FreeSB Installation Guide

1. Introduction

Purpose

This document provides step-by-step instructions on the installation and configuration of FreeSB Enterprise Service Bus.

Quick Install

Background

FreeSB is a service-oriented application integration engine, enabling loosely coupled systems to plug-in to the bus and communicate with each other. The application uses SOAP as the transport mechanism, making use of the Apache Axis SOAP implementation.

FreeSB is comprised of the following components:

- **AuthService** – This is the authentication & authorisation service component which handles all security related matters within FreeSB
- **EsbClient** – This is the client module that applications use to "plug" into FreeSB
- **EsbCommons** - Contains all the libraries required by the FreeSB as well as code templates and build scripts
- **EsbDocs** – Contains the documentation for FreeSB
- **EsbLogger** – This handles all logging related activity within FreeSB and it's client applications
- **SupervisorService** – This is the management service for the FreeSB, it acts as both a web front-end in order configure the FreeSB, as well as a co-ordinator for FreeSB services to perform registering (or "plugging" in) and load-balancing/fail-over.
- **TestService** – Features a sample web application that showcases basic functionality such as logging in and sending log messages.

Dependencies

a. Java

FreeSB requires the Java 2 SDK, Standard Edition version 1.4 or higher to run. You can download it from http://java.sun.com and we recommend that you follow the Java 2 SDK installation instructions carefully, making sure to define the JAVA_HOME environment variable to point to the Java installation directory.

b. Ant

To build and run FreeSB, you will require Apache Ant, version 1.6 or higher, available from http://ant.apache.org as a free download. Please follow the installation instructions carefully, making sure to define the ANT_HOME environment variable to point to the Ant installation directory.

c. Database

FreeSB requires the use of a relational database in order to store the status of the system, various configuration data, log messages etc. The persistence engine used

by FreeSB is Hibernate, and as such FreeSB can only use the databases compatible with Hibernate (which are many). These are listed below:

- DB2 7.1, 7.2, 8.1
- MySQL 3.23, 4.0
- PostgreSQL 7.1.2, 7.2, 7.3, 7.4
- Oracle 8i, 9i
- Sybase 12.5 (JConnect 5.5)
- Interbase 6.0.1 (Open Source) with Firebird InterClient 2.01
- HypersonicSQL 1.61, 1.7.0
- Microsoft SQL Server 2000
- Mckoi SQL 0.93
- Progress 9
- Pointbase Embedded 4.3
- SAP DB 7.3

Instructions on configuring FreeSB to use your preferred database are provided in later sections. Our preferred database is PostgreSQL.

d. Application Server

All the major internal services (such as the authentication service, logging service etc) provided by FreeSB are servlets that are packaged into web archive (WAR) files and deployed. Hence, they require a servlet container in order to operate. FreeSB was developed using the Mortbay Jetty server (available at http://jetty.mortbay.org), which is a 100% Java implementation of the servlet specification. Most FreeSB releases are bundled with Jetty for convenience, but you may also provide your own copy of Jetty as long as you take care to define the JETTY_HOME variable indicating the installation directory of you own Jetty. Note that while other servlet containers should also work correctly, Jetty is for the moment the only server supported. The default port of Jetty is 8080.

e. LDAP server

FreeSB requires the use of an LDAP server for its authentication and authorisation functionality. You may provide your own LDAP server, but note that FreeSB requires “Manager” access i.e. the ability to add/delete/modify LDAP entries. This may cause a security concern if you’ll be using your corporate LDAP server, in which case a security scheme should be established where FreeSB can access only a particular root branch within the LDAP tree, and users duplicate to that branch.

If you do not currently have an LDAP server, or do not wish to use your corporate server, then you may download OpenLDAP from http://www.openldap.org and configure it as instructed.

Downloading FreeSB

3.1 Getting releases

FreeSB uses SourceForge.net to host the project, where you may download the latest binary release. Go to the following URL:

http://sourceforge.net/projects/esb/

3.2 Getting source code

While releases of FreeSB are regular, to access the very latest version, you must download the source code via CVS and build FreeSB yourself.

Follow the instructions on http://sourceforge.net/cvs/?group_id=95289 to set up CVS if you have not done so already.

Each CVS module, listed below, must be checked out into the same directory.

- AuthService
- EsbClient
- EsbCommons
- EsbDocs
- EsbLogger
- SupervisorService
- TestService

Building FreeSB Releases

Building FreeSB from the source code is a two-step process. First, you have to build a release - that is, take all the individual projects and create a "version" of the FreeSB. Then you must check the configuration, compile and deploy the services.

4.1 Building A Release

Building a release of FreeSB requires the use of Apache Ant, and for Jetty to already be installed and configured for your machine, with the "JETTY_HOME" environment variable set.

Once you've checked out all the CVS modules required (see Chapter 3), navigate to the "EsbCommons" directory and from there, you may create a release using the following ant targets:

```
ant release
```

or

```
ant release_jetty
```

The first command will create a FreeSB release (a directory and ZIP file) that will not be packaged with Jetty; instead it will assume the JETTY_HOME variable is the location to deploy services.

