-mails or sent e- mails
In	MailHistoryId	Integer	Identifier assigned to the e- mail within the retrieved mail history
In	AttachmentPathIds	<u>TWSAttachmentIdList</u>	Path of attachment identifiers. When the attachment has been added directly to the e-mail, only the attachment identifier is needed. However, the attachment to be retrieved may be included within another attachment in mail format (.eml extension). In such case you must specify the path of identifiers needed to browse through the different e-mails that have been added as attachments to the e-mail
Out	FileData	Binary	Attachment content

# Return value

Integer. When the returned value is <0, the attachment could not be retrieved. If the value is 0, the attachment has been successfully retrieved.

## C# syntax

public Integer AgentWS.GetMailHistoryAttachment(WideString SessionId ,
TWSMailType MailType , Int32 MailHistoryId , TWSAttachmentIdList
AttachmentPathIds , out Binaty FileData);

# GetMailHistoryAttachmentAsMail

# Description

Retrieves an attachment in mail format (*TWSMail*) from an e-mail found through a search within the mail history.

## **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
In	MailType	<u>TWSMailType</u>	Indicates if the search has been performed through received e-mails or sent e- mails
In	MailHistoryId	Integer	Identifier assigned to the e- mail within the retrieved mail history
In	AttachmentPathIds	<u>TWSAttachmentIdList</u>	Path of attachment identifiers. When the attachment has been added directly to the e-mail, only the attachment identifier is needed. However, the attachment to be retrieved may be included within another attachment in mail format (.eml extension). In such case you must specify the path of identifiers needed to browse through the different e-mails that have been added as attachments to the e-mail
Out	Mail	<u>TWSMail</u>	Attachment content in mail format

# Return value

Integer. When the returned value is <0, the attachment could not be retrieved. If the value is 0, the attachment has been successfully retrieved.

C# syntax

public Integer AgentWS.GetMailHistoryAttachmentAsMail(WideString SessionId
, TWSMailType MailType , Int32 MailHistoryId , TWSAttachmentIdList
AttachmentPathIds , out TWSMail Mail);

# GetMailHistory

# Description

Retrieves the information about an e-mail found through a search within the mail history.

### **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
In	MailHistoryId	Integer	Identifier assigned to the e-mail within the retrieved mail history
Out	Mail	<u>TWSMail</u>	Information about the e-mail

### Return value

Integer. When the returned value is <0, the e-mail could not be retrieved. If the value is 0, the e-mail has been successfully retrieved.

# C# syntax

public Integer AgentWS.GetMailHistory(WideString SessionId , Int32
MailHistoryId , out TWSMail Mail);

# GetMailHistoryLog

## Description

Retrieves the mail history using certain search filtering fields.

### **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
In	MailType	<u>TWSMailType</u>	Indicates if you wish to search through received e-mails or sent e-mails
In	Mailld	Integer	Mail identifier to be used as a filter to locate a specific e-mail. If the value is

			set to 0, the search is filtered by sender (incoming mail) or recipient (outgoing mail), time range and state. Otherwise, those filtering fields are ignored
In	Sender	WideString	Full or partial e-mail address of the sender (incoming mail) or the recipient (outgoing mail) to be used as a filter. Only if the <i>Mailld</i> parameter is set to 0
In	StartDate	DateTime	Date that marks the start of the time period within which you wish to perform the search for e-mails. Only if the <i>Mailld</i> parameter is set to 0
In	<i>EndDate</i>	DateTime	Date that marks the end of the time period within which you wish to perform the search for e-mails. Only if the <i>Mailld</i> parameter is set to 0
In	MailStatusFiltei	<u>TWSMailStatusFilter</u>	Mail status to filter results (All, Unanswered, In progress, Completed or Suspended for incoming mail and All, Unsent and Sent for outgoing mail). Only if the <i>Mailld</i> parameter is set to 0
Out	MailHistoryLog	TWSMailHistoryList	Mail history matching the specified filtering fields

### Return value

Integer. When the returned value is <0, the mail history could not be retrieved. If the value is 0, the mail history has been successfully retrieved.

## C# syntax

public Integer AgentWS.GetMailHistoryLog(WideString SessionId ,
TWSMailType MailType , Int32 MailId , WideString Sender , DateTime
StartDate , DateTime EndDate , TWSMailStatusFilter MailStatusFilter , out
TWSMailHistoryList MailHistoryLog);

# GetMailsSuspended

### Description

Retrieves the list of e-mails that have been suspended by the agent.

# **Parameters**

Flag Parameter Type Description

In SessionId WideString Session control identifier

Out *Mails* <u>TWSMailSuspendedList</u> List of e-mails suspended by the agent

### Return value

Integer. When the returned value is <0, the list of suspended e-mails could not be retrieved. If the value is 0, the list of suspended e-mails has been successfully retrieved.

## C# syntax

public Integer AgentWS.GetMailsSuspended(WideString SessionId , out TWSMailSuspendedList Mails);

# GetManagedSkills

### Description

Retrieves the list of skills that the agent login can control (pause/resume). Only for the OpenGate platform.

## **Parameters**

Flag Parameter Type Description

In SessionId WideString Session control identifier

# Return value

Integer. When the returned value is <0, the list of skills under agent control could not be retrieved. If the value is 0, the list of skills under agent control has been successfully retrieved.

### C# syntax

public Integer AgentWS.GetManagedSkills(WideString SessionId , out TWSSkillList Skills);

# GetNoticeBody

### Description

Retrieves the notice body from a specific notice.

## **Parameters**

Flag Parameter Type Description

In SessionId WideString Session control identifier

In Integer Notice identifier used to retrieve the notice

body

Out *Body* WideString Notice body

#### Return value

Integer. When the returned value is <0, the notice body could not be retrieved. If the value is 0, the notice body has been successfully retrieved.

## C# syntax

public Integer AgentWS.GetNoticeBody(WideString SessionId , Int32 Id , out WideString Body);

# **GetNoticesHeaders**

# Description

Retrieves the list of notices of the agent. The notice body is not included.

#### **Parameters**

Flag Parameter Type Description

In SessionId WideString Session control identifier

Out NoticeHeaders TWSNoticeHeaderList List of notices

## Return value

Integer. When the returned value is <0, the list of notices could not be retrieved. If the value is 0, the list of notices has been successfully retrieved.

#### C# syntax

public Integer AgentWS.GetNoticesHeaders(WideString SessionId , out TWSNoticeHeaderList NoticeHeaders);

# GetOutboundContactHistory

## Description

Retrieves the history of contacts previously made to the outbound record currently being handled.

#### **Parameters**

Flag Parameter Type Description

In SessionId WideString Session control identifier

TWSOutboundContactHistoryList History of outbound Out History contacts

#### Return value

Integer. When the returned value is <0, the history of contacts could not be retrieved. If the value is 0, the history of contacts has been successfully retrieved.

# C# syntax

public Integer AgentWS.GetOutboundContactHistory(WideString SessionId , out TWSOutboundContactHistoryList History);

# GetOutgoingMail

### Description

Retrieves the content and attachments of the outgoing e-mail (reply) that is currently defined.

### **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
In	Mail	<u>TWSMail</u>	Content and attachments of the current outgoing e-mail

### Return value

Integer. When the returned value is <0, the outgoing e-mail could not be retrieved. If the value is 0, the outgoing e-mail has been successfully retrieved.

# C# syntax

public Integer AgentWS.GetOutgoingMail(WideString SessionId , out TWSMail Mail);

# GetPartialMailAttachment

# Description

Retrieves an attachment to the partial reply associated with the incoming e-mail that is being handled.

### **Parameters**

Flag Parameter Type Description

In SessionId WideString Session control identifier

In AttachmentId Integer Identifier of the attachment to be retrieved

from the partial reply

Out FileData Binary Attachment content

### Return value

Integer. When the returned value is <0, the attachment could not be retrieved. If the value is 0, the attachment has been successfully attached.

## C# syntax

public Integer AgentWS.GetPartialMailAttachment(WideString SessionId ,
Int32 AttachmentId , out Binary FileData);

# GetPhoneDescriptions

### Description

Retrieves the list of all alternative phone descriptions (Home, Mobile, Work...) defined in the system.

