How to Set Up Meteor

1. Download the Meteor software from the User Documentation section of www.MeteorNetwork.org.
2. Install Java SDK (see Appendix A).
   - Add [Java root directory]\bin to your PATH environment variable.
   - Set JAVA_HOME environment variable
3. Install Tomcat (see Appendix B).
4. Deploy Meteor WAR(s) (see Appendix C).
   - Modify .properties files, as necessary (see Appendix D).
5. Generate Private and Public Keys (see Appendix E)
   - If you're testing, create a self-signed certificate. For production you will need a valid signed certificate from a certificate authority, such as Verisign or Thawte.
   - Add necessary SSL directives to Tomcat configuration files.
   - Add necessary private key information to Meteor .properties files.
   - Send public key to Meteorhelpdesk@studentclearinghouse.org so it can be added to the Meteor registry.
6. Install and set up the database (for data provider)
   - Create database and tables from sql source.

Appendix A: Installing Java SDK

The Java Development Kit (SDK) is available at http://www.oracle.com/technetwork/java/javase/downloads/index.html. The Meteor Technical Team recommends that you use the latest version of SDK that is compatible with your application server. Up-to-date installation instructions for your environment are available at http://java.sun.com.

IMPORTANT: Downloading the Java Runtime Environment (JRE) is not sufficient to run Tomcat (Tomcat uses the javac executable that is not present in JRE).

Java 6 is recommended for running Meteor 4.

Appendix B: Installing Tomcat

The latest version of Apache Tomcat is available at http://tomcat.apache.org/. The Meteor Technical Team recommends following the latest installation information at the Tomcat web site for your specific environment. Tomcat 7 is recommended for running Meteor 4. Tomcat 6 is recommended for running Meteor 4 as a Shibboleth Service Provider.

Appendix C: Deploying WAR Files (Tomcat installation)

Deploying the Meteor WAR (Web ARchive) files is fairly straightforward. Simply place a copy of the WAR file(s) into the Tomcat webapps directory and start Tomcat. Tomcat automatically decompresses the file and places its contents into the necessary location.

Appendix D: Modifying .properties Files

Your instance of Meteor must be configured before connecting to the Meteor Registry. This is done by modifying the .properties files within the provider directories in the Tomcat webapps folder (%TOMCAT_HOME\webapps*provider\WEB-INF\classes). The .properties files are plain text and can be modified using any text editor (e.g., Notepad or TextPad for a Windows environment; Vi or Emacs for a Linux environment).

Below are the modifications necessary for each Provider instance, including changes for production and stand-alone environments. If a file does not exist within the specified directory, it must be created. Please refer to the Meteor 4.0 Implementation Guide for more detailed information about the individual attributes in each of the property files discussed below.

Required Files

The files required for each provider are as follows:

UI Provider (found in %TOMCAT_HOME\webapps\uiprovider\WEB-INF\classes)
- uiprovider.properties
- authentication.properties
- views.properties

Access Provider (found in %TOMCAT_HOME%\webapps\accessprovider\WEB-INF\classes)
- accessprovider.properties
- authentication.properties
- directory.properties*
- wss4j.properties

Data Provider (found in %TOMCAT_HOME%\webapps\dataprovider\WEB-INF\classes)
- dataprovider.properties
- authentication.properties
- directory.properties*

Index Provider (found in %TOMCAT_HOME\webapps\indexprovider\WEB-INF\classes)
- indexprovider.properties
- authentication.properties
- directory.properties*
- indexresponse.properties

Meteor Registry (found in %TOMCAT_HOME\webapps\meteorregistry\WEB-INF\classes)
- directory.properties (not same as AP, DP, and IP versions)
- directorydata.properties**

Sample Login **(stand-alone only; do NOT use in production)**
(found in %TOMCAT_HOME\webapps\samplelogin\WEB-INF\classes)
- samplelogin.properties

** identical files: directory.properties used in Access and Data Providers*

*** directorydata.properties:* ***NEVER use this file in production****. Used in Access, Data and Index Providers (not needed if connecting to an LDAP server)*

Configuring UI Provider – Default look-and-feel as XSLT transforms to the raw XML data from AP.

