Quick Start Guide of Middleware Installation

Product Version:	Middleware Version 1.0
Reviewed by:	RFID Middleware Team
Author:	RFID Middleware Team

Published: July 2007

Table of Contents

Introduction

This technical article is intended to provide a "getting started" point for RFID middleware. After following this technical article, users should be able to install and setup the middleware, and also its additional components.

The CUHK middleware implements the EPCglobal standard, “The Application Level Events (ALE) specification, version 1.0”. The middleware package also includes:

- Management Console - provides a bird-view of the running system and provides administrative and management functions.
- Reader Emulator – provides hardware reader emulation to the middleware
- Tag Capturer – demonstrates the basic functionalities of the middleware and integration with third-party EPCIS repository

This technical article does not explain the system design and the technical details of the middleware. Readers are expected to have a basic level of understanding of the EPCGlobal ALE standard, which may be referenced from the site *http://www.epcglobalinc.org/*.

Preparing the Installation

The installation of RFID middleware requires the following software to be installed in the machine:

- Java Development Kit (JDK) 1.5
- MySQL 5.0
- CUHK RFID 1.0

Installing Java Development Kit (JDK) 1.5

Install Java Development Kit (JDK) 1.5

- run jdk-1_5_0_10-windows-i586-p.exe
- make sure to include the bin directory is in your system PATH
- make sure JAVA_HOME environment variable is set

Installing MySQL 5.0

Run mysql-5.0.15-win32_setup.exe

- choose Standard Configuration in MySQL server instance configuration
- note the password of the root user
- make sure to include the bin directory is in your system PATH

Installing CUHK RFID 1.0

Run cuhkrfid_1.0.exe

Testing Installation of Middleware

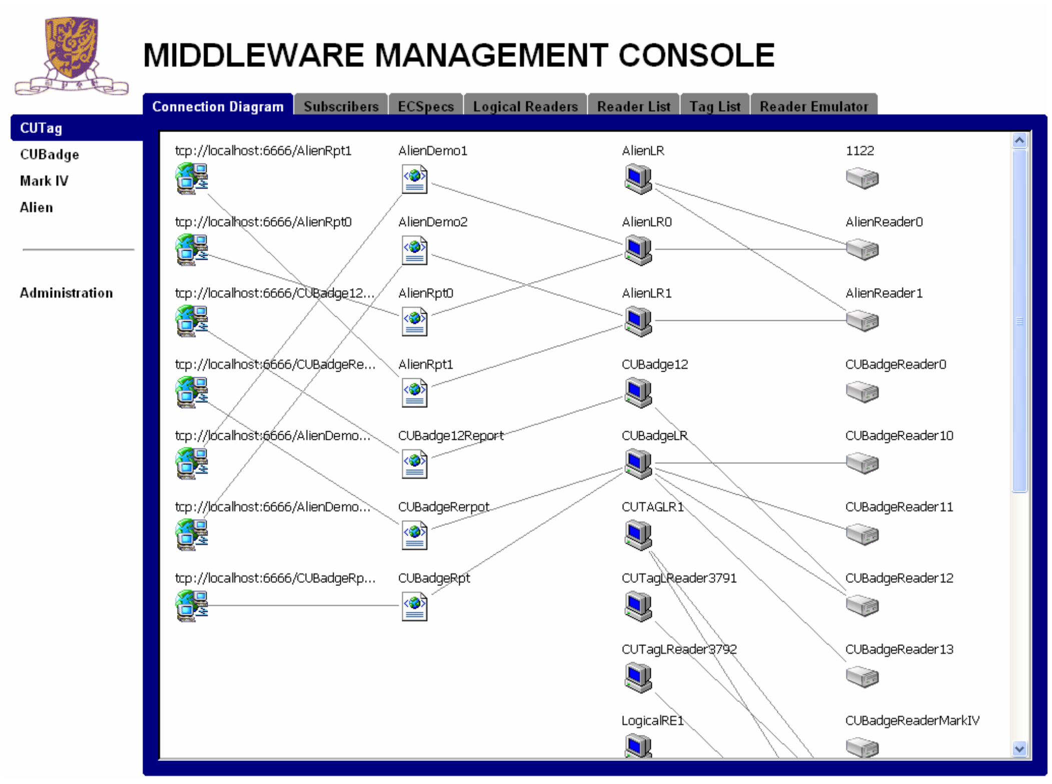
When you set up your lab computers, install and configure your systems in the following order:

Here we use a simple scenario to test the successful installation of the middleware.

