

Tutorial: Live E! Server Operation

Created: 22nd Aug. 2007
Last Updated: 6th Dec. 2010
Live E! Technical Working Group

Outline

This tutorial describes outline and operational procedure of wide-area sensor network (Live E!). Live E! system is composed of autonomously distributed servers and sensors which are administrated by independent organizations. Sensor data of these organizations can be shared in the Live E! community, which will lead to sensor applications: e.g., ubiquitous environment, facility management, agriculture, natural disaster management and education.

Section 1: Live E! System Overview

Live E! system is composed of distributed servers which autonomously form a community and share sensor data over the Internet. These distributed servers are administrated by independent organizations that are interested in and actually operate sensors. Sensor data are shared in the Live E! server network and any user who joins the community can retrieve those data from the network system.

Section 2 describes how to install a Live E! server: i.e., (1)download Live E! server package, (2)install the package and (3)join the Live E! server network.

Section 3 gives a brief introduction to server operation. Making accounts for authorizing sensors is required, and this section demonstrates how to make an account, authenticate and authorize sensors. After accounting sensors, profiles and data can be registered and uploaded. A sensor profile gives additional information to a sensor, which describes location of the sensor, type of the sensor and other information. Profiles are used not only for giving additional information to sensor applications but also indexing sensor on various application data models in the global server network.

Section 4 shows how to retrieve sensor data from the global Live E! server network briefly. Every server can retrieve data from the global network; i.e., the target servers that contain data which matches user query can be automatically selected and data can be retrieved from each server and merged. Data retrieval interface is provided by SOAP web services. Sensor application developers can easily create application software using the web service.

Section 5 gives a server redundancy technique to get robustness and improve service availability of an operational unit: organization. Sensor data and services can be replicated. They are synchronized by master / slave architecture.

Section 6 describes schema-based profile management for improving profile consistency and application layer interoperability. Application objects which are described on sensor profiles are administrated by the schema.

Section 7 gives note that any organization should know when developing new sensors for Live E!. The new sensor must be approved and registered at Live E! authority to use it in the global community. The new sensor will be included in the profile schema after the approval.

Section 2: Server Installation

The followings are the installation steps of a Live E! server.

- (1) Base system installation
- (2) Live E! system installation
- (3) Live E! server configuration

This manual describes an example of installing a Live E! server onto Ubuntu 10.04 LTS Linux. The following instructions are written under the assumption that the readers have basic knowledge

about Linux.

Live E! technical WG welcomes the report of installing a Live E! server onto other distributions of Linux or operation systems. Please make a report, if you have succeeded of installation, to live-e-tech@hongo.wide.ad.jp, which is a mailing list of Live E! technical WG.

1.1. Base System Installation

The installation of base software components on Ubuntu 8.04:

- JDK 6: Java Developers Kit
- Apache 2: Web Server
- Tomcat 6: Web Servlet Engine
- Axis 1.4: SOAP Web Service Engine
- PostgreSQL 8.4: Database Management System

Note: make sure that these components can be downloaded and installed by changing apt-line (/etc/apt/sources.list).

1.1.1. JDK 6 Installation

1.1.1.1. Make a download folder

```
$ mkdir -p ~/Downloads/LiveE
```

1.1.1.2. Install "unzip" command by:

```
$ sudo apt-get install -y unzip
```

1.1.1.3. Backup the "apt-get" sources.list file

```
$ sudo cp -p /etc/apt/sources.list /etc/apt/sources.list.bak
```

1.1.1.4. For Install JDK 6, modify the apt-get list

```
$ sudo vi /etc/apt/sources.list
```

Add following sentence

```
-----  
deb http://archive.canonical.com/ lucid partner  
-----
```

Update

```
$ sudo apt-get update
```

1.1.1.5. Install JDK 6

```
$ sudo apt-get install -y sun-java6-jdk
```

1.1.2. Apache2 Installation

```
$ sudo apt-get install -y apache2
```

1.1.3. Tomcat 6 Installation

```
$ sudo apt-get install -y tomcat6
```

Confirm that the tomcat service is up by accessing "http://localhost:8080/" or "http://localhost:8180/" with your web browser.

*) Replace "localhost" appropriately depending on your network environment

Disable tomcat security mode for Axis web service engine:

1.1.4. Axis1.4 Installation

Download and extract axis-bin-1_4.tar.gz from “<http://ws.apache.org/axis/>”.

```
$ mkdir -p ~/Downloads/LiveE/Axis/  
$ cd ~/Downloads/LiveE/Axis/  
$ wget http://www.apache.org/dist/ws/axis/1_4/axis-bin-1_4.tar.gz  
$ cd /usr/local/  
$ sudo tar zvxf ~/Downloads/LiveE/Axis/axis-bin-1_4.tar.gz  
$ sudo ln -s /usr/local/axis-1_4 /usr/local/axis
```

Make a directory copy of /usr/local/axis/webapps/axis into /var/lib/tomcat6/webapps/, and change its permission setting

```
$ sudo cp -r /usr/local/axis/webapps/axis /var/lib/tomcat6/webapps/  
$ sudo chown -R tomcat6.nogroup /var/lib/tomcat6/webapps/axis
```

Reboot tomcat:

```
$ sudo /etc/init.d/tomcat6 restart
```

Confirmation of Axis execution (this is just a confirmation of execution, the confirmation of complete installation is still required after installation of jar files in the next steps):

Refer to “<http://localhost:8080/axis/>” or “<http://localhost:8180/axis/>” with your web browser.

*) Replace "localhost" appropriately depending on your network environment

Installation of required and optional JAR files.

and are recommended to be done under the GUI interface, because on the step it is required to be certificated at web interface.

. **JavaBeans Activation Framework 1.1 (activation.jar)**

Download jaf-1_1_fr.zip from:

<http://java.sun.com/products/javabeans/jaf/downloads/index.html> ↓

After you download it, please “cd” to the directory where the downloaded file is.

```
$ unzip -e jaf-1_1_fr.zip  
$ sudo cp jaf-1.1.1/activation.jar /var/lib/tomcat6/webapps/axis/WEB-INF/lib/
```

. **JavaMail API 1.4.3 (mail.jar)**

Download javamail-1_4.3.zip from:

<http://java.sun.com/products/javamail/downloads/index.html>

After you download it, please “cd” to the directory where the downloaded file is.

```
$ unzip -e javamail-1.4.3.zip  
$ sudo cp javamail-1.4.3/mail.jar /var/lib/tomcat6/webapps/axis/WEB-INF/lib/
```

. **XML Security (xmlsec.jar)**

Download xml-security-bin-1_4_3.zip from:

http://santuario.apache.org/dist/java-library/xml-security-bin-1_4_3.zip

```
$ mkdir -p ~/Downloads/LiveE/XMLsec/  
$ cd ~/Downloads/LiveE/XMLsec/  
$ wget http://santuario.apache.org/dist/java-library/xml-security-bin-1_4_3.zip  
$ unzip -e xml-security-bin-1_4_3.zip
```

```
$ sudo cp ~/Downloads/LiveE/XMLsec/xml-security-1_4_3/libs/xmlsec-1.4.3.jar
/var/lib/tomcat6/webapps/axis/WEB-INF/lib/
```

Restart tomcat:

```
$ sudo /etc/init.d/tomcat6 restart
```

Check axis installation status by its validation page, which is linked from “<http://localhost:8080/axis/>” or “<http://localhost:8180/axis/>”.

*) Replace "localhost" appropriately depending on your network environment

1.1.5. Apache tomcat connector (mod_proxy, mod_proxy_ajp) activation

```
$ sudo vi /etc/apache2/sites-available/default
```

Add in<VirtualHost>

```
Alias /axis "/var/lib/tomcat6/webapps/axis"
<Directory "/var/lib/tomcat6/webapps/axis">
    Options Indexes FollowSymLinks
    AllowOverride None
    Order allow,deny
    allow from all
</Directory>
```

```
$ sudo vi /var/lib/tomcat6/conf/server.xml
```

Replace the following sentence (in 93 row),

```
<!--
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
-->
```

by

```
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
```

```
$ sudo vi /etc/apache2/mods-available/proxy_ajp.conf
```

create “proxy_ajp.conf”

```
<Location /axis/>
    ProxyPass ajp://localhost:8009/axis/
</Location>
```

Load mod_prox and mod_proxy_ajp:

```
$sudo a2enmod proxy_ajp
```

Restart Apache2 & tomcat6:

```
$ sudo /etc/init.d/apache2 restart
```

```
$ sudo /etc/init.d/tomcat6 restart
```

Confirm the connection from Apache to Axis by checking “<http://localhost/axis/>” with your web browser.

