|  |  |  |
| --- | --- | --- |
| **FRIB_rgb** | **Facility for Rare Isotope Beams** U.S. Department of Energy Office of Science | Michigan State University East Lansing, MI 48824-1321 • Ph: (517) 355-9672 • Fax: (517) 353-5967 www.frib.msu.edu | MSU-Wordmark-PMS-567.eps |

Configuration Module - Installation Manual

Version 0.95

|  |  |  |  |
| --- | --- | --- | --- |
|  | Name | Signature | Date |
| Prepared By | Vasu Vuppala |  |  |
| Reviewed By |  |  |  |
| Approved By |  |  |  |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Description |
| 0.9 | 16-Dec-2012 | Vuppala, V | Initial |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1 Introduction 5](#_Toc356868959)

[1.1 Purpose 5](#_Toc356868960)

[1.2 Scope 5](#_Toc356868961)

[1.3 Definitions, Acronyms, and Abbreviations 5](#_Toc356868962)

[1.4 References 6](#_Toc356868963)

[1.5 Overview 6](#_Toc356868964)

[2 System Requirements 6](#_Toc356868965)

[3 Installation 6](#_Toc356868966)

[3.1 Introduction 6](#_Toc356868967)

[3.1.1 Deployment Environment 7](#_Toc356868968)

[3.1.2 Accessing the GUI and Service 7](#_Toc356868969)

[3.2 Pre Installation 8](#_Toc356868970)

[3.3 Install 8](#_Toc356868971)

[3.4 Post Installation 9](#_Toc356868972)

[4 Compiling Configuration Module 10](#_Toc356868973)

[5 Support 11](#_Toc356868974)

[Appendix A – Glassfish Installation 12](#_Toc356868975)

[Appendix B – Apache and Glassfish 13](#_Toc356868976)

List of Figures and Tables

[Figure 1 Proteus Home Page 10](#_Toc356868977)

[Table 1 Definitions, Acronyms, and Abbreviations 5](#_Toc356868978)

# Introduction

*Open EPICS Configuration Module* (or simply *Configuration Module*) is software for managing the configuration of an accelerator facility. The configuration of an accelerator facility includes information about its components, their properties, relationships among the components, their layout, measurements, etc.

*Configuration Module* consists of the following:

* A database to store the accelerator facility’s configuration data
* A Graphical User Interface (GUI) to the data
* Services to access the data.
  + RESTful Service
  + EPICS V4 service (under development).
* Module API:
  + REST Interface
  + Java Interface
  + Python Interface

## Purpose

The objective of this document is to describe the installation of *Configuration Module*.

## Scope

It is assumed that the required infrastructure (DBMS, Application Server, Web Server) is already installed. Even though we provide basic installation instructions for them, it is assumed that you are familiar with installation of these services.

## Definitions, Acronyms, and Abbreviations

Table 1 Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| Item | Description |
| DBMS | Database Management System |
| EPICS | Experimental Physics and Industrial Control System |
| ERD | Entity-Relationship Diagram |
| FRIB | Facility for Rare Isotope Beam |
| IOC | Input/Output Controller |
| RDBMS | Relational Database Management System |
| REST | Representational State Transfer |
|  |  |
|  |  |
|  |  |
|  |  |

## References

1. Configuration Module: Schema Design Manual, Open EPICS Web Site
2. Open EPICS Web Site, <http://openepics.sourceforge.net>
3. Configuration Module: API Manual, Open EPICS Web Site
4. PVManager, <http://pvmanager.sourceforge.net>
5. EPICS CAJ, <http://epics-jca.sourceforge.net/caj/>
6. ChannelFinder, <http://channelfinder.sourceforge.net/>

## Overview

The next section lists the system requirements for Configuration Module. Installation Section describes the installation procedure.

