Samsung Gear Application Programming Guide

Version 1.0

Revision History

Version	Date	Description
1.0	2014.07.22	Initial release

Table of Contents

1. OVERVIEW 6

1.1 ARCHITECTURE 6
1.2 APPLICATION TYPES 7
1.2.1 LINKED (MASTER-FOLLOWER) 7
1.2.2 INTEGRATED 8
1.2.3 STANDALONE 9
1.3 GEAR APPLICATION PACKAGE STRUCTURE 9

2. INSTALLING GEAR APPLICATION 12

2.1 INSTALLING WIDGETS ON SAMSUNG GEAR 12
2.2 UNINSTALLING WIDGETS ON SAMSUNG GEAR 13

3. CONFIGURING GEAR APPLICATION 14

4. CONFIGURATION MENU 16

4.1 CONFIGURING THE SETTINGS 17
4.1.1 Configuration File for Basic Settings UI 17
4.1.2 Configuration File for Complex Settings UI 26
4.1.3 Backup and Restore settings of link mode 28
4.2 STRUCTURE OF THE CONFIGURATION FILE (“PACKAGENAME_SETTTINGS.XML”) 29
4.2.1 <Application> 29
4.2.2 <AppSetting> or <ClockSetting> 30
4.2.3 <packageName> 30
4.2.4 <DisplayName> 30
4.2.5 <MotherPackageName> 30
4.2.6 <MotherPackageIntentAction> 31
4.2.7 <MotherActivity> 31
4.2.8 <MotherExtra> 32
4.2.9 <key> 32
4.2.10 <Value> 32
4.2.11 <Settings> 32
4.2.12 <MotherCondition> 33
4.2.13 <onTrue> 33
4.2.14 <onFalse> 34
4.2.15 <Item> 34
4.2.16 <Title> 36
4.2.17 <SubTitle> 36
4.2.18 <CheckBox> 36
4.2.19 <CheckboxItem> 37
4.2.20 <Radiobox> 37
4.2.21 <RadioboxItem> 38
4.2.22 <Condition> 38
4.2.23 <SwitchButton> 38
4.2.24 <SwitchItem> 39
4.2.25 <TimePicker> 39
4.2.26 <DatePicker> 40
4.2.27 <Favorites> 40
4.2.28 <FavoritesItem> 41

4.2.29 <DisplayName> ... 41
4.2.30 <ClassName> ... 41
4.2.31 <IsAppWidget> ... 41
4.2.32 <PackageName> ... 42
4.2.33 <PackageIntentAction> ... 42
4.2.34 <Extra> ... 42
4.2.35 <key> ... 43
4.2.36 <Value> ... 43
4.2.37 <ON-CLICK> ... 43
4.2.38 <PhoneSideIntentAction> ... 44
4.2.39 <PhoneSideExtra> ... 44
4.2.40 <GearSideIntentAction> ... 44
4.2.41 <GearSideExtra> ... 45
4.3 MULTI-LANGUAGE SUPPORT ... 45

5. DELIVERING APPLICATION SETTING RESULTS ... 47

5.1 SETTINGS FROM BASIC SETTINGS UI ... 47
5.1.1 <Application> ... 48
5.1.2 <packageName> ... 48
5.1.3 <SettingsResult> ... 48
5.1.4 <Item> ... 49
5.1.5 <CheckBox> ... 50
5.1.6 <Radiobox> ... 50
5.1.7 <SwitchButton> ... 50
5.1.8 <DatePicker> ... 51
5.1.9 <TimePicker> ... 51
5.1.10 <Favorites> ... 51
5.2 SETTINGS FROM COMPLEX SETTINGS UI ... 52

6. USING NOTIFICATION ... 53

6.1 CONFIGURING HOST-SIDE PROVIDER ... 53
6.2 SENDING NOTIFICATION (HOST -> WEARABLE) ... 54
6.3 ACKNOWLEDGING NOTIFICATION (WEARABLE -> HOST) ... 55
6.4 CLEARING NOTIFICATION (HOST -> WEARABLE) ... 56
6.5 SHOW ON DEVICE ... 56

COPYRIGHT ... 58

About This Document

This document describes how to install and uninstall Gear application and configure host application.

This document introduces Gear Manager and provides a simple example.

This document does not describe development tools and guides, which are covered elsewhere in the SDK documentation.

1. Overview

1.1 Architecture

Wearable device is paired with a host device, normally a smartphone or connected tablet, on which it depends for wider connectivity and various services. The host runs an application known as the Gear Manager, which communicates with an application on the wearable device known as the WearableManagerService, these are used for managing applications.

Applications themselves are also two-part: a Wearable-side widget and a corresponding piece running on the host, usually operating as a consumer/provider pair.

The following figure shows the architecture of the host and wearable software including Gear Manager.

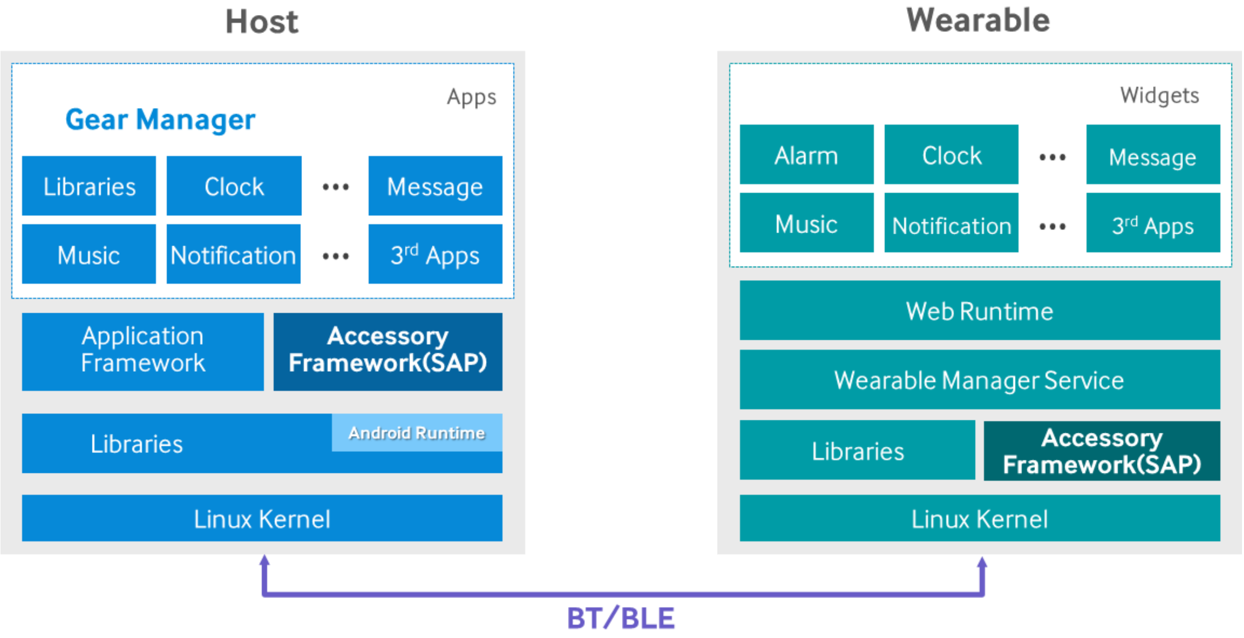


Figure 1: Architecture of the host and wearable device

- **Gear Manager:** You can use Gear Manager to install, configure and manage your Wearable-side Widget applications on Wearable device
- **WearableManagerService:** receives Wearable-side widget from Gear Manager and installs them on the wearable, and forwards followed setting information to Gear Manager.

1.2 Application Types

The following figure shows the application package structure required by Gear Manager.

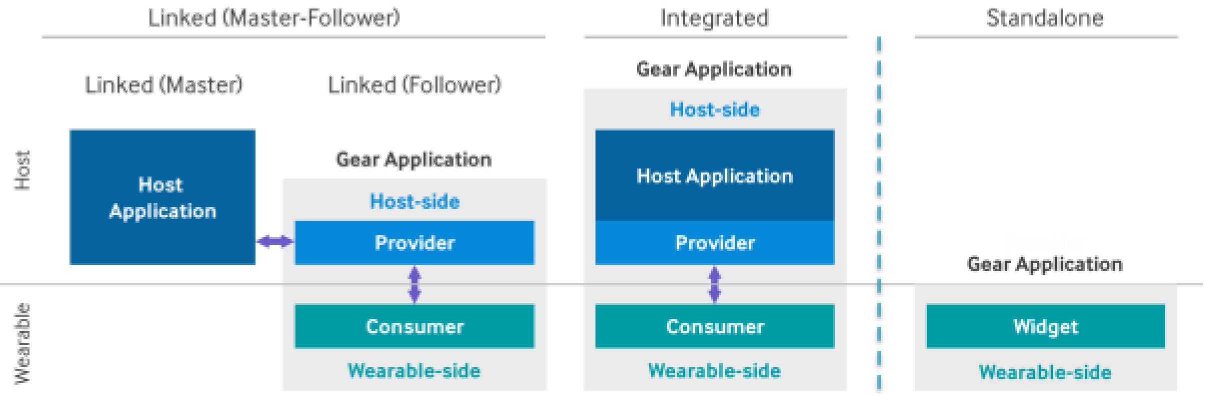


Figure 2: Application package structure

1.2.1 Linked (Master-Follower)

The Linked Type consists of the Host-side application and the Wearable-side application.

The Wearable-side application is not included with the general host application. You need to install the general host application and the Wearable-side application.

To enable host devices to recognize Linked applications and store recommendations, add the following permission to Android manifest file in the Host-side application (AndroidManifest.xml).

```
<uses-permission android:name=”com.samsung.WATCH_APP_TYPE.Linked_Follower”/>
```

To upload Gear applications to the Samsung Gear Apps Store, add the following metadata to the AndroidManifest.xml for the application that works with your wearable-side widget.

```
<meta-data android:name="GearAppType" android:value="wgt"/>
```

the version codes of Gear application must have different numbers.
If you try to register Gear applications with the same version code, the Samsung Gear Apps Store cannot register your new application.

