



DEVELOPER PROGRAM

SDL BEST PRACTICES
SDL HACKATHON
MWC AMERICAS 2017

AGENDA



- Start with a Clean Slate
 - Planning and Considerations
 - App Naming, App Icons, Languages, Using templates
 - Reacting to commands
 - Graphics
 - Using Softbuttons
 - Detecting display types
 - Using voice commands with Sync
 - Alert vs Perform Interaction
 - Handling Permission Changes
 - HMI State Changes
 - Listening for RPC responses
 - Avoiding common mistakes
-

START FROM SCRATCH

- When designing your user experience for the car:

DO's

- Start from scratch, even if you already have a production app.
- Consider inherent differences between using an app on a phone in your hand, and using an app while driving a vehicle.
- Take advantage of HMI user input and VR capabilities provided by AppLink like Softbuttons and Voice Recognition commands.

DON'T's

- Try to port the existing handset user experience and interface from an app in to a car.
- Try to match up functionality one for one, the in-vehicle interface is typically just a subset of functionality provided by an app.



PLANNING AND CONSIDERATIONS

- Plan:
 - Support for multiple AppLink / SDL based head units
 - Different capabilities
 - Display Types
 - Softbutton Availability
 - Vehicle Data Availability
 - All information is provided during the Register App Interface handshake process
- Performance in-vehicle - how to best initialize your app
- Options for loading voice commands
- Providing feedback regarding current app status
- Display vs. voice
- Ideally, near instant acknowledgment to any user input



PLANNING AND CONSIDERATIONS

- Navigating AppLink in the vehicle
- Language Considerations
 - Changing app language to match SYNC
 - Regional languages
- Inventory list of permissions need by app
- Submit app id request early with provided list
- Always check responses received from RPC requests
- Lock Screen Management and importance
- Always listen for notifications
- Always check the Register App Interface Response

APP NAME & APP ICON

DO's

- Select a short memorable App Name (Around 10 Characters)
- Use VRSynonyms and TTSSynonyms in Register App Interface Request to tweak the pronunciation
- Set an App Icon after the first HMI NONE state is received

DON'T's

- Try not to be too generic. (e.g. Name your App "Radio Player" or "Messages")
- Avoid failing to set your app icon because of app lifecycle events



Graphics

DO's

- Check the Register App Interface response to see if the head unit supports graphics
- Always upload images using the PutFile RPC before attempting to use
- Verify the PutFile response for Success prior to utilization
- Verify using the ListFile for already uploaded images.

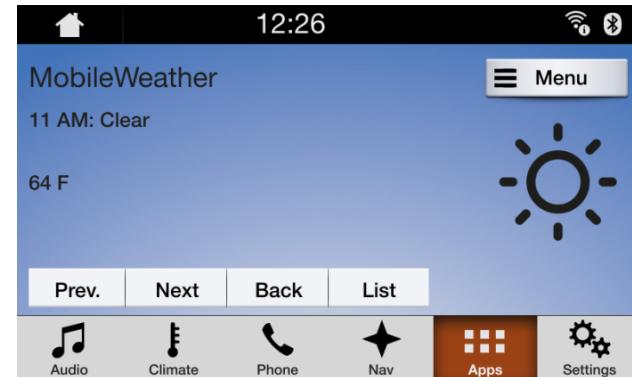
DON'T's

- Do not send images that have not been properly sized. For example, album art should not be larger than 185x185px.

INTERACTION WITH THE USER



Static Buttons



Soft Buttons

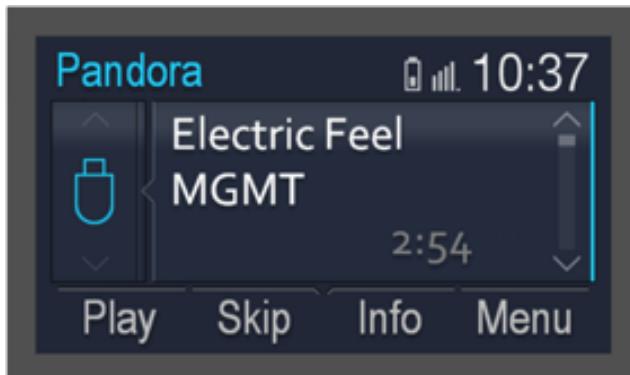


Voice Commands

Using Softbuttons

DO's

- Use short text only on Softbuttons
- Use icons that can be seen against a dark and light background



