Contents

Design

You are ready to get started learning about everything AppLink™ related. In this section, you will learn about:

- Driver Distraction – What it is, why it is important, and what does the various governments from around the world say about it.
- Initial Considerations – What you should think about before you start designing your application's user experience.
- Human Machine Interface (HMI) – Overview of the interfaces you will have access to with your application.
- Global Requirements – Requirements your application must meet in order to be distributed.
- Regional Requirements – Additional requirements your application must meet based on geographical location.
- Best Practices – Learn about the best ways to provide quality in-vehicle experiences.

Driver Distraction

Driver distraction is a very serious concern when developing an application for use in-vehicle. Rules and laws are always in flux and may change in the future, so it is important to consider these ramifications at the beginning of the development cycle. Another important consideration for applications that will be available globally, Driver Distraction rules and laws will vary by region.

When the SYNC® system is in “driver distraction” mode, the user will not be able to access certain areas of the HMI. For example, when the vehicle is in motion, phone pairing and sub level menus are not accessible. Designing a menu and voice tree that is quick, intuitive, and easy to use is critical.

To help with relaying information about when the vehicle motions, the OnDriverDistraction RPC message has been added. This allows the application to receive notifications when the vehicle begins to move over 5 Km/h (about 3 mph) and when the vehicle comes to stop.

Initial Considerations

Before you start developing, here are some very important considerations and questions you should ask yourself.

Display vs. Voice

When designing your experience, it is highly recommended to focus on voice interactions first and foremost.

On a mobile device, navigating through an application is entirely a visual process.

In a vehicle, especially while driving, application usage should be almost entirely accomplished using voice commands. This will help to eliminate the need to understand the driver distraction laws and guidelines that are continually changing around the world.

Questions To Ask Yourself

What Are The Major Features, or Functions, of My Application?

These nouns/verbs could be loaded as top-level voice commands, or AddCommands.

It is recommended that this list be less than 150 commands, to improve application initialization performance and for high voice recognition quality.

Do I Use Shortcuts, Favorites, Presets, Etc. Within My Application?

Consider automatically mapping these to the preset buttons in vehicle.

What Does a Brand New User Need to Know?

A welcome message with basic instructions could be given the first few times a user uses the application on SYNC® using the Speak function. Once they've used the application a set number of times, you can remove these helpful prompts.

What Does the Generic 'Help' Voice Command Do?

By default, SYNC® lists the application's AddCommands during the help prompt. The application can make the prompt more informative by changing the help prompt using the SetGlobalProperties command at any time.

Additionally, contextual help is very helpful! Often, a user can be in a certain 'section' of the application that has unique commands. Changing the help prompt to highlight those commands greatly increases user familiarity and knowledge of the application through SYNC® AppLink™.

Is There a Long List of Items to Select From Within My Application?

You can load these choices as items in a ChoiceSet and then call that ChoiceSet during a PerformInteraction.

Applications can load ChoiceSet with 100 items. If your PerformInteraction requires additional commands, you may reference additional ChoiceSets.

If your list of choices is known ahead of time, it is helpful to create these during your initialization phases, and simply reuse the ChoiceSet throughout your application's lifecycle.

Note: It is not recommended to consistently delete and create choice sets. If you must delete a ChoiceSet, it is suggested that you wait some time since it was last used. Immediately deleting a ChoiceSet after its PerformInteraction has returned could lead to undesired application behavior.

Does My Application Require a Logged In User Account to Operate?

If your application does, you should always have logic to catch applications that are running on AppLink™ without an active account and display a message notifying the user to log in when not driving.

What Commands Should My Audio Streaming Application Use?

Audio streaming applications should consider adding the following commands:

- Play
- Pause or Stop
- Resume
- Skip
- Skip back, if your application allow

What Buttons Should My Audio Streaming Application Use?

Audio streaming applications should register for the following button press events:

- Play/Pause (OK) Button
- Seek Backward
- Seek Forward

Human Machine Interface (HMI)

What is the HMI?

HMI is a term to describe the interface that the user/driver uses to interact with the vehicle. This interface includes a collection of presets, media buttons (seek forward/backward, tune up/down, and play/pause), menu items, and most importantly voice commands.

It is also important to note that not every vehicle has the same HMI. For example, some HMIs include touch displays while some only have simple text based displays. Some have hard buttons while some only have soft buttons on their touch screen.