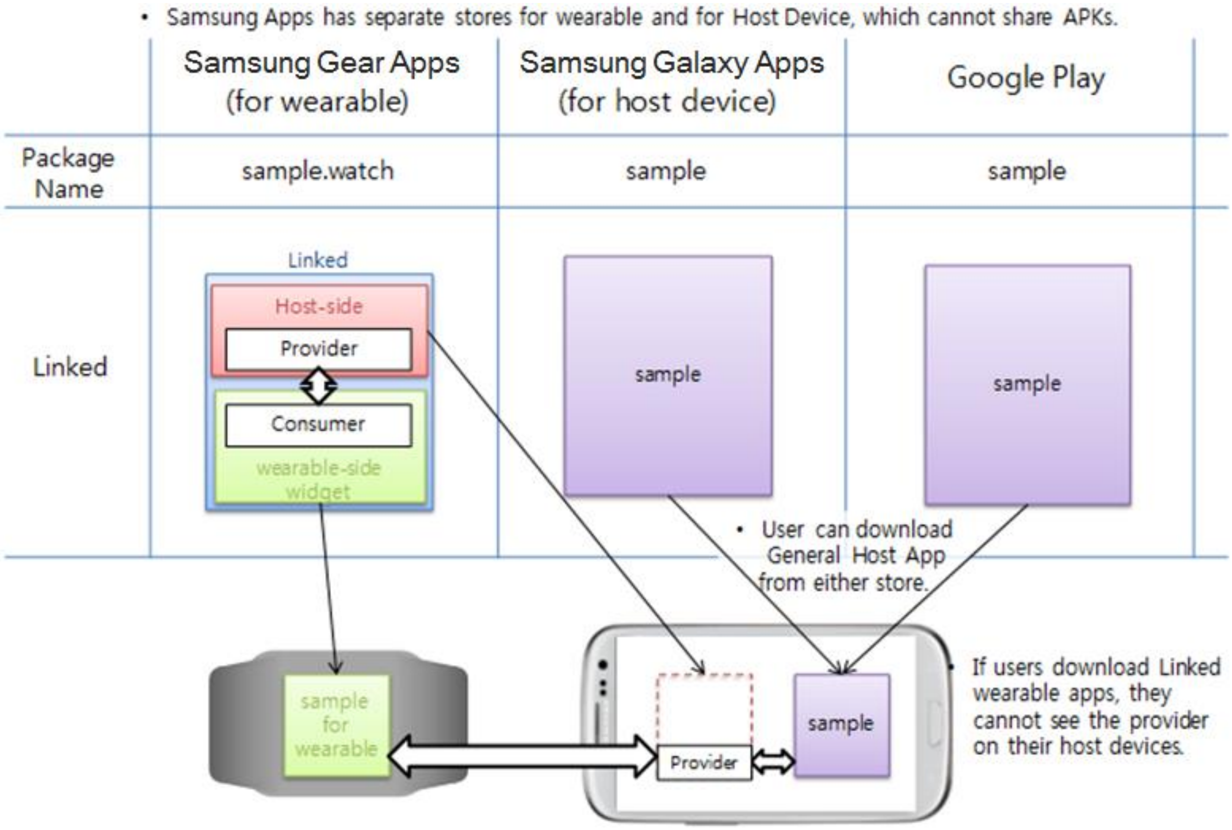

• Samsung Apps has separate stores for wearable and for Host Device, which cannot share APKs.
Samsung Gear Apps (for wearable)
Samsung Galaxy Apps (for host device)
Google Play
Package Name
sample.watch
sample
sample
Linked
Linked
Host-side
Provider
Consumer
wearable-side widget
sample
sample
• User can download General Host App from either store.
sample for wearable
Provider
sample
• If users download Linked wearable apps, they cannot see the provider on their host devices.

Figure 3: Linked Gear Application Installation

1.2.2 Integrated

Integrated applications consist of a host-side APK and a wearable-side widget.

The host-side APK contains the general host app. When users download an Integrated application, the general host app in the host-side APK is installed on Host devices

Integrated applications operate in the same manner as Linked applications, but they only have one (1) APK instead of the two (2) APKs of Linked applications. Integrated Gear package names must be different than those used for general host apps. If you use the same names, there can be problems when users update Integrated applications.

To enable host devices to recognize Integrated applications and store recommendations, add the following permission to your general host Android manifest file (AndroidManifest.xml).

```
<uses-permission android:name="com.samsung.WATCH_APP_TYPE.Integrated"/>
```

To upload Gear applications to Samsung Gear Apps Store, add the following metadata to the AndroidManifest.xml for the application that works with your wearable-side widget

```
<meta-data android:name="GearAppType" android:value="wgt"/>
```

the version codes of Gear application must have different numbers.
If you try to register Gear applications with the same version code, the SamsungApps Store cannot register your new application.

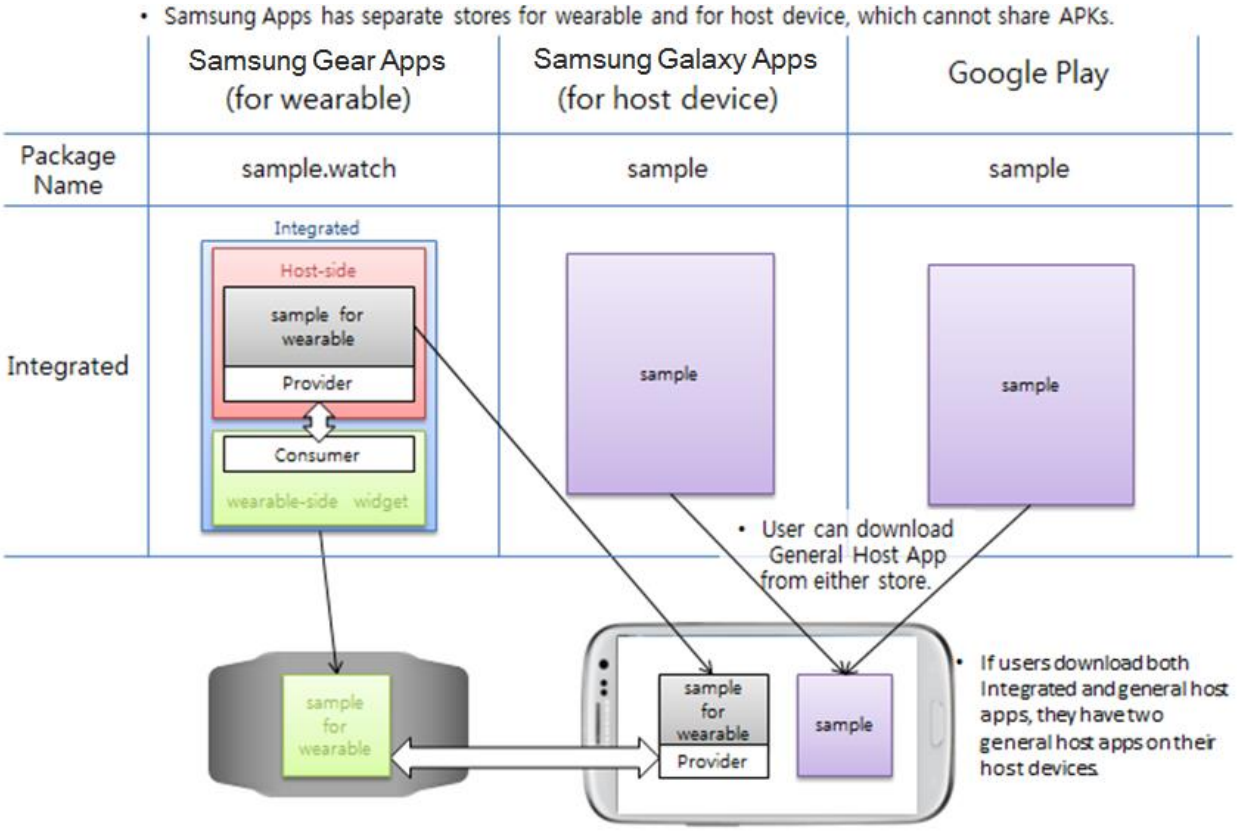

• Samsung Apps has separate stores for wearable and for host device, which cannot share APKs.
Samsung Gear Apps (for wearable)
Samsung Galaxy Apps (for host device)
Google Play
Package Name
sample.watch
sample
sample
Integrated
Integrated
Host-side
sample for wearable
Provider
Consumer
wearable-side widget
sample
sample
• User can download General Host App from either store.
sample for wearable
sample for wearable
Provider
sample
• If users download both Integrated and general host apps, they have two general host apps on their host devices

Figure 4: Integrated Gear application Installation

1.2.3 Standalone

You can use the standalone type, such as a Clock app, when your Gear application does not need a general host application, and can operate independently.

1.3 Gear Application Package Structure

The following figure shows the application package structure required by Gear Manager.

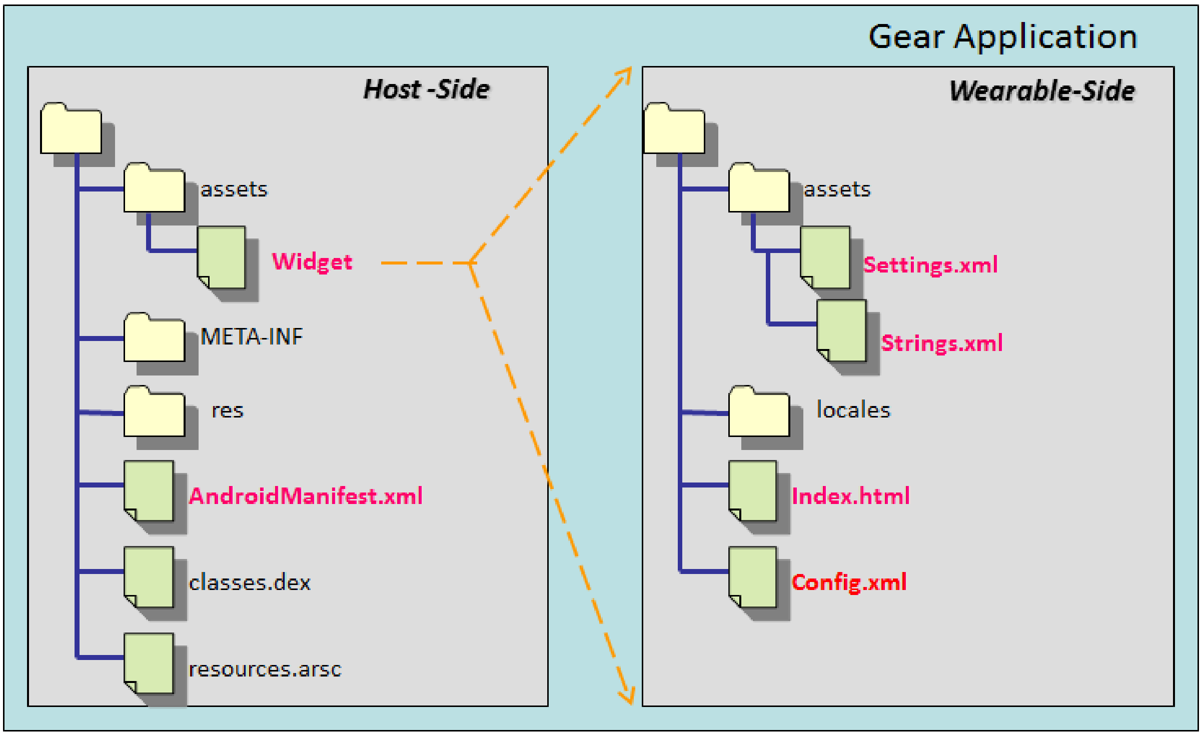


Figure 5: Internal package layout

To install an application on a wearable device, the application package must contain a host-side app and a wearable-side widget. The wearable-side widget must be in the assets folder of the host-side application.

When users install a host-side application, the following steps occur:

1. Android package manager install the Host-side application (from market or other root) on the host device.
2. Gear Manager sends the Wearable-side widget to WearableManagerService on the wearable device.
3. WearableManagerService installs the Wearable-side widget on the wearable device.

To use Gear Manager's settings and notification functionality, modify the host-side application AndroidManifest.xml file and the wearable-side widget config.xml and settings.xml files.

- The host-side application Android manifest file is used for host-side application installation..

- The wearable-side application `config.xml` file is used for wearable-side widget installation..