BUTTONS

- `onButtonPress()` Notification
 - Sent to app from SYNC whenever any button is pressed (steering wheel, center stack, touch screen, softbutton)
- LONG vs SHORT Press // BUTTONDOWN vs. BUTTONUP Modes
 - Example: SEEK button
 - Skip, Thumbs up, Fast forward

BUTTONS

- Presets
 - Up to 10 (0-9) buttons
 - Often used to mimic radio presets with hold & save
- Static button availability
 - Check the response to SubscribeButton RPC
 - SYNC will return NOT_SUPPORTED

USING VOICE WITH SYNC

- How to use it:
 - Register commands you want SYNC to recognize by loading grammars onto the system via a string of text or phonemes (phonetic sounds, e.g., read vs reed)
- How it works:
 - SYNC performs the recognition on behalf of the app
 - If recognition is ambiguous, it will provide an N-best list for the user to choose from.
 - If no match is found, it will ask the user to repeat the command

USING VOICE WITH SYNC

- Upon successful recognition:
 - Positive chime is played
 - `onCommand()` notification is sent to app with a commandID

APP COMMANDS AND SYNC SYSTEM COMMANDS

- SYNC itself includes embedded system voice commands that cannot be re-used by an app, such as ‘help,’ ‘exit,’ ‘cancel’
- SYNC executes these commands on behalf of an app

APP COMMANDS AND SYNC SYSTEM COMMANDS

- For example: “<AppName> Help”:
 - Option 1: setGlobalProperties() - The app populates the help prompt
 - App can set this at any time for contextual help
 - Option 2: Let SYNC populate this for you with the first few voice commands you load onto the system.
 - Not ideal: the list of commands in the help prompt may vary.

REGISTERING COMMANDS

- AddCommand
 - Top-level voice commands
 - Available throughout the app
- ChoiceSet
 - Presented to the user with PerformInteraction()
 - Good for large contextual lists

Reacting to commands

DO's

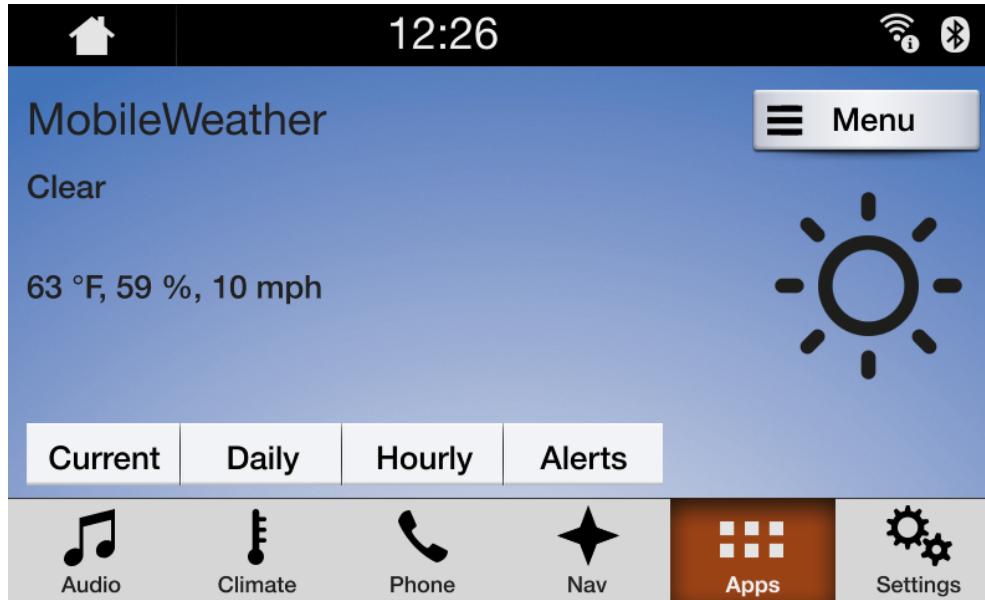
- If the user selects a command via Voice Recognition (VR) (check the onCommand notification, triggersource == "VR"), make sure to follow up via VR (for example by using performInteraction with the argument interactionMode = "BOTH", which will trigger a VR interaction.)
- On the other hand, if the user selects an option via menu(triggersource == "MENU") follow up with a manual performInteraction (interactionMode = "MANUAL_ONLY")
- If you need to load external items, inform the user (e.g. send a show command with the text "Loading..."

