

Orbit Technical Solution Description Draft v0.3

Template: Technical Solution Description for Device Software & Tools Software Draft v2.0

Contents

- 1 Contribution2
- 2 Architecture Overview2
 - 2.1 System architecture diagram2
 - 2.2 Software architecture diagram.....2
 - 2.3 Additional information5
- 3 Configurability6
- 4 Interface management.....6
 - 4.1 New or modified public and platform APIs.....6
 - 4.2 Changes to the HW adaptation interface.....6
- 5 Usage of Application Framework.....6
- 6 Inclusion of new Open Source software6
- 7 Dependencies on commercial 3rd party software.....6

1 Contribution

Development proposal ID

<Insert Orbit DP ID>

Feature description

Orbit is a collection of extension libraries on top of Qt that realizes the renewed S60 mobile user experience (as described in S60 Direct UI and Orbit development proposals).

Orbit deliverables include:

- Qt-based UI layer implementing scalable S60 Direct UI, including more than 50 UI widgets optimized for mobile user experience
- Input Framework for various hardware & touch input methods
- Control & Feedback Framework, including tactile feedback and gesture support
- Internationalization and localization enablers on Qt and Symbian OS, fulfilling mobile requirements
- High quality SDK APIs.

Orbit is device software and supersedes S60 Avkon UI Framework.

Orbit will be made available to Symbian Foundation under LGPL version 2.1. Contribution model for Orbit will be defined in collaboration with Qt contribution model.

Baseline

This feature introduces new functionality.

2 Architecture Overview

2.1 System architecture diagram

Orbit is a collection of user interface libraries and services. Consequently, it does not by nature cross the mobile device boundaries.

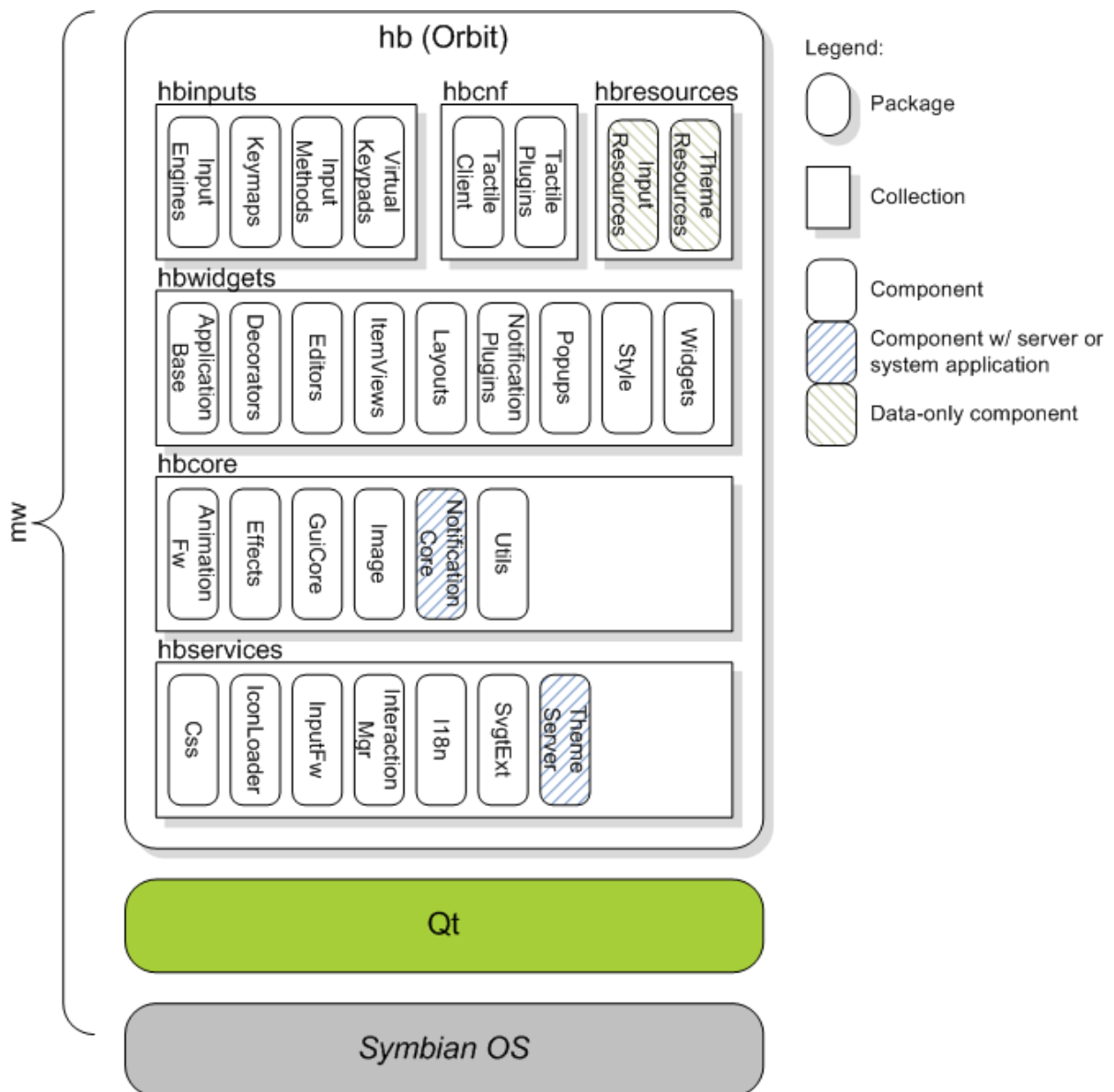
However, Orbit content uses graphics HW acceleration where available. At least OpenVG hardware acceleration support is recommended, although not technically mandatory.

