

WearML 开发人员指南



版本 1.1

### WearML 开发人员指南

# 目录

去律信息	2
修订信息	2
简介	3
WearML——嵌入源代码中	5
在 android:contentDescription 字段中嵌入指令	6
嵌入式 WearHF 指令示例	8
WearML——配置二进制代码	9
使用 RWExplorer 捕获 UI 树	10
JI 节点	11
固定节点	11
孚动节点	11
修改 UI 树	12
WearML 基本项	12
WearML 指令——标头项	14
WearML 指令——固定节点项	15
WearML 指令——浮动节点项	16
WeaML——开发流程	17
手动部署 UI 树修改	18

### 法律信息

• 使用 HMT-1 设备,以及任意内置的软件或由 RealWear 提供的其他配件,均须遵守您与

RealWear 签订的书面协议(即"先锋计划协议"、"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

草案	11-17-2016	Chris Parkinson	内部审查的初始版本
1.1	1-10-2017	Chris Parkinson	首次随 WearML 1.1 公开发布

### 简介

WearHF 是用于支持 HMT 头戴设备操作功能的基础软件服务。通过结合声音和头部动作,实现以免提方式使用标准的触摸驱动 Android 应用程序的目的。

WearHF 在大多数情况下是全自动的,不需要开发人员有任何输入即可免提启用应用程序。 对于专门为 HMT 头戴设备开发的应用程序,该系统运行良好。

然而,一些应用程序有着复杂的用户界面屏幕。WearHF 虽然能够以免提方式启用这些应用 程序,但有时候方法过于复杂。

这就是为什么要引入 WearML。WearML 允许根据应用从程序各个方面进行手动调整和定制,以实现自动免提功能。

在特定的应用程序上使用 WearML 可以将其从一个复杂的应用程序变为一个在免提模式下高效且直观地工作的应用程序。

WearML 不需要修改任何应用程序源代码,也不会改变原来的应用程序。只是加入一组提示和指令,可以使 WearHF 系统有效运行。

作为其有效性的一个示例,可参考以下关于在标准的 Skype 应用程序上运行 WearHF 的两个 屏幕截图。

第一个截图显示了当 WearHF 尝试免提启用应 用程序时发生的情况——这时界面上出现了 32 个可选按钮,没有一个按钮有与之相关的 逻辑名称,而且其中大部分都不是按钮! 要在聊天窗口中发消息,用户必须说"选择 项目 25",然后界面会出现一个虚拟键盘,用 户必须逐个字母地拼写。这种用户体验必然 不尽人意。





3 •		♥ 🖄 🖬 436	
<	meisgeghra, BellaSong1772, cindy.wang82,	<b>4 8 1</b>	
	Thursday Bellidong1772 Good	ére Call déo Call	٥
0	MildSoug1772 Please let me know the results		0
0	Thanks Trinh Hyperes		
Spea	ak Now here	Send Message	Δ
A		Show Help	

第二个截图显示了运行定制 WearML 的相同 应用程序。通过定制,屏幕上大多数可选按 钮都被忽略了,从而极大地清理了屏幕。新 的按钮允许用户再音频或视频模式间切换, 其他唯一活动的语音按钮被标记为"开始说 话"(speak now)和"发送消息"(send message)。

要发送消息,用户只需简单地说"开始说话"

即可启用语音听写服务。该听写服务允许用户对着耳机直接说话。当用户说完时,等待 2 秒左右,口语文本将立刻转译至聊天窗口。

如果用户对获取的文本满意,只需简单地说"发送消息"即可在 Skype 上发送文本消息。

因此,WearML的使用体验成为以免提方式操作 Skype 聊天的一种更为直观有趣的方式。

WearML——嵌入源代码中

WearML 是一种标记语言,可用来帮助 WearHF 服务在免提的情况下正确或更有效地启动 Android 应用程序。WearML 由可与用户界面(UI)树关联的指令列表组成。

在构建标准的 Android 应用程序时,开发人员必须通过将 UI 组件集合到 canvas (画布)上 来构建用户界面屏幕。可使用拖放式可视化编辑器 (例如 Android Studio)或以编程方式 在代码中完成。

无论以哪种方式,结果都是 ViewGroups (容器)和 Views (容器内的 UI 元素)的分层列表。

WearHF 通过在运行时遍历这些分层列表并提取特征来工作。例如,如果 WearHF 检测到有按钮嵌入到 UI 树中,将复制按钮文本并将其发送到语音识别器进行侦听。