HMI Concepts

Every HMI will have the same basic concepts that you will need to familiarize yourself with before you start developing. The main concepts that we will cover include:

- User-Initiated Interactions
- App-Initiated Interactions
- Presenting Information

User-Initiated Interactions

Command Hierarchy

The main way to receive interaction from the driver is through your Command Hierarchy, or AddCommands. This allows you to create a 'tree' structure of commands that a user will be able to issue. They can be menu only, voice only, or both menu and voice.

Note: any child commands might be limited due to driver distraction while the vehicle is in motion.

Button Interactions

Another form of receiving interaction is for you to register for Button interactions. This allows you to receive button presses and act accordingly. For example, as an audio app, you should subscribe to the 'OK' button, as users will use it to toggle between play and pause.

App-Initiated Interactions

Another way to interact with the user is by starting off the interaction programmatically. This allows you to preset a list of available choices to the user, and prompt them to make a selection.

Presenting Information

There are only a few ways of presenting information to the user. The first way is writing to the available templates, such as your application's main screen. While more advanced displays allow for more updatable fields and templates, you are always given a base

template to write to. The second is to send a text string for the voice engine to process and speak back to the user.

The combination of the two, display template updates and sending text to be processed into speech, is the best way of presenting information to the user. An example of this is requesting a popup with text-to-speech (TTS), or Alert.

HMI Status

While the RPC Reference documentation will enumerate all RPCs, notifications, and their parameters, there is one callback that deserves special mention. The OnHMIStatus RPC notification conveys information regarding the state of the HMI.

OnHMIStatus informs your application of three important pieces of information. The three pieces are:

- HMILevel
- AudioStreamingState
- SystemContext

HMILevel

AppLink™ uses the concept of HMILevels to describe the current state of your application with regards to the level at which the system can communicate with it (and vice versa). HMILevel is represented by four levels:

- FULL
- LIMITED
- BACKGROUND
- NONE

FULL

The FULL level indicates that your application has full access to the system's interface. An application can be moved into focus by various means, most commonly being when the user selects the application from the Mobile Applications menu, or when the system turns its attention back on the mobile application (e.g. after a phone call).

LIMITED

The LIMITED level indicates your application only has limited access to the system's interface. For example, if an audio app is in limited, the display is showing some screen other than the app's base screen, but you still have access to the vehicle's audio bus.

BACKGROUND

The BACKGROUND level indicates your application no longer has focus on the system. In this level, the system sees the current application as a secondary process (e.g. due to another application opening or a user-initiated event). The user cannot directly interact with your application (e.g. Push-to-talk (PTT), menu commands, etc.), and your application cannot directly interact with the user (e.g. PerformInteraction, button presses, etc.).

Depending on your app's classification and permissions granted on the system, your application may be able to generate certain controlled interactions with the user, such an Alert (with or without SoftButtons) from the background. This interaction may also be configured to allow a user to switch to your application.

NONE

The NONE level informs your application has registered its interface, but cannot interact with the user. Also, if your application is currently performing an operation, while in FULL, LIMITED, or BACKGROUND, and it receives NONE, all operations will be automatically terminated (e.g. interactions in-progress, Speaks in-progress, etc.).

AudioStreamingState

The system will also inform your application whether you have the ability to stream audio over the vehicle's audio bus. AudioStreamingState is represented by three states:

- AUDIBLE
- ATTENUATED
- NOT_AUDIBLE

AUDIBLE

The AUDIBLE state informs your application that it is currently allowed to stream audio.

ATTENUATED

The ATTENUATED state informs your application that any streaming audio is currently attenuated by some system audio or speech.

NOT_AUDIBLE

The NOT_AUDIBLE state informs your application that it is not allowed to stream audio.

SystemContext

The system will also inform your application of what is currently in process on the system (i.e. if the user is browsing a MENU, voice session is happening, etc.). SystemContext is represented by five states:

- MAIN
- VRSESSION
- MENU
- HMI_OBSCURED
- ALERT

MAIN

The MAIN state informs that there are no interactions (user-initiated or app-initiated) in progress, and your application's base template is fully visible on the display.

VRSESSION

The VRSESSION state informs your application that the system is currently in a VR session and possibly overlaying a popup with choices over your base template.

MENU
The MENU state informs your application that the user is currently browsing your in-app menu.

