





	User Guide	
ERICSSON 🔰	Open	2 (29)
No.	Date Rev	
ERA/LU/P-01:0159 Uen	2002-03-18 G	

# Contents

1	Introduction	4
1.1	Revision information	4
2	What is delivered	5
3	Simulator	6
3.1	Architecture	6
3.2	Capabilities	7
3.3	Known Bugs and Limitations	9
4	System Requirements	10
5	Installation Instructions	10
5.1	Installing With the Installer	10
5.2	Installing Without the Installer	11
6	The Simulator	11
6.1	General	11
6.2	Starting the Simulator	12
6.3	The Chatboard	16
6.4	Command Line Interface	17
7	Applications	18
7.1	Call Barring	18
7.2	Web Dial	21
8	3 <sup>rd</sup> Party applications	23
8.1	Introduction	23
8.2	Defining application IDs	23
8.3	Defining announcements	25
9	References	28
10	Trouble Shooting List	29

ERICSSON	Open		3 (29)
No.	Date	Rev	
ERA/LU/P-01:0159 Uen	2002-03-18	G	

#### 1 AGREEMENT

This Agreement between Ericsson Radio System and any party using ('User') the Ericsson OSA/Parlay Simulator and related documentation ('Material') shall be effective upon the usage of the Ericsson OSA/Parlay simulator.

#### 2 OWNERSHIP Provider retains full ownership to the Material at all times.

3 LICENSE

Ericsson hereby grants to the User a royalty free and non-transferable license to use the Material for educational and research purposes only. Any kind of commercial use of the Material is strictly prohibited. The User is entitled to make copies of the Material, irrespective of the media on which it is delivered, on the condition that the User retains in every copy such proprietary marks and symbols identical to those in the original Material distributed by Ericsson.

#### 4 DISTRIBUTION

The Material may not be distributed to third parties, except when used in education.

5 DISCLAIMER OF WARRANTY

COVERED MATERIAL IS PROVIDED UNDER THIS LICENSE ON AN 'AS IS' BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED MATERIAL IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGING. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE ON THE COVERED MATERIAL IS WITH THE USER. SHOULD ANY COVERED MATERIAL PROVE DEFECTIVE IN ANY RESPECT, THE USER (NOT THE PROVIDER) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED MATERIAL IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6 DEFAULT

The User, hereby, acknowledges that it is strictly forbidden to use the Material for any commercial purpose whatsoever. The Provider will claim damages for any breach of this agreement.

	User Guide		
ERICSSON 🔰	Open		4 (29)
No.	Date	Rev	
ERA/LU/P-01:0159 Uen	2002-03-18	G	

**INTRODUCTION** 

1

This User Guide has been written for the Ericsson OSA/Parlay Simulator, version 0.7 Beta.

OSA/Parlay is a new type of open API (Application Programming Interface) for todays and next generation's communication networks. It provides an additional layer of abstraction for the application developers. It enables telecom operators and service providers to offer the same services for all existing underlying networks – the fixed network, all mobile networks, IP networks – without adapting the application to network specific protocols.

With OSA/Parlay, developers do not have to worry about the operators' network protocols anymore. The OSA/Parlay open API is an exciting opportunity for software developers and application server providers since it represents a huge market potential. The OSA/Parlay API has been defined by the Parlay Group, a non-profit organisation supported by the major telecom suppliers, the IT industry and telecom network operators. The OSA/Parlay API is open to everybody.

In order to enable developers to test their application Ericsson developed the Ericsson OSA/Parlay Simulator. This tool simulates the Jambala Service Capability Server (J-SCS) and a telephony network. It is accompanied by a number of test applications.

The purpose of the simulator is to test applications, which are developed by service providers or other interested parties without the need for any network specific hardware. In this way, Ericsson is helping developers by speeding up the testing process. The simulator is also a convenient tool to be used in demonstrations for OSA/Parlay.

# 1.1 REVISION INFORMATION

G	2002-03-18	Updated for Ericsson OSA/Parlay Simulator version 0.7 Beta
F	2001-12-21	Updated for Ericsson OSA/Parlay Simulator version 0.6 Beta
Е	2001-11-19	Updated for Ericsson OSA/Parlay Simulator version 0.5 Beta
D	2001-09-27	Updated for Ericsson OSA/Parlay Simulator version 0.4 Beta
С	2001-08-29	Updated for Ericsson OSA/Parlay Simulator version 0.3 Beta (removed restrictions on releasing calls and starting multiple applications)
В	2001-07-11	Updated for Ericsson OSA/Parlay Simulator version 0.2 Beta
А	2001-06-05	First version

	User Guide	
ERICSSON 🔰	Open	5 (29)
No.	Date Rev	
ERA/LU/P-01:0159 Uen	2002-03-18 G	

Further information on the different releases can be found in the help menu of the simulator (select 'Release Notes').

# 2 <u>WHAT IS DELIVERED</u>

The Ericsson OSA/Parlay Simulator package includes the following items, which are categorised by the directory names.

#### **Root installation directory**

- User Guide.
- Release Notes.
- Run scripts for the Simulator.