但有时屏幕上的可点击按钮没有相关文本——这些按钮通常都是图像按钮。在这种情况下, WearHF将自动以"选择项目1"(selectitem1)的形式提供语音指令,并在控件旁边叠加 一个紫色数字。因此,用户必须说出"选择项目1"来激活该按钮。

对于简单的 UI 树来说,这种系统很有效——分析按钮文本、复选框和其他"可点击"项目 并传输至语音识别器。大多数情况下,用户可以说出屏幕控件的名称,但在少数情况下,可 能需要使用"选择项目 1"符号。

然而,真正难的是复杂的 UI 树。一些 UI 树可能包含 30 个或更多的可点击按钮,许多图形 图标都没有相关联的文本。这种屏幕上会出现大量需说出的紫色数字。

接下来就是 Wear ML 发挥作用的时刻。Wear ML 是一组可嵌入到 UI 树源代码中的提示和指令, 以指导 Wear HF 如何更好地处理用户界面。简单来说, Wear ML 可用于提示真正的口头命令而 不是"选择项目 1"命令,或者简单地告诉 Wear HF 忽略一组控件,且不启用这些控件的语 音功能。

当 WearML 指令添加至源代码后,这些指令会以文本的形式添加至 'android:contentDescription'字段,出现在每个 UI 组件中。

下表显示了用于嵌入源代码的 Wear ML 指令列表。

### 在 and roid: content Description 字段中嵌入指令

命令	源格式	描述
关闭数字	hf_no_number	关闭此元素和所有 UI 子元素的数字索引和叠加层(即没有选择
		项目1)
打开数字	hf_yes_number	打开此元素和所有 UI 子元素的数字索引和叠加层(用于逆转
		no_number 命令的效果)
		所有 UI 树默认设置成打开数字。
语音命令使用文本字	hf_use_text	明确指定此元素和任何后续的UI子元素都应该在语音识别器命
段		令中使用 android:text 字段的内容。
		该命令是用于覆盖在 android:text 和
		android:contentDescription 字段之间自动选择的默认行为。
		通常情况下, android:contentDescription 字段优先于
		android:text 字段。
语音命令使用描述字	hf_use_descrip	明确指定此元素和任何后续的UI子元素都应该在语音识别器命
段	tion	令中使用 android:description 字段的内容。
		该命令是用于覆盖在 android:text 和
		android:contentDescription 字段之间自动选择的默认行为。
		通常情况下, android:contentDescription 字段优先于
		android:text 字段。
关闭头部跟踪滚动	hf_scroll_none	明确关闭整个页面的头部跟踪滚动。
		WearHF将通过寻找"可滚动"的UI元素来确定是否需要滚动。
		hf_scroll_none 指令用于覆盖这种自动行为。
打开元素的水平头部	hf_scroll_hori	明确指定打开该页面的水平头部跟踪滚动,且滚动活动应针对
跟踪滚动	zontal	当前的 UI 元素。
打开元素的垂直头部	hf_scroll_vert	明确指定打开该页面的垂直头部跟踪滚动,且滚动活动应针对
跟踪滚动	ical	当前的 UI 元素。
		如果 UI 树上发现有可滚动的元素,将默认设置为垂直头部跟踪
		滚动。



命令	源格式	描述
关闭 PTT-Home	hf_no_ptt_home	关闭当前屏幕的 PTT-Home 行为。
行为		在正常使用情况下,按 PPT 按钮将自动切换至主屏幕。但该指令
		用于禁用当前屏幕的这种行为,允许应用程序充分利用 PPT 按钮
		本身。
		PPT 按钮用作标准按键/释放按键代码=500
描述文本字段输	hf_text_normal	描述一个文本字段,以帮助 WearHF 向用户呈现正确的数据输入工
入类型	hf_text_dictatio	具。
	n	当激活文本字段时,会出现一个虚拟键盘,允许用户输入文本。
	$hf\_text\_barcode$	默认显示一个完整的虚拟键盘,供用户说出字母和数字。
		指定听写字段将使 WearHF 直接显示文本听写对话框,允许用户直
		接说出文字。
		指定条形码字段将使 WearHF 显示条形码阅读器,并作为文本字段
		的键盘口。
		默认文本字段设置为 hF_text_nornal
		注释(TODO):这些功能在1月31日发布的版本上可用。还扩展
		了指定条形码类型的选项
		当出现标准虚拟键盘时,当前条形码和听写可作为子选项。

