

What's new for developers in Symbian OS v9.2?

Mark Shackman

Version 1.0

1. INTRODUCTION	1
2. ARCHITECTURAL ENHANCEMENTS AND NEW APIS	2
2.1. GRAPHICS AND MULTIMEDIA.....	2
2.2. HARDWARE	2
2.3. ENTERPRISE.....	2
2.4. IP TELEPHONY AND NETWORKING	3
2.5. MESSAGING.....	4
2.6. PLATFORM SECURITY.....	4
2.7. UI IMPROVEMENTS.....	4
2.8. CONNECTIVITY.....	5
2.9. DEVICE MANAGEMENT	6
2.10. PIM APPLICATIONS	7
3. CONCLUSION.....	8

1. Introduction

Symbian OS v9.2 is targeted to address the prosumer 3G handset market segment and represents a major step for the Symbian platform as it is the first product to be subject to a strict compatibility policy. This has delivered on Symbian's commitment to a backward binary compatible release to v9.1, whilst still providing key updates and improvements in the performance of the Symbian OS.

Enhancements introduced since Symbian OS v9.1 include:

- **Graphics and Multimedia** – support for additional input data types including YUV and YCbCr; Camera API enhancements
- **Hardware** – support for ARMv6
- **Enterprise** – consistent handling of time values; floating time support for Calendar entries; alarm services enhancements
- **IP Telephony and Networking** – RTCP, SIP and SDP implementations; automatic initial bearer selection
- **Messaging** – SMS support compliance with 3GPP release 6
- **UI Improvements** – SVG icon support; Vietnamese and Hindi language support
- **Connectivity** – publication of software install, backup and restore protocols; OBEX APIs
- **Device Management** – OMA DM-compliant management; OMA DM client alert mechanism; OMA DataSync support to version 1.2; additional synchronisation support
- **PIM Applications** – support for additional contact fields

These enhancements are discussed in section 2 below.

2. Architectural Enhancements and New APIs

2.1. *Graphics and Multimedia*

Colour space support in the image conversion library (ICL)

The image conversion library has been updated to support input data types, including YUV and YCbCr, which represent colour spaces other than RGB.

YUV is the natural input data type for both stills capture (JPEG) and Video (MPEG). YCbCr was defined in the ITU-601 (formerly CCIR-601) standard for use with digital component video. Camera hardware generally returns two types of data: pre-encoded JPEG, which is more common today, and YUV which is becoming increasingly popular. In order to support camera modules that expose YUV raw images and then encode these as JPEG images, the ICL supports YUV and YCbCr as input data types. Adding support for these ensures minimal need for format conversion, which in turn leads to potentially faster camera operations.

Camera API (ECAM) enhancements

The Camera API (ECAM) has been enhanced and extended to eliminate proprietary extensions and provide for future growth by supporting the following features:

Image resolution, ISO rating, optical zoom setting, digital zoom setting, image stabilisation, auto focus mode and type, manual focus distance, macro mode, white balance, aperture, shutter speed, manual flash power level, external flash, flash modes, flash compensation, flash compensation step, exposure compensation, exposure compensation step, metering mode, exposure modes, drive mode, bracket mode, bracket step values, bracket parameter, timer value, time lapse value, output file format, and picture orientation.

A new callback, providing a 'capabilities changed' notification, enables the dynamic addition and removal of pluggable camera devices.

2.2. *Hardware*

ARMv6 Support

Symbian OS v9.2 now supports the ARMv6 instruction set architecture, with code being optimised by Symbian's RVCT compiler for the ARM1136 core. The command-line build tools are capable of building with an ARMv6-specific option set and, as necessary, ARMv6-specific file locations for intermediate and final build products.

Symbian OS v9.2 debug APIs support ARMv6 debugging, including all ARMv6-specific instructions, and handle any MMU implications.

The Symbian OS kernel exploits the new features of ARMv6's MMU, including physically-tagged cache, single page table per address space ("multiple memory model"), and load/store exclusive instructions. Using the ARMv6 instruction set brings particular benefits to the performance of context switches: physically-tagged cache eliminates cache flushing, and one page table per address space reduces work compared with modification of a single page table. Additionally, the ARMv6 can mark pages as non-executable, further enhancing security on Symbian phones.