DON'T's

- Don't react to a manually selected command with a VR

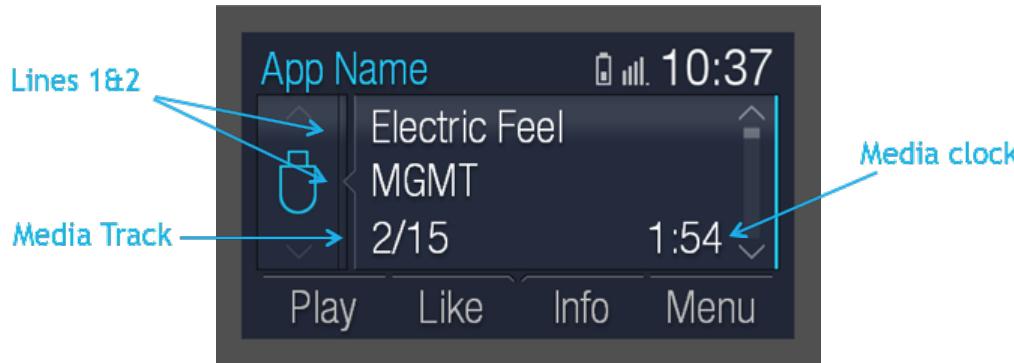
IN-VEHICLE DISPLAY

- SYNC 3 displays offer various layouts with text, image, and softbuttons.



IN-VEHICLE DISPLAY

- SYNC 1.1 displays offer 2-4 lines of text, a media clock timer, and a media track field



UTILIZING THE IN-VEHICLE DISPLAY

- Update meta-data
- Provide simple information to the user
- Update status information of the app, e.g., “Buffering”
- No scrolling/flashing text
- Provide relevant graphics

UTILIZING THE IN-VEHICLE DISPLAY

- Determine properties of current display
 - Text Fields
 - Graphics Support
 - Image Fields
 - Templates Available
 - Media Clock Formats
 - Number of Custom Presets
- Some units may not return all properties

LANGUAGES

- What is SYNC's language?
 - Changing app language to match SYNC (18 languages)
 - Regional languages
- The "RegisterAppInterface" RPC response will report SYNC's language to the app
- Changing app language to match SYNC
- ChangeRegistration()
- Ensure SYNC pronounces AppName correctly

LANGUAGES

DO's

- Switch to the language that SYNC is utilizing (check the Register App Interface Response)
- Listen for onLanguage change notifications from the system in case Sync's language is changed on the fly

DON'T's

- Avoid using static strings, be prepared to support multiple languages.

PERFORMANCE IN-VEHICLE

- If your app loads many commands:
 - Send a small number of the most important requests immediately, including:
 - Button subscriptions
 - Custom prompts (`SetGlobalProperties()`)
 - 4 or 5 of the most important commands
- Using a timer or background thread, send the remaining `AddCommand()` requests in batches

ALERT VS. PERFORMANCE INTERACTION

- Alerts(): short, informative messages that can be:
- Similar to push notifications on your phone
- Dismissed by: User closing, Timeout, User Input via Softbuttons