在 and roid: content Description 字段中嵌入 Wear IF 指令

注:多个嵌入式 WearHF 指令可在 and roid: content Description 中一行显示,通过使用"|" 字符来分隔命令。例如,

android:contentDescription=" hf\_no\_overlay/hf\_no\_ptt\_home"

(接近 UI 树顶部的该命令会导致所有的数字叠加层被关闭,并且当前屏幕的 PTT Home 功能 会被禁用。)

### 嵌入式 WearHF 指令示例

以下是一个简单的代码示例,显示了如何将 WearHF 指令嵌入到 Android 应用程序的 XML 格式文件中。

在该示例中,第一个 WearHF 指令会关闭屏幕上的所有数字叠加层。即,屏幕上任何地方都不显示紫色带圈数字。

第二个和第三个指令用于告知 WearHF 在字符串"zoomLevel1"和"zoomLevel2"中定义的 具体语音命令,从而激活两个单选按钮。注意,这些单选按钮只显示在屏幕上,但没有任何 标签。

向用户显示可用的语音命令是一种良好的实践。以下示例通过在两个单选按钮上增加一个文本标签来实现这个功能,显示'zoom\_level\_instruction'资源中定义的字符串'zoom level 1-2'。

<relativelayout< td=""></relativelayout<>
android:id="@+id/footer"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_gravity="bottom"
android:layout_alignParentBottom="true"
android:layout_alignParentStart="true"
android:contentDescription="hf_no_number">
<textview< td=""></textview<>
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="@string/zoom_level_instruction"/>
<radiobutton< td=""></radiobutton<>
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/radioButton1"
android:contentDescription="@string/zoom_level_1"
android:onClick="onZoomLevel1" />
<radiobutton< td=""></radiobutton<>
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/radioButton2"
android:contentDescription="@string/zoom_level_2"
android:onClick="onZoomLevel2" />

代码示例——WearML 指令可用于关闭树中的所有原色的紫色带圈数字,并可用于为UI 元素 提供明确的语音命令,这些UI 元素通常没有与之相关的文本。 WearML——配置二进制代码

WearML 嵌入式指令在应用程序的源代码级别使用时效果很好。但在很多情况下,应用程序的源代码不可用,只有二进制、预编译的应用程序可用。

在这些情况下,WearML 也可用于修改和控制这些二进制和第三方应用程序的免提性质。

为使 WearML 与二进制应用程序共同作用,应用程序的用户或管理员必须基本编写所需的 UI 变更脚本,并将其作为特殊文件存储在耳机上。这些变更将由 WearHF 读取并在运行时修改 二进制应用程序。

整个过程已经简化,并纳入了 RWExplorer 工具中,可从 RealWear 开发人员网站上获取。基本流程如下所示:



使用 RWExplorer 捕获当前运行的二进制应用程序的 UI 树。

在文本编辑器中编辑 UI 树以应用 WearHF 指令

将修改过的 UI 树保存回 EIMT 以供运行时使用

在二进制应用程序中使用 WearML 的工作流程



### 使用 RWExplorer 捕获 UI 树

RWExplorer 有一组用于编辑 WearML 的特殊控件。点击主窗口上的"WearML"按钮即可显示这些控件。



点击 WearML 按钮可访问这些 WearML 控件

WearML 窗口本身有两个按钮:

### 捕获

该按钮将对当前可见的应用程序窗口执行新的捕获操作。捕获的 UI 树将作为可编辑的 XML 文档在标准文本编辑器中呈现

刷新

该按钮将使可编辑 XML 文档中所做的任何更改都保存至耳机中,并且当前的应用程序会 重新加载这些更改信息。

### UI 节点

UI 树由 UI 节点的层次树表示,大多数 Android 开发人员都很熟悉。RWExplorer 捕获的树非 常密切地反映了用于生成 Android 应用程序的 layout.xml 文件。