2.3. *Enterprise*

UTC and Automatic DST support

All components of Symbian OS v9.2 have been updated to handle time in a consistent manner. In particular, Symbian APIs that use time (specifically `TTIME`) will always use Universal Time (UTC), unless explicitly documented to be in local time.

Automatic DST (daylight saving time) also means that the user does not need to be involved in determining the DST value for a particular location. Once the user sets the location on the phone, DST will be applied automatically, allowing for a better user experience.

Floating time support for Calendar entries

Symbian OS v9.2 provides the ability to create both repeating and non-repeating calendar entries, including To-Do entries and entries with alarms, in floating time (i.e. local time independent of time zone). Floating time is required to ensure that a day event is always shown on the same calendar day, or a wake-up call will happen at 7am local time, no matter where the user is.

Enhancements to alarm services

Symbian OS v9.2 contains a number of enhancements to the alarm services APIs (agenda server, alarm server, app framework UIKON alarm alert server) including:

- giving UI applications access to details of alarms in the alarm queue
- allowing different actions for an alarm. Alarms need to be richer than just a sound and this change makes it possible for an alarm to start a video, radio station, MP3 track, etc. Such an alarm will (as with "normal" alarms) go off even if the phone is on standby

Several alarms (either clock or agenda alarms) may be queued to go off at the same time. These enhancements allow the handling of such simultaneous alarms in a flexible manner e.g. to list all alarms that are due to go off, prioritise alarms, etc. Currently only the first alarm in the alarm queue is visible to the UI, and the next alarm is only visible when the user acknowledges the first alarm.

2.4. IP Telephony and Networking

Real time control protocol (RTCP) for Push-to-Talk over Cellular

A subset of the Real Time Control Protocol (RTCP) is provided, which is suitable for use with Push-to-Talk over Cellular (PoC) clients. This conforms to the PoC consortium specification v1.0 and RFC 3550.

Session Initiation Protocol (SIP) and Session Description Protocol (SDP)

Symbian OS v9.2 provides a SIP protocol implementation which is suitable for, but not limited to, Instant Messaging (IM), Push-to-Talk over cellular (PoC), IP telephony and presence based applications. The APIs allow applications to interact at the 'transaction' and the 'dialogue' levels (these levels are defined in the relevant IETF RFCs).

A SPI which enables extensions by adding message types and methods without modifying the existing implementation is provided; this is usable by applications. SIP events are logged by the existing call logging engine. The level of logging is similar to that provided for CS calls, and includes missed, received and originated invitations. SIP is fully integrated with Symbian's networking protocols and runs over both IPv4 and IPv6.

Automatic Initial Bearer Selection when Connecting to a Remote Service

Symbian OS v9.2 provides an API for automatic initial bearer selection, based on currently available bearers, a specified bearer priority and a specified network. The API permits:

- specifying bearer priority
- setting-up of a connection with automatic bearer selection

Research suggests that there will be around 100million phones by 2010 supporting multiple bearers (e.g. WLAN, 3G, GPRS...) and therefore potentially using this feature. This feature is key for enabling bearer mobility on Symbian OS-based devices without which application developers will have to individually implement this feature.

This will enable any on-line application to easily take advantage of automatic bearer selection. Assuming that the UI will provide a sensible default, this will enable simple bearer prioritisation for all online applications. This feature can also help users manage the cost of accessing mobile services by allowing them to predefine which bearers should be used for access when multiple bearers are available.

2.5. Messaging

SMS Compliant with 3GPP R6

Symbian OS v9.2 supports the latest version, Release 6 (2004), of the SMS specification, in line with the 3GPP specifications (3GPP TS 23.040 V6.5.0 and 3GPP TS 23.038 V6.1.0). Changes have been made to the CSmsMessage API to:

- allow hyperlinks (URLs) to be inserted and extracted from an SMS message (3GPP TS 23.040 sections 9.2.3.24 and 9.2.3.24.12). The SMS stack which uses CSmsMessage also handles a new Information Element (IE) field in the SMS where the hyperlink information is stored.
- allow inclusion of a return address that is different to the sender's address. This provides a mechanism to reply with an SMS to a message initially sent as Cell Broadcast (3GPP TS 23.040 sections 9.2.3.24 and 9.2.3.24.10.11.15).
- enable the reception and subsequent handling of a video message waiting indication in an SMS (3GPP TS 23.040 section 9.2.3.24.2).
- enable the reception and subsequent handling of enhanced voice mail information in an SMS such as list of voice mail messages, time a message was left, duration and caller etc (3GPP TS 23.040 sections 9.2.3.24 and 9.2.3.24.13).
- allow the sending and processing of SMS messages that indicate automatic deletion (3GPP TS 23.038 section 4).

