bea®

BEAWebLogic RFID Edge Server™

Using the Reader Simulator

Version 2.1
Revised: June 29, 2006

Contents

1. Introduction and Roadmap

2. Using the Reader Simulator

Index

Introduction and Roadmap

The following sections describe the contents, audience for, and organization of this guide—*Using the Reader Simulator*:

- "Document Scope and Audience" on page 1-1
- "Guide to This Document" on page 1-1
- "Related Documentation" on page 1-2

Document Scope and Audience

This document describes the Reader Simulator for BEA WebLogic RFID Edge Server, its configuration options, and its graphical user interface (GUI).

This document is intended primarily for Java programmers who develop RFID client applications that use the Application Level Events (ALE) API. System administrators can also use the reader simulator for testing purposes: for example, verifying communication between the simulator and an edge server.

Guide to This Document

This document is organized as follows:

- This chapter, "Introduction and Roadmap," describes the scope of this guide, summarizes new features, and lists related documentation.

- *"Using the Reader Simulator"* provides an overview of the Reader Simulator's capabilities, instructions on how to configure the simulator, and describes the Reader Simulator GUI.

Related Documentation

This document is a part of the WebLogic RFID Edge Server documentation set, which also includes the following documents:

- *Product Overview* provides descriptive information about WebLogic RFID Edge Server, including use cases, key benefits, component architecture, functional overview, RFID concepts and terminology, and developer and RFID resources.

- *Installing WebLogic RFID Edge Server* describes how to install and configure WebLogic RFID Edge Server.

- *Using the RFID Edge Server Administration Console* describes how to use the RFID Administration Console to configure ECSpecs, ECReports, RFID devices, filters, and workflows.

- *Programming with the ALE and ALEPC APIs* describes the Application Level Events (ALE) and ALEPC programming interfaces (APIs) that you use to create applications that interact with WebLogic RFID Edge Server by reading and writing electronic product code (EPC) tags.

- *RFID Reader Reference* describes how to configure the RFID devices supported by the RFID Edge Server.

- *Workflow Reference* describes how to configure and use the workflow modules included with the WebLogic RFID Edge Server.

- *ALE and ALEPC Javadoc* provides reference documentation for the ALE and ALEPC Java packages that are provided with the WebLogic RFID Edge Server software.

Using the Reader Simulator

The following sections describe using the WebLogic RFID Edge Server Reader Simulator:

Reader Simulator Overview

Your WebLogic RFID Edge Server installation includes a Reader Simulator. The Reader Simulator is a graphical Java application that provides a minimal simulation of a ThingMagic Mercury4 reader or, with minor configuration, a minimal simulation of a Printronix printer.

Note: The Reader Simulator is a development tool. It is provided as-is for use by application developers, and is designed to emulate a subset of the ThingMagic Mercury4 reader and Printronix printer protocols. Because it has no production context, it has not been rigorously tested. The simulator is reasonably functional but not fully robust; keep this in mind when experimenting with its features and options.

The simulator can mimic reading and writing both Gen1 and Gen2 tags, and supports the following tag formats:

gid-64-i	sgln-64	sgtin-96	sgln-96	usdod-64
sgtin-64	giai-64	sscc-96	giai-96	usdod-96
sscc-64	grai-64	gid-96	grai-96	

In order for the RFID Edge Server and the Reader Simulator to communicate, they must be configured to use the same host and port, and the device that the simulator is mimicking must be the device the RFID Edge Server is configured to see. By default:

- Communication between the Reader Simulator and the Edge Server uses host 127.0.0.1 and port 5050.

- The Reader Simulator mimics a ThingMagic Mercury4 reader.

- The RFID Edge Server is configured to view RFID device SimReadr as a Mercury4 reader with two logical readers, ConnecTerra1 and ConnecTerra2.

There are two ways to modify the behavior of the Reader Simulator: by providing optional command-line arguments when running the Reader Simulator startup script (RunReaderSim), or by modifying the script itself. For information of configuring the simulator, see "Configuring the Reader Simulator" on page 2-7.

There are two ways to modify how the Edge Server views the default Reader Simulator device, SimReadr: the Administration Console or the RFID_EDGE_HOME/etc/edge.props file (where RFID_EDGE_HOME is the directory where you installed the WebLogic RFID Edge Server software). The default at installation time is to configure readers by using the Administration Console. For information on how to modify the reader type and properties associated with a Reader Simulator device, see "Configuring the RFID Edge Server View of the SimReadr Device" on page 2-12.

