ANDROID SDK - RELEASE NOTES

Release notes and changelog for the TTEC Android SDK.

v6.4.0 – Fri 23, 2018

`Expertconnect-6.4.0.aar`

- Chat - High Level - The top navigation bar buttons can now be customized. The behavior, colors, fonts, and icons can be changed. The top two navigation buttons are called the "Back" button (left) and the "End" button (right).

  For documentation on customizing this, refer to our integrator example documentation: https://github.com/humanifydev/SDK-android-integrator#customizing-the-top-navigation-bar-button-items-in-chat

  (PAAS-2608)

- Chat - High Level - Corrected chat messages showing "\n" or "\t" characters when returning to the chat window from backgrounding or moving to another activity in the app. Now, these messages will be correctly converted into their respective control characters (\n = new line, \t = tab). This also restores consistency in chat bubble sizing when returning because the control messages being displayed would cause wrapping (e.g. "hello\nworld").

  (PAAS-2669)

- Chat - The Android SDK should now correctly send the inactive state notification when the chat is backgrounded or the user moves to a different activity. From the agent desktop client, the associate will now see feedback in the display when the Android user moves away from the chat window.

  (PAAS-2399)

- Chat - Added a new function "validateAPI" that allows validation tests of the API server. This might be used to handle the rare scenario that the user still has network connection at the device but lost the ability to route to the API server.This function will return true or false in a completion handler if the API server is accessible and healthy.

  Example:

```
expertConnect.validateAPI(new ExpertConnect.validateAPIListener() {
  @Override
  public void validateAPI(boolean connected) {
    if (connected)
      { // API is reachable and healthy. }
  }
});
```

(PAAS-2636)

- Chat - Low level - Now the SDK will attempt to reconnect the WebSocket if it detects a network recovery and was previously connected to a chat. This should happen automatically when the OS passes notification of a successful reachability event.

  In addition, a new function was added for callbacks when the chat object loses and recovers reachability. To capture these events implement the following listener in required place.

```
//Added the Listener:
private NetworkConnectionMonitor.NetworkConnectionListener listener;

NetworkConnectionMonitor.getInstance().addNetworkConnectionListener(list
ener = new NetworkConnectionMonitor.NetworkConnectionListener() {
  @Override
  public void onConnectionChanged(boolean connected) {
    if(connected)
      { // network connected. }
    else
      { // network disconnected. }
  }
});

//Removed the Listener:
NetworkConnectionMonitor.getInstance().removeNetworkConnectionListener(l
istener);
```

(PAAS-2637)

- Localization - The string for the "No agents available" message default has been modified to the text "No agents available." (in all languages) to mimic the value in the iOS SDK. The key for this string is "expertconnect_no_agents_available".

(PAAS-2633)

- Chat - The chat should now attempt to automatically reconnect if it detects a WebSocket "close" while a chat is active. A rare scenario where this could occur is if the server is restarted or shut down while a chat is in progress. The chat object will attempt to reconnect every 5 seconds with a backoff of adding 5 more seconds each retry.

  (PAAS-2644)

v6.3.2 – Jan 18, 2018

`Expertconnect-6.3.2.aar`

- Chat - The SDK will now send active/inactive state messages when the SDK detects the chat view has entered the background. For example, when a user navigates to another view or presses the home button to navigate away from the app. When this happens, the agent should now see the "[User] has navigated away from chat..." red system text on their desktop client.
  (PAAS-2579)

- Localizations - Added support for Danish, Dutch, Finnish, Norwegian, Polish, Portuguese, and Swedish.
  (PAAS-2528, PAAS-2595)

- Chat - Added a new function for retrieving chat transcripts. ConversationID is the only input parameter and is optional. If a value is present, the function will return the transcript for the given conversationID. If left blank (nil), the function will return all conversation history for the current journeyID. The output is a completion block with an array of Humanify SDK chat message objects (ChatTextMessage, ChatStateMessage, etc).