The UI Provider is coupled with an Access Provider. This coupling is enforced using digital certificates. All UI provider requests to an Access Provider are digitally signed. The Access Provider will only accept requests that are signed by a certificate in its configured trust store. Please refer to the properties file for details.

The uiprovider.tokenproviderclass property is used to specify the interface to whatever authentication implementation the UI provider will use (e.g., Shibboleth). Meteor 4 comes with a sample token provider (works with the sample login WAR) and a Shibboleth token provider. You must implement the TokenProvider interface of the UI WAR to use other authentication implementations. See the Token Provider implementation guide for further instructions.

These files can be found in
%TOMCAT_HOME\webapps\uiprovider\WEB-INF\classes.

uiprovider.properties

```
# URL to Access Provider web service endpoint
# The Access Provider must trust the certificate you provide in the UI provider's
authentication.properties
accessprovider.url=http://localhost:8080/accessprovider/services/AccessProviderService

# Connection and read timeouts between the UI provider and the Access Provider.
#   accessprovider.connection.timeout - time in ms to wait for a connection to be made
#       Default is 30000
#   accessprovider.receive.timeout - time in ms to wait for the Access Provider to return query
results
#       Default is 240000 (four minutes)
accessprovider.connection.timeout=30000
accessprovider.receive.timeout=240000

# TokenProvider implementation
#   Creates a Meteor SecurityToken from whatever login request/session info
#   is provided by your authentication technology (e.g., Shibboleth, OpenAM)
# org.meteornetwork.meteor.provider.ui.SampleTokenProvider
#      Sample provider for testing purposes. DO NOT USE IN PRODUCTION
uiprovider.tokenproviderclass=org.meteornetwork.meteor.provider.ui.token.SampleTokenProvider
#uiprovider.tokenproviderclass=org.meteornetwork.meteor.provider.ui.token.ShibbolethTokenProvider

# This is the contact url provided when the users encounter HTTP page not found,
# access denied, and internal server errors
uiprovider.errorcontacturl=http://www.contactus123456.com

# Determines whether or not to show the query status splash screen.
# Any value other than 'Yes' disables the splash.
#
uiprovider.splashenabled=Yes
```

Configuring UI Provider (continued)

authentication.properties

```
# This specifies the private key the UI provider will sign requests for Meteor data with.
# Only access providers that trust this UI Provider's certificate will process requests.

# Supported keystore types: JKS, PKCS12
org.apache.ws.security.crypto.merlin.keystore.type=PKCS12

# This is the path to the actual keystore that
# holds the private key.  Be careful where you
# put this file.  Under no circumstances should
# you put the keystore in a location that is accessible
# to the web!!!!!
org.apache.ws.security.crypto.merlin.keystore.file=C:/Projects/Meteor/certs/ap.pfx
# Password needed to read the keystore
org.apache.ws.security.crypto.merlin.keystore.password=meteor

# Keystore alias to point to a specific private key
# in the keystore
org.apache.ws.security.crypto.merlin.keystore.alias=ap

# Keystore password for the private key
org.apache.ws.security.crypto.merlin.keystore.private.password=meteor

# ######################################################
#
# Internal WSS4J properties - should not be changed
#
org.apache.ws.security.crypto.provider=org.apache.ws.security.components.crypto.Merlin
#
# ######################################################
```

Configuring UI Provider (continued)