1. Open Management Console:
   - Create a reader called "READER1"
   - Create a logical reader called "LREADER1"
   - Create a mapping such that logical reader "LREADER1" mapped to reader "Reader1"
2. Open Tag Capturer, in Menu, choose "Add a View"
   - Title: ecspec1
   - Logical Readers: LREADER1
   - Scope of Tags: urn:epc:pat:gid-96:1000001.0.[1-100]
   - Set of EPCs: current
   - Submit the Request
3. Open Reader Emulator:
   - Server Address: localhost
   - Reader ID: READER1
   - Reader Cycle: 1000
   - Tags: urn:epc:tag:gid-96:1000001.0.2
   - Start the emulation
4. The Tag Capturer reports the tags received.

Management Console Guide

Connection Diagram



In this tab, Subscriber, ECSpec, Logical Reader, Reader and their relations are displayed in real time with a 5-second refresh rate (configurable).

Subscribers

In this tab, details of the subscriber are listed out.

ECSPecs

A list of the ECSpec is provided, user can click on “[Display XML]” to see the corresponding XML content.

Logical Readers

A list of the Logical Readers is shown in a tree structure. User can click on the "+" icon to see the Readers encapsulated by a particular Logical Reader. User may also use the "Expand All" button to see all the details. Users can also click to add and remove logical readers

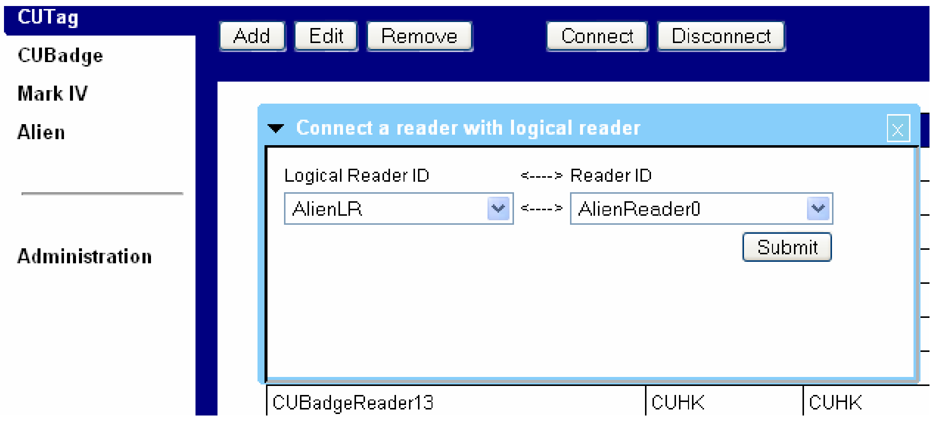
Reader List

Details of the Readers are listed out. Normally, a reader instance will be automatically added to the system, once it is turn on. User may also click on the "Add" button to create a new Reader purposely.

User can click on the "Edit" button to modify the details of a particular Reader.

Also, a reader instance will be automatically removed from the system, once it is turn off. User may also click on the "Remove" button to delete a new Reader purposely.

User can click on the "Connect" button to link up a Reader to a Logical Reader. A Logical Reader may connect to one ore more Readers.



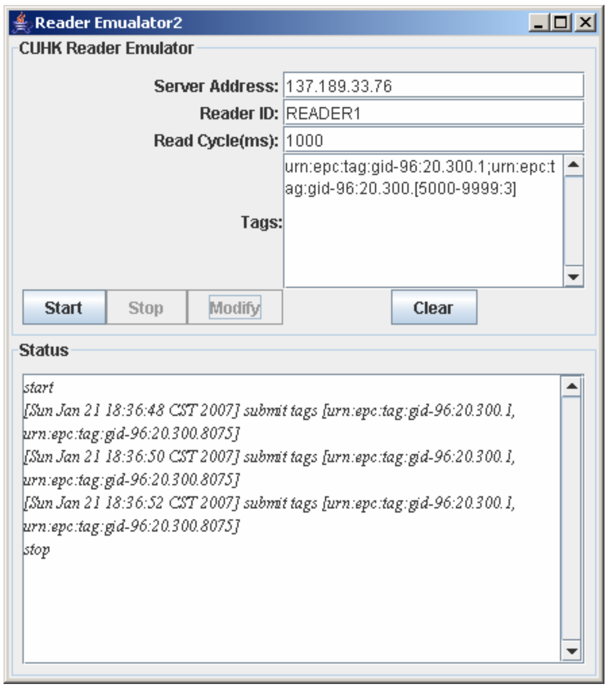
Logical Reader ID <----> Reader ID
AlienLR <----> AlienReader0

CUBadgeReader13	CUHK	CUHK

User can click on the "Disconnect" button to remove the linkage between a particular Reader and a Logical Reader.

Reader Emulator Guide

The reader emulator is the software which generates user defined tags periodically and sends to the middleware. It facilitates development and testing without the need of the hardware reader.



