



INTELLIGERE FLS

Copyright (c) 2009
<http://www.gnstudio.com>

This file is part of Intelligere FLS

Intelligere FLS is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version.

Intelligere FLS is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with Intelligere SCS; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA.

@package Intelligere FLS
@version RC 1.0
@idea maker Giorgio Natili
@author Giorgio Natili <g.natili@gnstudio.com>

Contents

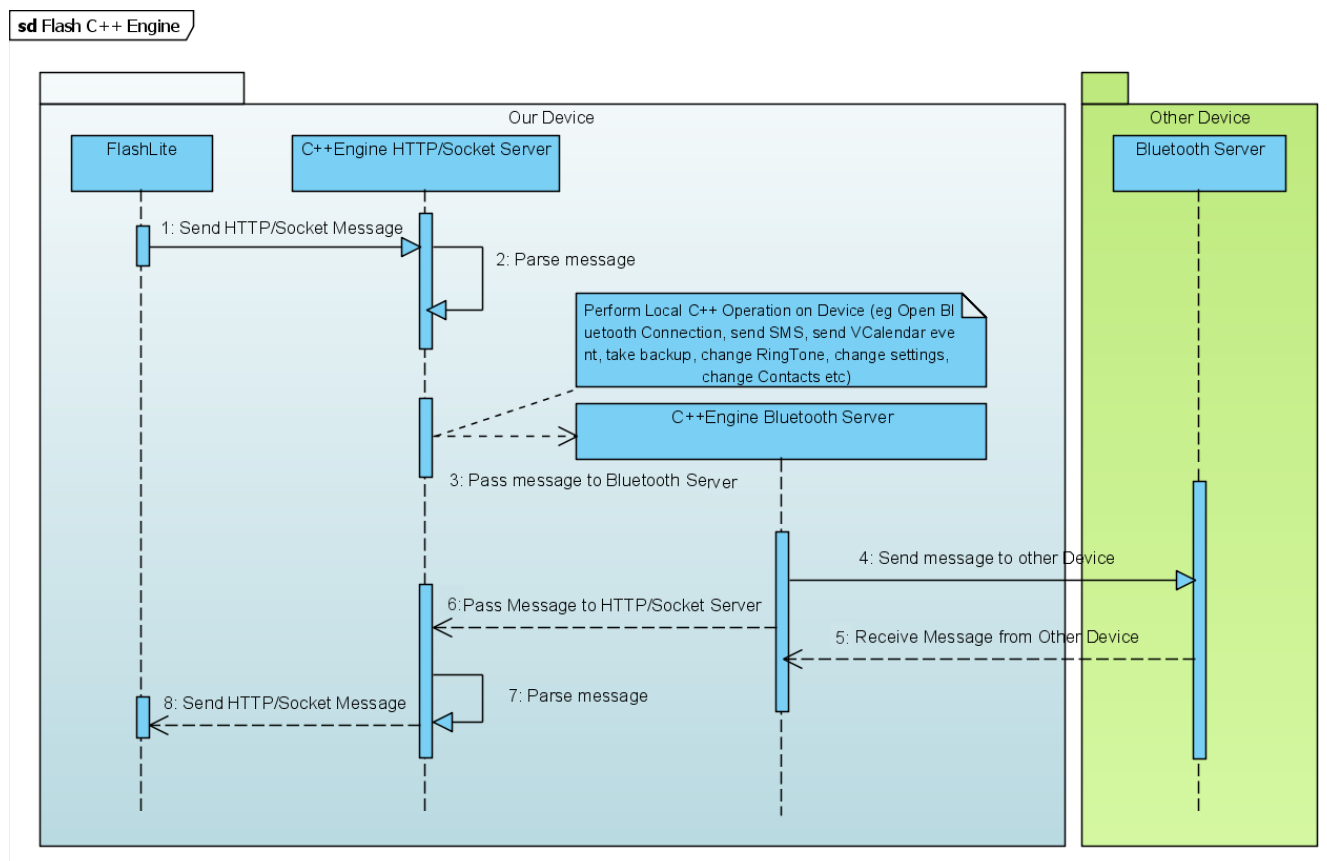
- Overview of the software
- Configure environment
- Windows Vista configuration
- Compile the projects
- Test the main demo in the emulator
- Demo description

Overview of the software

Intelligere Flash Lite Server (FLS) is a c++ open source component that gives to you the possibility to interact with the API of a symbian device from Flash Lite.

Using FLS you can extend the capabilities of your Flash Lite application making some simple HTTP calls sending commands and getting back data in this SWF file.