  Note that the current/last chat conversationID can be retrieved with:
  ```
  String conversationId =
  expertConnect.getIdentityManager().getLastConversationId();
  ```

  Example:

  ```
  ExpertConnectApiProxy api =
  ExpertConnectApiProxy.getInstance(this);
  String conversationId=
  expertConnect.getIdentityManager().getLastConversationId();

  api.getTranscriptForConversation(conversationId, new
  Callback<List<Message>>() {
    @Override
    public void success(List<Message> messages, Response response) {
  ```

```
    if (messages != null && !messages.isEmpty())
    {
      // Happy path. We have history data.
    } else {
      // No history data found.
    }
  }
  @Override
  public void failure(RetrofitError error)
  {
    // An error retrieving history.
  }
});
```

(PAAS-2565)

v6.3.0 – Jan 05, 2018

Expertconnect-6.3.0.aar

- Corrected a crash caused by an incompatibility including the "appcompat" library v7.26.1.0 and higher. The SDK was incorrectly overriding the "getLayoutInflater" function this library used. This version will remove the override and prevent future collisions.
  (PAAS-2577)

- Forms - Landscape mode is now disabled in high level forms.
  (PAAS-2578)

- Chat - Chat messages from the client are now queued for sending to increase reliability of message ordering. Each chat message will wait for the previous message response before being sent. Other SDK API calls outside of chat message sending are not affected by this queuing (such as breadcrumbs, decisioning calls, etc). This behavior is enabled by default and can be disabled by calling the following line once before chat starts:
  `ExpertConnect.getInstance(this).useMessageQueuing(false);`
  (PAAS-2345)

- Chat - Corrected an issue that would leave the chat in an unknown state if the client reconnected after a long period of being disconnected. This would typically occur when the user backgrounded the app or put the phone to sleep for an extended period of time (configurable on the server side, default 30 minutes). The SDK will now poll for the chat status after a short period (5 seconds) and if it finds that the chat was disconnected, it will issue a "disconnectedWithMessage" callback to delegates of ECSStompChatClient. The reason will be "idleTimeout" and the terminatedBy will be "system".

(PAAS-2380)

- Chat - Added the capability to add custom data fields to a chat start which will be displayed at the associate desktop console. The (optional) parameter accepts a dictionary of key-value pairs which will be displayed in the details portion of the associate desktop client.
  Example:
  ```
  HashMap<String, String> channelOptions = new HashMap<String,
  String>();
  channelOptions.put("department", "Call Center");
  channelOptions.put("userType", "Student");
  api.startChat(DEMO_SKILL, null, channelOptions);
  ```
  (PAAS-2313)

- Chat - Corrected a situation where disconnecting from a voice callback escalated from a chat session would disconnect both the callback and the chat. Now, when disconnecting the voice, the user is returned to the chat.
  (PAAS-2403)

- Forms - The "thumbs" treatment for the form type "options" has been graphically enhanced. The type now shows a thumbs up / thumbs down graphic for the choices clickable by the user. This now mirrors the iOS behavior.
  (PAAS-2435)

v6.2.2 – Oct 29, 2017

`Expertconnect-6.2.2.aar`

- Forms - The high level form view has been updated to better support the Android "TalkBack" accessibility feature. The view should now read the display in order, highlight form fields as "buttons", and set focus to the question text when the user navigates to the next or previous item.
  (PAAS-2384)

- Chat - Corrected a situation where the ECSStompChatDelegate would send a "Processing error" to the delegate if the associate sent a string of messages rapidly. This is a server bug that will be fixed in a server patch and often only occurs under a barrage of messages from the associate client. The correction in the SDK is that the WebSocket will detect this error and automatically reconnect. This should be transparent to the user, and the error will no longer be passed to the delegate.
  (PAAS-2438)

- Chat - The high level chat has a new string "expertconnect_chat_exit_title" that can be used to customize the dialog title when the user attempts to leave a chat in progress. Formerly, this used the "expertconnect_chat" string, which was also used in other places. The new

string default value is "Chat" is the same as the previous string was to maintain backwards compatibility. To customize this string, use the following:
<string name="expertconnect_chat_exit_title">Chat</string>
(PAAS-2445)

v6.2.1 – Oct 18, 2017

`Expertconnect-6.2.1.aar`

Fixes

- Chat - Corrected an issue that could cause the user to have to push the "back" button multiple times to back out of the high-level chat activity. This would occur in a certain situation when using the tab bar control and multiple screen transitions. The SDK chat activity should now correctly only require one action to hide/destroy it.
  (PAAS-2401)

v6.2.0 – Oct 6, 2017

`Expertconnect-6.2.0.aar`

Additions

- Chat - The high level chat UI now supports clickable addresses and phone numbers. Addresses will launch the native maps app to the given location, and a phone number will invoke the native call dialing dialog. These links will appear automatically as detected in a normal text message from the associate.
  Example message typed from an associate:
  - Hello, you can find us at 100 Cupertino Plaza, San Francisco CA
  - Hello, you can call us at 555-555-1234

  (PAAS-2234)

