Virtual Agent Integration with E-mail

# Overview

We’ve heard several requests to have an e-mail transcript sent to other mailboxes after the Chime session has ended. This sample shows how a post-conversation virtual agent (download or click-through the Virtual Agent Presentation slideshow if you don’t know what this is) can accomplish this by wrapping up *all* of the session data into a body of text and sending it off as an e-mail. Below is some explanation for how to do this in C# and above you can see a few screenshots (under SMTP Integration) showing the flow of integration and how the dispatched e-mail transcript will look.

# Note about references

The project this post-conversation virtual agent sample resides in has a reference to ExtensionLibrary.dll (this can be downloaded in the Virtual Agent SDK), this has the necessary interface definitions to integrate with Chime. Also, this class uses the following .NET references:

using System;

using System.Collections.Generic;

using System.Net;

using System.Net.Mail;

using System.Text;

# Create a class that implements the IVirtualAgent interface

Assuming you can do this, and you have returned a post-conversation virtual agent type on the Load method, there’s not much more to do.

# Get the session data from Chime

In the SeekerConnected method, the virtual agent should ask the PluginManager for the session’s PostChatData, that is all the data that was known when the seeker entered the queue, and the other data the was added over the course of being connected with a live agent, such as chat messages, skill tags, etc. In this example, we ask for the “clean” PostChatData which is stripped of HTML markup and doesn’t contain empty seeker data values. Then we send that data to be dispatched in an e-mail, and finally we tell the PluginManager to disconnect us. For post-conversation virtual agents, the keepAlive value is ignored by the PluginManager because the session is already completed. i.e., “dead”.

 public bool SeekerConnected(int sessionId)

 {

 va\_state = VirtualAgentState.Busy;

 PostChatData chatData = \_pluginManager.PostChatEventClean(sessionId);

 DispatchEmail(chatData);

 bool keepAlive = true;

 \_pluginManager.DisconnectVirtualAgent(sessionId, keepAlive);

 va\_state = VirtualAgentState.Online;

 return true;

 }

# Send the session data as an e-mail

Now that we have the session data, we can format it into an e-mail message and send it to a mailbox, using credentials that are authenticated into the network. (Note, the WriteToString method just builds a string out of the different object fields of PostChatData).

 private void DispatchEmail(PostChatData chatData)

 {

 var data = Convert.FromBase64String(\_password);

 var pw = Encoding.UTF8.GetString(data);

 var Credential = new NetworkCredential(\_userName, pw);

 var \_client = new SmtpClient(\_host, \_port)

 {

 EnableSsl = \_useSsl,

 Credentials = Credential,

 };

 var msg = new MailMessage(\_userName, \_targetEmailAddress, chatData.question, WriteToString(chatData));

 \_client.Send(msg);

 }