#### **Applications**

- Call Barring application including source code and sequence diagrams.
- Web Dial application including source code.
- Both applications have executable scripts to create javadocs.

#### Config

- Configuration files for the simulator.
- Configuration files with terminal settings.
- Configuration file with simulator network settings.

# IDL

- The IDL for generating the Parlay API source files.
- Note: Javadocs for the Parlay API can be downloaded as a separate package from Mobility world (<u>http://www.ericsson.com/mobilityworld/sub/open/solutions/parlay/index.ht</u> ml?PU=parlay).

#### Jars

- Jar file with the Parlay API classes (Generated with the IDL file)
- Jar file with the Simulator classes

#### Images

• Images for the simulator and the terminals.

# Sounds

- Announcements
- Ringtones and busytones for the terminals

#### Tools

# ERICSSON

	Open		6 (29)
No.	Date	Rev	
ERA/LU/P-01:0159 Uen	2002-03-18	G	

- To make the Simulator running without a network connection in case of start-up problems.
- Windows 98 patch that will be installed automatically.

# 3 <u>SIMULATOR</u>

# 3.1 ARCHITECTURE

The diagram below in figure 3.1-1 depicts the architecture of the Ericsson OSA/Parlay Simulator



# Figure 3.1-1

The simulator package consists of the following components:

1. Name server

The name server enables communication between the NetworkSim, the simulated Jambala service capability server (J-SCS) and the application(s).

2. J-SCS lite

The J-SCS lite is a simulated version of the real J-SCS. The J-SCS lite contains the following capabilities:

- Framework (FW), Used for authentication and discovery of services as defined by the OSA/Parlay specification.
- Generic Call Control (CC), Enables the generic Call Control services, such as handling call events, routing etc.
- Generic User Interaction (UI). Enables the generic User Interaction services, such as digit reception, announcement handling etc.

# 3. NetworkSim

The NetworkSim simulates a basic telecommunications network, in which mobile terminals can be defined in a graphic user interface (GUI).

ERICSSON 📕	Open		7 (29)
No.	Date	Rev	
ERA/LU/P-01:0159 Uen	2002-03-18	G	

# 4. Application(s)

Applications as such are not part of the simulator, but test-applications are included in the simulator package in order to test and demonstrate the use of the simulator. It is possible to test your own applications on the simulator (see also Chapter 7).

# 3.2 CAPABILITIES

The J-SCS component implements a subset of both the Generic Call Control Service Interface and the Generic User Interaction service as defined by OSA/Parlay (see Reference [1]). This subset excludes for example Multi-Party Call Control, Multi-Media Call Control and Conference Call Control.

The main features of the **Framework** service (based on API towards applications) that are implemented are:

- IpInitial
  - initiateAuthentication
- IpAuthentication
  - requestAccess
- IpAPILevelAuthentication
  - selectAuthenticationMethod
  - authenticate
  - abortAuthentication
- IpAppAPILevelAuthentication
  - authenticate
  - abortAuthentication
  - signServiceAgreement
  - terminateServiceAgreement
  - terminateAccess
- IpAccess
  - obtainInterface
  - obtainInterfaceWithCallback
  - selectService
  - signServiceAgreement
  - terminateServiceAgreement
  - endAccess
- IpServiceDiscovery
  - listServiceTypes
  - describeServiceType
  - discoverService

User Guide	
------------	--

ERICSSON 📕	Open		8 (29)
No.	Date	Rev	
ERA/LU/P-01:0159 Uen	2002-03-18	G	

• listSubscribedServices

The main features of the **Generic Call Control** service that are implemented are:

- IpCallControllManager
  - createCall
  - enableCallNotification
  - changeCallNotification
  - disableCallNotification
  - getCriteria
- IpService
  - setCallback
  - setCallbackWithSessionID
- IpAppCallControllManager
  - callAborted
  - callEventNotify
- IpCall
  - routeReq
  - release
  - deassign
  - getCallInfoReq
  - setCallChargePlan
- IpAppCall
  - routeReq
  - getCallInfoRes (sent at call disconnection when an intermediate report was requested via getCallInfoReq. The report does not contain a realistic TpCallReleaseCause and no resource connection time)
  - getCallInfoErr (sent immediately when a final report was requested via getCallInfoReq)
  - callEnded (the resulting TpCallEndedReport does not contain a realistic TpCallReleaseCause)
  - callFaultDetected

The following features of the Generic Call Control service are not supported:

- Load control
- Advice of charge (setAdviceOfCharge)
- Collecting further digits (getMoreDialledDigitsReq)

User	Guide
000.	0 4140

ERICSSON	Open		9 (29)
No.	Date	Rev	
ERA/LU/P-01:0159 Uen	2002-03-18	G	

• Call supervision (superviseCallReq)

The main features of the **Generic User Interaction** service that are implemented are:

- IpUIManager
  - createUICall
- IpService
  - setCallback
  - setCallbackWithSessionID
- IpUICall
  - sendInfoReq
  - sendInfoAndCollectReq<sup>1</sup>
  - release
  - abortActionReq
- IpAppUICall
  - sendInfoRes
  - sendInfoErr
  - sendInfoAndCollectRes
  - sendInfoAndCollectErr)
  - abortActionRes
  - abortActionErr (does not occur with the simulator)

The following features of the Generic User Interaction service are *not* supported:

- Non call-related user interaction (createUI)
- User Interaction notifications (enableUINotification, disableUINotification, userInteractionEventNotify)
- Recording of a message (recordMessageReq)

# 3.3 KNOWN BUGS AND LIMITATIONS

The following bugs are known at the time of release:

• The Simulator might give errors in case two applications are having triggers on the same phone numbers and events.