2.2 Software architecture diagram

Naming

When used as a prefix or otherwise concise form, the name "Orbit" is abbreviated "hb".

Collections and components



Collection descriptions

Orbit itself is a new package. The functionality provided is organized into collection as follows, according to the dependencies and build order.

Orbit Services (*mw.hb.hbservices*) includes features that do not have dependencies to Orbit's Graphics Scene specialization (in *mw.hb.hbcore.GuiCore*). This collection contains theming support, high-performance SVG Tiny rendering, internationalization extensions, interaction manager for control & feedback plug-ins, input framework, icon loader, and specialized cascading stylesheet engine. Components in this collection are also likely to be used by Qt Symbian platform plugins.

Orbit Core (*mw.hb.hbcore*) provides core functionality for graphics view –based UI software. The components in this collection include UI base classes, animation framework, UI effects support, specialized graphics items, core for notification management, and UI utilities.

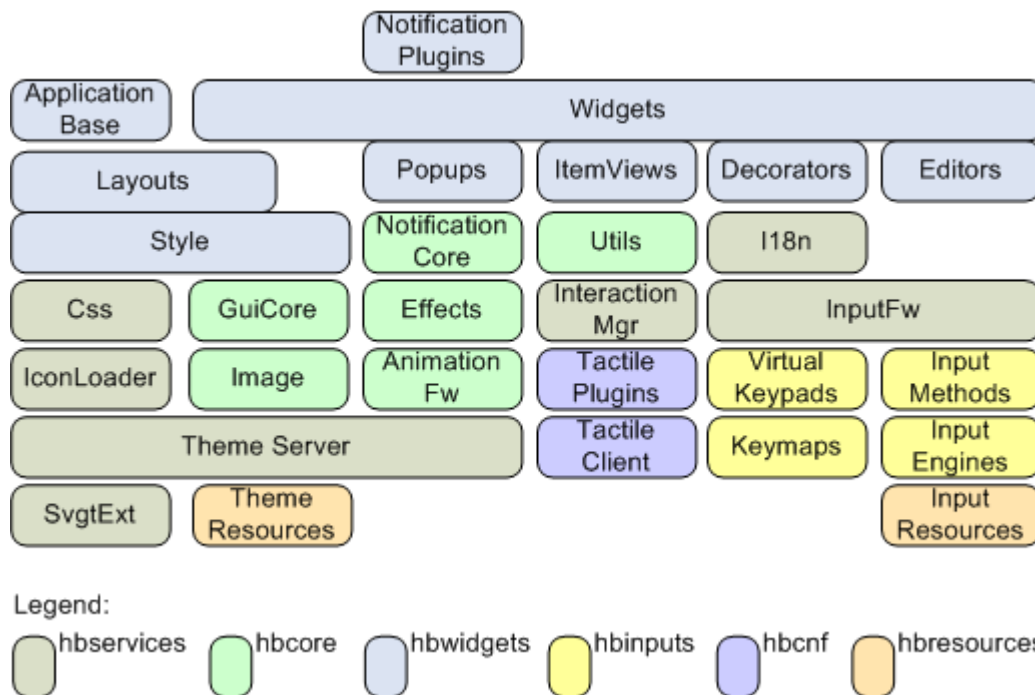
Orbit Widgets (*mw.hb.hbwidgets*) provides graphics view –based library of mobile UI widgets, including editors, item views, dialogs, and decorators. Additionally, the components in this collection contain convenience application base classes, scalable layouts support, styling support, and concrete notification plugin implementations.

Orbit Inputs (*mw.hb.hbinputs*) provides input methods, keymaps, and input engines together with their UI elements, such as on-screen virtual keypads.