# System Requirements

The service and GUI portions of *Configuration Module* have been developed using Java EE.It needs the following for its operation:

* Java 7
* MySQL Database Management System Version 5. *Configuration Module* has been tested with MySQL version 5.1.
* Java EE 6 Application Server
* A Web Server (preferably Apache 2)
* Open EPICS Lattice/Model Module Version 1.0

*Configuration Module* has been tested in the following environment:

* MySQL version 5.1.63 running on Debian Wheezy
* Glassfish version 3.1.2.2 running on Debian Wheezy
* Apache version 2.2 running on Debian Wheezy

However, it should work with MySQL or Glassfish running on other platforms such Windows Server 2008, Windows 7, OS X, or other distributions of Linux.

# Installation

## Introduction

Configuration Module contains the following packages:

* Database Schemas: One or more SQL DDL files containing SQL commands to create the database structures. It creates two databases (schemas) in MySQL:
  + discs\_conf: stores the configuration data. This schema name can be easily changed. The module does not depend on the name. It accesses the data through the database resource name (see the Pre Installation Section).
  + discs\_model: stores the lattice/model data. Changing this schema name requires changes to the source code. This dependency will be removed in the coming versions.
* Conf-Core: A WAR file to be installed on a Java Application Server
* Module API: Java library (JAR file) and Python module files
* Source: Source code for the service, GUI, API, documentation, and sample programs

### Deployment Environment

You need the following logical servers to install Configuration Module:

* An application server that supports Java EE 6. Configuration Module has been tested with Glassfish 3 but you may use other servers such as JBoss or WebSphere. Conf-Core package is installed on the application server. A database server. Database Schemas are installed on the database server. Configuration Module is developed and tested using MySQL 5.1.
* A Web Server: Running Glassfish on the HTTP port (80) is not advised. Hence it is kept behind a web server. Preferred web server is Apache 2 but you may any other web server that can interact with the application server. Nothing from Configuration Module is installed on the web server.

Module API is installed on developer workstations or packaged with user applications. Source package is generally installed on developer workstations only.

### Accessing the GUI and Service

Configuration Module GUI is a web application to be accessed through a browser. Based on your installation it can be accessed in one the following ways:

* You may access it directly through the application server: [http://app-server.domain:8080/conf where ‘app-server.domain](http://app-server.domain:8080/conf%20where%20'app-server.domain)’ is the name of your application server which is running on port 8080.
* A better way is to access it through a web server (direct access through the application server is disabled). In such case your URL will be something like [http://web-server.domain/conf where ‘web-server.domain](http://web-server.domain/conf%20where%20'web-server.domain)’ is the name of your web server. The subsequent versions of the GUI will authenticate users, so it is advised that HTTPS protocol is enabled on the web server, and the GUI be accessed over HTTPS instead of HTTP.

Configuration Module’s API is also accessed over the web, so the above methods hold for the service too. The base URL for the REST service will be <http://web-server.domain/conf/rs/v0>. The API is described in detail in [3].

## Pre Installation

The following steps are to be done first, and only once.

1. Download and install MySQL.
2. Create following schemas (the schemas are available on the Configuration Module’s website [2]):
   1. *Configuration Module* schema
   2. Open EPICS Lattice/Model schema
3. Download and install Glassfish. See Appendix A – Glassfish Installation for details.
4. Configure Glassfish:
   1. Login to the Glassfish administrator console (at https://glassfishserver:4848)
   2. Go to Resources => JDBC => JDBC Connection Pools
      1. Create new pool called "ConfPool" (you may choose any other name).
      2. Resource Type = javax.sql.ConnectionPoolDataSource
      3. Datasource Classname = com.mysql.jdbc.jdbc2.optional.MysqlConnectionPoolDataSource
      4. Additional Properties

ServerName = <Name of Server> ex. localhost, qa01

DatabaseName = discs\_conf

User = <Database user> eg. confapp

Password = <Database user's password>

* 1. Resources => JDBC => JDBC Resources
     1. Create a new resource called "org.openepics.conf.data". You must use this exact name for the resource.
     2. Pool Name = ConfPool (or the pool name used in Step b above).
  2. Resources => JNDI => Custom Resources
     1. Create a new resource called "org.openepics.conf.props".
     2. Choose Resource Type as java.util.Properties
     3. If you want Configuration Module to use ChannelFinder [6] for EPICS Channel information, add the following properties:
        1. ChannelFinder.Enable: set the value to ‘true’ or ‘false’. Configuration Module will use ChannelFinder only if the value is true.
        2. ChannelFinder.ServiceURL: Set the value to the service URL for the ChannelFinder.