Note: You can create multiple instances of a Reader Simulator, and multiple devices on the Edge Server to access those simulators. For information on creating multiple simulators, see "Configuring Multiple Reader Simulators" on page 2-13.

The best way to learn about the capabilities of the Reader Simulator is to experiment with it. After you become familiar with this manual, a good hands-on introduction is to run the sample programs in RFID_EDGE_HOME/samples that use the Reader Simulator. See the Sample Java

Applications section of the *Programming with the ALE and ALEPC APIs* manual for descriptions of these sample programs and the procedures for using them.

Starting and Stopping the Reader Simulator

Prerequisite: WebLogic RFID Edge Server must be installed on your system. See *Installing WebLogic RFID Edge Server*.

Starting the Reader Simulator

It is best to start the Reader Simulator, RFID Edge Server, and RFID Edge Server Administration Console at the same time. If you run the Reader Simulator without also starting the RFID Edge Server, the simulator will wait for a connection that never comes. And, because the Administration Console uses the Edge Server as an intermediary for all communication with the simulator, you generally start all three programs at the same time.

You start the programs by invoking scripts in the `RFID_EDGE_HOME/bin` directory. Start one program per Console window so that you can see any messages the programs write. For example, for an installation on a Linux platform where `RFID_EDGE_HOME` is `/usr/local/bea/rfid_edge20`, using a separate window for each program:

```
# cd /usr/local/bea/rfid_edge20/bin
# ./RunReaderSimulator.sh
# ./RunEdgeServer.sh
# ./RunAdminConsole.sh
```

It makes no difference in which order you start the three programs. If you start the Edge Server when the simulator is already running, the Edge Server will connect immediately. If you start the simulator after the Edge Server, the Edge Server will connect to the simulator whenever an application attempts to connect to the simulator, for example, when you run the `RunQuickTest` script.

If you look at the Reader Simulator GUI after the Edge Server connects to it, you will see that the simulator's Reader Status has changed to `Connected`.

Stopping the Reader Simulator

In the Reader Simulator, select **File →Exit**.

Reader Simulator Screen and Menus

The Reader Simulator has one main screen which contains:

- Antenna panes, one for each configured antenna.

- File, Protocol, and Help menus.

- A modifiable Reliability percentage (Reliability %) field for setting the simulated reliability of the reader.

- A Simulate reader downtime field for setting the number of seconds to use when simulating a reader failure.

- A Reader Status message, which is set to Connected when the Edge Server connects to the simulator.

Tag definitions are configured on the main screen, and status messages are displayed on this screen. The following tables describe the main screen and the menus:

- Table 2-1, "Main Screen and Antenna Pane Objects," on page 2-4

- Table 2-2, "File Menu Options," on page 2-5

- Table 2-3, "Protocol Menu Options," on page 2-5

- Table 2-4, "Help Menu Options," on page 2-6

Table 2-1 Main Screen and Antenna Pane Objects

Name	Description
EPC 1:, EPC2: etc. check boxes (one per tag)	Check each tag you want the reader to see. Uncheck the tags you do not want the reader to see.
Tag format pull-down list (one per tag).	Contains a list of supported tag formats. Lets you specify the formats of the tags that the Reader Simulator is seeing.
EPC numeric value fields (three per tag)	Editable fields that contain the numeric value of the EPC.

Table 2-1 Main Screen and Antenna Pane Objects

Antenna Status (one per antenna)	Each antenna can be `Active` or `Idle`. An antenna is active when the Edge Server is asking it to do something (for example, when a program has invoked ALE API methods such as `immediate`, `poll`, and `subscribe`).
Reliability (%)	Often, tag readers do not see all the tags in their fields. You can use this field to simulate various levels of reader reliability for testing.
Simulate reader downtime	Lets you simulate a reader being unavailable for a specified number of seconds, then becoming available again. Used for testing.
Reader Status message	`Connected` when the RFID Edge Server is connected to the simulator; `Not Connected` otherwise.

Table 2-2 File Menu Options

Name	Description
Exit	Quits the application.

