Honeywell

# Guided Work Solutions

GWS App 4.3

# Localization Guide

# Disclaimer

Honeywell International Inc. ("HII") reserves the right to make changes in specifications and other information contained in this document without prior notice, and the reader should in all cases consult HII to determine whether any such changes have been made. HII makes no representation or warranties regarding the information provided in this publication.

HII shall not be liable for technical or editorial errors or omissions contained herein; nor for incidental or consequential damages resulting from the furnishing, performance, or use of this material.



Web Address: automation.honeywell.com

# Patents

For patent information refer to hsmpats.com

# TABLE OF CONTENTS


## Chapter 1 - Introduction ..... 1

What is Localization? ..... 1

Supported Languages ..... 1

Key Terminology ..... 4

Speech Recognition ..... 4

Text-to-Speech (TTS) Configurations ..... 5

Using Embedded TTS in Android devices ..... 5

Changing System/Google TTS ..... 5

Associated Documentation ..... 6

Getting Help ..... 8

Additional Documentation ..... 8

Honeywell Voice Reseller Services ..... 8

Honeywell Voice Technical Support ..... 8

Honeywell Voice Customer Service ..... 8

Honeywell Voice Hardware Repair ..... 9

## Chapter 2 - Localization in GWS App ..... 11

Localization Files ..... 11

Resource Fallback Heirarchy ..... 11

Fallback Rules ..... 12

Validation Checklist ..... 12

Localization Content ..... 12

Modifying Localization ..... 13

Vocab Words ..... 15
Common Content ..... 21
Upload Your Localization ..... 22
Android Devices ..... 22
Talkman Devices ..... 23
Chapter 3 - Additional Information ..... 25
FAQ ..... 25

CHAPTER

# 1 INTRODUCTION

This document describes the process of performing localization for the GWS App.

## What is Localization?

Localization means adapting an app, in this case GWS App, so it feels natural and easy to use for people in different countries or regions. It's more than just translating words, it is about making the whole experience fit the local culture.

## Supported Languages

Starting with version 4.3, GWS App officially supports translations for twelve languages. These languages are identified in the Official Translations column below. Officially supported means the GWS App is delivered with resource (resx) files that provide translation of the keys used in the GWS App.

For other languages, the resxfiles can be created using the instructions later in this document.

All the languages listed in the following table can be used with BlueSteak on Android and Talkman devices. Pick Up and Go (PnG) support is noted in the table.

| Language Name | Locale Code | Talkman Devices | Android Devices | Pick Up and Go (PnG) | Official Translation |
|---|---|---|---|---|---|
| Arabic (Worldwide) | ar_WW | Yes | Yes | Yes (arAE, arSA) | No |
| Arabic (Emirates) | ar_AE | No | Yes | Yes | No |
| Arabic (Saudi Arabia) | ar_SA | No | Yes | Yes | No |
| Bengali | bnIN | No | No | No | No |
| Bulgarian | bgBG | No | Yes | Yes | No |

| Language Name | Locale Code | Talkman Devices | Android Devices | Pick Up and Go (PnG) | Official Translation |
|---|---|---|---|---|---|
| Czech | cs_CZ | Yes | Yes | Yes | No |
| Danish | da_DK | Yes | Yes | Yes | No |
| German (Germany) | deDE | Yes | Yes | Yes | Yes |
| Greek | el_GR | Yes | Yes | Yes | No |
| English (Australia) | en_AU | Yes | Yes | Yes | No |
| English (India) | en_IN | Yes | Yes | Yes | No |
| English (United Kingdom) | en_GB | Yes | Yes | Yes | No |
| English (United States) | enUS | Yes | Yes | Yes | Yes |
| Spanish (Spain) | esES | Yes | Yes | Yes | Yes |
| Spanish (Mexico) | esMX | Yes | Yes | Yes | Yes |
| Persian | faIR | No | Yes | Yes | No |
| Finnish | fi_FI | Yes | Yes | Yes | No |
| French (Canada) | frCA | Yes | Yes | Yes | Yes |
| French (France) | fr_FR | Yes | Yes | Yes | Yes |
| Hebrew | heIL | No | Yes | Yes | No |
| Hindi | hi_IN | Yes | Yes | No | No |
| Croatian | hr_HR | Yes | Yes | No | No |
| Hungarian | hu_HU | Yes | Yes | Yes | No |

