

Live E! サーバ運用チュートリアル

作成: 2007年08月22日

最終更新: 2010年12月06日

Live E! 技術WG

概要

このドキュメントは2007年8月から運用を開始した広域センサネットワーク(Live E!システム)の概要について述べたものである。新規Live E!システムでは、複数のサーバによる連携、すなわち分散運用が可能になる。分散運用により組織単位での自由なセンサシステム運用が行われると共に、各組織で集められたセンサデータは全体で共有され、流通が可能なシステムとなる。このドキュメントでは、新規Live E!システムの概要を示し、サーバのインストール方法、センサ登録と運用方法、そして、新規センサの開発方法について述べる。

第1章 Live E! システム概要

新規Live E! システムは、複数のLive E!サーバから構成され、これらが自律的に結合し、センサデータが全体で共有可能となっている。Live E! システムは、数々のセンサ運用組織から構成され、センサデータは、各組織で運用するLive E!サーバにアップロードされ保管される。グローバルなデータ検索は、いずれかのLive E!サーバ(通常は各組織で運用する自サーバ)にクエリを発行することで行われ、センサアプリケーションにデータを提供することが可能となっている。

Live E! サーバを導入するには、まず (1)Live E!公式サイトからソフトウェアをダウンロード後インストールし、(2)Live E!サーバツリーのどこかと論理的な結合を作成することで行われる。詳細については第2章で述べる。

Live E! システムのセンサは、サーバを運用する者により登録される。センサにはサーバに登録されることで、アクセスパスワードが発行され、センサプロファイルの登録、およびデータアップロードの認証が行われる。プロファイル登録およびデータアップロードはサーバが提供する専用のSOAPインタフェースにより行われる。センサの認証は、非意図的な誤りによって生じるエラーを防ぐために導入されたセキュリティ機構である。センサ登録システムおよびアップロード機構の詳細は、第3章で述べる。

グローバルなセンサ検索とデータ読出しは、すべてのLive E!サーバに備わった機能で、センサアプリケーションは、任意のサーバにクエリを発行することで、データを取得することが可能である。クエリを受け付けたサーバは、バックにあるサーバ群から適切なサーバを選び出し、データを解決し、提供する。検索システムについては、第4章で述べる。

新規Live E! システムは、自動的な同期機構によりデータおよびサービスの冗長化が実現されている。冗長運用の方法は、第5章で述べる。

Live E! システムでは、センサアプリケーションレベルでの相互接続性を実現するために、アプリケーション・オブジェクトを管理し、プロファイル一貫性を確保する機構が設けられている。第6章ではプロファイル一貫性の実現機構について述べる。

第7章では、新規センサを開発し、Live E! に導入する場合の注意事項について述べる。新規センサに関連する情報は、新しいアプリケーション・オブジェクトとして承認され、システム全体で共有される必要があり、Live E! Authorityに登録申請を必要とする。

第2章 サーバ・インストール

Live E! サーバのインストールは、次のステップで行われる。

- (1) 基本コンポーネントのインストール
- (2) Live E!コンポーネントのインストール
- (3) Live E!サーバの設定

本書は、Ubuntu 10.04 上にLive E!のサーバをインストールした事例を元に構成されている。下記のマニュアルは、大学院生レベルのLinux知識でLive E!サーバを立ち上げられるように書いてある。Ubuntu 10.04はLong Time Support (LTS) であり、もっとも相応しいLinux ディストリビューションであると考えている。

実際のインストール事例を元に構成されているため、本書に記載されているソフトウェア・バージョンが最新でない場合があるかもしれない。そのような場合は、適宜、最新バージョンのソフトウェアを利用して構わないが、詳細に渡る変更については、各自で対応してもらいたい。

なお、Live E!技術WGでは、他のOSやLinuxディストリビューションでのインストール事例も集めたいと考えている。Windowsでのインストールを挑戦した方などは、インストールの経過を是非、live-e-tech @ hongo.wide.ad.jp にレポートしていただければと思う。

2.1. 基本コンポーネントのインストール

Live E! サーバに必要な基本コンポーネントのインストール。

- JDK6: Java Developers Kit, Java開発ツール一式
- Apache2.2: Webサーバ
- Tomcat 6: Webサーブレットエンジン
- Axis 1.4: SOAP Webサービスエンジン
- PostgreSQL 8.4: データベース管理システム

2.1.1. JDK6のインストール

```
$ sudo add-apt-repository 'deb http://archive.canonical.com/ lucid partner'  
$ sudo apt-get update  
$ sudo apt-get install sun-java6-jdk
```

2.1.2. Apache2のインストール

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

2.1.3. Tomcat6のインストール

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

ブラウザで<http://localhost:8080/>もしくは<http://localhost:8180/>にアクセスし、Tomcatのページが表示されることを確認。

注) "localhost"の部分は適宜、ネットワーク環境に応じて設定のこと。

2.1.4. Axis1.4のインストール

<http://ws.apache.org/axis/>から、axis-bin-1_4.tar.gzをダウンロード後展開。

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

/usr/local/axis/webapps/axisを/var/lib/tomcat6/webapps/にコピー後、パーミッション設定。

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

Tomcat再起動

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

Axis実行確認

ブラウザで<http://localhost:8080/axis/>もしくは<http://localhost:8180/axis/>にアクセス。Apache-Axisのページが表示されることを確認。

注) "localhost"の部分は適宜、ネットワーク環境に応じて設定のこと。

各種JARファイルのインストール

注)I, IIの操作を行う際は、Downloadの際に利用規約に同意する事を求められるのでGUIを用いFirefoxなどのwebブラウザを使用することが好ましい。

I. JavaBeans Activation Framework 1.1 (activation.jar)

<http://java.sun.com/products/javabeans/jaf/downloads/index.html>のページをたどり、jaf-1_1_fr.zipをダウンロード。展開して得られるactivation.jarを/var/lib/tomcat6/webapps/axis/WEB-INF/libにコピー。

```
$ sudo cp activation.jar /var/lib/tomcat6/webapps/axis/WEB-INF/lib
```