通常情况下,有一个顶级的 View 项目,该项目封装了一些 UI 视图子组件,这些组件本身可 以包含自己的子项。



示例——UI 树表示为UI 元素的分层列表

在本文档中,每个注释的 UI 元素将被称为节点。标准应用程序中的节点可能是一个按钮、 一个复选框或任何其他重要的 UI 控件。

WearML 的理念是,操纵这些节点使其在免提环境中更易于使用。

例如,节点可以更改,从而为按钮提供一个简单的语音激活命令。

当 WearML 修改节点时,实际上不修改原始应用程序,而是在应用程序上绘制叠加层以实现 其视觉外观。

### 固定节点

固定节点是直接对应于屏幕上用户界面组件的节点。例如,按钮(Button)或文本视图(Text View)组件就是一个固定节点。

### 浮动节点

浮动节点是不对应于 UI 树中项目的节点。浮动节点代表屏幕上特定的鼠标点击点。

### 修改 UI 树

对应用程序 UI 树的修改本质上是指标记出对应用程序所做的更改。这是通过引用要更改的 节点并将这些更改保存到文本文件中完成的。

例如,如果用户要修改上述屏幕,将选项1(Choice1)、选项2(Choice2)和选项3(Choice3) 替换成新的语音命令(苹果、梨和橘子),相应的WearML文件将如下所示:



- WearML 不需要包含不变更的项目
- WearML 文件不需要保留原始 UI 树的分层结构

### WearML 基本项

### 各 WearML 文件必须在前几行包含下列项目:

<WearML>
<Package>待修改的应用程序包名</Package>
<Language>en\_US</Language>
<UniqueIdentifier content\_description= "key\_words"/>
...此处插入 UI 变更作为要变更的 UI 节点列表...
</WearML>

### 包名

包名是指修改的应用程序的完整包名。

语言

修改的语言代码包含在文件中。



该参数允许不同的修改文件被应用于每种语言,允许 WearML 实现本地化。如果使用其他语言,但没有找到其他的 WearML 语言文件,则将使用第一个 WearML。

唯一标识符(UniqueIdentifier)

许多应用程序可能有着简单的包名,但可能有多个屏幕。唯一标识符用于通过特定的WearML 文件链接至特定屏幕。

唯一标识符是与待修改屏幕的 UI 树中一段数据相匹配的一个字符串。唯一标识符可在 UI 树的 contentDescription、id 或文本字段中搜索。

### WearML 指令——头标签

WearML	
Document Element	必须
Package	
com.example.app	必须:捕获屏幕时自动生成。是指 Wear ML 所属的包名。
Language	
ex_EX	可选: 表示此文件的目标语言。如果未指定,将默认为屏
	幕使用的语言。
UniqueIdentifier	
id=" y"	必须:至少必须指定其中一个属性来标识屏幕在整个包中
text=" yy"	的唯一组件 (View)。
<pre>content_description=" yyy"</pre>	

### <WearML>

<Package>com.android.calculator</Package>

<Language>en\_US</Language>

<UniqueIdentifier id="calc\_app\_1"/>

</WearML>

WearML 指令——固定节点项

所有节点项属于<View>元素。

View	
root=" true false"	可选:只能应用到最顶层的 View 元素。根的属性被应用到所有 View
	子项,除非这些子项定义了属性。
id="y"	必须:至少必须指定其中一个属性来标识组件。
text=" yy"	
content_description	
=" yyy"	
<pre>speech_command=" te</pre>	可选: 定义语音命令的来源。
<pre>xt content_descript</pre>	text 将从组件获取文本属性。
ion no xxxx"	content_description 将使用组件的属性。
	no 将关闭组件上的语音命令。
	xxxx 您也可以在此处提供自定义语音命令,例如, xxxx
overlay_show_number	可选:打开或关闭紫色数字叠加层。(默认=是)
=" yes no"	
overlay_show_text=	可选:打开或关闭文本标签。标签文本将从设置的 speech_command
"yes no"	中获取。(默认=否)
overlay_persists="	可选:数字和/或叠加层不会消失。(默认=否,消失)
yes no"	
overlay_orientation	可选: 文本叠加方向(默认=右)
="left right top b	
ottom"	