The second command will create a FreeSB release (a directory and ZIP file) that will include Jetty, and hence all deployments will use this embedded Jetty.

Note that you may pass an environment variable, "freesb.version" to specify if you want the release directory and corresponding ZIP file to be versioned, i.e. the following command will create a "FreeSB-1.0RC2" directory and corresponding "FreeSB-1.0RC2.zip" ZIP file:

```
ant -Dfreesb.version=1.0RC2  release
```

Deploying FreeSB

Once you have downloaded a release from the web, or a release has been built (see Chapter 4), it's time to configure it and deploy FreeSB's services.

5.1 Checking Configuration

Please navigate to the directory where the FreeSB release is located. If you have downloaded a release from the Internet, simply extract the ZIP file to a directory of your choice. Open the following file within that directory:

```
build.properties
```

This file contains all the properties required to configure and deploy FreeSB, make the changes that are required. Please see Appendix A for a complete reference.

Due to the nature of some of the support libraries FreeSB uses, most configuration changes require the services to be re-deployed once the changes have been performed.

5.1 Initialising Database

Once you have edited the section to do with database settings in the build.properties configuration file (making sure to create the corresponding user in your own database), in order to create the tables and initialise them with certain default values, please run the following ant target:

```
ant create_db
```

Remember to perform this just **once,** as running it again will completely clear out your existing database tables.

5.2 Creating LDAP schema

Once you have edited the section to do with LDAP settings in the build.properties configuration file, run the following ant target to initialise the LDAP schema:

```
ant create_ldap_schema
```

Again, do this only once, it is designed to enter the default users and roles required by FreeSB.

5.3 Deploying Services

The next process is to deploy the various FreeSB services to the container. Please note that as FreeSB supports distributed services, it is possible to deploy each service on different machines (with different IP addresses). It is in fact possible to deploy the same service multiple times across different machines (only one per IP address), to activate automatic load balancing – as long as all machines use the same database and LDAP server, and as long as all machines are synchronized to the same time/data (preferably through the Network Time Protocol, NTP).

Run the following command to deploy all services at once:

```
ant build_all
```

or

You can deploy individual services (to different machines) using the following commands:

```
ant SupervisorService
ant AuthService
ant EsbLogger
ant TestService
```

Please note that the “SupervisorService”, “AuthService” and “EsbLogger” must be deployed on at least one machine, in order for the FreeSB to operate. The “TestService” can be deployed now, but it is recommended to read the FreeSB Developer’s Guide for a walkthrough on that particular service, including it’s deployment.

Also note that if you have modified the source code in a release, you will need to re-create the EsbClient to reflect the changes:

```
ant EsbClient
```

5.4 Starting Jetty

Run the following command to start Jetty from all machines you have deployed FreeSB services to:

```
ant start_jetty
```

This will take a minute or two, as all the services need to “plug” themselves into the Enterprise Service Bus.

5.5 Verifying Install

Run the following command to verify that the installation succeeded (it will attempt to access a few of the FreeSB services):

```
ant validate_install
```

The message output should indicate “OK” if everything worked successfully. If not, refer to the troubleshooting section.

Appendix A: Fields in properties file

A. build.properties:

Field	Description
supervisor.service.username	This is the username used in HTTP BASIC authentication to access the supervisor service. This value is used by the container to define it's security realm.
supervisor.service.password	This is the password used in HTTP BASIC authentication to access the supervisor service. This value is used by the container to define it's security realm.
supervisor.service.url	This is the URL that will be used by all EsbClients to access the management facilities of the SupervisorService. It is reccommended to use DNS round-robin techniques in order have one domain name point to multiple machines with a SupervisorService installed on each. This prevents a single instance of the SupervisorService being saturated with management requests by clients.
hibernate.connection.username	The username of the database required by FreeSB. "Hibernate" is the database persistence component used by FreeSB.
hibernate.connection.password	The password of the database required by FreeSB. "Hibernate" is the database persistence component used by FreeSB.
hibernate.dialect	This is a setting which tells FreeSB what compatible database to use. For a list of compatible databases and their dialects, please refer to Appendix B.
hibernate.connection.driver_class	The class name of the relevant JDBC driver for your database.
hibernate.connection.url	The URL of the database you are connecting to. Depending on the database driver, this URL will be significantly different from one database to the next.
ldap.url	The URL of your LDAP server.