- The wearable-side widget `settings.xml` file is used for configuring the widget. For more information, see Adding a Configuration Menu to a Wearable Side Widget. If you want to get more information, please refer to Chapter 5.

Gear Manager uses the XML settings file to configure the Wearable-side Widget.

2. Installing Gear Application

Users cannot directly install or uninstall wearable-side widgets on wearable device, instead Gear Manager can be used for these functions.
But user doesn't do additional install process.
Just install Host-side application on the host device and Gear Manager will install Wearable-side widget on wearable device.

The following figure illustrates the sequence.

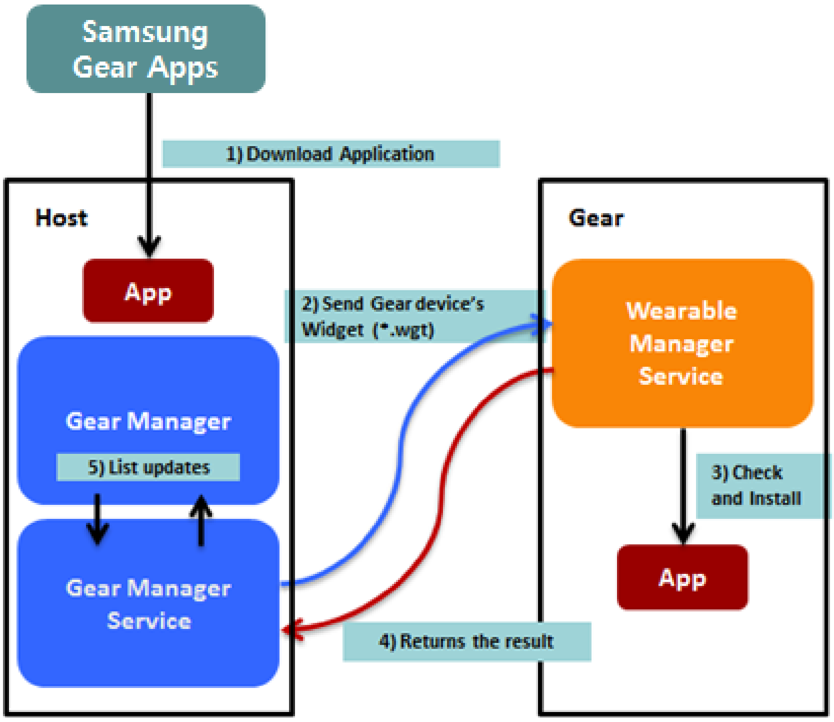


Figure 6: Installation Diagram

2.1 Installing Widgets on Samsung Gear

To install wearable-side widgets,

1. Users download an APK file with a wearable-side widget to a host device.
2. Gear Manager extracts the wearable-side widget from the assets folder of the host-side APK and transfers it to WearableManagerService on the wearable device.
3. WearableManagerService installs the widget.
4. WearableManagerService checks the category of the widget.

   WearableManagerService checks for `<tizen: category name=”com.samsung.wmanager.WATCH_CLOCK” />` in the `config.xml` file of the installed wearable-side widget to distinguish a clock widget from a general widget (clock widgets get special handling).

5. WearableManagerService caches the widget info and sends it to Gear Manager.
6. Gear Manager updates the widget list after it receives the widget info from WearableManagerService. It updates the clocks if is a clock widget, otherwise it updates **My apps**->**all** if the widget is not a clock widget

2.2 Uninstalling Widgets on Samsung Gear

To uninstall wearable-side widgets,

1. Click **Gear Manager**->**My apps**.
2. Select the trash icon in the upper right corner.

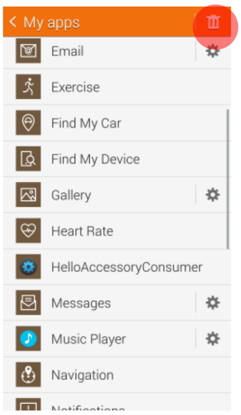


Figure 7: Uninstall Widgets

3. Select the application that you want to delete.

3. Configuring Gear Application

If you want to make a host-side provider for a wearable-side widget, you must provide Androidmanifest.xml.

To register your application with Gear Manager, add "com.samsung.wmanager.APP" to the uses-permission (highlighted in yellow) element of the host-side application's Android manifest file.

If your application is a Linked (Master-Follower) application, add the master app information using metadata tags (highlighted in green) to the AndroidManifest.xml. Gear Manager uses this information to notify users if the master application is absent or is later uninstalled from the Host.

meta-data	Description
master_app_name	Describes the master app name
master_app_packagename	Describes the master package name
master_app_s amsungapps_deeplink master_app_playstore_deeplink master_app_3rd_url_deeplink	Describes the market deep link of the master app
GearAppType	Describe application type. This is new feature for Samsung Gear. If you want to make application for Gear1, you don't need this meta-data.

Tablet 1: Meta data

When it comes to determining app priorities in the China model, master_app_3rd_url_deeplink is higher than master_app_samsungapps_deeplink.

In non-China models, master_app_samsungapps_deeplink is higher than master_app_playstore_deeplink.

Sample Code

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.samsung.android.app.watchmanager"
    android:versionCode="1"
    android:versionName="1.0" >
    <uses-sdk
        android:minSdkVersion="14"
        android:targetSdkVersion="17" />
    <uses-permission android:name="com.samsung.wmanager.APP"/>
    <application
        <activity>
                …
        </activity>
    <meta-data
    android:name="master_app_name"
            android:value="master App"/>
    <meta-data
      android:name="master_app_packagename"
           android:value="com.example.masterapp"/>
    <meta-data
      android:name="master_app_samsungapps_deeplink"
           android:value="Samsungapps deeplink URL"/>
    <meta-data
           android:name="master_app_playstore_deeplink"
           android:value="playstore deeplink URL"/>
    <meta-data
           android:name="master_app_3rd_url_deeplink"
           android:value="3rd url deeplink URL"/>

    <meta-data
           android:name="GearAppType"
           android:value="wgt"/>
    </application>
</manifest>
```

4. Configuration Menu

If you wish to use Gear Manager to manage the configuration of your wearable-side application, include a `settings.xml` file in the settings folder in the wearable-side widget. This file tells Gear Manager what settings information is required for a wearable-side widget.

Note
Gear Manager uses the XML configuration file for launching the basic or complex settings UI. The XML settings file uses a String ID format to support multiple languages. To support multi languages for your application, add the required strings in all the supported languages. The file name must be settings.xml.

The following figure shows the steps that occur when a wearable–side XML settings file is sent to Gear Manager on a host device:

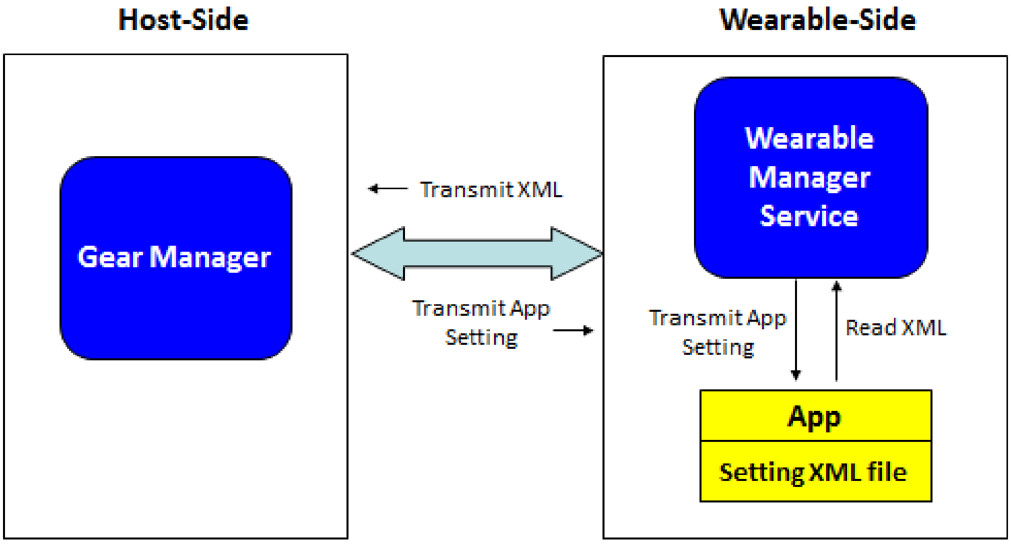


Figure 8: Setting Diagram

The following sequence of steps occurs when Gear Manager settings are used:

1. WearableManagerService imports and saves the settings.xml file under the settings folder.
2. WearableManagerService converts the String ID-format XML file to the current language set, and sends it to Gear Manager.
3. Depending on whether XML or Link Mode is specified in the settings file, Gear Manager calls the basic settings UI(XML) or the complex settings UI (Link Mode).
4. When the user is done configuring the settings, Gear Manager passes the resulting XML to WearableManagerService.
5. WearableManagerService delivers the XML file to the app.
6. The app parses the XML file to get the correct settings values.

4.1 Configuring the Settings

You can share and configure the settings for your application in the following ways:

- **Basic settings UI derived from XML settings file**

  This basic method uses the settings XML file to construct basic views such as CheckBox and RadioBox. For more information on the structure of the XML file, see Configuration File for Basic Settings UI.

- **Complex settings UI from your application**

  Link Mode allows you to create a complex settings page when the configuration is beyond the capabilities of the basic settings UI. You supply 'package name + activity name' in the settings file and Gear Manager launches the corresponding custom settings page when necessary. For more information on using Link Mode, see Configuration File for Complex Settings UI.

4.1.1 Configuration File for Basic Settings UI

Use the following guidelines for creating the XML settings file for your widget when you are using the basic settings UI:

- Use the settings/strings.xml file in your app to define the values used for the strings in the XML settings file (settings/settings.xml)
- Add "@" in the String ID in strings.xml to distinguish between a Tag (that uses the String ID) and a non-Tag.

  <string name="Clock_Name">Weather Clock</string>

  <DisplayName>@Clock_Name</DisplayName>

- Values between open and close Tag elements should not contain spaces.

  <Title>@Show_Date</Title>

- The package Name and Condition elements do not use String IDs.

Sample Code : strings.xml

