Honeywell

Release Notes
Honeywell Development Kit 2.0

October 30 2024

Verifying Download Files

The release files for Honeywell Development Kit (DevKit) 2.0 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-2.0-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

Framework Updates

DevKit now uses the MAUI (Multi-platform App UI) and .NET 8 framework.

- Xamarin.Forms, Xamarin.Android, and Xamarin.iOS are no longer used.

- DevKit no longer supports .NET Core 3.1 on A700x.

- .NET 6 continues to be supported on A700x

Visual Studio 2022 is required for development. Visual Studio 2019 is no longer supported.

MAUI and .NET 8 are growing products that are still in different stages of development with many service releases and versions. Therefore Honeywell is recommending specific versions that are considered the most stable and compatible with DevKit 2.0.

Honeywell recommends:

- Visual Studio 2022 Pro LTSC v17.9.6

- .NET MAUI 858 123 4567 (this is set in Directory.Build.props)

- .NET 8.0 SDK 8.0.200 or later

- A700x devices run VoiceCatalyst 4.6 ECS001 minimum, 4.7.1 ECS005 or later is recommended

See Developing with DevKit 2.0 later in this document for more information.

Standby Screen

A standby screen is displayed if the Bluetooth headset is disconnected at any point in the workflow while speech recognition is enabled. The screen disappears when the headset reconnects successfully.

Bluetooth Printing

DevKit supports Bluetooth printing for A700x. A new *DirectControlLanguagePrinter* class is created and supports two public print methods:

- **Print**: This method allows the caller to receive a success or failure notification via an asynchronous callback

- **PrintAsync**: This method allows the caller to wait for the print request which returns a Task<bool>.

Honeywell has tested this feature with the following printers:

  - Honeywell RP2d

  - HoneywellRP4d

  - Zebra QL320 Plus

  - Zebra QL420 Plus

Device Support

A700x

- DevKit no longer supports .NET Core 3.1.

- A700x devices must support .NET 6.0. This requires VoiceCatalyst 4.6 ECS001 minimum, however VoiceCatalyst 4.7.1 ECS005 or later is recommended.

- VADs for A700x must use the previously optional VAD size reduction/two modules concept. With MAUI, these VoiceArtisan-based VAD files cannot safely contain references to NuGet packages. Therefore, in multi-device (Android or iOS with A700x) environments, the business logic must be separated from the UI elements. This results in a single module for the business logic, a UI elements file for A700x, and a UI elements file for Android and iOS.

Android

- Support is added for the Honeywell CT47 with Android 12.

- DevKit no longer supports Android N (7.x) devices.

- Android O (8.0) is the minimum supported version.

iOS

- This release includes beta support for iOS devices. While the testing on iOS devices has been less thorough, it can be considered as a proof of concept.

Developing with DevKit 2.0

> **IMPORTANT**
> Additional details on developing with DevKit 2.0 and migrating from DevKit 1.x can be found in the help distributed with DevKit.

Changes

- DevKit has changed how REST and TCP requests are persisted for later retransmission. This is because .NET has dropped support for the *BinaryFormatter* class on Android and iOS. *BinaryFormatter* was used to serialize requests to persistent storage on DevKit 1.x. DevKit apps for all Android and iOS devices do not send requests persisted with *BinaryFormatter* when upgrading a DevKit app from a 1.x DevKit version to a 2.x DevKit version. Honeywell recommends carefully managing updates of DevKit apps on Android/iOS devices, using the old version of the app to transfer any pending persisted requests before upgrading to DevKit 2.x. A700x devices do not permit changing .TAS or .VAD files when persisted requests exist and do not have this issue. Replace any references of *IFormatter* in code to instead use *ICompatFormatter*. The changed files in sample apps are
devkit\BasePickingModule\Services\Communications\BasePickingTCPSocketQueue.cs
and devkit\VoiceLinkModule\Services\Communications\VoiceLinkTCPSocketQueue.cs
For more information on the changes to *BinaryFormatter*, see
https://github.com/dotnet/designs/blob/main/accepted/2020/better-obsoletion/binaryformatter-obsoletion.md
and https://learn.microsoft.com/en-us/dotnet/standard/serialization/binaryformatter-security-guide
- An update is required to support RT10 screen rotation when started up in the dock because Microsoft.Maui.Essentials has replaced Xamarin.Essentials. The previous instructions are now updated to change

```
if (DeviceInfo.Idiom.Equals(DeviceIdiom.Tablet))
```

to

```
if (DeviceInfo.Idiom.Equals(DeviceIdiom.Tablet) || (DeviceInfo.Idiom.Equals
(DeviceIdiom.Desktop) && IsTablet()))
```

- Removed/updated some comments, which included internal references.

