

# WearHF

## 编程风格指南



版本 1.0

# WearHF 编程风格指南

法律信息.....	3
修订信息.....	3
简介.....	4
看到什么说什么.....	5
数字叠加.....	6
关闭数字叠加.....	7
自定义数字叠加.....	8
隐藏命令.....	9
显示帮助.....	10
应用程序风格指南.....	11
分辨率.....	11
字体大小.....	11
应避免情况.....	12
选择语音命令.....	13
长词与短词.....	13
硬词和软词.....	13
HMT 设备编程工具.....	14

## 法律信息

- 使用 HMT-1 设备，以及任何嵌入在此类设备中的软件或由 RealWear 提供的软件，均须遵守您与 RealWear 签订的书面协议（即“先锋项目协议”、“保密协议（NDA）”等）（统称为“协议”）。
- 本文档是技术指南，而不是法律文件。文档中的任何内容均不会更改协议的任何条款，也不授予您超出协议之外的任何权利或许可。
- 本指南是 RealWear 根据您与 RealWear 签订的保密协议向您提供的保密信息，因此您不得将本指南透露给其他方。
- 本指南包含第三方软件（如 Android Studio）的引用。对第三方软件的使用需遵循适用的第三方许可规定。
- 本指南为熟练的程序员提供了有关 HMT-1 设备编程的介绍，但不保证任何结果。
- RealWear 可能会不时更新本指南。您应该确保使用的是最新版本。

## 修订信息

V1.0	10-23-2016		初始版本

## 简介

HMT-1 和 HMT-1Z1 头戴式计算设备代表了下一代可穿戴的坚固计算平台。这些设备的独特性在于免提操作——没有触摸屏幕、不需要滑动、不需要手势——整个计算机都通过语音和简单的头部动作进行操作。



然而，同时，HMT-1 和 HMT-1Z1 也是标准的 Android 6.0 设备——运行标准的 Android 应用程序并使用标准的 Android 应用程序开发工具。

当然，这一切都是通过设计实现的。HMT 设备必须具有免提功能才能在现实生活中真正有用。但与此同时，不要求应用程序开发人员改变现有的 Android 应用程序来加入免提功能。这就是为什么要使用 WearHF™。

WearHF 是运行 Android 应用程序的一种创新方法。它本质上是一个系统服务，嵌入在 Android 操作系统及其应用程序之间的操作系统核心上。

WearHF 设计用于在运行时自动检查所有应用程序，提取屏幕上显示的用户控件信息，并将其转换为免提操作，如语音命令或头部动作控件。所有这一切都以闪电般的速度无缝发生，而且应用程序开发人员无需了解语音识别或动作引擎。WearHF 在二进制应用程序上运行——这意味着现有的 Android 应用程序可以部署到 HMT 设备上并免提运行，不需要修改任何代码。

这就是为什么，HMT-1 和 HMT-1Z1 无需特别的 SDK。

也就是说，了解 WearHF 的工作方式将使开发人员能够优化HMT 用户的体验，并能够根据需要为特定用例配置 WearHF...



## 看到什么说什么

所有按钮、菜单、复选框和屏幕上出现的其他可点击用户界面组件在运行时会自动询问并提取其名称。这些名称会被送入始终开启的语音识别器。

因此,如果一个应用程序向用户提供一个屏幕按钮标记为“我的应用程序”,用户只需说出“我的应用程序”即可激活该按钮。就是这么简单!

应用程序本身对语音识别器一无所知——这些都是自动发生的,在应用程序中是不可见的。



在编程层面, WearHF 会检查当前屏幕的 UI 树并查找被标记为“可点击”的组件。该标记默认设置为按钮和文本字段,并可以在 `layout.xml` 中显式指定。

一旦识别出可点击的 UI 元素, WearHF 将从组件中提取文本供语音识别器使用。

该文本具有 `android:text` 的属性

如果文本属性为空或相当于 `NULL`, 将使用 `android:contentDescription` 属性。

## 数字叠加

除了定位可点击的 UI 元素并使用语音识别器中的元素名称之外，WearHF 还会自动指定另一种方式来选择 UI 元素。这就是所谓的数字叠加。

屏幕上的第一个可点击 UI 元素将被标记为项目“1”。默认情况下，该 UI 元素左上角会显示一个紫色的圆形标记，用户现在可说出“选择项目 1”来以编程方式点击该元素。

页面上的所有可点击的 UI 元素都重复使用该方法，每个元素可获得唯一的数字标识符。

除了说出该组件的全名外，用户也可以选择说出该数字。例如。在此处显示的截屏中，用户可以说“选择项目 2”或者“账户”来以编程方式打开“账户”选项。



此数字叠加主要用于屏幕上没有文本表示的项目（例如图像按钮），或用户不能读出的文本表示（例如，'文件名：40d38xy.doc'）。实际示例可参见上述截图中的数字叠加 1：搜索图标。该图标没有任何文本可读出来激活图标，所以用户将可以说“选择项目 1”来打开搜索功能。

