Honeywell

Release Notes
Honeywell Development Kit 3.0

July 22 2026

Verifying Download Files

The release files for Honeywell Development Kit (DevKit) 3.0 are zip files signed with a certificate. This allows a user to verify the files they download have not been modified. After downloading the files, use the process below to verify authenticity before unzipping the files.

1. Java JDK (version 1.8 or newer) must be installed.
2. Execute the jarsigner executable against the file to verify. Example:
   jarsigner –verify Honeywell-DevKit-3.0-signed.zip

Code Changes

Changes to code, including breaking API changes, can be found in the CHANGELOG.md file in the devkit directory, included in the DevKit zip file. Some of these changes are also included later in this document.

New Features in Honeywell DevKit

Custom Embedded Training

Custom Embedded Training is now available for BlueStreak on Android devices. This adds the ability to specify phrases composed of vocab words. These phrases are then trained during Enrollment or Update Training.

For Talkman devices this behavior is unchanged. These phrases are specified using the Embedded Training tab when creating or editing a Task Package.

For Android devices, these phrases are specified in the EmbeddedTraining.xml file found in the configuration directory for the app.

- Dash Vocab Template Update
  - If currently using "dash" in the vocab word the user is asked to retrain.
  - Template name for "dash" vocab word is now '-'.

    - This change was made to follow a similar implementation as whole_number_separator vocab word.
- Visual Changes
    - During training, embedded training phrases are space separated.
    - Previous embedded training "101" is now displayed as "1 0 1".

  This change improves readability for embedded training. For example: "alpha bravo" vs. "alphabravo"

Update Train

Update Train can be configured to enable training of all numbers in one update train instance for Android devices.

The DevKitResources.resx file has a new resource key:

```
`UpdateTrainSelectWordWorkflowActivity_AllNumbers` : `All Numbers`
```

When performing update train and numbers are in the vocabulary the "All Numbers" option is added to the top of the list in the current and all vocab sets.

Control Word Training

Specific control words are now set to the recognizer for embedded training for Android devices. The behavior changes as follows:

- Operators are asked to train the phrase "Control Help".
- Operators are asked to train "Say Again" vocab six times.
- Operators are to train the phrases "Control Report Problem" and "Control Noise Sample" if they are voice enabled. Refer to DevKit documentation for details on voice enabling these items.

VoiceConsole Parity Features

Additional device reporting features are added for Android devices.

> **IMPORTANT**
> VoiceConsole 6.5 or later is required to support these features.

Device State

Device State is supported on Talkman devices but functionality was limited for Android devices. The **Device State** is displayed on the **View Devices** table within VoiceConsole.

Android devices report the following states:

- **On**: Reported when a licensing attempt begins. This state report previously existed, but was sent during every connection attempt.
- **Ready**: Reported when the home screen is reached with no operator signed in or after an operator had signed off.
- **In Use**: Reported after an operator has successfully signed in.
- **In SmartTalk Call**: Reported when a SmartTalk call starts. The prior state (Ready or In Use) is automatically restored when the call ends.
- **Off**: Reported when the application is closing. This state previously existed, but was sent during license release.

> NOTE
> The Android OS may terminate the app without cleanup in some scenarios or on a force-stop. This prevents the Off state from being sent.

Device Last Message

The following events are logged in the device activity history within VoiceConsole. The **Last Message** is displayed on the **View Devices** table within VoiceConsole.

- **Terminal Ready**: Sent when the home screen is reached and the device becomes available.
- **Operator Signed Off**: Sent when an operator explicitly signs off.
- **Call Started**: Sent when a SmartTalk call begins.
- **Call Ended**: Sent when a SmartTalk call ends.

VoiceNotes and Memos

SRCOMM Support

SRCOMM support is added for VoiceNote and Memo record and playback on Android devices. This includes tone support for latency, start recording, and stop recording. Only PCM format is supported. Stereo PCM plays only the left channel.

Sampling Rate

The Recording Sample Rate is configurable. This setting affects both SRCOMM and HFP. The available sampling rates are:

- 16000: This is the default rate and is the recommended setting for Transcription Server 1.8 and later.
- 11025: This setting is recommended for versions of the Transcription Server earlier than 1.8.

IMPORTANT
SRX3 firmware 6.13 or later is required to support 16000 Hz. Earlier versions only support 11025 Hz.

The sampling rate is set in the VoiceAndAudioConfig repository.

```
{
    "Repositories": {
        "VoiceAndAudioConfig": {
            "VoiceRecordingSampleRate": "11025"
        }
    }
}
```

NOTE
The RecordingSampleRate that was previously used in this repository has been discontinued.

Language Model Downloads

Language model downloads are automated for Android devices. A new LanguageAvailability repository is added to specify language models to be downloaded or excluded.

In this example, en-GB and it-IT are added to the automatic download and pt-PT is excluded.

```
{
  "Repositories": {
    "LanguageAvailability": {
      "Additional": "en-GB,it-IT",
      "Excluded": "pt-PT"
    }
  }
}
```

The automatic language model download starts on the Home screen after the license is authorized. On the start or return to the Home screen a background task is triggered to download language models that are not available on the device. The list is determined as follows:

- Add languages specified in LanguageAvailabilityService.
- Add languages which have external resource files pushed to the device.
- Add languages specified in the 'Additional' list in LanguageAvailability as shown above.
- Remove languages specified in the 'Excluded' list in LanguageAvailability as shown above.

Rapid Scanning

Rapid Scanning is available for Talkman and Android devices. This feature allows a user to scan, queue, and process multiple barcodes back-to-back. This feature is best suited for scenarios involving repeated scans of the same type, such as batches of identical SKUs and is off by default. This is available for *GetValue*, *GetLongValue*, and *GetFloat* intents. A *ScanMode* property has been added to the *ValuePropertiesObject* for each intent. The scan modes available are:

- **ScanMode.Single**: Accepts only the first valid scan.
- **ScanMode.Multiple**: Allows successive valid scans to queue up and be processed.

This property is set as follows:

```
wfo.ValueProperties.ScanMode = ScanMode.Multiple;
```

> IMPORTANT
> To ensure backward compatibility, if **DisableScanning** is set to true, it overrides the **ScanMode** regardless of what was set when creating the workflow object.

When more than one rapid scan is queued, the application does not speak the prompts encoded in the intents nor does it update screens with scan results (on Android devices) while processing and submitting the queued scans if the intent's prompt is `null`. When this occurs, users are not able to see each scan result displayed in the value entry box. They only see the first of a batch of rapid scans displayed. If queueing up alternating *GetValue*, *GetLongValue*, and *GetFloat* intents the scans queue up and are processed across the different types. Scans are evaluated for validity based on each individual intent.

There are instances in which the queued up scans are cleared.