views.properties

```
rawxml.(class)=org.springframework.web.servlet.view.xslt.XsltView
rawxml.sourceKey=meteorData
rawxml.url=/WEB-INF/classes/xsl/pages/rawxml.xsl

awardSummary.(class)=org.springframework.web.servlet.view.xslt.XsltView
awardSummary.sourceKey=meteorData
awardSummary.url=/WEB-INF/classes/xsl/pages/AwardSummary.xsl

repaymentSummary.(class)=org.springframework.web.servlet.view.xslt.XsltView
repaymentSummary.sourceKey=meteorData
repaymentSummary.url=/WEB-INF/classes/xsl/pages/Repayment.xsl

aversion.(class)=org.springframework.web.servlet.view.xslt.XsltView
aversion.sourceKey=meteorData
aversion.url=/WEB-INF/classes/xsl/pages/AwardDefault.xsl

awardDetail.(class)=org.springframework.web.servlet.view.xslt.XsltView
awardDetail.sourceKey=meteorData
awardDetail.url=/WEB-INF/classes/xsl/pages/AwardDetails.xsl

disbursement.(class)=org.springframework.web.servlet.view.xslt.XsltView
disbursement.sourceKey=meteorData
disbursement.url=/WEB-INF/classes/xsl/pages/Disbursement.xsl

references.(class)=org.springframework.web.servlet.view.xslt.XsltView
references.sourceKey=meteorData
references.url=/WEB-INF/classes/xsl/pages/References.xsl

repaymentDetail.(class)=org.springframework.web.servlet.view.xslt.XsltView
repaymentDetail.sourceKey=meteorData
repaymentDetail.url=/WEB-INF/classes/xsl/pages/RepaymentDetail.xsl

consolidated.(class)=org.springframework.web.servlet.view.xslt.XsltView
consolidated.sourceKey=meteorData
consolidated.url=/WEB-INF/classes/xsl/pages/Consolidated.xsl

querySsn.(class)=org.springframework.web.servlet.view.xslt.XsltView
querySsn.sourceKey=sourceXml
querySsn.url=/WEB-INF/classes/xsl/pages/Query.xsl

queryStatus.(class)=org.springframework.web.servlet.view.xslt.XsltView
queryStatus.sourceKey=sourceXml
queryStatus.url=/WEB-INF/classes/xsl/pages/QueryStatus.xsl

error.(class)=org.springframework.web.servlet.view.xslt.XsltView
error.sourceKey=sourceXml
error.url=/WEB-INF/classes/xsl/pages/Error.xsl

denied.(class)=org.springframework.web.servlet.view.xslt.XsltView
denied.sourceKey=sourceXml
denied.url=/WEB-INF/classes/xsl/pages/Denied.xsl

sessionExpired.(class)=org.springframework.web.servlet.view.xslt.XsltView
sessionExpired.sourceKey=sourceXml
sessionExpired.url=/WEB-INF/classes/xsl/pages/SessionExpired.xsl
```

Configuring AccessProvider – This component calls the data providers, aggregates the data, and applies business rules.

accessprovider.properties files can be found in %TOMCAT_HOME\webapps\accessprovider\WEB-INF\classes.

accessprovider.properties

```
# How long do you want to wait for all of the
# Data Providers to respond? This value is in
# milliseconds. The default value is 20 seconds
# (which is 20000 milliseconds)
dataprovider.timeout=20000
```

authentication.properties

```
# This key is *your* identifier into the Meteor
# registry. If you have any questions on what
# to put in this value, please contact the
# maintainers of the Meteor Registry
authentication.identifier=[Your Meteor Org ID]

# What provider type are you? Valid values are:
#   AccessProvider
authentication.providertype=AccessProvider

authentication.process.identifier=1

# Do you want to sign your assertions? If not then make this value "No"
# Realize that in a production environment everyone will probably require a signed
# assertion.  So if you don't sign yours then you probably won't get any data
# Only change this to "No" when you know that the entity you are communicating
# with does not require the assertion to be signed
#org.apache.ws.security.saml.issuer.signAssertion=false
org.apache.ws.security.saml.issuer.signAssertion=true

# Supported keystore types: JKS, PKCS12
org.apache.ws.security.crypto.merlin.keystore.type=PKCS12

# This is the path to the actual keystore that
# holds the private key.  Be careful where you
# put this file.  Under no circumstances should
# you put the keystore in a location that is accessible
# to the web!!!!!
org.apache.ws.security.crypto.merlin.keystore.file=C:/Projects/Meteor/certs/ap.pfx
# Password needed to read the keystore
org.apache.ws.security.crypto.merlin.keystore.password=meteor

# Keystore alias to point to a specific private key in the keystore
org.apache.ws.security.saml.issuer.key.name=ap

# Keystore password for the private key
org.apache.ws.security.saml.issuer.key.password=meteor


# ###################################################
#
# Internal WSS4J properties - should not be changed
#
org.apache.ws.security.crypto.provider=org.apache.ws.security.components.crypto.Merlin
org.apache.ws.security.saml.issuer=meteornetwork.org
org.apache.ws.security.saml.issuer.cryptoProp.file=authentication.properties
org.apache.ws.security.saml.issuer.sendKeyValue=false
#
# ###################################################
```

