

Introducing Multi-Capability Skills: Expand Your Device Experience by Combining Smart Home and Custom Skill Models into a Single Skill

Ahmed El Araby

May 28, 2020

Share:   

Smart Home Skills

Connected Devices

Smart Home

News



Today, we are introducing the general availability of [Multi-Capability Skills \(MCS\)](#), a new way to combine the [Smart Home Skill](#) and [Custom Skill](#) models into a single skill. Now, smart home device developers can publish and maintain a single skill that enables both the smart home and custom features for their devices, simplifying their skill maintenance and creating a more seamless customer experience. With MCS, you can extend the built-in Smart Home skill utterances with your own custom voice interaction model to support nearly any feature, without customers needing to enable and invoke two separate skills. For example, you can leverage the Smart Home model’s built-in support for utterances that turn a device on or off, and have your skill support custom utterances like “Alexa, ask Roomba to vacuum the kitchen”, or “Alexa, ask *myskill* what can this device do?”, all in the same skill. And with MCS, if you already have a live skill, you can simply update the additional smart home or custom skill model without needing to rebuild from scratch.

A Single, Simplified Development Solution

Multi-capability skills simplify developers’ back-end management. With MCS, you can now publish and maintain a single skill that enables both the smart home skill and custom skill features, allowing your device to more seamlessly support nearly any feature without needing to configure and maintain two separate custom and smart home skills. Until today, smart home device developers needed to choose to either create a [Smart Home Skill](#) to leverage a defined set of built-in smart home capabilities, create a [Custom Skill](#) to support additional capabilities not included in the Smart Home API, or create both skills. However, choosing a single skill model may not accommodate all features for all devices, and while a developer could build both a custom and smart home skill, it added the complexity of managing two skill experiences and requiring customers to enable multiple skills. For example, smart router owners can use

 German (Deutsch)

© 2010-2021, Amazon.com, Inc. und Tochtergesellschaften. Alle Rechte vorbehalten.

[Nutzungsbedingungen](#)

[Dokumentation](#)

[Foren](#)

[Blog](#)

[Alexa Developer Home](#)

custom skill so users could say “Alexa, test my internet speed”. Instead, with MCS developers can now support these features in a single skill.

Create Higher Quality Device Experiences for Your Customers

Simplifying your development process doesn’t mean compromising customer experience. With MCS, customers no longer need to search for or enable multiple skills to access all the features of their Alexa-connected device. MCS removes the friction of customers needing to remember different skill names, allowing customers to access all the expanded smart home features with a single invocation name. For example, by building a multi-capability skill, Dyson enabled its customers to interact more naturally with their Alexa-connected devices. Customers can control their Dyson fans with commands like “Alexa, set the fan speed to 5,” or “Alexa, set Oscillation to wide,” and set night modes and quiet modes in their daily routines, all features previously not available in a single skill experience.

Likewise, with MCS Alexa customers in the US with compatible TP-Link routers can access expanded device features in a single skill. Using pre-built Smart Home API commands, customers can control their internet access across some or all of their household’s connected devices by saying commands like “Alexa, pause internet for Timmy’s iPad.” In addition, TP-Link offers unique voice controls using its custom model, like “Alexa, ask TP-Link to enable gaming mode.” MCS allowed the combination of these device control features in a single skill without customers needing to enable separate skills.

How to Get Started Building Multi-Capability Skills

You can create a multi-capability skill by adding an additional model to a base skill using the Alexa Developer Console, through [SMAPI](#) directly - the [Alexa Skills Kit Command-Line Interface \(ASK CLI\)](#) and updating your skill logic to handle requests from those models. You can also update existing skills through the [Alexa Developer Console](#) as part of the skill configuration workflow and you can create MCS on your live skill development version, letting you test expanded feature support before submitting for certification. To revert to the original skill state, you can also delete the new model as long as you haven’t submitted the skill.

To create a new MCS with both smart home and custom skill capabilities:

1. Follow the instructions in [Create a skill](#) to create your new skill:

Subscribe

* **Business Email Address:**

* **Country:**

* **Last Name:**

Create a new skill

Cancel

Create skill

Model: Custom

Host: Provision your own

Skill name

My first MCS

12/50 characters

Default language

English (US)

More languages can be added to your skill after creation

1. Choose a model to add to your skill

There are many ways to start building a skill. You can design your own custom model or start with a pre-built model. Pre-built models are interaction models that contain a package of intents and utterances that you can add to your skill.