- Updated the UI layout in .NET MAUI for:

  - Receiving, Retail, workflow to match previous releases that were based on Xamarin.Forms, as well as fixed the indentation in XAML code files for the same workflow.

  - Spacing and HeightRequest properties of StackLayout

- Updated the CustomPicker control, in order to fix the known MAUI picker control's bug that causes it to open automatically when navigating between different screens, or when acquiring and/or losing focus in other controls within the same screen.

- Removed unused RESX entries.

- General latency reductions on Android and A700x. Be aware that the MAUI Android debugger performs noticeably slower than the Xamarin debugger, but Release builds on Android should be roughly the same speed or faster compared to DevKit 1.x.

- Added a new method, *RequestWorkflowObjectContainerAsync* to *GuidedWorkRunner.IMobileDataExchange*, along with default implementations for the interface and all concrete implementations provided in DevKit NuGets. The method is used to improve performance in some cases in VAD files. If you use a custom implementation of *GuidedWorkRunner.IMobileDataExchange* for VADs, consider overriding the interface-provided method to improve performance. Most customers will not need to do this.

- Voice Console Cloud Deployment has changed to Voice Console SaaS Deployment. Below are the RESX keys that have changed and their new values:

  - **VoiceConsoleCloudBased**: Voice Console SaaS

  - **VoiceConsoleCloudBased_Info**: Licensing is done via VoiceConsole SaaS, templates are retrieved from and saved to VoiceConsole SaaS and logs are stored on the file system as well as sent to VoiceConsole SaaS when logging is enabled.

- Updated the Android minSdk version to Android 8.0 (API level 26). Android 7 and 7.1 are no longer be supported. In order to avoid a crash update the SupportedOSPlatformVersion in any .csproj files as well as in any AndroidManifest.xml files.

  - .csproj example:

    ```
    <SupportedOSPlatformVersion
    Condition="$([MSBuild]::GetTargetPlatformIdentifier('$(TargetFramework)')) ==
    'android'">26.0</SupportedOSPlatformVersion>
    ```

  - AndroidManifest.xml example:

    ```
    <uses-sdk android:minSdkVersion="26" android:targetSdkVersion="29" />`
    ```

- Review the following application styling items if problems arise:

- Change the xmlns schema "https://xamarin.com/schemas/2014/forms

```
https://xamarin.com/schemas/2014/forms
```

to

```
https://schemas.microsoft.com/dotnet/2021/maui
```

  in the App.xml file and other xaml files.
- Make sure that the proper package references shown below are included in the GWS.csproj file

```
<PackageReference Include="Microsoft.Maui.Controls" Version="$(MauiVersion)" />
<PackageReference Include="Microsoft.Maui.Controls.Compatibility"
Version="$(MauiVersion)" />
<PackageReference Include="Microsoft.Maui.Essentials" Version="$(MauiVersion)"
/>
```

- In App.xaml.cs, make sure that the constructor has changed to only have the parameter *IApplicationStateChangeServices*.

```
App(IApplicationStateChangeService s) : base(s)
```

Breaking Changes

- **VocabWordInfo** objects look up localization keys when their properties are first accessed. They should not be created with WorkFlow activity (WFA)-scoped localization keys. In the past, objects would return different values based on the current WFA. With this release, for a performance improvement, they save their localized value when first called. This is not expected to be a common situation, and should not impact most applications.

- Changed the *TargetFramework* of DevKit NuGets to generally support .NET 6.0 for Talkman and .NET 8.0 for Android and iOS. This forces the sample project .csproj files, in gws.sln, to update their *TargetFramework* properties to match. For instance, VoiceLinkModule\VoiceLinkModule.csproj sets its target frameworks like this:

```
<TargetFrameworks>net6.0;net8.0</TargetFrameworks>
```

  which produces two DLLs as output: one for use on Talkman with .NET 6.0, and the other for use on Android and iOS with .NET 8.0. When updating an existing app to DevKit 2.0, change your projects' .csproj files to match the TargetFrameworks in the sample project that is most like your app's equivalent project.