Configuring AccessProvider (continued)

directory.properties

```
# Which Java class should be loaded to handle the Directory requests
#
directory.class=org.meteornetwork.meteor.common.registry.RegistryWebServiceClient
#directory.class=org.meteornetwork.meteor.registry.PropertyRegistryManager

#####################################################################
## For RegistryWebServiceClient
##  directory.ws.url - primary registry webservice url
##  directory.ws.failover.url - failover registry webservice url
##
##  These URLs must be different from each other
##
##  Note: do not remove these properties without also commenting
##  out the jaxws:client definitions in cxf-servlet.xml
#####################################################################
directory.ws.url=http://localhost:8080/meteorregistry/services/RegistryService
directory.ws.failover.url=http://localhost:8080/meteorregistry/services/RegistryService-
failover
```

wss4j.properties

```
# Access Provider certificate trust store. Only requests
# signed with certificates in this trust store will be accepted.
# Your UI provider authentication.properties will need to be
# configured with a certificate that is trusted by this specified
# keystore.
#
org.apache.ws.security.crypto.merlin.truststore.file=C:/Projects/Meteor/certs/ap.pfx

# Password to access trust store
org.apache.ws.security.crypto.merlin.truststore.password=meteor

# Type of trust store. Supported types: PKCS12, JKS
org.apache.ws.security.crypto.merlin.truststore.type=PKCS12

# ######################################################
#
# Internal WSS4J properties - should not be changed
#
org.apache.ws.security.crypto.provider=org.apache.ws.security.components.crypto.Merlin
org.apache.ws.security.saml.issuer=meteornetwork.org
#
# ######################################################
```

Configuring DataProvider – Access back-end data and format it as a Meteor response. dataprovider.properties files can be found in %TOMCAT_HOME\webapps\DataProvider\WEB-INF\classes. Sample settings are included for XML files.

dataprovider.properties

```
# class name of your DataServerAbstraction implementation.
# This class will be loaded into the Spring context
default.data.server=org.meteornetwork.meteor.provider.data.FileDataServer

# Configuration for file data server
filedataserver.directory=C:/Projects/Meteor/workspace/meteor40/testxml
filedataserver.format=TestCase%{9}

# What is the minimum authentication level that you will allow
# requests from?  This MUST be an integer!!!
accessprovider.minimum.authentication.level=2

# Data Provider Data - Used to identify your responses to the access provider

# DataProvider.Data.usepropertydata - if Y, data provider software will add
#  the following data to all meteor response messages. Otherwise, it is expected
#  that the data server abstraction implementation will add data provider data
#  to all responses.
DataProvider.Data.usepropertydata=Y
DataProvider.Data.Name=My Data Provider
DataProvider.Data.ID=12345
DataProvider.Data.URL=http://www.yourmeteorhome.com
# Type:
#   S       - School
#   LO      - Loan Originator
#   LRS     - Loan Repayment Servicer
#   G       - Guaranty Agency
#   DA      - Disbursing Agent
#   GSP     - Grant & Scholarship Provider
#   FAT     - Financial Aid History Transcript
DataProvider.Data.Type=LRS
DataProvider.Data.Contacts.PhoneNum=(555) 555-5555
DataProvider.Data.Contacts.Email=meteor@yourmeteorhome.com
DataProvider.Data.Contacts.Addr=555 Meteor Parkway
DataProvider.Data.Contacts.Addr2=Suite 555
DataProvider.Data.Contacts.City=Meteor
# State can only be two characters
DataProvider.Data.Contacts.StateProv=ME
DataProvider.Data.Contacts.PostalCd=55555
```

