RFID Suite branch - Developer Documentation
RFID Suite branch - Developer Documentation

This document is destinated to developers who want to extend the RFID Suite branch.

Cote to the user documentation

The current version is the Tag *rfidsuite-200809*

Developer Skill (TDB)

- OSGi and iPOJO for edges and premises development
- JavaEE for EPCIS and ONS development
- GWT for Management Console development

Architecture (TBD)

The RFID Suite' architecture is based on the EPS (Edge-Premise-Server) model.

- Edges are directly connected to readers and sensors (and actuators) using wired (USB,Serial,OneWire,Ethernet...) or wireless (Bluetooth,ZigBee,IrDA) communications.
- Each Premise clusters a set of Edges to provide security and persistence facilities. It filters and relays the event flow to the server. Generally, there is one premise per warehouse or store.
- The Server collects the event flows from the Premises.

In the RFID Suite, edges communicate with the premises with connectors and premises communicate with the server(s) with connectors.

In some situation, edges and server can be merged in one computer.

Global Design

The global design of the RFID Suite is discribed in this document.

JMX convention for the RFIDSuite MBeans

ObjectNames MBeans objectnames must follow the naming conventions recommended by Sun .

The current convention is

- *rfid:type=reader* for readers
- *rfid:type=sensor* for sensors
- *rfid:type=listener* for listeners

In the future, the domain will become *org.ow2.aspirerfid*

Remark: the OW2 Aspire project could propose a set of standard JMX object names to EPC Global as aaddition to the RM (Reader Management) specification (for instance a *org.epcglobalinc:type=reader*). Attributes TODO Operations TODO

Readers, Sensors and Actuators Documentation

The documentation to configure and improve the readers and sensors is described in this document.

Project sources organization

Directory	Contains
branches/rfidsuite	the root
branches/rfidsuite/applications	contains the concrete applications, such as the Museum
branches/rfidsuite/rfidsuite	contains the core of the RIFD suite
branches/rfidsuite/rfidsuite/bundles	contains the bundles deployed on the edges and on the premises
branches/rfidsuite/rfidsuite/common	contains the necessary elements to deploy any component of the RFID Suite
branches/rfidsuite/rfidsuite/deployment	contains the components to deploy and to install the RFID Suite
branches/rfidsuite/rfidsuite/mailet	contains the elements to use a mail server
branches/rfidsuite/rfidsuite/manuals	contains the intallation, administration and user manuals.
branches/rfidsuite/rfidsuite/midlet	contains the architecture to use NFC in an smartphone
branches/rfidsuite/rfidsuite/server	contains the implementation of the JEE part of the RFID Suite
branches/rfidsuite/rfidsuite/server/epcis	contains the EPCIS EAR
branches/rfidsuite/rfidsuite/server/ons	contains the ONS EAR
branches/rfidsuite/readers	contains the bundles of the readers deployed on the edges
branches/rfidsuite/sensors	contains the bundles of the sensors deployed on the edges
branches/rfidsuite/actuators	contains the drivers of actuators (TBD)
branches/rfidsuite/printers	contains the drivers of RFID printers/encoders (TBD)
branches/rfidsuite/installers	contains the installers of the demo packs (TBD)

Extending the RFID Suite How to develop a new reader

What is a reader ? TBD

A reader is a iPOJO component packaged in an OSGi bundle.

The component requires a org.osgi.service.event.EventAdmin service to send org.osgi.service.event.Event object containing the ALE event.

Here is an example of rfid event sent by a fictive reader to the EventAdmin at the edge level:

- topic: *org/ow2/aspirerfid/rfidtopic/rfid*
- with the following properties:
  - event.topics=org/ow2/aspirerfid/rfidtopic/rfid
  - publisher.instance.name=fictivereader
  - *rfid.readerguid*=fictiveguid
  - *rfid.readername*=fictive
  - *rfid.tagguid*="357654321000001123456789"
  - *timestamp*=1234444400859

Events must be sent on topics matching "org/ow2/aspirerfid/rfidtopic/" in order to be included in ALE reports.

The component provides a javax.management.DynamicMBean service for the Reader configuration (serial port, ethernet port, speed, antenna ...).

Nota Bene:

- The current (ie tags/rfidsuite-200809) version of the component provides a standard MBean service for the Reader Management but is currently moving to iPOJO JMX handler (see below).
- The next version MUST use the iPOJO 1.2.0 JMX handler to simplity the MBean definition
- The next version MUST also use the iPOJO 1.2.0 CM handler to configure the reader
- The next version MUST also use the iPOJO 1.2.0 EA handler to post events.

How to develop a new sensor

What is a sensor ? A sensor can measure physical data about a identified objects or about the environment in which objects are temporally or permanently. A modern sensor can provide the instaneous measurement. Moreover, it can log the history (limited or cyclic) of the measurements taken periodically. It can also log the history of threshold infringements. WireAdmin-based sensors In the current version (ie tags/rfidsuite-200809), the driver of a sensor is a iPOJO component packaged in an OSGi bundle. It is attached to the environnement of the indentified objects and collect only the instantaneous measurements.

The component provides a org.osgi.service.wireadmin.Producer service to push/update sensor data. The type of sensor data (called flavors in the WireAdmin terminology) is not limited to a list but it must be Comparable. Typical classes are org.osgi.util.measurement.Measurement, org.osgi.util.position.Position, javax.microedition.location.Location, java.lang.Double, java.lang.Float, java.lang.Long ...