*) Replace "localhost" appropriately depending on your network environment

1.1.6. PostgreSQL 8.4 Installation

```
$ sudo apt-get install -y postgresql-8.4
```

Configuration of PostgreSQL:

Modify some sentences at the end of /etc/postgresql/8.4/main/pg_hba.conf as follows: i.e., replace “indentsameuser” and “md5” with “trust”.

```
$ sudo vi /etc/postgresql/8.4/main/pg_hba.conf
```

local	all	postgres		trust
# TYPE	DATABASE	USER	CIDR-ADDRESS	METHOD
local	all	all		trust
# IPv4	local connections			
host	all	all	127.0.0.1/32	trust
# IPv6	local connections			
host	all	all	:::1/128	trust

Edit /etc/postgresql/8.4/main/postgresql.conf:

```
$ sudo vi /etc/postgresql/8.4/main/postgresql.conf
```

Replace the following sentence (in 59 row),

```
#listen_address = 'localhost'
```

by

```
listen_address = '*'
```

Restart PostgreSQL:

```
$ sudo /etc/init.d/postgresql-8.4 restart
```

Installation of PostgreSQL JDBC Driver:

Download postgresql-8.4-702.jdbc4.jar from <http://jdbc.postgresql.org/download.html> and copy it to /var/lib/tomcat6/webapps/axis/WEB-INF/lib/

```
$ mkdir -p ~/Downloads/LiveE/JDBC/
```

```
$ cd ~/Downloads/LiveE/JDBC/
```

```
$ wget http://jdbc.postgresql.org/download/postgresql-8.4-702.jdbc4.jar
```

```
$ sudo cp ~/Downloads/LiveE/JDBC/postgresql-8.4-702.jdbc4.jar
```

```
/var/lib/tomcat6/webapps/axis/WEB-INF/lib/
```

Restart Tomcat:

```
$ sudo /etc/init.d/tomcat6 restart
```

1.2. Live E! system component Installation

Download Live E! server package from <http://live-e2.hongo.wide.ad.jp/dist/>. This section describes the installation steps of a Live E! service. The installation involves settings of LIVEE_HOME and CLASSPATH environmental variables. The structure of the package (live-e-datamanager-0.9.x.zip) is:

```
live-e-datamanager-0.9.x
+ bin -- Daemon Controller
+ lib -- JAR Library
| + live-e-datamanager-0.9.x.jar
+ conf -- Configurations
```

```

| + env.vars
| + livee_config.xml
| + deploy_slaveLiveE.wsdd
| + undeploy_masterLiveE.wsdd
| + undeploy_slaveLiveE.wsdd
+ schema -- Database Schema
| + live-e-datamanager-0.9.sql
+ log -- Output Logs
+ sample
- profile_template.xml -- Template sensor profile

```

1.2.1. Package extraction

Extract the package as follows:

```

$ mkdir -p ~/Downloads/LiveE/LiveE/
$ cd ~/Downloads/LiveE/LiveE/
$ wget http://live-e2.hongo.wide.ad.jp/dist/live-e-datamanager-0.9.14.zip
$ cd /usr/local/
$ sudo unzip ~/Downloads/LiveE/LiveE/live-e-datamanager-0.9.14.zip
$ sudo ln -s live-e-datamanager-0.9.* livee

```

Change the permission of log directory:

```

$ sudo chown -R tomcat6.nogroup /usr/local/livee/log

```

1.2.2. Environment variable setting

Most of the environmental variable settings are made by /usr/local/livee/conf/env.vars. At first, edit this file so that the versions and file paths are correct. Then, carry out this file before executing Live E! related programs by setting into startup scripts.

Confirmation of settings

```

$ sudo mv /usr/local/livee/conf/env.vars /usr/local/livee/conf/env.vars.bak
$ sudo vi /usr/local/livee/conf/env.vars

```

Check the versions and the paths of Axis, Tomcat, Live E!, Java, and modify them if needed.

Replace

```

-----
export JAVA_HOME=/usr/lib/jvm/java-1.5.0-sun/
export CATALINA_HOME=/usr/share/tomcat5.5
export AXIS_HOME=/usr/local/axis
export LIVEE_HOME=/usr/local/livee
export LIVEE_JAR_NAME=live-e-datamanager-0.9.13.jar
export CLASSPATH=.:$AXIS_HOME/lib/axis-
ant.jar:$AXIS_HOME/lib/axis.jar:$AXIS_HOME/lib/commons-discovery-
0.2.jar:$AXIS_HOME/lib/commons-logging-
1.0.4.jar:$AXIS_HOME/lib/jaxrpc.jar:$AXIS_HOME/lib/log4j-
1.2.8.jar:$AXIS_HOME/lib/saaj.jar:$AXIS_HOME/lib/wsdl4j-
1.5.1.jar:$CATALINA_HOME/common/lib/activation.jar:$CATALINA_HOME/common/lib/m
ail.jar:$CATALINA_HOME/common/lib/xmlsec-
1.4.2.jar:$CATALINA_HOME/common/lib/postgresql-8.3-
603.jdbc3.jar:$LIVEE_HOME/lib/"$LIVEE_JAR_NAME"
-----

```

By

```

-----
export JAVA_HOME=/usr/lib/jvm/java-6-sun/
export CATALINA_HOME=/var/lib/tomcat6
export AXIS_HOME=/usr/local/axis
export LIVEE_HOME=/usr/local/livee
export LIVEE_JAR_NAME=live-e-datamanager-0.9.14.jar
export CLASSPATH=.:$AXIS_HOME/lib/axis-
ant.jar:$AXIS_HOME/lib/axis.jar:$AXIS_HOME/lib/commons-discovery-
0.2.jar:$AXIS_HOME/lib/commons-logging-
1.0.4.jar:$AXIS_HOME/lib/jaxrpc.jar:$AXIS_HOME/lib/log4j-
1.2.8.jar:$AXIS_HOME/lib/saaj.jar:$AXIS_HOME/lib/wsdl4j-
1.5.1.jar:$CATALINA_HOME/webapps/axis/WEB-
INF/lib/activation.jar:$CATALINA_HOME/webapps/axis/WEB-
INF/lib/mail.jar:$CATALINA_HOME/webapps/axis/WEB-INF/lib/xmlsec-
1.4.3.jar:$CATALINA_HOME/webapps/axis/WEB-INF/lib/postgresql-8.4-
702.jdbc4.jar:$LIVEE_HOME/lib/"$LIVEE_JAR_NAME"
-----

```

Import into Tomcat6

```

$ sudo cp /etc/init.d/tomcat6 /etc/init.d/tomcat6.bak
$ sudo vi /etc/init.d/tomcat6

```

Add the following line

```

./usr/local/livee/conf/env.vars
into /etc/init.d/tomcat6 below the following two lines
./usr/lib/init-functions (in row 42 )
./etc/default/rcS (in row 45 )

```

Import into terminals

```

$ sudo cp /etc/profile /etc/profile.bak
$ sudo vi /etc/profile

```

Add the following line

```

./usr/local/livee/conf/env.vars
to /etc/profile (in row 3)

```

1.2.3. Confirmation of the Configuration

Carry out this:

```

$ source /etc/profile

```

then, check the CLASSPATH variable:

```

$ echo $CLASSPATH
./usr/local/axis/lib/axis-ant.jar:/usr/local/axis/lib/axis.jar:/usr/local/axis/lib/commons-discovery-
0.2.jar:/usr/local/axis/lib/commons-logging-
1.0.4.jar:/usr/local/axis/lib/jaxrpc.jar:/usr/local/axis/lib/log4j-
1.2.8.jar:/usr/local/axis/lib/saaj.jar:/usr/local/axis/lib/wsdl4j-
1.5.1.jar:/var/lib/tomcat6/webapps/axis/WEB-
INF/lib/activation.jar:/var/lib/tomcat6/webapps/axis/WEB-
INF/lib/mail.jar:/var/lib/tomcat6/webapps/axis/WEB-INF/lib/xmlsec-
1.4.3.jar:/var/lib/tomcat6/webapps/axis/WEB-INF/lib/postgresql-8.4-
702.jdbc4.jar:/usr/local/livee/lib/live-e-datamanager-0.9.14.jar

```

check the JAVA_HOME variable:

```
$ echo $JAVA_HOME  
/usr/lib/jvm/java-6-sun/
```

check the LIVEE_HOME variable:

```
$ echo $LIVEE_HOME  
/usr/local/livee
```

1.2.4. Live E! Database Creation

Create a database which works in the backend of Live E! service, and load the schema of Live E! database.

```
$ sudo su postgres  
postgres@ $ createdb livee  
postgres@ $ cd /usr/local/livee/schema  
postgres@ $ psql livee -f live-e-datamanager-0.9.sql  
... Messages will be produced (check that no error messages are presented) ...  
postgres@ $ exit  
$
```

1.2.5. Live E! component installation in Axis

Create a file link to include live-e-datamanager into Axis as follows:

```
$ cd /var/lib/tomcat6/webapps/axis/WEB-INF/lib/  
$ sudo ln /usr/local/livee/lib/live-e-datamanager-0.9.14.jar live-e-datamanager.jar
```

Restart Tomcat:

```
$ sudo /etc/init.d/tomcat6 restart
```

1.2.6. Live E! service deployment on Axis web service

The services that should be deployed are different, depending on the mode (master/slave) of the server. Be sure to deploy the correct service.