<sup>&</sup>lt;sup>1</sup> Input is submitted by pressing the Yes-button (on the softphone keypad) or by appending a hash (#) to the input.

ERICSSON	Open	10 (29)	
No.	Date Rev		
ERA/LU/P-01:0159 Uen	2002-03-18 G		

- When an application is started for the second time after a terminateServiceAgreement it might happen that the simulator closes itself because of an internal error. Example: Call barring is started and closed by pressing on 'terminate'. After that Call Barring is started again and when a subscribed number is called or the application is terminated again the problem might occur.
- The Simulator does not support follow-on calls; when the terminating leg is disconnected, the origating leg is disconnected as well.

#### SYSTEM REQUIREMENTS

4

The minimum system requirements are specified below:

- Pentium II or higher (for PC systems)
- 64 Mb internal memory (128 Mb recommended)
- 50 Mb free disk space for downloading and installing.
- Soundcard (optional, e.g. for ringtones)
- Microphone (optional, for recording of announcements)
- Network connection<sup>2</sup>
- Windows 98, 2000, ME, NT4.0, Linux, UNIX
- Java 2 Standard Edition version 1.3.1<sup>3</sup>

# 5 INSTALLATION INSTRUCTIONS

# 5.1 INSTALLING WITH THE INSTALLER

The installer will automatically install the simulator and the test application on the target system.

5.1.1 Downloading and using the installer

Note: it is recommended to uninstall any existing installations of the simulator before installing a new release.

- Go to the mobility world internet site,
   <u>http://www.ericsson.com/mobilityworld/open/solutions/parlay/index.html</u>
- Select 'Tools & Enablers'
- Download *install.exe* to your hard disk
- Run the 'install' program and follow the steps.

 $<sup>^{2}</sup>$  For Windows 98 and later versions there is a patch available that makes the network connection superfluous.

<sup>&</sup>lt;sup>3</sup> Not required when using the installer for the simulator because it will be automatically installed.

		User Guide		
ERICSSON 🔰		Open		11 (29)
		Date	Rev	
ERA/LU/P-01:0159 Uen		2002-03-18	G	
5.2	INSTALLING WITHOUT	THE INSTALLE	R	
	It is also possible to instal for people with Java expe	l the simulator rience.	manually. This is	recommended only
5.2.1	Download and install JAV	<u>A 2 SDK</u>		
	Download and install J	lava 2 SDK 1.3	.1 from <u>http://java.</u>	<u>sun.com/j2se/1.3/</u> .
5.2.2	Download and install the E	Ericsson OSA/I	Parlay simulator	
	<ul> <li>Go to the mobility worl</li> <li>Select 'Tools &amp; Enable</li> <li>Download Simulator.z.</li> <li>Extract the files in Sim</li> <li>For UNIX only: After u executable (chmod -1</li> </ul>	d internet site ers' <i>ip</i> to your hard <i>nulator.zip</i> to a nzipping it is no R a+x *.sh).	(see paragraph 5. disk dedicated folder ( ecessary to make	1.1) e.g. C:\Simulator). the shell files
	For last-minute informatio installation directory.	n, see the file F	RELEASE_NOTE	S.TXT in the
5.2.3	Configure PATH and CLA	SSPATH settir	<u>igs</u>	
	Two files will be created in and javaenv.sh (UNIX). The PATH and CLASSPATH t	n the installation nese files are υ ο the correct v	n directory: JavaE sed to set the envalues.	inv.bat (Windows) vironment variables
	To set the environment va platform and edit the follow	riables, open t wing line:	he script file for th	e corresponding
	set JAVA_1	HOME_DIR=no	t_set_yet	
	Change the value 'not_se Development environment the file.	et_yet' into th t (JDK 1.3) is i	ne correct director nstalled (e.g. c:\jo	y where the Java k1.3.x) and save
6	THE SIMULATOR			
6.1	GENERAL			
	This paragraph shows the and applications. If the Sir installer, this has been do directly by shortcuts in the When you install the Simu environment or host settin	e commands ar mulator has be ne already. In t start menu (W lator manually gs then the tex	nd settings for staten en installed by us hat case, you car /indows). or when you wan at below gives mo	rting the simulator ing the automatic n start the simulator t to change re information.
	In the installation folder, th	nere are a num	ber of batch files.	These are:
	<ul> <li>JavaEnv.bat/ javaenv.s</li> <li>Files used by the start- variables (Windows and</li> </ul>	sh ·up scripts for s id UNIX respec	setting the Java e stively)	nvironment

ERICSSON 💋	Open		12 (29)	
No.	Date	Rev		
ERA/LU/P-01:0159 Uen	2002-03-18	G		

run.bat/run.sh
 Files used for running the simulator (Windows and UNIX respectively).