- Chat - Added parameter "priority" to enable customization of the chat priority when starting a low-level chat. This parameter will set the chat's priority as it comes to the server affecting how quickly it moves through the queue. Valid values are 1-10, or -1 to take the default priority configured by the server. The following static int values are defined by the SDK and included in *ChannelRequest.java* class:

```
ExpertConnectChatPriorityUseServerDefault = -1;
ExpertConnectChatPriorityLow = 1;
ExpertConnectChatPriorityNormal = 5;
ExpertConnectChatPriorityHigh = 10;
```

```
*Example Implementation to start low level chat:*
ChannelRequest channelRequest = new
ChannelRequest.Builder(ChatActivity.this)
.setTo(skill)
.setFrom(expertConnect.getUserId())
.setSubject("help")
.setMediaType(ChannelRequest.MEDIA_TYPE_CHAT)
.setPriority(ChannelRequest.ExpertConnectChatPriorityHigh)
.build();
api.createChannel(channelRequest);
```

●

    (PAAS-2328)

● Chat - The way a transfer works has been enhanced. First, some older legacy messages are now suppressed in the SDK. These messages were not localized and were not customizable by the SDK. They have been replaced by SDK control messages. When a transfer occurs, the SDK will receive a "RemoveParticipant" message indicating the agent who is leaving the chat. Next, a channel state change of "queued" will occur, indicating the chat is now queued for the transfer agent to pick up. When the agent picks up, an "AddParticipant" message will arrive.
For the high-level chat UI, All three of these messages will be indicated in the chat with an in-line message (not in a chat bubble). These messages are customizable by overriding the following strings in your app's strings.xml file (default values shown):
<string name="expertconnect_joined">You are connected with %s</string>
<string name="expertconnect_left">%s has left the chat</string>
<string name="expertconnect_chat_transfer">The chat is being transferred…</string>>
In addition, there are 4 string replacements that can occur in the ChatJoin and ChatLeave messages:
%s - Backwards compatible. Firstname if found, otherwise, full name.
firstname - Agent's first name
lastname - Agent's last name
userid - Agent's userID
Example: chatJoinMessage = "You are connected with [firstname] [lastname] ([userid]).";
(PAAS-2210)

Fixes

● Chat - The low level chat implementation will now automatically attempt to fetch a new authentication token if the token had expired and the host app attempts to reconnect to chat. Previously, this would result in a "received bad response code from server 401" error being sent to the didFailWithError callback. If the authentication process fails our retry, it will then pass an error to the callback.
(PAAS-2365)

v6.1.2 – Aug 16, 2017

`Expertconnect-6.1.2.aar`

Additions

- Chat - The SDK now provides a method to gather the disconnection reason and who terminated the chat. These two fields can now be found in the ChannelState object. The new fields are:

  disconnectReason - The reason for a chat disconnect. Possible values:
  DISCONNECT_REASON_DISCONNECT_BY_PARTICIPANT
  DISCONNECT_REASON_IDLE_TIMEOUT
  DISCONNECT_REASON_ERROR
  DISCONNECT_REASON_UNKNOWN

  terminatedBy - Which entity caused the chat to end. Possible values:
  TERMINATED_BY_CLIENT
  TERMINATED_BY_ASSOCIATE
  TERMINATED_BY_SYSTEM
  TERMINATED_BY_ADMIN
  TERMINATED_BY_ERROR
  TERMINATED_BY_QUEUE
  TERMINATED_BY_UNKNOWN

  As an example, when an idle timeout occurs, the system should set terminatedBy = TERMINATED_BY_SYSTEM and disconnectReason = DISCONNECT_REASON_IDLE_TIMEOUT.
  When an associate (agent) ends a chat, terminatedBy = TERMINATED_BY_ASSOCIATE and disconnectReason = DISCONNECT_REASON_DISCONNECT_BY_PARTICIPANT.

  Example Implementation:

```
api.registerGetConversationEvent(conversationEventReceiver = new
ApiBroadcastReceiver<ConversationEvent>() {
@Override
public void onSuccess(Context context, ConversationEvent result) {
if (result instanceof ChannelState) {
ChannelState state = (ChannelState) result;
if (ChannelState.STATE_DISCONNECTED.equals(state.getState()))
{ ExpertConnectLog.Debug("Stomp disconnect notification",
"DisconnectReason = " + state.getDisconnectReason() + " and TerminatedBy
= " + state.getTerminatedBy()); }
}
}
@Override
```

```
public void onError(Context context, ApiException error) {
}
});
```

(PAAS-2149)

- Voice Callbacks - The last channelID for voice callbacks is now saved internally in the SDK and sent along with future form submits. This allows for post-call survey data to display on the Call Detail Report for voice callbacks.
(PAAS-2152)