| Language Name | Locale Code | Talkman Devices | Android Devices | Pick Up and Go (PnG) | Official Translation |
|---|---|---|---|---|---|
| Indonesian | id_ID | Yes | Yes | Yes | No |
| Italian | it_IT | Yes | Yes | Yes | No |
| Japanese | jaJP | Yes | Yes | Yes | Yes |
| Korean | ko_KR | Yes | Yes | Yes | No |
| Malay | msMY | No | Yes | Yes | No |
| Dutch (Belgium) | nlBE | Yes | Yes | Yes | No |
| Dutch (Netherlands) | nlNL | Yes | Yes | Yes | Yes |
| Norwegian | no_NO | Yes | Yes | Yes | No |
| Polish | pl_PL | Yes | Yes | Yes | No |
| Portuguese (Brazil) | ptBR | Yes | Yes | Yes | Yes |
| Portuguese (Portugal) | ptPT | Yes | Yes | Yes | Yes |
| Romanian | ro_RO | Yes | Yes | No | No |
| Russian | ruRU | Yes | Yes | Yes | Yes |
| Slovak | sk_SK | Yes | Yes | Yes | No |
| Slovenian | sl_SI | Yes | Yes | No | No |
| Serbian | sr_RS | Yes | Yes | No | No |
| Swedish | sv_SE | Yes | Yes | Yes | No |
| Chinese (Sichuan) | sz_CN | No | Yes | Yes | No |

| Language Name | Locale Code | Talkman Devices | Android Devices | Pick Up and Go (PnG) | Official Translation |
|---|---|---|---|---|---|
| Thai | th_TH | Yes | Yes | Yes | Yes |
| Turkish | tr_TR | Yes | Yes | Yes | No |
| Chinese (Simplified) | zh_CN | Yes | Yes | Yes | Yes |
| Chinese (Hong Kong) | zh_HK | Yes | Yes | Yes | Yes |
| Chinese (Traditional) | zh_TW | Yes | Yes | Yes | Yes |

# Key Terminology

The following terms are used throughout this document.

- Resource (.resx) file: A file containing all keys for localization. A resource file ban be base (no language identified, which is American English) or can contain a two or four digit locale identifier, such as es (Spanish) or es-MX (Mexican Spanish).
- Common content: Keys (words) that the app shows or speaks.
- Vocab words: Keys (words) that the app shows or speaks and can recognize when spoken by the user.
- Phonetic substitution: A string to generate an alternate pronunciation the word. For example "led" for the word "lead" when referring to the metal.
- Pick Up & Go (PnG): Speaker independent speech recognition. user training is not required but can be used to improve speech recognition.
- BlueStreak: BlueStreak speech recognition is speaker dependent. Each user must train templates that aid in speech recognition.

# Speech Recognition

GWS App uses Pick & Go (PnG) or BlueStreak for speech recognition. Depending on the recognition engine the app uses, be aware that:

- PnG operators can use multiple languages.
- BlueStreak operator cans only be associated to one language.

For more information on speech recognition, refer to the Speech Recognition Guide available at help.honeywellaidc.com.

# Text-to-Speech (TTS) Configurations

By default, the GWS App uses the native TTS software of the device.

- The A700x uses the TTS included in VoiceCatalyst. The available TTS Engines supported in VoiceCatalyst are the same ones listed as supported for Talkman devices in Supported Languages.
- Most Android devices use Google Mobile Services (GMS) for TTS functionality. These TTS engines may be pre-installed or downloaded from the Google Play store.
- Non-GMS Android devices may have an inferior TTS engine or no TTS engine.

## Using Embedded TTS in Android devices

The GWS App contains Embedded TTS Engines for the following languages:

- Chinese (Simplified) - zh_CN
- Chinese (Hong Kong) - zh_HK

To enable embedded TTS for zh-CN and zh-TW, use the following configurations in either OnetimeStartupSettings.config or PersistentWorkflowSettings.config. A “true” value in each language indicates the Embedded TTS Engine is activated:

```
{
  "Repositories":
  {
    "EmbeddedTtsAVoiceLocaleSupportedOverride" : {
      "zh-CN": "true",
      "zh-HK": "true"
    },
  }
}
```

## Changing System/Google TTS

GWS App changes the TTS (Text-To-Speech) language automatically based on loaded globalization culture info using the Google TTS. It automatically downloads the TTS data for the selected language. If user wants to download / update TTS manually on an android device, they can follow below steps:

1. Open **Settings**
    a. Unlock your Android device.
    b. Tap the **Settings** app (usually a gear icon).
2. Navigate to **Accessibility** or **System**
    a. Scroll down and tap **Accessibility**, or
    b. Tap **System** > **Languages & input**.
3. Tap on **Text-to-Speech Output**
    a. Under **Accessibility** or **Languages & Input**, find and tap **Text-to-Speech Output**.
    b. This opens the TTS settings.
4. Select **Preferred Engine**
    a. You'll see options like **Google Text-to-Speech Engine** or other installed engines.
    b. Tap the gear icon next to your preferred engine (usually Google TTS).
5. Change Language
    a. Tap **Language**.
    b. A list of supported languages appears.
    c. Select the language you want (e.g., English (US), Spanish, French, etc.).
6. Test the Voice (Optional)
    1. Tap **Play** or **Listen to an example** to hear how the voice sounds in the selected language.
7. Adjust Speech Rate and Pitch (Optional)
    a. You can also adjust:
        - **Speech rate**: how fast it talks
        - **Pitch**: how high or low the voice sounds
8. Exit **Settings**
    a. Once you've made your changes, press the **back** button or **home** button to exit.

# Associated Documentation

- **Guided Work Solutions Checklist Plug-In Product Description**: This document contains the information related to this specific plug-in included in this release. The Checklist Plug-In can be used to integrate Honeywell Voice with several Host Systems.
- **Guided Work Solutions Checklist Plug-In Integration Guide**: This document contains the installation and configuration information and integration specifications for the Checklist

Plug-In.

- **Guided Work Solutions Developer Guide**: This document contains detailed information about GWS concepts required to create new plug-ins or customizations.
- **Guided Work Solutions ToolVad User Guide**: This document describes the ToolVad utility, which is used to convert an existing Talkman VAD into a VAD that allows the Talkman to use the plug-in embedded mode.
    - **ToolVad 2.0** is used for most VAD conversions
    - **ToolVad 1.1.1** must be used when embedding the Distribution or Checklist Plug-Ins due to a known issue.
- **Guided Work Solutions ToolVad Release Notes**: This document contains information on the latest ToolVad release.

# Getting Help

## Additional Documentation

Additional documentation may be found in your product package and on online partner portals.

Find most Honeywell Voice technical documentation at help.honeywellaidc.com.

## Honeywell Voice Reseller Services

If you purchased equipment or services through a Honeywell Voice reseller, please contact your reseller first for support or to purchase a support plan.

## Honeywell Voice Technical Support

Submit incidents or questions to honeywell.custhelp.com or contact Honeywell Technical Support Center:

- **Americas**
  Email: VoiceTechnicalSupport@Honeywell.com
  Phone: +1(866) 862-7877
- **Europe, Middle East, Africa**
  Email: VoiceTechnicalSupport@Honeywell.com
  Phone: +44 (0) 1344-65-6123
- **Rest of World**
  Email: VoiceTechnicalSupport@Honeywell.com
  Phone: +1 (412) 376-9384

To report support incidents or ask technical questions for other Honeywell devices, visit honeywell.com/PSStechnicalsupport.

## Honeywell Voice Customer Service

For order placement or customer service inquiries:

- **North America, Latin America**
  Email: VoiceCustomerServiceAmericas@Honeywell.com
  Phone: +1(866)862-6553
- **Europe, Middle East, Africa, Turkey**
  Email: voicecustomerserviceEMEA@honeywell.com
  Phone: +44 (0) 1698-915777
- **Japan**
  Email: csjapan.pss@honeywell.com

Phone: +81-3-6730-7344

- **Brazil**
  Email: ACSHSMCentraldepedidos@honeywell.com
  Phone: +55 (31) 2391-5600
- **Asia Pacific**
  Email: VoiceCustomerServiceAPAC@honeywell.com
  Phone: +44 16989 15777

## Honeywell Voice Hardware Repair

For returns or to check the status of a Return Material Authorization (RMA) for Voice hardware products:

- **Americas**
  Email: VoiceRMA@Honeywell.com
  Phone: +1 (866) 417-6988
- **Europe, Middle East, Africa**
  Email: VoiceEMEARMA@honeywell.com
  Phone: +1 (866) 417-6988
- **Rest of World**
  Email: VoiceRMA@Honeywell.com

For returns or to check the status of an RMA for other Honeywell hardware products, visit the SPS RMA portal: sps-support.honeywell.com/s/pss/pss-rma

CHAPTER

2

# LOCALIZATION IN GWS APP

## Localization Files

GWS App uses .resx files to use and modify localization. These files are XML files. You need a .resx file for each language or local language you want to target. For example:

- GWExternalResources.resx - Base language
- GWExternalResources.fr.resx - Neutral language, in this case French
- GWExternalResources.es-MX.resx - Locale, in this case Spanish (Mexico)

As you see, the name of the files takes the following form:

GWExternalResources<.Locale>.resx

Where:

- The optional <.Locale> represents the locale of the resource file contents.