II. JavaMail API 1.4 (mail.jar)

<http://java.sun.com/products/javamail/downloads/index.html>のページをたどり、javamail-1.4.3.zipをダウンロード。展開して得られるmail.jarを /var/lib/

tomcat6/webapps/axis/WEB-INF/lib にコピー.

```
$ sudo cp mail.jar /var/lib/tomcat6/webapps/axis/WEB-INF/lib
```

III. XML Security (xmlsec.jar)

<http://santuario.apache.org/dist/java-library/> から, xml-security-bin-1_4_3.zipをダウンロード. 展開後得られるlib/xmlsec-1.4.3.jar を /var/lib/tomcat6/webapps/axis/WEB-INF/lib にコピー.

```
$ cd
$ wget http://santuario.apache.org/dist/java-library/xml-security-bin-1_4_3.zip
$ unzip xml-security-bin-1_4_3.zip
$ cd xml-security-bin-1_4_3/libs
$ sudo cp xmlsec-1.4.3.jar /var/lib/tomcat6/webapps/axis/WEB-INF/lib
```

注) unzipコマンドはインストールされていない可能性がある. その場合は,

```
$ sudo apt-get install unzip
を実行するとインストールできる
```

Tomcatを再起動する.

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

<http://localhost:8080/axis/>及び <http://localhost:8180/axis/> の検証画面に進み, 正しくインストールされたことを確認.

注) "localhost"の部分は適宜, ネットワーク環境に応じて設定のこと.

2.1.5. Apache Tomcatコネクタ(mod_proxy, mod_proxy_ajp)の有効化

TomcatとApache2.2を接続するために, mod_proxy, mod_proxy_ajpを有効にする.
/etc/apache2/mods-available/以下にproxy_ajp.confを作り, 以下の設定を書き込む.

```
$ sudo vi /etc/apache2/mods-available/proxy_ajp.conf
<Location /axis/>
    ProxyPass ajp://localhost:8009/axis/
    Order allow,deny
    allow from all
</Location>
```

/etc/apache2/site-available/defaultに以下の設定を追加.

```
$ sudo vi /etc/apache2/sites-available/default
<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>
. . .
```

```
</VirtualHost>
```

/var/lib/tomcat6/conf/server.xmlの以下の行のコメントアウトを外す.

```
$ sudo vi /var/lib/tomcat6/conf/server.xml
<!--
  <Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
-->
```

を以下のように変更.

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

mod_proxy, mod_proxy_ajpの読み込み

```
$sudo a2enmod proxy_ajp
```

Apache2再起動

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

Tomcat再起動

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

Apache経由でのAxis実行確認

ブラウザでhttp://localhost/axis/にアクセス、Apache-Axisのページが表示されることを確認.

注) "localhost"の部分は適宜、ネットワーク環境に応じて設定のこと.

2.1.6. PostgreSQL8.4のインストール

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

PostgreSQLの設定

/etc/postgresql/8.4/main/pg_hba.conf 末尾を以下のように変更. (indent sameuser および md5を trustに書き換える)

```
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
```

/etc/postgresql/8.4/main/postgresql.confの編集

```
$ sudo vi /etc/postgresql/8.4/main/postgresql.conf
# listen_address = 'localhost'
```

を,

```
listen_address = '*'
```

に変更する.

PostgreSQLを再起動

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

PostgreSQL JDBC Driver のインストール

<http://jdbc.postgresql.org/download.html>から、`postgresql-8.4-702.jdbc4.jar`をダウンロード後、`/var/lib/tomcat6/webapps/axis/` にコピー。

```
$ cd
$ wget http://jdbc.postgresql.org/download/postgresql-8.4-702.jdbc4.jar
$ sudo cp postgresql-8.4-702.jdbc4.jar /var/lib/tomcat6/webapps/axis/
```

Tomcatを再起動

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

2.2. Live E! コンポーネントのインストール

Live E! サーバ・コンポーネントを<http://live-e2.hongo.wide.ad.jp/dist/> からダウンロードし、インストールする。インストールには環境変数の設定を含む。パッケージ(`live-e-datamanager-0.9.x.zip`)のファイル構成を下記に示す。

```
live-e-datamanager-0.9.x
+ bin -- デーモンの起動 および 操作コマンド
+ lib -- JARライブラリ
| + live-e-datamanager-0.9.x.jar
+ conf -- 設定関係
| + env.vars
| + livee_config.xml
| + deploy_masterLiveE.wsdd
| + deploy_slaveLiveE.wsdd
| + undeploy_masterLiveE.wsdd
| + undeploy_slaveLiveE.wsdd
+ schema
| + live-e-datamanager-0.9.sql
+ log -- ログ出力
+ sample
  + profile_template.xml -- テンプレート・プロファイル
```

2.2.1. ファイルの展開

パッケージファイルを下記のように展開し、各種環境変数の設定を行う。

```
$ cd
$ wget http://live-e2.hongo.wide.ad.jp/dist/live-e-datamanager-0.9.x.zip
$ cd /usr/local/
$ sudo unzip ~/live-e-datamanager-0.9.x.zip
$ sudo ln live-e-datamanager-0.9.x livee
展開後、logのパーミッションを設定する。
$ sudo chown -R tomcat6.nogroup /usr/local/livee/log
```

2.2.2. 環境変数の設定

必要な環境変数設定スクリプトは、`/usr/local/livee/conf/env.vars` に置かれている。まず、このファイルを編集し、バージョン番号やファイルパスなどを整える。その後、各種実行プログラムの開始前に読み込ませることで、Live E!サーバが動作するようになる。各種実行プログラムには、Live E!デーモン、Tomcat6、ターミナルが含まれる。

