

Distributed Data Framework: Quick Start Guide

Table of Contents

License	1
1. Quick Start Tutorial	2
1.1. Installing (Quick Start)	2
1.1.1. Quick Install Prerequisites	2
1.1.2. Quick Install of SolrCloud	2
1.1.3. Quick Install of DDF	3
1.1.4. Quick Install of DDF on a remote headless server	4
1.2. Certificates (Quick Start)	5
1.2.1. Demo Certificate Authority (CA)	5
1.2.1.1. Creating New Server Keystore Entry with the CertNew Scripts	5
1.2.1.2. Dealing with Lack of DNS	7
1.2.2. Creating Self-Signed Certificates	8
1.2.2.1. Creating a custom CA Key and Certificate	8
1.2.2.2. Sign Certificates Using the custom CA	8
1.2.3. Updating Settings After Changing Certificates	9
1.3. Configuring (Quick Start)	9
1.4. Adding New User Roles	10
Index	11

License

Copyright (c) Codice Foundation.

This work is licensed under a [Creative Commons Attribution 4.0 International License](#).

This document last updated: 2026-08-03.

1. Quick Start Tutorial

This quick tutorial enables installation, configuration, and usage of a basic instance of DDF.

NOTE

This tutorial is intended for setting up a test, demonstration, or trial installation of DDF. For complete installation and configuration steps, see [Installing](#).

These steps demonstrate:

- ✓ [Prerequisites](#).
- ✓ [Quick Install of SolrCloud](#).
- ✓ [Quick Install of DDF](#).

1.1. Installing (Quick Start)

These are the basic requirements to set up the environment to run a DDF.

WARNING

For security reasons, DDF cannot be started from a user's home directory. If attempted, the system automatically shuts down.

1.1.1. Quick Install Prerequisites

Hardware Requirements (Quick Install)

- At least 4096 MB of memory for DDF.
 - This amount can be increased to support memory-intensive applications. See [Memory Considerations](#).

Java Requirements (Quick Install)

Follow the instructions outlined in [Java Requirements](#).

WARNING

Check System Time

Prior to installing DDF, ensure the system time is accurate to prevent federation issues.

1.1.2. Quick Install of SolrCloud

1. Download a preconfigured Solr distribution [zip file](#) [↗](#).
2. Unzip the Solr zip file.

3. Run `<solr_directory>/bin/solr -e cloud`.
 - a. Press enter to default to 2 nodes.
 - b. Enter 8994 for node 1 port.
 - c. Enter 8995 for node 2 port.
 - d. Press enter for all other prompts to accept defaults.

1.1.3. Quick Install of DDF

WARNING

The DDF installation cannot succeed unless SolrCloud is running. See [Quick Install of SolrCloud](#) for instructions.

1. Download the DDF [zip file](#) .
2. Install DDF by unzipping the zip file.

Windows Zip Utility Warning

The Windows Zip implementation, which is invoked when a user double-clicks on a zip file in the Windows Explorer, creates a corrupted installation. This is a consequence of its inability to process long file paths. Instead, use the java jar command line utility to unzip the distribution (see example below) or use a third party utility such as 7-Zip.

WARNING

NOTE

If and only if a JDK is installed, the jar command may be used; otherwise, another archiving utility that does not have issue with long paths should be installed

Use Java to Unzip in Windows(Replace `<PATH_TO_JAVA>` with correct path and `<JAVA_VERSION>` with current version.)

```
"<PATH_TO_JAVA>\jdk<JAVA_VERSION>\bin\jar.exe" xf ddf-2.31.1.zip
```

3. This creates an installation directory, which is typically created with the name and version of the application. This installation directory is referred to as `<DDF_HOME>`. (Substitute the actual directory name.)
4. Edit `<DDF_HOME>/etc/custom.system.properties` and update `solr.cloud.zookeeper=localhost:2181` to `solr.cloud.zookeeper=localhost:9994`
5. Start DDF by running the `<DDF_HOME>/bin/ddf` script (or `ddf.bat` on Windows).
6. Startup may take a few minutes.
 - a. Optionally, a `system:wait-for-ready` command (aliased to `wfr`) can be used to wait for startup to

complete.