#### **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
Out	AlternativePhoneDescriptions	TWSAlternativePhone DescriptionList	List of alternative phone descriptions

#### Return value

Integer. When the returned value is <0, the alternative phone descriptions could not be retrieved. If the value is 0, the alternative phone descriptions have been successfully retrieved.

### C# syntax

public Integer AgentWS.GetPhoneDescriptions(WideString SessionId , out TWSAlternativePhoneDescriptionList AlternativePhoneDescriptions);

# GetPredefinedText

### Description

Retrieves a mail predefined text.

# **Parameters**

Flag	Parameter	Type	Description

In SessionId WideString Session control identifier

In Id Integer Predefined text identifier

Out *Data <u>TWSPredefinedText</u>* Predefined text

### Return value

Integer. When the returned value is <0, the predefined text could not be retrieved. If the value is 0, the predefined text has been successfully retrieved.

### C# syntax

public Integer AgentWS.GetPredefinedText(WideString SessionId , Int32 Id ,
out TWSPredefinedText Data);

# GetPredefinedTextTreeItems

### Description

Retrieves the list of predefined text groups and predefined texts to which the agent has access to handle the current mail contact.

# Parameters

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

In SessionId WideString Session control identifier

Out PredefinedTexts <u>TWSPredefinedTextTreeItemList</u> List of available groups

and predefined texts that can be used to handle the current mail contact

# Return value

Integer. When the returned value is <0, the list of groups and predefined texts could not be retrieved. If the value is 0, the list of groups and predefined texts has been successfully retrieved.

# C# syntax

public Integer AgentWS.GetPredefinedTextTreeItems(WideString SessionId ,
out TWSPredefinedTextTreeItemList PredefinedTexts);

# **GetQCode**

# Description

Retrieves the qualification code selected for the current contact.

#### **Parameters**

Flag	Paramete	r Type	Description

In SessionId WideString Session control identifier

Out *QCode* Integer Selected qualification code

## Return value

Integer. When the returned value is <0, the qualification code could not be retrieved. If the value is 0, the qualification code has been successfully retrieved.

## C# syntax

public Integer AgentWS.GetQCode(WideString SessionId , out Int32 QCode);

# GetQInfo

Deprecated. Refer to GetOInfo3.

# GetQInfo2

Deprecated. Refer to GetOInfo3.

# GetQInfo3

## Description

Retrieves the information about the qualification of the current contact.

## **Parameters**

Flag	Parameter	Type	Description

In SessionId WideString Session control identifier

Out *QInfo TWSQInfo3* Qualification information

Return value

Integer. When the returned value is <0, the qualification information could not be retrieved. If the value is 0, the qualification information has been successfully retrieved.

### C# syntax

public Integer AgentWS.GetQInfo3(WideString SessionId , out TWSQInfo3
QInfo);

# **GetScheduledQCodes**

### Description

Retrieves the list of scheduled qualification codes for a specific outbound service.

#### **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
In	ServiceId	Integer	Outbound service identifier
Out	<i>QCodes</i>	<u>TWSQCodeList</u>	List of scheduled qualification codes

## Return value

Integer. When the returned value is <0, the scheduled qualification codes could not be retrieved. If the value is 0, the scheduled qualification codes have been successfully retrieved.

## C# syntax

public Integer AgentWS.GetScheduledQCodes(WideString SessionId , Int32 ServiceId , out TWSQCodeList QCodes);

# **GetServerStatus**

## Description

Retrieves the list of Presence server states.

### **Parameters**

Flag	Parameter	Type	Description
In	SessionId	WideString	Session control identifier
Out	Server	<u>TWSServer</u>	Presence Server state information
Out	AdditionalServers	TWSServerList	Additional Presence servers state information (Presence Mail

Interactions Server, Presence Web Interactions Server and Presence Recording Server)

## Return value

Integer. When the returned value is <0, the server states could not be retrieved. If the value is 0, the server states have been successfully retrieved.

# C# syntax

```
public Integer AgentWS.GetServerStatus(WideString SessionId , out
TWSServer Server , out TWSServerList AdditionalServers);
```

# **GetServerTime**

### Description

Test method: Retrieves the current date and time of the machine where the Web Services are installed.

### **Parameters**

No parameters

## Return value

DateTime

# C# syntax

public DateTime AgentWS.GetServerTime();

# GetServiceAgents

### Description

Retrieves the list of agents that are working in a specific service.

# **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
In	ServiceId	Integer	Identifier of the service from which the list of agents will be retrieved
Out	AgentsInService	TWSAgentServiceInformationList	List of agents working

Enghouse Interactive Presence Suite

in the service

#### Return value

Integer. When the returned value is <0, the list of agents could not be retrieved. If the value is 0, the list of agents has been successfully retrieved.

#### C# syntax

public Integer AgentWS.GetServiceAgents(WideString SessionId , Int32 ServiceId , out TWSAgentServiceInformationList AgentsInService);

# GetServiceCallSound

## Description

Retrieves the sound defined for a specific service to generate a warning on the agent's PC when receiving or answering a call.

# **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
In	SoundType	TWSSoundType	Indicates if the sound must be played when receiving or answering the contact call
In	ServiceId	Integer	Identifier of the service for which the sound should be retrieved
Out	SoundData	Binary	Content of the sound file
Out	LoopSound	Boolean	Indicates if it is necessary to repeat the warning sound until the agent answers the contact call. Only if the <i>SoundType</i> parameter is set to <i>DELIVERED</i>

### Return value

Integer. When the returned value is <0, the sound could not be retrieved. If the value is 0, the sound has been successfully retrieved.

### C# syntax

```
public Integer AgentWS.GetServiceCallSound(WideString SessionId ,
TWSSoundType SoundType , Int32 ServiceId , out Binary SoundData , out
Boolean LoopSound);
```

# GetServiceName

# Description

Retrieves the name of the service that the current contact belongs to.

#### **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
Out	ServiceName	WideString	Service name

## Return value

Integer. When the returned value is <0, the service name could not be retrieved. If the value is 0, the service name has been successfully retrieved.

### C# syntax

public Integer AgentWS.GetServiceName(WideString SessionId , out WideString ServiceName);

# GetServiceQCodes

## Description

Retrieves the list of qualification code groups and qualification codes (equal to or higher than 100) that belong to a specific group in a particular service.

### **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
In	ServiceId	Integer	Identifier of the service that the group belongs to
In	Group	Integer	Identifier of the qualification code group from which the list of groups and qualification codes will be retrieved. If the value if set to 0, the groups and qualification codes that don't belong to any group will be retrieved
Out	<i>QCodes</i>	<u>TWSQCodeList</u>	List of groups and qualification codes

## Return value

Integer. When the returned value is <0, the groups and qualification codes could not be retrieved. If the value is 0, the groups and qualification codes have been successfully retrieved.

# C# syntax

public Integer AgentWS.GetServiceQCodes(WideString SessionId , Int32 ServiceId , Int32 Group , out TWSQCodeList QCodes);

# GetSessionCallSound

## Description

Retrieves the general sound for the acoustic warning that the agent may hear when receiving a call outside any Presence service (an internal call, for instance).

### **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
Out	SoundData	Binary	Content of the sound file
Out	SoundLoop	Boolean	Indicates if it is necessary to repeat the warning sound until the agent answers the call

#### Return value

Integer. When the returned value is <0, the sound could not be retrieved. If the value is 0, the sound has been successfully retrieved.

### C# syntax

public Integer AgentWS.GetSessionCallSound(WideString SessionId , out Binary SoundData , out Boolean SoundLoop);

# **GetSessionInfo**

# Description

Retrieves the information about the agent work session.

# Parameters

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
Out	SessionInfo	<u>TWSSessionInfo</u>	Information about the agent work session

## Return value

Integer. When the returned value is <0, the work session information could not be retrieved. If the value is 0, the work session information has been successfully retrieved.

## C# syntax

public Integer AgentWS.GetSessionInfo(WideString SessionId , out TWSSessionInfo SessionInfo);

# GetSoftPhoneBooks

### Description

Retrieves the information about the entries of all phone books assigned to the agent, and the services that are available for placing direct calls to agents.

### **Parameters**

Flag	Parameter	Type	Description
In	SessionId	WideString	Session control identifier
Out	SoftphoneBooks	<u>TWSSoftPhoneInfo</u>	Phone books assigned to the agent and services that are available for placing direct calls to agents

## Return value

Integer. When the returned value is <0, the phone books and services could not be retrieved. If the value is 0, the phone books and services have been successfully retrieved.