## Resource Fallback Heirarchy

When the GWS App loads resources for a given culture, it does not use a single resx file in isolation. Instead, it resolves each key through a strict three-level fallback chain:

1. Locale (e.g.. es-MX)
    - If found, the key is used.
    - If key is not found, the next level is used.
2. Neutral (e.g., es)
    - If found, the key is used.
    - If key is not found, the next level is used.
3. Base (en — GWExternalResources.resx)

This means that if a key is missing in the locale file, the system looks for it in the neutral file. If it is also missing there, the system falls back to the base English file. This ensures no key is dropped.

## Fallback Rules

| Situation | Result |
|---|---|
| Locale file exists and key is found | Key resolved from locale |
| Locale file exists but key missing<br>Neutral has it | Key resolved from neutral |
| Locale file exists but key missing<br>Neutral is also missing key | Key resolved from base |
| Only locale file exists, no neutral | Recognition fails or raises a validation error |
| Neutral file missing entirely | Recognition fails for any language in that family |

## Validation Checklist

Before deploying a localized configuration, verify:

- Base file (GWExternalResources.resx) is present and contains the full command set.
- Neutral file exists for every language family used (e.g., es for es-MX and es-ES).
- Neutral file contains all keys from the base file.
- Locale file only overrides keys that differ from the neutral file.
- No locale file exists without the corresponding neutral file.

# Localization Content

A localization file contains <data></data> tags which the app uses for localization. Each data block corresponds a phrase, word or number. For instance, this tag is used for the affirmative button text on Android:

```
<data name="AffirmativeButtonText">
  <value>Yes</value>
  <comment>"Yes" button text.</comment>
</data>
```

In this example, *AffirmativeButtonText* is known as a key. The GWS App uses keys in a GWExternalResources.resx to populate the text you see on the screen or the phrases you can hear. **You should not modify the key**, but you can modify the value that is between <value></value> tags though there are some exceptions, see Vocab Words.

So, if you want to localize the previous key for Spanish, you would modify the value. In this case, this tag has to be placed in GWExternalResources.es.resx

```
<data name="AffirmativeButtonText">
  <value>Si</value>
  <comment>"Yes" button text.</comment>
</data>
```

Tips:

- Keep the base key names in GWExternalResources.resx unchanged (for example: VCOMMAND01, VCOMMAND01_Display, VCOMMAND01_Spoken).
- In the per-language files (GWExternalResources.es.resx, etc.) provide localized _Display and _Spoken values for each VCOMMAND defined in the base file.
- _Display is the human-facing text. It preserves accents and punctuation.
- _Spoken is the ASR-friendly (Automatic Speech Recognition)/normalized form. Prefered entry is plain ASCII, lower-case, remove any diacritics or provide phonetic variant. This is what recognition and ToolVad expect for training.

# Modifying Localization

Before modifying localization for GWS App, you must have:

- A GWExternalResources.resx file, which is a base localization file. This file contains all keys the app uses for localization.
- A GWExternalResources.resx could be found in the delivered 'ExternalResources' folder that ships with each GWSApp release; it is the authoritative base resource file and must reside alongside any GWExternalResources.<locale>.resx files.
- A text editor that supports XML.
- If you are creating a locale resx file such as French (Canadian) you must end up with three files:
    1. GWExternalResources.resx (the base file)
    2. GWExternalResources.resx.fr (the neutral language file)
    3. GWExternalResources.fr-CA.resx (the locale file)

IMPORTANT
Before starting to modify key values be aware of the types of content you want to localize:

- Common content: The ones the app only shows or speaks. This is the most common item to localize.
- Vocab words: The ones that the app can recognize by voice, show and speak. They are used for phonetic substitution.

To start modifying localization:

1. Create a copy of GWExternalResources.resx. You should add the language code for the language you want to target. See supported languages above.

   For example, GWExternalResources.fr-CA.resx for French (Canada).