設定の確認

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

を実行し、Axis, Tomcat, Live E!, Javaのディレクトリパスや各JARライブラリのバージョン番号を上記までに行ったインストールの環境に合わせて変更する。なお、本チュートリアルに基づきインストールを行うと以下のように書き換える。

```
$sudo vi /usr/local/livee/conf/env.vars
export JAVA_HOME=/usr/lib/jvm/java-6-sun/
export CATALINA_HOME=/var/lib/tomcat6
export AXIS_HOME=/usr/local/axis
export LIVEE_HOME=/usr/local/livee
export LIVEE_JAR_NAME=live-e-datamanager-0.9.x.jar
export CLASSPATH=.: "$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"
```

Tomcat6 への環境のインポート

`/etc/init.d/tomcat6`を編集する。

```
$ sudo vi /etc/init.d/tomcat6
. /lib/lsb/init-functions
```

```
if [ -r /etc/default/rcS ]; then
    . /etc/default/rcS
```

```
fi
```

の下に以下の1行を追加

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

ターミナルへのインポート

```
$ sudo vi /etc/profile
一行目に
```

```
. /usr/local/livee/conf/env.vars
と追記。
```

2.2.3. 環境変数の確認

```
$ source /etc/profile
を実行し、環境変数を読み込む。
```

CLASSPATH変数の確認

```
$ 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
```

JAVA_HOME変数の確認

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

LIVEE_HOME変数の確認

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

2.2.4. Live E! データベースの作成

Live E! サービスのバックエンドで動くデータベースを作成し、スキーマファイルを読み込ませる。

データベース作成 & スキーマ読み込み

```
$ sudo su postgres
postgres@ $ createdb livee
postgres@ $ cd /usr/local/livee/schema
postgres@ $ psql livee -f live-e-datamanager-0.9.sql
... メッセージが流れる ...
postgres@ $ exit
$
```

2.2.5. AxisへのLive E!コンポーネントのインストール

次に、AxisからLive E!のJARコンポーネントを参照できるように、リンクを作成する。

```
$ cd /var/lib/tomcat6/webapps/axis/WEB-INF/lib/
$ sudo ln -s /usr/local/livee/lib/live-e-datamanager-0.9.x.jar live-e-datamanager.jar
```

2.2.6. AxisでのLive E!サービスのデプロイ

提供するサービスの種類は、Masterサーバとして動作させる場合と、Slaveサーバとして動作させる場合とで異なる。下記のように別々の設定を行うこと。

Masterサーバとは、運用組織の中で核となるサーバのことで、Slaveサーバは冗長化目的のために設置され、Masterサーバと同期するサーバのことである。初めてLive E!サーバを設置する場合は、基本的にMasterサーバである。詳細は「2.3 Live E!サーバの設定と起

動」にて解説する。

Masterサーバとして動作させる場合

```
$ cd /usr/local/livee/conf/  
$ java org.apache.axis.client.AdminClient -p 8180 deploy_masterLiveE.wsdd
```

Slaveサーバとして動作させる場合

```
$ cd /usr/local/livee/conf/  
$ java org.apache.axis.client.AdminClient -p 8180 deploy_slaveLiveE.wsdd
```

注) Ubuntu パッケージでTomcatをインストールした場合に -p 8180を指定する。
Tomcatのportが8080の際は-p 8080を代わりに用いる。

2.3. Live E! サーバの設定と起動

Live E! サーバの設定は、livee_config.xmlに書かれている。詳細は、Appendix Aに記し、ここではMasterサーバのみによる運用に必須な内容として、Live E!サーバツリーの概要および運用ポリシーと、近隣トポロジ情報の設定方法を解説している。

設定ファイルはXMLで書かれており、近隣トポロジ情報は、下記の階層にある。

```
c:liveeConfiguration  
  + c:neighbor  
    + c:topology  
      + a:neighbor
```

ここで、

```
xmlns:c="http://live-e.org/Configuration/2007/03/"  
xmlns:a="http://live-e.org/Administrator/2007/03/"  
である。
```

Live E! サーバの名前付け規則はDNSによく似ている。ルートサーバは、ドット(.)で表現され、ルート直下のサーバには、jp. やtw. のように、名前を割り当てる(国を表すコードが望ましい)。それぞれのサーバは全体のLive E!サーバツリーのサブ・ツリーの運用責任を持ち、子サーバには、hoge.jp. や、hogehoge.jp などの名前を(jp.が)割り当てることにする。Live E! システムでは、それぞれの名前指定される運用単位のことをサイトと呼ぶ。それぞれのサイトでは、複数のサーバを用いて、サービスやデータをmaster/slave方式で冗長化することが可能。自ノードに対して、親の関係にあるサイトをparentサイト、子の関係にあるサイトをchildサイトと呼ぶ。自ノードおよび、冗長ノードで構成されるサイトのことをcolleagueサイトと呼び、siblingの関係にあるサイトと区別している(図1)。

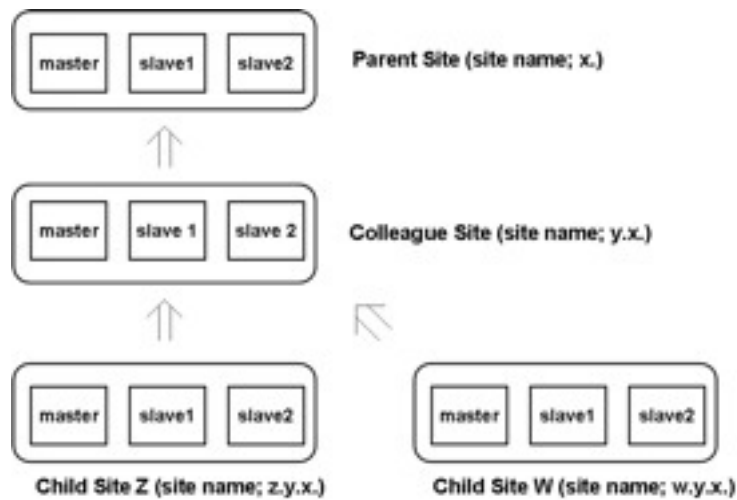


図1: 近隣トポロジ情報

サーバが、図1の関係にあるとき、設定は図2のようになる。

```

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