User Guide

• run\_without\_parlay.bat/run\_without\_parlay.sh Files used for running the NetworkSim without any parlay interaction.

#### Note:

The simulator and applications can be run on separated machines. The application run scripts support the following two command line arguments to determine the location of the name server (see also the individual script files):

- ORBInitialHost the IP-address of the machine where the ORB is running (default: the local machine).
- ORBInitialPort The corresponding port number (default: 1500)

The ORB settings for the Simulator can be found in the file 'config/simulator.cfg'

# 6.2 STARTING THE SIMULATOR

Executing the 'Run.bat' (Windows) or the 'run.sh' (UNIX) script in the installation directory will start the simulator. When the installer is used, a shortcut to the start-up script will be created in the Windows 'Start' menu, which can be used instead. The following steps are automatically performed.

First a name server is started. After that the J-SCS lite and network Simulator is started. The output window will look like Figure 6-1.



Figure 6-1: Start-up trace window for the Simulator

	User Guide	
ERICSSON 💈	Open	13 (29)
No.	Date Rev	
ERA/LU/P-01:0159 Uen	2002-03-18 G	

#### Next, the NetworkGUI (Figure 6-2) will appear on the desktop.

	📒 Er	icsson OSA,	/Parlay Sin	nulator	- UX
	Elle	Terminal	Mindow	Help	
	22 🛤				
About				2	4
			Welco	me to the Ericsson OSA/Parlay Simulator Version M.m.(asild number n)	
1		Compa	the to work	eyyyo- emm>-edd> affbe inint OSA I Pariav API group of March: 3GPP T9 29 199 (2001-03)	
		Compa	able to morn	on the joint operation of group of march, operating as the (approxid)	
100		Eri	csson M	obility World is an enhanced program to support 3rd	
202		party	y develop	ers. Please visit http://www.ericsson.com/mobilityworld	
	' 1	o learn mo	vre about	OSA/Parlay or to become an "Ericsson Associate" member.	
		-			
		By us	sing this sin	ulator you accept the terms and conditions stated in the user guide.	
				ОК	

Figure 6-2: Start-up screen of the Simulator

To be able to use the simulator, first click away the 'about'-window. To test the Network Simulator, different terminals can be added.

# 6.2.1 <u>Functionality of the NetworkGUI</u>

Create a terminal

- Or click '<u>N</u>ew... Insert' on the menu
- Press the terminal-icon:

	User Guide	
ERICSSON	Open	14 (29)
No.	Date Rev	
ERA/LU/P-01:0159 Uen	2002-03-18 G	

Llear Guida

The following window (Figure 6-3) appears:

📁 Terminal properties		×
Network address Address String:	123	-
Subaddress string: Address name:		
Address plan:	E.164	
Address presentation:	P_ADDRESS_PRESENTATION_ALLOWED	
Address screening:	P_ADDRESS_SCREENING_USER_NOT_VERIFIED	
Appearance Window title:	John	-
Terminal type:	T68 💌	
	<u>O</u> K <u>C</u> ancel	1

#### Figure 6-3: Terminal properties window

For the new terminal, a number of properties can be set. These properties are divided in two groups:

Network Address.
 The meaning of these properties can be looked up in the Parlay specifications.

*Note:* The address presentation field is not used in the Simulator.

2. Appearance.

The 'Window Title' property defines the name that is shown on the terminal window. When this field is empty, the 'Address String' will be used as title.

With the 'Terminal type' property, you can choose a different telephone model (appearance only).

After entering an address string (telephone number) for the terminal and pressing the 'OK-button, the terminal will appear in the NetworkGUI (see for example Figure 6-4). It is also possible to choose a different type of terminal from the list and to specify a title for the terminal window.

	User Guide		
ERICSSON	Open		15 (29)
No.	Date	Rev	
ERA/LU/P-01:0159 Uen	2002-03-18	G	

The following terminals are available in the network simulator.



6.2.2 <u>Test the NetworkGUI</u>

- Create two terminals (for example John and David, John has the number 123 and David has the number 456)
- Initiate a call from terminal John to terminal David by entering the number 456 via the keypad (in the phone-window) and clicking the 'Yes'-button of the originating terminal (Figure 6-4).

	User Guide	User Guide		
ERICSSON	Open		16 (29)	
No.	Date F	Rev		
ERA/LU/P-01:0159 Uen	2002-03-18	G		



Figure 6-4: Example of created terminals

- Click the 'Yes'-button of the terminating terminal to answer the call. The call is now established.
- Click the 'No'-button of either terminal to release the call.

#### 6.3 THE CHATBOARD

Instead of using the keypad, it is also possible to use the Ericsson Chatboard<sup>™</sup> as an input device for the terminals.

The Ericsson Chatboard<sup>™</sup> can be activated and deactivated by pressing the right mouse button in the terminal window and than selecting the 'Chatboard' option from the pop-up menu (see Figure 6-5).

	User Guide	
ERICSSON	Open	17 (29)
No.	Date Rev	
ERA/LU/P-01:0159 Uen	2002-03-18 G	

Lloor Cuido



Figure 6-5: Example of how to show the chatboard

# 6.4 COMMAND LINE INTERFACE

To give better support for testing applications and specially for automated testing, the Simulator has a command line interface.