### C# syntax

public Integer AgentWS.GetSoftPhoneBooks(WideString SessionId , out TWSSoftPhoneInfo SoftphoneBooks);

# GetStopReasons

# Description

Retrieves the stop reasons that can be selected by the agent when stopping the work session.

## **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
Out	Reasons	TWSReasonList	List of stop reasons

## Return value

Integer. When the returned value is <0, the stop reasons could not be retrieved. If the value is 0, the stop reasons have been successfully retrieved.

### C# syntax

public Integer AgentWS.GetStopReasons(WideString SessionId , out TWSReasonList Reasons);

# GetTemplate

## Description

Retrieves the body and fields of a mail template.

### **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
In	Id	Integer	Template identifier
Out	Data	<u>TWSMail</u>	Template body
Out	Fields	<u>TWSTemplateFieldList</u>	List of fields that have been added to the template

# Return value

Integer. When the returned value is <0, the template body and fields could not be retrieved. If the value is 0, the template body and fields have been successfully retrieved.

## C# syntax

public Integer AgentWS.GetTemplate(WideString SessionId , Int32 Id , out TWSMail Data , out TWSTemplateFieldList Fields);

# GetTemplateAttachment

### Description

Retrieves an attachment from a mail template.

# **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
In	TemplateId	Integer	Template identifier

In AttachmentId Integer Identifier of the attachment to be retrieved

from the template

Out FileData Binary Attachment content

### Return value

Integer. When the returned value is <0, the attachment could not be retrieved. If the value is 0, the attachment has been successfully retrieved.

## C# syntax

public Integer AgentWS.GetTemplateAttachment(WideString SessionId , Int32 TemplateId , Int32 AttachmentId , out Binary FileData);

# GetTemplateTreeItems

### Description

Retrieves the list of template groups and templates to which the agent has access to handle the current mail contact.

#### **Parameters**

Flag	Parameter	Type	Description
i iag	i di di lictoi	1 9 0 0	Description

In SessionId WideString Session control identifier

Out *Templates <u>TWSTemplateTreeItemList</u>* List of available groups and

templates that can be used to handle the current mail contact

### Return value

Integer. When the returned value is <0, the list of groups and templates could not be retrieved. If the value is 0, the list of groups and templates has been successfully retrieved.

### C# syntax

public Integer AgentWS.GetTemplateTreeItems(WideString SessionId , out TWSTemplateTreeItemList Templates);

# GetTimeZones

### Description

Retrieves the list of time zones that have been added to the outbound service that the current outbound contact belongs to.

### **Parameters**

Flag	Parameter	Type	Description

In SessionId WideString Session control identifier

Out *TimeZones <u>TWSTimeZoneList</u>* List of time zones added to the

outbound service

## Return value

Integer. When the returned value is <0, the time zones could not be retrieved. If the value is 0, the time zones have been successfully retrieved.

### C# syntax

public Integer AgentWS.GetTimeZone(WideString SessionId , out TWSTimeZoneList TimeZones);

# GetUnreachableOCodes

#### Description

Retrieves the list of unreachable qualification codes for a specific outbound service.

#### **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
In	ServiceId	Integer	Outbound service identifier
Out	<i>QCodes</i>	<u>TWSQCodeList</u>	List of unreachable qualification codes

#### Return value

Integer. When the returned value is <0, the unreachable qualification codes could not be retrieved. If the value is 0, the unreachable qualification codes have been successfully retrieved.

### C# syntax

public Integer AgentWS.GetUnreachableQCodes(WideString SessionId , Int32 ServiceId , out TWSQCodeList QCodes);

# GetWaitingCallMessageAlert

## Description

Retrieves the number of queued calls for the services which are connected to the agent session.

## **Parameters**

Flag Parameter Type Description

In SessionId WideString Session control

identifier

Out WaitingCallMessageAlert <u>TWSWaitingCallAlertList</u> List of service queued

calls

## Return value

Integer. When the returned value is <0, the service queued calls could not be retrieved. If the value is 0, the service queued calls have been successfully retrieved.

### C# syntax

public Integer AgentWS.GetWaitingCallMessageAlert(WideString SessionId ,
out TWSWaitingCallAlertList WaitingCallMessageAlert);

# HoldCall

# Description

Holds the active call.

#### **Parameters**

Flag Parameter Type Description

In Session/d WideString Session control identifier

## Return value

Integer. When the returned value is <0, the call could not be held. If the value is 0, the call has been successfully held.

### C# syntax

public Integer AgentWS.HoldCall(WideString SessionId);

# InitializeOutgoingMail

Deprecated. Refer to LoadMailOutInfo.

# InsertOutboundRecord

Deprecated. Refer to *InsertOutboundRecord2*.

# InsertOutboundRecord2

# Description

Inserts a new outbound record into a service.

# **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
In	ServiceId	Integer	ID of the service for which the record will be inserted
In	LoadId	Integer	ID of the load into which the record will be included
In	SourceId	Integer	This parameter provides a reference to the ID assigned to the customer in the system integrated with the Presence Suite. Valid values must range from 1 to 2147483647, both inclusive
In	Name	WideString	Contact name
In In	Name TimeZone	WideString WideString	Contact name  ID of the default time zone for the record. Only if the Enable support for time zones option is checked in the service.  The time zone must have been added to the service, otherwise an error occurs and the record is not inserted.  If this parameter contains an empty string, the service default time zone is assigned to the record

1: Record not scheduled (call as initial)

2: Record scheduled to a set

date and time

In Phone WideString

Contact phone number. If the value set for this parameter contains any blank spaces, they will be removed when inserting the record. When the Check phone numbers when inserting outbound record option is enabled for the service, the system will check whether the phone number is added to any of the enabled Do-Not-Call lists that are associated to that service

Phone Time Zone In

WideString

ID of the time zone for the contact phone number. Only if the Enable support for time zones option is checked in the service.

If the value of the *AutomaticTimeZoneDetection* parameter is set to TRUE, the time zone will be automatically assigned according to the contact phone prefix using the area codes defined in the system.

The time zone cannot be assigned to the phone if it is not added to the outbound service. In that case the time zone specified in the *TimeZone* parameter will be used

In **AlternativePhones**  WideString

Contact alternative phone numbers separated by commas. If the value set for this parameter contains any blank spaces, they will be removed when inserting the record. Only if the Enable alternative phone numbers option is checked in the service.

When the Check phone numbers when inserting outbound record option is enabled for the service, the system will check whether the phone numbers are added to any of the enabled Do-Not-Call

Presence Suite

Enghouse Interactive

lists that are associated to that service.

The number of phone nos. that will be processed depends on the value set for the Alternative phone numbers to manage option

In AlternativePhoneDescriptions WideString

IDs of the descriptions associated with each alternative phone number separated by commas. If the value set for this parameter contains any blank spaces, they will be removed when inserting the record. Only if the Enable alternative phone numbers option is checked in the service.

All the alternative phone numbers must have a valid phone description that has been added to the outbound service, otherwise an error occurs and the record is not inserted.

The number of phone nos. that will be processed depends on the value set for the Alternative phone numbers to manage option

In AlternativePhoneTimeZones WideString

IDs of the time zones associated with each alternative phone number separated by commas. Only if the Enable alternative phone numbers and the Enable support for time zones options are checked in the service.

If the value of the AutomaticTimeZoneDetection parameter is set to TRUE, the time zone will be automatically assigned according to each alternative phone prefix using the area codes defined in the system.