HMI_OBSCURED
The HMI_OBSCURED state informs your application's base template is currently being obscured by some kind of system or another app's popup or alert.

ALERT
The ALERT state informs your application, while in the BACKGROUND level, that your requested alert is currently being displayed.

Head Unit Capabilities
As each Head Unit was created specifically for different vehicle lines, they have slightly different physical capabilities, as well as AppLink versioning capabilities, that you will need to consider during your Design phase.

AppLink 1.0 through 1.2

SYNC®, or Center Information Display (CID), Features

Screen
- Two lines of Dynamic Text (“MainField1” and “MainField2”)
    - Limited by 20 characters
- Media clock (e.g. “9:15”)

Center Stack Buttons
- 10 preset buttons
- OK (Play/Pause)
- For Media Based Applications:
    - 4-way controller
        - Up = Tune Up

        - Down = Tune Down
        - Left = Seek Backward
        - Right = Seek Forward
    - Tune Up/Down dial
        - Clockwise = Tune Down
        - Counter-Clockwise = Tune Up

Steering Wheel Buttons

- OK (Play/Pause)
- Push-To-Talk (PTT)
- For Media Based Applications:
    - Seek Forward
    - Seek Backward

SYNC® with MyFord®, or Multi-Function Display (MFD), Features

Screen

- Two lines of Dynamic Text (“MainField1”, “MainField2”)
  - Limited by dynamically by the physical size of the characters
- Media clock (e.g. “9:15”)

Center Stack Buttons

- 10 preset buttons
- OK (Play/Pause)
- For Media Based Applications:
  - Seek Forward
  - Seek Backward

Steering Wheel Buttons

- OK (Play/Pause)
- Push-To-Talk (PTT)
- For Media Based Applications:
  - Seek Forward
  - Seek Backward

SYNC® with Voice Navigation, or Next Generation Navigation (NGN), Features

Screen

- For Media Based Applications:
  - Three lines of Dynamic Text (“MainField1”, “MainField2”, and “MediaTrack”)
    - Limited by dynamically by the physical size of the characters
  - Media clock (e.g. “9:15”)
  - Four On-Screen Soft Buttons
    - First four AddCommands will be displayed
- For Non-Media Based Applications:
  - One line of Dynamic Text (“MainField1”)

        - Limited by dynamically by the physical size of the characters

Center Stack Buttons

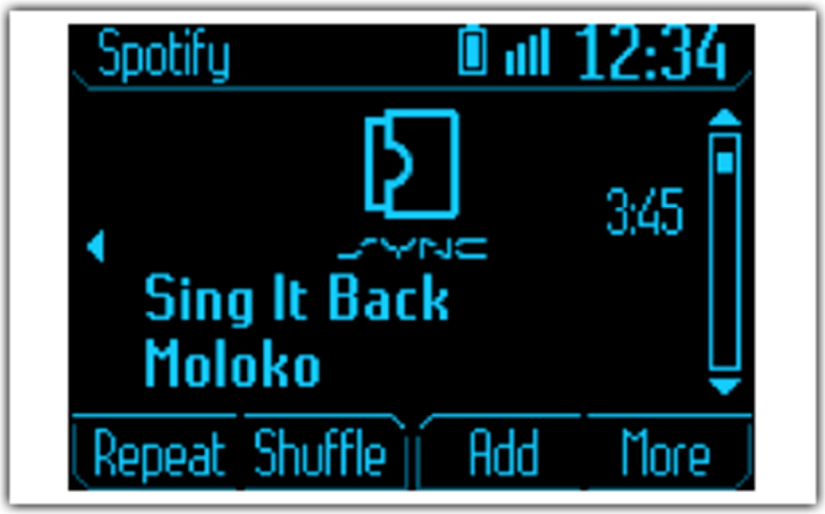
- 6 preset buttons
- OK (Play/Pause)
- For Media Based Applications:
    - Seek Backward
    - Seek Forward
    - Tune Dial
        - Clockwise = Tune Up
        - Counter-Clockwise = Tune Down