Configuring DataProvider (continued)

authentication.properties

```
# This key is *your* identifier into the Meteor
# registry. If you have any questions on what
# to put in this value, please contact the
# maintainers of the Meteor Registry
#
authentication.identifier=[Your Meteor Org ID]

# What provider type are you? Valid values are:
#   DataProvider
authentication.providertype=DataProvider

# Do you want to sign your assertions? If not then make this value "No"
# Realize that in a production environment everyone will probably require a signed
# assertion.  So if you don't sign yours then you probably won't get any data
# Only change this to "No" when you know  that the entity you are communicating
# with does not require the assertion to be signed
#org.apache.ws.security.saml.issuer.signAssertion=false
org.apache.ws.security.saml.issuer.signAssertion=true

# Supported keystore types: JKS, PKCS12
org.apache.ws.security.crypto.merlin.keystore.type=PKCS12

# This is the path to the actual keystore that
# holds the private key.  Be careful where you
# put this file.  Under no circumstances should
# you put the keystore in a location that is accessible
# to the web!!!!!
org.apache.ws.security.crypto.merlin.keystore.file=C:/Projects/Meteor/certs/ap.pfx
# Password needed to read the keystore
org.apache.ws.security.crypto.merlin.keystore.password=meteor

# Keystore alias to point to a specific private key
# in the keystore
org.apache.ws.security.saml.issuer.key.name=ap

# Keystore password for the private key
org.apache.ws.security.saml.issuer.key.password=meteor

# ######################################################
#
# Internal WSS4J properties - should not be changed
#
org.apache.ws.security.crypto.provider=org.apache.ws.security.components.crypto.Merlin
org.apache.ws.security.saml.issuer=meteornetwork.org
org.apache.ws.security.saml.issuer.cryptoProp.file=authentication.properties
org.apache.ws.security.saml.issuer.sendKeyValue=false
#
# ######################################################
```

Configuring DataProvider (continued)

directory.properties

```
# Which Java class should be loaded to handle the Directory requests
#
directory.class=org.meteornetwork.meteor.common.registry.RegistryWebServiceClient
#directory.class=org.meteornetwork.meteor.registry.PropertyRegistryManager

#####################################################################
## For RegistryWebServiceClient
##  directory.ws.url - primary registry webservice url
##  directory.ws.failover.url - failover registry webservice url
##
##  These URLs must be different from each other
##
##  Note: do not remove these properties without also commenting
##  out the jaxws:client definitions in cxf-servlet.xml
#####################################################################
directory.ws.url=http://localhost:8080/meteorregistry/services/RegistryService
directory.ws.failover.url=http://localhost:8080/meteorregistry/services/RegistryService-
failover
```

Configuring IndexProvider – Returns a list of data providers with data for a Social Security number. This configuration is for a stand-alone environment only. These files can be found in %TOMCAT_HOME\webapps\indexprovider\WEB-INF\classes.

indexprovider.properties

```
# Class name of your IndexServerAbstraction implementation.
# This class will be loaded into the Spring context
default.index.server=org.meteornetwork.meteor.provider.index.PropertiesIndexServer

# only required for PropertiesIndexServer. Delete for others.
# This properties file must be located in your classpath.
# Leave the ".properties" off the file name.
propertiesindexserver.propertyfile=indexresponsesample

# data returned in the IndexResponseMessage, including loan locator info
IndexProvider.ID=IP
IndexProvider.Name=National Student Clearinghouse
IndexProvider.URL=http://loanlocator.org
```

Configuring IndexProvider (continued)

authentication.properties

```
# This key is *your* identifier into the Meteor
# registry. If you have any questions on what
# to put in this value, please contact the
# maintainers of the Meteor Registry
#
authentication.identifier=[Your Meteor Org ID]

# What provider type are you? Valid values are:
#   DataProvider
authentication.providertype=DataProvider

# Do you want to sign your assertions? If not then make this value "No"
# Realize that in a production environment everyone will probably require a signed
# assertion.  So if you don't sign yours then you probably won't get any data
# Only change this to "No" when you know  that the entity you are communicating
# with does not require the assertion to be signed
#org.apache.ws.security.saml.issuer.signAssertion=false
org.apache.ws.security.saml.issuer.signAssertion=true

# Supported keystore types: JKS, PKCS12
org.apache.ws.security.crypto.merlin.keystore.type=PKCS12

# This is the path to the actual keystore that
# holds the private key.  Be careful where you
# put this file.  Under no circumstances should
# you put the keystore in a location that is accessible
# to the web!!!!!
org.apache.ws.security.crypto.merlin.keystore.file=C:/Projects/Meteor/certs/ap.pfx
# Password needed to read the keystore
org.apache.ws.security.crypto.merlin.keystore.password=meteor

# Keystore alias to point to a specific private key
# in the keystore
org.apache.ws.security.saml.issuer.key.name=ap

# Keystore password for the private key
org.apache.ws.security.saml.issuer.key.password=meteor

# ######################################################
#
# Internal WSS4J properties - should not be changed
#
org.apache.ws.security.crypto.provider=org.apache.ws.security.components.crypto.Merlin
org.apache.ws.security.saml.issuer=meteornetwork.org
org.apache.ws.security.saml.issuer.cryptoProp.file=authentication.properties
org.apache.ws.security.saml.issuer.sendKeyValue=false
#
# ######################################################
```