### <WearML>

<Package>com.android.calculator</Package>

<Language>en\_US</Language>

<UniqueIdentifier id="calc\_app\_1"/>

<View id=button1 overlay\_show\_text="yes" overlay\_orientation="top" />

</WearML>

## WearML 指令——浮动节点项

所有节点项属于<View>元素。

View	
ху="х, у"	必须: 当检测到语音命令时点击屏幕上的 XY 点
text=" yy"	该节点使用的文本和语音命令
overlay_show_number=	可选:打开或关闭紫色数字叠加层。(默认=是)
"yes no"	
overlay_show_text=" y	可选: 打开或关闭文本标签。标签文本将从设置的
es no"	speech_command 中获取。(默认=否)
overlay_persists=" ye	可选:数字和/或叠加层不会消失。(默认=否,消失)
s no"	
overlay_orientation=	可选: 文本叠加方向(默认=右)
"left right top bott	
om"	
text_field=" dictatio	可选: 在文本字段元素上, 该项将指示应该打开哪个键盘。
n keyboard barcode"	
barcode=" any  qr code	可选:将定义正在扫描哪种类型的条形码。如果 text_field 未
128   upc   ean   "	设置成条形码将忽略。(默认=任何)
global_commands=" yes	可选: 禁用所有全局命令并隐藏"显示帮助", 与应用哪个组件
no"	无关。
broadcast_results=" y	可选: 通过单独的目标
es no"	<b>com. realwear. wearhf. intent. action. SPEECH_EVENT</b> 广播 ASR
	结果,包括置信度。(默认=否)
	这将影响所有的命令,无论添加至哪个视图。

1

### WeaML——开发流程

Use <u>RWExplorer</u> to capture the UI tree for the currently running binary application

Q Ourse Brising 172 Links month	0.00
88	
0.0	
© Contention of the second second	
Barts and here proceeded	
8 0	0
0000000	() Dew Hele

使用 RWExplorer 捕获当前运行的二进制应用程序的 UI 树。

点击"捕获"(Capture)按钮来捕获新的 XML 格式文件。 系统将要求开发人员输入名称以保存文件——文件名应可读。 选择了文件名后, XML 文件将以文本编辑方式呈现

2 Edit the UI tree in a text editor to apply WearHF directives

countries.	
Calendary and the second s	
(has	
(Anno 1999) and an and a second s	
s/bases	0000
	A Second state of the seco

在文本编辑器中编辑 UI 树以应用 WearHF 指令

编辑 XML 文本文件以移除不需要修改的所有 UI 元素。最终的 WearML 应大大删减,只显示需要更改的那些元素。

3	
S	ave modified UI tree back to HMT for runtime
u	se

C methoday Bulleting (117, 100), months		
0 11.0	Se L	Ï
C For the last number of a smaller franker		¢
Contract of the second		
N	INTER O	0
at 70 M in th 19 70	III Shaw Help	

将修改过的 UI 树保存回 HMT 以供运行时使用

首先保存文本编辑器视图,然后按下 RWExplorer 中的"刷新"(refresh)按钮,将更改保 存到耳机中。

保存后,结果将上传到耳机,并导致当前应用程序加载新的 UI 更改。

手动部署 UI 树修改

WearML 文件可以由开发人员生成,也可以由其他方提供。WearML 文件必须放置在以下目录中的耳机文件系统上:

• /external storage/.wearml/

在这个文件夹中,每个需要优化的软件包都包含一个子文件夹。而在包文件夹内将会有一些 WearML文件,每个文件对应一个需要在应用程序中优化的页面。

例如,要修改 Skype 应用程序中包名为 'com. skype. raider\_7.22.0.322.apk' 的 UI 树的两 页,相应的 WearML 文件可能为

- /external storage/.wearml/com.skype.raider/main\_layout.xml
- /external storage/.wearml/com.skype.raider/chat\_layout.xml

WearHF 示例——更改主屏幕

	() Show Help	
		0
My Programs	My Camera	My Files
	_ M	
WH OF	Jan 9, 1970	100%

### 这个简单的示例显示了如何修改简单的应用程序,如主屏幕。



# <WearML> <Package>com.realwear.launcher</Package> <Language>en\_US</Language> <UniqueIdentifier content\_description="My Programs"/> <View content\_description="My Programs" overlay\_show\_text="yes" overlay\_show\_number="no" overlay\_persists="yes"/> <View xy="400,180" text="Start the Camera" overlay\_show\_text="yes" overlay\_persists="yes" /> </WearML>

简单修改——在"我的程序"(My Programs)按钮上添加叠加层,然后添加浮动节点以激活 "我的相机"(My Camera)按钮。