```
<resource>
   <string name="Clock_Name">Weather Clock</string>
   <string name="24_Hour_Format">Use 24-hour format</string>
   <string name="Hour_Format_Item1">1:00 PM</string>
    ……
</resource>
```

Sample Code : settings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<Application xmlns:android="http://schemas.android.com/apk/res/android"
        type="clock" version="001"language="english">
<packageName>samsung.com.watch.weatherclock</packageName>
<DisplayName>@Clock_Name</DisplayName>
<Settings>
    <Item id="24format" title_type="title_subtitle" setting_type="checkbox">
       <Title>@24_Hour_Format</Title>
                <SubTitle options="checkbox">@Hour_Format_Item2</SubTitle>
                <CheckBox id="24format" checked="yes">
                        <CheckBoxItem>@Hour_Format_Item1</CheckBoxItem>
                        <CheckBoxItem>@Hour_Format_Item2</CheckBoxItem>
                </CheckBox>
    </Item>
    <Item id="showdate" title_type="title" setting_type="checkbox">
           <Title>@Show_Date</Title>
           <CheckBox checked="yes"/>
    </Item>
    <Item id="weathertype" title_type="title_subtitle" setting_type="radiobox">
                    <Title>@Weather_Type</Title>
                    <SubTitle>@Weather_Type_Item1</SubTitle>
                    <Radiobox id="weathertype" number="2" selected="1">
                             <RadioboxItem>@Weather_Type_Item1</Radio_Item>
                              <RadioboxItem>@Weather_Type_Item2</Radio_Item>
                    </Radiobox>
    <Item id="syncphonesettings" title_type="title" setting_type="checkbox">
                    <Title>@Sync_Phone_Settings</Title>
                    <CheckBox checked="yes"/>
    </Item>
    <Item id="unit" title_type="title_subtitle" setting_type="radiobox"
                                                            conditional="enable">
                    <Title>@Unit</Title>
                    <SubTitle options="radiobox">@Unit_Item1</SubTitle>
                    <Radiobox id="unit" number="2" selected="1">
                              <RadioboxItem>@Unit_Item1</RadioboxItem>
                              <RadioboxItem>@Unit_Item2</RadioboxItem>
                    </Radiobox>
                    <Condition type="enable">syncphonesettings</Condition>
    </Item>
    <Item id="audiorefresh" title_type="title_subtitle" setting_type="radiobox"
conditional="enable">
                <Title>@Audio_Refresh</Title>
                <SubTitle options="selected">@Audio_Refresh_Item1</SubTitle>
                <Radiobox id="autorefresh" number="6" selected="4">
                        <RadioboxItem>@Audio_Refresh_Item1</RadioboxItem>
                        <RadioboxItem>@Audio_Refresh_Item2</RadioboxItem>
                        <RadioboxItem>@Audio_Refresh_Item2</RadioboxItem>
                        <RadioboxItem>@Audio_Refresh_Item4</RadioboxItem>
                        <RadioboxItem>@Audio_Refresh_Item5</RadioboxItem>
                        <RadioboxItem>@Audio_Refresh_Item6</RadioboxItem>
                <Radiobox>
                <Condition type="enable">syncphonesettings</Condition>
    </Item>
    <Item id="schduletype" title_type="title_subtitle" setting_type="radiobox">
             <Title>@Selecte_event_to_show</Title>
            <SubTitle>@Next_event</SubTitle>
             <Radiobox id="unit" number="2" selected="1" conditional_value=”1”>
                    <RadioboxItem>@Next_event</RadioboxItem>
                    <RadioboxItem>@Today_events</RadioboxItem>
             </Radiobox>
     </Item>
     <Item id="displaytime" title_type="title_subtitle" setting_type="radiobox" 
conditional="enable">
               <Title>@Show_for</Title>
               <SubTitle>@3seconds</SubTitle>
               <Radiobox id="unit" number="2" selected="1">
               <RadioboxItem>@3seconds</RadioboxItem>
               <RadioboxItem>@5seconds</RadioboxItem>
               <RadioboxItem>@10seconds</RadioboxItem>
                </Radiobox>
                <Condition type="enable">schduletype</Condition>
     </Item>
     <Item id="refresh" title_type="title_subtitle" setting_type="checkbox">
                    <Title>@Refresh_With_Power_Button</Title>
                    <SubTitle>@Refresh_With_Power_Button_DESC</SubTitle>
                    <CheckBox checked="yes"/>
      </Item>
      <Item id="restart" title_type="title_subtitle" setting_type="switch">
                    <Title>@Restart_switch</Title>
                    <SwitchButtonswitch_checked="yes"/>
      </Item>
      <Item id="change" title_type="title_subtitle" setting_type="switch">
                    <Title>@change_switch</Title>
                    <SubTitle options="switch">@switch_off</SubTitle>
                    <SwitchButton id="change"switch_checked="no">
                            <SwitchItem>@switch_on</SwitchItem>
                            <SwitchItem>@switch_off</SwitchItem>
      </Item>
      <Item id="memo" title_type="title_subtitle" setting_type="inputbox">
                        <Title>@memo_input</Title>
                        <SubTitle>@Input_DESC</SubTitle>
      </Item>
      <Item id="set_date" title_type="title_subtitle" setting_type="datepicker">
                <Title>@set_date</Title>
                <SubTitle>@not_setdate</SubTitle>
                <DatePicker id=”set_date” year=”2013” month=”1” day=”1”>
              </DatePicker>
      </Item>
      <Item id="set_date" title_type="title_subtitle" setting_type="datepicker">
                     <Title>@set_time</Title>
                     <SubTitle>@not_settime</SubTitle>
                     <TimePicker id=”set_time” hour=”1” minute=”12”>
                    </ TimePicker >
     </Item>
     <Item id="favorite" title_type="title" setting_type="favorites">
                        <Title>@favorites_test</Title>
                        <Favorites id="favorite" number="3" selected="2">
                            <FavoritesItem>
                                  <DisplayName>@widget_1</DisplayName>
                            <ClassName>com.samsung.android.app.widget1</ClassName>
                                  <IsAppWidget>false</IsAppWidget>
                            </FavoritesItem>
                            <FavoritesItem>
                                  <DisplayName>@widget_2</DisplayName>
                        <ClassName>com.samsung.android.app.widget2</ClassName>
                                  <IsAppWidget>false</IsAppWidget>
                            </FavoritesItem>
                            <FavoritesItem>
                                  <DisplayName>@widget_3</DisplayName>
                        <ClassName>com.samsung.android.app.widget3</ClassName>
                                  <IsAppWidget>true</IsAppWidget>
                             </FavoritesItem>
                        </Favorites>
     </Item>
     <Item id="linksettig" title_type="title" setting_type="link">
               <Title>@weather_setting</Title>
               <PackageName>com.android.testclock</PackageName>
               <PackageIntentAction>com.android.testclock1.setting</PackageIntentAction>
               <Extra>
                   <Key>SETTING_MODE</Key>
                   <Value>3</Value>
              </Extra>
     </Item>
     <Item id="transfernow" title_type="title" setting_type="button">
              <Title>@transfernow</Title>
              <ON-CLICK>
    <PhoneSideIntentAction>intent.action.sacmera.transfernow</PhoneSideIntentAction>
                 <PhoneSideExtra>
                      <Key>SETTING_MODE</Key>
                      <Value>0</Value>
                 </PhoneSideExtra>
    <GearSideIntentAction>intent.action.sacmera.transfernow</GearSideIntentAction>
                 <GearSideExtra>
                      <Key>SETTING_MODE</Key>
                      <Value>0</Value>
                 </GearSideExtra>
              </ON-CLICK>
      </Item>
    </Settings>
</Application>
```

When the string IDs are resolved, the XML settings file looks as follows:

Sample Code

```
<?xml version="1.0" encoding="utf-8"?>
<Applicationxmlns:android="http://schemas.android.com/apk/res/android" type="clock"
                                     version="001"language="english">
        <packageName>samsung.com.watch.weatherclock</packageName>
        <DisplayName>Weather Clock</DisplayName>
        <Settings>
             <Item id="24format" title_type="title_subtitle"
                                                     setting_type="checkbox">
                <Title>Use 24-hour format</Title>
                <SubTitle options="checkbox">13:00</SubTitle>
                <CheckBox id="24format" checked="yes">
                        <CheckBoxItem>1:00 PM</CheckBoxItem>
                        <CheckBoxItem>13:00</CheckBoxItem>
                </CheckBox>
             </Item>
           <Item id="showdate" title_type="title" setting_type="checkbox">
                <Title>Show date</Title>
                <CheckBox checked="yes"/>
           </Item>
           <Item id="weathertype" title_type="title_subtitle"
                                                             setting_type="radiobox">
                <Title>Weather Type</Title>
                <SubTitle>Current</SubTitle>
                <Radiobox id="weathertype" number="2" selected="1">
                        <RadioboxItem>Current</Radio_Item>
                        <RadioboxItem>3-day forecast</Radio_Item>
                </Radiobox>
            <Item id="syncphonesettings" title_type="title" setting_type="checkbox">
                <Title>Sync phone's settings</Title>
                <CheckBox checked="yes"/>
            </Item>
            <Item id="unit" title_type="title_subtitle" setting_type="radiobox"
                                                             conditional="enable">
                <Title>Unit</Title>
                <SubTitle options="radiobox">℃</SubTitle>
                <Radiobox id="unit" number="2" selected="1">
                        <RadioboxItem>℃</RadioboxItem>
                        <RadioboxItem>℉</RadioboxItem>
                </Radiobox>
                <Condition type="enable">syncphonesettings</Condition>
            </Item>
            <Item id="audiorefresh" title_type="title_subtitle"
                                     setting_type="radiobox" conditional="enable">
                <Title>Audio refresh</Title>
                <SubTitle options="selected">Every 6 hours</SubTitle>
                <Radiobox id="autorefresh" number="6" selected="4">
                    <RadioboxItem>None</RadioboxItem>
                    <RadioboxItem>Every 1 hour</RadioboxItem>
```

```
                       <RadioboxItem>Every 3 hour</RadioboxItem>
                       <RadioboxItem>Every 6 hour</RadioboxItem>
                       <RadioboxItem>Every 12 hour</RadioboxItem>
                       <RadioboxItem>Every 24 hour</RadioboxItem>
                    </Radiobox>
                    <Condition type="enable">syncphonesettings</Condition>
               </Item>
             <Item id="schduletype" title_type="title_subtitle" setting_type="radiobox">
                     <Title>Select events to show</Title>
                    <SubTitle>Next event</SubTitle>
                     <Radiobox id="unit" number="2" selected="1" conditional_value=”1”>
                         <RadioboxItem>Next Event</RadioboxItem>
                         <RadioboxItem>Todays’s events</RadioboxItem>
                     </Radiobox>
             </Item>
             <Item id="displaytime" title_type="title_subtitle" setting_type="radiobox"
conditional="enable">
                     <Title>Show_for</Title>
                   <SubTitle>3 seconds</SubTitle>
                     <Radiobox id="unit" number="2" selected="1">
                         <RadioboxItem>3 seconds</RadioboxItem>
                         <RadioboxItem>5 seconds</RadioboxItem>
                           <RadioboxItem>10 seconds</RadioboxItem>
                      </Radiobox>
                      <Condition type="enable">schduletype</Condition>
              </Item>
                <Item id="refresh" title_type="title_subtitle" setting_type="checkbox">
                        <Title>Refresh with power button</Title>
                        <SubTitle>Update weather information automatically when
                        you press the power button to show this clock</SubTitle>
                        <CheckBox checked="yes"/>
                </Item>
                <Item id="restart" title_type="title" setting_type="switch">
                        <Title>Restart</Title>
                        <SwitchButtonswitch_checked="yes"/>
                </Item>
                <Item id="change" title_type="title_subtitle" setting_type="switch">
                        <Title>Change</Title>
                        <SubTitle options="switch">OFF</SubTitle>
                        <SwitchButtonid="change"switch_checked="no">
                                <SwitchItem>ON</SwitchItem>
                                <SwitchItem>OFF</SwitchItem>
                </Item>
                <Item id="memo" title_type="title_subtitle" setting_type="inputbox">
                        <Title>Memo</Title>
                        <SubTitle>Please input memo</SubTitle>
                </Item>
                <Item id="set_date" title_type="title_subtitle" setting_type="datepicker">
                        <TitleSet Date</Title>
                        <SubTitle>Not Set</SubTitle>
                        <DatePicker id=”set_date” year=”2013” month=”1”
```