Fixes

- Corrected a crash that could occur when the SDK code attempted to cast an Activity as a FragmentActivity. The following error would be seen in the console log:
```
java.lang.ClassCastException:
com.name.oa.activity.AppLinkLockScreenActivity cannot be cast to
android.support.v4.app.FragmentActivity
```
(PAAS-2221)

- Breadcrumbs - Corrected an issue that would cause each breadcrumb call to generate a new breadcrumb Session and journeyID. This would cause an excessive number of journeys and sessions to be created and fragment the information for a session.
(PAAS-2220)

- Chat - Corrected an issue that would cause a crash when tapping the photo button in chat. This would cause a FileUriExposedException error and crash. The error only duplicates on API 24 and higher.
(PAAS-2192)

v6.1.0 – Jul 19, 2017

`Expertconnect-6.1.0.aar`

Additions

- **IMPORTANT:** The Picasso library included in the SDK has been upgraded to version 2.5.2. The compile line in Gradle config should be updated to the following:
```
compile 'com.squareup.picasso:picasso:2.5.2'
```
(PAAS-2063)

- Localization - Added translation strings for Portuguese-Brazil (pt-br) and Portuguese-Portugal (pt).

(PAAS-2104)

- Forms - The SDK will now send a previous chat channelID along with a form submit command. This ensures that the "Call Detail Report" will have the post-chat form data in the results. For reporting purposes, the form data will only be included in the report if the form is submitted within 5 minutes of the end of the conversation.
(PAAS-2118)

- Chat - Added specific customizations for the timestamp text font and color. The styles to modify is:

```
<style name="ExpertConnect.TextAppearance.Chat.Timestamp"
parent="ExpertConnect.TextAppearance" />
```

Example Implementation:

```
//Override the following style in host app styles.xml file.
<style name="ExpertConnect.TextAppearance.Chat.Timestamp"
parent="ExpertConnect.TextAppearance">
  <item name="android:textColor">@color/dark_gray</item>
  <item name="android:textStyle">bold</item>
  <item name="android:textSize">10sp</item>
</style>
```

(PAAS-2137)

- The Android SDK now supports automatic generation of journeyID's and breadcrumb sessionID's. The integrator no longer needs to call startJourney() and wait for completion before SDK functions can be called. Now, any API call made from the SDK will cause the server to generate one and pass it back to SDK.
This requires Humanify API version 5.6.0 or greater.
(PAAS-1122)

Fixes

- Answer Engine - Corrected an HTTP 500 error when a user selects "thumbs down" as their rating for the selected article.
(PAAS-1408)

- Answer Engine / Forms - The "thumbs down" graphic has been changed to a filled-in graphic. Previously, it was an outline with no fill. This now matches the "thumbs up" graphic.
(PAAS-2092)

- Chat - Inline Forms - If the user submits an inline form from chat, the message bubble will now display the text "submitted". Previously, the user could tap again to submit the form multiple times.
(PAAS-2041)

v6.0.0 – Jun 19, 2017

`Expertconnect-6.0.0.aar`

Additions

- Localization - Added support for Spanish-Spain (es-ES) and Italian (it). Base Spanish (es) was also updated with minor language corrections.
  (PAAS-1793)

Fixes

- Forms - Corrected a crash that would occur when the "Next" button was pushed in the form view and the host app's "buildToolsVersion" was 25.x or higher. The crash stack trace would mention the "ViewPager" object.
  (PAAS-1986)

- Chat - Corrected an issue where sending hyperlinks from the Humanify Expert Desktop platform would appear to be empty chat bubbles. This was due to the text color being the same color as the background. A new style was added to customize this color and the default color was changed to black.
  `<item name="expertconnect_text_color_link">@android:color/black</item>`
  (PAAS-2022)

v5.9.1 – Jun 1, 2017

`Expertconnect-5.9.1.aar`

Fixes

- Chat - The Humanify SDK has the capability to accept PDF files and display them in chat. This is an optional feature. Previously, a permission check for external_storage (required by the Android OS since 6.0) was always displayed regardless of whether PDFs were used or not. Now, this has been updated to only show the check at the time the PDF is attempted to be displayed. In addition, we have added a parameter that allows disabling this check altogether:
  `expertConnect.setAllowStoragePermissionCheck(false); // This would disable the permission check.`

  In addition, the external_storage permission dialog can be customized:
  `<string name = "expertconnect_storage_permission_title" >Permission Required</string>}`
  `<string name = "expertconnect_storage_permission_message" >Application requires storage permission to display PDF and Video files</string>`

  Note: receiving PDF files will not work if the permission check is not called beforehand. If you

are using this feature, make sure to call the check yourself or leave it enabled in the SDK. (PAAS-1987)

- Forms - The "closedWithForm()" callback function will now be invoked if the user presses the device's "back" or the "navigation up arrow" button. In addition, if an error occurs while loading the form, the "formErrorRaised()" callback function is called.