See the following documents for details on target frameworks, C# language versions, and multi-targeting:

- https://learn.microsoft.com/en-us/dotnet/standard/frameworks
- https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/configure-language-version for details
- https://learn.microsoft.com/en-us/dotnet/standard/library-guidance/cross-platform-targeting#multi-targeting

- Removed the *TinyMessenger* class which is unused by DevKit. If you were using it and need it back, then its source code remains available at https://github.com/grumpydev/TinyIoC/blob/master/src/TinyMessenger/TinyMessenger.cs.

- Updated the embedded *TinyIoC* class to the latest version. This change improves performance but has a breaking change. It now throws an exception in cases where a singleton registration is resolved along with a default named constructor parameter provided in a registration lambda. In older versions of *TinyIoC* the named parameter was ignored. A code change is necessary to prevent the exception and provide the same behavior as before. For example a registration like this would have worked in the past, but now fails:

```
TinyIoC.TinyIoCContainer.Current.Register<IFoo>(TinyIoCContainer c,
NamedParameterOverloads p) => c.Resolve<ISomeIntefaceExtendingFoo>(p));
```

If you have code like this in your app, fix the issue by removing the `p` variable in the `Resolve` call:

```
TinyIoC.TinyIoCContainer.Current.Register<IFoo>(TinyIoCContainer c,
NamedParameterOverloads p) => c.Resolve<ISomeIntefaceExtendingFoo>());
```

- After upgrading the DevKit 1.x to DevKit 2.x, ODR/persisted transaction data is no longer stored in a binary-formatted fashion. Rather, the data is stored in a JSON-formatted fashion, and ODR/persisted transaction data that remains behind, to be sent after migrating from DevKit 1.x to DevKit 2.x, for any transactions marked true for persistence (persist=true) in the APIs exposed by IRESTQueue or ITCPSocketQueue (or derived classes), are processed/deserialized based on their default implementation, without any expectation of data loss. If you have previously extended either of these two interfaces, it is highly recommended that you read the documentation associated with the *Honeywell.GuidedWork.Communication.Services.Communication.IMsNrbfDeserializationService* interface, in order to avoid any transaction data loss. Additionally, backward migration of ODR/persisted transaction data from DevKit 2.x to DevKit 1.x is not possible or supported.

- The *MainActivity* class for Android apps has been simplified, with many features having moved to its base class (*GuidedWorkMainActivity*). See devkit\GWS.Android\Platforms\Android\MainActivity.cs for an example of what needs to be in your class when porting an application to DevKit 2.0. You may add your own extensions, but the only current DevKit requirement is that you set the *RequestedOrientation* property as desired, and set attributes on the *MainActivity* class as desired.

- Removed support for Captuvo mobile phone scanner sleds on iOS.

- The *XplatDependencyOverrides* class for Android and iOS apps is responsible for registering some dependencies or dependency overrides that were previously registered in *MainActivity* and *SplashActivity* classes in DevKit 1.x. In particular, note that GWS.*App* is now registered in *XplatDependencyOverrides*.

- Removed support for Visual Studio extension (VSIX) files.

- Removed the Selection (*WarehousePicking*) and Retail (*OrderPicking*) workflows from DevKit. The *Receiving* workflow remains as an example for some custom workflow views.

- Removed the Honeywell.GuidedWork.RetailModule NuGet that supported these removed workflows, along with undocumented classes in other NuGets that support the removed workflows.

- Changed the type of IGuidedWorkStore.*WorkflowObjectList* from *IEnumerable<WorkflowObject>* to *List<WorkflowObject>*.

- Removed undocumented *Selected* property from the slot message returned by menu intents. The property was removed because of a race condition in older versions of DevKit. See the method *DecodeEnquireDamage* in LAppPickOrderLogic.cs for an example of checking for selected menu items.

- The *IModal.ShowToastDialog* and *IModal.ShowAlertDialogAsync* methods do not display dialogs if called when a DevKit application is not the foreground app.

