

What's new for developers in Symbian OS v9.1?

Mark Shackman

Version 1.1

1.	EVOLUTION OF SYMBIAN OS	1
2.	ARCHITECTURAL ENHANCEMENTS	2
2.1.	GENERIC FILESERVER HOOKS.....	2
2.2.	BLUETOOTH.....	2
2.3.	GENERIC QOS (QUALITY OF SERVICE).....	3
2.4.	INTEROPERABILITY AND DEVICE MANAGEMENT	3
3.	NEW APIS	3
3.1.	BASE SERVICES APIS.....	3
3.2.	MESSAGING API	4
3.3.	UI AND GRAPHICS APIS.....	4
3.4.	CALENDAR API	5
3.5.	TIME ZONE LOCALISATION/DEPRECATION OF WORLD SERVER	6
3.6.	CONTROL FRAMEWORK.....	6
4.	JAVA ENHANCEMENTS.....	7
5.	NEW AND UPDATED TOOLS.....	7
6.	CONCLUSION.....	7

1. Evolution of Symbian OS

Symbian OS v9.1 brings Platform Security to the recently-introduced real-time kernel (EKA2). When coupled with the Symbian Signed programme, developers, phone manufacturers, network operators and end-users can be confident that Platform Security provides a more reliable, private and trustworthy open environment in which to work.

The introduction of a new tool chain (see section 5 below) means that code compiled for previous versions of Symbian OS is not binary-compatible with Symbian OS v9.1 code. In some cases, the introduction of enhanced Platform Security has necessitated changes to APIs that have broken source compatibility. Applications will therefore need to be recompiled in order to run on Symbian OS v9.1 and later.

Enhancements introduced since Symbian OS v8.1 include:

- Generic hooks are provided in to the file server
- Bluetooth stereo headset profiles and Bluetooth extended-SCO links
- Network quality of service requirements can be specified by applications
- SyncML APIs now conform to OMA standards
- Telephony now includes support for – R-USIM, CDMA network roaming support, third party OTA API, NAM programming mode and R5 IMS
- Symbian OS v9.1 provides a native RTP (Realtime Transport Protocol) stack. This stack can be used by licensee and 3rd party applications

These enhancements are discussed in section 2 below.

There have also been a number of changes to the functionality available to developers, particularly in the following areas:

- Base Services contains enhanced APIs for error handling, a writable, resizable buffer descriptor class, RBuf, and new IPCs and communications mechanisms
- A new Send-As API allows applications to create and send messages
- The graphics subsystem has been extended and enhanced
- A new Calendar API replaces the Agenda Model, adding group scheduling support

These changes are discussed in section 3 below.

2. Architectural Enhancements

Symbian OS v9.1 is built on top of the real-time kernel (EKA2). The old version of the kernel (EKA1) is no longer available.

2.1. *Generic fileserver hooks*

Generic hooks are provided in Symbian OS v9.1 so that multiple vendors can hook into the file system simultaneously; for example allowing concurrent on-access virus scanning and encryption.

2.2. *Bluetooth*

Bluetooth stereo audio headsets

Symbian OS v9.1 provides the relevant Bluetooth protocols and infrastructure to allow devices to support Bluetooth stereo audio headsets. Bluetooth stereo headsets allow stereo quality audio to be streamed from the phone to the headset so that the phone can act as a media server thus enhancing the phone's credentials as an entertainment device.

Generic Remote Control Framework

To support stereo headsets, a generic, bearer-agnostic, remote control framework is provided by the Bluetooth subsystem. The framework supports both target and controller roles and allows applications to send and receive remote control commands to and from remote devices in a bearer-agnostic manner.

Extended-SCO Support

Extended-SCO (eSCO) is a Bluetooth v1.2 feature that allows various SCO link parameters to be specified by the host. The existing Symbian API for handling SCO links (CBluetoothSynchronousLink) has been extended to also support eSCO.

PAN

Bluetooth PAN profile is designed to make a Bluetooth piconet look like an ethernet network at an application level. Symbian OS v9.1 provides support for a single active PAN network at a time, which means that each remote device that connects to the Symbian device will be merged into the same PAN network.

2.3. Generic QoS (Quality of Service)

Symbian OS v9.1 provides an interface to allow applications to specify the QoS they require from their network. This includes support to allow a number of sockets to bundle to an individual channel. The interface provides an unencumbered future proof solution for setting the QoS of a channel, supporting the use of QoS by all protocol types.

2.4. Interoperability and Device Management

The SyncML APIs have been enhanced to comply with OMA standards. Specifically, these include OMA Data Synchronisation v1.1 for Agenda & Contacts, OMA Device Management v1.1.2 for device management and support also for OMA client provisioning v1.2.