2. Open the copy with your text editor. You should see a text file like this:

```
410     <data name="CancelButtonText">
411       <value>Pas plus</value>
412       <comment>"Cancel" button text.</comment>
413     </data>
414     <data name="CasesLabel">
415       <value>cas :</value>
416       <comment>Application screen label for the number of
    cases.</comment>
417     </data>
418     <data name="CasesPrompt">
419       <value>{0} cas</value>
420       <comment>Spoken prompt stating the number of cases.</comment>
421     </data>
422     <data name="confirmation_of_spoken_value">
423       <value>{0}, n’est-ce pas?</value>
424       <comment>Spoken prompt asking user for confirmation of a spoken
    value.</comment>
425     </data>
426     <data name="confirmation_of_spoken_value_and_anchor">
427       <value>{0}, {1}, n’est-ce pas?</value>
428       <comment>A spoken prompt asking the user to confirm their spoken
    value and anchor</comment>
429     </data>
430     <data name="ConfirmDeleteWorkerWorkflowActivity_Header">
431       <value>Supprimer ce travailleur?</value>
```

```
    <comment>Application screen header</comment>
  </data>
  <data name="CurrentProductLabel">
    <value>Point actuel :</value>
    <comment>Application screen label for the current item.</comment>
  </data>
```

3. Modify the value for the key you want to localize.

   For example, let's say you want to modify the value for **MenuItemDialogue_ConfirmMultiple_Prompt** key and translate it to French. The original value is:

```
<data name="MenuItemDialogue_ConfirmMultiple_Prompt">
  <value>You Selected: {0}, and {1}, correct?</value>
</data>
```

   The text you want to translate to French is **You Selected: {0}, and {1}, correct?**. The translated text should be: **Vous avez sélectionné : {0} et {1}, c'est bien cela?**. Then, you can update the value in the .resx file:

```
  <data name="MenuItemDialogue_ConfirmMultiple_Prompt">
    <value>Vous avez sélectionné : {0} et {1}, c'est bien cela?</value>
  </data>
```

4. If the key has a value within <comment><\comment> tags, don't change this value.
5. Repeat steps 3 and 4 for all the keys in the .resx file.

## Vocab Words

The app uses these words or phrases for:

- Speech recognition.
- Phonetic substitution.

For most vocabs, each word or phrase has three keys:

- Vocab value
  - This is the phrase for the vocab.
  - The app uses this value for storing templates in Voice Console.
  - You **must not modify** this value.

    - Its length must not exceed 74 characters.
    - It must not contain these special characters: |\\?*<\":>+[]/'
- Display value
    - The app uses this key for displaying the vocab.
    - Modify this value according to your target language.
- Spoken value
    - The app uses this value for phonetic substitution.
    - Modify this value according to your target language.

> **NOTE**
> Vocab and spoken values always should be a lowercase string.

One example is the VCOMMANDS. Each VCOMMAND needs to have three keys:

```
<data name="VCOMMAND02" xml:space="preserve">
  <value>dock</value>
  <comment>Displayed during voice training of command 02</comment>
</data>
<data name="VCOMMAND02_Display" xml:space="preserve">
  <value>Dock</value>
  <comment>Displayed during voice training of command 02</comment>
</data>
<data name="VCOMMAND02_Spoken" xml:space="preserve">
  <value>dock</value>
  <comment>The phonetic word for command 02</comment>
</data>
```

For other phonetic substitution content, the same key is used for the vocab, spoken and display value:

```
<data name="phonetic_h" xml:space="preserve">
  <value>hotel</value>
  <comment>The phonetic word for the letter "h"</comment>
</data>
```

You can help the app to pronounce the phrases better by changing the spoken value.

For instance, the word lead (a type of metal) or lead (to guide). If you want to help the app to pronounce the metal, you should change the spoken value to "led":

```
<data name="VCOMMAND02" xml:space="preserve">
  <value>lead</value>
  <comment>Displayed during voice training of command 02</comment>
</data>
<data name="VCOMMAND02_Display" xml:space="preserve">
  <value>lead</value>
  <comment>Displayed during voice training of command 02</comment>
</data>
<data name="VCOMMAND02_Spoken" xml:space="preserve">
  <value>lead</value>
  <comment>The phonetic word for command 02</comment>
</data>
```