  Callback functions:
  ```
  public void formErrorRaised(Activity context, ApiException exception)
  public boolean closedWithForm(Activity context, Form form)
  ```

  To set the form callback listener:
  ```
  expertConnect.setFormListener(true, this);
  ```

  (PAAS-2008)

v5.9.0 – May 22, 2017

`Expertconnect-5.9.0.aar`

Additions

- Forms - The high level form screen has been updated to include new interface functions and member variables. The host app can now react when the user answers a question, submits a form, or add custom behaviour when the user clicks "Close" to exit the form view.

  This can be done by using the following two steps:

  1. Implement the ExpertConnectConversationApi.FormListener interface in required host app Activity/Fragment to receive events such as answered questions, submitted forms, etc.

  ```
  void answeredFormItem(Activity context, FormItem formItem, int index);
  void submittedForm(Activity context, Form form, String name, ApiException exception);
  boolean closedWithForm(Activity context, Form form);
  ```

  2. Set the FormListener from the same host host app Activity/Fragment to receive the values to interface functions.

  ```
  public void setFormListener(boolean
  showFormSubmittedView,ExpertConnectConversationApi.FormListener
  formListener) { }
  ```

  showFormSubmittedView - Whether or not the submitted view is shown after form submission.If true, will show the "form submitted" page after last question is answered. false will do nothing. Set to false if you want to customize the transition after the survey is answered straight on to another view.

  formListener - Set the FormListener from host app to receive the events

Example:
```
 public class ExampleHostAppActivity extends AppCompatActivity implements
ExpertConnectConversationApi.FormListener {

  @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        ExpertConnect.getInstance(this).setFormListener(true, this);
    }

     @Override
    public void answeredFormItem(Activity context, FormItem formItem, int
index) {}

    @Override
    public void submittedForm(Activity context, Form form, String name,
ApiException exception) {}

    @Override
    public boolean closedWithForm(Activity context, Form form) {
        return true;
    }
  }
```

New Functions from ExpertConnect Class:
```
expertConnect.getFormName(); // Retrieves the current form's name.
expertConnect.getForm(); // Retrieves the current form, including any
data input filled out by the user already.
```

(PAAS-1907)

- Chat - The chat instance will now close the channel when the user closes the application cleanly (from settings or the recent app list). The SDK will not close the channel if the user "force stops" the app as Android does not allow for any code to run in this situation. (PAAS-1928)

- Direct API calls (low level) - All API calls to the server should now use the same method template (retrofit). These function calls allow for synchronous and asynchronous implementations. Formerly, some of the API calls below were using a broadcast method. This is still in place for backwards compatibility, but we strongly recommend updating to use the retrofit asynchronous model. This model will be the one documented going forward and the one found in the integrator example apps as of SDK 5.8.0. The following API calls now have a retrofit implementation:
    - Create Journey – api.createJourney()
    - Set Journey Context – api.setJourneyContext()
    - Send Single BreadcrumbsAction – api.breadcrumbSendOne()

- - Send Multiple BreadcrumbsAction – api.breadcrumbQueueBulk()
  - Send BreadcrumbsSession – api.breadcrumbsSession()
  - Post Decision Data – api.postDecisionData()
  - Post Answer Rate – api.postAnswerRate()
  - Post form – api.postForm()
  - Close Conversation – api.closeConversation()
  - Post Extended Profile – api.registerPostExtendedProfileResponse()

  (PAAS-628)

- JavaDocs are now available for the SDK. They can be found in the SDK-Android-Integrator GitHub repo under the /JavaDoc folder. For reference, the following article details how to add JavaDoc for a .aar file to your Android project: http://stackoverflow.com/questions/24882854/add-local-javadoc-to-local-aar-in-android-studio#answer-32566073
  (PAAS-1839)

Fixes

- Corrected an issue where the user-agent header passed to the server was not reflecting the SDK's version but instead showing the integrator app's version. The new user-agent header will now show all of these versions. Example:
  MyApp/1.0.0 EXPERTconnect/5.9.0 (iOS/10.3.0)
  (PAAS-1913)

- Forms - Form submission from the high-level form view now contains the added data fields to properly display the form results to the Expert Desktop agent console. The form data will still show up using older versions of the SDK but the form fields will not have populated labels.
  (PAAS-1918)

- Chat - Our Humanify agent client has the capability to send PDF files to the client. If you are implementing support for this, your host app will need to add permissions for displaying PDF files. Add the following line in "build.gradle" of your app:
  dependencies
  {
      compile 'com.joanzapata.pdfview:android-pdfview:1.0.4@aar'
  }
  (PAAS-1861)

- Chat - Added missing localization for the "No Internet Connection" string seen during chat if internet connection is lost.
  (PAAS-1908)

- Chat - Added missing Italian localization for the "Leave Chat Queue?" dialog message seen if the user attempts to exit a chat while in queue.