```

図2: 近隣トポロジ情報の設定

実際、図3のトポロジ構成でシステムを構築した場合、fr.でのトポロジ設定、および、fire04.fr.でのトポロジ設定を図4および図5に記す。

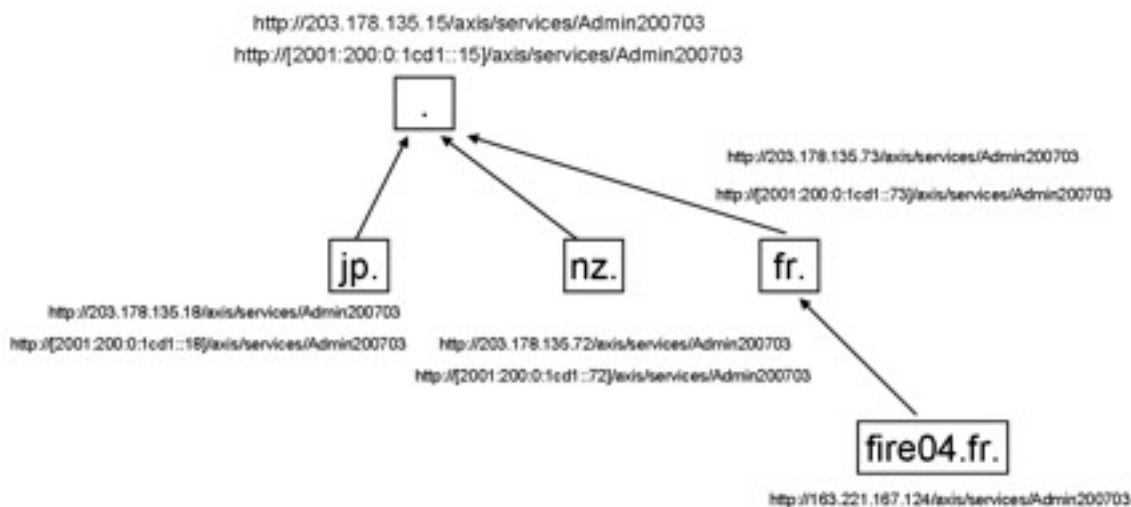


図3: 実際的なサーバツリーの例

```

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

図4: サーバ"fr."での近隣トポロジ設定

```

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

図5: サーバ"fire04.fr."での近隣トポロジ設定

設定後、Administratorを起動する。

Administratorの起動方法

```

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

Tomcatの再起動

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

2.4. サーバの動作テスト

```
$ java org.livee.test.TestAdmin200703  
XMLメッセージが出力される。 "<error . . ."が出ていないことを確認。
```

2.5. ログについて

/usr/local/livee/log/
にログが出力される。デフォルトで出力されるログファイル名は、次の通りである。

- live-e.log --- 動作のログ
- live-e-warning.log --- 警告メッセージ
- live-e-error.log --- エラーメッセージ

live-e-warning.log および live-e-error.logに何か出力されていないか確認すること。質問は、live-e[at]mri.co.jpまで。メールによるメッセージ通知機能もある。第3.3章を参照。

2.6. 自動起動の設定

以下のシンボリックリンクを作成する。

```
/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  
作成はすべてrootユーザーで行う。
```

```
$ 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
```

administrator.shの編集。

```
# vi /usr/local/livee/bin/administrator.sh  
case文の前に  
. /usr/local/livee/conf/env.vars  
の一文を挿入。
```

マシン全体を再起動後

を実行し、

```
"java org.livee.data.command.Administrator"  
が稼動していることを確認する。
```

第3章 サーバ運用

3.1. センサ管理と認証

サーバを運用する者は、センサをシステムに登録し、センサ認証を行い、センサが提供する情報に対して責任を持たなければならない。Live E! サーバシステムには、コマンドラインによるセンサ管理コマンドが用意されている。

3.1.1. センサIDのフォーマット

I. 複合センサID

複合センサとは、数種類の子センサ(e.g., 温度センサ, 湿度センサなど)で構成されているセンサのこと。フォーマットは、"センサ管理者のFQDN / センサモデル / 設置環境 / "の形式になっており、グローバルユニーク性を持つ必要がある。例えば、奈良先端科学技術大学院大学のB206 部屋に設置されたセンサ(WM918)の複合センサID は、下記のようにすることで、ユニーク性を持つことにする。

例) live-e.naist.jp/WM918/B206/

FQDN / センサモデル / 設置環境 /

注) 最後はスラッシュ(/)で終端すること

II. センサID

このID は、各単体センサに割り当てられ、"センサグループID / センサタイプ"の形に整形すること。現在規定されているタイプを表に示す。

表: センサの種類とセンサタイプの表記

センサの種類	センサタイプ表記
温度(°C)	Temperature
湿度(%)	Humidity
雨量(mm/h)	RainFall
日雨量(mm/day)	DayRainFall
風向(360方位系)	WindDir
風速(m/s)	WindSpeed
CO2濃度 (ppm)	CO2
照度(lx)	Illuminance

奈良先端科学技術大学院大学の温度センサのID は次のようになる。

例) live-e.naist.jp/WM918/B206/Temperature

グループID / センサタイプ

注) 最後はスラッシュ(/)で終端しないこと。

3.1.2. センサ登録

新しくシステムにセンサを登録する場合は、システムにログインし、次のコマンドを実

行する。

```
$ 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は、新規にセンサをシステムに登録することを意味し、-iでセンサIDを指定、-pでアクセスパスワードを指定、-oでセンサ所有者を指定、-mで所有者のメールアドレスを指定している。このセンサ管理コマンドの詳細は、Appendix Bに記載する。

