



## **Meteor 4 Developers Guide**

Version 1.0

Contributors:  
Richard Allen  
John Lazos



# Table of Contents

Revision History .....	2
1. Source Code .....	4
2. JDK.....	4
3. Maven.....	4
4. CXF and WSS4J.....	4
5. Tomcat.....	4
6. Dependency Tree.....	5
8. Build Order.....	6
9. Eclipse with Maven.....	6

## 1. Source Code

The source code must be obtained from the National Student Clearinghouse

## 2. JDK

Java 6 is required for Meteor 4

## 3. Maven

The Meteor 4 build is implemented using Maven. Install Maven and set up a local repository.

- Download Maven 3.0.3 at <http://maven.apache.org/>

## 4. CXF and WSS4J

Meteor 4 uses secure web services, SOAP with WS-S implemented using CXF and WSS4J. Install the 1.6.4 version of WSS4J that includes the Tomcat classloading bug fix. If WSS4J 1.6.4 is not available in the Maven central repository, a snapshot version is available in CVS (wss4j-1.6.4-SNAPSHOT.jar).

```
mvn install:install-file -Dfile=wss4j-1.6.4.jar -DgroupId=org.apache.ws.security -DartifactId=wss4j -Dversion=1.6.4 -Dpackaging=jar
```

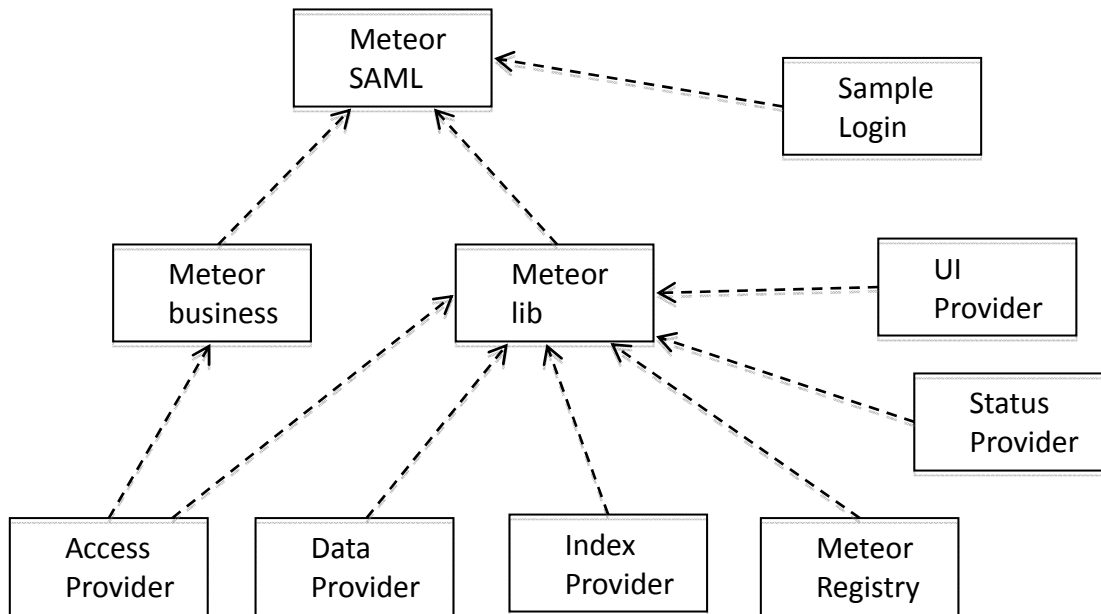
## 5. Tomcat

Tomcat 5, 6, or 7 with Java 6 is required.

- Tomcat 6 and Apache HTTP Server are required to integrate the UI Provider with Shibboleth.

## 6. Dependency Tree

Dependency Diagram



## 7. Directories

meteor → accessprovider  
meteor → dataprovider  
meteor → indexprovider  
meteor → meteorbusiness  
meteor → meteorlib  
meteor → meteorregistry  
meteor → meteorsaml  
meteor → samplelogin  
meteor → uiprovider

## 8. Build Order

The following Maven commands can be used to build Meteor components.

- a. Access Provider, build AP, example UI, and sample login WAR files.
  - `cd ./meteor/meteorsaml`
  - `mvn clean package install`
  - `cd ./meteor/meteorlib`
  - `mvn clean package install`
  - `cd ./meteor/meteorbusiness`
  - `mvn clean package install`
  - `cd ./meteor/accessprovider`
  - `mvn clean package`
  - `cd ./meteor/uiprovider`
  - `mvn clean package`
  - `cd ./meteor/samplelogin`
  - `mvn clean package`
  
- b. Data Provider, build DP WAR file.
  - `cd ./meteor/meteorsaml`
  - `mvn clean package install`
  - `cd ./meteor/meteorlib`
  - `mvn clean package install`
  - `cd ./meteor/dataprovider`
  - `mvn clean package`
  
- c. Index Provider, build IP and registry wrapper WAR files.
  - `cd ./meteor/meteorsaml`
  - `mvn clean package install`
  - `cd ./meteor/meteorlib`
  - `mvn clean package install`
  - `cd ./meteor/meteorregistry`
  - `mvn clean package`
  - `cd ./meteor/indexprovider`
  - `mvn clean package`

## 9. Eclipse with Maven

The following Maven command can be used to maintain Eclipse project configuration and JAR dependencies.

- `mvn eclipse:clean eclipse:eclipse`

## 10. Meteor Components

**accessprovider** – WAR and SOAP endpoint for XML aggregated data results with business rules applied.

**dataprotider** – WAR and SOAP endpoint for XML data results from one source.

**indexprovider** – WAR and SOAP endpoint for XML list of loan locator links and DP ids that have data for an SSN.

**meteorregistry** – WAR and SOAP endpoint for XML with valid public keys and URLs to contact for data.

**uiprovider** – WAR and XSLT for converting XML aggregated data results into the default look and feel HTML. Shibboleth headers are processed here when a Shibboleth session is present.

**samplelogin** – WAR and sample code for authentication providers wishing to generate SAML assertions representing trusted users on the network.

**meteorsaml** – JAR used to generate SAML assertions representing trusted users on the network.

**meteorlib** – JAR with base classes used by other Meteor 4 components.

**meteorbusiness**- JAR with best source, loan de-duplication, and other business logic.

**statusprovider** – WAR file used by the Clearinghouse to ping data providers and check their status.