ldap.manager.name	The LDAP distinguished name of the “Manager” account i.e. the user who has permission to create/update/delete entries in the LDAP database.
ldap.manager.password	The password for the LDAP “Manager” account.
ldap.base	The root branch within the LDAP database where all the FreeSB entries (such as users and roles) will go. Perhaps the “Manager” account could be set to access only this branch, to maintain your organization’s security guidelines. Note that all other LDAP distinguished names in the build.properties file must refer to the same organization.
ldap.admin.username	The username of the FreeSB administrator account. This is what is used to log into the SupervisorService web front-end.
ldap.admin.password	The password of the FreeSB administrator account. This is what is used to log into the SupervisorService web front-end.
ldap.admin.distuingishedname	The LDAP distinguished name of the FreeSB administrator account. FreeSB will create this account in the LDAP database on installation.
ldap.admin.lastname	The last name of the LDAP administrator account.
supervisor.service.ldap.distuingishedname	The LDAP distinguished name of the SupervisorService. This entry is used to authenticate the SupervisorService as being a valid FreeSB service. Created by FreeSB on installation. Not recommended to change the name “Supervisor Service” in it, only the organisational unit and orgranisation to reflect your organization’s LDAP directory structure
supervisor.service.ldap.lastname	The last name of the SupervisorService LDAP account. Should be left alone as “Service” just for conventions sake.
supervisor.service.ldap.username	The username that the SupervisorService will use to register with FreeSB on startup.

supervisor.service.ldap.password	The password that the SupervisorService will use to register with FreeSB on startup.
The remaining block of LDAP configuration entries for the Auth Service, Logger Service and Test Service are all similar to the above Supervisor, except of course referencing their own details.	
esb.service.ldap.role	This is the LDAP role that all FreeSB services must belong to.
supervisorgui.access.ldap.role	This is the LDAP role that all users of the SupervisorService web front-end must belong to in order to gain access. It is assigned to the FreeSB Administrator by default.
supervisorgui.role.modifier.ldap.role	This is the LDAP role that allows modification of LDAP roles within FreeSB via the SupervisorService web front-end. It is assigned to the FreeSB Administrator by default, and hence, the FreeSB Administrator can assign it to others.
supervisorgui.user.modifier.ldap.role	This is the LDAP role that allows modification of LDAP users within FreeSB via the SupervisorService web front-end. It is assigned to the FreeSB Administrator by default.
supervisorgui.config.modifier.ldap.role	This is the LDAP role that allows modification of the various configuration parameters (such as this very list) within FreeSB via the SupervisorService web front-end. It is assigned to the FreeSB Administrator by default.
auth.service.default.principal.expirytime	This is the default expiry time of a “Principal” - a token generated when a user or service is authenticated, and is used to maintain and track that authentication within a session. It is possible to over-ride this value manually, so modification is not usually required.

auth.service.default.principal.extensiontime	This is the default extension time of a “Principal”. Every time a user or service uses FreeSB functionality that requires security (such as interacting with other services) the Principal is refreshed, and if a higher time-to-live figure is required, this value is used to extend that TTL.
auth.service.http.soap.access.path	This is the path on the web server that allows SOAP communication with the AuthService.
auth.service.http.soap.access.port	This is the port on the web server that provides access to the AuthService web application.
supervisor.service.http.soap.access.path	This is the path on the web server that allows SOAP communication with the SupervisorService.
supervisor.service.http.soap.access.port	This is the port on the web server that provides access to the SupervisorService web application.
logger.service.http.soap.access.path	This is the path on the web server that allows SOAP communication with the EsbLogger.
logger.service.http.soap.access.port	This is the port on the web server that provides access to the EsbLogger web application.

Appendix B: Compatible Database Dialects

RDBMS	Dialect
DB2	`net.sf.hibernate.dialect.DB2Dialect`
MySQL	`net.sf.hibernate.dialect.MySQLDialect`
SAP DB	`net.sf.hibernate.dialect.SAPDBDialect`
Oracle (any version)	`net.sf.hibernate.dialect.OracleDialect`
Oracle 9	`net.sf.hibernate.dialect.Oracle9Dialect`
Sybase	`net.sf.hibernate.dialect.SybaseDialect`
Sybase Anywhere	`net.sf.hibernate.dialect.SybaseAnywhereDialect`
Progress	`net.sf.hibernate.dialect.ProgressDialect`

RDBMS	Dialect
Mckoi SQL	`net.sf.hibernate.dialect.MckoiDialect`
Interbase	`net.sf.hibernate.dialect.InterbaseDialect`
Pointbase	`net.sf.hibernate.dialect.PointbaseDialect`
PostgreSQL	`net.sf.hibernate.dialect.PostgreSQLDialect`
HypersonicSQL	`net.sf.hibernate.dialect.HSQLDialect`
Microsoft SQL Server	`net.sf.hibernate.dialect.SQLServerDialect`
Ingres	`net.sf.hibernate.dialect.IngresDialect`
Informix	`net.sf.hibernate.dialect.InformixDialect`
FrontBase	`net.sf.hibernate.dialect.FrontbaseDialect`

Troubleshooting

Q: Building a service(s) does not deploy to where I want it.

A: It is possible it is deploying to an incorrect JETTY_HOME directory. Make sure that if you specify a JETTY_HOME variable in your environment, it is pointing to the correct directory. If you have both a JETTY_HOME environment variable and you use the release of FreeSB that packages Jetty, and you wish to deploy to your JETTY_HOME directory, make sure you delete the “Jetty” directory in your FreeSB release directory once it is unpacked.