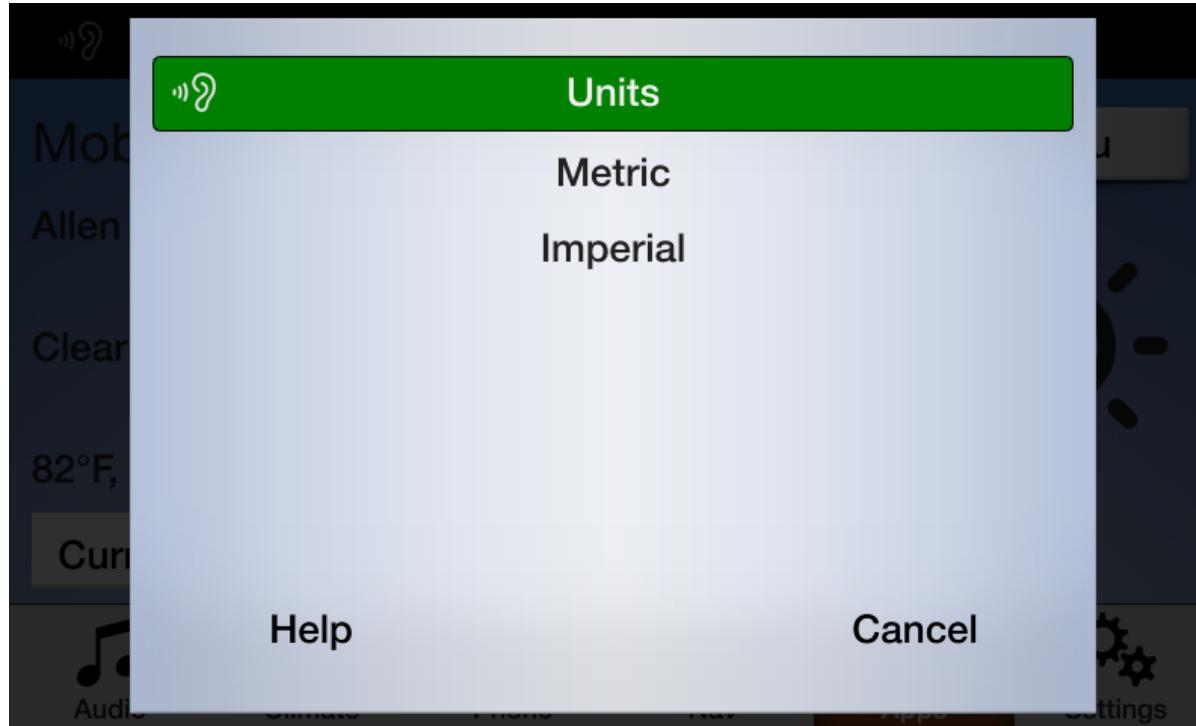
ALERT VS. PERFORMINTERACTION



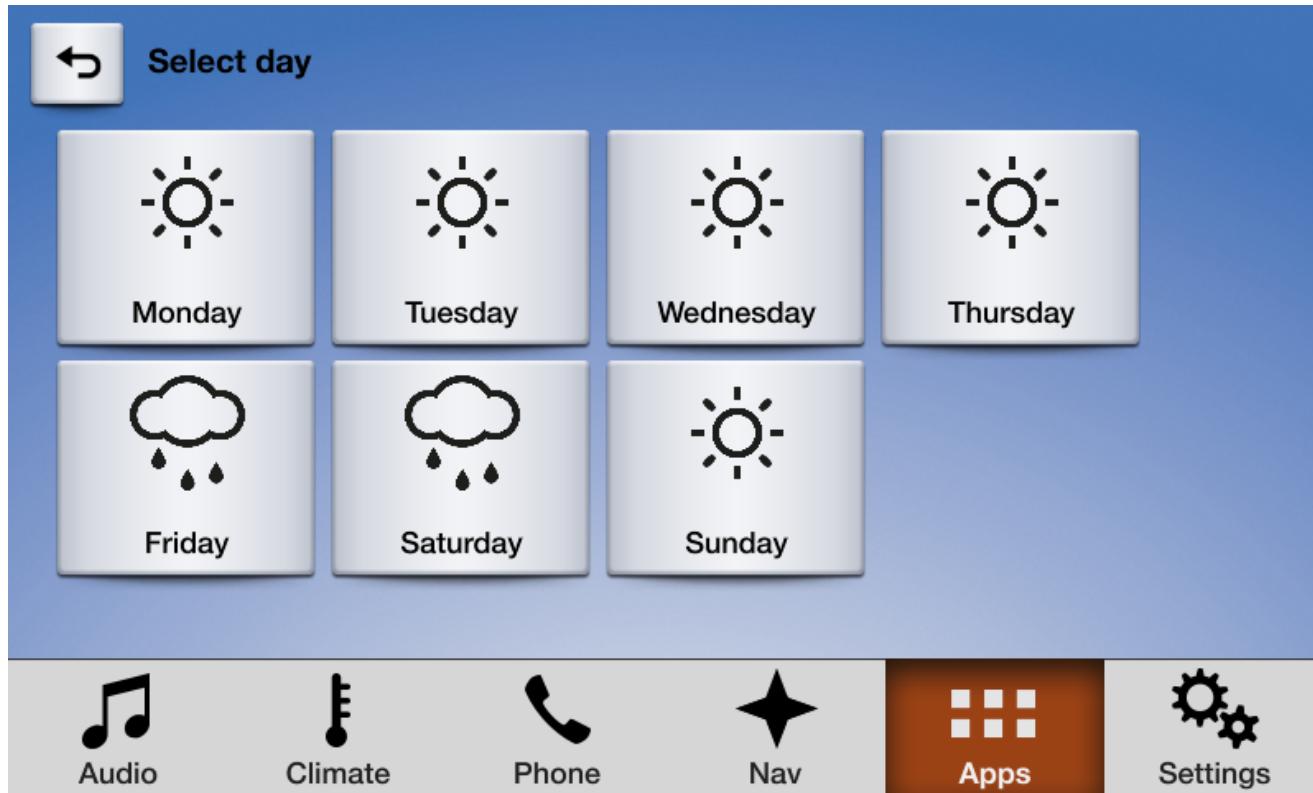
ALERT VS. PERFORMINTERACTION

- `PerformInteraction()`: dedicated one-time question and answer sessions that require a user's response from a set of choices.
 - Example:
 - App initial prompt: “Please select a state.”
 - User: “Michigan” (1 of 50 items in the ChoiceSet of states)
- If your application requires the user to choose from multiple options (more than 2), you should use the `PerformInteraction()` API.

ALERT VS. PERFORMINTERACTION



ALERT VS. PERFORMANCE INTERACTION



PROVIDING FEEDBACK TO THE USER

- Instant acknowledgment to any user input
- Providing updates during buffering or other delays
- How to provide feedback? e.g., Display vs. voice
