**Basic Setup**

1. **Create a backup of your project.** I’m 96% sure nothing in my plugin will break your project, but it never hurts to back it up before installing a new plugin.
2. Install Instant Messages plugin by copying it into your project’s Plugins folder.
3. Move the files in the Audio and Graphics folders included into the same respective folders in your project.
4. Explore the settings in Plugins > Instant Messages > 000\_Settings.
5. Define the Groups and Contacts for your game in Plugins > Instant Messages > 002\_Configuration\_GroupContact.
6. Define the Conversations for your game in Plugins > Instant Messages > 003\_Configuration\_Conversation.
7. To access the Instant Messages app, you need to call pbInstantMessages. You can create a Pause Menu or Pokegear option to do this, depending on what your game setup looks like. Review the definitions of these for your game, duplicate one, and create one for Instant Messages using this function.
8. If you wish to customize the graphics for your game, explore the Graphics > UI > Instant Messages, or the emojis in Graphics > Icons

**Documentation**

**Functions**

pbReceiveIM(ConversationID, Silent)

* Description: Sends the player a message.
* Arguments:
  + ConversationalID
    - Required. Set to the ID of the conversation you want to send to the player.
  + Silent
    - Optional. Set to true to make the message be sent to the player without any notification (unless it’s an Important message).
    - If this isn’t set, receiving the message will notify the player.
* Examples:
  + pbReceiveIM(:OAK\_TEST) => Sends the :OAK\_TEST conversation.
  + pbReceiveIM(:QUIET\_MESSAGE, true) => Sends the :QUIET\_MESSAGE conversation without any notification to the player.

pbPendDelayedIM(ConversationID, steps: <val>, time: <val>)

* Description: Pends an Instant Message to be sent to the player after a set number of steps/amount of time. For example, you might want your rival to send you a message 60 minutes after the battle you just had.
* Arguments:
  + ConversationalID
    - Required. Set to the ID of the conversation you want to pend to be sent to the player.
  + steps: <val>
    - Optional. Replace <val> with an integer representing the minimum number of steps the player needs to take before it is received. This only applies if PASSIVE\_TRIGGER\_TYPE is set to 1 or 3.
    - If this isn’t set, it defaults to the value set in PASSIVE\_STEP\_MIN
  + time: <val>
    - Optional. Replace <val> with an integer representing the minimum number of in-game minutes that needs to pass before it is received. This only applies if PASSIVE\_TRIGGER\_TYPE is set to 2 or 3.
    - If this isn’t set, it defaults to the value set in PASSIVE\_TIME\_MIN
* Examples:
  + pbPendDelayedIM(:OAK\_DELAYED) => Pends the :OAK\_DELAYED conversation to be sent to the player later.
  + pbPendDelayedIM(:OAK\_DELAYED, steps: 30) => Pends the :OAK\_DELAYED conversation to be sent to the player after they take 30 or more steps.
  + pbPendDelayedIM(:OAK\_DELAYED, steps: 30, time: 60) => Pends the :OAK\_DELAYED conversation to be sent to the player after they take 30 or more steps and/or 60 in-game minutes have passed.

pbPendRandomIM(ConversationID)

* Description: Pends an Instant Message to be included in the PASSIVE\_TRIGGERS\_RANDOM\_POOL. This can be used to have conversations be randomly sent to the player passively without having to rely on Switches, Variables, or a piece of Code. For example, you can pend a random message after talking to a minor NPC, without having to create a switch to keep track that you talked to them.
* Arguments:
  + ConversationalID
    - Required. Set to the ID of the conversation you want to add to the PASSIVE\_TRIGGERS\_RANDOM\_POOL.
* Examples:
  + pbPendRandomIM(:OAK\_RANDOM) => Pends the :OAK\_RANDOM conversation to be sent passively as if it were part of the PASSIVE\_TRIGGERS\_RANDOM\_POOL.

pbInstantMessages

* Description: Opens the Instant Messages app, showing the list of all messages the player can view. There are no arguments for this function.

pbInstantMessagesDirect(GroupID)

* Description: Use if you want to directly open a group, instead of the Instant Message app’s menu view.
* Argument:
  + GroupID
    - Required. Set to the ID of the group you want to open directly.
* Example:
  + pbReceiveIM(:PROFOAK) => Opens the :PROFOAK conversation.

pbSetIMTheme(Color)

* Description: Changes the theme color of the Instant Messages app.
* Arguments:
  + Color
    - Required. Set to a string that is the name of a folder in the Graphics > UI > Instant Messages > Themes folder.
* Example:
  + pbSetIMTheme(“Blue”) => Sets the theme of the Instant Messages app to be blue.

pbHasReceivedIM?(ConversationID)

* Description: Returns **true** if the player has already received a specific conversation.
* Arguments:
  + ConversationalID
    - Required. Set to the ID of the conversation you want to check if the player has already received.
* Example:
  + pbHasReceivedIM?(:OAK\_TEST) => Returns **true** if the player has already received the :OAK\_TEST conversation.

pbHasUnreadIM?