The time zone cannot be assigned to the phone if it is not added to the outbound service. In that case the time zone specified in the *TimeZone* parameter will be used. The number of phone nos. that will be processed depends on

			the value set for the Alternative phone numbers to manage option
In	ScheduleDate	DateTime	Date and time for the scheduling. Only if the <i>Status</i> parameter is set to 2. If the support for time zones is enabled for the service, the scheduled date and time will match the date and time of the time zone that is specified in the <i>PhoneTimeZone</i> parameter
In	CapturingAgent	Int64	Capturing agent login. Supports up to 18 digits. If the value is 0, the outbound record is not captured by any agent
In	Priority	Integer	Record priority
In	Comments	WideString	Record comments
In	CustomData1	WideString	First custom data value to be assigned to the record. This is useful to add business data. Supports up to 100 alphanumeric characters. Only if the Enable custom fields option is checked for the service in the Custom fields: Record tab
In	CustomData2	WideString	Second custom data value to be assigned to the record. This is useful to add business data. Supports up to 100 alphanumeric characters. Only if the Enable custom fields option is checked for the service in the Custom fields: Record tab
In	CustomData3	WideString	Third custom data value to be assigned to the record. This is useful to add business data. Supports up to 100 alphanumeric characters. Only if the Enable custom fields option is checked for the service in the Custom fields: Record tab
In	CallerId	WideString	Phone number that will be displayed at the destination of calls made to the record. Only

if the Enable outgoing calls identification option is checked in the service

In CallerName WideString Descriptive text that will be

displayed at the destination of calls made to the record. Only if the installed platform is OpenGate and the Enable outgoing calls identification option is checked in the service

In AutomaticTimeZoneDetection Boolean Indicates if the automatic time

zone detection will be performed according to the phone prefix and using the area codes defined in the system. Only if the Enable support for time zones and the Enable automatic time zones by phone prefix options are checked in the service

### Return value

Integer. When the returned value is <0, the outbound record could not be inserted. If the value is 0, the outbound record has been successfully inserted.

C# syntax

public Integer AgentWS.InsertOutboundRecord2(WideString SessionId , Int32
ServiceId , Int32 LoadId , Int32 SourceId , WideString Name , WideString
TimeZone , Int32 Status , WideString Phone , WideString PhoneTimeZone ,
WideString AlternativePhones , WideString AlternativePhoneDescriptions ,
WideString AlternativePhoneTimeZones , DateTime ScheduleDate , Int64
CapturingAgent , Int32 Priority , WideString Comments , WideString
CustomData1 , WideString CustomData2 , WideString CustomData3 , WideString
CallerId , WideString CallerName , Boolean AutomaticTimeZoneDetection);

# IsValidQCode

### Description

Indicates if a qualification code is valid for the service that the current contact belongs to.

#### **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
In	<i>QCode</i>	Integer	Qualification code that will be checked

Out *IsValid* Boolean Indicates if the qualification code is valid for the service

## Return value

Integer. When the returned value is <0, the qualification code could not be checked. If the value is 0, the qualification code has been successfully checked.

#### C# syntax

public Integer AgentWS.IsValidQCode(WideString SessionId , Int32 QCode ,
out Boolean IsValid);

# LoadMailOutInfo

## Description

Loads the data associated to a mailbox (templates, predefined texts, and address books) in order to create an outgoing e-mail.

## **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	AnsiString	Session control identifier
In	MailBoxId	Integer	Mailbox ID of the outgoing e-mail that you want to create

#### Return value

Integer. When the returned value is <0, the outgoing e-mail cannot be initialized. If the value is 0, the outgoing e-mail is successfully initialized.

## C# syntax

public Integer AgentWS.LoadMailOutInfo(AnsiString SessionId , Int32 MailBoxId);

# Login

#### Description

Performs the agent login to the Presence system and to the station in the ACD system.

## **Parameters**

Flag	Parameter	Туре	Description
In	AgentLogin	Int64	Agent identifier
InOut	SessionId	WideString	Session control identifier

In	Password	WideString	Agent's password
In	Station	Int64	ACD station where the agent will be logged in
In	ClearCalls	Boolean	Indicates if all calls will be hung up automatically at the physical station before the agent is logged in
In	AgentVersion	WideString	Agent client version
In	LocalIP	WideString	Agent client IP address
In	Language	WideString	Language to launch localizable elements. Supported values are 'EN' (English) and 'ES' (Spanish)
In	LCIDScripting	Integer	Locale culture identifier used to specify Web Scripting locale settings

## Return value

Integer. When the returned value is <0, the agent could not be logged in. If the value is 0, the agent has been successfully logged in.

# C# syntax

public Integer AgentWS.Login(Int64 AgentLogin , ref WideString SessionId ,
WideString Password , Int64 Station , Boolean ClearCalls , WideString
AgentVersion , WideString LocalIP , WideString Language , Int32
LCIDScripting);

# Logout

### Description

Logs the agent out of the Presence system and of the station in the ACD system.

# **Parameters**

Flag	Parameter	Type	Description
In	SessionId	WideString	Session control identifier
In	ReasonCode	Integer	Logout reason code

### Return value

Integer. When the returned value is <0, the agent could not be logged out. If the value is 0, the agent has been successfully logged out.

### C# syntax

public Integer AgentWS.Logout(WideString SessionId , Int32 ReasonCode);

# MakeCall

## Description

Makes a call to the specified phone number.

## **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
In	Phone	WideString	Phone number to be called
In	PhoneType	<u>TWSPhoneType</u>	Indicates if the specified phone number is an external phone number or an internal extension
In	AppData	WideString	Additional information associated with the outgoing call. Only if the Store outgoing calls of agent option is checked for the login

# Return value

Integer. When the returned value is <0, the call could not be made. If the value is 0, the call has been successfully made.

# C# syntax

public Integer AgentWS.MakeCall(WideString SessionId , WideString Phone ,
TWSPhoneType PhoneType , WideString AppData);

# MakeConsultationCall

# Description

Makes a consultation call to the specified phone number.

# **Parameters**

Flag	Parameter	Type	Description
In	SessionId	WideString	Session control identifier
In	CalledPhone	WideString	Phone number to be called
In	PhoneType	<u>TWSPhoneType</u>	Indicates if the specified phone number is an external phone number or an internal extension

In AppData WideString Additional information associated with

the outgoing call. Only if the Store outgoing calls of agent option is

checked for the login

#### Return value

Integer. When the returned value is <0, the consultation call could not be made. If the value is 0, the consultation call has been successfully made.

## C# syntax

public Integer AgentWS.MakeConsultationCall(WideString SessionId ,
WideString CalledPhone , TWSPhoneType PhoneType , WideString AppData);

# MakeContactCall

# Description

Used only for outbound services in preview mode to generate the call to the contact.

### **Parameters**

Flag	Parameter	Type	Description
In	SessionId	WideString	Session control identifier
In	AppData	WideString	Additional information associated with the outgoing call. Only if the Store outgoing calls of agent option is checked for the login

### Return value

Integer. When the returned value is <0, the call could not be generated. If the value is 0, the call has been successfully generated.

# C# syntax

public Integer AgentWS.MakeContactCall(WideString SessionId , WideString AppData);

# MakeDirectAgentCall

## Description

Places a direct call to an agent through an inbound service for which direct calls to agents are enabled.

### **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
In	ServiceId	Integer	Identifier of the inbound service that supports direct calls
In	Login	Int64	Login of the agent who will receive the call
In	AppData	WideString	Additional information associated with the outgoing call. Only if the Store outgoing calls of agent option is checked for the login

### Return value

Integer. When the returned value is <0, the direct call could not be placed. If the value is 0, the direct call has been successfully placed.

#### C# syntax

public Integer AgentWS.MakeDirectAgentCall(WideString SessionId , Int32 ServiceId , Int64 Login , WideString AppData);

# MakeDirectAgentConsultationCall

# Description

Places a direct consultation call to an agent through an inbound service for which direct calls to agents are enabled.

### **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
In	ServiceId	Integer	Identifier of the inbound service that supports direct calls
In	Login	Int64	Login of the agent to receive the consultation call
In	AppData	WideString	Additional information associated with the outgoing call. Only if the Store outgoing calls of agent option is checked for the login

## Return value

Integer. When the returned value is <0, the direct consultation call could not be placed. If the value is 0, the direct consultation call has been successfully placed.

### C# syntax

public Integer AgentWS.MakeDirectAgentConsultationCall(WideString SessionId , Int32 ServiceId , Int64 Login , WideString AppData);

# MakeServiceCall

## Description

Makes an external call to the specified phone number using the phone prefix defined for the outbound service that the current contact belongs to. If there is no phone prefix defined for the service, the one defined in the Presence Server Configuration program in the Prefix for outgoing calls parameter is used.