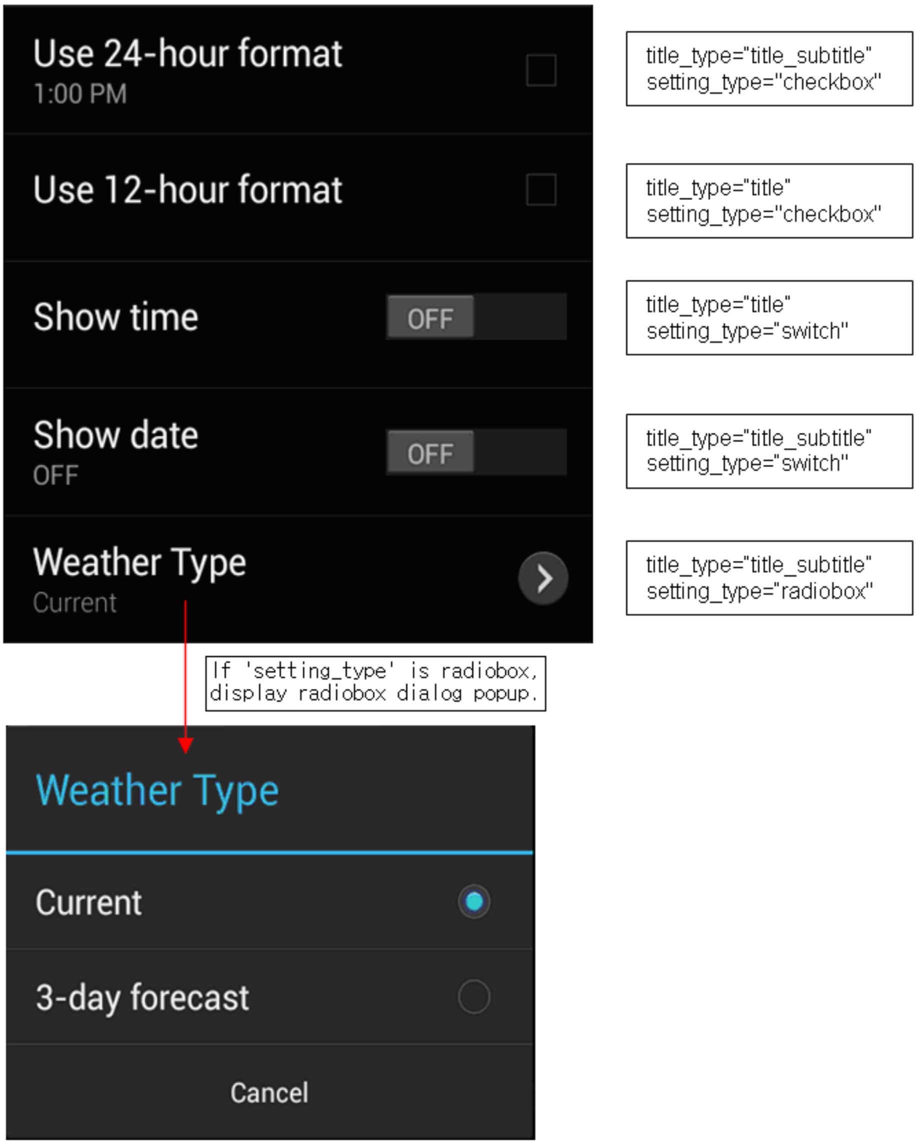
```
day=”1”></DatePicker>
                </Item>
                <Item id="set_time" title_type="title_subtitle" setting_type="timepicker">
                        <Title>Set Time</Title>
                        <SubTitle>Not Set</SubTitle>
                        <TimePicker id=”set_time” hour=”1” minute=”12”></TimePicker>
                </Item>
                <Item id="favorite" title_type="title" setting_type="favorites">
                     <Title>Favorite Test</Title>
                     <Favorites id="favorite" number="3" selected="2">
                          <FavoritesItem>
                               <DisplayName>Widget_1</DisplayName>
                               <ClassName>com.samsung.android.app.widget1</ClassName>
                            <IsAppWidget>false</IsAppWidget>
                          </FavoritesItem>
                          <FavoritesItem>
                               <DisplayName>Widget_2</DisplayName>
                               <ClassName>com.samsung.android.app.widget2</ClassName>
                            <IsAppWidget>false</IsAppWidget>
                          </FavoritesItem>
                          <FavoritesItem>
                                <DisplayName>Widget_3</DisplayName>
                                <ClassName>com.samsung.android.app.widget3</ClassName>
                            <IsAppWidget>true</IsAppWidget>
                          </FavoritesItem>
                        </Favorites>
           </Item>
           <Item id="linksettig" title_type="title" setting_type="link">
              <Title>Weather Setting</Title>
              <PackageName>com.android.testclock</PackageName>
              <PackageIntentAction>com.android.testclock1.setting</PackageIntentAction>
              <Extra>
                  <Key>SETTING_MODE</Key>
                  <Value>3</Value>
              </Extra>
           </Item>
           <Item id="transfernow" title_type="title" setting_type="button">
              <Title>Transfer now</Title>
              <ON-CLICK>

<PhoneSideIntentAction>intent.action.sacmera.transfernow</PhoneSideIntentAction>
                 <PhoneSideExtra>
                      <Key>SETTING_MODE</Key>
                      <Value>0</Value>
                 </PhoneSideExtra>

<GearSideIntentAction>intent.action.sacmera.transfernow</GearSideIntentAction>
                 <GearSideExtra>
                       <Key>SETTING_MODE</Key>
                       <Value>0</Value>
                 </GearSideExtra>
              </ON-CLICK>
           </Item>
       </Settings>
    </Application>
```

The settings.xml file for the application is structured as follows:

Sample Code

```
    <?xml version="1.0" encoding="UTF-8"?>
    <Application xmlns:android="http://schemas.android.com/apk/res/android"
type="application" version="001" language="english">
        <packageName>languages0</packageName>
        <DisplayName>@setting</DisplayName>
        <Settings>
                <Item id="radiobox" title_type="title_subtitle" setting_type="radiobox">
                        <Title>@radiobox</Title>
                        <SubTitle options="radiobox">White</SubTitle>
                        <Radiobox id="radiobox" number="5" selected="1">
                                <RadioboxItem>White</RadioboxItem>
                                <RadioboxItem>Red</RadioboxItem>
                                <RadioboxItem>Blue</RadioboxItem>
                                <RadioboxItem>Green</RadioboxItem>
                                <RadioboxItem>Yellow</RadioboxItem>
                        </Radiobox>
                </Item>
                <Item id="timepicker" title_type="title_subtitle"
setting_type="timepicker">
                        <Title>@timepicker</Title>
                        <SubTitle>24-hour format</SubTitle>
                        <TimePicker id="timepicker" hour="10" minute="26"></TimePicker>
                </Item>
                <Item id="switch" title_type="title_subtitle" setting_type="switch">
                        <Title>@switch</Title>
                        <SubTitle options="switch">ON</SubTitle>
                        <SwitchButton id="switch" switch_checked="yes">
                                <SwitchItem>ON</SwitchItem>
                                <SwitchItem>OFF</SwitchItem>
                        </SwitchButton>
                </Item>
                <Item id="checkbox" title_type="title_subtitle" setting_type="checkbox">
                        <Title>@checkbox</Title>
                        <SubTitle options="checkbox">OFF</SubTitle>
                        <CheckBox id="checkbox" checked="no">
                                <CheckBoxItem>ON</CheckBoxItem>
                                <CheckBoxItem>OFF</CheckBoxItem>
                        </CheckBox>
                </Item>
                <Item id="datepicker" title_type="title" setting_type="datepicker">
                        <Title>@datepicker</Title>
                        <DatePicker id="datepicker" year="2013" month="12"
day="25"></DatePicker>
                </Item>
                <Item id="inputbox" title_type="title_subtitle" setting_type="inputbox">
                        <Title>@inputbox</Title>
                        <SubTitle>InputBox</SubTitle>
                </Item>
        </Settings>
    </Application>
```

The following figure shows the Settings UI displayed on screen for the sample XML settings file used earlier in this document.



Use 24-hour format
1:00 PM □

title_type="title_subtitle"
setting_type="checkbox"

Use 12-hour format □

title_type="title"
setting_type="checkbox"

Show time OFF

title_type="title"
setting_type="switch"

Show date
OFF OFF

title_type="title_subtitle"
setting_type="switch"

Weather Type
Current

title_type="title_subtitle"
setting_type="radiobox"

If 'setting_type' is radiobox,
display radiobox dialog popup.

Weather Type

Current ●

3-day forecast ○

Cancel

Figure 9: Setting UI

4.1.2 Configuration File for Complex Settings UI

You can use the Link Mode to have Gear Manager call your own settings page if your Wearable-side widget requires a complex settings UI.

Use the following guidelines for creating the XML settings file for your widget when you are using Link Mode:

- DisplayNameTag is a required field
- You must configure your settings directly in your application by sending Intents. For more information, see the example below.

Sample Code

```
<?xml version="1.0" encoding="utf-8"?>
<Applicationxmlns:android="http://schemas.android.com/apk/res/android"
                                type="clock" version="001"language="english">
        <packageName>samsung.com.watch.weatherclock</packageName>
        <DisplayName>@Clock_Name</DisplayName>
        <MotherPackageName>com.sec.android.app.clockpackage</MotherPackageName>
        <MotherPackageIntentAction>com.sec.android.test.status.setting</MotherPackageIntent
Action>
    <!-- You must use one of both MotherPackageIntentAction and MotherActivity
    <MotherActivity> com.sec.android.test.status.MenuSettingsActivity<MotherActivity>
    -->
        <MotherExtra>
                <Key>Name</Key>
                <Value>WeatherClock</Value>
        </MotherExtra>
</Application>
```

The resulting settings.xml file for providing a link to your application's settings is as follows:

Sample Code

```
<?xml version="1.0" encoding="utf-8"?>
<Applicationxmlns:android="http://schemas.android.com/apk/res/android"type=""
                        version=""language="">
                <packageName></packageName>
                <DisplayName></DisplayName>
             <MotherPackageName></MotherPackageName>
                <MotherPackageIntentAction></MotherPackageIntentAction>
                <!—You must use one of MotherPackageIntentAction and MotherActivity
                <MotherActivity></MotherActivity>
                -->
                <MotherExtra>
                        <Key></Key>
                        <Value></Value>
                </MotherExtra>
</Application>
```

Your widget must provide the implementation for configuring the settings for Gear Manager to activate it.

Note
If you need to have Gear Manager call another package's activity, declare it in your host-side application's Android manifest file.