Configuring IndexProvider (continued)

directory.properties

```
# Which Java class should be loaded to handle the Directory requests
#
directory.class=org.meteornetwork.meteor.common.registry.RegistryWebServiceClient
#directory.class=org.meteornetwork.meteor.registry.PropertyRegistryManager

#####################################################################
## For RegistryWebServiceClient
##  directory.ws.url - primary registry webservice url
##  directory.ws.failover.url - failover registry webservice url
##
##  These URLs must be different from each other
##
##  Note: do not remove these properties without also commenting
##  out the jaxws:client definitions in cxf-servlet.xml
#####################################################################
directory.ws.url=http://localhost:8080/meteorregistry/services/RegistryService
directory.ws.failover.url=http://localhost:8080/meteorregistry/services/RegistryService-
failover
```

Configuring Meteor Registry – Web service wrapper for the Meteor Registry look-ups. This configuration is for a stand-alone environment only. These files can be found in %TOMCAT_HOME\webapps\meteorregistry\WEB-INF\classes.

directory.properties

```
# Which Java class should be loaded to handle the Directory requests
#
#directory.class=org.meteornetwork.meteor.registry.LDAPRegistryManager
directory.class=org.meteornetwork.meteor.registry.PropertyRegistryManager

#####################################################################
## For PropertyRegistryManager
##
##  directory.properties.directorydata - the name of the .properties file
##          that contains the directory data. Do not include the ".properties"
##          extension
#####################################################################
directory.properties.directorydata=directorydata

#####################################################################
##   Production LDAP Server - anonymous access using ssl
##          Note: do not delete these properties without also removing
##          the applicationContext-ldap.xml file from the contextConfigLocation
##          in web.xml
#####################################################################
directory.ldap.providerurl=ldap://65.202.239.207:389
directory.ldap.failover.providerurl=ldap://65.202.239.207:389
#directory.ldap.providerurl=ldaps://ldap.meteorregistry.com:636
directory.ldap.basedn=o=meteorregistry.com
directory.ldap.authentication.type=simple
directory.ldap.authentication.principal=cn=matadmin,o=meteorregistry.com
directory.ldap.authentication.credentials=meteortest
```

Configuring Meteor Registry (continued)