## 关闭数字叠加

数字叠加是控制应用程序的一个备用方法，对于未明确设计用于 HMT 设备的应用程序尤其有用。

然而，数字叠加也可能会分散注意力，而且在设计 HMT 使用的应用程序时也不需要。

要关闭数字叠加功能也很简单。只需将一个特殊命令嵌入到表单顶部的 `android:contentDescription = hf_no_overlay` 字段中。

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent" android:layout_height="fill_parent"
    android:id="@+id/topLevel"
    android:contentDescription="hf_no_overlay"
/>
```

数字叠加可以在按顺序分层级进行控制。在 `layout.xml` 文件中，`android:contentDescription = hf_no_overlay` 可出现在屏幕上任何可点击 UI 元素的中。当解析到该命令，数字叠加功能将从此处停止工作，直到出现配对命令为止。

配对命令为 `android:contentDescription = hf_yes_overlay`，同样可以写在 `layout.xml` 文件的任何需要启用数字叠加的地方。

开发人员可以选择在使用这两个命令的之间部分，关闭部分屏幕的数字叠加功能，并在其他部分重新打开数字叠加功能。

这在文件浏览器等应用程序中非常有用。屏幕中心的文件图标将需要使用数字叠加，因为不保证能够读出文件名。然而，文件列表上方或下方的用户控件可以很好地定义成供语音识别器使用，而且不需要使用数字叠加。

## 自定义数字叠加

数字叠加功能可以进一步改善，即让开发人员设计叠加层的外观、感觉和位置，并让 WearHF 系统分配和处理编号部分。

要通知 WearHF 使用自定义叠加层，需定义一个 UI 文本视图控件，并告诉 WearHF 使用该控件作为数字叠加层。这一点通过向文本视图控件添加以下内容描述来完成：

**`android:contentDescription = hf_overlay_number`**

如果 WearHF 检测到页面上只有一个 `hf_overlay_number` 命令，将自动关闭数字叠加层，转而使用已标记的文本视图元素。系统将用数字填充文本视图，并使用文本视图控件的字体、颜色和大小。

## 隐藏命令

有时候为用户提供屏幕上不可见的一些语音命令可能更为方便。

这一点可通过将按钮添加到屏幕布局，并将其可见属性设置为隐藏来实现。

WearHF 仍然会接收这些按钮并听取这些语音命令是否说出。一旦说出这些语音命令，这些按钮将以编程方式激活。然而，这些按钮在屏幕上不可见。

## 显示帮助

记得使用“显示帮助”的方法来查看这些隐藏命令的列表。

WearHF 的设计理念是屏幕上应显示所有重要的语音命令，因此用户不会疑惑该怎么说。

然而，某些应用程序可能会提供较长的语音命令列表，其中一些命令通常不那么重要，并且不需要一直显示在屏幕上。

在这种情况下，强烈建议开发人员使用内置的“显示帮助”功能来弹出这些附加的、不可见命令的列表。

“显示帮助”是一种通用命令，用户可在任何时候在任何屏幕中使用。

默认情况下，说出“显示帮助”命令将在窗口中弹出一个可用语音命令的列表，或者至少弹出当前屏幕上不可见的命令列表。



应用程序可以在运行时使用特殊的 `Broadcast Intent` 填充此列表。该功能名称为 `'com.realwear.wearhf.intent.action.UPDATE_HELP_COMMANDS'`。

帮助窗口的命令列表使用以下标识符作为字符串数组提供：

`'com.realwear.wearhf.intent.extra.HELP_COMMANDS'`。

```
final ArrayList<String> helpCommands = new ArrayList<>();
helpCommands.add("Command One");
helpCommands.add("Command Two");

