

S60 3rd Edition

Internet Radio Reference Design Document

Copyright © 2005, 2006 Nokia.

This material, including documentation and any related computer programs, is protected by copyright controlled by Nokia. All rights are reserved. Copying, including reproducing, storing, adapting or translating, any or all of this material requires the prior written consent of Nokia. This material also contains confidential information, which may not be disclosed to others without the prior written consent of Nokia.

Nokia is a registered trademark of Nokia Corporation. S60 and logo is a trademark of Nokia Corporation. Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. Other company and product names mentioned herein may be trademarks or tradenames of their respective owners.

Change history:

Version	Date	Status	Comments
1.0	04-04-06	Public release	First public release

Table of contents:

1.	Document control	4
1.1	References	4
1.2	Documentation conventions.....	4
1.3	Abbreviations	4
1.4	Definitions.....	4
2.	Introduction.....	5
2.1	Motivation	5
3.	Design	6
3.1	Class diagrams	6
3.1.1	UI Framework.....	6
3.1.2	Application Engine	6
3.2	Plugin	7
3.2.1	SHOUTcast® and protocol.....	7
3.2.2	SHOUTcast® Plugin Adapter	8
3.2.3	Decoder dependency	10
3.2.4	SHOUTcast® Stream Events	10
4.	Usage.....	11
4.1	Building.....	11
4.2	Signing and Installation.....	11
4.3	Playlist Management	11
4.4	Navigation.....	11
5.	Miscellaneous	15
5.1	SVG Application Icon.....	15

1. Document control

1.1 References

[1]	<i>Series 60 3rd Edition SDK for Symbian OS</i>	www.s60.com
[2]	<i>S60 Platform: Porting from 2nd to 3rd Edition</i>	www.s60.com
[3]	<i>Developer Certificate FAQ and Troubleshooting</i>	www.symbiansigned.com

1.2 Documentation conventions

Code is shown with `Courier New` font.

1.3 Abbreviations

ECOM	A Symbian OS plugin architecture component. Maintains a dynamic registry of libraries (plugins). The libraries must be built in conformance with the ECOM specifications. ECOM is used for registering and loading pluggable components. See reference document [1] for details.
MIME	Multipurpose Internet Mail Extensions. A standard that specifies how messages must be formatted so that they can be exchanged between different systems.

1.4 Definitions

Plugin	A specialized module (library) that provides multimedia services (e.g. audio, video, codec).
Platform Security	A platform-wide security framework introduced in Symbian OS version 9 to control access to sensitive APIs.

2. Introduction

The S60 Internet Radio reference application is a fully working example of an audio application developed for the S60 3rd Edition platform. The application supports SHOUTcast® streaming audio playback. (See www.shoutcast.com) Currently, MP3 and AAC+ streams are supported.

The application also supports local playback of audio files in the following formats: MP3, AAC, eAAC+, MP4, M4A, WMA, 3GPP, AMR, and WAV. Note that some formats may not be supported on some S60 3rd Edition products. This is dependent on the codecs that are available on specific phone models, and is not a limitation of the reference application.

SHOUTcast is a registered trademark of America Online, Inc. in the United States and/or other countries

2.1 Motivation

The S60 development community could benefit from an open source project that shows how an audio application might be written using the S60 3rd Edition SDK. It is provided as a learning tool and to stimulate the creation of innovative S60 audio applications.

3. Design

The S60 Internet Radio application consists of two modules: the application UI and the application engine. The application UI provides the user interface logic and presentation of the engine data while the application engine allocates resources, provides logic for audio playback, and interacts with other subsystems such as file system and communication.

The UI and Engine model allow for the individual components to evolve independently. For example, the current Internet Radio Reference UI, which is based on a single view architecture, can be replaced with a more advanced UI with multiple views without affecting the engine.

3.1 Class diagrams

3.1.1 UI Framework

This is the S60 view architecture of the S60 Internet Radio application. The application has a single view.