If the server works as a master server:

```
$ cd /usr/local/livee/conf/  
$ java org.apache.axis.client.AdminClient deploy_masterLiveE.wsdd
```

If the server works as a slave server:

```
$ cd /usr/local/livee/conf/  
$ java org.apache.axis.client.AdminClient deploy_slaveLiveE.wsdd
```

1.3. Live E! Server Configuration

Live E! server configuration is described in livee_config.xml. This section only gives how to configure neighbor topology information. Live E! manages a server community, which is connected over the Internet by tree-structured topology. Each server must be included into the community (by an upper site) and also can include other servers into the community (as lower sites). This section shows the configuration of neighbor topology information that implements the architecture. For the detail of server configuration, see Appendix A.

The configuration file is written in XML, and the neighbor topology information is located at /c:liveeConfiguration/c:neighbor/c:topology/a:neighbor.

Here,

```
xmlns:c="http://live-e.org/Configuration/2007/03/"
```


xmlns:a="http://live-e.org/Administrator/2007/03/"

The naming rule of Live E! server node is very similar to that of domain name system(DNS). The root server is identified by dot(.) and each server just under the root has a country-code name basically: e.g., "jp." and "tw.". These servers have authority of operating their sub-trees and they can allocate names on their child servers on their own authority; e.g., "jp." server can allocate "hoge.jp." and "hoge hoge.jp." to their child servers. These names and corresponding servers and organizations are called "site" in Live E!. Each site can setup redundant servers(Section 5).

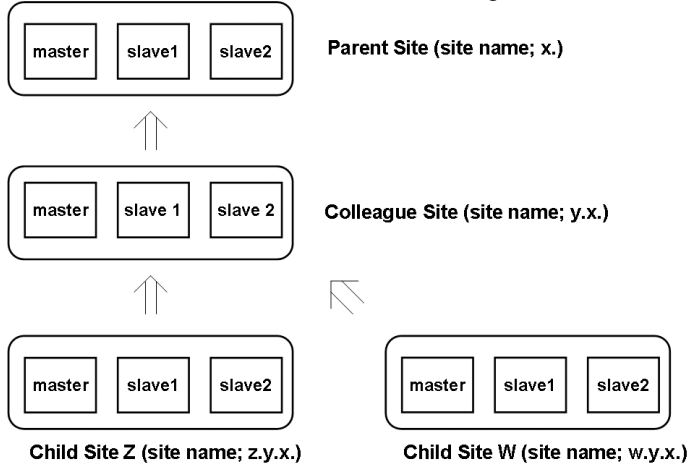


Figure 1: Neighbor Topology Example for site "y.x."

The configuration of neighbor topology information for server "y.x." in figure 1 is shown in figure 2.

```
<neighbor xmlns="http://live-e.org/Administrator/2007/03/">
  <site type="parent" name="x.">
    <server name="Admin200703" url="http://master.---/axis/services/Admin200703" />
    <server name="Admin200703" url="http://slave1.---/axis/services/Admin200703" />
    <server name="Admin200703" url="http://slave2.---/axis/services/Admin200703" />
  </site>
  <site type="colleague" name="y.x.">
    <server name="Admin200703" url="http://master.---/axis/services/Admin200703" />
    <server name="Admin200703" url="http://slave1.---/axis/services/Admin200703" />
    <server name="Admin200703" url="http://slave2.---/axis/services/Admin200703" />
  </site>
  <site type="child" name="z.y.x.">
    <server name="Admin200703" url="http://master.---/axis/services/Admin200703" />
    <server name="Admin200703" url="http://slave1.---/axis/services/Admin200703" />
    <server name="Admin200703" url="http://slave2.---/axis/services/Admin200703" />
  </site>
  <site type="child" name="w.y.x.">
    <server name="Admin200703" url="http://master.---/axis/services/Admin200703" />
    <server name="Admin200703" url="http://slave1.---/axis/services/Admin200703" />
    <server name="Admin200703" url="http://slave2.---/axis/services/Admin200703" />
  </site>
</neighbor>
```

Figure 2: Configuration of Neighbor Topology in "y.x."

Practically, when the server tree is constructed as in figure 3, the configurations at "fr." and "fire04.fr." can be set as figure 4 and figure 5 respectively.

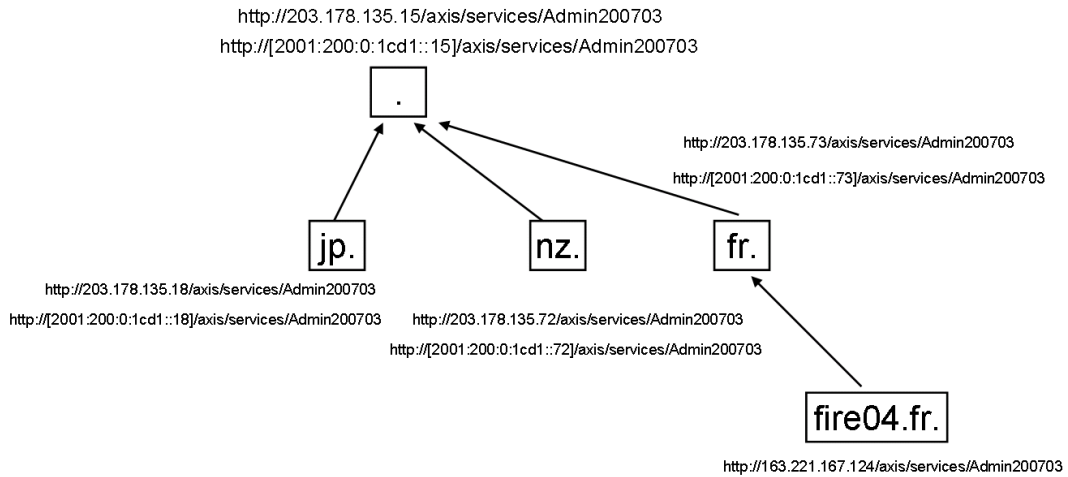


Figure 3: Example of a practical server tree

```
<neighbor xmlns="http://live-e.org/Administrator/2007/03/">
  <site type="parent" name=".">
    <server name="Admin200703" url="http://[2001:200:0:1cd1::15]/axis/services/Admin200703" />
    <server name="Admin200703" url="http://203.178.135.15/axis/services/Admin200703" />
  </site>
  <site type="colleague" name="fr.">
    <server name="Admin200703" url="http://[2001:200:0:1cd1::73]/axis/services/Admin200703" />
    <server name="Admin200703" url="http://203.178.135.73/axis/services/Admin200703" />
  </site>
  <site type="child" name="fire04.fr.">
    <server name="Admin200703" url="http://163.221.167.124/axis/services/Admin200703" />
  </site>
</neighbor>
```

Figure 4: Neighbor Topology Configuration at “fr.”

```
<neighbor xmlns="http://live-e.org/Administrator/2007/03/">
  <site type="parent" name="fr.">
    <server name="Admin200703" url="http://[2001:200:0:1cd1::73]/axis/services/Admin200703" />
    <server name="Admin200703" url="http://203.178.135.73/axis/services/Admin200703" />
  </site>
  <site type="colleague" name="fire04.fr.">
    <server name="Admin200703" url="http://163.221.167.124/axis/services/Admin200703" />
  </site>
</neighbor>
```

Figure 5: Neighbor Topology Configuration at “fire04.fr.”