- A behavior change in MAUI affects how views are managed when the app is running in the background or if the display is powered off. Pushing and Popping views from the stack hangs until the app returns to the foreground. DevKit 2.0 includes updates to address this issue on Android. However if your application includes custom graphics you should test its behavior while running in the background and with the display powered off.

- The *XamarinPageViewAdapter* class is now internal instead of public.

Enhancements

- Enhanced A700x logging by disabling Continuous Socket logs to reduce log bloat and adding a *TransportsLogEnabled* task setting to enable them.

- Changed captions added by the prompt captioning setting to apply to all spoken prompts as opposed to just initial spoken prompts.

- Log certificate information and error details if server certificate validation fails

- Enhanced A700x logging by logging Intents from the Python portion of a VAD rather than C#, reducing the number of logging calls in the process. When using Mock Catalyst, Intents are now logged only in <Your Project>\bin\Debug\net6.0\temp\mock_log.txt and not displayed in the debug console. Slots continue to be logged in the debug console.

- Changed the format of logged Intents. The log entry now is a single line unless it must be wrapped, rather than one line per property.

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
Region specific Locale Fix bug to support region specific locale setting for neutral language in PickUpAndGoSpecificLocales in LanguageAvailabilityServices.cs.	
Bluetooth Headset Lists As Scanner Fixed a bug that allowed Bluetooth headsets to be detected/listed as scanners under Settings > Scanner Settings > Current Secondary Scanner.	
Invalid Date Prompts Not Displayed Fixed issue with invalid date prompts that were not displaying.	
Delay for First Prompt Fixed an issue with TTS utterance that caused a delay in the very first spoken prompt.	
Speech Recognition Performance Fixed an issue where speech recognition performance was degraded when exiting standby mode by lowering a SRX headset microphone boom.	
Duplicate Automation IDs Fixed an issue with duplicate Automation IDs in VoiceLinkServerSettingsView.xaml.	
Automation IDs Reset Fixed an issue with UserEntrySubview where Automation IDs were reset when properties change in the UserEntrySubview class.	
Sign-On Loop Fixed a bug that caused an endless sign-on loop that occurred after lowering the SRX headset mic boom, and returning from the standby screen, which is displayed as a result of raising the SRX headset mic boom.	

Issue Description	Issue ID

decimal-places Property

Fixed an issue where the decimal_places property is not able to get a decimal part in a float intent and try to find the specific node by the key that tried to build.

Configuration Settings

Fixed issue where some configuration settings were not being notified of configuration change events.

Vocabulary Not Added

Fixed a bug where vocabulary was not added to a vad's config.xml when vocabulary only existed in parent culture. For example when there is a vocabulary list for "es-MX" that does not have one of the words in "es" it will now use the vocab from the "es" list.

Race Condition Processing Messages

Fixed a bug that caused a crash in an A700x application when a race condition occurred when processing messages sent from dotnet to Artisan components.

Speak Ahead Ignored

Fixed an Android/iOS issue where accepted recognitions are sometimes ignored, incorrectly logging "Speak ahead ignored" in cases when "Speak ahead" must not be ignored.

VOSMB-11015

Race Condition with Menu Intents

Fixed a race condition with processing menu intents.

VOSMB-10426

Serial Number Reported as 1

Fixed an issue where Talkman device serial number reported as 1 because the device properties were not getting loaded during app start.

VOSMB-10393

Log Line Length

Changed maximum C# log line length before wrapping from 950 to 462 to address the issue where logs would appear disordered from auto-wrapping.

VOSMB-10116

Issue Description	Issue ID
Language Selection Crash Fixed bug that makes neutral language selectable for PnG and leads to crash. This can happen after pushing both neutral and specific region external resources, like GWExternalResources.de.resx and GWExternalResources.de-DE.resx, to GWS application.	VOSMB-8676
Minimum Length Validation Message Setting the MinLengthValidationMessageEnabled intent property to true on GetValue caused an error message to be displayed for Android and spoken for A700x if the length of the input data is less than the minimum allowed. Fixed for A700x devices the error message was not spoken if the input data was less than the minimum allowed length when the MinLengthValidationMessageEnabled property was enabled.	VOSMB-8055
Menu Intents Fixed an issue with Menu intents. The application response to the "say again" global word was the wrong index prompt instead of the instruction prompt. MenuItemDialogue_InvalidIndex RESX key will not be repeated when "say again" global word is spoken.	VOSMB-7948
Control Help Incorrect Response Fixed issue where Control Help message was incorrect after speaking values.	VOSMB-1012