Server Address: 137.189.33.76
Reader ID: READER1
Read Cycle(ms): 1000
Tags: urn:epc:tag:gid-96:20.300.1;urn:epc:tag:gid-96:20.300.[5000-9999:3]

```
start
[Sun Jan 21 18:36:48 CST 2007] submit tags [urn:epc:tag:gid-96:20.300.1,
urn:epc:tag:gid-96:20.300.8075]
[Sun Jan 21 18:36:50 CST 2007] submit tags [urn:epc:tag:gid-96:20.300.1,
urn:epc:tag:gid-96:20.300.8075]
[Sun Jan 21 18:36:52 CST 2007] submit tags [urn:epc:tag:gid-96:20.300.1,
urn:epc:tag:gid-96:20.300.8075]
stop
```

Server Address: the hostname/IP address of the middleware

Reader ID: the ID of the emulated reader

Read Cycle: the emulator generates tags and submit them to the middleware in every specified cycle

Tags: the tags generated, in the form of a ";" separated tag list. The tags specified can be fixed or with a random generation rule. A fixed tag is a tag specified in EPC Tag URI format, e.g. urn:epc:tag:gid-96:20.300.1. Tags with random generation rules are just like fixed tags, but its numeric field can be in [lo-hi:cycle] format. This means that a random value between lo and hi is generated and will be stable for the specified cycle (we don't want the value to be changed in every read cycle). 1 or more numeric fields can be in this format.

For example, with urn:epc:tag:gid-96:20.300.[5000-9999:3], here is one of the possible output sequences:

read cycle 1, epc:tag:gid-96:20.300.5899 (5000 < 5899 < 9999)

read cycle 2, epc:tag:gid-96:20.300.5899 (stable for 3 read cycles)

read cycle 3, epc:tag:gid-96:20.300.5899

read cycle 4, epc:tag:gid-96:20.300.6233 (a new value is generated)

Tag Capturer Guide

The Tag Viewer is an application demonstrating the functionalities of the ALE middleware. Basically, it allows you to define and subscribe ECSpec in the middleware and receive reports from it.

Configuration

There is a file named "**config.properties**" which contains the following parameters:

- ***server.host, server.port***

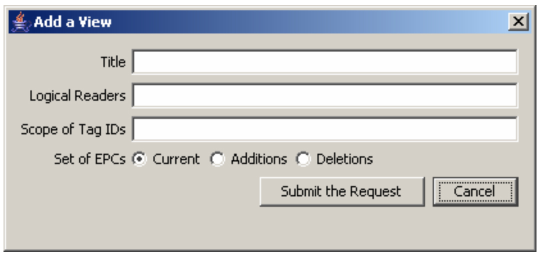
  These parameters specify the hostname and the port number of the ALE middleware. The default value is `localhost, 8080`.

- ***client.host, client.port***

  These parameters specify the hostname and the port number of the tag viewer as seen by the middleware. For example, if the tag viewer machine is running on private IP address, and the middleware is running in public Internet, then a proxy is required to be set up in order to route the traffic from middleware to the client. The ***client.host*** and ***client.port*** in this case is the proxy's public address and the port for traffic routing. The default value is `localhost, 6666`.

View

The Tag Viewer is working on "View", which is an abstraction of ECSepc and ECReport.



To add a view, the following parameters have to be specified:

- ***Title***

  This parameter corresponds to the name of the ECSpec, specName.

- ***Logical Readers***

  This parameter corresponds to the one in ECSpec. Semicolon ";" can be used to separate list of logical reader names.

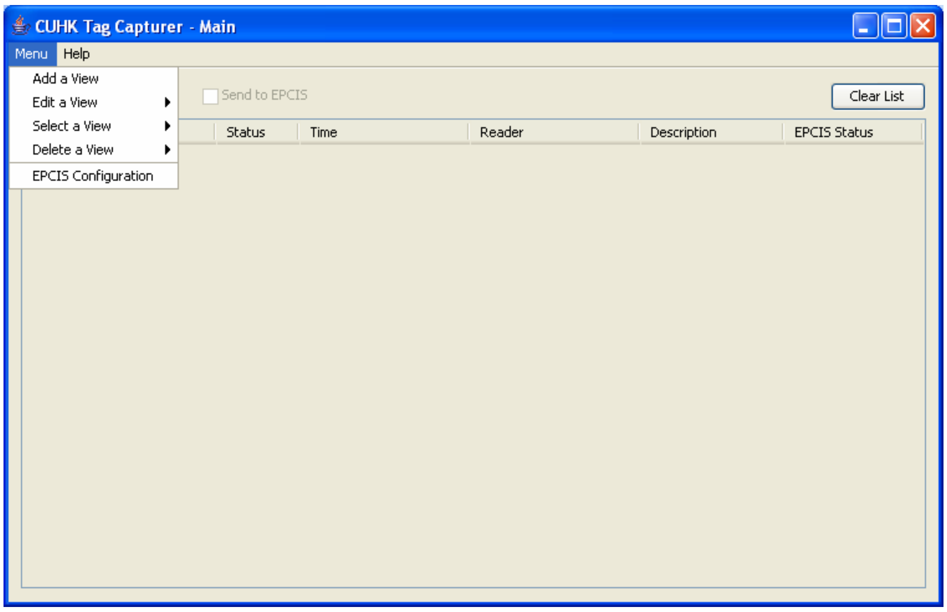
- ***Scope of Tag IDs***

  This parameter corresponds to the includePatterns in the ECFilterSpec. Semicolon ";" can be used to separate a list of tag patterns.