final Intent intent = new Intent("com.realwear.wearhf.intent.action.UPDATE_HELP_COMMANDS");
intent.putStringArrayListExtra("com.realwear.wearhf.intent.extra.HELP_COMMANDS", helpCommands);
intent.putExtra("com.realwear.wearhf.intent.extra.SOURCE_PACKAGE", getPackageName());
intent.setPackage("com.realwear.wearhf");
sendBroadcast(intent);
```

注意：“显示帮助”中提供的帮助命令仅供用户参考。任何文本组合都可以使用。语音识别器不限制使用的命令文本。

## 应用程序风格指南

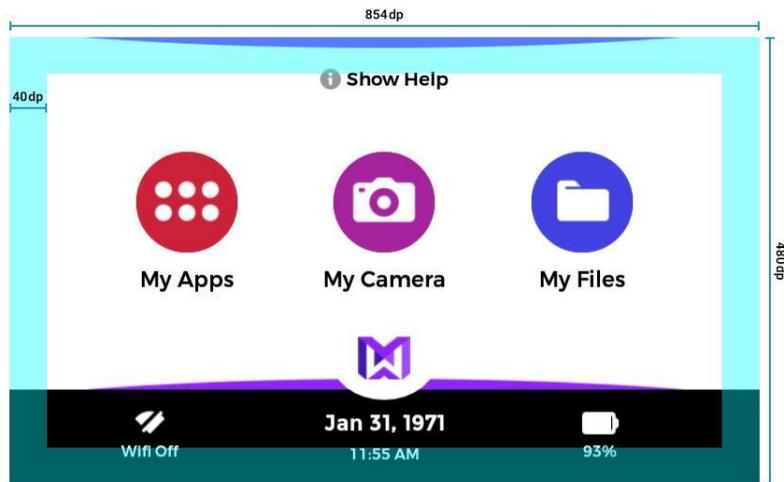
HMT 应用程序设计为快速访问信息。这些应用程序应简洁并简单易用，指导用户快速高效地访问所需信息。

在风格方面，应用程序在设计为高对比度时效果最佳。通常使用大字体和亮色的白色背景。

## 分辨率

屏幕分辨率为 854dp X 480dp。

还建议在屏幕周围保留一个 40dp 的禁用区。这样目镜的定位将更为广泛——用户可以将目镜放在离眼睛较远处，但确保仍然可以看到所有相关信息。



## 字体大小

字体应足够大，以便用户清楚地查看。推荐的最小字体大小是 16sp，但是可以使用 14sp 作为常规文本。

## 应避免情况

小的文本信息和不明显的着色方案（如深灰色上使用浅灰色）将不能在小的、高对比度显示屏上很好地转换。

重要的信息和命令应远离屏幕边缘，最好显示在屏幕的中心位置。

## 选择语音命令

语音命令应仔细选择。

不恰当的语音命令将造成 99% 的识别度与低于 50% 的识别度之间的差异。反过来，这意味着用户有着良好的应用体验与决定不再使用该应用之间的差异。

语音命令的选择有一些基本规则：

- 避免一个词的命令
- 词语越长，识别效果越好（在方便的前提下）
- 词语越难（发音上），识别效果越好

## 长词与短词

词语越长，识别器听取和分析语音命令的时间就越长。

一个词的命令，无论对用户而言有多合理，通常会导致较低的 ASR 性能。

诸如“现在继续”之类的词语比“OK”更容易被识别。“OK”这类短词很容易被一些背景噪声或咳嗽等错误识别。然而，“现在继续”足够长，因此不会被错误识别。注意，“继续”也许能够很好地发挥作用，但是打破了“不要使用一个词作为命令”的基本规则。

设计中经常使用的一个技巧是在命令之前添加一个无意义词语。例如，WearHF 使用“我的照片”和“我的文档”。在这些命令中，第一个词语是一个无意义词语，第二个词才是行动命令。

## 硬词和软词

这里的硬和软是指音节的难易程度。发音柔和的词不容易被识别器识别，而发音突出的词语更容易捕获。

例如，一个常用的命令可能是“**Home**”以跳转至主屏幕。然而，这是一个软词，没有什么明显特征可供识别器识别。但如果将“**Home**”替换成“**Previous**”，将达到更好的识别效果，因为“**Previous**”有两个突出的辅音。然而，“**Previous**”也是一个短词。WearHF 中使用的更好的解决方案是“Go Previous”。

## HMT 设备编程工具

HMT 设备本质上是标准的 Android 6.0 设备。这意味着所有标准的 Android 编程工具和开发环境都可以用于开发 HMT-1 和 HMT-1Z1 的应用程序。

多数 Android 开发人员使用 Google 的 Android Studio 进行开发工作。我们也一样！我们使用所有的标准插件、标准的 SDK 和帮助程序，而且不需要任何定制的 SDK 来使用 HMT-1 或 HMT-1Z1。



然而，我们开发了一个叫“RWExplorer”的特殊开发工具。RWExplorer 是运行于 Windows PC 和 Mac 电脑的一个桌面工具，可将 HMT-1 的屏幕镜像到桌面。这意味着作为一名开发人员，您不需要继续在 HMT 微型显示屏上查看应用程序正在执行的操作，而是直接在桌面上查看相同屏幕的大型版本。此外，您可以使用 PC 鼠标和键盘来驱动 HMT 设备上的用户界面。



RWExplorer 是一个非常方便的工具，可以立即在 RealWear Developer Portal 上下载。