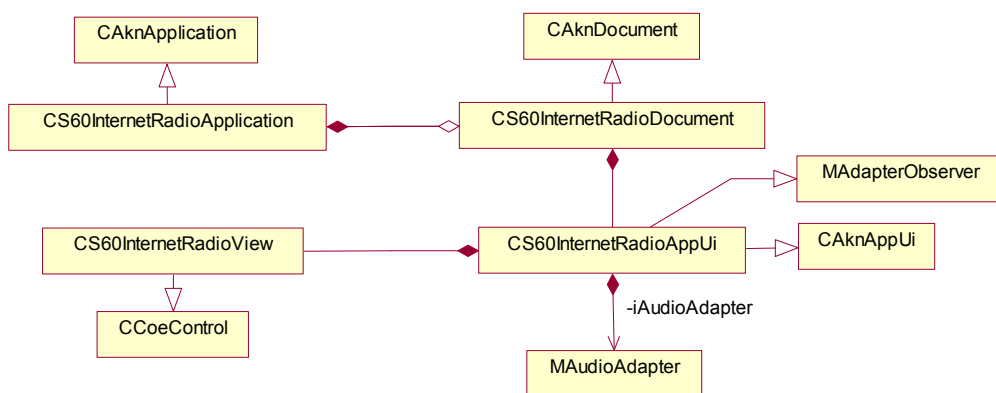


Figure 1 UI Class Relationships

3.1.2 Application Engine

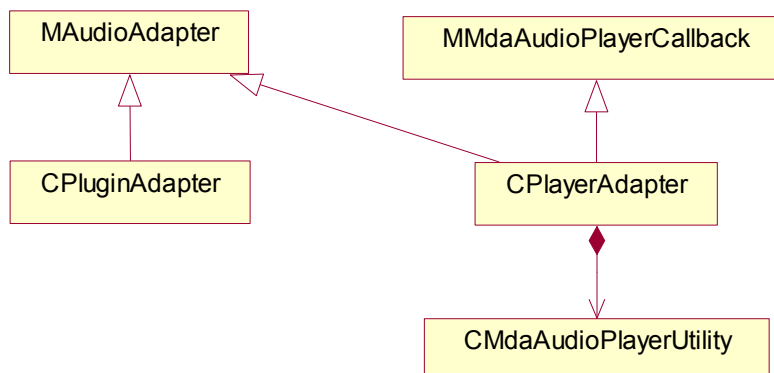


Figure 2 Application Engine Class Relationships

The application engine defines a generic interface for audio playback with class `MAudioAdapter`. All adapters derive from `MAudioAdapter` and implement the adapter interface. The application UI interacts with the engine via the `MAudioAdapter` interface.

The `CPluginAdapter` class is an adapter for plugin components. The application engine is designed for extensibility via the ECOM plugin mechanism. Additional features such as the SHOUTcast stream playback feature are implemented as an ECOM plugin to the S60 Internet Radio application. The local file playback function is a standard built-in feature of the S60 Internet Radio application and therefore not implemented as a plugin.

The `CPlayerAdapter` class is an adapter for the `CMdaAudioPlayerUtility` which is used to open, play, and obtain information, such as metadata, from a file.

The Application UI receives events from the adapter by implementing the `MAdapterObserver` interface.

3.2 Plugin

As mentioned earlier, the ECOM plugin architecture is used to provide a mechanism to add features to the S60 Internet Radio application. The SHOUTcast® stream playback feature is implemented as an ECOM plugin.

The SHOUTcast plugin is selected by the standard ECOM resolution. In order for the ECOM resolution to be able to find and instantiate the plugin, some details about the interface must be given to ECOM. The SHOUTcast plugin uses the ECOM registration resource file to register the `CShoutcastAdapter` interface UID, the implementation UID, and other information with ECOM since the interface can have many implementations. Once registered, it can be “discovered” through the ECOM resolution process.