#### **Parameters**

Flag	Parameter	Type	Description
In	SessionId	WideString	Session control identifier
In	Phone	WideString	Phone number to be called
In	AppData	WideString	Additional information associated with the outgoing call. Only if the Store outgoing calls of agent option is checked for the login

### Return value

Integer. When the returned value is <0, the service call could not be requested. If the value is 0, the service call has been successfully requested.

# C# syntax

public Integer AgentWS.MakeServiceCall(WideString SessionId , WideString Phone , WideString AppData);

# NoticesAcknowledge

#### Description

Confirms that the existence of new notices has been acknowledged so that the event indicating that there are new notices (*AGM\_MESSAGEALERT\_EVENT*) is not resent until there are other new notices.

#### **Parameters**

Flag	Parameter	Type	Description
In	SessionId	WideString	Session control identifier

## Return value

Integer. When the returned value is <0, the notices could not be acknowledged. If the value is 0, the notices have been successfully acknowledged.

## C# syntax

public Integer AgentWS.NoticesAcknowledge(WideString SessionId);

# **PauseSkills**

### Description

Pauses the specified skills to temporarily stop receiving contacts that are queued to them. Only for the OpenGate platform.

### **Parameters**

Flag	Parameter	Type	Description
In	SessionId	WideString	Session control identifier
In	Skills	TWSSkillExtensionList	List of skills to be paused
Out	ResultList	TWSSkillActionResultList	List of results of the pause actions performed on the skills

### Return value

Integer. When the returned value is <0, the skills could not be paused. If the value is 0, the skills have been successfully paused.

## C# syntax

```
public Integer AgentWS.PauseSkills(WideString SessionId ,
TWSSkillExtensionList Skills , out TWSSkillActionResultList ResultList);
```

# RecordPause

### Description

Pauses the recording of the active call.

# **Parameters**

Flag Parameter Type Description

In SessionId WideString Session control identifier

### Return value

Integer. When the returned value is <0, the recording could not be paused. If the value is 0, the recording has been successfully paused.

#### C# syntax

public Integer AgentWS.RecordPause(WideString SessionId);

# RecordStart

## Description

Starts or resumes the recording of the active call.

### **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
In	<i>AppData</i>	WideString	User data attached to the recording
Out	RecordId	Integer	Resulting recording identifier

### Return value

Integer. When the returned value is <0, the recording could not be started or resumed. If the value is 0, the recording has been successfully started or resumed.

### C# syntax

public Integer AgentWS.RecordStart(WideString SessionId , WideString AppData , out Int32 RecordId);

# RecordStop

# Description

Stops the recording of the active call.

#### **Parameters**

Flag Parameter Type Description

In SessionId WideString Session control identifier

# Return value

Integer. When the returned value is <0, the recording could not be stopped. If the value is 0, the recording has been successfully stopped.

## C# syntax

public Integer AgentWS.RecordStop(WideString SessionId);

# RequestOutboundACDCall

Deprecated. Refer to RequestOutboundACDCall2.

# RequestOutboundACDCall2

# Description

Requests a new manual outbound ACD contact for the agent in a particular outbound service.

### **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
In	ServiceId	Integer	ID of the outbound service that the new contact will be requested from. The support for manual outbound ACD calls must be enabled in the service
In	SourceId	Integer	This parameter provides a reference to the ID assigned to the customer in the system integrated with the Presence Suite. Valid values must range from 1 to 2147483647, both inclusive.  If the value for this parameter is set to 0, the source ID will be automatically assigned by the Presence system
In	Phone	WideString	Contact phone number. When the Check phone numbers when inserting outbound record option is enabled for the service, the system will check whether the phone number is added to any of the enabled Do-Not-Call lists that are associated to that service
In	Name	WideString	Contact name
In	Scheduled	Boolean	Indicates if the call is captured for the agent at the scheduled date and time. If the value is set to FALSE, the manual outbound ACD call is made immediately
In	ScheduleDate	DateTime	Date and time to schedule the contact. Only if the <i>Scheduled</i> parameter is set to TRUE.

Specifying a date and time earlier than the current

date and time in Presence Server causes the manual outbound ACD call to be made immediately. Otherwise the call is captured for the agent at the scheduled date and time

In Comments WideString Contact comments

### Return value

Integer. When the returned value is <0, the request could not be made. If the value is 0, the request has been successfully made.

### C# syntax

public Integer AgentWS.RequestOutboundACDCall2(WideString SessionId ,
Int32 ServiceId , Int32 SourceId , WideString Phone , WideString Name ,
Boolean Scheduled , DateTime ScheduleDate , WideString Comments);

# ResumeSkills

## Description

Resumes the specified skills to start receiving contacts again that are queued to them. Only for the OpenGate platform.

# **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
In	Skills	TWSSkillExtensionList	List of skills to be resumed
Out	ResultList	TWSSkillActionResultList	List of results of the resume actions performed on the skills

# Return value

Integer. When the returned value is <0, the skills could not be resumed. If the value is 0, the skills have been successfully resumed.

### C# syntax

```
public Integer AgentWS.ResumeSkills(WideString SessionId ,
TWSSkillExtensionList Skills , out TWSSkillActionResultList ResultList);
```

# RetrieveCall

## Description

Retrieves a held call.

#### **Parameters**

In SessionId WideString Session control identifier

In CallId Integer Identifier of the call to be retrieved

### Return value

Integer. When the returned value is <0, the call could not be retrieved. If the value is 0, the call has been successfully retrieved.

#### C# syntax

public Integer AgentWS.RetrieveCall(WideString SessionId , Int32 CallId);

# RetrieveMailSuspended

### Description

Retrieves an e-mail that was suspended by the agent.

### **Parameters**

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

In SessionId WideString Session control identifier

In Mailld Integer Identifier of the suspended e-mail to be

retrieved

### Return value

Integer. When the returned value is <0, the suspended e-mail could not be retrieved. If the value is 0, the suspended e-mail has been successfully retrieved.

# C# syntax

public Integer AgentWS.RetrieveMailSuspended(WideString SessionId , Int32
MailId);

# SaveOutgoingMailAsPartial

### Description

Saves the outgoing e-mail content and the attachments added so far through the <u>AddOutgoingMailAttachment</u> method as a partial reply to the incoming e-mail.

### **Parameters**

Flag Parameter Type Descrip	ption
-----------------------------	-------

In SessionId WideString Session control identifier

In *PartialMail TWSOutgoingMail* Outgoing e-mail content

### Return value

Integer. When the returned value is <0, the outgoing e-mail could not be saved. If the value is 0, the outgoing e-mail has been successfully saved.

# C# syntax

public Integer AgentWS.SaveOutgoingMailAsPartial(WideString SessionId ,
out TWSOutgoingMail PartialMail);

# SelectLine

## Description

Sets the call line as active in the phone extension.

### **Parameters**

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

In SessionId WideString Session control identifier

In Calld Integer Identifier of the call associated with the line

to be set as active

## Return value

Integer. When the returned value is <0, the call line could not be set as active. If the value is 0, the call line has been successfully set as active.

# C# syntax

public Integer AgentWS.SelectLine(WideString SessionId , Int32 CallId);

# SendDTMFTones

#### Description

Sends DTMF (Dual Tone Multifrequency) tones to the current active call.

### **Parameters**

Flag Parameter Type Description

In	SessionId	WideString	Session control identifier
In	Tones	WideString	String containing the tones to be sent. This string may include numeric values from 0 to 9 and the characters # and *. The maximum length of this string is of 32 characters (tones). If a string longer than 32 characters is specified, the first 32 characters will be sent and the remaining characters will be ignored

### Return value

Integer. When the returned value is <0, the DTMF tones could not be sent. If the value is 0, the DTMF tones have been successfully sent.

### C# syntax

public Integer AgentWS.SendDTMFTones(WideString SessionId , WideString Tones);

# SendOutgoingMail

### Description

Sends the outgoing e-mail to reply to the mail contact that is being handled. Before calling this method, the outgoing e-mail must have been defined using the *InitializeOutgoingMail*, *SetOutgoingMail* and *AddOutgoingMailAttachment* methods.

# **Parameters**

Flag	Parameter	Type	Description
In	SessionId	WideString	Session control identifie

# Return value

Integer. When the returned value is <0, the outgoing e-mail could not be sent. If the value is 0, the outgoing e-mail has been successfully sent.