- When entering an intent that does not support this feature (non-Value intents).
- When entering an intent that is set to **ScanMode.Single**.
- When entering an intent that has **DisableScanning** set to `true` (regardless of **ScanMode**).
- When an invalid scan is encountered anywhere during the processing of queued scans. An invalid scan consists of any scan that fails validation. For example:
  - If any **ExpectedScannedValues** are set and the scanned value does not match any of those values.
  - The scanned value does not fall within set **MinValue** and **MaxValue**, if applicable.
  - A scanned Float value does not fall between the **MinWholeDigit**, **MaxWholeDigit**, **MinDecimalDigits**, and **MaxDecimalDigit** parameters, if applicable.
  - The scanned Float value is not of the appropriate character set, if applicable. For example, the scan contains a negative value when negative is not allowed.

GetLongValue

UseFloatValidation

UseFloatValidation is an optional feature for the *GetLongValue* Intent to enforce float-only numeric entry rules

- This new property controls whether the long value entry behaves normally or enforces float-specific validations. It defaults to false - float validation is disabled by default.
- Only the first decimal point is recognized. Subsequent decimal-point words are ignored.
- Behavior is consistent with Float Value Dialogue.
- Multiple decimal points are no longer permitted when this property is enabled.

Operator Login by Number

> **NOTE**
> This feature can be disabled. See Disable Login Features.

Android devices support operator login by operator number, a feature previously only supported for Talkman devices. Operators can now log in by entering their operator number. The existing team-based and scan-based login methods are still supported.

- A new *SignInWithOperatorNumber* workflow intent (*WorkflowIntent.SignInWithOperatorNumber*) is available for workflows that require operator number only confirmation when the operator is already identified.
- A new *GuidedWorkOperatorNumberSignInView* user interface view handles operator-number-only entry without operator selection.
- The following RESX key-value pairs have been added to DevKitResources.resx:

Key Name	Value
Generic_SelectLoginOptions_Prompt	Operator Login Options
Generic_SignOn_Failed_Operator_Number_Entry_Blank	Operator Number is required
Generic_SignOn_Failed_Operator_Number_Entry_DidNot_Match	Operator Number {0} not found
Generic_SignOn_Failed_DueTo_an_Error	Sign on failed due to an error

Key Name	Value
Validation_SignInWithOperatorNumber_Number_Required	Operator Number is required
Validation_SignInWithOperatorNumber_Number_Invalid	Operator Number must contain only digits
Validation_SignInWithOperatorNumber_Number_TooLong	Operator Number must be {0} digits or less
Validation_SignInWithOperatorNumber_Number_Missing	Operator Number entry not found
GuidedWork_Operator_Teams_SignIn	Operator Teams
GuidedWork_Operator_Number_SignIn	Operator Number
GuidedWork_SignInWithOperatorNumber_Label	Enter Number

- Added support for password-required login at the workflow level during the operator load-by-number login feature. VoiceLink and LApp uses *CreateSignInIntent* to align with existing devkit login flow which helps to prepopulate the *UserID* field automatically based on operator number which is associated to the operator. Workflows can now explicitly require password entry as part of operator sign-on, enabling partner-defined authentication flows during operator number login.
- LApp (devkit/LAppModule/BusinessLogic/LAppLoginStateMachine.cs): LApp now uses a dedicated post-sign-on login state machine to validate LApp server credentials after VoiceConsole operator selection. Team-based sign-on goes directly to credential validation using the selected operator's existing credentials. Operator-number sign-on shows an LApp password screen with the operator identifier pre-filled and locked, validates those credentials in the background, and on invalid password returns to the password screen with the password cleared so the operator can retry without repeating operator selection.
- The following RESX key-value pairs have been added to devkit/LAppModule/LAppResources.resx:

Key Name	Value
LApp_GetPassword_Header	Enter Password
LApp_SignOn_Failed_InvalidPassword	Invalid password, please try again

Key Name	Value
LApp_BackgroundActivity_Header_ ValidatingCredentials	Validating credentials

- VoiceLink (devkit/VoiceLinkModule/VoiceLinkLoginStateMachine.cs): VoiceLink now supports operator-number sign-on with password entry, following the same flow as LApp but without the LApp-specific credential validation. The operator identifier is pre-filled and locked on the password screen, and after successful sign-on the credentials are stored for use in API calls from VoiceLink.
- The following RESX key-value pairs have been added to devkit/VoiceLinkModule/VoiceLinkResources.resx:

Key Name	Value
VoiceLink_GetPassword_Header	Enter Password
VoiceLink_SignOn_Failed_InvalidPassword	Invalid password, please try again

- Base Picking (devkit/BasePickingModule/BasePickingLoginStateMachine.cs): Base Picking now supports operator-number sign-on without password entry. After entering a valid operator number, the operator is signed in immediately without a password prompt, and the operator identifier is stored for use in API calls from Base Picking. If customers want to add password validation for Base Picking, they can follow LApp/VoiceLink implementation which has the most complete login flow.
- Simple App (devkit/SimpleAppModule/SimpleAppLoginStateMachine.cs): Simple App now supports operator-number sign-on without password entry, following the same flow as Base Picking. After entering a valid operator number, the operator is signed in immediately without a password prompt, and the operator identifier is stored for use in API calls from Simple App.
- Receiving (devkit/ReceivingModule/ReceivingStateMachine.cs): Receiving does not use the Simplified pattern so it does not support any of the new login features. It continues to use the legacy login flow and is unaffected by these changes.
- The new login changes are specific to Android and iOS devices only. The LApp password-screen/state-machine path is wired through *devkit/LAppGWRunnerModule* (devkit/LAppGWRunnerModule/LAppGWRunnerModule.csproj targets *net9.0-android;net9.0-ios*). Talkman devices use the separate *devkit/LApp.Artisan* app path (devkit/LApp.Artisan/LApp.Artisan.csproj, devkit/LApp.Artisan/LAppArtisanModule.cs) and does not route through **LAppGWRunnerModule**, so Talkman sign-on behavior is unchanged by this update. Same with VoiceLink, which uses *devkit/VoiceLinkModule* (devkit/VoiceLinkModule/VoiceLinkModule.csproj) and is unaffected by the new login changes.

NFC-Based Sign In

> **NOTE**
> This feature can be disabled. See Disable Login Features

NFC-based operator sign-in is available for Android devices by using the SRX3 electronic module tap-to-load. Operators can load their profile by tapping the SRX3 electronic module to the Android device.

- NFC sign-in is triggered by the NFC tag in the SRX3 electronic module, not by the headband. Tapping only the headband does not provide the NFC payload used for operator lookup, so no NFC auto sign-in is performed.
- The headset Bluetooth device address is read from the NFC tag and used to look up the associated operator in VoiceConsole via the new *GetOperatorConfigByHeadsetId* API.
- After a successful NFC read, Bluetooth Out-of-Band (OOB) pairing is initiated automatically by Android system, Pairing and Connecting the headset.
- If SRCOMM is enabled (**PreferSRCOMM** set to true), the audio route is coordinated so that the SRCOMM connection is established after the NFC-initiated Bluetooth pairing completes.
- Configure sign-off behavior for NFC auto-login. Set SkipNfcAutoLogin = true as default after sign-on success, ensuring consistent behavior across workflows by preventing immediate re-trigger of NFC auto-login when returning to login/home screens.
  - This prevents automatically re-signing the same operator from a headset that remains connected after sign-off.
  - This allows the next operator to select a different login path (for example Teams or Number) from Operator Login Options.
  - Workflows can set SkipNfcAutoLogin = false on sign-off when each operator has a single headset association and they want the same operator to be loaded again after sign-off.