With this interface the properties of the simulator can be overwritten at the start-up. To list all possible commands one can type the –help command while starting the simulator.

Example in run.bat

java -cp jars\simulator.jar;jars\parlay3gpp.jar \_Simulator -help.

The following commands are supported:

-help	Shows this screen
-configFile value	Set the location of the config file
-noAboutBox	No about box at start up
-noNameServer	Simulator starts without nameserver

ERICSSON 📕	Open		18 (29)
No.	Date	Rev	
ERA/LU/P-01:0159 Uen	2002-03-18	G	

-ORBInitialPort value	Set value of the ORB port
-ORBInitialHost value	Set value of the ORB host
-cosNamingPort value	Set value of the cosnaming responder port
-cosNamingHost value	Set value of the cosnaming responder host
-EPCTfiles value	Set used epct files (separator ',' no spaces)
-staticProcs value	Set used static procs. (separator ',' no spaces)
-releaseTimeout value	Timeout (ms) for release of a call after the call has ended

# 7 <u>APPLICATIONS</u>

The Call Barring application and the Web Dial application are delivered with the simulator as test/example applications. These applications are described in the following chapters.

# 7.1 CALL BARRING

## 7.1.1 <u>Overview</u>

The Call Barring application is delivered together with the Ericsson OSA/Parlay simulator to serve as an example of an application that makes use of the Generic User Interaction Service.

The Call Barring Application enables a user to prevent people from setting up a call to his/her number, unless they have the correct PIN code. The users of this application are registered in a user database. Only users who have registered can make use of the application.

The application consist of two parts:

- Management part: this enables the user to change his/her PIN code.
- Traffic Part: this part performs the actual Call Barring service.

# 7.1.1.1 Management part

The following sequence describes a call scenario to the management part:

- Set up a call to the service number (default '667').
- The user is requested to enter his/her current PIN Code<sup>4</sup>.
- If the correct PIN code is entered, the user is requested to enter the new PIN code.
- The newly entered PIN code is repeated and the call is released.
- If an incorrect PIN code is entered, the call is released immediately.

If a user was not registered in the Call Barring application yet, his/her number will be added to the User database.

<sup>&</sup>lt;sup>5</sup> Input is submitted by pressing the Yes-button (on the softphone keypad) or by appending a hash (#) to the input.

|--|

ERICSSON	Open		19 (29)
No.	Date	Rev	
ERA/LU/P-01:0159 Uen	2002-03-18	G	

# 7.1.1.2 Traffic part

The following sequence describes a call scenario to the traffic part:

- Set up a call to a user of the Call Barring Application.
- The user is requested to enter the PIN code for the destination number that was dialled.
- If the correct PIN is entered, a call will be set up to the specified destination number.
- If an incorrect PIN is entered, the user is requested to re-enter the PIN code, after three incorrect attempts the call is released.

Calls to destination numbers, which are not registered in the application, are handled normally.

## 7.1.2 Installing and running the application

The Call Barring application is delivered together with the Ericsson OSA/Parlay simulator. It is installed in the application directory:

#### <installationdir>\applications\CallBarring

Running the 'run.bat' (Windows) or the 'run.sh' (UNIX) scripts in the application directory will start the application (when using the installer a shortcut to the start-up script is created in the Windows 'Start' menu, which can be used instead). Before starting, make sure the simulator is running.

When the application is ready to process incoming calls the application log window looks like Figure 7-1.



Figure 7-1: Call Barring application log.

ERICSSON 📕	Open	20 (29)
No.	Date Rev	
ERA/LU/P-01:0159 Uen	2002-03-18 G	

The initial user database is simulated using a text file named '*users.ini*'. Users can be added manually in this file. This file is loaded on start-up and not used during runtime, any changes should be made before the application is started.

The file 'config.ini' contains default values for constants used throughout the application, if necessary it is possible to adapt these values before starting the application.

#### 7.1.3 <u>Source code</u>

The Java source files for this application are included and are stored in the subdirectory:

<installationdir>\simulator\applications\CallBarring\src

The application has the following package structure:

- com.ericsson.cba.common common application classes and common constant classes
- com.ericsson.cba.management management specific classes and management specific constant classes
- com.ericsson.cba.management.states state classes defining the different management input states
- com.ericsson.cba.traffic traffic specific classes and traffic specific constant classes
- *com.ericsson.cba.traffic.states* state classes defining the different traffic input states
- com.ericsson.cba.users classes defining the application users and user database

Note: Other packages can be found under:

- com.ericsson.datastructures
- com.ericsson.configuration
- com.ericsson.parlay
- com.ericsson.tracedebug

These packages contain standard classes with utilities that can be used by applications.

Details can be found in the Java documentation that is delivered together with the source code. This documentation can be accessed by opening the *'index.html'* file in the *'<installationdir>\Simulator\CallBarring\javadocs'* directory.

The full source code of the Call Barring application can be compiled by running the '*Compile.bat*' (Windows) or Compile.sh (UNIX) script in the application directory.