# C# syntax

public Integer AgentWS.SendOutgoingMail(WideString SessionId);

# SendOutgoingMailHistory

#### Description

Sends the outgoing e-mail to reply to an incoming e-mail from the mail history. Before calling this method, the outgoing e-mail must have been defined using the <a href="mailto:linetaling.com/mailto

### **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
In	MailboxId	Integer	Identifier of the mailbox that the incoming e-mail from the mail history belongs to
In	Mailld	Integer	Identifier of the incoming e-mail from the mail history

### Return value

Integer. When the returned value is <0, the outgoing e-mail could not be sent. If the value is 0, the outgoing e-mail has been successfully sent.

### C# syntax

public Integer AgentWS.SendOutgoingMailHistory(WideString SessionId ,
Int64 MailBoxId , Int64 InboundMailId);

# SetAlternativePhones

## Description

Sets the alternative phone numbers of the current contact.

### **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
In	Phones	<u>TWSOutboundPhoneList</u>	List of alternative phone numbers of the current contact

### Return value

Integer. When the returned value is <0, the alternative phone numbers could not be set. If the value is 0, the alternative phone numbers have been successfully set.

## C# syntax

public Integer AgentWS.SetAlternativePhones(WideString SessionId ,
TWSOutboundPhoneList Phones);

# SetAppData

## Description

Defines additional information to be attached to the contact information. It is useful to add business data. This additional information is stored in the APPDATA field of the PCO\_INBOUNDLOG table (refer to the *Presence Repository* manual). For inbound services only.

#### **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
In	AppData	WideString	Additional information to be attached to the contact information. Supports up to 30 characters

### Return value

Integer. When the returned value is <0, the information could not be set. If the value is 0, the information has been successfully set.

### C# syntax

public Integer AgentWS.SetAppData(WideString SessionId , WideString AppData);

# SetContactData1

### Description

Sets the first custom data value for the outbound or inbound contact that is being handled. This is useful to add business data. Supports up to 100 alphanumeric characters.

Only if the Enable custom fields option is checked for the service in the Custom fields: Contact tab (outbound services) or the Custom fields tab (inbound services).

### **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
In	ContactData	WideString	First custom data value to be assigned to the contact

### Return value

Integer. When the returned value is <0, the custom data value could not be set. If the value is 0, the custom data value has been successfully set.

## C# syntax

public Integer AgentWS.SetContactDatal(WideString SessionId , WideString ContactData);

# SetContactData2

## Description

Sets the second custom data value for the outbound or inbound contact that is being handled. This is useful to add business data. Supports up to 100 alphanumeric characters.

Only if the Enable custom fields option is checked for the service in the Custom fields: Contact tab (outbound services) or the Custom fields tab (inbound services).

#### **Parameters**

Flag	Parameter	Type	Description
In	SessionId	WideString	Session control identifier
In	ContactData	WideString	Second custom data value to be assigned to the contact

#### Return value

Integer. When the returned value is <0, the custom data value could not be set. If the value is 0, the custom data value has been successfully set.

### C# syntax

public Integer AgentWS.SetContactData2(WideString SessionId , WideString ContactData);

# SetContactData3

### Description

Sets the third custom data value for the outbound or inbound contact that is being handled. This is useful to add business data. Supports up to 100 alphanumeric characters

Only if the Enable custom fields option is checked for the service in the Custom fields: Contact tab (outbound services) or the Custom fields tab (inbound services).

#### **Parameters**

Flag	Parameter	Type	Description
In	SessionId	WideString	Session control identifier
In	ContactData	WideString	Third custom data value to be assigned to the contact

### Return value

Integer. When the returned value is <0, the custom data value could not be set. If the value is 0, the custom data value has been successfully set.

### C# syntax

public Integer AgentWS.SetContactData3(WideString SessionId , WideString ContactData);

# SetCustomData1

## Description

Sets the first custom data value for the outbound record associated to the contact that is being handled. This is useful to add business data. Supports up to 100 alphanumeric characters. For outbound services only.

Only if the Enable custom fields option is checked for the service in the Custom fields: Record tab.

#### **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
In	CustomData	WideString	First custom data value to be assigned to the outbound record

### Return value

Integer. When the returned value is <0, the custom data value could not be set. If the value is 0, the custom data value has been successfully set.

# C# syntax

public Integer AgentWS.SetCustomDatal(WideString SessionId , WideString CustomData);

# SetCustomData2

# Description

Sets the second custom data value for the outbound record associated to the contact that is being handled. This is useful to add business data. Supports up to 100 alphanumeric characters. For outbound services only.

Only if the Enable custom fields option is checked for the service in the Custom fields: Record tab.

### **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier

In CustomData WideString Second custom data value to be assigned to

the outbound record

### Return value

Integer. When the returned value is <0, the custom data value could not be set. If the value is 0, the custom data value has been successfully set.

### C# syntax

public Integer AgentWS.SetCustomData2(WideString SessionId , WideString CustomData);

# SetCustomData3

## Description

Sets the third custom data value for the outbound record associated to the contact that is being handled. This is useful to add business data. Supports up to 100 alphanumeric characters. For outbound services only.

Only if the Enable custom fields option is checked for the service in the Custom fields: Record tab.

### **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
In	CustomData	WideString	Third custom data value to be assigned to the outbound record

### Return value

Integer. When the returned value is <0, the custom data value could not be set. If the value is 0, the custom data value has been successfully set.

# C# syntax

public Integer AgentWS.SetCustomData3(WideString SessionId , WideString CustomData);

# SetNoticeRead

### Description

Sets a notice as read.

### **Parameters**

Flag Parameter Type Description

In	SessionId	WideString	Session control identifier
----	-----------	------------	----------------------------

In *NoticeId* Integer Identifier of the notice to mark as read

### Return value

Integer. When the returned value is <0, the notice could not be set as read. If the value is 0, the notice has been successfully set as read.

# C# syntax

```
public Integer AgentWS.SetNoticeRead(WideString SessionId , Int32
NoticeId);
```

# SetOutgoingMail

## Description

Sets the whole outgoing e-mail content (i.e., subject, sender, recipient, body, etc.), except for the attachments which should be added using the <u>AddOutgoingMailAttachment</u> method. The <u>SendOutgoingMail</u> method can be called in order to send the outgoing e-mail.

#### **Parameters**

Flag	Paramet	er Type	Description
i iag	i di di lict	or rypc	Description

In SessionId WideString Session control identifier

In *Mail* TWSOutgoingMail Outgoing e-mail content

# Return value

Integer. When the returned value is <0, the outgoing e-mail could not be set. If the value is 0, the outgoing e-mail has been successfully set.

### C# syntax

```
public Integer AgentWS.SetOutgoingMail(WideString SessionId ,
TWSOutgoingMail Mail);
```

# **SetQCode**

# Description

Sets the qualification code for the current contact. This is the code (Sale, Not interested, etc.) which the contact will be qualified with.

### **Parameters**

Flag Parameter Type Description

In SessionId WideString Session control identifier

In *QCode* Integer Qualification code

## Return value

Integer. When the returned value is <0, the qualification code could not be set. If the value is 0, the qualification code has been successfully set.

## C# syntax

public Integer AgentWS.SetQCode(WideString SessionId , Int32 QCode);

# SetOInfo

Deprecated. Refer to SetOInfo3.

# SetOInfo2

Deprecated. Refer to SetOInfo3.

# SetOInfo3

# Description

Sets the information about the qualification of the current contact.

### **Parameters**

Flag Parameter Type Description

In SessionId WideString Session control identifier

In *QInfo* <u>TWSQInfo3</u> Qualification information

## Return value

Integer. When the returned value is <0, the qualification information could not be set. If the value is 0, the qualification information has been successfully set.

# C# syntax

public Integer AgentWS.SetQInfo3(WideString SessionId , TWSQInfo3 QInfo);

# SetSourceId

# Description

Sets the source ID (customer identifier) of an inbound contact.

## **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
In	SourceId	Integer	This parameter provides a reference to the ID assigned to the customer in the system integrated with the Presence Suite. For example, it allows to link the contact data stored in the Presence repository to the record data stored by an integrated application. The value of this field must range from 1 to 2147483647, both inclusive

### Return value

Integer. When the returned value is <0, the source ID could not be set. If the value is 0, the source ID has been successfully set.