  In LAppBusinessLogic, set `SkipNfcAutoLogin = false` on sign-off to allow the same operator to be reloaded via NFC.

```
private Task<CoreAppSMState> ProcessSignOffAsync()
{
    // Disable sign-off immediately so the menu won't reappear while sign-off
processing is underway.
    SignOffEnabled = false;
    SkipNfcAutoLogin = false; // set this to enable NFC auto-login after sign-off.
    // Return to sign-on state (UI / state machine will present sign-on).
    return Task.FromResult(LAppBusinessLogic.SignOnState);
}
```

- Required application change Add `<uses-permission android:name="android.permission.NFC" />`

to your AndroidManifest.xml.

- The following RESX key-value pairs have been added to DevKitResources.resx:

Key Name	Value
Generic_SignOn_Failed_NFC_NoAssociation	No association found for the connected headset. Please load your operator using the Teams option.
Generic_SignOn_NFC_Headset_Association_Confirmation	Headset currently associated with operator {0}. Is this correct?
Generic_SignOn_Operator_Number_Association_Confirmation	Operator Number {0} is associated with operator {1}. Is this correct?
GuidedWork_NfcRequiredDialog_Title	NFC Required
GuidedWork_NfcRequiredDialog_Message	NFC is disabled. Please enable NFC to continue.
GuidedWork_NfcSupportedButDisabledText	NFC is supported but currently disabled.

Disable Login Features

Compile time and runtime configuration supports enabling or disabling Operator Number login and NFC electronic module login. Default is enabled for both features.

- Comment this line in XplatDependencyOverrides.cs to disable Operator Number login.

```
Honeywell.Firebird.WorkflowEngine.OperatorNumberLoginEnabled.Enable(container);
```

- Comment this line in XplatDependencyOverrides.cs to disable NFC electronic module login.

```
Honeywell.Firebird.WorkflowEngine.NfcEModuleLoginEnabled.Enable(container);" in
`XplatDependencyOverrides.cs
```

- Use the OnetimeStartupSettings.config file with the following configuration included to change the app defaults:

```
{
    "Repositories": {
```

```
        "OperatorLoginConfig": {
            "OperatorNumberLoginEnabled": "true",
            "NfcEModuleLoginEnabled": "true"
        }
    }
}
```

Login Screen

The login screen layout is redesigned for an improved user experience. The operator team selection list is now displayed on every login screen. Previously selected teams are no longer automatically restored after logout.

Configure Login

Added an overload for the ConfigureLogin method that enables customization of the screen used to obtain the UserId/Operator. This change also removes the automatic *OperPrep* state that was previously triggered immediately after SignOn in Voice Console, delegating the responsibility of invoking the *OperPrep* state to the workflow itself at the desired step.

Parameters:

- **startState**: The state that is the first state configured
- **destinationState**: The state to return to after the operator name and password are retrieved
- **operIDConfigName**: The name of the configuration setting that stores the resulting operator id
- **operLoginAction**: The asynchronous action to be performed to sign the operator into a service
- **displayLoginState**: The state to display for the login process in case custom screen is needed.
- **retrieveOpersBgndMsgKey**: (Optional) localization key to be displayed while retrieving operators from VoiceConsole
- **signOnBgndMsgKey**: (Optional) localization key to be displayed while signing on an operator

Use case for the overload method

- Use this overload when building workflows that extend SimplifiedBaseBusinessLogic, and
- Want to use a different login Screen than devkit SignIn screen.

Operator Teams Login Options

Operator team option at login has been updated.

- After an operator logs off, the login screen displays all available login options.
- If the back button is pressed during the operator login process, the system returns to the screen displaying all operator teams.
- If an error occurs and all options cannot be displayed, an "Unable to display all login options" error is shown.
- If the back button is pressed and the system does not return to the login options screen, a "Navigation error." error message is shown.

Structured Event Logging

Added structured event logging with VoiceCatalyst-compatible format for Android and Talkman devices.

A new structured event log is added for specific communication and network events. Only the events listed below use this format. Regular application log messages are not affected.

- Events are sent to Log.txt in the Android app's Log directory
- Events are included in VoiceConsole logs for Android and Talkman devices
- The log is formatted so that each structured event gets a timestamp, monotonic millisecond counter (ms since device boot, CLOCK_MONOTONIC), and message following VoiceCatalyst's format:

  ```
  07:01:34.234 - 337726: SURVEY: Signal Strength: 42% ( 28 33 ... )
  ```

  Existing Android DevKit Log format is unchanged:

  ```
  2026-01-12 07:01:34.235 [INFO]  CoreAppStateMachine - [3] Leaving background
  activity state and going to ExecuteSignOnPrep
  ```

  Note that VoiceConsole logs continue to prepend a date/time stamp on each line:

  ```
  (1/12/26 12:01:35 PM UTC) 07:01:34.234 - 337726: SURVEY: Signal Strength: 42% ( 28
  33 ... )
  (1/12/26 12:01:35 PM UTC) 2026-01-12 07:01:34.235 [INFO]  CoreAppStateMachine -
  [3] Leaving background activity state and going to ExecuteSignOnPrep
  ```

Automatic log file entries:

- **Communication events**: HTTP requests/responses and TCP socket data transfers in *RESTQueue* and *TCPSocketQueue*

- Data sent/received with timestamps, byte counts, and status codes
- Works by detecting special COMM_SEND:, COMM_RECV:, and COMM_RECORDS: patterns in log messages

- Network events: for Android devices:
  - **Signal Strength**: RSSI changes converted to 0-100 percentage
    - Throttled to max 1 event per 30 seconds
    - Keeps rolling history of last 10 values
  - **Access Point**: BSSID (MAC address) of connected WiFi access point
    - Requires Location Services (GPS) enabled - refer to the Important note at the end of this section for details
  - **Successful Roaming**: Device switched to different access point while staying connected
  - **Unsuccessful Roaming**: Failed attempt to switch access points
    - - Best-effort detection via *SupplicantState* transitions (some roaming at OS/driver level may not be visible)
  - **WLAN Association**: Band (2.4GHz/5GHz/6GHz), channel, MAC, and RSSI when connecting

Required Application Changes

If you are updating an existing application, you must add the following permissions or the app cannot start:

- AndroidManifest.xml — add the two location permissions:

```
<!-- Required for WiFi BSSID on Android. On Android 12+, COARSE must be declared
before FINE. -->
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

- MainApplication.cs — add the same permissions to the *requiredPermissions* array in *GetAndroidPermissionsRequestorSingleton*