After the configuration, start “Administrator”

```
$ sudo chmod 755 /usr/local/livee/bin/administrator.sh
$ . /usr/local/livee/bin/administrator.sh start
```

Restart Tomcat

```
$ sudo /etc/init.d/tomcat6 restart
```

2.4. Server Test (Running Status)

```
$ java org.livee.test.TestAdmin200703
```

This command gives server status in XML format. Check that it does not give any error message.

2.5. Server Log

Server log will be put in /usr/local/livee/log/

- live-e.log --- Normal Log
- live-e-warning.log --- Warning Message
- live-e-error.log --- Error Message

Check live-e-warning.log and live-e-error.log to confirm that it is working properly. Any questions are welcomed at live-e[at]mri.co.jp. These server log messages can be transferred by e-mail, see Section 3.3.

2.6. Set Automatic Activation

Create the following Symbolic links.

```
/etc/init.d/livee -> /usr/local/livee/bin/administrator.sh  
/etc/rc2.d/S92livee -> /etc/init.d/livee  
/etc/rc3.d/S92livee -> /etc/init.d/livee  
/etc/rc4.d/S92livee -> /etc/init.d/livee  
/etc/rc5.d/S92livee -> /etc/init.d/livee
```

```
$ sudo su -
```

```
# ln -s /usr/local/livee/bin/administrator.sh /etc/init.d/livee  
# ln -s ../init.d/livee/etc/rc2.d/S92livee  
# ln -s ../init.d/livee/etc/rc3.d/S92livee  
# ln -s ../init.d/livee/etc/rc4.d/S92livee  
# ln -s ../init.d/livee/etc/rc5.d/S92livee
```

Edit “administrator.sh”

```
# vi /usr/local/livee/bin/administrator.sh  
the line before “case”, input the following one line.
```

```
./usr/local/livee/conf/env.vars
```

Reboot the machine, and confirm to activate automatically.

```
$ ps auxwww | grep Administrator
```

if there is a term "java org.livee.data.command.Administrator", the activation success.

Section 3: Server Operation

3.1. Sensor Registration

Live E! server operators must authorize sensors with their authority. This authorization can be performed by sensor accounting: i.e., sensor registration and sensor authentication. Live E! server system has a sensor accounting command to achieve that account-based authorization.

3.1.1. Sensor ID Format

□. Combined Sensor ID

Live E! defines “combined sensor” as a packaged sensor of multiple sensors (e.g., temperature sensor and humidity sensor). Each combined sensor must have globally-unique ID. Live E! defines ID format that enables the global-uniqueness as follows:

CombinedSensorID ::= FQDN of the organization “/” sensorModel “/” free-format “/”

For example, a combined sensor ID of the sensor deployed at the room B206 of “Nara Institute of Science and Technology (NAIST)”, which sensor model is WM918, can be:

live-e.naist.jp/WM918/B206/
(FQDN of the organization / sensorModel / free-format /)

Note: Combined sensor ID must be ended with slash (/).

□. Element Sensor ID

This ID should be allocated on every elemental sensor that can work as an independent sensor. The format should be:

ElementSensorID ::= CombinedSensorID sensorType

The current approved sensor types are listed here:

- Temperature
- Humidity
- RainFall
- DayRainFall
- WindDir
- WindSpeed
- Illuminance

For example, the element sensor ID for “temperature” sensor of live-e.naist.jp/WM918/B206/ is: live-e.naist.jp/WM918/B206/Temperature

Note: element ID must NOT be ended with slash (/).

3.1.2. Sensor Accounting

To make an account for a new sensor, login to the server and use “Authorization” command as follows:

```
$ java org.livee.data.command.Authorization add -i hongo.wide.ad.jp/WM918/elab/ -p 00000000 -o "Hideya Ochiai" -m jo2lxq@hongo.wide.ad.jp
```

“add” means new sensor registration, -i specifies the combined sensor ID, -p specifies password, -o specifies owner name of the sensor, and -m specifies mail address of the sensor owner. For the detail of the command, see Appendix B.

3.1.3. Sensor Profile Registration

After accounting a sensor, profile of the sensor must be registered to the server by “ProfileManager” command or any other profile registration tools. A sensor profile gives additional information to its sensor: e.g., geographical address, vendor name, latitude and longitude. Sensor profile is also used for indexing sensor for applications. For profile format, see Appendix C.

ProfileManager command:

```
$ java org.livee.data.command.ProfileManager set -f FILEPATH -p 00000000
```

The template file is extracted at:

```
$LIVEE_HOME/sample/profile_template.xml
```

Profile management service is also provided by a web service. For the detail of the service, see Appendix D.

Note: SensorRegisterDX can register sensor profile via the Web service. This software is available at <http://live-e.naist.jp/SensorRegisterDX/>

3.1.4. Sensor Data Upload

Sensor data uploading service is provided by a web service (DataUpload200703). All the sensors must use this service to publish its observed sensor data.

The web service (DataUpload200703) has the following three remote methods for several upload granularities. See also Appendix E.

String uploadElement(String data)
String uploadCombined(String data)
String uploadCollection(String data)

Armadillo sensor hosts for WM918 / WXT510 which use the web service are available at <http://live-e.hongo.wide.ad.jp/DataUpload200703/>

3.2. Local Data Retrieval

```
$ java org.livee.data.command.LocalDataRetriever
```

```
Usage: java org.livee.data.command.LocalDataRetriever COMMAND [OPTIONS]
```

```
COMMAND:= getProfileSchema | getLatestDataAll  
          | getLatestData | getDataHourlyAggregated  
          | getDataDailyAggregated | getDataMonthlyAggregated  
          | getProfileAll | getProfile
```

```
OPTIONS:=(OPTION)* | e
```

```
OPTION:= -i ID | -s START_TIMESTAMP | -e END_TIMESTAMP  
         | -l LOCALE | -tz TIMEZONE
```

```
$ java org.livee.data.command.CSVArchiver
```

```
Usage: java org.livee.data.command.CSVArchiver ID [OPTIONS]
```

```
OPTIONS:=(OPTION)* | e
```

```
OPTION:= -s START_TIMESTAMP | -e END_TIMESTAMP
```

3.3. Logging and Message Notification

Server log will be put in /usr/local/livee/log/

- live-e.log --- Normal Log
- live-e-warning.log --- Warning Message
- live-e-error.log --- Error Message

These messages can be transferred by e-mail. Figure 6 shows logging system configuration. The configuration is written at /liveeConfiguration/logging in the configuration file (livee_config.xml).

```
<logging>
  <smtpServer></smtpServer>
  <normal>
    <mail></mail>
    <file>live-e.log</file>
  </normal>
  <warning>
    <mail></mail>
    <file>live-e-warning.log</file>
  </warning>
  <error>
    <mail></mail>
    <file>live-e-error.log</file>
  </error>
</logging>
```

Figure 6: Logging System Configuration

To receive messages by e-mail, specify SMTP server in “smtpServer” and mail addresses in “mail” XML nodes. Then, restart “Administrator” and “Tomcat”.

Section 4: Global Sensor Search and Data Retrieval

Server installation and joining in the global Live E! server network enables to search and retrieve sensor data from the global community. Live E! server provides search and retrieval service interface by a web service (GlobalDataProvider200703). This service provides the following functionalities.

- Profile Schema Retrieval
- Sensor Profile Retrieval
- Sensor Data Retrieval
- Language Locale Selection
- Time Zone Selection
- Geographical Region Selection
- Aggregated Values (e.g., Maximum, Minimum and Average)
- Server Search

Please also refer to Appendix F, which describes the detail of the service.

Server administrator can also search and retrieve sensor from the server console.

```
$ java org.livee.data.command.GlobalDataResolver
```

Section 5: Redundant Services and Data for Robustness

Redundancy is necessary for improving service and data robustness. Live E! enables the redundancy with Master / Slave technique. Slave servers synchronizes their master server, and data search links, retrieval interfaces and data itself will be redundantly saved. An operational unit is called “site” in Live E!. Any sites can setup redundant servers in this way.

The following is the setting steps.

1. Slave server configuration
2. Service deployment
3. Binding with the Live E! server tree

5.1. Slave Server Configuration

5.1.1. Mode setting

In the configuration file(livee_config.xml), alter

```
<mode>master</mode>
with
<mode>slave</mode>
```

5.1.2. Master server setting

```
<slave>
  <masterAdminURL>http://live-e2.hongo.wide.ad.jp/axis/services/Admin200703 </masterAdminURL>
  . . .
</slave>
```

Alter the underlined URL with your master's Admin200703 service URL.