ERICSSON 🗲	Open	21 (29)
No.	Date Rev	
ERA/LU/P-01:0159 Uen	2002-03-18 G	

7.2 WEB DIAL

#### 7.2.1 <u>Overview</u>

The Web Dial application is delivered together with the Ericsson OSA/Parlay simulator to serve as an example of an application that makes use of the Generic Call Control Service.

The Web Dial application shows how to create an application initiated call.

## 7.2.2 Installing and running the application

When the application is ready to connect two parties, the application log file window looks like the following window (Figure 7-2).

👹 Log	_ 🗆 ×
[Trace Issuer] class com.ericsson.parlay.application.s [Action] signService [Msg] - signing service ServiceToken = serviceprofile1_test_application1 SigningAlgorithm = P_NONE	erviceprovisio 📥
[Trace Issuer] class com.ericsson.parlay.application.s [Action] getAgreementText [Msg] - getting agreement text (blocking)	erviceprovisio
[Trace Issuer] ServiceDiscovery [Action] signServiceAgreement [Msg] - signing service agreement IpAccess: org.open_service_access.fw_client.trust_ar Service token = serviceprofile1_test_application1 Agreement text = IAgreeToUseTheService Signing algorithm = P_NONE	nd_securitylp
Pause GUI	Terminate

# Figure 7-2: Web Dial application log

To create an application initiated call, at least two terminals must exist (see Figure 6-4).

Fill in the numbers of those terminals in the Web Dial window (see Figure 7-3) and press 'Create Call'.

	User Guide	
ERICSSON	Open	22 (29)
No.	Date Rev	
ERA/LU/P-01:0159 Uen	2002-03-18 G	

👹 ₩eb Dia		
A-number	123	
B-number	456	
Create call		

#### Figure 7-3: Web Dial main window

When the terminal of the A-party is ringing press 'Yes'. The terminal gets into state 'Waiting' and the B-party will be called. Press 'Yes' when the terminal of the B-party is ringing. A- and B-party will be connected then.

With Web Dial you can test all kinds of scenarios. E.g. the terminals do not exist (routing failure) or are busy.

# 7.2.3 <u>Source Code</u>

The Java source files for this application are included and are stored in the subdirectory:

#### <installationdir>\simulator\applications\WebDial\src

The application has the following package structure:

• com.ericsson.wda.common common application classes and common constant classes

Note: Other packages can be found under:

- com.ericsson.datastructures
- com.ericsson.configuration
- com.ericsson.parlay
- com.ericsson.tracedebug

These packages contain standard classes with utilities that can be used by applications.

The full source code of the Web Dial application can be compiled by running the '*Compile.bat*' (Windows) or Compile.sh (UNIX) script in the application directory.

ERICSSON 📕	Open		23 (29)
No.	Date	Rev	
ERA/LU/P-01:0159 Uen	2002-03-18	G	

## 8 <u>3<sup>RD</sup> PARTY APPLICATIONS</u>

## 8.1 INTRODUCTION

The Simulator can be used to test and demonstrate 3<sup>rd</sup> party applications. These applications may require a different configuration of the Simulator than provided by default. For example, announcements other than those configured for the Call Barring test application might be needed.

It is outside of the scope of this document to describe all possible configurations, but the following types of configuration will discussed:

- Defining application IDs
- Defining announcements

Changing the configuration of the Simulator involves updating certain fairly low-level configuration files<sup>5</sup>.

This method of changing the configuration (including the syntax used by the configuration files) is subject to change without notice. Since an incorrect configuration may cause the Simulator not to function properly, it is recommended to always make a backup of the impacted files and to keep the changes to a minimum.

#### 8.2 DEFINING APPLICATION IDS

Application IDs (and their keys) are used by the framework to authenticate client applications<sup>6</sup>. By default, the following application IDs are predefined: 'test\_application1', 'test\_application2' and 'test\_application3'. All applications use the key '3A47B193823F7D88' based on DES\_56 encryption.

If this predefined set of application IDs is insufficient, it is possible to add additional IDs. To this end, the file 'fw\_config.epct' in the 'config' directory must be updated.

One entry in this file is used to identify all client applications:

apps.ids = "test\_application1, test\_application2, test\_application3"

For each of the above application IDs, a set of application specific properties is defined:

test_applicationX.props	=	"authType,P_DES_56,P_NONE,name,email"
test_applicationX.authType	=	"P_OSA_AUTHENTICATION"
test_applicationX.P_DES_56	=	"3A47B193823F7D88"
test_applicationX.P_NONE	=	"key_is_not_needed"
test_applicationX.name	=	"test_applicationX"
test_applicationX.email	=	"test_applicationX@ericsson"

<sup>&</sup>lt;sup>5</sup> In the actual J-SCS also graphical management interfaces are available.

<sup>&</sup>lt;sup>6</sup> The process of authenication involves obtaining the IpInitial and handling of

IpAppAPILevelAuthentication:authenticate() (by the application).