```
// Required for WiFi BSSID (Access Point MAC address)
// On Android 12+, COARSE must be requested before FINE
Android.Manifest.Permission.AccessCoarseLocation,
Android.Manifest.Permission.AccessFineLocation,
```

- If these changes are not made the app cannot start. DevKit treats all permissions in *requiredPermissions* as mandatory. The startup flow blocks until the user grants them or is directed to app Settings.
- Impact on device deployment:

- First-time install / manual install users see an additional permission prompt for Location when launching the app for the first time. This is expected behavior. Accept the prompt otherwise the app blocks at the permission request screen on every launch.

> IMPORTANT
> Granting the app permissions is NOT enough. The device-wide Location Services toggle MUST also be turned ON (**Settings** > **Location** > Use location = ON).
>
> - If Location Services are off, the app logs warnings (GetCurrentBssid() returned null or empty) and Access Point, Roaming, and WLAN events do not work. The either are missing or contain no meaningful data.
> - This is an Android operating system limitation, not an application bug. Android requires both the permission and the system-wide Location toggle to be on before providing the real WiFi BSSID to any app.
> - For MDM deployments: If your organization uses an MDM system to silently grant permissions, you must update your MDM configuration to also grant ACCESS_COARSE_LOCATION and ACCESS_FINE_LOCATION. Without this update, the permissions are not granted, there is no notification, and the app blocks at the permission request screen on every launch. Also, ensure device policy enables Location Services system-wide, not just the app permissions. Consult your MDM install guide for instructions on adding these permissions to your device profile.

Privacy Consent Updates

The user interface for Privacy Consent has been updated. This does not change the functionality of this feature that was first implemented in DevKit 2.4.

Visual Changes

- The screen no longer displays the Honeywell Voice Automated Solutions Privacy Notice (help.honeywellaidc.com/privacy/Content/Privacy.htm).
- The Honeywell Acknowledgement Checkbox now contains a hyperlink to the Honeywell Voice Automated Solutions Privacy Notice.
  - Once clicked, the link opens on the device's native browser
  - If a user rejects the Honeywell Voice Automated Solutions Privacy Notice by clicking the back button, a pop-up is displayed informing them that they must accept the notice to continue with the application.
  - If the link can not be accessed (see section below), a pop-up is displayed containing a QR code for the user to scan. The user can scan this code from a personal devices and is directed to the Honeywell Voice Automated Solutions Privacy Notice.

Link Accessibility

The Honeywell Voice Automated Solutions Privacy Notice link may not be reachable in the following scenarios:

- The device's browser has been disabled via MDM.
- The network does not allow for outside navigation.
- The link has been disabled via OnetimeStartupSettings.config or XplatDependencyOverrides.cs. See section below.
- An error occurred while launching the browser.

In all of these scenarios, the pop-up containing the QR code is displayed.

Disable Embedded Privacy Notice Link

> NOTE
> These actions do not disable the feature, only the embedded link on the Privacy Consent screen. If this is disabled, clicking the link displays the QR code pop-up.

The embedded Privacy Notice link can be disabled by either of the following methods:

- Uncomment the line below in XplatDependencyOverrides.cs:

```
Honeywell.Firebird.WorkflowEngine.PrivacyNoticeLinkDisabled.Disable(container);
```

- Use the OnetimeStartupSettings.config file with the following configuration included:

```
{
    "Repositories": {
        "PrivacyConsentConfig": {
            "PrivacyNoticeLinkDisabled": "true"
        }
    }
}
```

RESX Updates

The following RESX key-value pairs have been removed from to GuidedWorkResources.resx:

Key Name	Value
PrivacyConsentHoneywellAcknowledgementText	I consent to the collection, use and processing of my personal data, which may include speech templates, voiceprints, and other biometrics, for data analytics and improvement of Honeywell's products and services, as detailed in Honeywell's Voice Automated Solutions – Operator Privacy Notice, available online. I acknowledge that I have read and understood that notice, including my right to withdraw consent and to have my personal data deleted.

The following RESX key-value pairs have been added to GuidedWorkesources.resx:

Key Name	Value
PrivacyConsentHoneywellAcknowledgementTextStart	I consent to the collection, use and processing of my personal data, which may include speech templates, voiceprints, and other biometrics, for data analytics and improvement of Honeywell's products and services, as detailed in
PrivacyConsentHoneywellAcknowledgementTextLink	Honeywell's Voice Automated Solutions – Operator Privacy Notice
PrivacyConsentHoneywellAcknowledgementTextEnd	available online. I acknowledge that I have read and understood that notice, including my right to withdraw consent and to have my personal data deleted.
PrivacyConsentActionRequiredText	ACTION REQUIRED! Select the checkboxes below to consent and Continue.

The following RESX key-value pairs have been added to GuidedWorkCoreResources.resx:

Key Name	Value
PrivacyConsent_Alert_QR_Code_Header	Scan QR Code

Key Name	Value
PrivacyConsent_Alert_QR_Code_Label	Honeywell Privacy Notice
PrivacyConsent_Alert_ConsentRequired_Header	Consent Required
PrivacyConsent_Alert_ConsentRequired_WarningLabel	You must consent to continue working.
PrivacyConsent_Alert_ConsentRequired_QuestionsLabel	If you have any questions about providing consent, please contact your supervisor.
PrivacyConsent_Alert_Close	Close

Android Device Serial Numbers

An additional section is necessary in the AndroidManifest.xml file for retrieving serial numbers from Zebra devices using Android 11 and newer when using the AccurateSerialNumber feature. This section is added to AndroidManifest.xml in the generated XML using Android's AndroidManifestOverlay feature.

The following section is added:

```
<queries>
  <provider android:authorities="oem_info" />
  <package android:name="com.zebra.zebracontentprovider" />
  <package android:name="com.symbol.emdk.emdkservice" />
</queries>
```

For more details on AndroidManifestOverlay see:

- learn.microsoft.com/en-us/dotnet/android/building-apps/build-items#androidmanifestoverlay
- developer.android.com/build/manage-manifests

Overflow Menu VoiceEnabled

A new *VoiceEnabled* property is added to *InteractiveItem*. The default is true.

- Overflow menu items with *VoiceEnabled* = false are shown in the overflow menu UI but are not loaded into the speech recognizer as out-of-band words. These items can only be activated by tapping.
- Existing items are unaffected because the property defaults to true.

- For a custom overflow menu item that should be tap-only (for example, a destructive or safety-critical action), set *VoiceEnabled* = false on that InteractiveItem.
- On Talkman, items with *VoiceEnabled* = false are also excluded from the additional vocabulary dictionary built by `_get_additional_vocab` in the VAD.

SQLite Dependencies

- Removed SQLite dependency from Receiving workflow: Replaced database implementation with in-memory storage using new *InMemoryDatabaseContainer* and *InMemoryDatabase* classes. The SQLite-based DatabaseContainer class was removed. Note that data is not persisted across executions. This code is for demonstration purposes only.
- Removed SQLite dependency from configuration repositories: Configuration repositories are now kept in memory and persisted as JSON files.
  - Automatic migration occurs on first launch after upgrade. Existing configuration data will be transferred from SQLite to the new JSON format. The SQLite database is deleted after successful migration.
  - Note that the configuration service is designed for application settings and preferences, not runtime data storage.
  - Persisting data in configuration repos on Talkman devices is not supported. If your deployment has been using configuration repos to persist data on a Talkman device, that data is not migrated to the new format and is available after upgrading. Contact Technical Support if you need assistance.

Device Support

Support is added for the following devices:

- Honeywell A700XP with VoiceCatalyst 5.0 or later
- Honeywell CT70 with Android 15

Miscellaneous Changes

- Updated AsrEngine version from 1.6.5 to 1.7.4, which is built with improved security in compiler and options and includes:
  - Fixes for simulator builds of iOS apps on M-series Macs.
  - Updates to support Embedded Training.
- Added compiled bindings (see learn.microsoft.com/en-us/dotnet/maui/fundamentals/data-binding/compiled-bindings?view=net-maui-9.0) to devkit folder views:
  - BasePickingGWRunnerModule\Controllers\BasePickingConfigurationSettingsController.cs
  - BasePickingGWRunnerModule\ViewModels\BasePickingConfigurationSettingsViewModel.cs

    - BasePickingGWRunnerModule\Views\XamarinPageViews\BasePickingConfigurationSettingsView.xaml
    - BasePickingGWRunnerModule\Views\XamarinPageViews\BasePickingServerSettingsView.xaml
    - LAppGWRunnerModule\Views\XamarinPageViews\LAppSettingsView.xaml
    - ReceivingModule\Views\XamarinPageViews\ReceivingView.xaml.cs
    - SimpleAppGWRunnerModule\Controllers\SimpleAppSettingsController.cs
    - SimpleAppGWRunnerModule\ViewModels\SimpleAppSettingsViewModel.cs
    - SimpleAppGWRunnerModule\Views\XamarinPageViews\SimpleAppSettingsView.xaml
    - VoiceLinkGWRunnerModule\Views\XamarinPageViews\VoiceLinkServerSettingsView.xaml

  Along with those changes, note a new ServerSettingsVisible property use in these files:

    - SimpleAppGWRunnerModule\ViewModels\SimpleAppSettingsViewModel.cs
    - SimpleAppGWRunnerModule\Controllers\SimpleAppSettingsController.cs
- ConfigurationDataService now allows processing settings when a configuration repository for that category is not registered. In this scenario, the configuration is directly applied using the ConfigService, using the provided key as configuration category.
- Fixed a are situation where whitespace trimming was not preserved for non-boolean configuration values. Trimming already existed but the result was discarded for non-boolean values. This could impact values with leading whitespace such as ' ENC=...', which is now correctly trimmed and decrypted when they were previously been stored with the whitespace intact.
- Partially revert the VGWSC-7686 bug fix delivered in DevKit 2.5 (see the DevKit 2.5 Release Notes). This keeps the behavior where the “Sign off” option is removed from the overflow menu after pushing the ApplicationSettings.config file. This occurs when SignOff is included in the DisabledVocabWords list in ApplicationSettings.config file.

Breaking Changes

- Removed support for mapping workflow names to work item types in initial_wfa.json
- StateMachine *PrepSelectTeamDisplay* and triggers *PrepSignOnDisplaySelectTeam*, *PrepSignOnDisplayFailed* have been removed as part of the login screen redesign. If you have a customized login workflow that uses these states or triggers, you need to update your workflow to remove any references to them and implement the new login screen design.
- Enhanced SSL pinning validation with improved logging.
    - *PinningUtil* now requires constructor dependency injection
        - *PinningUtil* no longer exposes a static `_ServerConfigRepository` field
        - The following method overloads have been removed

        - *VerifyPinnedPublicKey(string currentPubKey, string serverAuthority, IServerConfigRepository serverConfigRepository)*
        - *PinningValidation(HttpRequestMessage sender, X509Certificate2 cert, X509Chain chain, SslPolicyErrors error, IServerConfigRepository serverConfigRepository)*
    - If your code called these overloads, you must now instantiate *PinningUtil* via constructor injection and call the remaining public methods without the *IServerConfigRepository* parameter.
    - Code must now instantiate *PinningUtil* via constructor injection:

```
// New approach - constructor dependency injection (required)
// Example with additional dependencies (from VIOTCPSocketQueue)
public class VIOTCPSocketQueue : TCPSocketQueue, IVIOTCPSocketQueue
{
    private readonly PinningUtil _PinningUtil;

    public VIOTCPSocketQueue(
        IFileSystem fileSystem,
        ICompatFormatter formatter,
        PinningUtil pinningUtil,          // ← Inject here
        TimeProvider? timeProvider = null)
        : base(fileSystem, formatter, pinningUtil)
    {
        _PinningUtil = pinningUtil;  // Store if needed for custom certificate
validation
    }