7. The Command Console displays.

Command Console Prompt

```
ddf@local>
```

1.1.4. Quick Install of DDF on a remote headless server

If DDF is being installed on a remote server that has no user interface, the hostname must be updated in the configuration files and certificates.

NOTE Do not replace *all* instances of **localhost**, only those specified.

Configuring with a new hostname

1. Update the `<DDF_HOME>/etc/custom.system.properties` file. The entry `org.codice.ddf.system.hostname=localhost` should be updated to `org.codice.ddf.system.hostname=<HOSTNAME>`.
2. Update the `<DDF_HOME>/etc/users.properties` file. Change the `localhost=localhost[...]` entry to `<HOSTNAME>=<HOSTNAME>`. (Keep the rest of the line as is.)
3. Update the `<DDF_HOME>/etc/users.attributes` file. Change the `"localhost"` entry to `"<HOSTNAME>"`.
4. From the console go to `<DDF_HOME>/etc/certs` and run the appropriate script.
 - a. *NIX: `sh CertNew.sh -cn <hostname> -san "DNS:<hostname>"`.
 - b. Windows: `CertNew -cn <hostname> -san "DNS:<hostname>"`.
5. Proceed with starting the system and continue as usual.

Configuring with an IP address

1. Update the `<DDF_HOME>/etc/custom.system.properties` file. The entry `org.codice.ddf.system.hostname=localhost` should be updated to `org.codice.ddf.system.hostname=<IP>`.
2. Update the `<DDF_HOME>/etc/users.properties` file. Change the `localhost=localhost[...]` entry to `<IP>=<IP>`. (Keep the rest of the line as is.)
3. Update the `<DDF_HOME>/etc/users.attributes` file. Change the `"localhost"` entry to `"<IP>"`.
4. From the console go to `<DDF_HOME>/etc/certs` and run the appropriate script.
 - a. *NIX: `sh CertNew.sh -cn <IP> -san "IP:<IP>"`.
 - b. Windows: `CertNew -cn <IP> -san "IP:<IP>"`.
5. Proceed with starting the system and continue as usual.

File Descriptor Limit on Linux

- For Linux systems, increase the file descriptor limit by editing `/etc/sysctl.conf` to include:

```
fs.file-max = 6815744
```

NOTE

- (This file may need permissions changed to allow write access).
- For the change to take effect, a restart is required.

1. *nix Restart Command

```
init 6
```

1.2. Certificates (Quick Start)

DDF comes with a default keystore that contains certificates. This allows the distribution to be unzipped and run immediately. If these certificates are sufficient for testing purposes, proceed to [Configuring \(Quick Start\)](#).

To test federation using 2-way TLS, the default keystore certificates needs to be replaced, using either the included [Demo Certificate Authority](#) or by [Creating Self-signed Certificates](#).

If the installer was used to install the DDF and a hostname other than "localhost" was given, the user is prompted to upload new trust/key stores.

If the hostname is `localhost` or, if the hostname was changed *after* installation, the default certificates do not allow access to the DDF instance from another machine over HTTPS (now the default for many services). The Demo Certificate Authority must be replaced with certificates that use the fully-qualified hostname of the server running the DDF instance.

1.2.1. Demo Certificate Authority (CA)

DDF comes with a populated truststore containing entries for many public certificate authorities, such as Go Daddy and Verisign. It also includes an entry for the DDF Demo Root CA. This entry is a self-signed certificate used for testing. It enables DDF to run immediately after unzipping the distribution. The keys and certificates for the DDF Demo Root CA are included as part of the DDF distribution. This entry must be removed from the truststore before DDF can operate securely.