5.2. Service Deployment

Confirm the existence of live-e-datamanager-0.9.x.jar in WEB_INF/lib/ directory in Axis on Tomcat, and deploy services for slave settings as follows:

```
$ cd /usr/local/livee
$ cd conf
$ java org.apache.axis.client.AdminClient deploy_slaveLiveE.wsdd
```

Note: If master services are already deployed,

```
$ java org.apache.axis.client.AdminClient undeploy_masterLiveE.wsdd
$ java org.apache.axis.client.AdminClient deploy_slaveLiveE.wsdd
```

Then, reboot tomcat to enable.

5.3. Binding with the Live E! server tree

The operator must have the parent site registered the redundant service list. In neighbor topology configuration, redundant services must be set at 'site type="colleague"' as follows:

```
<site type="colleague" name="jp.">
  <server service="Admin200703" url="http://master.domain.com/axis/services/Admin200703" />
  <server service="Admin200703" url="http://slave0.domain.com/axis/services/Admin200703" />
  <server service="Admin200703" url="http://slave1.domain.com/axis/services/Admin200703" />
</site>
```

When a child site operator notifies their topology update, configure the topology information at the corresponding site 'type="child" name="xxx"'. .

Section 6: Sensor profile consistency management with schema

Live E! system has a single profile schema maintained by Live E! authority to improve sensor profile consistency and to index sensors using their profile. Live E! authority publishes the schema at the root server of the tree, and the schema will be automatically disseminated to all the server on down links of the tree. Using the schema, profiles are verified periodically or at the registration, and if mismatches are found, warning messages will be posted.

The following is the detail of the profile schema.

Structure

```

<profileSchema xmlns="http://live-e.org/Schema/2007/03/">
  <schema name="location" class="combined" type="string" value="*" multilanguage="true" />
  <schema name="latitude" class="combined|element" type="float" />
  . . .
</profileSchema>

```

Schema attributes and the meanings

Attribute	Meaning
name	The name of the attribute in profile.
class	The layer of the attribute to appear in profile (regular expression). collection -- combined sensor collection layer combined -- combined sensor layer element -- element sensor layer value -- data layer
type	The type of the attribute value in profile. boolean, integer, float, time, string
value	The allowed values of the attribute (regular expression).
multilanguage	Enable / Disable multilanguage extension When multilanguage="true", this attribute can support multi-languages with the following rule. Language_Dependent_AttrName ::= AttrName " _ " LanguageCode See Appendix G for the available LanguageCode e.g.,) location_jpn, location_eng, location_tha, location_fre
delegation	Delegate to external application domain. (Reserved)
description	The semantics of the attribute. (Reserved)

Section 7: Requirements for developing and deploying new sensors

To develop a new sensor and to use it in Live E! environment;

1. Create a system that uploads data using the web service (DataUpload200703).
2. Apply new sensor information to Live E! authority

For the first requirement, a developer can download the tool from <http://live-e.hongo.wide.ad.jp/DataUpload200703/>

For the second requirement, please contact to live-e[at]mri.co.jp with information about the new type of sensor: i.e., vendor, model, and measurement type. After a small discussion, sensorVendor, sensorModel, and sensorType will be approved and determined in Live E!.

Appendix A: Configuration (livee_config.xml)

livee_config.xml

```

<liveConfiguration xmlns="http://live-e.org/Configuration/2007/03/">
  <mode>master</mode>
  <dbAccess>jdbc:postgresql:livee</dbAccess>
  <master>
    <aggregator>
      <start>2007-07-01T00:00:00.0000000+09:00</start>
      <interval>300</interval>
      <hourAggregator>
        <thresholdTime>70</thresholdTime>
        <thresholdTimeBase>minute</thresholdTimeBase>
      </hourAggregator>
      <dayAggregator>
        <thresholdTime>1</thresholdTime>
        <thresholdTimeBase>day</thresholdTimeBase>
      </dayAggregator>
      <monthAggregator>
        <thresholdTime>1</thresholdTime>
        <thresholdTimeBase>month</thresholdTimeBase>
      </monthAggregator>
    </aggregator>
    <profileSchemaLoader>
      <interval>3600</interval>
    </profileSchemaLoader>
    <childProfileMerger>
      <interval>3600</interval>
      <cacheValidityTime>7200</cacheValidityTime>
    </childProfileMerger>
    <dataManagerProfileMerger>
      <interval>1200</interval>
    </dataManagerProfileMerger>
    <profileChecker>
      <interval>600</interval>
      <errOutputLevel>warning</errOutputLevel>
    </profileChecker>
    <sensorLivingChecker>
      <interval>3600</interval>
      <thresholdTime>3</thresholdTime>
      <thresholdTimeBase>day</thresholdTimeBase>
      <errOutputLevel>warning</errOutputLevel>
    </sensorLivingChecker>
  </master>
  <slave>
    <masterAdminURL>http://live-e2.hongo.wide.ad.jp/axis/services/Admin200703</masterAdminURL>
    <copyAdmin>
      <interval>600</interval>
    </copyAdmin>
    <copyDataManager>
      <masterURL>http://live-e2.hongo.wide.ad.jp/axis/services/DataManagerReplication200703</masterURL>
      <latestDataUpdate>
        <interval>60</interval>
      </latestDataUpdate>
      <profileUpdate>
        <start>2007-07-01T00:00:00.0000000+09:00</start>
        <interval>300</interval>
        <thresholdTime>600</thresholdTime>
        <thresholdTimeBase>second</thresholdTimeBase>
        <scopeTime>300</scopeTime>
        <scopeTimeBase>second</scopeTimeBase>
      </profileUpdate>
      <archiveDataUpdate>
        <start>2007-07-01T00:00:00.0000000+09:00</start>
        <interval>300</interval>
        <thresholdTime>600</thresholdTime>
        <thresholdTimeBase>second</thresholdTimeBase>
        <scopeTime>300</scopeTime>
        <scopeTimeBase>second</scopeTimeBase>
      </archiveDataUpdate>
    </copyDataManager>
  </slave>
  <neighbor>
    <topology>
      <neighbor xmlns="http://live-e.org/Administrator/2007/03/">
        <site type="barant" name="live-e.org">

```


Appendix B: Sensor Administration Command

Use “Authorization” command to register / leave sensors.

```
$ java org.livee.data.command.Authorization
```

Usage: java org.livee.data.command.Authorization COMMAND [OPTIONS]

COMMAND ::= list|add|update|close|restore

OPTIONS ::= (OPTION)* | ε

OPTION ::= -i ID| -p PASSWORD | -o OWNER_NAME | -m OWNER_MAIL_ADDRESS

Five sub commands are provided.

- list (list the registered sensors)
- add (new sensor registration)
 - Required: ID, Password, OwnerName, OwnerMailAddress
- update (sensor entity update including the password)
 - Required: ID; Optional: Password, OwnerName, OwnerMailAddress
- close (stop and leave the sensor from the external link)
 - Required: ID
- restore
 - Required: ID

Appendix D: Profile Management Service

Live E! web service for sensor profile registration and modification.

Web Service: ProfileManagement200703

```
String getAvailableLocales();  
String getProfileSchema();  
String getProfileAll(String tz,String lang);  
String getProfile(String id);  
String setProfile(String data);  
String updatePassword(String id, String old_pass, String new_pass);
```

String getAvailableLocales();

This method gives the list of available language locales.

String getProfileSchema();

This method gives the system profile schema (see Section 6).

String getProfileAll(String tz,String lang);

This method gives the whole sensor profile registered in the server with time zone(tz) and in language (lang). This method can be used for obtaining registered sensor list.

String getProfile(String id);

This method retrieves a registered sensor profile specified by “id”. The retrieved profile format differs from that of DataProvider200703 in that this method gives multi-language profile at the same time (Appendix C). Users can use this method for modifying and updating sensor profile with setProfile method.

String setProfile(String profile);

This method registers sensor profile formatted as in Appendix C.