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 O (8.0)

The above are the minimum recommended device 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 Devices

- Honeywell CT30 XP
- Honeywell CT40
- Honeywell CT40 XP
- Honeywell CT45
- Honeywell CT47
- Honeywell CT60
- Honeywell CT60 XP
- Honeywell CN80
- Honeywell CN80G
- Honeywell CK65
- Honeywell CW45
- Honeywell ScanPal EDA51
- Honeywell ScanPal EDA71
- Honeywell A700x

Third-Party Devices

- Zebra WT6000
- Zebra TC5x
- Zebra TC7x

Headsets

- Honeywell SRX-SL Light Industrial Use Headset
- Honeywell SRX2 Wireless Headset (with Hands-Free Profile support)
- Honeywell SRX3 Wireless Headset

Scanners

- Honeywell CT30 XP On-board Scanner
- Honeywell CT40 On-board Scanner
- Honeywell CT40 XP On-board Scanner

- Honeywell CT45 On-board Scanner
- Honeywell CT60 On-board Scanner
- Honeywell CT60 XP On-board Scanner
- Honeywell CN80 On-board Scanner
- Honeywell CN80G On-board Scanner
- Honeywell CK65 On-board Scanner
- Honeywell CW45 On-board Scanner
- Honeywell EDA51 On-board Scanner
- Honeywell EDA71 On-board Scanner
- Honeywell A730x On-board Scanner
- Honeywell 8670 Ring Scanner

Android Device Operating System Support

- Android O (8.1.0)
  - Honeywell CT40
  - Honeywell CT60
  - Honeywell CK65
  - Honeywell ScanPal EDA51
  - Honeywell ScanPal EDA71
- Android P (9)
  - Honeywell CT40
  - Honeywell CT40 XP
  - Honeywell CT60
  - Honeywell CT60 XP
  - Honeywell CK65
- Android 10
  - Honeywell CT40
  - Honeywell CT40XP
  - Honeywell CT60
  - Honeywell CT60XP
  - Honeywell CK65

- Android 11
  - Honeywell CT30 XP
  - Honeywell CT40
  - Honeywell CT40XP
  - Honeywell CT45
  - Honeywell CT60
  - Honeywell CT60XP
  - Honeywell CK65
- Android 12
  - Honeywell CW45
  - Honeywell CT47

Honeywell A700x Software Support

- A700x devices must support .NET 6.0. This requires VoiceCatalyst 4.6 ECS001 minimum, however VoiceCatalyst 4.7.1 ECS005 or later is recommended.

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

Management Server Support

- VoiceConsole 5.5 or later for A700x
- VoiceConsole 6.1 or later for Android device support

Issues Reported in this Release

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**
HVMA Language Not Switching Properly In Honeywell Voice for Manhattan Active (HVMA), when a new operator is loaded using a different voice than the previous operator, the new operator may hear a mix of both languages. **Workaround**: Remove the battery and reboot the device.	**VOSMB-11078**
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**
Disable Vocab and Alphas English Only Disabling vocab words or Alpha by pushing an ApplicationSettings.config file is only working for English	**VOSMB-10896**
Application Scheme Labels When an ExternalResources RESX file is pushed to the device, the application scheme lists "Microservices" instead of "Enterprise Voice". **Workaround**: If "Enterprise Voice" is not present in the list, select "Microservices".	**VOSMB-10778**

Previously Reported Limitations

Issue Description	Issue ID
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**
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**
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**
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**
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**

Issue Description	Issue ID
"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**
Do Not Press Next Button Quickly If you have untrained words and get to the template training instructions screen, if you tap the Next button in rapid succession, it can cause the application to shut down unexpectedly.	**VOSMB-230**
Audio to the Bluetooth Headset Can Fail In some situations, the audio no longer comes through a Bluetooth headset even though it is still paired to the device. **Workaround**: Unpair and re-pair the headset.	**VOSMB-28**
Errors while Transmitting Files to VoiceConsole A number of errors may appear in the device logs when transmitting logs to VoiceConsole. In spite of these errors, all log files are eventually transmitted successfully.	**VOSMB-570**
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**