3.1.3. プロファイルの登録

センサのIDをシステムに登録後、プロファイル読み込みコマンドもしくは、プロファイル登録ツールを用いて、システムにプロファイルに登録する。

コマンドか読み込ませるには、まず、プロファイルをXMLで記述し、その後、下記のProfileManagerコマンドを実行する。プロファイルのフォーマットはAppendix Cに示す。

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

プロファイルのテンプレートは

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

にある。

プロファイル操作機能は、Webサービス(ProfileManagement200703)でも提供されている。このWebサービスの詳細は、Appendix Dに示す。

このWebサービスを利用するツールとしては、SensorRegisterDXがある。SensorRegisterDXは、

<http://live-e.naist.jp/SensorRegisterDX/>
から入手可能。

3.1.4. センサデータのアップロード

データアップロードはWebサービス(DataUpload200703)を利用して行う。

DataUpload200703は、下記3つの操作を提供している(詳細はAppendix Eを参照)。

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

このアップロードインタフェースを使うセンサユニットには、気象センサWM918やWXT510に対応のアルマジロがある。アルマジロに関する問い合わせは、live-e-wg[at]wide.ad.jp。ソースコードは、

<http://live-e.hongo.wide.ad.jp/DataUpload200703/>
から入手可能。

3.2. ローカルデータの読出しと確認

\$ java org.livee.data.command.LocalDataResolver
により、ローカルサーバに蓄積されたデータをXML形式で取り出すことができる。

\$ java org.livee.data.command.CSVArchiver
では、指定したセンサ(単体センサ)を期間指定で、CSV形式のデータに抽出することができる。

3.3. ログとメッセージ通知機能

ログファイルは、\$LIVEE_HOME/logに出力される。デフォルトでは、

- live-e.log --- 動作のログ
- live-e-warning.log --- 警告メッセージ
- live-e-error.log --- エラーメッセージ

に出力されるが、これらをメールで通知することも可能である。設定ファイル(livee_config.xml)において、階層/liveeConfiguration/loggingは、図6のようになっている。

```
<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>
```

図6: ログ関係設定項目

smtpServerを指定し、標準ログ(normal)、警告メッセージ(warning)、エラーメッセージ(error)の通知先メールアドレスをmailに記述する。AdministratorおよびTomcatを再起動すれば、設定は反映される。

第4章 グローバルなセンサ検索とデータ読出し

サーバを設置し、Live E!サーバツリーと結合すると、グローバルなLive E! ネットワークからセンサ検索およびデータ抽出が可能になる。Live E! サーバは、Webサービス(GlobalDataProvider200703)により、これらの機能を提供する。

GlobalDataProvider200703は、下記の機能を提供する。

- プロファイルスキーマの取得
- センサプロファイルの取得

- ・ センサデータ(最新値)の取得
- ・ タイムゾーン指定機能
- ・ 言語ロケール指定機能
- ・ 検索対象エリア指定機能
- ・ アグリゲーション値(平均値, 最大値, 最小値)提供機能
- ・ サーバ検索機能

詳細はAppendix Fに掲載する。

Live E! サーバのコンソールからもサーバおよびセンサ等の検索ができる。

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

第5章 サービスとデータの冗長化

Live E! では、センサの管理ドメインをサイトと呼び、サイトごとにサービスおよびデータをMaster/Slave方式によって冗長化が可能である。Slaveサーバは、Masterサーバと同期し、Live E!サーバツリーやデータの読出し/検索サービスの冗長化が実現される。以下、冗長サービスの導入方法を解説する。

1. Slaveサーバ設定
2. サービスのデプロイ
3. Live E!サーバツリーとの結合

5.1. Slaveサーバ設定

5.1.1. 動作モード設定

設定ファイル livee_config.xmlにおいて、

```
<mode>master</mode>
```

を

```
<mode>slave</mode>
```

に書き換える。

5.1.2. Masterサーバ指定

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

下線部()を、MasterサーバのAdmin200703サービスに書き換える。

5.2. サービスのデプロイ

5.2.1. Slave用サービスのデプロイ

tomcatのaxisのWEB_INF/libディレクトリに、live-e-datamanager-x.x.x.jar があることを確認し、下記を実行しSlave用サービスのデプロイを行う。


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

注) もし、以前にMaster用サービスをデプロイしている場合は、

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

を実行後、

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

を実行すること。

5.2.2. Tomcat再起動

Tomcatを再起動する。

5.3. Live E! サーバツリーとの結合

親ノードに冗長化されたサーバリスト(Admin200703サービスのリスト)を登録してもらう。また、Masterサーバで管理される近隣トポロジ情報で、site type="colleague"に下記のように冗長サーバを設定。

```
<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>
```

子ノードからの構成更新通知があった場合も、対応するサイトtype="child" name="xxx"について、同様に冗長形態を明示的に指定すること。

第6章 スキーマによるプロファイル一貫性の保証

センサプロファイルに一貫性を持たせるために、Live E!システムは単一スキーマで管理され、Live E! Authorityから発行されたプロファイルスキーマの配布機構が設けられている。Live E! サーバツリーの上位側(ルートサーバ)でスキーマを規定後、発行し、それを下位側へ伝播させることにより、システム全体でのスキーマを統一させる。スキーマにより型規定(e.g., 文字列型, 整数型, 浮動小数点数型)が可能になる他、アプリケーション・オブジェクトの名前に関する規定も可能になるため、表記揺れをなくすることができる。具体的には、プロファイル登録時などにスキーマによるベリファイ・チェックを行い、表記揺れが発生した場合には、警告メッセージが発行される。