Table 2-3 Protocol Menu Options

Name	Description
Gen1	When the Gen1 protocol is selected, each tag is treated as a Class 1 Gen 1 tag.
Gen2	When the Gen2 protocol is selected, each tag is treated as a Class 1 Gen 2 tag. In addition, each tag has an Advanced button. (The text of the Advanced button turns red when the simulator receives a command that changes any of the Gen2 fields, except the EPC.) Click the Advanced button to display a dialog box where you can configure the Gen2-specific attributes (descriptions follow this table). When you finish, you can click Apply to exit and apply the attribute values, or click Cancel to exit without applying the attribute values.

This list describes the Gen2 attributes you can set in the Advanced dialog box for each tag:

- The current tag name and value (for example.: EPC 1: urn:epc:tag:gid-64-i:10.10.1) is printed, followed by a user-controlled Locked check box.

- User Memory input field followed by a user-controlled Locked checkbox.

- TID input field followed by a user-controlled Locked checkbox.

- Access Password input field populated with 8 zeros. If the Access password is non-zero, the tag can never be in the open state. This means that if the Access Password is set on a tag, the client (RFID Edge Server) must provide the password to write to the tag, or to read either of the password fields.

- Kill Password input field populated with 8 zeros. If the Kill Password is non-zero, the client (RFID Edge Server) must provide the password in order to kill (permanently disable) the tag. If the Kill Password is not set, the client cannot kill the tag.

- Killed check box. Simulates a killed tag. (It does not matter what the Kill password is when selecting the Killed check box.)

Locked checkboxes simulate password protecting memory fields. A memory field can be locked or unlocked. Note that the Gen2 specification lists four different states: *unlocked*, *permanently unlocked*, *secured*, and *permanently secured*. The Reader Simulator simulates two of those states: *unlocked* and *secured*; there is no simulation for *permanently unlocked* or *permanently secured*. Also locking is supported only for three of the memory fields: there is no simulated locking for the Access Password and Kill Password fields.

Refer to the Gen2 specification for detailed information about the exact meaning of the different lock types.

Table 2-4 Help Menu Options

Name	Description
Help	Displays a link to this guide.
About RFID Reader Simulator	Displays version and copyright information.

Configuring the Reader Simulator

You can use command-line options to configure the Reader Simulator behavior, or edit the `RunReaderSim` script in your `RFID_EDGE_HOME/bin` directory. Editing the `RunReaderSim` script provides more configuration options:

Note: You cannot configure the behavior of the Reader Simulator through the Administration Console or `edge.props`. Use the Console and `edge.props` to configure how the Edge Server *views* the Reader Simulator, not how the simulator behaves.

Using Command-Line Options

You can use the following command-line options when you invoke the `RunReaderSim` script:

- `-epcIndexTableURL` *EPCINDEXTABLEURL*

- `-printer`

- `-printerPort` *port*

These options are described in .

For information on how to configure the Reader Simulator to emulate a Printronix printer, see .

Editing the RunReaderSim Script

Here is the `RunReaderSim` script that is installed on a Linux system (line breaks inserted for readability). The Windows script provides the same set of variables.

```
#!/bin/sh

# Copyright (C) 2003-2006 by BEA Systems, Inc.
# All rights reserved; use is subject to license terms.
# US and international patents pending.

RFTA_ROOT=${RFTA_ROOT-`dirname $0`/..}
JAVA_HOME=${CT_JAVA_HOME-/usr/local/bea/jdk150_06}

READER_PORT=${READER_PORT-5050}
TAG_COUNT=${TAG_COUNT-7}
ANTENNA_COUNT=${ANTENNA_COUNT-2}
```

```
EPCINDEXTABLEURL=
     ${EPCINDEXTABLEURL-"http://onsepc.com/ManagerTranslation.xml"}
exec "$JAVA_HOME/bin/java" -jar "$RFTA_ROOT/lib/readersim.jar"
     -epcIndexTableURL "$EPCINDEXTABLEURL" "$@" $READER_PORT $TAG_COUNT
     $ANTENNA_COUNT
```

This script is preconfigured at installation to use your system's JVM and to point to the RFID_EDGE_HOME/lib directory to find the Java classes needed by the simulator.