    // Use the injected instance in your custom validation logic
    private bool ValidateServerCertificate(
        object sender,
        X509Certificate? certificate,
        X509Chain? chain,
        SslPolicyErrors sslPolicyErrors)
    {
        // ... validation logic ...
        return _PinningUtil.VerifyPinnedPublicKey(publicKeyString, HostServer);
    }
}
```

- Deprecations and Future Breaking Changes
    - The Responded event in *IGuidedWorkRunner* is now marked as Obsolete and is to be removed in future versions.

- The following classes and methods no longer perform any function and thus have been marked Obsolete and are being evaluated for removal in future versions:
  - *SubmitScanResult* method in *GuidedWorkScannerView* and *GuidedWorkBluetoothScannerView*.
  - *GuidedWorkValidatingScannerView* as well as its *SubmitScanResult* method.
  - *GuidedWorkValidatingBluetoothScannerView* as well as its *SubmitScanResult* method.
  - *GuidedWorkMenuItemScannerView* as well as its *SubmitScanResult* method.
  - *GuidedWorkMenuItemBluetoothScannerView* as well as its *SubmitScanResult* method.
  - *GuidedWorkSignInScannerView* as well as its *SubmitScanResult* method.
  - *GuidedWorkSignInBluetoothScannerView* as well as its *SubmitScanResult* method.

> **IMPORTANT**
> Contact Technical Support (voicetechnicalsupport@honeywell.com) if you need access to any of these classes for your applications.

Issues Fixed in this Release

> **NOTE**
> Resolved issues are included in the CHANGELOG.md file in the devkit directory, included in the DevKit zip file.

Issue Description	Issue ID
GWS Apps Permission Error The GWS App could prompt for permissions after the initial install (or after clearing the app data) then display a "Permission Required" message even though the permissions were granted. Add the 'PostNotifications' permission in *GetAndroidPermissionsRequestorSingleton* method in MainApplication.cs. This ensures that notification permission is checked during app start up. Refer to devkit\GWS.Android\Platforms\Android\MainApplication.cs for details. The value of the key 'Text_PermissionsRationale' in the DevKitResources.resx file for all GWS App supported languages has been updated to include notification permission.	VGWSC-13825
GWS App Min/Max Length The GWS App GetFloat instruction did not work properly with min/max lengths set when the delimiter (selected by language) was a symbol other than dot '.'.	VGWSC-13005
GetDigits Negative Vocab When the GWS App got the instruction for GetDigits and AllowNegative is true, the intent created is adding '-' to the AllowedCharacterSet which prevented negative vocab from being recognized.	VGWSC-12737
Sign Off at Date Intent On a Talkman device, trying to sign off at a date intent was unsuccessful as it did not sign off and proceeded to the next intent.	VGWSC-12708
GWS App Settings Not Saved On an Android device, if a user made changes on the Application Settings screen then pressed the Back button, those changes were not saved.	VGWSC-12652

Issue Description	Issue ID
App Crash when no License is Available When licensing using VoiceConsole SaaS and no license is available, enabling auto release of lease then pressing the Ready button can crash the app. This change added new resource keys in GuidedWorkCoreResources.resx: Key Name: VoiceConsole_LicenseConfiguration_Missing Value: VoiceConsole license configuration is missing. Please contact your supervisor. Key Name: VoiceConsole_LicenseCheckFailed Value: VoiceConsole license validation failed. Please contact your supervisor.	VGWSC-12506 VGWSC-7749
Scanning Sign On Scanning a sign on incorrectly converted all user name characters to lowercase. In addition to preserving case, a new error message prevents a crash if the username is not found. This change added new resource keys in GuidedWorkCoreResources.resx: Key Name: Generic_SignOn_Failed_Invalid_Credentials Value: Sign-in failed. Please verify your credentials and try again.	VGWSC-12470 VGWSC-11948
Error Reloading Application When a task package was reloaded to a Talkman device an error code 1104 could occur.	VGWSC-12460
DelayTime not Set If DelayTim wasn't set to zero for each intent it would override the DigitEntrySpeechDelay and cause it to default to 0.700 (700 ms). DelayTime now defaults to 0 for Talkman devices and 700 ms for Android devices.	VGWSC-12450
Min/Max Decimal Places from Scanner Value When an Android device provided scanned input it was not properly validated for minDecimalPlaces and maxDecimalPlaces.	VGWSC-12382
Headset Boom Events not Handled Raising or lowering the microphone boom during a workflow transition the boom rotation event was lost and never processed.	VGWSC-12237

Issue Description	Issue ID
Race Condition after Raising Boom Raising the microphone boom quickly after an utterance could cause the application to remain in standby state without presenting the standby screen/prompt to the user.	VGWSC-12236
Pick Up & Go not Activated for Chinese The logic was not properly checking for GWExternalResources.zh-CN.resx, GWExternalResources.zh-HK.resx, or GWExternalResources.zh-TW.resx resource files.	VGWSC-12051
Sign Off from Overflow Menu Tapping Sign Off from the overflow menu caused an invalid intent when sign off was included in the disabled vocab list.	VGWSC-11995
Smart Talk Connection not Made After a successful call from Smart Talk (SRCOMM enabled), after the call is disconnected, trying to make another call without switching back to Guided Work app is unsuccessful.	VGWSC-11992
Unable to Hear Voice in Smart Talk Call If Operator 1 is in a call with Operator 2 and Operator 1 turns the headset off then on, operator 1 can hear operator 2 after the headset reconnects, but operator 2 cannot hear operator 1.	VGWSC-11991
Overflow Menu The overflow menu sometimes contained items that did not make sense and selecting those items may put the device into a state where only the Back button could recover.	VGWSC-11951
Sign Off Overflow Menu Option The Sign off option was available on the Overflow Menu on screens where it should not be an available option.	VGWSC-11949
GetLongFloat Instruction The GetLongFloat VIO instruction allowed input values without any digit after the decimal point (such as '12.') without triggering a validation a validation message when UseFloatValidation was set to true.	VGWSC-11884

Issue Description	Issue ID
Not Validating Min/Max Length On a Talkman device using GetDigit VIO instructions, the app was not performing the min/Max length validation before the min/max range validation.	VGWSC-11776
Invalid Value Accepted In GetDigits instruction, if an invalid value is entered and the user executes a command during the invalid value prompt, the invalid value was accepted and the command execution continued.	VGWSC-11267
Check Digit Field Overlaps with Header The Check Digit input field could be overlapped by the header when the keyboard opens.	VGWSC-10932
Headset Switch on Home Screen If a user is on the Home screen, switching headsets displays a prompt to return to the home screen.	VGWSC-10129
Volume Change After Headset Disconnect Cycle On Android devices, the headset volume could be reduced or muted after a disconnect/reconnect cycle.	VGWSC-9848
Loudness Difference There was a difference in actual loudness using Play Example if the loudness was set from Voice and Audio settings than with using the overflow menu to set the same loudness setting.	VGWSC-7713
Main Menu Prompt not Replayed In MenuItem Intents after an invalid option was spoken, only the error message was played and repeated when an operator speaks “say again” instead of replaying the main menu prompt.	VGWSC-7680

Issue Description	Issue ID
Help Message Arrangement The help message was not arranged properly, and now begins with the instruction-related guidance, then lists the characters that can be spoken along with the anchor word, and ends with the section containing additional commands. This change added a new resource key in GuidedWorkCoreResources.resx: Key Name: default_help_multiple_words_and_one_anchor Value: followed by {0}	VGWSC-7679