Custom

Design a unique experience for your users. A custom model enables you to create all of your skill's interactions.

Flash Briefing

Give users control of their news feed. This pre-built model lets users control what updates they listen to.

"Alexa, what's in the news?"

Smart Home

Give users control of their smart home devices. This pre-built model lets users turn off the lights and other devices without getting up.

"Alexa, turn on the kitchen lights"

Music

Give users complete control of their music. This pre-built model lets users search, pause, skip, or shuffle in your skill.

"Alexa, play music by Lady Gaga"

Video

Let users find and consume video content. This pre-built model supports content searches and content suggestions.

"Alexa, play Interstellar"

2. Click “MODELS”:

alex developer console

< Your Skills Sample custom Build Code Test Distribution Certification Analytics

English (US)

CUSTOM

Interaction Model

Utterance Conflicts (0)

Invocation

Intents (5)

Built-in Intents (5)

AMAZON.FallbackIntent

AMAZON.CancelIntent

AMAZON.HelpIntent

AMAZON.StopIntent

AMAZON.NavigateHomeIntent

Slot Types (0)

JSON Editor

Interfaces

Endpoint

Intent History

Annotation Sets

Display

MODELS

ACCOUNT LINKING

PERMISSIONS

How to get started

Alexa Skills Kit Developer Tutorial for Programmers: Building with the Alexa De... Watch later Share

amazon alexa Developer Console: Build

Resources

Update your live skill instantly

Qualify for live updates to your skill if changes are made to slot values and/or sample utterances. Learn more about live updates to your skill.

Catalog Management

Use Catalog management for managing slot types with large, constantly changing slot values. Catalog management is currently available only on Alexa Skill Management API (SMAP) and the Alexa Skills Kit Command Line Interface (ASK CLI).

Make Money

Create an In-Skill Product and add it to your skill. This enables you to sell premium content to your customers via voice.

Feature Updates & Releases

New Alexa Skills Kit Features and Tools.

Getting Started: A Comprehensive Course (Cake Walk)

Cake Walk: Build an Engaging Alexa Skill. This course is for software developers, voice developers, solution architects, user interface (UI) developers, voice designers, or anyone with beginner-level coding experience.

Documentation

Refer to our technical documents for detailed guides on building custom skills.

Skill builder checklist

Complete these steps to be able to test your skill using the simulator in

1. Invocation Name > Enter an invocation name for your skill

2. Intents, Samples, and Slots > Add at least one intent and one sample utterance

3. Build Model > Successfully build your interaction model

4. Endpoint > Set a web service endpoint to handle skill requests

In-Skill Products > Create an in-skill product and add it to your skill.

3. In this example we will add a smart home model to the skill. Note, that in order to add a smart home model to an existing custom skill your new skill must meet the prerequisites of smart home skill development:

alex developer console

< Your Skills Sample Custom Build Code Test Distribution Certification Analytics

English (US)

CUSTOM

MODELS

IN-SKILL PRODUCTS

ACCOUNT LINKING

PERMISSIONS

Models

Use this page to add or remove eligible models from this skill.

Save

MODEL NAME	DESCRIPTION	REQUIREMENT	STATUS
Custom	Design a unique experience for your users. A custom model enables you to create all of your skills interactions.	Custom model service endpoint needs to be set and the voice interface needs to be defined with invocation name, sample utterances, and intents.	<input checked="" type="checkbox"/>
Smart Home	Give users control of their smart home devices. "Alexa, turn on the kitchen lights"	Smart home service endpoint needs to be set and Account Linking needs to be configured.	<input type="checkbox"/>

4. Confirm your changes and save:

alex developer console

< Your Skills Sample Custom Build Code Test Distribution Certification Analytics

English (US)

CUSTOM

MODELS

IN-SKILL PRODUCTS

ACCOUNT LINKING

PERMISSIONS

Models

Use this page to add or remove eligible models from this skill.