String updatePassword(String id,String old_passwd,String new_passwd);

This method can be used for updating sensor access password. “old_passwd” and “new_password” must be encoded in BASE64.


```

<?xml version="1.0" encoding="UTF-8" ?>
<sensorGroup authorization="MDAwMDAwMDA=" class="combined" id="live-e.naist.jp/WM918/B206/" xmlns="http://live-
e.org/DataType/2007/03/">
  <sensor id="live-e.naist.jp/WM918/B206/Temperature">
    <value time="2007-06-27T00:00:00.0000000+09:00">25.5</value>
    <value time="2007-06-27T00:10:00.0000000+09:00">25.3</value>
    ...
  </sensor>
  <sensor id="live-e.naist.jp/WM918/B206/Humidity" >
    <value time="2007-06-27T00:00:00.0000000+09:00">56.5</value>
    <value time="2007-06-27T00:10:00.0000000+09:00">56.8</value>
    ...
  </sensor>
  <sensor id="live-e.naist.jp/WM918/B206/Pressure" >
    <value time="2007-06-27T00:00:00.0000000+09:00">1003</value>
    <value time="2007-06-27T00:10:00.0000000+09:00">1006</value>
    ...
  </sensor>
  <sensor id="live-e.naist.jp/WM918/B206/WindDir" >
    <value time="2007-06-27T00:00:00.0000000+09:00">352</value>
    <value time="2007-06-27T00:00:20.0000000+09:00">132</value>
    ...
  </sensor>
  <sensor id="live-e.naist.jp/WM918/B206/WindSpeed" >
    <value time="2007-06-27T00:00:00.0000000+09:00">2.4</value>
    <value time="2007-06-27T00:00:20.0000000+09:00">3.5</value>
    ...
  </sensor>
  <sensor id="live-e.naist.jp/WM918/B206/RainFall" >
    <value time="2007-06-27T00:00:00.0000000+09:00">0.0</value>
    <value time="2007-06-27T00:10:00.0000000+09:00">1.0</value>
    ...
  </sensor>
</sensorGroup>

```

String uploadCollection(String xml)

This method is provided for uploading multiple combined sensor data at the same time in a transaction. Sensor access passwords must be encoded in BASE64 and must be put on each sensor field.

```

<?xml version="1.0" encoding="UTF-8" ?>
<sensorGroup class="collection" xmlns="http://live-e.org/DataType/2007/03/">
  <sensorGroup authorization="MDAwMDAwMDA=" class="combined" id="live-e.naist.jp/WM918/B206/" >
    ...
  </sensorGroup>
  <sensorGroup authorization="MDAwMDAwMDA=" class="combined" id="hongo.wide.ad.jp/WM918/elab/" >
    ...
  </sensorGroup>
  <sensorGroup authorization="MDAwMDAwMDA=" class="combined" id="im.unl.dendai.ac.jp/WM918/roof/" >
    ...
  </sensorGroup>
  ...
</sensorGroup>

```


Appendix F: Global Sensor Data Search/Retrieval

Live E! web service for searching and retrieving sensor data in the global Live E! network.

Web Service: GlobalDataProvider200703

```
String getProfileSchema()
String getProfileAll(String tz,String lang)
String getProfile(String id,String tz,String lang)
String getProfileByAreaRect(double north,double south,double east,double west, String tz,String lang);
String getArchiveProfile(String id,String start,String end,String tz,String lang)
String getLatestDataAll(String tz,String lang)
String getLatestData(String id,String tz,String lang)
String getLatestDataByAreaRect(double north,double south,double east,double west,String tz,String lang);
String getArchiveCombinedData(String id,String start,String end,String tz,String lang)
String getDataHourlyAggregated(String id,String start,String end,String tz,String lang)
String getDataDailyAggregated(String id,String start,String end,String tz,String lang)
String getDataMonthlyAggregated(String id,String start,String end,String tz,String lang)
String search(String query)
```

String getProfileSchema();

This method gives the system sensor profile, see Section 6.

String getProfileAll(String tz,String lang);

This method gives all the sensor profile globally with time zone (tz) in language (lang).

```
<sensorGroup class="collection" xmlns="http://live-e.org/DataType/2007/03/">
  <sensorGroup class="combined" id="hongo.wide.ad.jp/WM918/elab/" latitude="35.65"
longitude="139.345" location="江崎研究室" sensorVendor="AmbientWeather"
sensorModel="WM918" . . . >
  <sensor id="hongo.wide.ad.jp/WM918/elab/Temperature" sensorType="Temperature" />
  <sensor id="hongo.wide.ad.jp/WM918/elab/Humidity" sensorType="Humidity" />
  <sensor id="hongo.wide.ad.jp/WM918/elab/Pressure" sensorType="Pressure" />
  . . . sensor . . .
</sensorGroup>
. . . sensorGroup class="combined" . . .
</sensorGroup>
```

E.g., getProfileAll("JST","Japanese");

String getProfile(String id,String tz,String lang);

This method retrieves and gives specified sensor profile by "id" with time zone(tz) in language (lang).

```
<sensorGroup class="combined" id="hongo.wide.ad.jp/WM918/elab/" latitude="35.65"
longitude="139.345" location="Esaki Laboratory" sensorVendor="AmbientWeather"
sensorModel="WM918" . . . xmlns="http://live-e.org/DataType/2007/03/" >
  <sensor id="hongo.wide.ad.jp/WM918/elab/Temperature" sensorType="Temperature" />
  <sensor id="hongo.wide.ad.jp/WM918/elab/Humidity" sensorType="Humidity" />
  <sensor id="hongo.wide.ad.jp/WM918/elab/Pressure" sensorType="Pressure" />
  . . . sensor の繰り返し . . .
</sensorGroup>
```

E.g., getProfile("hongo.wide.ad.jp/WM918/elab/","JST","English");

String getProfileByAreaRect(double north,double south,double east,double west, String tz,String lang);

This method retrieves sensor profile with geographical region specification [south,north]x[west,east]. North Pole: +90, South Pole: -90, 180°E: +180, and 180°W: -180.

String getArchiveProfile(String id, String start, String end, String tz, String lang)

This method retrieves sensor profile update logs with time specification[start,end]. “start” and “end” must be formatted with W3CTimestamp expression.

```
<sensorGroup class="collection" xmlns="http://live-e.org/DataType/2007/03/">
  <sensorGroup class="combined"
    created="2007-07-25T00:52:55.00000000+09:00"
    expired="2007-07-25T06:09:59.00000000+09:00"
    id="hongo.wide.ad.jp/WM918/elab/" latitude="35.65" longitude="139.345"
    location="江崎研究室"
    sensorVendor="AmbientWeather" sensorModel="WM918" . . . >
    <sensor created="2007-07-25T00:52:55.00000000+09:00"
      expired="2007-07-25T06:09:59.00000000+09:00"
      id="hongo.wide.ad.jp/WM918/elab/Temperature" sensorType="Temperature" />
    . . . sensor . . .
  </sensorGroup>
  <sensorGroup class="combined"
    created="2007-07-25T06:09:59.00000000+09:00"
    expired="2007-07-30T06:30:58.00000000+09:00"
    id="hongo.wide.ad.jp/WM918/elab/" latitude="35.65" longitude="139.345"
    location="江崎研究室(工学部 2 号館)" sensorVendor="AmbientWeather"
    sensorModel="WM918" . . . >
    <sensor created="2007-07-25T06:09:59.00000000+09:00"
      expired="2007-07-30T06:30:58.00000000+09:00"
      id="hongo.wide.ad.jp/WM918/elab/Temperature" sensorType="Temperature" />
    . . . sensor . . .
  </sensorGroup>
  . . . sensorGroup class="combined" . . .
</sensorGroup>
```

E.g., getArchiveProfile("hongo.wide.ad.jp/WM918/elab/",
"2007-01-01T00:00:00.00000000+09:00",
"2007-08-01T00:00:00.00000000+09:00",
"JST","Japanese")

String getLatestDataAll(String tz,String lang)

This method gives all the latest sensor data with time zone (tz) in language (lang).

```

<sensorGroup class="collection" xmlns="http://live-e.org/DataType/2007/03/">
  <sensorGroup class="combined" id="hongo.wide.ad.jp/WM918/elab/" latitude="35.65"
longitude="139.345" location="江崎研究室" sensorVendor="AmbientWeather"
sensorModel="WM918" . . . >
    <sensor id="hongo.wide.ad.jp/WM918/elab/Temperature" sensorType="Temperature">
      <value time="2007-07-30T12:34:23.0000000+09:00">26.7</value>
    </sensor>
    <sensor id="hongo.wide.ad.jp/WM918/elab/Humidity" sensorType="Humidity">
      <value time="2007-07-30T12:34:23.0000000+09:00">57.2</value>
    </sensor>
    . . . sensor . . .
  </sensorGroup>
  . . . sensorGroup class="combined" . . .
</sensorGroup>

```

E.g., `getLatestDataAll("JST","Japanese");`

String getLatestData(String id,String tz,String lang)

This method gives the latest sensor data of specified sensor (id) with time zone (tz) in language (lang).