Error on Headset Connect/Disconnect Connecting and disconnecting a Bluetooth headset to an Android device can result in an SRX SDK exception.	VGWSC-7672
GWS App Notification Permission On initial launch the application asks for GWS App notification permission without re-prompting if the user does not allow. This change added new resource keys in GuidedWorkResources.resx: Key Name: Text_EnablePermissionsInSettings Value: Permissions have been permanently denied. Please enable them in Settings. Key Name: Text_GoToSettings Value: Go to Settings	VGWSC-7655
Operator Team Name If a single letter was used as the Team Spoken Name from an Android device, the following error was displayed: "Error in voice dialogue, please report problem. Use screen to complete current task and continue".	VGWSC-7650
Boom Lift While Recording Lifting the microphone boom while recording stops the recording but does not send the app into standby. This can result in prompts to such as "Say Ready to begin recording" but any responses is not recognized until the boom is lowered.	VGWSC-7638

Issue Description	Issue ID
Menu Item Selection Order When using the GWS App on an Android device, selecting multiple menu items in non-seqential order still displayed the options on the prompt in numerical order. (for example: 1,2,3,4 rather than as selected 4,2,1,3)	VGWSC-7637
Noise Sample without Headset If a workflow was started with no headset attached, the Android device displays the "No Bluetooth Headset Connected" screen but continues the noise sample rather than the standby by message.	VGWSC-7632
iOS Overflow Menu Items in the overflow menu were not always anchored in the correct location.	VGWSC-7490
"Standing by" not Spoken Flipping up the microphone boom causes the display to correctly show standby mode but the TTS does not announce "Standing by".	VGWSC-7363
Operator Team List not Sorted Alphabetically The DevKit application operator is sorted alphabetically, however in the operator team list they are not.	VGWSC-7361
Operator Switch Prompts Restart When a task switched between two operators, there was no issue. But switching to a third operator could case this error "Error occurred, see your supervisor. To restart application say Ready"	VGWSC-7359
UI Elements Horizontally Offset Items displayed in the App can be horizontally offset by a large gap, causing items to be split and shown on two lines.	VGWSC-7341
GetFloat Confirmation The GetFloat confirmation could speak "instead of" for the decimal separator in non-English locales.	VGWSC-6847
Some GWS App Messages Always in English Localization was not provided for certain phrases.	VGWSC-5066

Issue Description	Issue ID
Anchor Words and Talkman Help When using anchor words in the GWS App, the Talkman Help commands did not send the list of Anchor Words that can be spoken at the prompt. This was not a problem on Android devices.	VGWSC-4311
Error 1101 Sent when Operator is Changed in Workflow When the operator is changed in the middle of the workflow, the application sends an 1101 error before restarting the application to continue with the new operator.	VGWSC-4226
Connection Refused Error Talkman device logs may show this message "Receiving socket error: ConnectionRefusedError(111, 'Connection refused')" before the application has started. This message was replaced with the more appropriate "Waiting for GuidedWork app to start...” message.	VGWSC-4008
No Message Sent The NotAllowedPrompt did not send a message if it was empty and now sends a default message.	VGWSC-1339
Application Loop in Operator Selection If no operators were created in VoiceConsole and VoiceConsole was used for licensing, an Android device could loop after selecting all operators in the GWS App instead of giving the "Unable to retrieve operator files" message like a Talkman device does. This change added a new resource key in GuidedWorkCoreResources.resx: Key Name: VoiceConsole_NoOperatorsAvailable Value: No Operators Available	VGWSC-1091
Natural Number Prompts Although the app recognizes natural numbers such as 100, when the device speaks the number it says one zero zero.	VGWSC-777

System Requirements

The following devices and software were tested for this release of DevKit.

Minimum Android Device Specifications

- **Processor**: Qualcomm Snapdragon 410 MSM8916 1.2 GHz quad-core
- **Memory**: 2GB RAM
- **Storage**: 8GB/16GB Flash
- **WLAN**: EEE 802.11 a/b/g/n radio
- **Bluetooth**: Bluetooth Class 4.0, Bluetooth HFP (Hands-Free Profile) version 1.6
- **Operating System**: Android 11

The list above is the minimum recommended specifications. If the device does not meet or exceed these specifications, the following symptoms may occur:

- Poor audio quality
- Slow application screen responsiveness
- Delayed input entry

Hardware

The following devices were tested for this release.

Honeywell Android Devices

Device	Supported Android Versions	SRCOMM Supported SKUs *	Landscape Support
CK65	11	L0N	No
CK67 (Model CK67X1N)	14		No
CN80			No
CN80G			No
CT30 XP	11	L1N, X0N	No
CT37	14	X1N	No
CT40	11, 13		No
CT40 XP	11, 12		No
CT45	11, 12, 13	X0N	No
CT45 XP	13		No
CT47	12	X1N	No
CT60	11, 13		No
CT60XP	11, 13		No
CT70	15		No
CW45	12		Yes
ScanPal EDA57	12		No

* SRCOMM was tested with these particular SKUs. Because SRCOMM support is dependent on the installed radio only these SKUs are supported. An SRX3 headset is required to support SRCOMM with these devices.

Honeywell Talkman Devices

Device	VoiceCatalyst Version	SRCOMM Supported
A700XP	5.0 or later	Yes
A700x	4.7.1 ECS012 or later	Yes

Zebra Android Devices

Device	Supported Android Versions	SRCOMM Supported	Landscape Support
WT6300	11, 12, 13, 14	No	Yes
TC5x	11, 12, 13	No	No
TC7x	11, 12, 13, 14	No	No

Headsets

- Honeywell SRX-SL Light Industrial Use Headset
- Honeywell SRX2 Wireless Headset (with Hands-Free Profile support)
- Honeywell SRX3 Wireless Headset
  - Firmware version 6.09 or later is required to support SRCOMM with BlueStreak speech recognition on Android devices.
  - Firmware version 6.13 or later is required to support SRCOMM with Pick Up and Go speech recognition on Android devices.

Scanners

- On-board scanners for Honeywell Android devices listed above
- Honeywell A730x
- Honeywell A730XP
- Honeywell 8670 Ring Scanner

Printers

- Honeywell RP2d
- Honeywell RP4d
- Intermec PB22
- Intermec PB50
- Zebra QL320 Plus
- Zebra QL420 Plus

> **NOTE**
> Honeywell A700 series and A500 devices are not supported.

Management Server Support

- VoiceConsole 5.6.3. or later for A700x.
- VoiceConsole 5.6.3, 5.6.5, 6.3.2, and 6.4 or later for A700XP. An ECS is required for 5.x versions.
- VoiceConsole 6.2.1 or later for Android device support

Issues Reported in this Release

Issue Description	Issue ID
Standby Screen not Displayed after Headset Disconnect If the Bluetooth headset is disconnected (or Bluetooth is turned off) the Standby screen is not displayed and either a "Noise Sample Failed" or "Training Paused" message is displayed. **Workaround**: After either the "Noise Sample Failed" or "Training Timed Out" screen reconnect the headset and the workflow resumes.	VGWSC-14149
Noise Sample Failed During Template Training When using SRCOMM if the Bluetooth headset goes out of range during template training and later reconnects the application fails to resume training and displays "Application Noise Sample has failed" and does not proceed. **Workaround**: Force close (kill) the application and restart the training process.	VGWSC-13725

Issue Description	Issue ID
Serial Number not Displayed When AccurateSerialNumber is enabled and a device that does not exist in VoiceConsole is added to VoiceConsole, the screens may not reflect the device serial number. **Workaround**: Closing and reopening the app forces VoiceConsole to properly display the serial number.	VGWSC-13160
No Error when User Speaks "dash dash" When using GetLongDigit and GetLongFloat with AllowNegative no error message is given when the user speaks dash twice, such as "--12". Instead the application returns to the same VIO instruction.	VGWSC-5304

Previously Reported Issues

Issue Description	Issue ID