以下、プロファイルスキーマの詳細を解説する。

スキーマの構造

```

<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>

```

下記に、schemaの属性値とその意味を記す。

属性値	意味
name	プロフィールにおける属性名
class	プロフィールにおける階層 (正規表現で示される) collection -- 複合センサの集合階層 combined -- 複合センサの階層 element -- 単体センサの階層 value -- センサデータの階層
type	プロフィールにおける属性の型 boolean -- 真偽値型 integer -- 整数型 float -- 浮動小数点数型 time -- タイムスタンプ型 string -- 文字列型
value	プロフィールにおける属性値が取りうる値 (正規表現で記述)
multilanguage	プロフィールにおける属性が、多言語に対応する(trueの場合)か否か multilanguage="true" の場合、 言語依存属性名 = 属性名+"_"+言語コード でプロフィールに多言語を組み込むことが可能。利用可能な言語コードは、Appendix Gを参照のこと。 例) location_jpn は、日本語表記で、 location を記述 location_eng は、英語表記で、 location を記述 location_tha は、タイ語表記で、 location を記述
delegation	スキーマチェックを外部アプリケーション・ドメインに委譲する。 (詳細は未定)
description	属性名に対する意味表現 (詳細は未定)

第7章 新規センサの開発と導入における注意事項

新規にセンサを開発し、Live E!センサとして利用したい場合は、

1. アップロード用Webサービス(DataUpload200703)で送信可能なシステム作成
2. 新規センサに関する情報(e.g., 製造社名, 型番, 観測項目)の通知が必要となる。

前者は <http://live-e.hongo.wide.ad.jp/DataUpload200703/> から提供されているラ

イブラリツールを使って作成可能で、後者は `live-e[at]mri.co.jp`へセンサの製造会社、型番、観測項目の情報を連絡のこと。 `sensorVendor`, `sensorModel`, `sensorType`が決定され、Live E!のセンサとして正式に承認される。

Appendix A: 設定ファイル(livee_config.xml)

ここでは、Live E!サーバの設定ファイル(livee_config.xml)の例を記す。

livee_config.xmlの例

```

<liveConfiguration xmlns="http://live-e.org/Configuration/2007/03/">

<mode>master</mode>
<dbAccess>jdbc:postgresql:live</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>

```

livee_config.xml 設定内容

```

- mode -- 動作モード (master|slave)
- dbAccess -- データベースアクセスURL
+ master -- マスターシステム
+ aggregator -- データアグリゲーションシステム
- start -- デフォルト・アグリゲーション開始日時 (2007-07-01 00:00:00)
- interval -- アグリゲーションプロセス実行頻度 (300sec)
+ hourAggregator -- 1時間のアグリゲーションシステム
- thresholdTime -- 開始閾値 (70minute)
- thresholdTimeBase -- 単位(second|minute|hour|day|month)
+ dayAggregator -- 1日のアグリゲーションシステム
- thresholdTime -- 開始閾値 (1day)
- thresholdTimeBase -- 単位(second|minute|hour|day|month)
+ monthAggregator -- 1ヶ月のアグリゲーションシステム
- thresholdTime -- 開始閾値 (1month)
- thresholdTimeBase -- 単位(second|minute|hour|day|month)
+ profileSchemaLoader -- 親ノードからのスキーマ取得システム
- interval -- 取得頻度 (3600sec)
+ childProfileMerger -- 子ノードからのプロフィール読み込みシステム
- interval -- 取得頻度 (3600sec)
- cacheValidityTime -- キャッシュ時間 (7200sec)
+ dataManagerProfileMerger -- DataManagerからのプロフィール読み込みシステム
- interval -- 取得頻度 (600sec)
+ profileChecker -- プロファイルのスキーマ適合性チェックシステム
- interval -- 実行頻度 (600sec)
- errOutputLevel -- 不適合メッセージ出力レベル (null|occurrence|warning|error)
+ sensorLivingChecker -- センサデータの到達性チェックシステム
- interval -- 実行頻度 (3600sec)
- thresholdTime -- 停止判断閾値 (3day)
- thresholdTimeBase -- 単位(second|minute|hour|day|month)
- errOutputLevel -- 不適合メッセージ出力レベル (null|occurrence|warning|error)

+ slave -- スレーブシステム
- masterAdminURL -- マスターサーバのAdminサービスURL
+ copyAdmin -- Adminサービスのコピー
- interval -- コピー頻度 (600sec)
+ copyDataManager -- データマネージャのコピー
+ latestDataUpdate -- 最新データのアップデート(コピー)
- interval -- コピー頻度 (60sec)
+ profileUpdate
- start -- デフォルトコピー開始日時 (2007-07-01 00:00:00)
- interval -- コピー頻度 (300sec)
- thresholdTime -- コピー開始時刻閾値 (600sec)
- thresholdTimeBase -- 単位(second|minute|hour|day|month)
- scopeTime -- コピー対象範囲時間 (300sec)
- scopeTimeBase -- 単位(second|minute|hour|day|month)
+ archiveDataUpdate
- start -- デフォルトコピー開始日時 (2007-07-01 00:00:00)
- interval -- コピー頻度 (300sec)
- thresholdTime -- コピー開始時刻閾値 (600sec)
- thresholdTimeBase -- 単位(second|minute|hour|day|month)
- scopeTime -- コピー対象範囲時間 (300sec)
- scopeTimeBase -- 単位(second|minute|hour|day|month)

+ neighbor -- 近隣ノード情報管理システム
- topology -- 近隣情報 (neighbor xmlns="http://live-e.org/Administrator/2007/03/")

```


Appendix B: センサ管理コマンド

センサの管理を行うには、次のJavaクラスを実行する。

```
$ 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
```