String getLatestDataByAreaRect(double north, double south, double east, double west, String tz, String lang)

This method retrieves the latest sensor data with geographical region specification [south,north]x[west,east]. North Pole: +90, South Pole: -90, 180°E: +180, and 180°W: -180.

String getArchiveCombinedData(String id, String start, String end, String tz,String lang)

This method retrieves raw sensor data archive with time specification[start,end]. "start" and "end" must be formatted with W3CTimestamp expression.

```

<sensorGroup class="combined" id="hongo.wide.ad.jp/WM918/elab/" latitude="35.65"
longitude="139.345" location="江崎研究室" sensorVendor="AmbientWeather"
sensorModel="WM918" . . . xmlns="http://live-e.org/DataType/2007/03/">
  <sensor id="hongo.wide.ad.jp/WM918/elab/Temperature" sensorType="Temperature">
    <value time="2007-07-30T00:00:00.0000000+09:00">26.7</value>
    <value time="2007-07-30T00:01:00.0000000+09:00">26.7</value>
    <value time="2007-07-30T00:02:00.0000000+09:00">26.8</value>
    <value time="2007-07-30T00:03:00.0000000+09:00">26.8</value>
    . . . value . . .
  </sensor>
  . . . sensor . . .
</sensorGroup>

```

E.g., `getArchiveCombinedData("hongo.wide.ad.jp/WM918/elab/",
"2007-07-30T00:00:00.0000000+09:00",
"2007-07-30T01:00:00.0000000+09:00",
"JST","Japanese")`

String getDataHourlyAggregated(String id, String start, String end, String tz, String lang)

This method retrieves hourly aggregated sensor data with time specification[start,end]. “start” and “end” must be formatted with W3CTimestamp expression. Aggregated values are maximum, minimum and average values in a region.

```
<sensorGroup class="combined" id="hongo.wide.ad.jp/WM918/elab/" latitude="35.65"
longitude="139.345" location="江崎研究室" sensorVendor="AmbientWeather"
sensorModel="WM918" . . . xmlns="http://live-e.org/DataType/2007/03/">
  <sensor id="hongo.wide.ad.jp/WM918/elab/Temperature" sensorType="Temperature">
    <aggValue aggType="avg" timeScope="hour"
      time="2007-07-28T06:00:00.0000000+09:00" >26.7</aggValue>
    <aggValue aggType="max" timeScope="hour"
      time="2007-07-28T06:00:00.0000000+09:00" >27.3</aggValue>
    <aggValue aggType="min" timeScope="hour"
      time="2007-07-28T06:00:00.0000000+09:00" >25.5</aggValue>
    <aggValue aggType="sum" timeScope="hour"
      time="2007-07-28T06:00:00.0000000+09:00" >1602</aggValue>
    <aggValue aggType="count" timeScope="hour"
      time="2007-07-28T06:00:00.0000000+09:00" >60</aggValue>
    . . . aggValue . . .
  </sensor>
  . . . sensor . . .
</sensorGroup>
```

E.g., `getDataHourlyAggregated("hongo.wide.ad.jp/WM918/elab/", "2007-07-28T06:00:00.0000000+09:00", "2007-07-28T09:00:00.0000000+09:00", "JST", "Japanese")`

String getDataDailyAggregated(String id, String start, String end, String tz, String lang)

This method retrieves daily aggregated sensor data with time specification[start,end]. “start” and “end” must be formatted with W3CTimestamp expression. Aggregated values are maximum, minimum and average values in a region.

String getDataMonthlyAggregated(String id, String start, String end, String tz, String lang)

This method retrieves monthly aggregated sensor data with time specification[start,end]. “start” and “end” must be formatted with W3CTimestamp expression. Aggregated values are maximum, minimum and average values in a region.

String search(String query)

This method searches servers that match “query”.

Appendix G: Available Language Locales

Available language locales in August, 2007.

These are based on ISO639-2.

http://www.loc.gov/standards/iso639-2/php/code_list.php

Language Code	Spelling
jpn	Japanese
eng	English
fre	French
ger	German
kor	Korean
chi	Chinese
tha	Thai

Live E! is going to append locales in the near future.

Appendix H: Available Time Zones

Etc/GMT+12	America/Denver	America/Detroit
Etc/GMT+11	America/Edmonton	America/Eirunepe
MIT	America/Hermosillo	America/Fort_Wayne
Pacific/Apia	America/Inuvik	America/Grand_Turk
Pacific/Midway	America/Mazatlan	America/Guayaquil
Pacific/Niue	America/Phoenix	America/Havana
Pacific/Pago_Pago	America/Shiprock	America/Indiana/Indianapolis
Pacific/Samoa	America/Yellowknife	America/Indiana/Knox
US/Samoa	Canada/Mountain	America/Indiana/Marengo
America/Adak	Etc/GMT+7	America/Indiana/Vevay
America/Atka	MST	America/Indianapolis
Etc/GMT+10	MST7MDT	America/Iqaluit
HST	Mexico/BajaSur	America/Jamaica
Pacific/Fakaofu	Navajo	America/Kentucky/Louisville
Pacific/Honolulu	PNT	America/Kentucky/Monticello
Pacific/Johnston	SystemV/MST7	America/Knox_IN
Pacific/Rarotonga	SystemV/MST7MDT	America/Lima
Pacific/Tahiti	US/Arizona	America/Louisville
SystemV/HST10	US/Mountain	America/Montreal
US/Aleutian	America/Belize	America/Nassau
US/Hawaii	America/Cancun	America/New_York
Pacific/Marquesas	America/Chicago	America/Nipigon
AST	America/Costa_Rica	America/Panama
America/Anchorage	America/El_Salvador	America/Pangnirtung
America/Juneau	America/Guatemala	America/Port-au-Prince
America/Nome	America/Managua	America/Porto_Acre
America/Yakutat	America/Menominee	America/Rio_Branco
Etc/GMT+9	America/Merida	America/Thunder_Bay
Pacific/Gambier	America/Mexico_City	America/Toronto
SystemV/YST9	America/Monterrey	Brazil/Acre
SystemV/YST9YDT	America/North_Dakota/Center	Canada/Eastern
US/Alaska	America/Rainy_River	Cuba
America/Dawson	America/Rankin_Inlet	EST
America/Ensenada	America/Regina	EST5EDT
America/Los_Angeles	America/Swift_Current	Etc/GMT+5
America/Tijuana	America/Tegucigalpa	IET
America/Vancouver	America/Winnipeg	Jamaica
America/Whitehorse	CST	SystemV/EST5
Canada/Pacific	CST6CDT	SystemV/EST5EDT
Canada/Yukon	Canada/Central	US/East-Indiana
Etc/GMT+8	Canada/East-Saskatchewan	US/Eastern
Mexico/BajaNorte	Canada/Saskatchewan	US/Indiana-Starke
PST	Chile/EasterIsland	US/Michigan
PST8PDT	Etc/GMT+6	America/Anguilla
Pacific/Pitcairn	Mexico/General	America/Antigua
SystemV/PST8	Pacific/Easter	America/Aruba
SystemV/PST8PDT	Pacific/Galapagos	America/Asuncion
US/Pacific	SystemV/CST6	America/Barbados
US/Pacific-New	SystemV/CST6CDT	America/Boa_Vista
America/Boise	US/Central	America/Campo_Grande
America/Cambridge_Bay	America/Bogota	America/Caracas
America/Chihuahua	America/Cayman	America/Cuiaba
America/Dawson_Creek	America/Coral_Harbour	America/Curacao