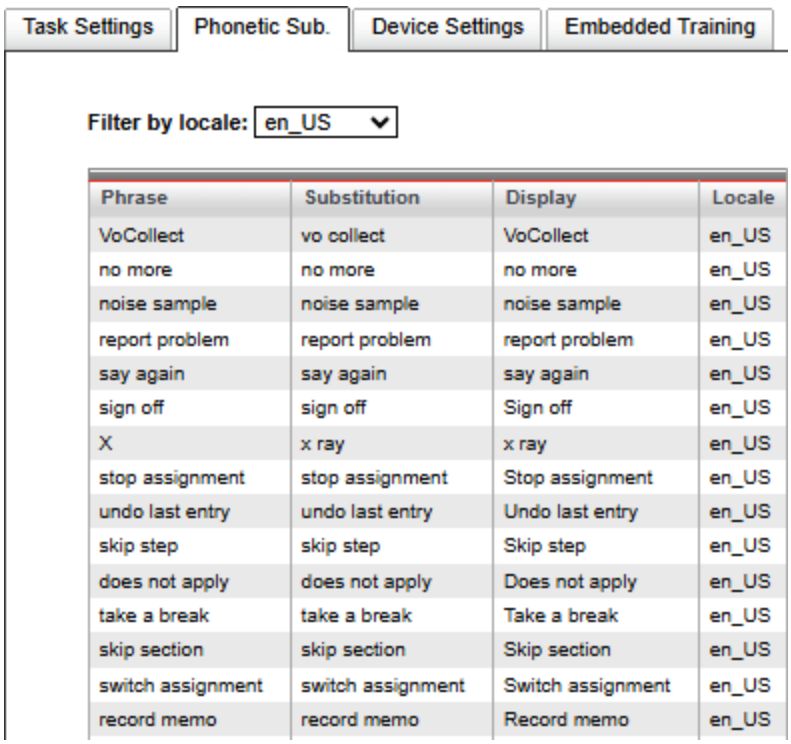
<-- Spoken value changed to led -->

```
<data name="VCOMMAND02_Spoken" xml:space="preserve">
  <value>lead</value>
  <comment>The phonetic word for command 02</comment>
</data>
```

## Phonetic Substitution

When speaking a prompt, the application replaces the vocab value to the value provided in the spoken key. Wherever the vocab value is found inside a screen, the word is automatically changed to the display value of the language that was provided via resx.

You can find the **Phonetic Substitution** values in Voice Console for Talkman devices in the task package screen:




| Phrase | Substitution | Display | Locale |
|---|---|---|---|
| VoCollect | vo collect | VoCollect | en_US |
| no more | no more | no more | en_US |
| noise sample | noise sample | noise sample | en_US |
| report problem | report problem | report problem | en_US |
| say again | say again | say again | en_US |
| sign off | sign off | Sign off | en_US |
| X | x ray | x ray | en_US |
| stop assignment | stop assignment | Stop assignment | en_US |
| undo last entry | undo last entry | Undo last entry | en_US |
| skip step | skip step | Skip step | en_US |
| does not apply | does not apply | Does not apply | en_US |
| take a break | take a break | Take a break | en_US |
| skip section | skip section | Skip section | en_US |
| switch assignment | switch assignment | Switch assignment | en_US |
| record memo | record memo | Record memo | en_US |

Values can be added to this table for any other phonetic substitutions needed. Note that adding a value to this table only enables the word to be substituted, but it doesn't enable the value to be a valid vocab to be used within the application.

For users migrating from VIO, GWS App 4.3 and later automatically maps the following VIO Vocabs. This happens when Backward Compatibility is turned on and Server. mode is used. The following equivalent mappings are made in both Android and Talkman.

| Vio Vocab | New Vocab Phrase |
|---|---|
| VCONFIRM | ready |
| VYES | yes |
| VNO | no |
| VOPTIONS | options |
| VSIGNOFF | signoff |
| VCANCEL | cancel |

Example: To manage subtitution for VSIGNOFF, use the existing entry for signoff instead of adding a new row for managing VSIGNOFF.

For Plug-In users, prompts using VIO Vocabs need to be updated to the available vocabs described in the following section.

## Vocab Words Keys

You can reference this table when translating vocab words.

| Vocab | SpokenKey | DisplayKey |
|---|---|---|
| no more | VocabWord_NoMore | VocabWord_NoMore |
| noise sample | VocabWord_NoiseSample | VocabWord_NoiseSample |
| report problem | OverflowMenuItem_ ReportProblem | OverflowMenuItem_ ReportProblem |
| say again | VocabWord_SayAgain | VocabWord_SayAgain |
| sign off | VSIGNOFF_Spoken | VSIGNOFF_Display |
| . | whole_number_separator_ phonetic | whole_number_separator |
| continue | Phonetic_Continue | Display_Continue |
| help | Phonetic_Help | Display_Help |
| louder | vocab.louder | vocab.louder |
| negative | vocab.negative | vocab.negative |
| no | VocabWord_No | VocabWord_No |
| options | VocabWord_Option | Display_Option |
| ready | phonetic_accept_entry_word | ReadyButtonText |
| sleep | vocab.sleep | vocab.sleep |
| softer | vocab.softer | vocab.softer |
| wakeup | vocab.wakeup | vocab.wakeup |
| yes | VocabWord_Yes | VocabWord_Yes |