(PAAS-1909)

- Chat - Added missing Italian translation for the "chat timeout" message that would occur if the user went idle for longer than the idle timeout in a chat.
(PAAS-1910)

- Voice Callback - Fixed a crash that would occur if the user was in an active voice call, backgrounds the app, and then a disconnect is signaled from the agent.
(PAAS-1860)

v5.8.1 – April 20, 2017

`Expertconnect-5.8.1.aar`

Fixes

- Corrected an issue that would cause the SDK to crash if the new debug callback method introduced in 5.8.0 was not implemented by the host app. A workaround to solve this issue prior to receiving Android SDK 5.8.1 is to implement the debug callback method (see Addition #1 in 5.8.0 below).

v5.8.0 – April 18, 2017

`Expertconnect-5.8.0.aar`

Additions

- Added a debugging callback that allows your app to integrate SDK level debugging into your debugging output of choice. To integrate, implement the following:

```
expertConnect.setLoggingCallback(new ExpertConnect.LoggingCallback() {
  @Override
  public void getLog(int logLevel, String tag, String message, Throwable error) {
     // Add this debug message to your debug output integration.
     Log.e(levelString, message, error);
  }
});
```

Note: ECSLogLevel is an enum with the following values: ECSLogLevelError, ECSLogLevelWarning, ECSLogLevelDebug, ECSLogLevelVerbose, ECSLogLevelNone
(PAAS-1776)

- A new function 'getResponseFromEndpoint' is now exposed that can be used to perform custom API calls to Humanify. This is useful for integrators who want to parse the response JSON manually. The function returns JSON in the form of a string variable.

```
ExpertConnectApiProxy apiProxy =
ExpertConnectApiProxy.getInstance(context);
String endPoint = "answerengine/v1/top10?num=5";
apiProxy.getResponseFromEndpoint(endPoint, new Callback<String>() {
    @Override
    public void success(String result, retrofit.client.Response response)
{

    }
    @Override
    public void failure(RetrofitError error) {
    }
});
```
(PAAS-1677)

- Chat - Behavior has been improved when the network is interrupted while the user is in a chat session. A red bar will slide in from the top of the window when network loss is detected and will disappear automatically when connection is regained. In addition, if the user gets reconnected to a chat that has timed out, a dialog will be shown alerting the user that the chat session has ended. The network error bar and this dialog are customizable as shown below.

  To Change the background color of network bar, override the following value in **colors.xml**:
  ```
  <color name="expertconnect_network_bar_background">#ffeb3b</color>
  ```

  To Change the network bar text color override the following value in colors.xml
  ```
  <color name="expertconnect_network_bar_text">#424242</color>
  ```

  Two new overridable strings have been added:
  ```
  /* Usage: Displayed in a red bar at the top of the chat window when the
  network connection is lost. */
  <string name="expertconnect_chat_queue_network_error">No internet
  connection.</string>

  /* Usage: Dialog displayed when the user background the app and returns
  while in-queue for a chat. This is the dialog message. */
  <string name="expertconnect_chat_queue_disconnected_message">Your chat
  request has timed out. Please try again.</string>
  ```
  (PAAS-1708)

Fixes

- Chat - Corrected a web socket configuration issue that could potentially cause messages to arrive from the server out of order. With the previous configuration, a significant server delay processing messages could cause unexpected behavior at the agent's client, such as being unable to accept a chat.
  (PAAS-1737)

- Chat - Fixed an issue where the user could get stuck on the queue screen indefinitely. Now, if the SDK detects any timeout, disconnect, or network error, it will display the generic alert that says chat has timed out and to please retry.
  (PAAS-1781)

- Chat - Corrected an issue that could cause the Android chat to become hung after recovering from a temporary network loss. This issue would occur if the chat session was disconnected while the network was unavailable on the mobile device. Now, the chat should reconnect and correctly show any messages sent while the network was down (up to and including a chat disconnect message).
  (PAAS-1804)

- Chat - The message displayed for "less than one minute remaining" in chat queue will now correctly be displayed only for estimated wait times of 60 seconds or less. Previously, rounding caused this message to be shown for an ETA of 90 seconds or less.
  (PAAS-1773)