When the client wants to use the `CShoutcastAdapter` interface, it asks ECOM to list all implementations that matches. The client can then ask ECOM to create one of the interface implementation on the list. ECOM instantiates the interface implementation and returns the interface implementation to the client. Subsequently, the client uses the interface normally.

In general, this is how the SHOUTcast plugin is launched. For more in-depth information, refer to [1].

The following sub-chapters describe SHOUTcast protocol and the SHOUTcast plugin in more detail.

3.2.1 SHOUTcast® and protocol

The SHOUTcast® streaming audio system was developed by Nullsoft Inc. for IP-based network like the internet. It enables anyone to stream audio from their PC and enables anyone with a client

capable of receiving and decoding the stream to listen to the audio. Currently, the stream formats supported are MP3 and AAC+.

The client application connects to the SHOUTcast server and requests a stream in the same way a web browser requests a page from a web server – by issuing an HTTP request. The following is an example request and response message.

An example HTTP request:

```
GET path HTTP/1.0\r\n
User-Agent:OSSPlayer/1.0\r\n
Accept:*/*\r\n
icy-metadata:1\r\n
Connection:close\r\n
\r\n
```

The SHOUTcast server responds with the following example header:

```
ICY 200 OK\r\n
icy-noticel:<BR>This stream requires <a
href="http://www.winamp.com/">Winamp</a><BR>
icy-notice2:SHOUTcast Distributed Network Audio Server/Linux v1.x.x<BR>
icy-name:Server Name\r\n
icy-genre:Genre text\r\n
icy-url:http://www.shoutcast.com\r\n
content-type:audio/mpeg\r\n
icy-pub:1\r\n
icy-br:128\r\n
icy-metaint:8192
\r\n (end of header)
```

ICY 200 indicates that the request to the server was successful.

The SHOUTcast server responds with icy tags with information necessary to configure the SHOUTcast plugin for stream playback.

Content-Type indicates the MIME type of the stream, i.e. audio/mpeg (MP3) or audio/aacp (AAC+)

icy-br indicates the bitrate of the stream

icy-metaint:x indicates to the client that every x bytes of encoded audio data, there will be a metadata block.

After the header information, the audio data will begin.

3.2.2 SHOUTcast® Plugin Adapter

The SHOUTcast plugin design has the following class relationships.

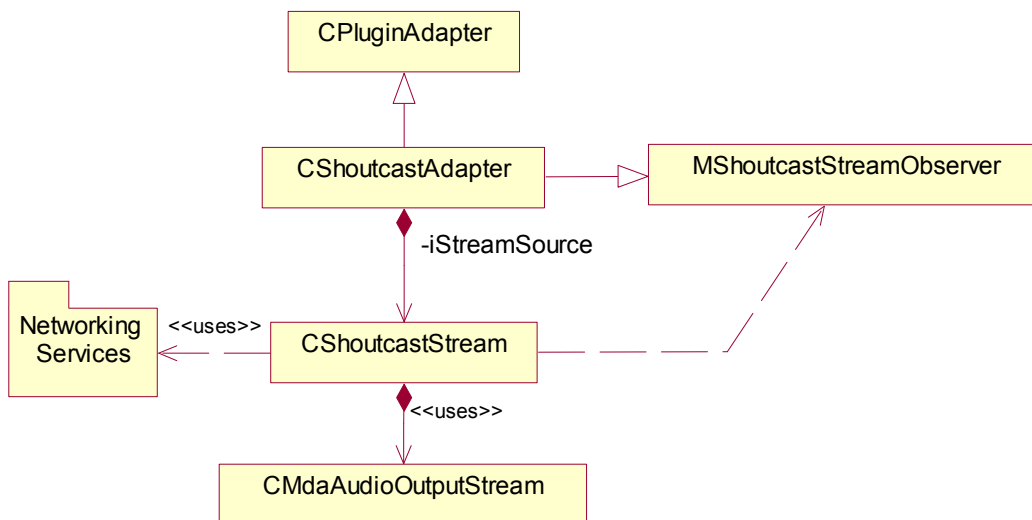


Figure 3 Plugin Class Relationships

The `CShoutcastAdapter` implements the `MAudioAdapter` interface. When the Shoutcast plugin is instantiated, it will look for playlists (files with `.pls` extension) within the `\Shoutcast\` directory of each drive. All playlists that are found will be parsed and added to the internal list of URL and the UI is updated with new menu items.