## Install

1. Obtain Conf-Core package (the WAR file). You can download it from [2], or generate it by compiling the source code. For latter, please see the section *Compiling Configuration Module.*
2. Deploy the WAR file
   1. Login as Glassfish user
   2. Run the command: asadmin –H app-server-name deploy –contextroot=/conf conf-0.92.war
3. Configuration Module uses PVManager [4] and CAJ [5] to access the control system. Depending on your EPICS setup, you may need to configure CAJ to enable the Configuration Module to access your control system. One of the easier ways to do so is as follows:
   1. Create a directory named .JCALibrary (note the ‘.’ at beginning) in the home directory of the user running the application server (glassfish)
   2. Create a text file named JCALibrary.properties in the directory. Refer [5] regarding the content of this file. Here is a sample JCALibrary.properties file:

#

# Configuration for EPICS using CAJ

#

com.cosylab.epics.caj.CAJContext.addr\_list=controlsgw.nscl.gov:6064

com.cosylab.epics.caj.CAJContext.auto\_addr\_list=false

#com.cosylab.epics.caj.CAJContext.max\_array\_size=32768

#com.cosylab.epics.caj.impl.reactor.lf.LeaderFollowersThreadPool.thread\_pool\_size=80

#com.cosylab.epics.caj.CAJContext.beacon\_period=15

#com.cosylab.epics.caj.CAJContext.repeater\_port=5069

#com.cosylab.epics.caj.CAJContext.server\_port=5068

## Post Installation

1. Load data. Configuration Module includes a data loader (in conf-dl directory). It loads data from MS Excel files. Documentation for the data loader is still under development. Please read the schema design document [1] which will help you with understanding the database structures, and loading data. If you want to try or test Configuration Module, you may download sample data that is provided on the web site [2].
2. Test the GUI. Access the application URL through a browser. You should see ‘Proteus: Configuration’ home page (Figure 1).
3. Test the REST Service. Try the following curl commands (replace ‘localhost:8080’ on the first line with your server’s name and optionally the port number).



