



Communications
Research Centre
Canada

An Agency of
Industry Canada

Centre de recherches
sur les communications
Canada

Un organisme
d'Industrie Canada

Software Communications Architecture v2.2 Reference Implementation Project

**SDRF Forum Contract
SDRF-04-A-0002-V0.00**

README

June 2007



TABLE OF CONTENTS

1. Installation Under Linux Fedora Core 3	1
1.1 CRC SCARI-Open Environment Setup.....	1
Configuration	2
1.2 Audio Setup for Desktop Environment.....	2
1.3 Java Orb Initialization	2
1.4 Parallel Port Configuration	2
2. SCARI-OPEN directory structure	3
3. Building SCARI-OPEN Node and Examples.....	5



1. INSTALLATION UNDER LINUX FEDORA CORE 3

1.1 CRC SCARI-Open Environment Setup

Below are the installation steps for CRC SCARI-OPEN under Linux Fedora Core 3 using the bash shell.

- Make sure Java SDK 1.4.2 has been installed
- Select a location to install SCARI-OPEN (e.g. \$HOME):
 `cd $HOME`
- Extract all the packaged SCARI-OPEN files:

```
tar xzvf scari-OpenRel1.1.tgz
```

- Set SCARI-OPEN Development Environment Variables

Update your user environment file ~/.bashrc as follows:

- `export JAVA_HOME={JAVA_INSTALLATION_HOME}`
- `export PATH=$PATH:$JAVA_HOME/bin`
- `export SCA_HOME={INSTALL_DIR}/scari-Open`
- `export CLASSPATH=$CLASSPATH:$SCA_HOME/classes`
- `export CLASSPATH=$CLASSPATH:$SCA_HOME/lib/openorb_event-1.3.0.patched.jar`
- `export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$SCA_HOME/lib`

SCARI-OPEN is already built so you don't have to compile the source code before using it. The C library files of the two applications provided as examples are built for Fedora Core 3.

If you are using another Linux distribution and/or version, we suggest that you read the section ***Building SCARI-OPEN Node and Application Examples*** and rebuild the libraries for this particular version.

CONFIGURATION

1.2 Audio Setup for Desktop Environment

VERY IMPORTANT: To prevent audio contention when running the application examples, the sound server for the Desktop Environment should be deactivated. To deactivate the sound server:


- 1 - At the command prompt, type : **kcmshell arts**
- 2 - Check off the **Enable the sound system** box .
- 3 - Click **Apply**. The sound server shuts down.

1.3 Java Orb Initialization

In order for the scenarios to execute properly, you must change the default TCP port for the ORB.

- In the home directory, create a file named orb.properties
- With a text editor, add the following line to orb.properties:
org.omg.CORBA.ORBInitialPort=1050

1.4 Parallel Port Configuration

	<p>CAUTION</p> <p>The commands below are only required if you have the RF transceiver box end. And you intend to run the analog FM Application.</p>
---	--

The SCARI-OPEN RFDevice needs to have access to the parallel port to control the RF Transceiver Box. In Linux, the access to the parallel port is only allowed by super-user. If you happened to run as root then you can ignore the following commands. Any regular user has to give super-user privilege to the sendDataToPort program which is used by the RFDevice to communicate with the RF Transceiver Box. The source code for the sendDataToPort program is located in {INSTALL_DIR}/scari-Open/demosources/devices/ for your perusal.

- From any shell become super-user (i.e. su)
cd \$SCA_HOME/demosources/Node1/profile
chown root sendDataToPort
chmod u+s sendDataToPort



2. SCARI-OPEN DIRECTORY STRUCTURE

.	All the scenario scripts and SCARI-OPEN Makefile.
Classes	All compiled Java classes.
IDL	All IDL with Makefile to generate java source and compile.
Scasources	Java sources code for all SCA implementation
Generated	All IDL generated code. There is no Makefile (use the one provided in the directory <i>IDL</i>).
SCA/CFImpl	Java files which implement only the CF functions that will be reused in any CF development. Code that represents a proprietary solution to an open issue in the SCA CF should be located in the SCA/CFextImpl. Most of the object implementations in this directory use the classes located in the “util/domainprofile” as they are also used by the GUIs and others. This directory also contains a Makefile.
SCA/CFextImpl	Java files which extend the basic generic CF with proprietary solutions to the SCA CF as specified in the SCA specs. For example, this is where the implementation of the DomainManager that provides special services for the ApplicationFactory is located. The IDL interface definition for the extended CF objects should be put in the \$SCA_HOME/IDL/CFext.IDL (if so, the Makefile will generate everything automatically). This directory also contains a Makefile.
SCA/LogServicesImpl	Java files which implement the LogServices interface with Makefile to compile them.
util/domainprofile	Java files which implement all the XML parsing. This is where the Domain Profile related code for the DeviceManager, DomainManager, and ApplicationFactory is located. A Makefile is also included.
util/install	Java files which implement Application installation.
Demosources	Java source code required for the platform and application demonstration.
Devices	Java files which implement the demonstration platform CF device-related functions. For example, the SCA does not specify how a DeviceManager gets a reference to its Log. However, for the demonstration platform, this has been implemented using the port concept which requires a connection in the DCD file. This directory contains this specific implementation of the DeviceManager.
Services	Java files which implement node specific services (i.e. Log). For the moment, this directory only contains a Log CORBA server to allow the Log to be started as a service from the DCD file.
Components	This serves the same purposes as the demosources/devices and demosource/services directories but for the other SCA components. For example, this directory contains an implementation of the DomainManager which gets a reference to its log through a Port.
applications	Java files which implement application specific code. This where an AssemblyController should be located but support code may also be put here.



	However, the Resources of a waveform application are located in a separate directory since they should made as reusable as possible.
resources/audioeffect	Java files which implement audio effect Resources.
resources/fmreceiver	Java files which implement FM Receiver Resources.
resources/fmtransmitter	Java files which implement FM Transmitter Resources.
Node1/profile	XML, java jar files and libraries that represent Node1 (DCD).
Waveforms/AudioEffect0	XML, java jar files and libraries that represent AudioEffect0 application (SAD). Requires single Node (Node1).
Waveforms/AnalogFM	XML, java jar files and libraries that represent AnalogFM application (SAD). Requires single Node (Node1).
Doc	Java documentation of all java source code contained in SCARI-OPEN



3. BUILDING SCARI-OPEN NODE AND EXAMPLES

In order for the Makefile to run properly, make sure you have performed the SCARI-OPEN environment setup properly as described in section 1.1

To start with a fresh build do the following in \$SCA_HOME

- make clean
- make

If the source files are modified, type the following in \$SCA_HOME

- make

Makefiles are also provided in all source-code directories

- scasources/SCA/CFImpl
- scasources/SCA/LogServicesImpl
- scasources/util
- scasources/util/domainprofile
- scasources/util/install
- demosources/devices
- demosources/services
- demosources/components
- demosources/applications
- demosources/resources/audioeffect
- scenarios/SCA/CFImpl
- scenarios/SCA/LogServicesImpl

To package SCARI-OPEN Node and Application Examples from the \$SCA_HOME directory:

- cd \$SCA_HOME
- make -C lib