- Chat - The english strings for the dialog displayed when the user tries to exit the queue have been updated to have consistent capitalization grammar. The word "leave" no longer has an uppercase L. Other localizations were already lowercased. New default values:

  ```
  <string name="expertconnect_leave_queue_yes">Yes, leave</string>
  <string name="expertconnect_leave_queue_no">No, stay</string>
  ```

  (PAAS-1768)

- Chat - The text displayed when the user tries to leave a chat queue was updated for the French language. A missing comma was added to the phrases "Non, rester" and "Oui, laisser".
  (PAAS-1762)

v5.7.1 – February 27, 2017

Expertconnect-5.7.1.aar

Additions

- Chat - The default estimated wait time strings have been updated to include the line "Please remain on this screen to keep your spot in queue." at the end. The string keys updated were:
  - expertconnect_chat_wait_time (plural - one, and other)
  - expertconnect_connect_chat_time_greater_than_five

  (PAAS-1677)

v5.7.0 – February 2017

Expertconnect-5.7.0.aar

Modifications

- Chat - a new dialog has been added to ask the user if they are sure they want to leave the chat queue. This dialog is displayed while the user is in the queue for a chat. The dialog has four new localized strings that can be overridden. The keys and default values are:
  - `<string name="expertconnect_leave_queue_title">Leave Chat Queue?</string>`
  - `<string name="expertconnect_leave_queue_message">By leaving now, you will lose your place in the chat queue.</string>`
  - `<string name="expertconnect_leave_queue_yes">Yes, Leave</string>`
  - `<string name="expertconnect_leave_queue_no">No, stay</string>`

  (PAAS-1604)

- Chat - The default user avatar image has been changed to a grey silhouette. This image will be seen if chat avatars are enabled and setAvatarImage() has not been called. Agent avatars are set using the Humanify Supervisor Console. If you would like to set the user's avatar, set the following field to your desired image file:

  `ExpertConnect.getInstance(getApplicationContext()).setAvatarImage(my Bitmap);`

  (PAAS-1163)

- Agent Availability - The "getDetailsForSkill" function has been removed from the SDK. Instead, the "getDetailsForExpertSkill" function should be used. To capture the estimated wait time, use the "estWait" field in the response object.

(PAAS-1255)

- Chat - All three localized strings used to display an estimated wait message to the user allow for the use of the estimated Wait time variable. These three localization strings can be added to your project to replace the default text provided by the SDK. To use the variable in your text string, add the following three character into your string: %d.
  Eg:

```
<plurals name="expertconnect_chat_wait_time">
<item quantity="one">Chat volume is low. Your wait time is expected to
be less than a minute. Please remain on this screen and you will be
connected with a HumanifyGuide.</item>
<item quantity="other">Your wait time is approximately %d minutes.
Please remain on this screen and you will be connected with a
HumanifyGuide.</item>
</plurals>
<string name="expertconnect_connect_chat_time_greater_than_five">Chat
volume is heavy. Your wait time is expected to be %d minutes(s). Please
remain on this screen and you will be connected with a
HumanifyGuide.</string>
```

(PAAS-1602)

Fixes

- Chat - Corrected an issue that would cause the chat "Send" button to display it's enabled/disabled state in reverse. When the button was functionally enabled, it would display as disabled (greyed out), and vise versa. The button should now show the correct enabled/disabled state.
  (PAAS-1172)

- Several fields were removed from JSON responses that were extra information and would always return NULL from the server. If your code is referring to one of these fields, it should be removed. The removed fields are:
  - `UserProfile.userid`
  - `AnswerEngineRateRequest.actionId`
  - `AnswerEngineRequest.navContext`
  - `AnswerEngineRequest.actionId`
  - `BreadcrumbAction.action`
  - `Channel.type`
  - `Channel.source`
  - `AnswerEngineHistoryItemDetail.actionId`
  - `ConversationHistoryItemDetail.actionId`
  (PAAS-631)

- Chat - Corrected an issue where the agent's avatar would display as a grey circle if a transfer was initiated. Now, the SDK should show the agent's avatar for a transfer message.