For example,

```
Packagename: com.samsung.android.test
PackageNameIntentAction: com.samsung.android.test.status.setting
Setting Activity: com.samsung.android.test.status.MenuSettingsActivity
```

Sample Code : Androidmanifest.xml

```
<!-- If using MotherPackageIntentAction Tag-->
<activity android:name="com.sec.android.test.status.MenuSettingsActivity" >
        <intent-filter>
                <action android:name="com.sec.android.test.status.setting" />
                <category android:name="android.intent.category.DEFAULT" />
        </intent-filter>
    </activity>

    <!-- If using MotherActivity Tag-->
    <activity android:name="com.sec.android.test.status.MenuSettingActivity" >
        <intent-filter>
        <action android:name="com.sec.android.test.status.MenuSettingActivity"/>
        <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
</activity>
```

In the sample code below, MotherPackageName, Activity, IntentAction, Key and Value are resolved for the MotherPackageName, MotherActivity, MotherPackageIntentAction, and MotherExtraelements.

Sample Code

```
<!-- If using MotherPackageIntentAction Tag-->
    Intent intent = new Intent(MotherPackageIntentAction);
    intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    intent.putExtra(Key, Value);
    intent.puExtra("FROM_CLOCKS", true /*if selecting the setting icon on Clock list, then
set true */);
    startActivity(intent);
```

```
<!-- If using MotherActivity Tag-->
    Intent intent = new Intent();
    intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    intent.setComponent(new ComponentName(MotherPackageName, MotherActivity));
    intent.putExtra(Key, Value);
    intent.puExtra("FROM_CLOCKS", true /*if selecting the setting icon on Clock list, then
set true */);
    startActivity(intent);
```

4.1.3 Backup and Restore settings of link mode

Because the Link Mode settings page belongs to another package, Gear Manager does not backup and restore the data. However, Gear Manager can indicate when data backup or restoration is required by broadcasting an intent when a backup or restore is performed on the host side. The other package can implement backup and restore features using this intent information.

Gear Manager broadcast intents.

Sample Code

```
Backup :
<!-- when user press the backup button settings in Gear Manager -->
<!-- when Gear and Gear Manager are disconnected -->
sendBroadcast(new Intent("android.intent.watchmanager.action.BACKUP"));

Restore :
<!-- when user press the restore settings button in Gear Manager -->
<!-- when user press the restore setup wizard button in Gear Manager -->
sendBroadcast(new Intent("android.intent.watchmanager.action.RESTORE"));
```

Another package can receive the intent by declaring an intent-filter in the receiver element in the host side `AndroidManifest.xml`.

Sample Code

```
<receiver android:name="com.samsung.android.test.status.receiver" >
    <intent-filter>
    <!-- when user press the backup settings button in Gear Manager -->
    <!-- when Gear and Gear Manager are disconnected -->
    <action android:name="android.intent.watchmanager.action.BACKUP" />
    <!-- when user press the restore settings button in Gear Manager -->
    <!-- when user press the restore setup wizard buttion in Gear Manager -->
    <action android:name="android.intent.watchmanager.action.RESTORE" />
    </intent-filter>
</receiver>
```

The example below demonstrates how to implement a receiver for receiving broadcast intents.

Sample Code

```
@Override
public void onReceive(Context context, Intent intent) {

        final String action = intent.getAction();

        if (action.equals("android.intent.watchmanager.action.BACKUP")) {
                // We can back up data here
                // ex) save as shared preference, save as files
        } else if (action.equals("android.intent.watchmanager.action.RESTORE")) {
                // We can restore data here
                // ex) save as shared preference, save as files
        }
}
```

4.2 Structure of the Configuration File (“PackageName_setttings.xml”)

4.2.1 <Application>

Syntax	<Application xmlns:android="http://schemas.android.com/apk/res/android" type="string resource" version=" string resource " language="string resource"> … </Application>	
Contained in	none	
Must Contain	none	
Can Contain	none	
Description	Application information	
Attribute	**type**	Application Type. “clock” and “application” are allowed.
	version	Application version
	language	Device language

4.2.2 <AppSetting> or <ClockSetting>

Syntax	<AppSetting> … </AppSetting> <ClockSetting> … </ClockSetting>
Contained in	<Application>
Description	To use the App setting UI, use the <AppSetting> Tag To use the Clock setting UI, use the <ClockSetting> Tag
Attributes	none

4.2.3 <packageName>

Syntax	<packageName></packageName>
Contained in	<Application>
Description	Package name
Attributes	None

4.2.4 <DisplayName>

Syntax	<DisplayName></DisplayName>
Contained in	<Application>
Description	Application display name
Attributes	None

4.2.5 <MotherPackageName>

Syntax	<MotherPackageName></MotherPackageName>
Contained in	<Application>
Description	The PackageName of the app to be called if Gear Manager calls another package's activity. Can be used, for example, to check whether the application declared is installed, otherwise configure the UI using the basic settings UI. If settings configuration requires a complex UI and cannot be reported to

	your application in XML format, then you can use Link Mode to have Gear Manager call your own settings page. **Note:** Omit <MotherPackageName>, <MotherPackageIntentAction>, <MotherExtra>, <key>, <value> tags if you only use the basic settings UI.
Attributes	None

4.2.6 <MotherPackageIntentAction>

Syntax	<MotherPackageIntentAction></MotherPackageIntentAction>
Contained in	<Application>
Description	The PackageName of the app to be called if Gear Manager calls another package's activity. Can be used, for example, to check whether the application declared is installed, otherwise configure the UI using the basic settings UI. If settings configuration requires a complex UI and cannot be reported to your application in XML format, then you can use Link Mode to have Gear Manager call your own settings page.
Attributes	None

4.2.7 <MotherActivity>

Syntax	<MotherActivity></ MotherActivity >
Contained in	<Application>
Description	The PackageName of the app to be called if Gear Manager calls another package's activity. Can be used, for example, to check whether the application declared is installed, otherwise configure the UI using the basic settings UI. If configuring the settings requires a complex UI and cannot be reported to your application in XML format, then you can use Link Mode to have Gear Manager call your own settings page.
Attributes	None

4.2.8 <MotherExtra>

Syntax	<MotherExtra></MotherExtra>
Contained in	<Application>
Description	Additional information for Intent for Activity and PackageName
Attributes	None

4.2.9 <key>

Syntax	<key></key>
Contained in	<Application>
Description	Key name for the Intent for Activity and PackageName. intent.putExtra("Version", "001");
Attributes	None

4.2.10 <Value>

Syntax	<value></value>
Contained in	<MotherExtra>
Description	Key value for the Intent for Activity and PackageName. intent.putExtra("Version", "001");
Attributes	None

4.2.11 <Settings>

Syntax	<Settings> … </Settings>
Contained in	<Application>
Description	Values for the settings.

Attributes	None

4.2.12 <MotherCondition>

Syntax	<MotherCondition id="string resource" packageName="string resource" minVersion="string resource”> … </MotherCondition >	
Contained in	<Settings>	
Must Contain	<onTrue> <onFalse>	
Description	For certain conditions (packageName, minVersion), invoke a different settings screen. Only supported for the XML method.	
Attribute	**id**	Must be ‘RequiredMother’
	packageName	Condition-1 Set the packageName. If the APK with the ‘packageName’ is installed on the device
	minVersion	Condition-2 Set the minimum version of APK with the ‘packageName’.

4.2.13 <onTrue>

Syntax	<onTrue> … </onTrue>
Contained in	<MotherCondition>
Must Contain	<Item>
Description	When MotherCondition is true, get the information for the item.
Attributes	None

4.2.14 <onFalse>

Syntax	<onFalse> … </onFalse>
Contained in	<MotherCondition>
Must Contain	<Item>

Description	When MotherCondition is false, get the information for the item.
	If the <Item> setting_type is "noitem" and the meta-data includes "pre_check_settings_condition", the Setting icon for this app is hidden in the Gear Manager display.
	`<onFalse>`
	`    <Item id="" title_type="" setting_type="noitem">`
	`    </Item>`
	`</onFalse>`
	Wearable-side app's meta-data information:
	`<application`
	`    <activity>`
	`        …`
	`    </activity>`
	`    <meta-data android:name="pre_check_settings_condition" android:value="1"/>`
	`</application>`

Attributes	None

4.2.15 <Item>

Syntax	<Item id="string resource" title_type="string resource" setting_type="string resource" conditional=”string resource”> … </Item>
Contained in	<Settings>
Must Contain	<Title>
Can Contain	<SubTitle> <CheckBox> <Radiobox> <Condition> <SwitchButton> <TimePicker>

	<DataPicker> <Favorites> <PackageName> <PackageIntentAction> <Extra> <Key> <Value> <ON-CLICK> <PhoneSideIntentAction> <PhoneSideExtra> <GearSideIntentAction> <GearSideExtra>	
Description	Information for each item.	
	id	Id of each Item.
	title_type	Title type “title” – only Title “title_subtitle” – Title and Subtitle
	setting_type	Setting type “checkbox” – CheckBox “radiobox” – RadioBox “switch”-Switchbutton “timepicker”-TimePicker “datepicker”-DatePicker” “favorites”-Favorites “inputbox” – Input box dialog display “link” – Open the activity of the specific app. “button” – Sendbraodcast for specific intent **Note:** If “setting_type” is “inputbox”, “timepicker” or “datepicker”, title_type must be “title_subtitle”.
	Conditional	Display the Condition. “enable” –Display the feature when the value of the Condition Tag is Enable.

4.2.16 <Title>

Syntax	<Title></Title>
Contained in	<Item>
Description	String ID of the title content
Attributes	none

4.2.17 <SubTitle>

Syntax	<SubTitle></ SubTitle>	
Contained in	< Item>	
Description	String ID of subtitle content.	
Attributes	**options**	Radiobox - shows the value that you set in the RadioBox in Sub Title. checkbox - shows the value that you set in the CheckBox in Sub Title. inputbox - shows the value that you set in the Inputbox dialog in Sub Title. timepicker - shows the value that you set in the TimePicker dialog in Sub Title. Datepicker – shows the value that you set in the DatePicker dialog in Sub Title.