2.6. Platform Security

OCSP-based SIS File Revocation

The software install engine uses OCSP (Online Certification Status Protocol, RFC2560) to check the validity of SIS file signatures. Checks are performed at install time, and on request post-install. At install time, if a valid OCSP response indicates that a SIS file signature has been revoked, installation will be aborted.

2.7. UI Improvements

Enable control of behaviour of child window when parent exits

A new API allows control of the behaviour of a child window when the parent exits - either the child will remain or it will also exit.

Auto detection of Japanese Character Set

Symbian OS v9.2 includes a character conversion method that can convert from any of the following character sets: Windows-31J (Microsoft code page 932), JIS (ISO-2022-JP-1), EUC, UTF8, and UCS2.

Vietnamese and Hindi Language Support

The text rendering and editing capabilities of Symbian OS have been extended to include the Quoc Ngu (accented Latin) and the Devanagari scripts.

Support for in-scene SVG icons

Symbian OS v9.2 allows application developers to use scalable SVG (scalable vector graphics) icons that are rendered in-scene with the rest of the GDI-drawn system GUI.

Application developers have to support multiple screen resolutions and screen sizes. Next generation smartphone UIs primarily target QVGA screens in the range 2.2" to 2.8" but moving into mid-tier devices requires support for a range of screen sizes and resolutions, for example QVGA, HVGA and VGA between 2.0" and 2.8". In order to achieve high quality UIs and good readability, generally all screen elements (e.g. icons and fonts) will have to maintain their physical size (i.e. scale) across these different screens and devices. As an example, an icon for a VGA screen will have to be twice the size of an icon for a QVGA screen. For an application icon that means 32*32 vs. 64*64 pixels.

2.8. Connectivity

Symbian Connect Protocols

The protocols for software install, backup & restore and file management are published.

Historically, some implementers have tweaked the protocols to satisfy their own needs, making it impossible to have a generic set of protocols for PC connectivity. This resulted in the introduction of the SCOM component as part of the PC suite. Publishing the protocols for Symbian OS v9.2 will result in standardisation, meeting the needs of the small community of PC developers looking to create software which works with Symbian based devices.

Notification at the start and completion of the final packet transmission of an OBEX put request

The OBEX API in Symbian OS v9.2 has been extended to offer an (OBEX client) application the ability to request notification when the final packet for a PUT request has:

- started transmission, and
- been transmitted.

Currently the OBEX client only raises a notification when the message has been acknowledged by the server.

New OBEX API to allow asynchronous Put and Get

The OBEX API in Symbian OS v9.2 has been extended to allow asynchronous responses and processing of Put and Get. The asynchronous behaviour operates alongside the current synchronous behaviour. However, only one mode - asynchronous or synchronous - can be used within one instantiation of the OBEX server. Multiple instantiations of the OBEX server are allowed.

Currently OBEX blocks the thread whilst waiting for an OBEX server application to react to a put or a get request. This synchronous behaviour is restrictive when large data objects are being transferred, as the server application has limited opportunity to prepare new data for transmission or store new incoming data.

2.9. Device Management

Application lifecycle management

Symbian OS v9.2 includes an OMA DM-compliant management object, which is a plug-in to the Symbian OS OMA DM framework. This connects to an OMA DM server to allow the server to manage the applications on a Symbian OS device.

The remote server, using the management object, is able to list the set of currently installed applications, request installation of new applications and request removal currently installed applications. The management object uses the Symbian OS software install component to carry out the requests and therefore supports the application types software installer supports currently including SIS files, JAR files and JAD files.

Installation of applications includes the ability to upgrade to a newer version.