The workflow of the application can be summarized as following



The main features of FLS are

- Launch native video player in full screen mode (`launchVP`)
- Download a file from the NET (`download`)
- Check the percentage of the download in progress (`status`)
- Create the APN without opening the native selector (`createAP`) -> certificate is requested
- Read variables without opening the APN selector (`loadVariables`)
- Stop all the downloads (`stop`)
- Close the whole application (`exit`)
- Create a file (`create`)
- Delete a file (`delete`)

@package Intelligere FLS

@version RC 1.0

@idea maker Giorgio Natili

@author Giorgio Natili <g.natili@gnstudio.com>

- Write content into a file deleting everything before (`write`)
- Append content to a file (`append`)
- Clear all the contents of a file (`clear`)
- Rename a file (`rename`)
- Check if a file exist (`exist`)
- Get all the content of a directory (`dir`)
- Create a directory (`mkdir`)
- Delete a directory (`rmdir`)

In the RC2 version planned by the end of June 2009 the following feature will be added

- Get a picture from the camera (`mkPict`)

Configure your environment

In order to use the Intelligere Flash Lite Server you have first of all to download the following software and tools (read the following paragraph for the exact sequence of installing)

- 1) Carbide 1.3 <http://www.forum.nokia.com/info/sw.nokia.com/id/21c7fd04-1299-43d5-b7ee-3692e215a277.html>
- 2) Nokia SDK S60 3rd edition <http://www.forum.nokia.com/info/sw.nokia.com/id/4a7149a5-95a5-4726-913a-3c6f21eb65a5/S60-SDK-0616-3.0-mr.html> (choose the 3rd edition 262 mega bytes option on the right)
- 3) Active Perl 5.6.1.635 <http://downloads.activestate.com/ActivePerl/Windows/5.6/>
- 4) Extensions Plug-ins for S60 3rd Edition and S60 5th Edition SDKs http://www.forum.nokia.com/info/sw.nokia.com/id/48a93bd5-028a-4b3e-a0b1-148ff203b2b3/Extensions_plugin_S60_3rd_ed.exe.html (choose again the 3rd edition 3 mega bytes option on the right)

Following the next steps in order to install all these software

- 1) Launch the Carbide installer and follow the instruction on the screen selecting the "express edition" that is for free
- 2) Launch the Active Perl installer and follow the instruction on the screen and keeping everything by default
- 3) Unzip the S60 SDK and launch the setup keeping the default settings and if your are prompted install the CSL Arm Toolchain too, restart your computer if prompted
- 4) Run tge Extensions_plugin_S60_3rd_ed installer

Windows Vista configuration

If you are using Vista you have to take care about the following recommendation in order to make your environment working

- Install "Sun Java Runtime for Vista" ! not 1.5, but version 6 from Sun Java(tm) Runtime.
- On the Carbide 1.2 setup.exe right click "Run as Administrator".
- Install Active Perl.msi. Make sure you install version 5.6.1.638, not the latest 5.8xxx. Be sure to also add ";c:\Perl\bin" to the PATH environment variable .
You can change the PATH variable at My Computer > Properties > Advanced System Configuration > Environment Variables > System Variables. You'll be asked to give Administrator Permission during the

process.

- In the Vista startup bar an icon has been made to start Carbide 1.2 right click and select copy, go on to the desktop and right click and select paste
- Right click on the Carbide shortcut select the "Compatibility". Click the button "Run this program in compatibility mode for" Windows XP (Service Pack2)
- Click the box marked "Run this program as an Administrator", then click "Apply" then click ok
- Emulator needs to add the right JRE version (Epic32\tools\ecmt\config\config.properties -> epdt.java.version.start=1.4.1,1.4.2,1.5,5.0,1.6,6.0)