The `CShoutcastStream` encapsulates the network interaction, the stream buffering, metadata extraction, and audio stream output interaction.

Network Interaction

When the client selects a URL, the `CShoutcastAdapter` constructs a `CShoutcastStream` object with the specified URL as an argument. The `CShoutcastStream` object opens a socket and begins establishing a connection with the SHOUTcast server. Once connection is established, the HTTP request is sent to the SHOUTcast server. Upon receiving the response header the `CShoutcastStream` parses the header and reads additional data from the socket.

Stream Buffering

Stream data is read into a circular buffer. The stream tries to maintain the data level at capacity but when the data level drops below a threshold, the stream will cause the playback to pause and rebuffering will occur.

Metadata Extraction

Metadata is extracted according to the icy-metadata interval received in the SHOUTcast response header, therefore, a byte count of the audio data needs to be maintained. At the point where metadata block should appear in the buffer, the first byte, multiply by 16, indicates the length of the metadata.



Since metadata can appear in the middle of the encoded audio data, the metadata is extracted and the encoded audio data is rejoined. The byte count is reset and the count starts again with the first byte of encoded data after the metadata.

Output Stream Interaction

The `CMdaAudioOutputStream` utility is used to play streaming audio. First, the `CMdaAudioOutputStream::Open()` is called to open an audio output stream. `MMdaAudioOutputStreamCallback::MaosOpenComplete()` callback is called by the framework when the stream has been opened and is ready to receive audio data. Then the `CMdaAudioOutputStream::SetAudioPropertiesL()` method is used to configure the output stream with the previously received, bitrates. The `CMdaAudioOutputStream::SetDataTypeL()` is used to set the audio format, the previously received content-type is translated to an equivalent FourCC, so the correct decoder can be selected. The stream data is written to the output with the asynchronous `CMdaAudioOutputStream::WriteL()`. When the data has been copied, the `CShoutcastStream` is notified with `MMdaAudioOutputStreamCallback::MaoscBufferCopied()`. The next `WriteL()` can be called once the acknowledgement is received.

3.2.3 Decoder dependency

The SHOUTcast plugin uses the `CMdaAudioOutputStream` utility to play streaming audio and the decoding of the audio stream is done by the audio subsystem that manages and allocates the audio decoders. The availability of certain decoders varies from products to products, therefore, when the system can't find an appropriate decoder to decode the stream, the `CMdaAudioOutputStream::SetDataTypeL()` method will leave with `KErrNotSupported`.

3.2.4 SHOUTcast® Stream Events

The Shoutcast engine can generate events in any state. Events are sent asynchronously to the observer via an event dispatcher. The dispatcher is an active object that does a context jump before sending the event to the observer to prevent re-entrant code.

4. Usage

4.1 Building

Both the application UI and engine project were built using the S60 3rd Edition SDK. They have been tested with Codewarrior IDE and command line.

Building S60 Internet Radio Reference:

```
cd s60internetradio\group  
  
Bldmake bldfiles  
Abld reallyclean armv5  
Abld build armv5
```

Building the Shoutcast Engine plugin:

```
cd ShoutcastEngine\group  
  
Bldmake bldfiles  
Abld reallyclean armv5  
Abld build armv5
```

4.2 Signing and Installation

The S60 Internet Radio application uses certain APIs that requires additional capabilities such as Network Services. In order to install the application to the phone, the installation files must be signed with a valid certificate. Since the S60 Internet Radio application is provided as a reference implementation, the SIS files were signed with a Developer Certificate and private key. During installation, the user will be asked to grant the necessary capabilities. If the necessary capabilities are granted, the installation will proceed, otherwise, the installation is aborted.