### C# syntax

public Integer AgentWS.SetSourceId(WideString SessionId , Int32 SourceId);

# SingleStepTransferCall

# Description

Transfers the active call to the specified phone number in a single step.

### **Parameters**

Flag	Parameter	Type	Description
In	SessionId	WideString	Session control identifier
In	Called	WideString	Phone number to which the active call is to be transferred
In	PhoneType	<u>TWSPhoneType</u>	Indicates if the specified phone number is an external phone number or an internal extension

#### Return value

Integer. When the returned value is <0, the call could not be transferred. If the value is 0, the call has been successfully transferred.

# C# syntax

public Integer AgentWS.SingleStepTransferCall(WideString SessionId ,
WideString Called , TWSPhoneType PhoneType);

# Single Step Transfer Direct Agent Call

## Description

Transfers the active call to an agent in a single step through an inbound service for which direct calls to agents are enabled.

### **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
In	ServiceId	Int64	Identifier of the inbound service to which the active call is to be transferred
In	Login	Int64	Agent login to which the active call is to be transferred

# Return value

Integer. When the returned value is <0, the active call could not be transferred. If the value is 0, the active call has been successfully transferred.

## C# syntax

public Integer AgentWS.SingleStepTransferDirectAgentCall(WideString SessionId , Int64 ServiceId , Int64 Login);

# **StartSession**

# Description

Starts the agent work session.

### **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier

### Return value

Integer. When the returned value is <0, the work session could not be started. If the value is 0, the work session has been successfully started.

```
C# syntax
```

```
public Integer AgentWS.StartSession(WideString SessionId);
```

# **StopSession**

# Description

Stops the agent work session.

#### **Parameters**

Flag	Parameter	Туре	Description
In	SessionId	WideString	Session control identifier
In	ReasonCode	Integer	Stop reason code

### Return value

Integer. When the returned value is <0, the work session could not be stopped. If the value is 0, the work session has been successfully stopped.

## C# syntax

```
public Integer AgentWS.StopSession(WideString SessionId , Int32
ReasonCode);
```

# Sum

## Description

Test method: Adds two integer values and returns the result of the addition.

### **Parameters**

Flag	Parameter	Туре	Description
In	А	Integer	First integer value
In	В	Integer	Second integer value

# Return value

Integer

## C# syntax

public Integer AgentWS.Sum(Int32 A , Int32 B);

# **TransferCall**

## Description

Transfers the call that was placed on hold when generating the current active call. The held call is transferred to the recipient of the active call.

### **Parameters**

Flag Parameter Ty	pe Description
-------------------	----------------

In SessionId WideString Session control identifier

#### Return value

Integer. When the returned value is <0, the call could not be transferred. If the value is 0, the call has been successfully transferred.

#### C# syntax

public Integer AgentWS.TransferCall(WideString SessionId);

# TransferSurveyCall

# Description

Transfers the current contact to a particular survey.

#### **Parameters**

Flag	Parameter	Type	Description
In	SessionId	WideString	Session control identifier
In	SurveyVDN	Int64	Survey VDN/service extension to transfer the contact to. It must match one of the VDNs/service extensions listed in the <i>Surveys</i> field of the <i>TWS/nboundContact</i> (for inbound contacts) or the <i>TWSOutboundContact</i> (for outbound contacts) type

#### Return value

Integer. When the returned value is <0, the contact could not be transferred to the survey. If the value is 0, the contact has been successfully transferred to the survey.

### C# syntax

public Integer AgentWS.TransferSurveyCall(WideString SessionId , Int65 SurveyVDN);

# UnblockWaitForEvent

# Description

Unblocks the WaitForEvent method.

## **Parameters**

Flag	Parameter	Type	Description

In SessionId WideString Session control identifier

### Return value

Integer. When the returned value is <0, the <u>WaitForEvent</u> method could not be unblocked. If the value is 0, the <u>WaitForEvent</u> method has been successfully unblocked.

# C# syntax

public Integer AgentWS.UnblockWaitForEvent(WideString SessionId);

# UnloadOutboundRecord

## Description

Unloads an outbound record from a service.

### **Parameters**

Flag	Parameter	Type	Description
In	SessionId	WideString	Session control identifier
In	ServiceId	Integer	Identifier of the service that the record will be unloaded from
In	LoadId	Integer	Identifier of the load where the record will be searched for. If the value is set to 0, the outbound record will be searched through all loads to unload it
In	SourceId	Integer	This parameter provides a reference to the ID assigned to the customer in the system integrated with the Presence Suite

# Return value

Integer. When the returned value is <0, the outbound record could not be unloaded. If the value is  $\ge 0$ , the outbound record has been successfully unloaded.

## C# syntax

public Integer AgentWS.UnloadOutboundRecord(WideString SessionId , Int32 ServiceId , Int32 LoadId , Int32 SourceId);

# WaitForEvent

## Description

Waits for a new agent session event. If there are pending events, it will return TRUE instantly. Otherwise, it will wait for events during 90 seconds (by default).

### **Parameters**

Flag Parameter Type Description

In SessionId WideString Session control identifier

## Return value

Integer. When the returned value is <0, there are no events or an error has occurred. If the value is 0, there are new events. Refer to the <u>GetEvent</u> method to read them.

# C# syntax

public Integer AgentWS.WaitForEvent(WideString SessionId);

# Chapter

# **Appendices**



# **Troubleshooting**

This section describes the most common errors that can occur when executing any of the methods supported by the Presence Agent Web Services.

 "The server committed a protocol violation. Section=ResponseHeader Detail=Header name is invalid."



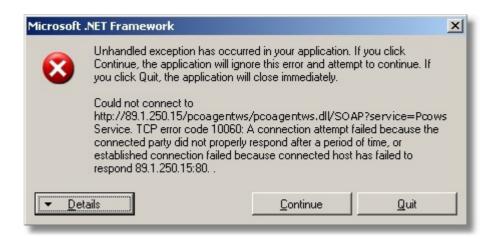
This error is generated when the unsafe header parsing is not enabled. Microsoft sets this value to false by default. Therefore, this error becomes more common if you are using a Microsoft programming language (for example, C# or Visual Basic).

To fix this issue, you must change the configuration file (.config) of your application to enable the unsafe header parsing. To do this, simply add the following lines to the end of the configuration file of your application:

```
<system.net>
  <settings>
     <httpWebRequest useUnsafeHeaderParsing="true" />
  </settings>
</system.net>
```

As a result, the configuration file would look like this:

2. "Could not connect to <URL>. TCP error code 10060: A connection attempt failed because the connected party..."



This error is generated when attempting to connect to the Presence Agent Web Services using the specified <URL>.

To fix this issue, you must check that the specified URL is correct and that the Presence Agent Web Services are installed and started on the machine that the URL points to. Depending on your programming language, this URL can be found in the (.config) configuration file of the application.

3. "The remote server returned an unexpected response: (405) Method not allowed."



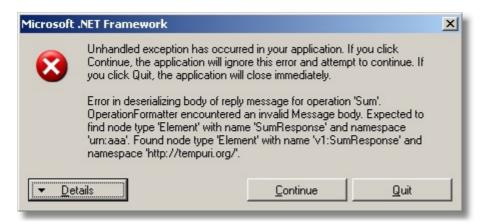
This error is generated when trying to execute a method whose reference is not found in the Presence Agent Web Services. It could also be because the specified <URL> is incorrect, therefore, the referenced method cannot be found. This error is generated when attempting to connect to the Presence Agent Web Services using the specified <URL>.

To fix this issue, you must check that the specified URL is correct and that the Presence Agent Web Services are installed and started on the machine that the URL points to. Depending on your programming language, this URL can be found in the (.config) configuration file of the application.

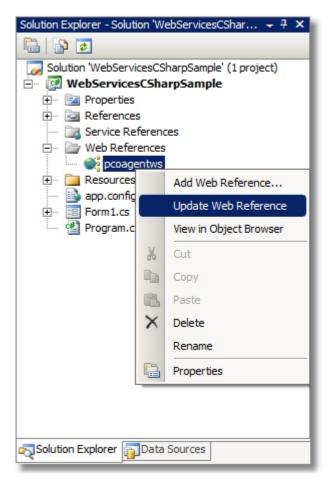
# 4. Other errors

For any other errors, the problem may be that a change made to the definition of the methods of the Presence Agent Web Services is not compatible with the current definition of the integrated application. While always trying to maintain the compatibility in the new versions of the Presence Agent Web Services, this is not always possible and the error may be due to this type of change.