4.2.18 <CheckBox>

Syntax	<CheckBox id="string resource" checked="string resource"> … </ CheckBox>	
Contained in	< Item>	
Description	**id**	Id of each Item
	checked	Initial value (yes/no)

4.2.19 <CheckboxItem>

Syntax	<CheckboxItem></ CheckboxItem>
Contained in	<Checkbox>
Description	The form 'Title + SubTitle +Checkbox' is used. This is used only if the 'SubTitle' element is changed using the status of the Checkbox element's 'checked' attribute. Set the SubTitle string ID for the contents to be displayed to a CheckBoxItem.
Attributes	none
Example	SubTitle has checkbox set as the option. The string ID for 'SubTitle' matches the string ID for 'CheckBoxItem' • 'checked' attribute is "yes": @Hour_Format_Item2 • 'checked' attribute is "no": @Hour_Format_Item1

Sample Code

```
<Item id="24format" title_type="title_subtitle" setting_type="checkbox">
    <Title>@24_Hour_Format</Title>
    <SubTitle options="checkbox">@Hour_Format_Item2</SubTitle>
    <CheckBox id="24format" checked="yes">
    <CheckBoxItem>@Hour_Format_Item1</CheckBoxItem>
    <CheckBoxItem>@Hour_Format_Item2</CheckBoxItem>
    </CheckBox>
</Item>
```

4.2.20 <Radiobox>

Syntax	<Radiobox id="string resource" number="string resource" selected="numeric resource"> … </ Radiobox>	
Contained in	< Item>	
Must Contain	<RadioboxItem>	
Description	Radio button.	
Attributes	**Id**	Id of each radiobox Item

	Number	Sets the number of RadioBoxes
	Selected	Sets the initial value
	Conditional_value	Used to enable/disable the specific <Item> This attribute should be used with the <Condition> Tag Set the value to a <RadioboxItem>

4.2.21 <RadioboxItem>

Syntax	<RadioboxItem></RadioboxItem>
Contained in	<Radiobox>
Description	String IDs of Radiobox items
Attributes	none

4.2.22 <Condition>

Syntax	<Condition type="string resource"></Condition>	
Contained in	<Item>	
Description	When the specific condition is met, set whether to display.	
Attribute	**type**	“enable”

4.2.23 <SwitchButton>

Syntax	<SwitchButton id="string resource" switch_checked="string resource"> … </SwitchButton>	
Contained in	<Item>	
Description	Switch Button	
	id	Id of each Item.
	switch_checked	Initial value that is checked

4.2.24 <SwitchItem>

Syntax	<SwitchItem></SwitchItem>
Contained in	<SwitchButton>
Description	The form 'Title + SubTitle +SwitchButton' is used. This is used only if SubTitle element is changed using the status of the SwitchButton element's 'switch_checked' attribute. Set the SubTitle string ID for the contents to be displayed to a SwitchItem.
Attributes	none
Example	SubTitle has switch set as the option. The string ID for SubTitle matches the string ID for SwitchItem. • 'switch_checked' attribute is "yes": @Hour_Format_Item1 • 'switch_checked' attribute is "no": @Hour_Format_Item2

Sample Code

```
<Item id="24format" title_type="title_subtitle" setting_type="switch">
    <Title>@24_Hour_Format</Title>
    <SubTitle options="switch">@Hour_Format_Item2</SubTitle>
    <SwitchButton id="24format" switch_checked="no">
    <SwitchItem>@Hour_Format_Item1</SwitchItem>
    <SwitchItem>@Hour_Format_Item2</SwitchItem>
    </CheckBox>
</Item>
```

4.2.25 <TimePicker>

Syntax	<TimePicker id="string resource" hour="string resource" minute="string resource"> … </TimePicker>	
Contained in	<Item>	
Description	Used when setting up a TimePickerDialog	
Attribute	**id**	Id of each Item.
	hour	Initial value for hour.

	minute	Initial value for minute

4.2.26 <DatePicker>

Syntax	<DatePicker id="string resource" year="string resource" month="string resource" day="string resource"> … </DatePicker>	
Contained in	<Item>	
Description	Used when setting up a DatePickerDialog.	
Attributes	**id**	Id of each Item.
	year	Initial value for year.
	month	Initial value for month.
	day	Initial value for day.

4.2.27 <Favorites>

Syntax	<Favorites id="string resource" number="string resource" selected="string resource" > … </Favorites>	
Contained in	<Item>	
Must Contain	<FavoritesItem>	
Description	Used when setting up Favorites. You must organize the List of Favorites using this Tag in order to support “Multi-Widgets”. This tag must not have a Sub Title element.	
Attribute	**id**	Id of each Item.
	number	Total number of FavoritesItem (that supports Mulit-Widgets).
	selected	Initial the selected favoritsItem

4.2.28 <FavoritesItem>

Syntax	<FavoritesItem> … </FavoritesItem>
Contained in	<Favroites>
Must Contain	<DisplayName> <ClassName>
Description	The number of FavoritesItem must be same as the ‘number’ attribute of the <Favorites> tag
Attribute	none

4.2.29 <DisplayName>

Syntax	<DisplayName> … </DisplayName>
Contained in	<FavoritesItem>
Description	Set “StringID”(Widget/Launcher) to be displayed on Favorite_List
Attribute	none

4.2.30 <ClassName>

Syntax	<ClassName> … </Class>
Contained in	<FavoritesItem>
Description	Set the “ClassName of a Widget (or Launcher) supporting Multi-Widget.
Attribute	none

4.2.31 <IsAppWidget>

Syntax	<IsAppWidget> …

	</IsAppWidget>
Contained in	<FavoritesItem>
Description	Whether Widget(or Launcher) is a widget or not
Attribute	none

4.2.32 <PackageName>

Syntax	<PackageName></PackageName>
Contained in	<Item>
Description	The PackageName of the app to be called if Gear Manager calls another package's activity in XML format. Check whether the application declared is installed. If installed, call startActvity with PackageIntentAction.
Attribute	none

4.2.33 <PackageIntentAction>

Syntax	<PackageIntentAction></PackageIntentAction>
Contained in	<Item>

Description	Clarify the intent to be operated in conjunction with <PackageName>.
	It calls another package's activity as show in the example below.
	`Intent intent = new Intent(PackageIntentAction);`
	`Intent.putExtra("Key", key);`
	`Intent.putExtra("Value", value);`
	`mContext.startActivity(intent);`

Attribute	none

4.2.34 <Extra>

Syntax	<PackageIntentAction></PackageIntentAction>
Contained in	<Item>

Must Contain	<Key> <Value>
Description	Additional information for Intent for Activity and PackageName
Attribute	none

4.2.35 <key>

Syntax	<key></key>
Contained in	<Extra> <PhoneSideExtra> <GearSideExtra>
Description	Key name for the Intent for Activity and PackageName. intent.putExtra("Version", "001");
Attribute	none

4.2.36 <Value>

Syntax	<value></value>
Contained in	<Extra> <PhoneSideExtra> <GearSideExtra>
Description	Key value for the Intent for Activity and PackageName. intent.putExtra("Version", "001");
Attribute	none

4.2.37 <ON-CLICK>

Syntax	<ON-CLICK> … </ON-CLICK>
Contained in	<Item>
	<PhoneSideIntentAction> <GearSideIntentAction>
Description	To send a broadcast to either the host devices side or to the wearable device

	side.
Attribute	none

4.2.38 <PhoneSideIntentAction>

Syntax	<PhoneSideIntentAction></PhoneSideIntentAction>
Contained in	<ON-CLICK>

Description	Clarify the intent to send broadcast, as shown in the example below.
	`Intent intent = new Intent(PhoneSideIntentAction);`
	`Intent.putExtra(“Key”, key);    // refer to 4.2.34`
	`Intent.putExtra(“Value”, value);    // refer to 4.2.35`
	`sendbroadcast(intent)`

Attribute	none

4.2.39 <PhoneSideExtra>

Syntax	<PhoneSideExtra></PhoneSideExtra>
Contained in	<ON-CLICK>
Must Contain	<Key> <Value> Refer to 4.2.34 , 4.2.35
Description	Additional information for Intent to send broadcast.
Attribute	none

4.2.40 <GearSideIntentAction>

Syntax	<GearSideIntentAction></GearSideIntentAction>
Contained in	<ON-CLICK>

Description	**Description**
	Clarify the intent to send broadcast, as shown in the example below.
	`Intent intent = new Intent(GearSideIntentAction);`
	`Intent.putExtra(“Key”, key);    // refer to 4.2.34`
	`Intent.putExtra(“Value”, value);    // refer to 4.2.35`
	`sendbroadcast(intent)`

Attribute	none

4.2.41 <GearSideExtra>

Syntax	<GearSideExtra></GearSideExtra>
Contained in	<ON-CLICK>
Must Contain	<Key> <Value> - Refer to 4.2.34 , 4.2.35
Description	Additional information for Intent to send broadcast.
Attribute	none

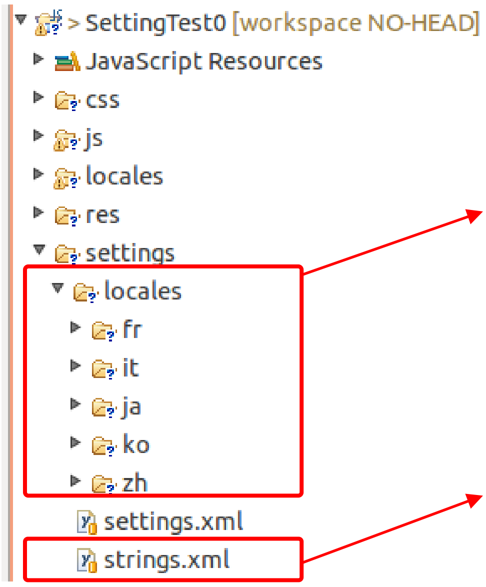
4.3 Multi-language Support

Gear Manager sends notifications about language changes on the host device to Gear. WearableManagerService receives these notifications and sends them to all the installed wearable-side widgets and prepares new `settings.xml` files for that language.

If your application has a corresponding language folder under settings/locales as in the example below and if there is a `strings.xml` with string tag values written in the required language, then Gear Manager displays the changed settings menu on the host device.

If there is no matching folder in locales, then the default `strings.xml` is used

If there is no matching folder in locales, then strings.xml in default strings.xml.



Add supported languages in your

Default string file will be used if there is no matching language support in locales folder.

Figure 10: Project composition

For more details of the multiple language support in Settings, refer to "Gear Wearable Device Application Programming – Getting Started" in the help contents of the Tizen SDK for Wearable.

5. Delivering Application Setting Results

5.1 Settings from Basic Settings UI

If you use a basic settings UI, then Gear Manger returns the results as follows:

Sample Code

```
<?xml version="1.0" encoding="utf-8"?>
<Applicationxmlns:android="http://schemas.android.com/apk/res/android" type="clock"
version="001" language="english">
    <packageName>com.sec.clock.weatherclock</packageName>
        <SettingsResult>
                <Item id="24format" title_type="title_subtitle" setting_type="checkbox">
                        <CheckBox id="24format" checked="yes"/>
                </Item>
                <Item id="showdate" title_type="title" setting_type="checkbox">
                        <CheckBox id="showdate" checked="yes"/>
                </Item>
                <Item id="weathertype" title_type="title_subtitle" setting_type="radiobox">
                        <Radiobox id="weathertype" number="2" selected="1">
                <Item id="unit" title_type="title_subtitle" setting_type="radiobox"
conditional="enable">
                        <Radiobox id="unit" number="2" selected="1">
                </Item>
                <Item id="audiorefresh" title_type="title_subtitle" setting_type="radiobox"
conditional="enable">
                        <Radiobox id="autorefresh" number="6" selected="1">
                </Item>
                <Item id="scheduletype" title_type="title_subtitle"
setting_type="radiobox">
                        <Radiobox id="scheduletype" number="2" selected="1">
                </Item>
            <Item id="displaytime" title_type="title_subtitle" setting_type="radiobox">
                        <Radiobox id="displaytime" number="2" selected="1">
                </Item>
                <Item id="refresh" title_type="title_subtitle" setting_type="checkbox">
                        <CheckBox checked="yes"/>
                </Item>
                <Item id=”memo" title_type="title_subtitle" setting_type="inputbox">
                        <SubTitle>InputBox2</SubTitle>
                </Item>
                <Item id=” set_date" title_type="title_subtitle" setting_type="datepicker">
                        <DatePicker id="set_date" year="1" month="1" day="21"/>
                </Item>
                <Item id=” set_time" title_type="title_subtitle" setting_type="timepicker">
                        <TimePicker id=" set_time " hour="17" minute="30"/>
                </Item>
                <Item id=” favorite" title_type="title" setting_type="favorites">
                        <Favorites id="favorite" number="3" selected="1"/>
                </Item>
            <Item id="linksetting" title_type="title" setting_type="link"/>
            <Item id="transfernow" title_type="title" setting_type="button"/>
    </SettingsResult>
</Application>
```