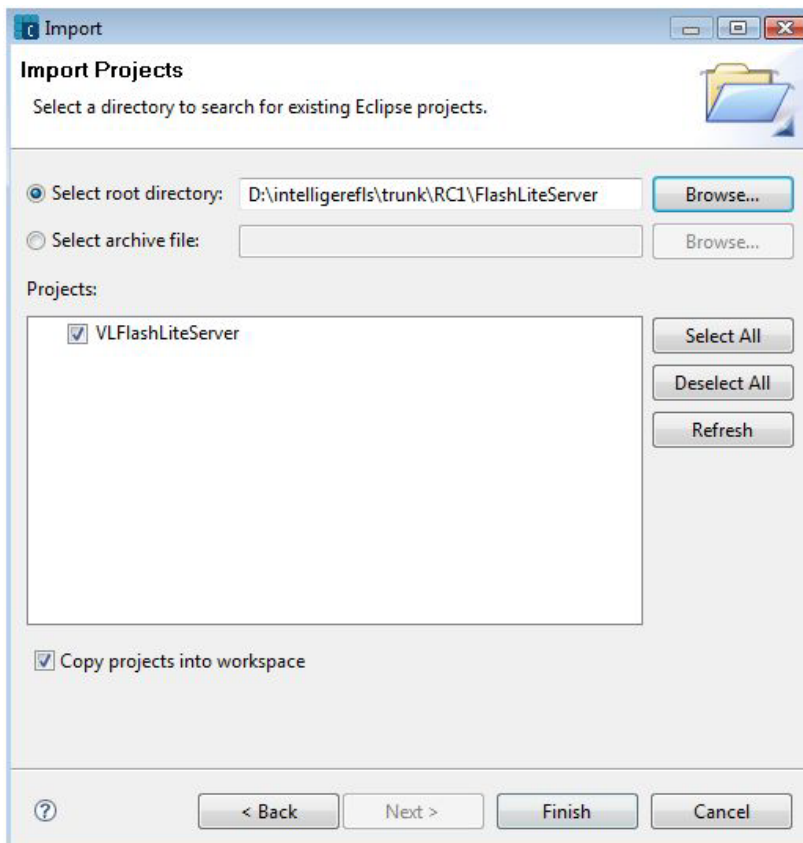
Compile the projects

Open Carbide and select your workspace and import the two projects you find in the RC1 folder of FLS source code as an existing project (right click and copliile).

probably you'll get the following errors

```
BLDMAKE ERROR: Directory "\Symbian\9.1\S60_3rd\EPOC32\" does not exist          fPlayer          Unknown
12364347180974
bldmake returned with exit value = 1          fPlayer          Unknown          12364347181595
```

This happens if you haven't put the workspace in the same drive in which the Symbian SDK is, so if you haven't do this create a new workspace in the C drive and import again the project in the new workspace making a copy in the workspace



In order to be able to see the application in the emulator you have to go under your Symbian SDK installation folder and copy in DRIVE:\Symbian\9.1\S60_3rd\Epic32\winscw\c\Data\Others\trusted the DemoTest.swf file, actually each time you want to create an application with FLS you have to out the SWF file you are using in



this folder too.

Test the main demo

In order to run the demo with the Carbide emulator you can simply run the fPlayer project you find in the source code, in the emulator you can test the following features

- Launch native video player in full screen mode (`launchVP`)
- Download a file from the NET (`download`)
- Check the percentage of the download in progress (`status`)
- Create the APN without opening the native selector (`createAP`) -> certificate is requested
- Read variables without opening the APN selector (`loadVariables`)
- Stop all the downloads (`stop`)
- Close the whole application (`exit`)
- Create a file (`create`)
- Delete a file (`delete`)
- Write content into a file deleting everything before (`write`)
- Append content to a file (`append`)
- Clear all the contents of a file (`clear`)
- Rename a file (`rename`)
- Check if a file exist (`exist`)
- Get all the content of a directory (`dir`)
- Create a directory (`mkdir`)
- Delete a directory (`rmdir`)

Remember that each time you use this project you are using the FLS (FlashLiteServer) you use this project as a dll in the others project, so if you do changes you have to build the FLS, build your project and then run the project in the emulator.

In the emulator the dll is automatically found, when you are delivering the sisi with the complete application you have to copy the LHS.sisx and LHS.sis file in the `Components` folder of your project.

In the .pkg file of your project you say to the application to install the dll if it's missing

```
IF NOT package(0xA00045F5)
; install LHS.sisx only if not already installed
@"..\Components\LHS.sisx", (0xA00045F5)
ENDIF
```

If you want to change the port on which the server is listening for the commands open the `FlashPlayerView` file you find in the `src` folder and change the argument you use in as a port

```
//TODO Set here the port which the server listens to
iEngine->StartLHSL(2002);
```

In the `Constants` file you find in the `inc` folder you can set which SWF file to use

```
_LIT(KLitSwfFileToLaunch, "\\data\\others\\trusted\\HelloWorld.swf");
```

If you want to change the name and description of the application you have to open the `FPlayer_loc.rls` file under the `data` folder and change the value of the constants

```
rls_string STRING_short_caption "FLS"
```

```
@package Intelligere FLS
@version RC 1.0
@idea maker Giorgio Natili
@author Giorgio Natili <g.natili@gnstudio.com>
```

```
rls_string STRING_r_long_caption "FLS demo"
```

The other demo are a full screen video player done with c++ and Flash Lite and the Bluetooth connectivity demo.

The building blocks of the HTTP communication can be summarized as following