	User Guide		
ERICSSON Z	Open		24 (29)
No.	Date	Rev	
ERA/LU/P-01:0159 Uen	2002-03-18	G	

In order to define additional application IDs (e.g. 'hello\_world\_appl'), the property 'apps.ids' must be updated as follows:

apps.ids = "test\_application1, test\_application2, test\_application3, hello\_world\_appl"

Next, the application specific properties must be defined by adding the following entries<sup>7</sup>:

hello_world_appl.props	=	"authType,P_DES_56,P_NONE,name,email"
hello_world_appl.authType	=	"P_OSA_AUTHENTICATION"
hello_world_appl.P_DES_56	=	"3A47B193823F7D88"
hello_world_appl.P_NONE	=	"key_is_not_needed"
hello_world_appl.name	=	"hello_world_appl"
hello_world_appl.email	=	" someone@somewhere"

At this stage the application ID 'hello\_world\_appl' has been defined but it does not yet have access to the service capabilities (Generic Call Control and Generic User Interaction).

In general, the use of a service capability may be limited for certain applications. Not every application may use all features of all service capabilities. A service profile is available for each collection of applications that use the same features of a service capability. Thus many applications can make use of the same service profile.

In the default configuration of the Simulator, all applications make use of the same service profile for Generic Call Control (serviceprofile1) and the same service profile for Generic User Interaction (serviceprofile3).

These service profiles are defined in the file 'fwinitiator.ini' in the 'config' directory. The default test applications are assigned to both profiles by means of the the following entries<sup>8</sup>:

serviceprofile1.ASSIGNMENTS = test\_application1 test\_application2 test\_application3 serviceprofile3.ASSIGNMENTS = test\_application1 test\_application2 test\_application3

In order to give the new application 'hello\_world\_appl' access to both call control and user interaction, the following updates must be made:

serviceprofile1.ASSIGNMENTS =

test\_application1 test\_application2 test\_application3 hello\_world\_appl serviceprofile3.ASSIGNMENTS =

test\_application1 test\_application2 test\_application3 hello\_world\_appl

<sup>&</sup>lt;sup>7</sup> It is outside of the scope of this document to explain each individual property of the application specific properties. It is recommended not to deviate from the given examples.

<sup>&</sup>lt;sup>8</sup> Note the seperation of application IDs by means of spaces.

	User Guide		
	Open		25 (29)
No.	Date	Rev	
ERA/LU/P-01:0159 Uen	2002-03-18	G	

# 8.3 DEFINING ANNOUNCEMENTS

The Call Barring application uses Generic User Interaction to send announcements via Parlay.

More precisely, the methods sendInfo() and sendInfoAndCollect() are used. These methods require a logical announcement ID as argument to identify the proper announcement.

When the logical announcement ID is received by the Simulator (or the actual SCS), a so called physical announcement ID is looked up, depending on the logical announcement ID. Next, the physical announcement ID is used to identify the proper announcement within the network. Graphically, this can be depicted as follows:



The distinction between logical and physical announcement IDs reduces the amount of assumptions an application must make about the underlying telecom network and increases flexibility.

Thus, in order to define a new announcement, two actions must be performed (in arbitrary order):

- 1. A mapping from the logical announcement ID to the physical announcement ID must be established.
- 2. A mapping from the physical announcement ID to the actual announcement must be established.

The next two chapters explain how to perform these actions in detail.

#### 8.3.1 Defining the mapping of logical announcement lds

The mapping of logical announcement IDs to physical announcement IDs is defined in the service profile(s) related to Generic User Interaction. This means that to change this mapping, the file 'fwinitiator.ini' in the 'config' directory must be updated.

By default, 18 announcement mappings are predefined in this file (as used by the Call Barring application).

The following extract shows how this mapping has been defined:

	User Guide		
ERICSSON 🔰	Open		26 (29)
No.	Date	Rev	
ERA/LU/P-01:0159 Uen	2002-03-18	G	

serviceprofile3.PROPERTY.P\_ANNOUNCEMENT\_MAPPING.NROFVALUES = 36 serviceprofile3.PROPERTY.P\_ANNOUNCEMENT\_MAPPING.VALUE.0 = 0 serviceprofile3.PROPERTY.P\_ANNOUNCEMENT\_MAPPING.VALUE.1 = 0 serviceprofile3.PROPERTY.P\_ANNOUNCEMENT\_MAPPING.VALUE.2 = 1 serviceprofile3.PROPERTY.P\_ANNOUNCEMENT\_MAPPING.VALUE.3 = 1

serviceprofile3.PROPERTY.P\_ANNOUNCEMENT\_MAPPING.VALUE.32 = 108 serviceprofile3.PROPERTY.P\_ANNOUNCEMENT\_MAPPING.VALUE.33 = 108 serviceprofile3.PROPERTY.P\_ANNOUNCEMENT\_MAPPING.VALUE.34 = 109 serviceprofile3.PROPERTY.P\_ANNOUNCEMENT\_MAPPING.VALUE.35 = 109

The very first entry defines the total number of announcement values. For each announcement, two values are required. The first value defines the logical ID, the second defines the physical ID.

Continuing the example of the previous chapter, suppose that the application 'hello\_world\_appl' requires one, new announcement. This application has been assigned to 'serviceprofile3' (for Generic User Interaction). Therefore, the mapping defined in 'serviceprofile3' must be updated.