The most common error message in this case is "Error in deserializing body of...", but other messages may also be shown.



To fix this issue, you must update the reference to the Presence Agent Web Services. This will cause the integrated application to get the definition of the methods again. To do this, open the application integrated with the programming language (C# in this example), then right-click on the reference for the Presence Agent Web Services and select the Update Service Reference option.



Once the updating process is complete, recompile and run the application to verify that the issue has been fixed.

# **Error Codes**

The most common error codes that might be returned by an operation of the Presence Agent Web Services are as follows:

Error code	Description
0	The operation has been successfully completed
-3	The request cannot be completed because the timeout value has been exceeded
-5	There are no available events
-16777216	The request to a server of the Presence Suite cannot be completed because the timeout value has been exceeded
-16777217	Unable to connect to a server of the Presence Suite
-16777222	The TCP/IP connection is not open
-16777224	This version of the Presence Agent Web Services is not supported by a server of the Presence Suite
-33554432	The service does not exist
-33554435	The service is disabled
-33554440	The service is paused
-33554442	The outbound service configuration does not support the operation
-33554444	The outbound service is out of service calling hours
-33554445	The outbound service does not exist
-33619968	An agent with the same login is already connected
-33619969	No agent is connected with this login
-33619970	The agent is already working
-33619971	The agent is not working
-33619972	The agent is not working in the service for which the request has been made
-33619973	The agent is not connected to any service
-33619975	Unable to perform the agent login to the station in the ACD system
-33619976	Unable to perform the agent login to the Presence system (the login or password is not valid)
-33685506	The qualification code does not exist for the service

-33685510	An error has occurred when generating the preview call
-33685511	There are no available phantom extensions to generate a preview call
-33685512	An error has occurred when inserting the outbound record into the database
-33685513	The outbound service load does not exist
-33685514	The current outbound contact does not exist in Presence Server
-33685515	The current inbound contact does not exist in Presence Server
-33685516	The phone number has been found in a Do-Not-Call list of the service
-33685517	The outbound service load is disabled
-33685518	The phone number does not contain any digits
-33685520	The default time zone for the outbound record is not added to the service
-33685521	At least one of the phone number descriptions is incorrect or is not added to the service
-33685522	Maximum number of pending requests that insert a new outbound record has been exceeded
-33751040	There are not available outbound records to be handled
-33816576	No data are associated to the contact with that key
-34013184	The maximum number of Presence Agent licenses has been exceeded
-34078722	An internal error has occurred in Presence Server when processing the operation
-34144256	Maximum number of connected agents has been exceeded
-34144257	Maximum number of additional data that can be attached to the contact has been exceeded
-34144516	Maximum number of pending manual outbound ACD calls by agent has been exceeded
-50331655	The call is already being recorded
-50331656	The recording does not exist
-50331657	The call to be recorded has not been found in the Presence Recording call list
-50331658	The call to be recorded has not been answered yet
-50331659	There are no active calls to record in the extension
-50331662	Unable to pause the recording

-50331663	Unable to resume the recording
-50331664	Unable to stop the recording
-50331666	The recording is already started
-50331667	The recording is already paused
-50331668	The recording is already stopped
-50593797	An internal error has occurred in Presence Recording when processing the operation
-67174402	The e-mail processing has already been finished
-67174405	The agent already has a suspended e-mail in progress
-67174409	The specified e-mail is not suspended
-150994945	An internal error has occurred in the Presence Agent Web Services when processing the operation
-150994946	An internal error has occurred in the Presence Agent Web Services when processing the operation
-150994947	The agent session cannot be started
-150994948	Unable to connect to the CTI server
-150994949	Unable to connect to the database
-150994950	The version of the database connection driver is not valid
-150994952	This version of the Presence Agent Web Services is not compatible with the Presence Server version
-150994953	Maximum number of agents connected to the same server has been exceeded
-150995199	A request has been made to perform an operation on an element that does not exist
-151060481	The agent session state is not valid
-151060482	The operation is not supported for this service type
-151060483	There is a call in progress in the agent extension
-151060484	An error has occurred when performing the agent login to the Presence system or the station in the ACD system
-151060485	An error has occurred when performing the agent logout of the Presence system or the station in the ACD system
-151060486	The integrated application does not allow the contact to be finished

-151060487	The minimum after-call work time has not been reached
-151060488	The contact cannot be finished because there are calls in the agent extension (all calls must be hung up)
-151060489	The agent session control identifier is not valid
-151060512	The old password is not correct. Passwords are case sensitive
-151060513	The new password cannot be empty
-151060514	The new password confirmation cannot be empty
-151060515	The new password and the new password confirmation do not match
-151060516	The password supplied does not meet the minimum complexity requirements. Please select another password that meets the following criteria: has at least the minimum length specified for strong passwords; is different from the latest password entered; does not contain the agent login number; contains at least three of the following four character groups: English uppercase characters (A-Z), English lowercase characters (a-z), numerals (0-9) and non-alphabetic characters (such as !, \$, #, %)
-151060735	The agent session does not exist
-151126017	The agent state is not valid
-151126018	The qualification code is not valid
-151126020	The scheduled date is out of calling hours (for a service or time zone)
-151126021	The group identifier does not exist
-151126022	The identifier does not correspond to any group
-151126023	The service is paused
-151126024	There are not available outbound records to be handled
-151126025	The agent cannot be connected to the service
-151126026	The agent cannot be connected to any service
-151126027	The contact has not been qualified
-151126029	The recording on demand is not enabled for the service
-151126030	The scheduled date for the record is out of the date limit set for the service
-151126031	The outbound service is out of service calling hours
-151126032	The phone number has been found in a Do-Not-Call list of the service
-151126034	The maximum size for an e-mail has been exceeded

-151126271	The service does not exist
-151191554	An error has occurred when changing the agent state
-151191555	The call has not been found
-151191556	An error has occurred when launching the CTI monitor in the agent extension
-151191557	Unable to place the call on hold
-151191558	The state is not valid for the call
-151191559	There are no active calls in the agent extension
-151191560	The agent extension has not been defined (check the agent seat configuration)
-151191561	Unable to hang up all calls (one of the calls might be ringing)
-166789121	The connection to Presence Server has been lost
-166789122	The connection to the CTI server has been lost
-166789123	The connection to the Presence Repository database has been lost
-166789124	The login has been disconnected from the extension by the ACD (e.g. because the extension handset has been hung up and the current agent configuration does not allow it)
-166789125	The connection to the CTI server node has been lost. Only for the OpenGate platform
-166789126	The connection to the Presence Agent Web Services has been lost
-268435457	An error has occurred in Presence OpenGate Proxy Server when processing the operation
-268435458	An internal error has occurred in Presence OpenGate Proxy Server when processing the operation
-268435462	The call has not been found in Presence OpenGate Proxy Server
-268435465	The call line has not been found in Presence OpenGate Proxy Server
-268435472	The agent has not been found in Presence OpenGate Proxy Server
-268435473	The agent state in Presence OpenGate Proxy Server is not valid to process the operation
-268435474	The agent extension state in Presence OpenGate Proxy Server is not valid to process the operation
-268435475	An error has occurred in Presence OpenGate Proxy Server when placing the call on hold

-268435476	An error has occurred in Presence OpenGate Proxy Server when retrieving the call
-268435477	An error has occurred in Presence OpenGate Proxy Server when hanging up the call
-268435478	An error has occurred in Presence OpenGate Proxy Server when making the call
-268435483	The agent extension has not been found in Presence OpenGate Proxy Server
-268435497	An error has occurred in Presence OpenGate Proxy Server when making the transfer
-268435502	An error has occurred in Presence OpenGate Proxy Server when making the conference call
-268435503	The device has not been found in Presence OpenGate Proxy Server
-268435527	The agent extension state in Presence OpenGate Proxy Server is not valid to make a transfer
-268435528	The maximum number of devices in a conference for Presence OpenGate Proxy Server has been reached
-285212672	The Presence Agent Web Services are not connected to the database
-285278208	An internal error has occurred in the database when processing the operation