* Description: Returns **true** if the player has unready Instant Messages. There are no arguments for this function.

**Manual Setup – Defining your own Contacts, Groups, and Conversations**

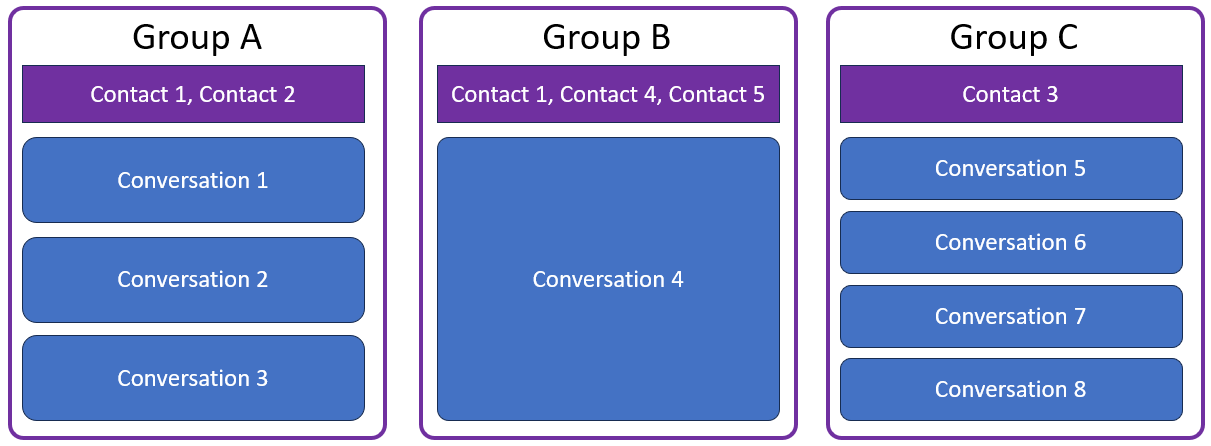
The core of this plugins involves defining conversations for your own game, and along with it, contacts and groups for those conversations. Review the definition instructions in the 002\_Configuration\_GroupContact and 003\_Configuration\_Conversation files for how to define these.

**General Structure of Contacts, Groups, and Conversations**

Conversations are the individual messages or sets of messages you send to the player. These act like conversations you would have in an event using Show Text, for instance.

Every conversation needs to be housed within a group. Groups can contain many separate conversations. Only conversations that the player has been sent will be visible at that time. But, if you send the player a conversation in a group that already has read conversations, the player will be able to review those old conversations when viewing the new one. This is similar to how you may have a group chat/text in real life, you have a conversation with that group yesterday, and then today you have a new conversation. You can still view all the messages from both conversations. Each individual group will appear as a separate item in the Instant Messages menu view.

Every group contains contacts. These are the definitions of NPCs that you have conversations with within that group. Contacts are shared across different groups and conversations.



*Diagram showing what the structure of 3 different groups could look like.*

**Feature Callouts**

* Choices the player makes during a conversation can have an impact on the game by allowing it to set a Game Variable. Review the information in 003\_Configuration\_Conversation for how to set that up.
* The player can fast forward through new conversations using the SPECIAL key. This can be turned off.
* The player can sort their conversations list using the SPECIAL key. This can be turned off.
* The player can pin specific messages using the ACTION key in their conversation list so they appear at the top. This can be turned off.
* You can use icons as emojis in conversation messages using the “<icon=>” feature supported by base Pokemon Essentials. However, the way that code is structured, there needs to be a visible character after the icon definition. So add an extra space if it's the last thing in the message or have additional text after an emoji.
* There are special types of messages you can include, such as :Picture, :Leave, and :GroupName. These allow you to send a picture message, make a contact look like it left the group, and change the group name while in the middle of the conversation. Review the information in 003\_Configuration\_Conversation for how to set these up.