First, an unused logical and physical announcement ID must be decided upon. In the default configuration, 200 could be used for both a new logical announcement ID, as well as a new physical announcement ID.

To define this mapping, two entries must be added<sup>9</sup>:

serviceprofile3.PROPERTY.P\_ANNOUNCEMENT\_MAPPING.VALUE.36 = 200 serviceprofile3.PROPERTY.P\_ANNOUNCEMENT\_MAPPING.VALUE.37 = 200

Since, two values have been added, the property 'NROFVALUES' must be increased by two:

serviceprofile3.PROPERTY.P\_ANNOUNCEMENT\_MAPPING.NROFVALUES = 38

By making these changes a mapping from 200 (logical ID) to 200 (physical ID) is established. The changes will take effect the next time that the Simulator is started.

<sup>&</sup>lt;sup>9</sup> The value immediately after VALUE is similar to an index.

	User Guide		
ERICSSON 🔰	Open		27 (29)
No.	Date	Rev	
ERA/LU/P-01:0159 Uen	2002-03-18	G	

#### 8.3.2 Defining the mapping of physical announcement IDs

Physical announcement IDs are mapped to actual announcements using the announcement manager of the simulator. In the menu bar, open the Windows menu and choose the Announcements menu item.

A new window appears that lists all announcements and their physical IDs:

Annour	ncement	nanager	_ 🗆 ×
Enabled	Id	Message	
<b>v</b>	1	Enter PIN √	
7	2	Re-enter PIN	
¥	3	PIN rejected	
¥.	4	Enter new PIN 🍕	
¥.	5	PIN changed 🤄	
¥.	6	Dial number 🚽	
¥	100	0.4	
¥	101	1.1	
¥	102	2 -8	
¥.	103	3 🚽	
¥.	104	4 📲	
¥	105	6 <b>4</b>	
¥.	106	6 📢	
¥	107	7 🕄	
¥	108	8 🕄	
7	109	9 📲	
Þ	=		Close

Buttons are available to change announcements, add new announcements and delete existing announcements. All changes are automatically saved.

To connect the physical announcement ID to a new announcement (e.g. an existing WAV-file), press the 'New'-button. A new entry appears:

Announcement manager			
Enabled	ld	Message	
<b>V</b>	1	Enter PIN 🚽	
¥	2	Re-enter PIN	
¥.	3	PIN rejected 📲	
¥.	4	Enter new PIN 🍕	
¥	5	PIN changed 🧏	
<b>V</b>	6	Dial number 📢	
<b>v</b>	100	0.4	
<b>v</b>	101	1 📢	
¥	102	2 1	
*	103	3 🖞	
¥.	104	4 🚽	
7	105	5 J	
¥.	106	6 📢	
<b>v</b>	107	7 🚽	
<b>v</b>	108	8 📢	
¥	109	9 <b>.</b> {	
¥.	7	New announcement	
•		Close	

	User Guide		
ERICSSON	Open	Open	
No.	Date	Rev	
ERA/LU/P-01:0159 Uen	2002-03-18	G	

Next, change the ID to the proper physical announcement ID (200 in this example) and enter the textual representation of the announcement:

Announcement manager			
Enabled	ld	Message	
<b>V</b>	1	Enter PIN 🚽	
¥	2	Re-enter PIN	
¥.	3	PIN rejected 🚽	
¥.	4	Enter new PIN 4	
7	5	PIN changed 4	
<b>v</b>	6	Dial number √	
<b>v</b>	100	0.4	
7	101	1 🚽	
7	102	2 1	
7	103	3 🖞	
7	104	4 🚽	
7	105	6 🚽	
7	106	6.4	
7	107	7 4	
7	108	8 🚽	
7	109	9 1	
*	200	Hello world	
•		🗋 🔟 📝 Close	

Finally, press 'edit announcement' to define the sound that is to be played when the announcement is used:



#### <u>REFERENCES</u>

9

Ref [1] OSA/Parlay APIs specification, 3GPP TS 29.198 (2001-03), Specification available on <u>http://www.3gpp.org</u>

	User Guide		
ERICSSON	Open	29 (29)	
No.	Date Rev		
ERA/LU/P-01:0159 Uen	2002-03-18 G		

Llear Guida

10 TROUBLE SHOOTING LIST

- The simulator does not start.

Check if the CLASSPATH variable has been correctly set, see chapter 5.2.3 (Configure PATH and CLASSPATH settings). Check if the ORBInitialPort is not already in use.

I cannot start the application. Make sure that the simulator is running without any errors. Make sure that the application uses the same ORBInitialPort as the simulator.

Make sure that the CLASSPATH and PATH variables are set correctly.

The Simulator and applications are very slow and produce errors. On some machines the simulator and applications are very slow while starting up and running. This might cause timeout errors in Call Barring. This problem often happens when:

- The machine has no network connection
- The machine has a dynamic IP address

Two things can be tried to solve this problem

- Add the current IP-address and machine name to the host file (under windows). This can be automatically done by using the hostname fix tool in the 'simulator'\tools directory. *Note:* In case of a dynamic IP address this has to be done every time the machine restarts.
- 2. Use 'ipconfig /release' (under windows) to clear the IP settings