directorydata.properties (do NOT use in production)

```
#[id].Name=[id]
#[id].Type=[provider type]
#[id].Version=4.0.0.0
#[id].AuthenticationProcessID=1
#[id].APCSR.AuthenticationLevel=[minimum authentication level this provider will accept for
this role]
#[id].BORROWER.AuthenticationLevel=[minimum authentication level this provider will accept
for this role]
#[id].FAA.AuthenticationLevel=[minimum authentication level this provider will accept for
this role]
#[id].LENDER.AuthenticationLevel=[minimum authentication level this provider will accept for
this role]
#[id].AccessProvider.Certificate=[absolute path to certifiate file (C:\...)]
#[id].AccessProvider.URL=[meteor connection url]
#[id].AccessProvider.Status=AC
#[id].AccessProvider.Aliases=[comma separated list of aliases] e.g. 12345,23456,34567

LTI_IP40.Name=LTI_IP40
LTI_IP40.Type=IndexProvider
LTI_IP40.Version.1=4.0.0.0
LTI_IP40.AuthenticationProcessID=1
LTI_IP40.APCSR.AuthenticationLevel=3
LTI_IP40.BORROWER.AuthenticationLevel=3
LTI_IP40.FAA.AuthenticationLevel=3
LTI_IP40.LENDER.AuthenticationLevel=3
LTI_IP40.IndexProvider.Certificate=C:/Projects/Meteor/certs/ap.cer
LTI_IP40.IndexProvider.URL=http://localhost:8080/indexprovider/services/IndexProviderService
LTI_IP40.IndexProvider.Status=AC

LTI_DP40.Name=LTI_DP40
LTI_DP40.Type=DataProvider
LTI_DP40.Version=4.0.0.0
LTI_DP40.AuthenticationProcessID=1
LTI_DP40.APCSR.AuthenticationLevel=3
LTI_DP40.BORROWER.AuthenticationLevel=3
LTI_DP40.FAA.AuthenticationLevel=3
LTI_DP40.LENDER.AuthenticationLevel=3
LTI_DP40.DataProvider.Certificate=C:/Projects/Meteor/certs/ap.cer
LTI_DP40.DataProvider.URL=http://localhost:8080/dataprovider/services/DataProviderService
LTI_DP40.DataProvider.Status=AC

LTI_AP40.Name=LTI_AP40
LTI_AP40.Type=AccessProvider
LTI_AP40.Version=4.0.0.0
LTI_AP40.AuthenticationProcessID=1
LTI_AP40.APCSR.AuthenticationLevel=3
LTI_AP40.BORROWER.AuthenticationLevel=3
LTI_AP40.FAA.AuthenticationLevel=3
LTI_AP40.LENDER.AuthenticationLevel=3
LTI_AP40.AccessProvider.Certificate=C:/Projects/Meteor/certs/ap.cer
LTI_AP40.AccessProvider.URL=http://localhost:8080/accessprovider/services/AccessProviderServi
ce
LTI_AP40.AccessProvider.Status=AC
LTI_AP40.AccessProvider.Aliases=12345,23456
```

Configuring Sample Login (do NOT use in production) – Sample authentication provider. This configuration is for a stand-alone environment only. These files can be found in %TOMCAT_HOME\webapps\samplelogin\WEB-INF\classes.

directory.properties

```
# UI provider to login to after authentication is successful.
samplelogin.uiprovider.url=http://localhost:8080/uiprovider/meteor/query.do

# The URL UI providers will call to get the security token
samplelogin.artifactresolver.url=http://localhost:8080/samplelogin/resolveArtifact

# Identifier for assertion subject name
samplelogin.authentication.identifier=samplelogin

# Authentication Process ID
samplelogin.authentication.process.identifier=1
```

Appendix E: Public & Private Keys and Certificates

Key generation is handled by Java keytool.

Step 1: Generate a keystore and key entry
From the command line, locate your Java /bin directory (can be found at JAVA_HOME/bin) and enter:

```
keytool –genkey –keyalg RSA –alias meteor –keystore [keystore name]
```

You will be asked to enter a keystore password. Choose a password and enter it when prompted to do so.

You will then be asked a series of questions regarding your organization. Enter your response for each question (questions are in bold; sample answers are in blue italics).

What is your first and last name? [Unknown] *www.mydomain.com* (enter your organization's domain name)

What is the name of your organizational unit? [Unknown] *Web Development*

What is the name of your Organization? [Unknown] *Acme Inc*

What is the name of your City or Locality? [Unknown] *New York City*

What is the name of your State or Province? [Unknown] *New York*

What is the two-letter country code for this unit? [Unknown] *US*

Is CN=www.mydomain.com, **OU**=Web Development, **O**=Acme Inc, **L**=New York City, **ST**=New York, **C**=US **correct? [no]** *yes*

Enter key password for <meteor>: *Return (if same as keystore password)*

You should specify the same password for the keystore and the key entry. Otherwise, you'll receive the following error message when you restart the Jakarta engine:

```
java.security.UnrecoverableKeyException: Cannot recover key
```

Finally, verify that your keystore was created. Run:

```
keytool –list –keystore [keystore name]
```

They keystore will be stored in your Java SDK/bin directory. In a Windows environment, it will be saved to your Documents and Settings folder (e.g., c:\Documents and Settings\zorro\.keystore). We recommended that you make and retain a backup copy of your keystore file in the event of a server crash.