The Producer service should have the following registration properties :

- *application* represents the application (eg application=outdoor temperature)
- *org.osgi.util.measurement.Unit* represents the SI unit if the flavor is org.osgi.util.measurement.Measurement (eg org.osgi.util.measurement.Unit=K)
- … other ??? TODO

Warning : You'd better prefer SI (International System) units such as meter for distance, meter/second for speed, radian for angle, etc for your modelized data.

Producers are automatically wired to consumers using the WireAdminBinder.

Nota Bene:

- The current version (ie tags/rfidsuite-200809) of the component provides a javax.management.DynamicMBean service for the Sensor Management.
- The next version MUST use the iPOJO 1.0.0 JMX handler to simplity the MBean definition
- The next version MUST also use the iPOJO 1.0.0 CM handler to configure the sensor

MonitorAdmin-based sensors TODO

Monitor Admin UPnP-based sensors TODO

State variables of :

- Switch Power (Binary Light, Dimmable Light)
- HVAC (Thermometer, Valve)
- ...

How to develop a new actuator

The current version of the RFID Suite does not have actuators (except the NXT rotor and Nabaztag ears and choregraphy).

Actuators can be represented

- as an OSGi service,
- as an OSGi UPnPDevice service (if the real device is an UPnP device),
- as an OSGi DPWSDevice service (if the real device is a DPWS device).

Suggestions relying on standards are welcomed. UPnP-based actuators TODO Actions of :

- Switch Power (Binary Light, Dimmable Light)
- HVAC (Valve)
- AV (Media Renderer, Media Server)
- ...

How to develop a new printer

A RFID printer can print texts, logos and barcodes on a label and in the same time it can encode (ie write custom data in the memory of) the tag inlaid in the label.

The current version of the RFID Suite does not have RFID printers. Suggestions relying on standards are welcomed. How to customize a NFC MIDLet (TBD by Andres Gomez)

New MIDLets can be developed by customizing the existing ones.

see branches/rfidsuite/rfidsuite/midlet How to develop a new filter

What is a filter ? TBD How to develop a new connector

What is a connector ? TBD How to secure connectors

Secured communications can be acheived using SSL/TLS or SSH tunneling.

- HTTP/SOAP connector : TBD
- JMS/SOAP connector : TBD
- SMTP/SOAP connector : TBD
- XMPP connector : TBD

Remark: the LDAP server (used for deployment) can be used for PK and certificates distribution. How to extend the EPCIS server Data Model

The sources organization is:

Directory	Contains
branches/rfidsuite/rfidsuite/server/epcis	contains the EPCIS EAR

The EPCIS is developed using EJB3.0.

The list of EnterpriseBeans of the current version (ie tags/rfidsuite-200809) are:

- EntityBeans
    - ...
- StatelessSessionBeans (Business)
    - ...
- StatelessSessionBeans (Facade)
    - ...
- StatefulSessionBeans (Business)
    - ...
- StatefulSessionBeans (Facade)
    - ...
- MessageDrivenBeans
    - ...

Remark: The EJB3.0 container (EasyBeans on JOnAS 4 and 5) maps automatically the EntityBeans to a relational database. A dump of the generated database schema is possible with FOSS or commercial tools such as MySqlAdmin (if your DBMS is MySql), sqlplus (if your DBMS is Oracle), JavaDB util (if you uses a JDBC driver to access to your DBMS), … Most of the time, each entity is mapped on a SQL table. How to extend the Management Console

The Management Console is developed using the GWT technology.

New menu items can be added in branches/rfidsuite/rfidsuite/server/epcis/gwt/src/main/java/org/ow2/aspirerfid/app/epcis/client/widget/menu/Menu.java How to improve the ONS Server

The sources organization is:

Directory	Contains
branches/rfidsuite/rfidsuite/server/ons	contains the ONS EAR

The ONS is developed using EJB3.0.

The list of EnterpriseBeans of the current version (ie tags/rfidsuite-200809) are:

- EntityBeans
    - ...
- StatelessSessionBeans (Business)
    - ...
- StatelessSessionBeans (Facade)
    - ...
- StatefulSessionBeans (Business)
    - ...
- StatefulSessionBeans (Facade)
    - ...
- MessageDrivenBeans
    - ...

The ONS provides only a WS-* based API.

The DNS-based API must be provided.

The RESTful-based API should be provided. Readers, Sensors and Actuators deployment

Usage of the OSGi Device Access spec is recommended How to customize the installers

TBD (Walter) How to customize the 1-click demopack TBD (Walter and Didier)

The goal of the demopack is to give a quick overview of the RFID Suite to new users by installing and running a full demonstration in less 2 minutes.

The current demopack packages inn a zip file:

- Ant lightweight distribution
- JOnAS lightweight distribution
- Felix
- Aspire EAR : epcis.ear, ons.ear
- Aspire bundles and dependencies

The current demopack is based on Ant build files (start.build.xml and stop.build.xml).

Build

Maven2 is mainly used to build the projects. However, some legacy requires Ant (but not Ivy).
RFID Suite branch - Developer Documentation (en)
Creator: xwiki:XWiki.donsez Date: 2008/10/11 12:35
Last Author: xwiki:XWiki.touseau Date: 2009/02/12 13:51