

Purpose

A majority of Alexa 3P high-value developers (HVDs) distribute their skills to users in non-English-speaking locales. In August, for example, Alexa B2BD had █ HVDs, about █ of which publish non-English skills in international locales (Figure 1). Alexa offers a wealth of technical documentation to support these global developers, but the content is integrated into our doc set. Some developers might not even know it's there.

We'd like to improve the findability of our locale-specific tech docs content. This initiative aligns with the Alexa B2BD group's goal of █ among global 3P HVDs. The objectives of this document are to explore options and recommend a solution for surfacing locale-specific tech docs content. We welcome your feedback and ideas.

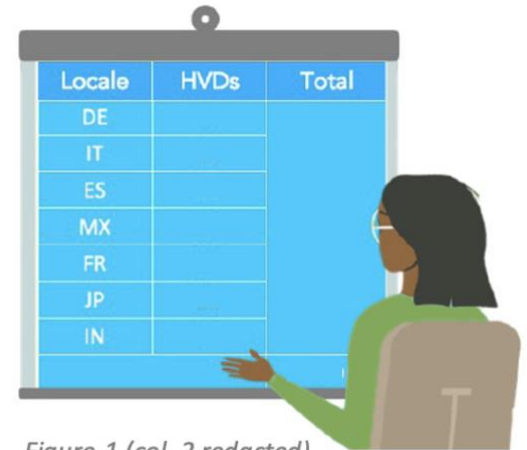


Figure 1 (col. 2 redacted)

Background

Attracting and retaining international high-value developers (HVDs) is a key priority of the B2BD group. The number of global HVDs █. According to the Alexa Developer Education (ADE) July 20█ Monthly Business Report (MBR), the number of global HVDs has █. That trend extended into August (█% of goal, and █ MoM).

HVD locales

Locale is determined by the language in which a skill is offered and the country in which it's distributed, such as en-US (English, United States). A skill's locale doesn't necessarily match the developer's preferred language or geographic location, but it does coincide with its users' language and country of residence. See Appendix 1 for details.

HVD definition

Until recently, any 3P developer with at least █ users was considered an HVD. Now an HVD is defined as any 3P developer that captures at least █% monthly active users (MAUs) in a given locale. Take █, for example, which manufactures Alexa-enabled █, █, and other smart home devices. In August, this vendor had █ monthly active users (MAUs) in the fr-FR locale. That's less than half the number needed to qualify as an HVD under the old definition. █'s user base in France, however, represents █% of the █ MAUs in that locale—more than twice the size needed to qualify under the revised definition.

Dx for global developers

As shown in Figure 2, locale-specific tech documentation is written in English, but gives customized advice for building skills in global locales. For example, [Support for Right-to-left Languages in the Responsive Components and Templates](#) helps developers build skills in Arabic and other right-to-left languages.

29

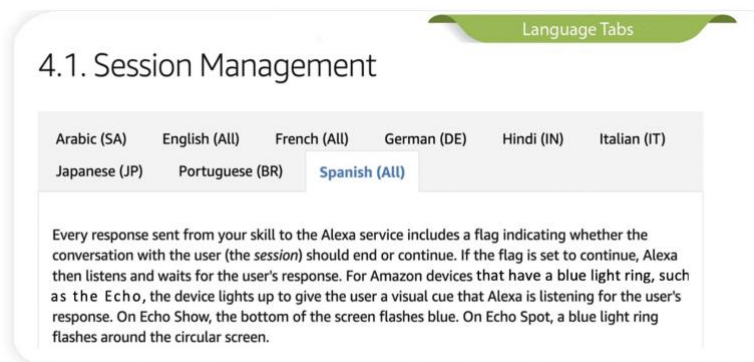


Figure 2

Alexa offers some locale-specific content as discrete documents, such as [Use Hindi in Your Alexa Skill](#) and [Speechcon Reference \(Interjections\): Japanese \(JP\)](#). Much of the documentation, however, is presented as tabular material within other pages. For example, on the [Voice Interface and User Experience Testing for a Custom Skill](#) page, locale-specific [session management content](#) appears on a set of tabs (see Figure 2).