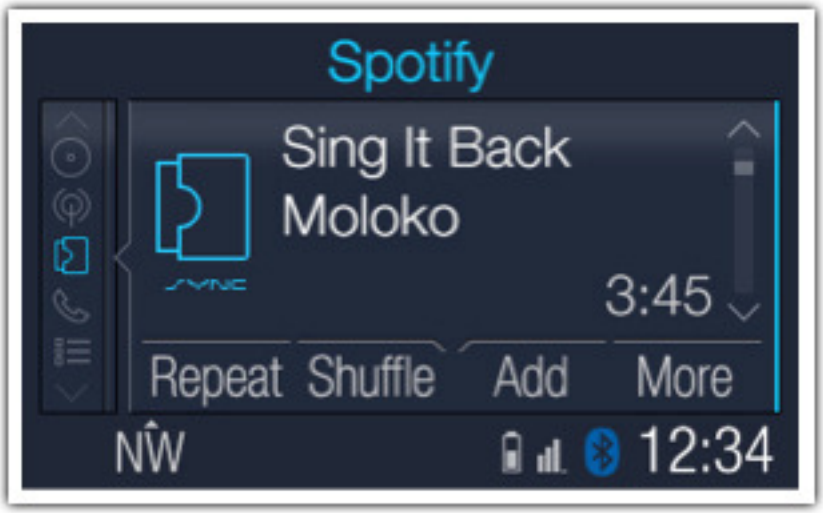
Steering Wheel Buttons

- OK (Play/Pause)
- Push-To-Talk (PTT)
- For Media Based Applications:
    - Seek Forward
    - Seek Backward

AppLink 2.0

SYNC® with MyFord®, or Multi-Function Display (MFD), Features





Screen

- Four lines of Dynamic Text (“MainField1”, “MainField2”, “MainField3”, and “MainField4”)
    - Limited by dynamically by the physical size of the characters
    - Pressing the Down arrow will toggle between Lines 1/2 and 3/4
- Media clock (e.g. “9:15”)
- Four SoftButtons

Center Stack Buttons

- 10 preset buttons
- OK (Play/Pause)
- For Media Based Applications:
    - Seek Forward
    - Seek Backward

Steering Wheel Buttons

- OK (Play/Pause)

- Push-To-Talk (PTT)
- For Media Based Applications:
    - Seek Forward
    - Seek Backward

Supported Icons

- On-board icon set available for use
    - See [Supported Icons]

Global Requirements

General

1. Your application must always be available in the mobile apps menu when the phone is connected to the vehicle.
2. When your application is running through AppLink™ and an incoming call is received, the media audio should be paused while your app is in BACKGROUND / NOT_AUDIBLE.
3. When your application is running through AppLink™ and an outgoing call is placed, the media audio should be paused while your app is in BACKGROUND / NOT_AUDIBLE.
4. Your application should pause or stop audio while in the BACKGROUND or NONE state.
5. Your application should minimally mute, preferably pause or stop, the media audio when transitioned to the NON_AUDIBLE (FULL or LIMITED) state.
6. Your application shall send a SetGlobalProperties to establish an advanced help prompt before sending any voice commands.
7. Your application must implement a lockscreen. See <HTML link to Lockscreen> for the full requirement.
8. When your application is in NONE state, has been disconnected from SYNC or receives an OnDriverDistraction notification DD_OFF, the application can allow regular interaction with the mobile device.
9. If using the term SYNC and/or AppLink in your application GUI, you must receive written permission from Ford to use any Ford trademarks and logos, and include a registered symbol and trademark in the following fashions if approved:
    - Ford SYNC®
    - SYNC® AppLink™
    - SYNC®
    - AppLink™
10. Applications must not register voice grammars using synonyms that include other application names, or conflict with on-board voice commands.
11. Applications must provide an audio or visual response to user interaction (button presses, VR, AddCommands, etc.).
12. Applications must not provide an incorrect or unexpected response to user interaction.

Regional Requirements

North America
No regional requirements at this time.

South America
No regional requirements at this time.

Europe
No regional requirements at this time.

Asia Pacific
No regional requirements at this time.

Best Practices

Button Management

There are a variety of options for an application to utilize when it is active on SYNC® AppLink™. After subscribing to the desired buttons, the application will be notified of button events (OnButtonEvent) and presses (OnButtonPress).
Each button press will have one of two modes: LONG or SHORT. This could be used for switching presets, saving a currently playing station or favorite to a preset, or even thumbing up or down a song.

Each button event will have one of two modes: BUTTONUP or BUTTONDOWN. For example, this could be used in conjunction with button press mode of LONG for fast forwarding through an audio stream.

