

语言和语言区域

本文内容：

>

解析语言资源所面临的挑战

>

对资源解析策略的改进

>

设计您的应用以支持附加语言区域

从 Android 7.0（API 级别 24）开始，Android 为多语言用户提供增强的支持，让他们可以在设置中选择多个语言区域。Android 通过大幅扩展受支持的语言区域数量并更改系统解析资源的方式来提供此功能。

本文档首先说明低于 7.0（API 级别 24）的 Android 版本中的资源解析策略，接着介绍 Android 7.0 中改进的资源解析策略，最后说明如何充分利用扩展的语言区域数量来支持更多的多语言用户。

解析语言资源所面临的挑战

在 Android 7.0 之前，Android 并非始终能够成功匹配应用和系统语言区域。

例如，假设您遇到了以下情况：

- 您的应用的默认语言为 `en_US` (US English)，但它也在 `es_ES` 资源文件中对西班牙语字符串进行了本地化。
- 将设备设置为 `es_MX`

当您的 Java 代码引用字符串时，系统会从默认 (`en_US`) 资源文件加载字符串，即使应用在 `es_ES` 下有本地化的西班牙语资源。这是因为当系统无法找到精确匹配时，它会继续通过将国家/地区代码从语言区域中剥离来查找资源。最后，如果未找到匹配，系统会恢复为默认模式，即 `en_US`。

如果用户选择应用根本不支持的语言（如法语），则系统也会默认显示 `en_US`。例如：

表 1. 没有精确语言区域匹配项的资源解析。

用户设置	应用资源	资源解析
fr_CH	默认值 (en)	尝试 fr_CH => 失败
	de_DE	尝试 fr => 失败
	es_ES	使用默认值 (en)
	fr_FR	
	it_IT	

在此示例中，系统在不知道用户是否理解英语的情况下显示英语字符串。目前，此行为很常见。

对资源解析策略的改进

Android 7.0（API 级别 24）可提供更稳健的资源解析，并自动查找更好的备用方法。不过，为了加速解析和提升可维护性，您应以最常用的母语存储资源。例如，如果您之前将西班牙语资源存储在 `es-US` 目录中，请将它们移动到 `es-419` 目录，该目录包含拉丁美洲西班牙语。同样，如果您在名为 `en-GB` 的文件夹中存储有资源字符串，则将此文件夹重命名为 `en-001`（国际英语），因为 `en-GB` 字符串的最常用母语为 `en-001`。以下示例介绍为什么这些做法可提升性能和资源解析的可靠性。

资源解析示例

使用 7.0 以上的 Android 版本时，以不同的方式解析表 1 中所描述的案例：

表 2. 针对没有精确语言区域匹配项时改进的资源解析策略。

用户设置	应用资源	资源解析
1. fr_CH	默认值 (en) de_DE es_ES fr_FR it_IT	尝试 fr_CH => 失败 尝试 fr => 失败 尝试 fr 的子项 => fr_FR 使用 fr_FR

现在，用户获得的是法语资源而不是英语。此示例还表明对于 Android 7.0 或更高版本，您为什么应将法语字符串存储在 `fr`（而非 `fr_FR`）中。此处的操作是匹配最接近的母语，从而使解析更快速且更具预见性。

除了这个改进的解析逻辑外，Android 现在还提供更多的用户语言以供选择。下面，我们将意大利语指定为附加用户语言，但假设应用不支持法语，再次尝试上面的示例。

表 3. 应用仅与用户的次优语言区域设置匹配时的资源解析。

用户设置	应用资源	资源解析
1. fr_CH 2. it_CH	默认值 (en) de_DE es_ES it_IT	尝试 fr_CH => 失败 尝试 fr => 失败 尝试 fr 的子项 => 失败 尝试 it_CH => 失败 尝试 it => 失败 尝试 it 的子项 => it_IT 使用 it_IT

用户仍会获取他们理解的语言，即使应用不支持法语。

设计您的应用以支持附加语言区域

LocaleList API

从 Android 7.0（API 级别 24）开始，Android 显示的 `LocaleList.getDefault()` API 可让应用直接查询用户已指定的语言列表。您可以使用此 API 创建更成熟的应用行为和更优化的内容显示。例如，搜索可以基于用户的设置以多种语言显示结果。浏览器应用可避免翻译以用户理解的语言显示的页面，键盘应用可自动启用所有适用的布局。

格式化程序

直到 Android 6.0（API 级别 23），Android 仅支持许多常用语言（en、es、ar、fr、ru）的一个或两个语言区域。由于每种语言只有几种变体，因此，应用可以通过在资源文件中将一些数字和日期存储为硬编码字符串解决此问题。不过，随着 Android 扩展了支持的语言区域集，即使在一个语言区域中，日期、时间、货币及类似信息也会存在显著差异。对您的格式进行硬编码会让最终用户困惑不已。因此，在针对 Android 7.0 或更高版本开发应用时请务必使用格式化程序代替硬编码数字和日期字符串。

例如，Android 7.0 和更高版本支持 27 个阿拉伯语言区域。这些语言区域可以共享大多数资源，但其中一些资源首选 ASCII 数字，另一些则首选本地数字。例如，如果您想要创建一个具有数字变量的句子，如“Choose a 4 digit pin”，则按如下所示使用格式化程序：

```
format(locale, "Choose a %d-digit PIN", 4)
```