If you are setting up a stand-alone Meteor system, you can self-sign your certificate (and skip steps 2-4):

```
keytool –selfcert –keystore [keystore name] –alias meteor
```

Self-signed certificates can only be used in a stand-alone or testing environment, but are not required. You must obtain a certificate from Thawte or Verisign to participate in Meteor. Steps 2-4 will guide you through the process of creating a certificate signing request (CSR) and importing a signed certificate.

Step 2: Generate a CSR of the newly created keystore and keyentry
Generate the certificate signing request as follows:

```
keytool –certreq –alias meteor –keyalg RSA –file certreq.csr –keystore [keystore name]
```

Enter the keystore password when prompted. The CSR file will be saved to your Java SDK/bin directory. In a Windows environment, it will be saved to your Documents and Settings folder (e.g., c:\Documents and Settings\zorro\certreq.csr).

Step 3: Submit your CSR to a signing authority (e.g., Verisign or Thawte)

Step 4: Import your signed certificate into your keystore (must be in PKCS#7 format)
Import the certificate into your keystore file as follows (where cert.crt is your certificate file):

```
keytool –import –alias meteor –trustcerts –file cert.crt –keystore [keystore name]
```

If you receive the error 'unsupported encoding,' add a line break after the last line in the certificate using Notepad or equivalent text editor (keytool is very sensitive to white space).

Step 5: Export your public key
Your public key is used in the Meteor registry to verify communication with your server. You will need to export your public key and send it to us during the Meteor registration process.

Export your key as follows (where [public key name].key is the public key file name):

```
keytool –export –alias meteor –file [public key name].key –keystore [keystore name]
```

When prompted, enter your keystore password. Press 'Enter' and you will see the "Certificate Stored…" message. Your public key file will be saved to the Java SDK/bin directory. In a Windows environment, it will be saved to your Documents and Settings folder (e.g., c:\Documents and Settings\zorro\[public key name].key).

NOTE: *When configuring Meteor with a key created using these steps, you should specify the keystore type as JKS.*

Document Change Log

Date	Version	Person	Changes
1/6/2005	0.8	Mike Mazanec	Initial Release
1/25/2005	0.9	Mike Mazanec	Added Appendices D (Deploying WARS) & E (.properties modifications)
1/27/05	1.0	Mike Mazanec	Added stand-alone .properties modifications, SDk-specific changes and JSSE installation
1/31/05	1.01	Mike Mazanec	Added .properties notes for version 3.0
2/14/05	1.02	Mike Mazanec	Corrected typo for JAVA_HOME
2/2205	1.03	Mike Mazanec	Updated .properties changes
2/23/05	1.04	Mike Mazanec	Fixed typos, added properties for XML Data Provider data (vs. MySQL) and Access Provider keystore and modifications for Access Provider web.xml (for use with SAML)
2/25/05	1.05	Mike Mazanec	Added directions for self-signing the certificate, changed directorydata.properties modifications, added in-document hyperlinks from TOC
3/16/05	1.06	Mike Mazanec	Removed directions for creating CATALINA_HOME, lines for using SampleDataServer and FileDataProvider for DP and notes regarding the MySQL connector and self-signed certificates.
4/7/05	1.07	Mike Mazanec	Modified Appendix F to reflect .default suffix on property files, and Appendix E to reflect Tomcat 5.x doesn't need JAR files
5/11/07	1.08	Tech Team	Revised to make current and checked for accuracy
03/12/08	1.09	Chris Cooper	Revised for changes to Meteor v3.3
03/12/08	1.10	Rick Allen	Revised after Tech Team conference call
03/17/08	1.10	Tech Team	Revised after Tech Team conference call
03/19/08	1.10	Tech Team	Revised after Tech Team conference call
5/11/2011	1.11	Tim Cameron	Updated logo and NCHELP references.
10/24/2011	2.0	Rick Allen	Added new properties files names and content
11/23/2011	2.0	John Lazos	Revised for Meteor 4.0
01/12/2012	2.0	Tim Cameron	Integration of formatting edits