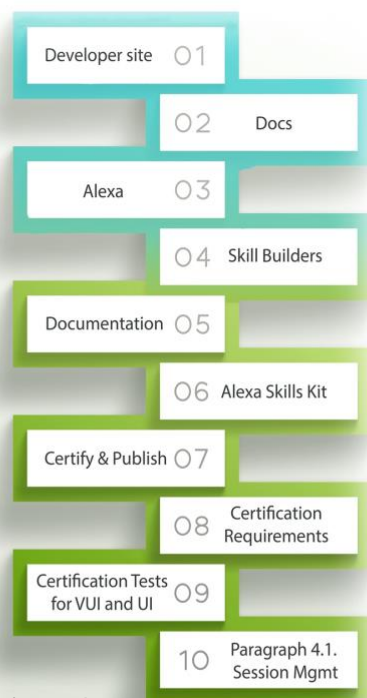


Figure 3

Options

We have extensive, up-to-date content for the locales in which we want to attract and retain Alexa HVDs. The placement of locale-specific documentation within relevant conceptual material is logical, but developers sometimes have to drill down many levels to find it. Take the session management tabs mentioned in the preceding section, for example. As shown in Figure 3, this tabbed content is nine levels below developer.amazon.com. How can we make locale-specific content easier to find?

Option 1: Do nothing

We could decide to leave well enough alone. After all, the information for global developers is available, even if it's not ready for its close-up. Inertia is free and requires no engineering or tech docs resources. Going this route, though, demonstrates neither customer obsession nor bias for action.

Option 2: Create a new interface

We could consolidate the embedded locale-specific content and place it together in a new interface. Doing so would highlight our commitment to expanding Alexa skills internationally. It would also showcase the depth of resources available for each locale.

However, some of the locale-specific information simply wouldn't make sense out of context. In addition, this approach might create a mirror image of the issue we're trying to solve. Instead of paging through conceptual documentation to search for language resources, developers would need to link out from the locale-specific content to find associated conceptual documentation. Furthermore, building a new interface would require significant engineering resources.

Option 3: Duplicate the content

Duplication would entail retaining the existing tabular material in context and, in addition, creating a new interface to house the collected content. This combined content would appear out of context, though, and building a new interface would consume scarce engineering resources. Moreover, duplication is not a best practice. The content would need to be maintained in two places, increasing the writers' workload and inevitably introducing conflicts.

Option 4: Create a locale-specific search feature (recommended)

Creating a locale-specific search feature would offer the advantages of a new interface without the complexity of content duplication or the expense of building a new interface. It would showcase our locale-specific content without the reproducing it.

A locale search could be accomplished with a Coveo in-product experience (IPX) similar to that used for our [reference content](#). Coveo is an AI-driven search platform to which we already subscribe. An IPX is an advanced Coveo filter that funnels tagged content into a predesigned interface. A user could click the de-DE tile, for example, to see an index of all language-specific content for Alexa developers producing skills in German. Aside from the staff resources required to tag the content, there's no additional cost to create an IPX.

We could add this search-by-locale feature to the developer documentation submenu. The title bar mock-up in Figure 4 shows how it might look. Alternatively, instead of "Locale," we could call it "Language Content" or

77 something similar. In addition, to elevate the visibility of the feature, we've approached the Horizon team
78 about integrating a locale search into the new developer portal.

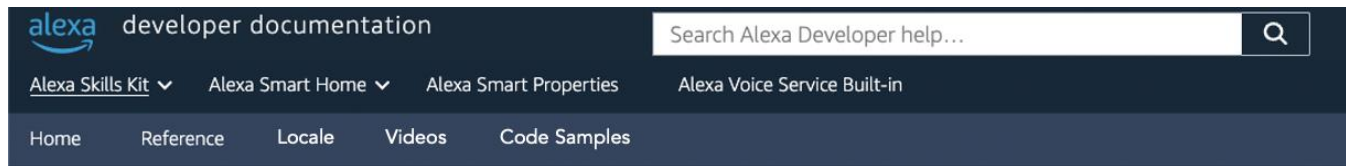


Figure 4

Next steps

79 The next steps are to get feedback on these options, consider any other proposed solutions, and settle on a
80 course of action. If the recommended option is selected, we can explore technical details, solicit customer
81 feedback, estimate the time commitment, and seek buy-in from the wider team.

Summary

82 Is the recommended solution sound? Have we overlooked any sources of supporting data? Have we missed any
83 possible risks? How might we tweak the proposed locale search feature to reach new developers or encourage
84 engagement among existing devs? We encourage you to share your thoughts and ideas for surfacing locale-specific
85 content to better serve our global skill builders.

86 **Appendix 1**

87 [Appendix content redacted]