5.1.1 <Application>

Syntax	<Application type="string resource" version="numeric resource" language="string resource"> … </Application>	
Contained in	none	
Must Contain	none	
Can Contain	none	
Description	Application information	
Attribute	**type**	Application type. “clock” and “application” allowed.
	language	Device language

5.1.2 <packageName>

Syntax	<packageName></packageName>
Contained in	<Application>
Description	Package name
Attributes	none

5.1.3 <SettingsResult>

Syntax	<SettingsResult> … </SettingsResult>
Contained in	<Application>
Must Contain	<Item>

Description	Wrapper for the settings results
Attribute	none

5.1.4 <Item>

Syntax	<Item id="string resource" title_type="string resource" setting_type="string resource" conditional=”string resource”> … </Item>	
Contained in	<Settings>	
Can Contain	<CheckBox> <Radiobox> <SwitchButton> <Inputbox> <DatePicker> <TimePicker> <Favorites>	
Description	Each Item in the settings	
Attribute	**Id**	Id of each Item
	title_type	Title type “title” – only Title, “title_subtitle” – Title and Subtitle
	setting_type	Setting Type “checkbox” – CheckBox “radiobox” – RadioBox “switch”-SwitchButton “timepicker”-TimePicker “datepicker”-DatePicker “favorites”-Favorites “inputbox” **Note:** If “setting_type” is “inputbox”, “datepicker” or “timepicker”, the Item element must have a Subtitle

		element.
	Conditional	Value of Condition tag. “enable”

5.1.5 <CheckBox>

Syntax	<CheckBox id="string resource" checked="string resource"/>	
Contained in	<Item>	
Description	Checkbox	
	id	Id of each Item
	checked	Initial value that is Checked.

5.1.6 <Radiobox>

Syntax	<Radiobox id="string resource" number="string resource" selected="numeric resource"/>	
Contained in	<Item>	
Description	Radiobox	
Attribute	**Id**	Id of each Item
	number	RadioBox item count
	selected	Selected item number

5.1.7 <SwitchButton>

Syntax	<SwitchButton id="string resource" switch_checked="string resource"/>	
Contained in	<Item>	
Description	Switch Button.	
Attribute	**id**	Id of each Item
	switch_checked	Initial value (on/off)

5.1.8 <DatePicker>

Syntax	<DatePicker id=”string resource” year=”string resource” month=”string resource” day=”string resource”/>	
Contained in	<Item>	
Description	Used when setting up a DatePickerDialog.	
	id	Id of each Item
	year	Set the default year in a DatePicker dialog
	month	Set the default month in a DatePicker dialog
	day	Set the default day in a DatePicker dialog

5.1.9 <TimePicker>

Syntax	<TimePicker id="string resource” hour= “string resource” minute=”string resource”/>	
Contained in	<Item>	
Description	Used when setting up a TimePickerDialog.	
Attribute	**id**	Id of each Item
	hour	Set the default hour in a TimePicker dialog
	minute	Set the default minute in a TimePicker dialog

5.1.10 <Favorites>

Syntax	<Favorites id="string resource" number="string resource" selected="string resource"/>	
Contained in	<Item>	
Description	Used when setting up Favorites.	
Attribute	**id**	Id of each Item
	number	Total number of FavoritesItem
	selected	Selected FavoritesItem

5.2 Settings from Complex Settings UI

When you use Link Mode, WearableManagerService sends the settings for your application with Intents, as shown in the table below.

Intent	Description
Intent Action	**Format:** REMOTE_SETTINGS + PackageName **Example:** <packageName>com.sec.clock.weatherclock</packageName> App’s Receiver filter: REMOTE_SETTINGS.com.sec.clock.weatherclock
Write Intent	intent.putExtra(Intent.EXTRA_TEXT, (xml strings));
Intent	sendBroadcast (intent);

Table 2: Link Mode settings

6. Using Notification

6.1 Configuring Host-side Provider

To add your widget to “Notifications for Gear” category, add "com.samsung.wmanager.ENABLE_NOTIFICATION" to the uses-permission element in the Android manifest file for the Host-side application. You can send notifications to Wearable-side without posting notification on notification panel.

Sample Code

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
       package="com.samsung.android.app.watchmanager"
    android:versionCode="1"
    android:versionName="1.0" >
    <uses-s
    android:minSdkVersion="14"
    android:targetSdkVersion="17" />

    <uses-permission android:name="com.samsung.wmanager.ENABLE_NOTIFICATION"/>

    <application
        <activity>
                 …
       </activity>

    <meta-data
         android:name="master_app_packagename"
                  android:value="com.example.gearMasterApp"/>
    <meta-data
    android:name="app_notification_maxbyte"
                  android:value="300 "/>

    </application>
</manifest>
```

- “master_app_packagename” : If you add your master package in the Android manifest it will not be displayed on General notifications category.
    - General notifications : Sending notifications on Android notification panel by the Host-side application.
- “app_notification_maxbyte” : Notification max byte can be adjusted. (Default : 500byte)

6.2 Sending Notification (Host -> Wearable)

Sending intent by action “com.samsung.accessory.intent.action.ALERT_NOTIFICATION_ITEM” will send notifications to Wearable Device.

Extra name	Type	Description	Note
NOTIFICATION_PACKAGE_NAME	String	Package name	Mandatory
NOTIFICATION_VERSION	int	Notification service version (3)	Mandatory
NOTIFICATION_TIME	Long	Received time	Optional
NOTIFICATION_MAIN_TEXT	String	Main title of contents	*
NOTIFICATION_TEXT_MESSAGE	String	Text contents	**
NOTIFICATION_APP_ICON	Byte []	Application Icon	Optional
NOTIFICATION_LAUNCH_INTENT	String	Launch intent on Host-side(Package name)	Optional
NOTIFICATION_LAUNCH_TOACC_INTENT	String	Launch intent on Wearable-side	Optional
NOTIFICATION_ITEM_IDENTIFIER	Integer	Applied Item DB ID	Optional
NOTIFICATION_ID	Integer	ID for notification sync.	Mandatory
NOTIFICATION_CUSTOM_FIELD1	String	Additional Field to 3rd Developers	Optional
NOTIFICATION_CUSTOM_FIELD2	String	Additional Field to 3rd Developers	Optional

*, ** : At least one is mandatory.

Sample Code

```
    public static final String ALERT_NOTIFICATION =
“com.samsung.accessory.intent.action.ALERT_NOTIFICATION_ITEM”;
    public static final int NOTIFICATION_SOURCE_API_SECOND = 3;
    Bitmap bitmapImg;

    // Put data to Intent
    Intent myIntent = new Intent(ALERT_NOTIFICATION);
    myIntent.putExtra("NOTIFICATION_PACKAGE_NAME", “com.example.gearApp”);
    myIntent.putExtra("NOTIFICATION_VERSION", NOTIFICATION_SOURCE_API_SECOND);
    myIntent.putExtra("NOTIFICATION_TIME", System.currentTimeMillis(););
    myIntent.putExtra("NOTIFICATION_MAIN_TEXT", “Title Text”);
    myIntent.putExtra("NOTIFICATION_TEXT_MESSAGE", ”Body text);
    byte [] byteArray = convertResizeBitmapToByte(bitmapImg);
    myIntent.putExtra("NOTIFICATION_APP_ICON", byteArray);
    myIntent.putExtra("NOTIFICATION_LAUNCH_INTENT", “com.example.gearMasterApp”);
    myIntent.putExtra("NOTIFICATION_LAUNCH_TOACC_INTENT", “com.example.gearSideApp”);
    sendBroadcast(myIntent);

    public byte[] convertResizeBitmapToByte(Bitmap bitmap){
           Bitmap scBitmap = Bitmap.createScaledBitmap(bitmap, 75, 75, false);
           ByteArrayOutputStream byteArrayStream = new ByteArrayOutputStream();
           scBitmap.compress(Bitmap.CompressFormat.PNG, 50, byteArrayStream);

           return byteArrayStream.toByteArray();
       }
```

6.3 Acknowledging Notification (Wearable -> Host)

- Intent Action : "com.samsung.accessory.intent.action.UPDATE_NOTIFICATION_ITEM"
- When a notification is read in Wearable-side, you can receive it by Broadcast Receiver in Host-side. Below information for the notification will be included.

Extra name	Type	Description
NOTIFICATION_PACKAGE_NAME	String	Package name
NOTIFICATION_ITEM_IDENTIFIER	Integer	Applied Item DB ID
NOTIFICATION_ID	Integer	ID for notification sync.
NOTIFICATION_CUSTOM_FIELD1	String	Additional Field to 3rd Developers
NOTIFICATION_CUSTOM_FIELD2	String	Additional Field to 3rd Developers

-

6.4 Clearing Notification (Host -> Wearable)

- Intent Action : "com.samsung.accessory.intent.action.CHECK_NOTIFICATION_ITEM"
- When a notification cleared in Host-side, you can send an intent to be synced on Wearable-side.

Extra name	Type	Description	Note
NOTIFICATION_PACKAGE_NAME	String	Package name	Mandatory
NOTIFICATION_ID	int	ID for notification sync. For the Host-side application.	Mandatory

6.5 Show On Device

- "Show on device" on Wearable-side will launch application in Host-side referring "NOTIFICATION_LAUNCH_INTENT" field.

Extra name	Type	Description
NOTIFICATION_LAUNCH_INTENT	String	Launch intent on Host-side(Package name)

- Additional intent will be delivered to let an application could do specific action.
- Intent Action : "com.samsung.accessory.intent.action.LAUNCH_NOTIFICATION_ITEM"
- Extra information for the notification.

  -

Extra name	Type	Description
NOTIFICATION_PACKAGE_NAME	String	Package name
NOTIFICATION_ITEM_IDENTIFIER	Integer	Applied Item DB ID
NOTIFICATION_ID	Integer	ID for notification sync.
NOTIFICATION_CUSTOM_FIELD1	String	Additional Field to 3rd Developers
NOTIFICATION_CUSTOM_FIELD2	String	Additional Field to 3rd Developers

- If the value of "NOTIFICATION_LAUNCH_INTENT" is "NO_LAUNCH", only intent will be delivered without launching the Host-side application.
- Package name should be checked before using the intent.

Sample Code

```
if (PACKAGE_NAME.equals(intent.getExtras().getString("NOTIFICATION_PACKAGE_NAME") == false)
    return;
```

Copyright

Though every care has been taken to ensure the accuracy of this document, Samsung Electronics Co., Ltd. can not accept responsibility for any errors or omissions or for any loss occurred to any person, whether legal or natural, from acting, or refraining from action, as a result of the information contained herein. Information in this document is subject to change at any time without obligation to notify any person of such changes.

Samsung Electronics Co. Ltd. may have patents or patent pending applications, trademarks copyrights or other intellectual property rights covering subject matter in this document. The furnishing of this document does not give the recipient or reader any license to these patents, trademarks copyrights or other intellectual property rights.

The document is subject to revision without further notice.

For more information, please visit http://developer.samsung.com/