Smart Talk Headset does not Connect After a first call is made from Smart Talk, the headset may not connect to the Android device (headset shows green blinking light) on a subsequent call from Smart Talk. **Workaround**: Make the second call using the make call option from the overflow menu in the app.	VGWSC-11992
Smart Talk Loss of Audio If a Smart Talk call is in progress and the headset disconnects and reconnects during the call for operator 1, operator 2 can no longer hear operator 1. **Workaround**: Disconnect the call and make the call again.	VGWSC-11991

Issue Description	Issue ID
SXR3 Headset Low Audio Certain Android 14 devices using HFP mode with SRX3 headset may occasionally have very low volume after the headset loses connection with the device and reconnects. Honeywell has observed this on CT32 and CT37 using Android 14 MR1 and later and it may be seen on other devices. **Workaround**: The following workarounds are available: • Reboot the Android device • Load a different version of Android OS (If available) • Use SRCOMM instead of HFP	VGWSC-9848
Unable to Install App If GWS App, GuidedWorkDemo App or Warehouse Demo is already installed trying to install a second one of these apps results in the following error message: **adb: failed to install .<*package name to be installed*> Failure [INSTALL_FAILED_DUPLICATE_PERMISSION: Package.<*package name to be installed*> attempting to redeclare permission com.hon.smarttalk.permission.RECEIVE_CONDITIONS already owned by <*package name already installed*>]**. **Workaround**: Delete the original app that is holding the Smart Talk permission and install the new app.	VGWSC-7360
Split Screen not Supported DevKit does not support split-screen mode on Android devices. **Workaround**: If a DevKit based app is opened in split-screen made close the app and restart.	VOSMB-14357
Vocab Duplicated Error Message When selecting an Operator Team from the Android device, selecting one with a single letter spoken name can result in the following message: "Error in voice dialogue, please report problem. Use screen to complete current task and continue" **Workaround**: Do not use single letters (A-Z) as operator Team Spoken Name.	VOSMB-13671

Issue Description	Issue ID
Application Not Switching Language When changing the application from one language to another, the TTS continued to speak the original language on an iOS device. **Workaround**: Navigate to the iOS language settings and switch the device to the new language	VOSMB-13173
SRCOMM: Headset Reconnect Log Entries Reconnecting the headset from the Headset Not Connected screen creates extraneous entries in the log file.	VOSMB-12844
SRCOMM: Switching to PnG may Fail If a user has a headset connected using SRCOMM in their workflow, turns the headset off, then exits the workflow and tries to enable Pick Up & Go (PnG), the result is a headset not connected message. When the headset is turned on, it connects using SRCOMM rather than HFP. **Workaround**: Toggle PnG while the headset is connected. Navigate to the home screen and select Ready.	VOSMB-12840
Not All Words Shown During Training The display dialog screen may not show all words during training and the training bar may jump to 100%	VOSMB-12801
SRCOMM: Not Available for Prompts Outside Workflow SRCOMM is not available for speaking TTS prompts outside the workflow, such as the prompts from the Voice & Audio Settings screen. A2DP is used instead.	VOSMB-12793
SRCOMM: ValueIntent Vocab Lost When using SRCOMM and GetValue intent without anchor words the GuidedWorkViewModel response property updates the Spoken Value and the most recent recognized vocab is lost. **Workaround**: Speak the utterance again.	VOSMB-12735
Select Operator Team Select Operator Team only appears the first time the app is used. **Workaround**: Users can access this option by pressing the Back button on the login screen.	VOSMB-12723

Issue Description	Issue ID
Progress Bar not Visible Navigating away from a screen that shows a progress bar (such as Update Train that has started, Noise Sample or Voice Training) to another screen such as Report Problem or Device Info does not show the progress bar when returning to that screen.	VOSMB-12707
Application Styling The Guided Work application may not have the expected styling.	VOSMB-12573
SRCOMM: Headset not Functioning During an extended period of non-use a headset connected via SRCOMM the headset may remain connected (flashing blue) but not functioning. **Workaround**:	VOSMB-12479
REST Error When Switching Licensing If switching licensing from VoiceConsole to Microservices, the following REST error may be displayed in the logs: Requests cannot be sent while RESTService is disabled. **Workaround**: None needed. The application works as expected despite the error logged.	VOSMB-12356
Vocab Training in Wrong Language If training is started and interrupted, and a different operator signs onto the device with a different language, the new operator may be asked to train words in the original language.	VOSMB-12318
SSL Connection Errors .NET 6 and later do not support weaker ciphers that were supported in .NET Core 3.1. This is an item that was omitted from the breaking changes list in DevKit 2.0.	VOSMB-12144
Site ID Change Once an Android devices has connected to VoiceConsole the device cannot be moved to a different site by downloading a new config file with a different Site ID. **Workaround**: Move the device manually or delete the device from the old site.	VOSMB-11691

Issue Description	Issue ID
iOS App not Supported in Background DevKit 2.0 does not support running the application on iOS in the background or while the screen is off. When the app is in the background, prompts and commands are not processed. NOTE This defect was first found in an earlier version of DevKit. Honeywell has confirmed the issue exists at least as far back as DevKit 1.9.1. **Workaround**: Running the app in the foreground is the only supported mode for iOS.	VOSMB-11147
Sign Off not in Overflow Menu When Sign Off is disabled as a Vocab Word it is also removed from the overflow menu.	VOSMB-10898
Voice Input Disabled When a vocab word is disabled by pushing an ApllicationSettings.config file voice input is disabled in all workflows.	VOSMB-10897
Hints Hints/response instructions are expected to work reliably for digits and alphabetic characters on Android and iOS. Other vocabulary words are not expected to work as part of hints.	VOSMB-8168
Do Not Press + and - Buttons on SRX3 Headset to Unpair When unpairing your SRX3 headset from an Android device, do not press the + and - buttons simultaneously. This procedure causes known issues with TTS and speech recognition. **Proper Procedure**: To unpair your SRX3 headset from an Android device, go to your device Bluetooth settings, display the Paired devices screen, select the headset, and tap FORGET to unpair the headset and the Android device.	VOSMB-1448
Scanning Data Can Override Priority Prompts Priority prompts can be overridden by scanning data at a screen where scanning is a valid input.	VOSMB-1415

Issue Description	Issue ID
Voice Dialogue Continues to Run During Background Activity When waiting for background spinner activity to complete, speech recognition may allow the operator to use some menu items through voice, e.g., Say Again and Help.	VOSMB-1406
Pressing Power Button on SRX-SL or SRX2 Headset Causes Unpairing If you press the Power button on your SRX-SL or SRX2 headset, the headset unpairs from the device. **Workaround**: Power off the headset and re-pair your device and headset.	VOSMB-1252
"GatewayTimeout" error message from Microservices If the mobile application has trouble reaching the Microservices host ("GatewayTimeout") while attempting to retrieve templates it may force the user to retrain all words for that workflow. **Workaround**: If you experience this behavior close and restart the app and log in again	VOSMB-971
State Machine Recommendation Do not create a state machine where the first state goes directly into a secondary state machine and sets the next trigger for return. The app becomes unresponsive when returning from the secondary state machine and the second state never runs.	VOSMB-465

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