- 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. E.g.,
 - SYNC®: “Please select a country.”
 - User: “USA”
 - SYNC®: “USA, please select a state.”
 - User: “Michigan”
 - SYNC®: “Michigan, please select a city.”
 - User: “Detroit.”

USING TEMPLATES

DO's

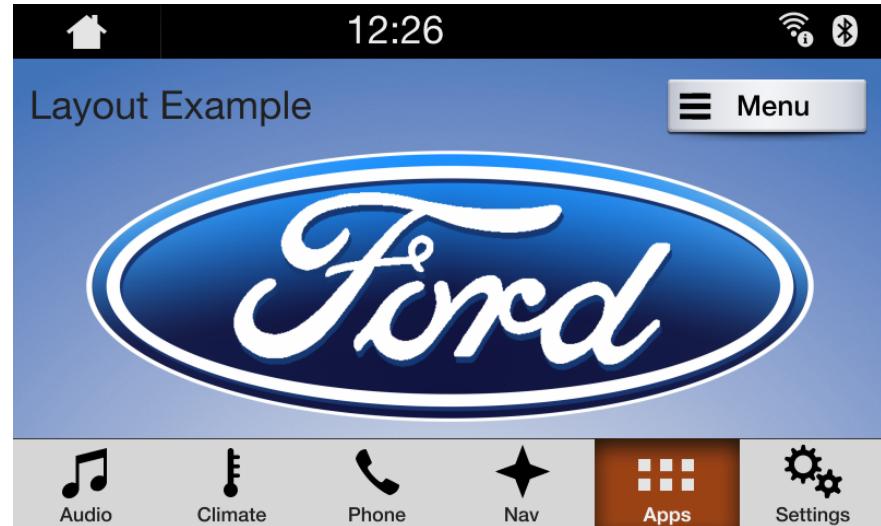
- Select the template that is most appropriate for your app.
- Switch templates if beneficial for your use case.

DON'T's

- Do not use the media template if your app is not a media based app
- Avoid utilizing text based images



DISPLAY LAYOUTS



FINAL CONSIDERATIONS

- Keep it Simple
- Consumption vs. Discovery
- Focus on using voice instead of the display.
 - Driver distraction rules prevent traversing through submenus

FORD DEVELOPER PROGRAM

- More information on best practices is available at developer.ford.com