Note: To run the simulator on the system where the RFID Edge Server is installed, you can simply execute this script. To run the simulator on a different machine, you need to create a script that finds the local JVM and WebLogic RFID Edge Server classes.

Table 2-5 describes the variables defined in the script. In addition, because you can insert the printer options in the script, those options are also repeated in this table.

Table 2-5 RunSimReader Variables and Options

Variable or Option	Description
ANTENNA_COUNT	The number of antennas that the simulated reader sees. Note that a ThingMagic Mercury4 reader has a maximum of four antennas. The default is 2 antennas.
CT_JAVA_HOME	An optional variable, which you can use to override JAVA_HOME.

Table 2-5 RunSimReader Variables and Options

Variable or Option	Description
EPCINDEXTABLEURL	The location of the EPC Company Prefix Index translation table. WebLogic RFID Edge Server uses this table to decode tags encoded in certain 64-bit formats, as defined by the EPCglobal EPC Tag Data Standard Version 1.1 rev 1.27.
	The default location is the EPCglobal web site: http://onsepc.com/ManagerTranslation.xml
	If you are running the Reader Simulator in an environment that is not connected to the Internet, you can use the local copy instead; the local copy is installed in the RFID_EDGE_HOME/etc directory. Edit the RunReaderSim file to point to the local file, using the following file: syntax:
	-epcIndexTableURL ManagerTranslation.xml
	In Windows, if the pathname to ManagerTranslation.xml contains spaces, replace the spaces with %20, for example: file:///C:/Program%20Files/bea/...
	Note: If you use a local file, it is up to you to keep it up to date by copying down subsequent versions from the EPCglobal site as necessary.
JAVA_HOME	The location of the JDK.
-printer	If present, the Reader Simulator will simulate a Printronix printer, in addition to a reader. You must configure a Printronix printer in the RFID Edge Server as well as including this option.
-printerPort *port*	Specifies the port that will receive the data to be printed.
	The default port is 9100.
READER_PORT	The port on which the Reader Simulator is listening.
	The script default port is 5050. If not specified in the script or via an environment variable, the program's default is port 8080.
RFTA_ROOT	The script equivalent of RFID_EDGE_HOME: the directory where the RFID Edge Server software is installed.
TAG_COUNT	The number of tags each antenna should see.
	The script default is 7 tags. If not specified, the program's default is 6 tags.

Configuring the Reader Simulator as a Printer

Table 2-5 describe the `-printer` and `-printerPort` options. This section provides some additional information about using the Reader Simulator as an RFID printer:

- The Reader Simulator can simulate a Printronix printer writing a tag. When you specify the `-printer` option, the simulator emulates the Printronix driver instead of the ThingMagic driver. Part of that emulation includes a pop-up window that emulates the Printronix printer printing a label; however, the simulation does not provide a full emulation of the Printronix Graphics Language (PGL) output.

- As shipped, the Reader Simulator is configured to run with two antennas; however, a Printronix printer has only one logical reader (antenna). The default configuration will print one label, but insert the same EPC in each of the configured antennas. To avoid this duplication, you can configure the simulator to use only one antenna.

- Note that you can, as shown in the procedure in this section, create a `SimPrinter` device, which lets you use both a `SimReadr` and a `SimPrinter` device. Configuring a printer device still allows you to use the Reader Simulator as a ThingMagic simulator.

The following lines demonstrate how to edit the `RunReaderSim` script to include the `-printer` property and to use `-printerPort` to specify a port other than 9100:

- **Windows:** `"%JAVA_HOME%\bin\java" -jar "%RFTA_ROOT%\lib\readersim.jar" -epcIndexTableURL "%EPCINDEXTABLEURL%" `**`-printer -printerPort 9101`**` %READER_PORT% %TAG_COUNT% %ANTENNA_COUNT%`

- **Linux:** `exec "$JAVA_HOME/bin/java" -jar "$RFTA_ROOT/lib/readersim.jar" -epcIndexTableURL "$EPCINDEXTABLEURL" `**`-printer -printerPort 9101`**` $READER_PORT $TAG_COUNT $ANTENNA_COUNT`

Notes: In the `RunReaderSim` script, the `-printer` and `-printerPort` properties should either be right after the `%*` (Windows) or `$@` (Linux), or you can replace the `%*` or `$@` with the properties (as was done in these examples).

The default printer port is 9100; you use the `-printerPort` option only when you want to specify a different port.

The following commands demonstrate how to perform the same function from the command line:

- **Windows:** `> RunReaderSim.bat `**`-printer -printerPort 9101`**

- **Linux:** `# ./RunReaderSim.sh `**`-printer -printerPort 9101`**

To use the Reader Simulator as a printer:

1. Using either the Administration Console or `edge.props`, whichever mode you work in, configure the RFID Edge Server to add a simulated printer device. For example, create a device with the following properties:

   – Reader Name: SimPrinter

   – Reader Type: Printronix T5000e and T5000r

   – Reader Hostname: 127.0.0.1 (must be same as the Reader Simulator)

   – Logical Reader Name: LogicalPrinter

   – PGL Form: If you supply a form, the printer simulation displays the "printed" label in a separate window. If you do not supply a form, you see a white label with nothing on it. (There is a sample PGL script in Configuring RFID Devices in the *Reader Reference* manual.)

   – You can use the default values for the rest of the fields. If you change the Reader Port number from 9100, remember to start the simulator with the `-printerPort` option and specify the new port number.

   Note: See the *RFID Reader Reference* for a full list of Printronix properties.

2. In the *Programming with the ALE and ALEPC APIs* book, follow the instructions in ImmediateProgramSample: Writing Tags to program an EPC value into a tag using the Reader Simulator. Make the following changes to the procedure:

   a. In the `RFID_EDGE_HOME/samples/ImmediateProgramSample` directory, edit the PCSpec.xml file. Change the name of the `<logicalReader>` to the one you used when creating the SimPrinter device. For example:

      ```
      <logicalReader>LogicalPrinter</logicalReader>
      ```

   b. Optional. Configure the Reader Simulator to use only one antenna. If you do not do this, you will see duplicate tags on the simulator.

   c. When starting the Reader Simulator, add the `-printer` option. For example, on a Linux system:

      ```
      ./RunReaderSim.sh -printer
      ```

   When you run the example, you should see the tag written to the simulator, and if you supplied a `.pgl` script, you should see the label written to the RFID Printer Simulator window.

Configuring the RFID Edge Server View of the SimReadr Device

From the WebLogic RFID Edge Server point of view, the Reader Simulator is just another reader. You can configure how the RFID Edge Server views the Reader Simulator device, `SimReadr`, using either the Administration Console *or* the `RFID_EDGE_HOME/etc/edge.props` file. These methods are mutually exclusive; see Two Approaches to Configuring Readers: edge.props or Administration Console in *Installing WebLogic RFID Edge Server* for a discussion about the two methods. The default at installation time is to configure readers using the Administration Console.

- "Using the Administration Console" on page 2-12

- "Using edge.props" on page 2-13

Using the Administration Console

The following procedure provides a short-hand introduction to the steps for using the Administration Console to modify the RFID Edge Server's view of the Reader Simulator. For more information about configuring readers via the Console, see Configuring and Monitoring Readers in *Using the Edge Server Administration Console*.

1. If you have more than one Edge Server defined, in the left pane of the Console, expand the Edge Server whose Reader Simulator you want to modify.

2. Select the **RFID Devices** node under that Edge Server.

3. In the RFID Devices pane, select `SimReadr`.

4. Click **Configure** to display the **Edit Reader** dialog box.

5. Edit the values of the properties you want to modify.

   Note: Although the Reader Simulator can mimic only ThingMagic and Printronix devices, the Reader Type pull-down list contains the full list of supported readers. (This is because the software does not prevent you from creating a device named `SimReadr`.) When changing the reader type of the Reader Simulator, restrict your choices to the ThingMagic and Printronix types.

6. Click **OK** to save the updated configuration, or click **Revert** to revert to the previously saved configuration, or click **Cancel** to discard all changes.

Using edge.props

The default `edge.props` file that is installed on your system contains the following block of properties definitions, which defines the Reader Simulator, `SimReadr`, as a ThingMagic Mercury4 reader with two antennas running on your local system and listening on port 5050.

```
com.connecterra.ale.reader.SimReadr.class =
 com.connecterra.ale.readertypes.ThingMagicMercury4PhysicalReader
com.connecterra.ale.reader.SimReadr.hostname = localhost
com.connecterra.ale.reader.SimReadr.port = 5050
com.connecterra.ale.reader.SimReadr.defaultRate = 0
com.connecterra.ale.reader.SimReadr.uhf2LogicalReaderName = ConnecTerra2
com.connecterra.ale.reader.SimReadr.uhf1LogicalReaderName = ConnecTerra1
```

If you are configuring the Edge Server's view of readers through `edge.props` rather than the Administration Console, edit these properties to change how the Edge Server views the Reader Simulator device.

Note: A single `edge.props` file may contain properties definitions for many readers. For testing and development purposes, you may want to keep some definitions (for example, for the simulator) in the file but inactive. You can prevent WebLogic RFID Edge Server from trying to communicate with a reader by commenting out the first line of a reader's property definitions.

Configuring Multiple Reader Simulators

This section shows you how to create a second Reader Simulator. You can extend this paradigm to create as many reader simulators, or as many printer simulators, as you want.

1. Make a copy of the `RunReaderSim` file; for example:

    ```
    cp RunReaderSim.bat RunReaderSim2.bat
    ```

2. Edit the copy, changing the `READER_PORT` number to something other than 5050; for example, set the port to 5051:

    ```
    set READER_PORT=5051
    ```

3. In the Administration Console, clone `SimReadr` to `SimReadr2`, and set the following properties for `SimReadr2`:

    – Set the Reader Port to 5051.

- For any configured antennas, change the logical names to distinguish them from those used by `SimReadr`; for example, for `SimReadr2`:

  - Set UHF Antenna 1 Logical Reader Name to `SimReader2-1`.

  - Set UHF Antenna 2 Logical Reader Name to `SimReader2-2`.

4. Start both Reader Simulators, each in its own Console window.

You should now see two Reader Simulators. Because they have the same number of antennas, the windows will be identical; there is no way for you to change the banner at the top of a simulator window. However, you can keep track of the windows by where you put them on your screen. For example, you could place each simulator window on top of its Console window.

Using the Reader Simulator with the QuickTest Utility

The Post-Installation Information section in *Installing WebLogic RFID Edge Server* describes how to use the QuickTest utility and the Reader Simulator to verify that the installation procedure succeeded. In the following steps, you observe the interactions of the QuickTest utility and the Reader Simulator to become familiar with some basic WebLogic RFID Edge Server features.

1. If you have not done so, run the test described in the Post-Installation Information section to verify the installation. Set up your desktop so you can see both the Reader Simulator and a new Console window at the same time.

2. In the new Console window, start the QuickTest utility by going to the `RFID_EDGE_HOME/bin` directory and running the following script:

```
RunQuickTest
```

In the Console window, you can see that QuickTest first connects to the Edge Server, then starts testing a logical reader called `ConnecTerra1`.

```
Connecting to edge server...
Finished connecting to edge server.
Testing logical reader ConnecTerra1...
```

Within the Reader Simulator, `ConnecTerra1` is called Antenna 1, so when QuickTest starts testing `ConnecTerra1`, the Reader Simulator shows Antenna 1's status turning to `Active`. Now look at the Console window for the tags that QuickTest read from `ConnecTerra1`:

```
Logical reader ConnecTerra1 read the following 7 EPCs:
urn:epc:tag:gid-64-i:10.50.5
urn:epc:tag:gid-64-i:10.40.4
```

```
urn:epc:tag:gid-64-i:10.10.1

urn:epc:tag:gid-64-i:10.30.3

urn:epc:tag:gid-64-i:10.70.7

urn:epc:tag:gid-64-i:10.20.2

urn:epc:tag:gid-64-i:10.60.6

Finished testing logical reader ConnecTerra1.
```

As you can see, these are the same EPC values for the seven tags that Antenna 1 in the Reader Simulator is seeing.

Now try changing what Antenna 1 of the Reader Simulator is seeing — uncheck all but the first EPC value.

3. Run QuickTest again. This time it reports seeing only one tag for Antenna 1.

```
Testing logical reader ConnecTerra1...

Logical reader ConnecTerra1 read the following EPC:

urn:epc:tag:gid-64-i:10.10.1

Finished testing logical reader ConnecTerra1.
```

To gain more experience, try using various combination of tags and antennas. You can also vary the Reliability % on the simulator, and see the change reflected in how many tags are read.

Index