| Vocab | SpokenKey | DisplayKey |
|---|---|---|
| - | phonetic_dash | symbol_dash |
| A | phonetic_a | phonetic_a |
| B | phonetic_b | phonetic_b |
| C | phonetic_c | phonetic_c |
| D | phonetic_d | phonetic_d |
| E | phonetic_e | phonetic_e |
| F | phonetic_f | phonetic_f |
| G | phonetic_g | phonetic_g |
| H | phonetic_h | phonetic_h |
| I | phonetic_i | phonetic_i |
| J | phonetic_j | phonetic_j |
| K | phonetic_k | phonetic_k |
| L | phonetic_l | phonetic_l |
| M | phonetic_m | phonetic_m |
| N | phonetic_n | phonetic_n |
| O | phonetic_o | phonetic_o |
| P | phonetic_p | phonetic_p |
| Q | phonetic_q | phonetic_q |
| R | phonetic_r | phonetic_r |
| S | phonetic_s | phonetic_s |
| T | phonetic_t | phonetic_t |

| Vocab | SpokenKey | DisplayKey |
|---|---|---|
| U | phonetic_u | phonetic_u |
| V | phonetic_v | phonetic_v |
| W | phonetic_w | phonetic_w |
| X | phonetic_x | phonetic_x |
| Y | phonetic_y | phonetic_y |
| Z | phonetic_z | phonetic_z |
| cancel | VocabWord_Cancel | GuidedWork_CancelButtonText |

## Common Content

For this type of content, the app can only speak or show them on screen.

Usually, they are:

- Prompts.
- Text for the user interface.
- Error messages.

For instance, the login button:

```
<data name="LoginButtonText" xml:space="preserve">
  <value>Login</value>
  <comment>"Login" button text.</comment>
</data>
```

Be cautious when a placeholder appears, like{0} or {1}, within the text you want to localize. You must not delete them from the text. They get replaced by another text when the app is running.

```
<data name="MenuItemDialogue_ConfirmMultiple_Prompt" xml:space="preserve">
  <value>You Selected: {0}, and {1}, correct?</value>
</data>
```

# Upload Your Localization

## Android Devices

Recommended:

- adb installed.
- An Android device with GWS App installed.
- All GWExternalResources<.Locale>.resx files you want to upload.

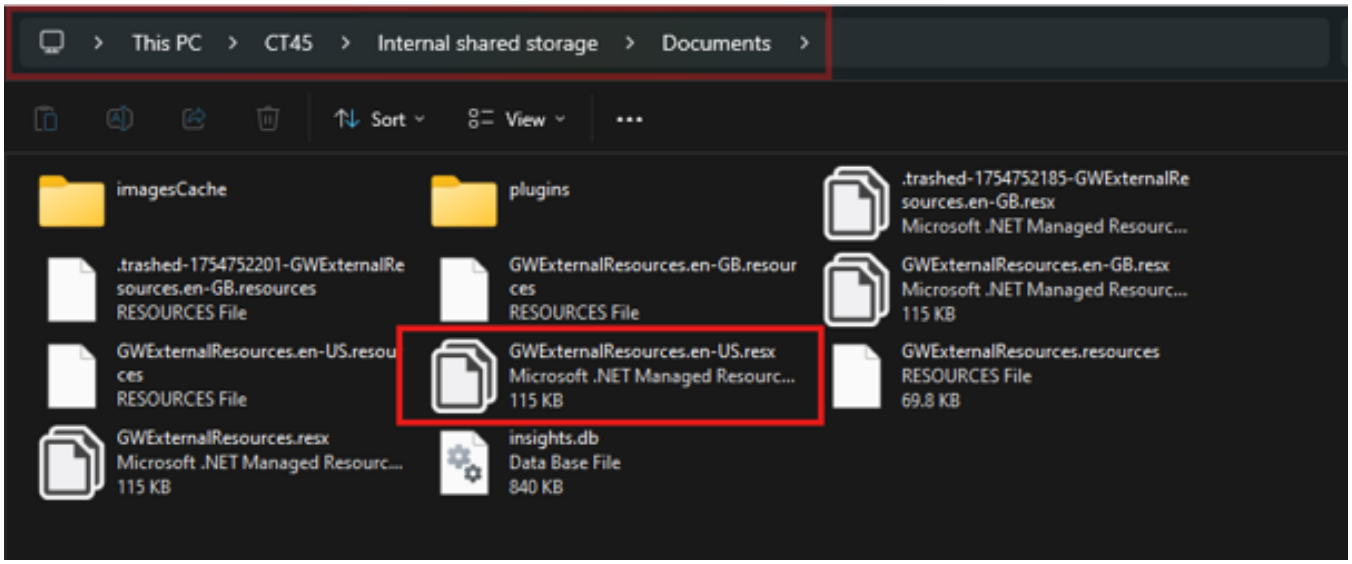
Steps:

1. Connect your Android device to your computer.
2. Turn on file sharing with your computer in your Android Device.
3. Push both GWExternalResources.resx and the modified GWExternalResources<.Locale>.resx via adb or Windows Explorer to /<InternalStorage>/Documents/ where <InternalStorage> denotes the root folder of the device.

   For example, with adb:

   ```
   adb push GWExternalResources.resx
   /storage/emulated/0/Documents/GWExternalResources.resx
   adb push GWExternalResources.fr-CA.resx
   /storage/emulated/0/Documents/GWExternalResources.resx
   ```

   and with Windows Explorer:



4. Open the app and transition to the home screen.
5. Change the app language to the language of the .resx you pushed.
6. You should see the changes already applied.

## Talkman Devices

Before starting, you must have:

- A GWS App .vad file.
- All GWExternalResources<.Locale>.resx files you want to upload.
- Toolvad, a command line interface.

**Recommended:**

- Put your .vad file in the same directory as Toolvad.

There are two versions of ToolVad.

- Refer to the *GWS ToolVad User Guide* 2.0 for most use cases.
- Refer to the *GWS ToolVad User Guide* 1.1. when embedding either the Checklist or Distribution Plug-In due to a known issue.

Steps when using ToolVad 2.x:

1. Put all your GWExternalResources<.Locale>.resx files inside a directory. Be sure the base GWExternalResources<.Locale>.resx is included.
2. Open PowerShell or Command Prompt where Toolvad is located.
3. Run the following command:

   Toolvad --vad-file <VAD FILE> --resources <.RESX DIRECTORY> --output-filename <OUTPUT_FILENAME>

   Where:

   - <VAD FILE> is your GWS App .vad file.
   - <.RESX DIRECTORY> is the directory that contains the .resx files.

   For example:

   ToolVad --vad-file VIO.net8.0-linux-arm64.vad --resources  plugins\custom\resources  --output-filename 5_plugin.vad
4. Toolvad creates a <NAME OF VAD>-plugin.vad file. Import this file as a Task into Voice Console.
5. Create a Task Package.
6. Load the Task Package to a A700x.

CHAPTER

3

# ADDITIONAL INFORMATION

The app follows a fallback procedure when the app doesn't find a key in the current culture the app is running. For example, if **User's culture setting** is es-MX (Spanish - Mexico):

- **Resource lookup starts**: .NET looks for a resource file named GWExternalResources.es-MX.resx.
- **If not found**: It falls back to GWExternalResources.es.resx (general Spanish).
- **Still not found**: It finally falls back to GWExternalResources.resx (default, which is English).

If you push a GWExternalResources.<locale>.resx that is not officially supported by Honeywell:

- UI strings from that file are used when the app runs in that culture. There is no need to add to LanguageNames.

**PnG availability**:

- If the apk contains the PnG models for the language, PnG is automatically enabled when any VoiceConsole versionis used.
- For VoiceConsole SaaS deployments, if a PnG model exists on the server and the device has network connectivity, the model is downloaded and PnG is enabled.
- For VoiceConsole On Prem deployments, it is not possible to download PnG models. The apk with the complete set of models needs to be used to support PnG.

# FAQ

- What is it in a base GWExternalResources.resx file?
  A base .resx file contains all the keys to localize GWS App (around +600 keys), for both Android and Talkman. The content of this .resx file is in English, which is the base language.

  Consider that:

  - GWExternal resx files contained in plug-in packages contain the keys for GWS App plus other keys the plug-in can use.
  - For custom plug-ins, you should add any key that your plug-in requires.
- Can I add VCOMMANDS to a task through Voice Console for a Talkman device?
  No, Voice Console doesn't support adding VCOMMANDS. You can only add VCOMMANDS

using Toolvad.

- If I am using different plug-ins, how does localization for each plug-in work?
  To effectively use localization for a plug-in, you have to push the GWExternalResources.resx for the plug-in the app uses at the moment on Android.

  For Talkman, you can only target the localization for one plug-in, either for Embedded or Service mode.

Honeywell
855 S Mint St
Charlotte, NC 28202

automation.honeywell.com

GWS App Localization Guide

May 2026