コマンドには、下記の5つが定義されている。

- list (登録センサの一覧表示)
- add (新規登録)
 - 必須パラメータ ID, Password, OwnerName, OwnerMailAddress
- update (パスワードも含めた内容更新)
 - 必須パラメータ ID; オプションPassword, OwnerName, OwnerMailAddress
- close (センサ停止)
 - 必須パラメータ ID
- restore (センサ復帰)
 - 必須パラメータ ID

closeコマンドは、何らかの事情により、センサによるデータ収集を停止させる場合に使う。センサをcloseすることで、アーカイブとしては残すが、現在のセンサ網へは明示的に公開せず、動作監視リストからも除外される。

restoreコマンドは、一度停止させたセンサを再び稼働させる場合に使う。

Appendix C: 登録プロフィールフォーマット

下記のように、センサに対する登録プロフィールを記述し、プロフィール登録コマンドやWebサービスで登録すること。

```
<?xml version="1.0" encoding="UTF-8" ?>
<sensorGroup authorization="MDAwMDAwMDA=" class="combined"
address_eng="ikoma-city, nara, Japan" address_jpn="奈良県生駒市" id="live-
e.naist.jp/WM918/B206/" latitude="34.7319" longitude="135.7339"
location_eng="NARA INSTITUTE OF SCIENCE AND TECHNOLOGY" location_jpn="奈
良先端科学技術大学院大学" xmlns="http://live-e.org/DataType/2007/03/">
  <sensor id="live-e.naist.jp/WM918/B206/Temperature"
sensorType="Temperature" />
  <sensor id="live-e.naist.jp/WM918/B206/Humidity" sensorType="Humidity" />
  <sensor id="live-e.naist.jp/WM918/B206/Pressure" sensorType="Pressure" />
  <sensor id="live-e.naist.jp/WM918/B206/WindDir" sensorType="WindDir" />
  <sensor id="live-e.naist.jp/WM918/B206/WindSpeed"
sensorType="WindSpeed" />
  <sensor id="live-e.naist.jp/WM918/B206/RainFall" sensorType="RainFall" />
</sensorGroup>
```

アクセスパスワードは、BASE64エンコーディング後、authorization属性値として埋め込むこと。その他の属性名は、プロフィールスキーマにより規定されており、詳細については、第6章を参照のこと。

Appendix D: プロファイル操作サービス

プロファイルの登録/変更などの操作を行うためのWebサービス

サービス: 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();

利用可能な言語ロケールのリストを取得する。

String getProfileSchema();

プロファイルスキーマを取得する(Appendix ?参照)

String getProfileAll(String tz,String lang);

センサリストを読み出すためのサービス。タイムゾーンtz, 言語ロケールlangでサーバに登録されているプロファイルをすべて取得。

String getProfile(String id);

センサ(id)の登録プロファイルを取得する。DataProvider200703でのgetProfileと異なり, すべての言語表記を一括取得できる(Appendix C)。ユーザがプロファイルを更新する場合に, まずこのメソッドを呼び, 変更を加えてから, setProfileメソッドを呼ぶこと。

String setProfile(String profile);

Appendix Cのフォーマットで記述したセンサプロファイルを登録する。

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

センサのアクセスパスワードを更新する。old_passwd, new_passwdに, それぞれBASE64エンコーディング後の旧パスワード/新パスワードを与えること。

Appendix E: センサデータアップロードサービス

センサデータをサーバにアップロードするためのWebサービス

サービス: DataUpload200703

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

String uploadElement(String xml)

センサ単体でデータアップロードを行う。複数の値を同時にアップロードできる。パスワードは、BASE64エンコーディング後、authorizationアトリビュートとして書き込むこと。

```
<?xml version="1.0" encoding="UTF-8" ?>
<sensor id="live-e.naist.jp/WM918/Temperature" authorization="MDAwMDAwMDA=" xmlns="http://live-
e.org/DataType/2007/03/" >
  <value time="2007-06-27T00:00:00.0000000+09:00">25.6</value>
  <value time="2007-06-27T00:10:00.0000000+09:00">25.5</value>
  ...
</sensor>
```

String uploadCombined(String xml)

複合センサ単位でデータアップロードを行う。各単体センサは同時に複数の値を送信できる。パスワードは、BASE64エンコーディング後、authorizationアトリビュートとして書き込むこと。

```
<?xml version="1.0" encoding="UTF-8" ?>
<sensorGroup authorization="MDAwMDAwMDAwMDA=" 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)

複数の複合センサのデータをまとめてアップロードする。パスワードは、それぞれの複合センサに対し、BASE64エンコーディング後、authorizationアトリビュートとして書き込むこと。

```
<?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: 広域センサデータ検索／読出し

広域にわたってセンサデータを検索し、読み出すWebサービス

サービス: 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();

システムで唯一のプロファイルスキーマを提供する(第6章).

String getProfileAll(String tz,String lang);

タイムゾーンtz, 言語langですべてのセンサのプロファイルを読み出す.

```
<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の繰り返し . . .
</sensorGroup>
. . . sensorGroup class="combined"の繰り返し . . .
</sensorGroup>
```

getProfileAll("JST","Japanese");の例

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

複合センサをidで指定し, タイムゾーンtz, 言語langでプロファイルを読み出す.

```
<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>
```

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

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

地表面での緯度経度エリア[south,north]×[west,east]に含まれるセンサプロフィールをタイムゾーンtz, 言語langで読み出す。緯度表現は、北極を+90.0, 赤道が0.0, 南極を-90.0とし、経度表現は、東経180度を+180.0, グリニッジ子午線を0.0, 西経180度を-180.0で表現する。

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

指定した複合センサのプロファイル更新履歴を期間指定[start,end]でタイムゾーンtz,言語langで取得する。startおよびendはW3Cのタイムスタンプ表現を用いる。


```

<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の繰り返し . . .
  </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の繰り返し . . .
  </sensorGroup>
  . . . sensorGroup class="combined"の繰り返し . . .
</sensorGroup>

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")
  の例

```

String getLatestDataAll(String tz,String lang)

タイムゾーンtz, 言語langで, すべてのセンサの最新値を取得する.

```

<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の繰り返し . . .
</sensorGroup>
. . . sensorGroup class="combined"の繰り返し . . .
</sensorGroup>

```

getLatestDataAll("JST","Japanese");の例

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

タイムゾーンtz, 言語langで, 指定した複合センサ(id)の最新値を取得する.