Orbit Control & Feedback (*mw.hb.hbcnf*) collection contains concrete implementation of tactile feedback support for widgets and the corresponding engine integration.

Orbit Resources (*mw.hb.hbresources*) provides data-only components that include reference graphics, other themable resources, and data for inputs functionality.

Logical architecture



Component deployment

Theme Server (*mw.hb.hbservices.ThemeServer*) is deployed as a server.

Notification Manager (within *mw.hb.hbwidgets.NotificationCore*) is deployed as a privileged application with client-server interface.

The rest of the Orbit content is deployed as dynamically linked libraries and generally run within the using application process.

Implementation language

Orbit components are implemented in Qt cross-platform C++. Exceptionally, the components (and/or parts of components) with direct dependencies to Symbian-only APIs are implemented in both Symbian C++ and, at least as stubs, in Qt cross-platform C++.

2.3 Additional information

Duplication of platform functionality

Orbit provides a Qt-based mobile user interface solution. Although the Symbian Foundation provides legacy user interface technologies and features, none of these existing solutions build on Qt or realize S60 Direct UI design.

Start-up and shutdown

Theme Server (*mw.hb.hbservices.ThemeServer*) and Notification Manager (in *mw.hb.hbcore.NotificationCore*) are started automatically on start-up, regardless whether they are explicitly in the start-up list, as they are used to across UI applications. Likewise, they are in practice running all the time.

None of the Orbit components is foreseen to need to perform cleanup operations during the power down.

Power management

Animation framework used by UI effects (*mw.hb.hbcore.Effects*) uses timers when an effect is running. These timers are both shared by Qt and also destroyed when an effect completes or the application goes to background.

ROM Memory Management

According to the current estimates, the total size of Orbit executable code is likely to remain under 1 megabyte.

Additionally, the reference UI graphics and optional content – such as input methods – may result in large data files.

RAM Memory Management

Orbit, as any Qt-based software, requires maximum stack size for applications.

Platform security

Theme Server (*mw.hb.hbservices.ThemeServer*) and Notification Manager (in *mw.hb.hbwidgets.NotificationCore*) access privileged interfaces, and thus require additional platform security capabilities.

Additionally, Notification Manager must implement a security policy towards its clients and plugins to forbid execution of privileged operations by untrusted clients.

Performance

Orbit uses QPainter and the active Qt graphics system of each client process.

Notwithstanding, Orbit user experience design assumes availability of at least OpenVG graphics HW acceleration and the provided content is optimized against this assumption. For example, the full range of UI effects functionality cannot be used without matching graphics hardware capabilities. Additionally, regardless of the active graphics system, vector graphics rendering has substantial performance gain when OpenVG HW acceleration is available.

Dependencies on tools

Building Orbit requires the usual make and resource compilation tools provided with Qt.

High-performance SVG Tiny support (*mw.hb.hbservices.SvgExt*) requires NVG converter.

Theming support (*mw.hb.hbservices.ThemeServer*), Effects (*mw.hb.hbcore.Effects*), and Layouts (*mw.hb.hbwidgets.Layouts*) can read the content generated by Theme, Effect, and Layout creation tools, respectively.

3 Configurability

Individual input methods (within *mw.hb.hbinputs*) and Control & feedback plugins (within *mw.hb.hbcnf*) are optional.

4 Interface management

4.1 New or modified public and platform APIs

Orbit package provides a significant number of new public and platform APIs.

As these new APIs renew and consequently replace especially Avkon functionality, the corresponding legacy APIs should be deprecated and/or removed. Orbit will not provide functional backward compatibility for such legacy APIs.

4.2 Changes to the HW adaptation interface

Orbit does not provide, change, or directly use HW adaptation interfaces.

Efficient usage of graphics hardware acceleration may indirectly require HW adaptation API extensions to enable, for example, resource sharing and window-level effects.

5 Usage of Application Framework

Orbit supersedes legacy Avkon user interface components and introduces a new theming system.

Orbit uses the localization enablers provided by Qt for translatable text strings.

6 Inclusion of new Open Source software

Orbit will be made available to Symbian Foundation under LGPL version 2.1, and requires Qt similarly available under LGPL.

The contribution model for Orbit, which is being defined, must allow code exchange between Orbit and Qt.

7 Dependencies on commercial 3rd party software

Optional Input methods (built using *mw.hb.hbservices.InputFw* and *mw.hb.hbinputs*) may require the corresponding commercial 3rd party input engines.

Optional Tactile feedback functionality (within *mw.hb.hbcnf.TactileClient*) may indirectly require the corresponding commercial 3rd party haptics engine.

Orbit package itself (*mw.hb*) does not include by default such commercial 3rd party software.