3. New APIs

3.1. Base Services APIs

A number of features have been added to the Base Services in Symbian OS v9.1 to aid developers and support Platform Security. These include:

New Leaving functions in RArray

Many of the methods of the dynamic array classes have been enhanced to include leaving versions of these functions, for better error condition handling. Specifically, Symbian OS v9.1 has added leaving versions of the existing methods in RPointerArray, RArray<T>, RArray<TInt> and RArray<TUint>.

RBuf descriptor

RBuf provides a new writable, dynamically sized, buffer descriptor class that is build-independent. This offers developers an easier-to-use replacement for HBufC, which should assist in improving program correctness. For further details, see the technical paper [Introduction to RBuf](#).

Publish and subscribe, message queues and shared buffer I/O

Publish and Subscribe allows the setting, retrieving and monitoring of system-wide variables and provides a new IPC mechanism for peer-to-peer communication between threads. In Symbian OS v9, Publish and Subscribe is deterministic and bounded (but this is not true where the mechanism has been backported to a previous version of Symbian OS). For further details, see the technical paper [Publish and Subscribe](#).

Message queues offer a peer-to-peer many-to-many communications paradigm. They allow threads to send messages without needing to know if the recipient is listening. A message queue can be globally visible, local to the current process or anonymous.

Shared buffer I/O is primarily intended for device driver communications. For further details on all three mechanisms, see the technical paper [New IPC mechanisms for Symbian OS](#).

3.2. Messaging API

New secure SendAs server

The original SendAs API ran in the client application process. Under the new platform secure OS, this model would require client applications to be trusted with several capabilities to be able to create a message. A new server, implementing the SendAs interface, has therefore been added, which allows applications with the correct capabilities to create and send messages.

The new API allows any client application to create messages in the Drafts folder of the message store and to launch an appropriate message editor for a given message type. It supports two methods of sending messages - confirmed or unconfirmed send. A confirmed send requires a confirmation by the UI MTM; typically it would query the user. An unconfirmed send does not require confirmation by the UI MTM.

The main client-side API is the RSendAs class, which the client application uses to connect to the SendAs server. The session object on the server-side is an instance of CSendAsSession.

Client applications create messages using the RSendAsMessage API. Opening an RSendAsMessage object on a SendAs server session creates a CSendAsMessage object in the SendAs server. There is a one-to-one mapping between an RSendAsMessage object and a CSendAsMessage object. This design allows client applications to create multiple messages concurrently.

3.3. UI and Graphics APIs

A number of enhancements have been made to the graphics and UI subsystems in Symbian OS v9.1.

Animated user interfaces

A new component in Symbian OS v9.1 provides animated graphics support. It allows the creation of animated System GUIs, including support for animated system and application icons.

Two types of animation are supported: basic client side animations (CBasicAnimation) and animated sprites that run in the window server thread (CSpriteAnimation).

Data is passed to the animation through a generic interface, implemented as a CAnimationDataProvider, to allow the re-use of generic animation code for different data formats. A concrete data provider, (CICLAnimationDataProvider) which animates data formats supported by the image converter library (ICL), for instance animated GIFs, has been implemented.

Icons replaceable while application runs

An application's icon can now be replaced while the application is running. This allows the icons to be customised, perhaps as part of a UI theme.

Control text alignment

Symbian OS v9.1 now provides the concept of the “maximum width” of a control, which only affects the controls’ calculation of their minimum size. Functions have been added to the `CCoeControl` class to vertically align the text in multiple horizontally adjacent controls of different concrete types; and to align the text in a control with a known baseline, allowing a control to inform the caller of the position of its text.

Screen orientation selection

Applications can select the orientation of the screen when a view is switched to, so that individual applications can have a landscape view when the rest of the device is in portrait or vice versa. A screen mode switch hook, `MCoeView::PrepareForViewActivation()`, is available which allows application intervention when switching screen mode.

Text layout

A font independent way to allow consistent layout of text on the screen (font metrics etc.) is provided. It allows horizontally adjacent controls containing text, for instance a text label and a text editor, to be aligned with each other.

Application language determination

The new method `RApaLsSession::ApplicationLanguage()` returns the language used by the application to display its UI.

Task list dismissal

The new method `CEikonEnv::DismissTaskList()` causes the task list, if displayed, to be closed.

Application font change

Symbian OS v9.1 provides a new method for changing the font currently being used by an application and notifying all necessary components that it has been changed. The change could be an attribute change such as the font size or it could be a completely different font.