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

タイムゾーンtz, 言語langで, 指定した地表面上の領域[south,north]×[west,east]の最新値を取得する. 緯度表現は, 北極を+90.0, 赤道が0.0, 南極を-90.0とし, 経度表現は, 東経180度を+180.0, グリニッジ子午線を0.0, 西経180度を-180.0で表現する.

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

指定した複合センサの生データ履歴を期間指定[start,end]でタイムゾーンtz,言語langで取得する. startおよびendはW3Cのタイムスタンプ表現を用いる.

```

<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の繰り返し . . .
</sensor>
. . . sensorの繰り返し . . .
</sensorGroup>

```

```
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")
```

の例

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

指定した複合センサの時間単位でのアグリゲート値を期間指定[start,end]でタイムゾーンtz,言語langで取得する。startおよびendはW3Cのタイムスタンプ表現を用いる。アグリゲート値とは、平均値、最大値、最小値など、一定区間を代表する値のこと。

```
<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の繰返し . . .
</sensor>
  . . . sensorの繰返し . . .
</sensorGroup>
```

```
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")
```

の例

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

指定した複合センサの日単位でのアグリゲート値を期間指定[start,end]でタイムゾーンtz,言語langで取得する。startおよびendはW3Cのタイムスタンプ表現を用いる。アグリゲート値とは、平均値、最大値、最小値など、一定区間を代表する値のこと。

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

指定した複合センサの月単位でのアグリゲート値を期間指定[start,end]でタイムゾーンtz, 言語langで取得する。startおよびendはW3Cのタイムスタンプ表現を用いる。アグリゲート値とは、平均値、最大値、最小値など、一定区間を代表する値のこと。

String search(String query)

クエリに適合するサーバの検索を行う。

クエリの方法

2段階のサーバ指定方法がある。

1. サーバ管理の名前系での指定方式
2. プロファイル要素を利用したフィルタ方式

1. 名前系での指定段階

完全一致の方法と、部分木選択の方法があり、name="hoge.jp."のようにサーバのサイト名で指定することや、name="*.jp."のような指定が可能。name="*"とするとすべてのサーバが検索対象となる。

2. プロファイルによるフィルタ方式

上記の方法で、対象とするサーバ範囲を決定後、その中で、プロファイル項目を条件にして候補サーバをフィルタする。フィルタの指定方式は、

target="combined" name="ATTRNAME" (TEST_OP="COMP_VALUE")*
に従う。target="combined"は、必須(target="element"は現在サポート外)、

TEST_OPには、lt,gt,eq,neq,lteq,gteq,containsがあり、比較対象とする値をCOMP_VALUEに記述する。

```
<query domain="admin" name="*" xmlns="http://live-e.org/Query/2007/03/">  
  <filter target="combined" name="latitude" gt="30" lt="40" />  
  <filter target="combined" name="longitude" gt="130" lt="140" />  
</query>
```

(latitude,longitude) が[30,40]×[130,140]の範囲にあるセンサを持つサーバを検索する例

```
<queryResponse xmlns="http://live-e.org/Query/2007/03/" >  
  <site name="jp." xmlns="http://live-e.org/Administrator/2007/03/">  
    <server service="Admin200703" url="http://203.178.135.14/axis/services/Admin200703">  
  </site>  
  <site name="." xmlns="http://live-e.org/Administrator/2007/03/">  
    <server service="Admin200703" url="http://203.178.135.15/axis/services/Admin200703">  
  </site>  
</queryResponse>
```

応答の例

Appendix G: 利用可能な言語ロケール

2007年08月時点での利用可能な言語ロケール
ISO639-2に準拠している。

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

利用可能な言語ロケールは徐々に追加していく予定。

Appendix H: 利用可能なタイムゾーン表記

利用可能なタイムゾーン表記.

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

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

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

Asia/Hong_Kong
Asia/Irkutsk
Asia/Kashgar
Asia/Kuala_Lumpur
Asia/Kuching
Asia/Macao
Asia/Macau
Asia/Makassar
Asia/Manila
Asia/Shanghai
Asia/Singapore
Asia/Taipei
Asia/Ujung_Pandang
Asia/Ulaanbaatar
Asia/Ulan_Bator
Asia/Urumqi
Australia/Perth
Australia/West
CTT
Etc/GMT-8
Hongkong
PRC
Singapore
Asia/Choibalsan
Asia/Dili
Asia/Jayapura
Asia/Pyongyang
Asia/Seoul
Asia/Tokyo
Asia/Yakutsk
Etc/GMT-9
JST
Japan
Pacific/Palau
ROK
ACT
Australia/Adelaide
Australia/Broken_Hill
Australia/Darwin
Australia/North
Australia/South
Australia/Yancowinna
AET
Antarctica/DumontDURville
Asia/Sakhalin
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
Pacific/Saipan
Pacific/Truk
Pacific/Yap
Australia/LHI
Australia/Lord_Howe
Asia/Magadan
Etc/GMT-11
Pacific/Efate
Pacific/Guadalcanal
Pacific/Kosrae
Pacific/Noumea
Pacific/Ponape
SST
Pacific/Norfolk
Antarctica/McMurdo
Antarctica/South_Pole
Asia/Anadyr
Asia/Kamchatka
Etc/GMT-12
Kwajalein
NST
NZ
Pacific/Auckland
Pacific/Fiji
Pacific/Funafuti
Pacific/Kwajalein
Pacific/Majuro
Pacific/Nauru
Pacific/Tarawa
Pacific/Wake
Pacific/Wallis
NZ-CHAT
Pacific/Chatham
Etc/GMT-13
Pacific/Enderbury
Pacific/Tongatapu
Etc/GMT-14
Pacific/Kiritimati