- ***Set of ECPs***

  This parameter corresponds to the setting in ECReportSetSpec. The ECSepc added in the middleware has a duration of 3 seconds in ECBoundarySpec.

Operations



- ***Add a View***

When a view is added, an ECSpec is defined and subscribed in the middleware. The report is displayed in the reporting window.

- ***Edit a View***

This action unsubscribes and undefines the selected ECSpec, then defines it again with new parameters and subscribe to it.

- ***Select a View***

This action displays the reports of selected view in the reporting window.

- ***Delete a View***

The action unsubscribes and then undefines the selected ECSpec.

- ***EPCIS Configurations***

This action allows modification of the URL of the Capture Interface for the EPCIS Events.

There is a Main View that displays all the reports in different views. In an individual view, users can check the box "**Alarm on New reports**" so that the reporting window will switch to that view when a report is received for that view. Besides, by checking on the checkbox "**Send to EPCIS**" will convert the reports into EPCIS Events and send to the EPCIS module via the specified URL of the Capture Interface.

There is a Main View that displays all the reports in different views. In an individual view, users can check the box "Alarm on New reports" so that the reporting window will switch to that view when a report is received for that view.

API Programming Guide

For application developers, there are 2 major java packages for the middleware.

- ***epcglobal.ale***

  This package contains the classes defined in the EPCGlobal ALE Standard. Example includes ECSpec, ECReports, etc. These classes are described in details in the specification.

- ***epcglobal.ale.soap***

  This package contains the classes that allow application to perform method calls to middleware via HTTP SOAP. Examples of operations include define, subscribe, etc.

  To include the library in your codes, simply put the jar files under the CLASSPATH of your application, tagviewer2\dist\lib\*.jar.

  The javadocs are placed under ale\docs\api.

XML & SOAP Binding

According to the EPCGlobal specification, the ECSpec and ECReport should be able to represented as XMLs, and the ALE API should be able to accessed via SOAP. Therefore, the middleware is developed with the foundation of XML to Java object transformation & SOAP accessibility.

The Java API for XML Web Services (JAX-WS, *https://jax-ws.dev.java.net/*) library is used. Before calling the ALE API via SOAP, you have to do initialization to get the implementation of ALEServicePortType interface.

```
ALEService service = new ALEService(new URL("http://
137.189.33.76:8080/ale-ws/aleservice?wsdl"), new
QName("urn:epcglobal:ale:wsdl:1", "ALEService"));


ALEServicePortType aleServicePort =
service.getALEServicePort();
```

The URL is the wsdl address for the ALE middleware. For default installation of the middleware, users may simply modify the IP address of the URL is required to change.

To do an ALE standard operation, you create its object and set appropriate parameters, and then go through the ALEServicePortType, take Define as an example:

```
Define parms = new Define();

parms.setSpec(sampleECSpec);

parms.setSpecName(“sampleSpec”);

port.define(parms);
```

By this, the input of the operation are transparently transformed into an EPCGlobal complaint XML object and sent via SOAP to the middleware, the result is returned as XML and converted back into java object.

In ALE standard, the notification mechanism is that the middleware will send back the ECReports as XML. Say you now get the inputstream, to convert the XML back into the corresponding java object, you then need JAXB to unmarshal it:

```
InputStream is = client.getInputStream();

JAXBContext jc =
JAXBContext.newInstance("epcglobal.ale");

Unmarshaller u = jc.createUnmarshaller();

JAXBElement element = (JAXBElement) u.unmarshal(is);

ECReports reports = (ECReports) element.getValue();
```

Tag Capturer Source Code Example

The tag viewer is bundled with source code. Indeed, it is a NetBeans project. The ALESOAPClient.java is the core class for doing SOAP communication. The SocketThrdServer.java demonstrates unmarshalling.

Conclusion

By following the steps in this technical article, you should have a fully functioning middleware. You have learnt how to use the Reader Emulator to create tag events, send to middleware for processing and obtains reports from the Tag Viewer.

Additional Resources

Application Level Events (ALE) Standard, Version 1.0

- http://www.epcglobalinc.org/standards/Application_Level_Event_ALE_Standard_Version_1.0.pdf