Transition effects

Several changes have been made to support transition effects, where screen elements may grow, shrink, move, fade in or fade out, while the background remains unchanged. For instance, a menu bar may appear to fade in gradually, changing from transparent to opaque.

Window transparency

A new 16 million colour display mode `EColor16MA` has been added.

3.4. Calendar API

A new version of the agenda model has been introduced (`CalendarAPI`). It is implemented as a thin wrapper over the existing agenda model implementation, which is now internal to Symbian. The new API provides all of the existing agenda model's functionality, but also supports some additional properties to enable group scheduling behaviour and to support the iCalendar standard.

The new Calendar API is not binary or source compatible with the existing API. The intention is to preserve the Calendar API in future releases, but to replace the underlying agenda model implementation with an entirely re-architected calendar component. It has the following features:

- Redundant classes and functions from the Agenda Model have been removed.

- New fixed time behaviour allows agenda entries and alarms to be displayed with correctly adjusted date and time when the user has moved into a different time zone, i.e. there is no need to re-queue alarms when this happens.
- A new class, `TCalendarTime`, which is used for all APIs, is introduced to replace `TTime`, `TAgDate` and `TAgDateTime`. This requires the client to set the time to system local or UTC time explicitly before passing the time parameters to the functions.
- Users can create and store entries with data that conforms to the iCalendar standard.
 - Meeting participation enhancement - The attendee attribute of an entry has been extended to fit the `ATTENDEE` property in iCal.
 - Group scheduling data storage – Entries can now hold properties defined in RFC2445 such as the unique identifier (UID), sequence number (SEQUENCE), Recurrence ID (RECURRENCE-ID), repeating events defined by the `RDATE` property, local time zone rules and the `METHOD` property.
 - Group scheduling entries that act on the same event can be stored and retrieved in an associative manner by means of a common UID.
- New observer features - The user can observe the changes in the file made by other clients.
- Calendar alarms will be managed in the Calendar server without interaction from clients.
- Todo list are no longer supported.

3.5. Time zone localisation/deprecation of World Server

The World Server API provides access to a database of country and city information. This is deprecated in v9.1 and has been replaced by a new component called Time Zone Localisation.

Time is now handled in a more consistent manner in all components of Symbian OS. In particular, Symbian APIs that use time will always be in Coordinated Universal Time (UTC), unless explicitly documented to be in local time. The responsibility for handling time zones and daylight saving has moved out of the kernel and into a new user-side time zone server.

This allows localised time zone names, including standard, daylight saving, short standard and short daylight saving names, to be defined in resource files. Localisable city names and city groups can also be defined, and cities can be associated with time zones.

3.6. Control framework

The control framework has new functionality that allows controls to:

- have their own zoom factor and zoom type;
- better support for getting and changing the application font;
- have a themeable system text that allows the text used inside controls to be drawn according to the current *theme* or *skin*;
- support aligning controls according to their text;
- support control layout managers to make it easier for compound controls to lay out their component controls.

4. Java Enhancements

The MIDP Java platform has been fully integrated into Symbian OS v9.1 to provide a seamless transition for Java developers. It has general performance improvements and support for the javax.obex package has been provided in Symbian's implementation of JSR082.

The performance of Symbian's Java graphical implementation has been improved between v8.1 and v9.1 by about 300% according to JBenchmark 1.0 and by about 500% according to JBenchmark 2.0.

Startup time for MIDlets has been reduced. The improvement may only be noticeable for larger and more complex MIDlets.

5. New and Updated Tools

Symbian OS v9.1 is built with ARM's RealView Compilation Tools (RVCT) 2.2, which fully supports the ARMv5 and later architectures. RVCT delivers substantial performance and footprint benefits compared to the earlier GCC compiler and, by implementing the ARM ABI (Application Binary Interface), allows applications to be built with alternative, compatible compilers.

Symbian OS v9.1 continues to support a GCC compiler for low-cost application development, with the required version having been updated from GCC 2.98 to GCC 3.4. As GCC 3.4 is compatible with RVCT 2.2, developers therefore have a choice between compilers.

The adoption of these new tools means that code compiled for previous versions of Symbian OS are not binary-compatible with Symbian OS v9.1 code. Applications will have to be recompiled to run on Symbian OS v9.1 and later.

6. Conclusion

For further information on these changes, see the Migration Guides in the Symbian OS Developer Library which comes with the DevKit.

[Back to Developer Library](#)

Want to be kept informed of new articles being made available on the Symbian Developer Network?

[Subscribe to the Symbian Community Newsletter.](#)

The Symbian Community Newsletter brings you, every month, the latest news and resources for Symbian OS.