For preset buttons that exist on a head unit, pre-populate those buttons with set features, for example, a user's favorite stations. If you cannot pre-populate those buttons with features, it's good to provide text-only feedback via an Alert that suggests the button is unused or a preset is not set.

On certain Head Units, buttons may not exist such as presets 7-0. You may want to check for the buttons available in the response of your RegisterAppInterface.

Certain head units also provide access to “soft buttons” on the display. These are top level buttons that can be used to quickly access certain functions. Any quick functions that require single button press functionality can be added here, like bookmarking or rating something, providing a quick update or media controls play and seek forward/backward. You may want to check for the buttons available in the response of your RegisterAppInterface.

Using the Display

There are anywhere between two and four lines of text available to your application via the Show command. As a general rule, this display is used to convey the current state of your application, either what station is playing, artist/track being streamed, distance to location, or other relatively continuous updates on Lines 1 and 2. Other information which might be useful, but not pertinent to functionality are best included on Lines 3 and 4, if available.

Choosing VR Commands

SYNC® global system VR commands are not to be used. Examples include, but not limited to:

help	voice settings help	DAB 1
voice settings help	voice settings	DAB 2

main menu help	audio	DAB 3
audio help	audio system	line in
audio system help	music	AV in
music help	CD	audio video
main menu	CD player	AUX
exit	disc	USB
cancel	disc player	iPod
interaction mode advanced	voice settings	MP3 player
interaction mode novice	radio	MP3
confirmation prompts on	AM	Bluetooth audio
confirmation prompts off	FM	phone
media candidate lists on	FM 1	iPhone
media candidate lists off	FM 2	Blackberry
phone candidate lists on	AM 1	<Device Name (USB, BT, or Phone)>
phone candidate lists off	DAB	

Note: This also applies to each translated word or phrase in all SYNC® supported languages.

Effectively Responding to a User with Voice or Display Updates

It is helpful to repeat voice command input as confirmation to commands that are triggered via voice. For example, a voice command to "Get Local Traffic" would be followed up with a Speak saying "Getting Local Traffic." An exception to this practice would be while streaming audio where you may not want to interrupt audio playback with a Speak, so an Alert with no text-to-speech could update the display-only instead, for example to indicate that you acknowledged the users input.

When designing your voice tree, be aware that you can trigger up to 3 voice prompts in a row using PerformInteraction. This is due to both driver distraction rules as well as ease-of-use. The following is an example:

- SYNC®: “Please select a country.”
- User: “USA”
- SYNC®: “USA, please select a state.”
- User: “Michigan”
- SYNC®: “Michigan, please select a city.”
- User: “Detroit.”

It is good practice to always confirm a voice command as seen in the above example where SYNC is confirming the previous response.

Note: You do NOT need to confirm the command by asking the user a yes or no question. SYNC®'s voice engine does this for you if an utterance is unclear.

Registering your Application

Once you send the registration message (if successful), you will receive a response with very important information about the system that you need to consider and add logic around. Within this response, you will learn important details about the system including items like the AppLink version, HMI language (Display and Voice), DisplayType, button capabilities, etc.

For example, if you detect the system is in French and you support French, you will want to use the ChangeRegistration API to set yourself to French and then load only French commands and display strings. If you don't, the user will hear a language mismatch prompt informing them that your application might not work as they expect.

Initializing your Application

Within your initialization of code, your app is going to want to register voice recognition commands, subscribe to buttons, and set up the help prompt, among other things. When initializing, consider performing the following actions in order to speed up processing time and ease-of-use for the user:

1. SetGlobalProperties to establish your application's help and timeout prompts
2. Send a Show command to update the display with a welcome or initialization message such as "Buffering…"
3. Subscribe to buttons required for your application
4. Register voice and menu commands using the AddCommand function.
5. Perform any other initialization, including creation of choice sets.

Note: If your application streams audio, it is best to play audio immediately, either a user's favorite station, last played content, or some other default. This may start immediately, and the display should be updated accordingly. It is best to perform this after button initialization.

Command and ChoiceSet Management

While DeleteCommand and DeleteInteractionChoiceSet are supported RPCs, only use them when appropriate. Avoid deleting commands and ChoiceSets that will knowingly be used again.

Vehicle Data

There are a number of vehicle data items, which are made available via the vehicle data APIs GetVehicleData and SubscribeVehicleData.