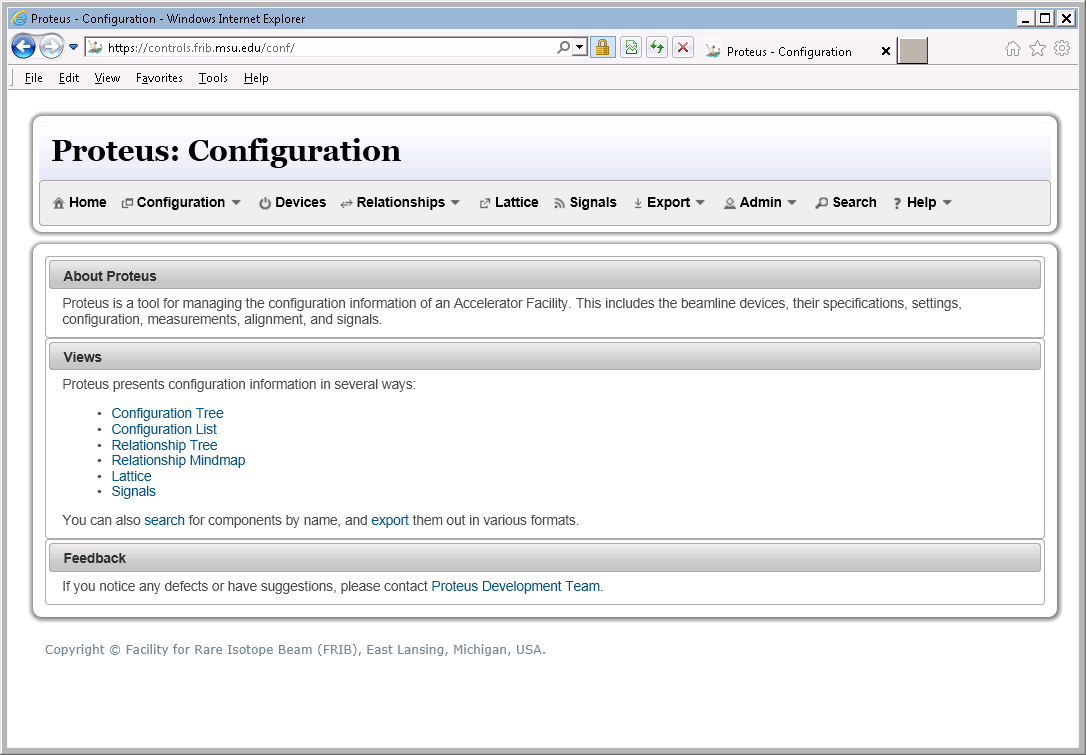


Figure 1 Proteus Home Page

# Compiling Configuration Module

* Checkout the source files from [2].
* The directory structure of the source is as follows. It contains a top-level directory named ‘conf’, and has the following sub-directories:
  + docs: source for documentation
  + design: design artifacts
  + Conf-core: GUI and REST Service
  + Conf-japi: Java API (a Netbeans project)
  + Conf-pyapi: Python API
  + Conf-samples: Sample programs that use the API
  + Conf-v4: EPICS V4 service (prototype)
  + Conf-dl: Data loader
  + Conf-ent: Configuration Module’s Entities
  + Conf-old: Lattice/Model entities
* Conf-Core package depends on conf-japi, conf-ent, and conf-olm. You may build them through Netbeans (Project -> Build) or Maven (run ‘mvn package’ in the corresponding directory).
* To generate the Conf-Core package:
  + Open it in Netbeans and start the build process (Project -> Build)
  + Or run ‘mvn package’ in conf-core directory

You must have access to the Internet to be able to build these projects because the required libraries are downloaded from the repositories automatically. You may have to run the build process a couple of times; sometimes the library files do not get downloaded during the first try. If you access the Internet through a proxy server, you will need to specify that in the maven configuration.

* The WAR file will be generated in the ‘target’ directory under conf-core. The API file (JAR file) will be generated in the ‘target’ directory under conf-japi.

# Support

Please visit <http://openepics.sourceforge.net> to report problems.

# Appendix A – Glassfish Installation

1. Download Glassfish (zip) distribution
2. Install it in /soft (/soft/glassfish3/...). You may choose some other directory.
3. Setup the startup scripts (/etc/init.d/glassfish):

update-rc.d glassdish defaults

1. Start Glassfish server. Set admin password:

asadmin change-admin-password

1. Enable secure/remote admin:

asadmin enable-secure-admin

1. Restart Glassfish server:

asadmin restart-domain

1. Add your organization’s CA certificate. This is needed only if the MySQL server is using SSL (which is recommended if the MySQL server is running on another server).

keytool -trustcacerts -import -alias "your\_ca\_Cert" -file your-cacert.pem -keystore /soft/glassfish3/glassfish/domains/domain1/config/cacerts.jks

1. Check the certificate:

keytool -list -v -keystore /soft/glassfish3/glassfish/domains/domain1/config/cacerts.jks

1. Restart Glassfish server

asadmin restart-domain

1. Configure for mod\_jk:

asadmin create-http-listener --listenerport 8009 --listeneraddress 0.0.0.0 --defaultvs server jk-connector

asadmin set configs.config.server-config.network-config.network-listeners.network-listener.jk-connector.jk-enabled=true

1. Patch installation

# Appendix B – Apache and Glassfish

1. Install Apache2 and libapache2-mod-jk (debian). Make sure mod\_jk is enabled (in /etc/apache2/mods-enabled)
2. Install glassfish 3.x
3. Create /etc/apache2/workers.prorperties with following content



1. Add the following to httpd.conf



1. Configure Glassfish

asadmin create-http-listener --listenerport 8009 --listeneraddress 0.0.0.0 --defaultvs server jk-connector

asadmin set configs.config.server-config.network-config.network-listeners.network-listener.jk-connector.jk-enabled=true

1. Create a site as follows (in /etc/apache2/sites/{available,enabled})



1. Restart Apache and Glassfish