OMA DM Client Alert Mechanism

The Symbian OS DM framework API has been extended to allow device software to alert the DM server that a condition has occurred, which requires intervention from the network operator or IT department. The implementation communicates with the server using the OMA DM 1.2 Generic Client Alert mechanism.

The DM framework conforms to the mandatory requirements of the OMA DM 1.2 specifications. In particular:

- generic client alert mechanism
- correlator
- using HTTP only over a secured transport, i.e. TLS
- updating the DM Account Object to the 1.2 version.

Processes calling the extended API will require the same platform security capabilities as existing clients of the DM framework.

This makes available a basis on which to provide OMA compliant remote diagnostic client software. The ability to provide customer care solutions and diagnostics solutions which are cost effective across a very large user base requires a solution which is standards based. As devices become more complicated, network operators and IT departments are finding more and more value in providing these solutions. This will also be used by asynchronous device management activity such as firmware update where the DM session has ended before the DM activity ended and there is need to report status back to the server.

Upgrade OMA DataSync support to version 1.2

The Open Mobile Alliance (OMA) DataSync protocol in Symbian OS v9.2 has been updated to be version 1.2 compliant. In particular, it includes:

- support for the suspend and resume features of OMA Data Sync 1.2
- new APIs to support filtering of synchronisations

The new APIs allow a filter to be applied to the data to be synchronised. The filters will be passed to the DataSync server as part of the session, in compliance with the OMA DS 1.2 protocol. The actual filters supported are determined by the data provider plug-in for each application service. The user interface API will be extended to allow the UI to interact with the user to select and modify the filters to be used.

Filtering of synchronisations allows a user to restrict the data that is available on small footprint devices compared to that which is held on the server. A prime example of this would be to filter out attachments in email.

Hierarchical synchronisation

The DataSync framework has also been extended to support hierarchical synchronisation in accordance with the OMA DS 1.1 specifications. It provides new APIs which allow an application service to synchronise data which is stored across multiple stores such as file system directories or email folders.

Synchronisation of email is a core requirement for enterprise solutions. The current solutions only require the synchronisation of the INBOX and OUTBOX, but enterprise users wish to be able to select which mail boxes are available on the device. A service provider who wishes to be able to synchronise data which is stored in a file system, such as pictures or MP3 files, requires hierarchical synchronisation to be able to select which sub-folders should be synchronised.

Protocol Support for Mobile ActiveSync v2.5.5

Symbian OS v9.2 provides APIs that implement the Mobile ActiveSync protocol version 2.5.5 with respect to synchronising email, contacts and agenda via a HTTP/HTTPS connection with a Microsoft Exchange 2003 server. This implementation follows the "Microsoft Protocol Reference Specification version 2.5.5" and the API will enable Exchange 2003 server synchronisation to be integrated into PIM application suites.

2.10. PIM Applications

Support for additional Contacts fields

Support for the fields given below has been added to both the Contacts database and vCard import/export, so that it is possible to synchronise these fields.

If the property is not found in the vCard 2.1 specification, the x-property extension format will be used.

The new fields to be added are:

- **Department**
vCard format: ORG:;<department name>
Data type of field: string
NOTE: The Versit parser supports parsing department data to and from ORG property in vCard in addition to the current "Company Name" field parsing. Department name is the second field in ORG property.
- **Assistant name**
vCard format: X-ASSISTANT
Data type of field: string
- **Assistant phone**
vCard format: X-ASSISTANT-TEL
Data type of field: string
- **Anniversary**
vCard format: X-ANNIVERSARY
Data type of field: Same as with BDAY property. The date will be exported in ISO 8601 basic format (e.g. 20050503) for uniformity. It will be possible to import both ISO 8601 basic and extended (e.g. 2005-05-03) formats
- **Spouse**
vCard format: X-SPOUSE
Data type of field: string

- **Children**
vCard format: X-CHILDREN
Data type of field: string
- **Class**
vCard format: X-CLASS
Data type of field: string
NOTE: The currently required values for the field are PUBLIC, PRIVATE and CONFIDENTIAL, but generic string datatype is appropriate for ensuring flexible expandability for values. The value defining responsibility is at the application side

3. Conclusion

A brief summary of the key features and technical specifications of Symbian OS v9.2 can be found on the Symbian web site in the [Symbian OS v 9.2 product sheet](#).

Further information on these changes will be available in 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.