(PAAS-1138)

- Chat - Corrected an issue that would cause messages from a previous conversation to be displayed in a subsequent chat if the user navigates away and returns to the chat view. (PAAS-1139)

- Chat - Corrected an issue that would cause an Answer Engine article sent by the agent to be displayed incorrectly in the chat view. The message would be narrow vertically with cutoff text. The answer engine preview should now display correctly. (PAAS-1141)

- Answer Engine History - Corrected a potential app crash when loading the answer engine history view. This view is manually invoked by a host app and is not a part of the normal Answer Engine operation. (PAAS-1142)

- Answer Engine History - The history items are now displayed in chronological order. Previously, a bug caused the items to be out of order. (PAAS-1409)

v5.6.0 - November 2016

`expertconnect-5.6.0.aar`

Fixes

- Forms – When the server response provides a form name that does not exist, a generic error message will be returned and the app should not crash. (PAAS-651)

- Voice Callback – Corrected the issue where the "End Call" button would not end the call when the phone sleeps during the call or if the call screen is minimized and the voice call screen goes to the top. (PAAS-681)

Modifications

- Removed the cacheCount and cacheTime from the expertconnect initialization and gave them default values. (PAAS-638)

v5.5.0 - October 2016

`expertconnect-5.5.0.aar`

Additions

- Localization for Chinese (simplified) has been added. (PAAS-167)

Fixes

- Chat - Corrected an issue that would cause the chat client to miss an idle timeout message if the device was asleep when the timeout occurred.
  (PAAS-25)

- Answer Engine – Corrected an issue that could cause the Answer Engine's "type ahead" feature to not return results when it should have.
  (PAAS-234)

- GetForm API – Corrected an issue where the app could crash if an invalid form name parameter was provided.
  (PAAS-640)

- Chat – Corrected a crash that could occur in some situations when the idle timeout message is received.
  (PAAS-676)

v5.4.0 - July, 2016

`expertconnect-5.4.0.aar`

Additions

- Rate response changes have been implemented. Clients will receive a min and a max field from the server in addition to treatment type. It will store this data as it displays the form, and when it returns a response to the server, it will pass back the min, max, and value. For binary (thumbs, etc.) min=-1, max=1, where -1=down and 1=up. Stars: min=1, max=10 (allows for half-star values).
  (EC-2111)

- Added a system message, "Your chat will timeout in 60 seconds due to inactivity." It will be displayed to the user prior to the chat timing out. This message is localized in all available languages.

(EC-2266)

- Added ability to send the region and language settings regardless of, or in spite of, what the end user's phone settings are.
  (EC-2460)

- Added localization support for German.
  (EC-2710)

- Added support for x-ia-context in global headers to all API calls. Contains the current journeymanager context in EXPERTconnect.journeyManagerContext.
  (EC-2839)

- Added localization support for English (UK).
  (EC-2997)

Fixes

- Fixed issue that occurred after ending a voice callback during a chat, where the conversation exchanged prior to the callback disappeared from the chat screen.
  (EC-2401)

- Fixed issue where the app crashed during a callback, if the user clicked "End Call."
  (EC-2469)

- Fixed issue where messages from the Expert Desktop did not appear on the Android chat screen, if the Android app ran in the background and then was brought back to the front.
  (EC-2480)

- Fixed issue where the chat was not terminated on the agent side in the Expert Desktop, if the user ended the chat on the Android phone.
  (EC-2489)

- Fixed issue where the chat screen displayed a "Try Reconnect" button when the voice call of an escalated chat was terminated.
  (EC-2492)

- Fixed problem that occurred when invalid input on a form was submitted, and an empty error message showed up. The user would not be unable to return to the chat screen.
  (EC-2493)

- Fixed issue that occurred during a chat, if the user tapped "<" to go back to the previous page and then tapped "continue chat" to return to the chat screen, the previous conversation did not appear on the screen.

(EC-2510)

- When using the integrator app localized to French or Spanish, the "No agents available" message was not translated properly.
(EC-2541)

Changes

- Agent Availability – The SkillDetail class has been moved from the package com.humanify.expertconnect.api.model to com.humanify.expertconnect.api.model.experts. This will require an update if your app is referencing this class in an import statement. (EC-3250)

- Journey Management – JourneyResponse no longer contains a getOrganization() method call.

- Your app's organization can be inferred from the identity delegate token so the app should never need to fetch or use the organization within the SDK. Any reference to the getOrganization() method call will cause a compile error and should be removed. (EC-3250)

v5.2.5

`expertconnect-5.2.5.aar`

Changes

- Removed Location Manager access from the SDK.

- The SDK will not include geolocation in breadcrumbs automatically.

v5.2.4

`expertconnect-5.2.4.aar`

Additions

- Previously the server returned a "No agents available" message in English only. Now this message will be displayed in English, Spanish or French, depending on the language selection on the device.
To customize this message, override the "no_agents_available" string for each language:

```
<string name="no_agents_available">No Agents Available!</string>
```

COPYRIGHT AND DISCLAIMER