To sign the application, obtain a certificate and key then use the `createsis` tool (available with the S60 3rd Edition SDK) from the command line to sign the SIS file. The `createsis` tool may also be used to generate a self-signed certificate to sign the SIS file.

See [3] for details about tools and certificate troubleshooting. For additional information about signing application, visit www.symbiansigned.com.

4.3 Playlist Management

When the S60 Internet Radio application is installed, an example playlist (`shoutcast.pls`) will be installed to the `\Shoutcast` directory of the installation drive. Additionally, the Shoutcast plugin will search for playlists in the `\Shoutcast` directory of every drive on the device. Found playlist(s) will be automatically parsed and added to the playlist menu.

4.4 Navigation

The UI uses standard S60 controls. The standard Options softkey offer a menu to select a station from the playlist, and a menu to toggle between local playback and internet radio mode. The center softkey toggles between play and pause while listening.

The following screenshots show the different views of the application.

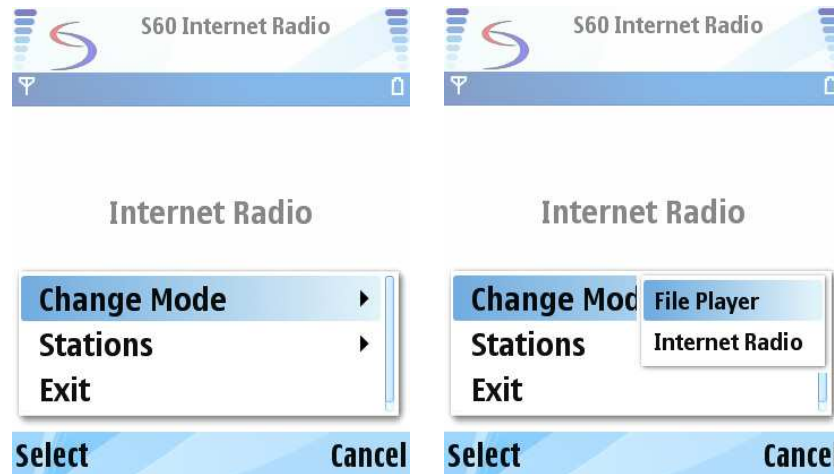


Figure 4 Application Menus

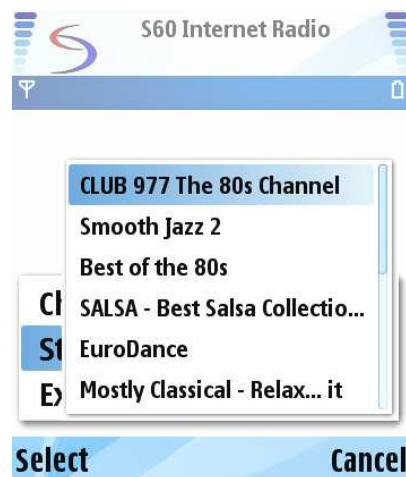


Figure 5 Playlist selection



Figure 6 Establishing connection



Figure 7 Streaming MP3 and Streaming AAC+



Figure 8 Rebuffering data

5. Miscellaneous

5.1 SVG Application Icon

The S60 Internet Radio icon is a scalable vector graphic icon created using the free SVG editing tool “Inkscape” available at www.inkscape.org. The image was saved in plain SVG format and converted to SVG-T format using the `svg2svgt` tool included in the S60 3rd Edition SDK. The SVG-T icon is built into the application by using a makefile started from the `bld.inf`. See the `S60InternetRadio bld.inf` file for an example of how this is done.

The icon building process has changed from the pre S60 3rd Edition process. See [2] for detail about porting and changes to the icon build process.