America/Dominica	America/Cordoba	Etc/Universal
America/Glace_Bay	America/Fortaleza	Etc/Zulu
America/Goose_Bay	America/Godthab	Europe/Belfast
America/Grenada	America/Jujuy	Europe/Dublin
America/Guadeloupe	America/Maceio	Europe/Lisbon
America/Guyana	America/Mendoza	Europe/London
America/Halifax	America/Miquelon	GB
America/La_Paz	America/Montevideo	GB-Eire
America/Manaus	America/Paramaribo	GMT
America/Martinique	America/Recife	GMT0
America/Montserrat	America/Rosario	Greenwich
America/Port_of_Spain	America/Sao_Paulo	Iceland
America/Porto_Velho	Antarctica/Rothera	Portugal
America/Puerto_Rico	BET	UCT
America/Santiago	Brazil/East	UTC
America/Santo_Domingo	Etc/GMT+3	Universal
America/St_Kitts	America/Noronha	WET
America/St_Lucia	Atlantic/South_Georgia	Zulu
America/St_Thomas	Brazil/DeNoronha	Africa/Algiers
America/St_Vincent	Etc/GMT+2	Africa/Bangui
America/Thule	America/Scoresbysund	Africa/Brazzaville
America/Tortola	Atlantic/Azores	Africa/Ceuta
America/Virgin	Atlantic/Cape_Verde	Africa/Douala
Antarctica/Palmer	Etc/GMT+1	Africa/Kinshasa
Atlantic/Bermuda	Africa/Abidjan	Africa/Lagos
Atlantic/Stanley	Africa/Accra	Africa/Libreville
Brazil/West	Africa/Bamako	Africa/Luanda
Canada/Atlantic	Africa/Banjul	Africa/Malabo
Chile/Continental	Africa/Bissau	Africa/Ndjamena
Etc/GMT+4	Africa/Casablanca	Africa/Niamey
PRT	Africa/Conakry	Africa/Porto-Novo
SystemV/AST4	Africa/Dakar	Africa/Tunis
SystemV/AST4ADT	Africa/El_Aaiun	Africa/Windhoek
America/St_Johns	Africa/Freetown	Arctic/Longyearbyen
CNT	Africa/Lome	Atlantic/Jan_Mayen
Canada/Newfoundland	Africa/Monrovia	CET
AGT	Africa/Nouakchott	ECT
America/Araguaina	Africa/Ouagadougou	Etc/GMT-1
America/Argentina/Buenos_Aires	Africa/Sao_Tome	Europe/Amsterdam
America/Argentina/Catamarca	Africa/Timbuktu	Europe/Andorra
America/Argentina/ComodRivadavia	America/Danmarkshavn	Europe/Belgrade
America/Argentina/Cordoba	Atlantic/Canary	Europe/Berlin
America/Argentina/Jujuy	Atlantic/Faeroe	Europe/Bratislava
America/Argentina/La_Rioja	Atlantic/Madeira	Europe/Brussels
America/Argentina/Mendoza	Atlantic/Reykjavik	Europe/Budapest
America/Argentina/Rio_Gallegos	Atlantic/St_Helena	Europe/Copenhagen
America/Argentina/San_Juan	Eire	Europe/Gibraltar
America/Argentina/Tucuman	Etc/GMT	Europe/Ljubljana
America/Argentina/Ushuaia	Etc/GMT+0	Europe/Luxembourg
America/Bahia	Etc/GMT-0	Europe/Madrid
America/Belem	Etc/GMT0	Europe/Malta
America/Buenos_Aires	Etc/Greenwich	Europe/Monaco
America/Catamarca	Etc/UCT	Europe/Oslo
America/Cayenne	Etc/UTC	Europe/Paris

Europe/Prague	Europe/Tallinn	Asia/Bishkek
Europe/Rome	Europe/Tiraspol	Asia/Dushanbe
Europe/San_Marino	Europe/Uzhgorod	Asia/Karachi
Europe/Sarajevo	Europe/Vilnius	Asia/Oral
Europe/Skopje	Europe/Zaporozhye	Asia/Samarkand
Europe/Stockholm	Israel	Asia/Tashkent
Europe/Tirane	Libya	Asia/Yekaterinburg
Europe/Vaduz	Turkey	Etc/GMT-5
Europe/Vatican	Africa/Addis_Ababa	Indian/Kerguelen
Europe/Vienna	Africa/Asmera	Indian/Maldives
Europe/Warsaw	Africa/Dar_es_Salaam	PLT
Europe/Zagreb	Africa/Djibouti	Asia/Calcutta
Europe/Zurich	Africa/Kampala	IST
MET	Africa/Khartoum	Asia/Katmandu
Poland	Africa/Mogadishu	Antarctica/Mawson
ART	Africa/Nairobi	Antarctica/Vostok
Africa/Blantyre	Antarctica/Syowa	Asia/Almaty
Africa/Bujumbura	Asia/Aden	Asia/Colombo
Africa/Cairo	Asia/Baghdad	Asia/Dacca
Africa/Gaborone	Asia/Bahrain	Asia/Dhaka
Africa/Harare	Asia/Kuwait	Asia/Novosibirsk
Africa/Johannesburg	Asia/Qatar	Asia/Omsk
Africa/Kigali	Asia/Riyadh	Asia/Qyzylorda
Africa/Lubumbashi	Asia/Tbilisi	Asia/Thimbu
Africa/Lusaka	EAT	Asia/Thimphu
Africa/Maputo	Etc/GMT-3	BST
Africa/Maseru	Europe/Moscow	Etc/GMT-6
Africa/Mbabane	Indian/Antananarivo	Indian/Chagos
Africa/Tripoli	Indian/Comoro	Asia/Rangoon
Asia/Amman	Indian/Mayotte	Indian/Cocos
Asia/Beirut	W-SU	Antarctica/Davis
Asia/Damascus	Asia/Riyadh87	Asia/Bangkok
Asia/Gaza	Asia/Riyadh88	Asia/Hovd
Asia/Istanbul	Asia/Riyadh89	Asia/Jakarta
Asia/Jerusalem	Mideast/Riyadh87	Asia/Krasnoyarsk
Asia/Nicosia	Mideast/Riyadh88	Asia/Phnom_Penh
Asia/Tel_Aviv	Mideast/Riyadh89	Asia/Pontianak
CAT	Asia/Tehran	Asia/Saigon
EET	Iran	Asia/Vientiane
Egypt	Asia/Baku	Etc/GMT-7
Etc/GMT-2	Asia/Dubai	Indian/Christmas
Europe/Athens	Asia/Muscat	VST
Europe/Bucharest	Asia/Yerevan	Antarctica/Casey
Europe/Chisinau	Etc/GMT-4	Asia/Brunei
Europe/Helsinki	Europe/Samara	Asia/Chongqing
Europe/Istanbul	Indian/Mahe	Asia/Chungking
Europe/Kaliningrad	Indian/Mauritius	Asia/Harbin
Europe/Kiev	Indian/Reunion	Asia/Hong_Kong
Europe/Mariehamn	NET	Asia/Irkutsk
Europe/Minsk	Asia/Kabul	Asia/Kashgar
Europe/Nicosia	Asia/Aqtau	Asia/Kuala_Lumpur
Europe/Riga	Asia/Aqtobe	Asia/Kuching
Europe/Simferopol	Asia/Ashgabat	Asia/Macao
Europe/Sofia	Asia/Ashkhabad	Asia/Macau

Asia/Makassar	Pacific/Saipan
Asia/Manila	Pacific/Truk
Asia/Shanghai	Pacific/Yap
Asia/Singapore	Australia/LHI
Asia/Taipei	Australia/Lord_Howe
Asia/Ujung_Pandang	Asia/Magadan
Asia/Ulaanbaatar	Etc/GMT-11
Asia/Ulan_Bator	Pacific/Efate
Asia/Urumqi	Pacific/Guadalcanal
Australia/Perth	Pacific/Kosrae
Australia/West	Pacific/Noumea
CTT	Pacific/Ponape
Etc/GMT-8	SST
Hongkong	Pacific/Norfolk
PRC	Antarctica/McMurdo
Singapore	Antarctica/South_Pole
Asia/Choibalsan	Asia/Anadyr
Asia/Dili	Asia/Kamchatka
Asia/Jayapura	Etc/GMT-12
Asia/Pyongyang	Kwajalein
Asia/Seoul	NST
Asia/Tokyo	NZ
Asia/Yakutsk	Pacific/Auckland
Etc/GMT-9	Pacific/Fiji
JST	Pacific/Funafuti
Japan	Pacific/Kwajalein
Pacific/Palau	Pacific/Majuro
ROK	Pacific/Nauru
ACT	Pacific/Tarawa
Australia/Adelaide	Pacific/Wake
Australia/Broken_Hill	Pacific/Wallis
Australia/Darwin	NZ-CHAT
Australia/North	Pacific/Chatham
Australia/South	Etc/GMT-13
Australia/Yancowinna	Pacific/Enderbury
AET	Pacific/Tongatapu
Antarctica/DumontDUrville	Etc/GMT-14
Asia/Sakhalin	Pacific/Kiritimati
Asia/Vladivostok	
Australia/ACT	
Australia/Brisbane	
Australia/Canberra	
Australia/Currie	
Australia/Hobart	
Australia/Lindeman	
Australia/Melbourne	
Australia/NSW	
Australia/Queensland	
Australia/Sydney	
Australia/Tasmania	
Australia/Victoria	
Etc/GMT-10	
Pacific/Guam	
Pacific/Port_Moresby	