Save

MODEL NAME	DESCRIPTION	REQUIREMENT	STATUS
Custom	Design a unique experience for your users. A custom model enables you to create all of your skills interactions.	Custom model service endpoint needs to be set and the voice interface needs to be defined with invocation name, sample utterances, and intents.	<input checked="" type="checkbox"/>
Smart Home	Give users control of their smart home devices. "Alexa, turn on the kitchen lights"	Smart home service endpoint needs to be set and Account Linking needs to be configured.	<input checked="" type="checkbox"/>

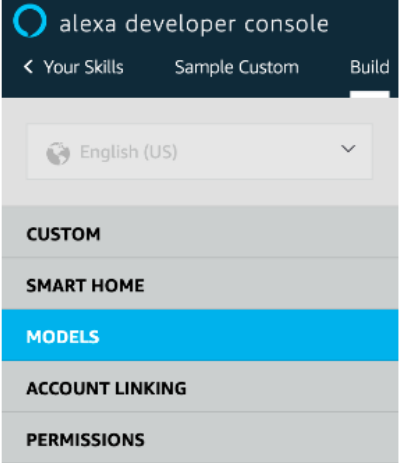
Confirm Changes

Models to be added

Smart Home

Cancel Save

5. That’s it, your skill is now a Multi-Capability Skill with two models in the selected locale (EN_US). Note that you have to add both models for each locale.



After adding a model to your skill, you will need to properly handle the functionality of the model in your skill logic in the same way you would previously handle custom or smart home requests in separate skills. For example, if you have an existing skill that uses a smart home model and you would like to update it to use a custom model, you would also need to add skill logic to handle the session requests from the custom interaction model either manually or with one of the Alexa Skill Kit Software Development Kits (SDKs) for [Node.js](#), [Java](#), or [Python](#). This means evaluating the incoming request and responding appropriately to incoming custom session or smart home directive requests.

How to Support State and Change Reporting With MCS

You can support [state and change reporting](#) for device endpoints when you upgrade a custom skill to MCS. Your skill needs to follow the guidelines of [sending events to Alexa gateway](#) and how to [authenticate a customer to Alexa with permissions](#). To allow existing customers to benefit from the state and change reporting experience, Alexa will run an automated silent process within a week from the upgrade date to send your skill an *AcceptGrant* directive for each existing customer. You will receive an email with the results.

Supporting Multiple Languages and Locales with MCS

If you plan to support more than one language or locale, your MCS must have an instance of each skill model type for each language. For example, if you intend to support English in both Canada and the US, you will need a custom model and a smart home model for each language. The custom model should at least handle the basic Amazon built-in intents.

Drive Customer Awareness of Your Device’s Expanded Features

When you upgrade an existing smart home skill to MCS by adding a custom model, you should educate your users about the new custom capabilities and how to use them. Let users know how they can interact with the custom model capabilities by adding your device’s newly supported utterances on your skill description page, product website, and other mediums to drive awareness of your enhanced voice experience.

Learn More

You can read more about Multi-Capability Skills in the documentation [here](#), with details on how to upgrade existing skills using the developer console or with the ASK CLI, and additional FAQs.

Back to Top

Alexa Skills Kit

- [Alexa Skills Kit](#)
- [Learn](#)
- [Design](#)
- [Build](#)
- [Launch](#)

Resources

- [Getting Started](#)
- [Tutorials](#)
- [Documentation](#)
- [Developer Forum](#)
- [Agencies and Tools](#)

Alexa Voice Service

- [Alexa Voice Service](#)
- [Learn](#)
- [Design](#)
- [Build](#)
- [Launch](#)

AVS Resources

- [Getting Started](#)
- [AVS Device SDK](#)
- [AVS API](#)
- [Dev Kits for AVS](#)

Connected Devices

- [Alexa Smart Home](#)
- [Alexa Gadgets](#)

Agreements

- [Agreements and Terms](#)
- [Program Materials License Agreement](#)
- [Amazon Developers Services Portal Terms of Use](#)

Blogs

- [Alexa Skills Kit Blog](#)
- [Device Makers Blog](#)
- [AWS Blog](#)
- [Alexa Science](#)

Support

- [Amazon Developer Support](#)
- [Contact Us](#)
- [Forums](#)
- [Manage Email Preferences](#)

Follow Us:

