

[Reader Core Proxy](#)

Reader Core Proxy

-
- [1 Introduction](#)
 - [2 Users Guide](#)
 - [2.1 Manual deployment](#)
 - [2.2 Configuration](#)
 - [3 Developers Guide](#)
 - [3.1 Acquiring the code](#)
 - [3.2 Component analysis](#)
-

Introduction

This section provides information regarding the deployment of the Reader Core components available at the [AspireRFID SVN branches](#). Before getting on with the deployment of the components, you could check the developers section which describes how to build the software from the source.

Users Guide

The developed bundles implement the role of a EPCglobal RFID Reader Protocol (RP) v.1.1 compliant reader. That means we may take a non-RP compliant reader and by implementing a Java interface into an OSGI bundle, we could make the reader AspireRFID enabled.

You can download the bundled package that contains Apache Felix with the required configuration and the OSGI bundles from [here](#) or you may proceed to manual deployment using the following instructions.

You can start Felix with the following command (while being in the root directory of the installation) and set the profile name as "core" when asked. *java -jar bin/felix.jar* Manual deployment

To deploy the bundles you need first to get the latest version of [Apache Felix](#) and unzip the archive in a directory in which you have read/write access. Before starting the server, you need to edit its configuration file located under the conf folder. You need to add the following line somewhere in the file:

```
mosgi.jmxconsole.rmiport.core=9995
```

Then you may start the server:

```
java -jar bin/felix.jar
```

You will then be asked to enter a profile name; you should type in the word "core" (without the quotes) and press enter. Then the server starts and you will be prompted with the felix shell (>). You may type help to get a list of the available commands.

Now you need to deploy some required bundles, which can be found at [the forge SVN](#).

These are the following (note the sequence):

1. OSGi R4 Compendium Bundle
2. MOSGi JMX-MX4J Agent Service

Aspire Wiki - ObjectWeb - ReaderCoreProxy

3. MOSGi JMX rmiregistry
4. MOSGi JMX-MX4J RMI Connector
5. Apache Jakarta log4j Plug-in
6. OW2 Bundles :: Externals :: Commons Collections
7. castor
8. antlr- osgi
9. commons-logging- osgi
10. commons-lang- osgi
11. commons-configuration- osgi

To deploy a bundle you will need the following command:

```
start file:"full_bundle_location"
```

where *full_bundle_location* is the location of the bundle on your disk.

To **skip the manual deployment process**, you can download the config.aspire.properties file and replace the original felix configuration file. Assuming that the required bundles exist in the {felix.home}bundle directory, when you start felix, the bundles will be auto-deployed.

You may then proceed in deploying the RP bundles in the same way. The RP bundles are the following. (Please note the sequence which is essential in order to avoid any problems):

1. JaxBLib
2. TDT Implementation bundle
3. Reader Protocol bundle (API)
4. Reader Protocol implementation bundle
5. Simulator reader
6. Intermec IF5 reader proxy

Configuration

You may configure Reader core proxy using JMX. We have developed a plug-in for AspireID, the [JMX plugin](#), which makes the underlying mechanics of configuration transparent to the user. You may find documentation regarding the use of the plug-in at the link above. Another way of configuration through JMX is via the JMX console, the jconsole tool, which is provided by the Sun Java JDK. You may connect to the server with the following command (assuming that felix is run at the local machine):

```
jconsole service:jmx:rmi:///jndi:rmi://localhost:9995/core
```

A graphical user interface should be displayed providing means of using the exposed JMX functionality. You may see the available functionality under the menu rfid->service->RPreader. For details regarding the available functionality please refer to the developers guide.

Developers Guide

As it is said in the introduction, this component is based on the OSGi framework. Thus, it is composed by a number of bundles, each implementing a different kind of functionality. The TDT bundle is covered in the [Tag Data Translation](#) section. Acquiring the code

You can get the source code of the bundles through [Aspire SVN](#). You should get all the source code under the path aspire/trunk/bundles/. The projects under bundles are build with maven and they can be imported to eclipse. Component analysis

The Reader Protocol bundle defines the API that is used by the implementation bundle. The reason we followed this strategy is to allow for other developers to implement their own implementations if such an alternative is required. The operations defined through the API are also the ones that are exported as

Aspire Wiki - ObjectWeb - ReaderCoreProxy

management operations through JMX.

The reader protocol implementation bundle is based on the codebase of [Fosstrak](#) reader project. This bundle is responsible for creating the JMX MBean and exporting it through MOSGI. Moreover, it maintains the configuration which can be updated through JMX. The bundle's jar initially contains a ReaderProtocol.xml file with a basic configuration. Every update that is made to the configuration is saved on an external configuration file that is saved under the user's home directory with the name ReaderProtocol.xml. The initial configuration is loads the RFID reader simulator implementation. If there is not such a configuration file under the user's home directory for some reason (e.g. initial deployment or deleted file) the default configuration is loaded.

In order for the component to be of any use, a HAL implementation is required. The HAL implementation is the actual reader with which the proxy module communicates. Within this module the standard RP calls are translated into proprietary calls.

Anyone who is interested in creating a bundle implementing the HAL should implement the following Java interface, which is located in the Reader protocol implementation bundle

org.ow2.aspirerfid.reader.rp.impl.hal.HardwareAbstraction

Then the implementation bundle should be deployed in the OSGI container and the implementation class should be identified in the RP configuration through JMX.

[Reader Core Proxy](#) (en)

Creator: xwiki:XWiki.nektarios Date: 2009/01/29 17:57

Last Author: xwiki:XWiki.yluo Date: 2010/06/08 14:22

Copyright (c) 2008-2010, [Aspire](#)