GetVehicleData will provide a point-in-time value for the vehicle data items being requested.

SubscribeVehicleData will provide an update to vehicle data every second, if and only if it has changed. For an element like vehicle speed, you may get an update each second, unless the speed is constant (as in cruise control) in which case the notification will not be send again. For items like PRNDL status, you may only receive this when it changes.

Some of the elements made available are:

- Speed
- Engine Speed (RPM)
- Fuel Level
- Fuel Economy
- Vehicle Location
- Odometer
- Fuel Level State
- Fuel Consumption
- External Temperature
- PRNDL State
- Tire Pressure

For a complete list, please look at the SYNCProxy documentation.

When designing an application to use vehicle data, it is important to note that each vehicle may have differing sets of data and not all data may be available for each vehicle line. Please check the vehicle data availability documentation per region for further information. If data is not available, it is recommended to inform the user that their vehicle does not have the appropriate availability of the data required for your application.

Alert vs PerformInteraction

Alerts are short informative messages that can simply be dismissed after a short period of time. They are similar to any push notification that you receive today on your smartphone.

The Alert API also has the ability to collect a quick response from the user through the use of SoftButtons. If your application requires the user to choose from muttiple options (more than 2), you should use the PerformInteraction API.

PerformInteractions are dedicated one-time question and answer sessions that require a user's response from a set of a few to many choices.

Using ScrollableMessage

ScrollableMessage is a useful way to show and display text to a user. In some markets like Europe and Asia, ScrollableMessage can be used at any time, to show a message or other relevant information that wouldn't fit on an Alert. It is important to note that in North America, ScrollableMessage is limited in use and cannot be used while the vehicle is in motion. Please be advised that you may find these messages are blocked from being shown in that region. To detect this condition, use the OnDriverDistraction notification on vehicles before showing the message, or if you detect ScrollableMessage

is rejected due to driver distraction restrictions, simply use a Speak or another method to convey the information.

System's Language Change Event

In the event a system-wide language change occurs and your application has added voice commands, you must first unregister yourself. Then, you must re-register and re-load your voice commands. If you support the new language that the system is set to, you must also change the language parameters in the RegisterAppInterface request.

Android™

While Android™ devices connect over Bluetooth®, this adds both additional functionality and complexity as described below.

Auto Start

To have your application show up in the Mobile Applications menu (requirement 5.1.1), you will have to implement specific functionality that monitors connections to Bluetooth® devices. Typically, this is easily done with intent listeners and a background service that acts accordingly when certain events occur (e.g. reboots or power cycles to the Bluetooth® device, Bluetooth® toggles, etc.).

Below is a list of some basic use cases that you can use as starting point to see if you manage the SyncProxy correctly:

- Device power cycles
- First application run
- Bluetooth® toggling

Other Android™ Complexities

When a Bluetooth disconnect occurs the proxy will always notify the app via an onProxyClosed notification. Unfortunately the timing for this notification is not consistent across all devices. This problem can be avoided if the app listens for an ACL_DISCONNECT and treats it as an onProxyClosed (dispose and recreate the proxy by calling the reset function, take off the lockscreen, etc.). The Hello AppLink and AppLink Tester applications in the SDK are examples on how this can be accomplished.

iOS

With iOS users, they expect apps to remain running, possibly continuing to streaming audio, when the application is backgrounded, and this fact is especially true when it is operating through SYNC® AppLink™. If you don't implement the following background mode, the application will be disconnected from SYNC® whenever the application is backgrounded, specifically seen with phone calls.

In the Xcode project .plist file, simply add a "UIBackgroundModes" item, which is converted to "Required background modes". Then, add at least this item, "external-

accessory." Other background modes (such as audio, navigation, etc.) can be added if needed.

Also, make sure the application declares "com.ford.sync.prot0" as a supported external accessory protocol. This is required to communicate with SYNC over External Accessory.

Sample .plist file

The application can only establish a connection with SYNC® AppLink™ while it is in the foreground. We also recommend that the phone is locked for SYNC® AppLink™ to work properly for iOS versions prior to iOS5.

When two or more SYNC® AppLink™ enabled applications are present on an iOS device, the application that is in the foreground will maintain that connection. If a second application then connects to SYNC® AppLink™, the first application's connection to SYNC® AppLink™ will be terminated and will have to reconnect when it is foregrounded.