1.2.1.1. Creating New Server Keystore Entry with the CertNew Scripts

To create a private key and certificate signed by the Demo Certificate Authority, use the provided

scripts. To use the scripts, run them out of the `<DDF_HOME>/etc/certs` directory.

*NIX Demo CA Script

For *NIX, use the `CertNew.sh` script.

```
sh CertNew.sh [-cn <cn>|-dn <dn>] [-san <tag:name,tag:name,...>]
```

where:

- `<cn>` represents a fully qualified common name (such as "`<FQDN>`", where `<FQDN>` could be something like `cluster.yoyo.com`)
- `<dn>` represents a distinguished name as a comma-delimited string (such as "`c=US, st=California, o=Yoyodyne, l=San Narciso, cn=<FQDN>`")
- `<tag:name,tag:name,...>` represents optional subject alternative names to be added to the generated certificate (such as "`DNS:<FQDN>,DNS:node1.<FQDN>,DNS:node2.<FQDN>`"). The format for subject alternative names is similar to the OpenSSL X509 configuration format. Supported tags are:
 - `email` - email subject
 - `URI` - uniformed resource identifier
 - `RID` - registered id
 - `DNS` - hostname
 - `IP` - ip address (V4 or V6)
 - `dirName` - directory name

If no arguments specified on the command line, `hostname -f` is used as the common-name for the certificate.

Windows Demo CA Script

For Windows, use the `CertNew.cmd` script.

```
CertNew (-cn <cn>|-dn <dn>) [-san "<tag:name,tag:name,...>"]
```

where:

- `<cn>` represents a fully qualified common name (such as "`<FQDN>`", where `<FQDN>` could be something like `cluster.yoyo.com`)
- `<dn>` represents a distinguished name as a comma-delimited string (such as "`c=US, st=California, o=Yoyodyne, l=San Narciso, cn=<FQDN>`")
- `<tag:name,tag:name,...>` represents optional subject alternative names to be added to the

generated certificate (such as "DNS:<FQDN>,DNS:node1.<FQDN>,DNS:node2.<FQDN>"). The format for subject alternative names is similar to the OpenSSL X509 configuration format. Supported tags are:

- **email** - email subject
- **URI** - uniformed resource identifier
- **RID** - registered id
- **DNS** - hostname
- **IP** - ip address (V4 or V6)
- **dirName** - directory name

The **CertNew** scripts:

- Create a new entry in the server keystore.
- Use the hostname as the fully qualified domain name (FQDN) when creating the certificate.
- Adds the specified subject alternative names if any.
- Use the Demo Certificate Authority to sign the certificate so that it is trusted by the default configuration.

To install a certificate signed by a different Certificate Authority, see [Managing Keystores](#).

After this proceed to [Updating Settings After Changing Certificates](#).

WARNING

If the server's fully qualified domain name is not recognized, the name may need to be added to the network's DNS server.

1.2.1.2. Dealing with Lack of DNS

In some cases DNS may not be available, and the system needs to be configured to work with IP addresses.

Options can be given to the **CertNew** Scripts to generate certs that work in this scenario.

***NIX**

From <DDF_HOME>/etc/certs/ run:

```
sh CertNew.sh -cn <IP> -san "IP:<IP>"
```

Windows

From <DDF_HOME>/etc/certs/ run:

```
CertNew -cn <IP> -san "IP:<IP>"
```

After this proceed to [Updating Settings After Changing Certificates](#), and be sure to use the IP address instead of the FQDN.

1.2.2. Creating Self-Signed Certificates

If using the Demo CA is not desired, DDF supports creating self-signed certificates with a self-signed certificate authority. This is considered an advanced configuration.

Creating self-signed certificates involves creating and configuring the files that contain the certificates. In DDF, these files are generally Java Keystores (**jks**) and Certificate Revocation Lists (**crl**). This includes commands and tools that can be used to perform these operations.

For this example, the following tools are used:

- openssl
 - Windows users can use: [openssl](#) for windows.
- The standard Java [keytool certificate management utility](#) [↗](#).
- [Portecle](#) can be used for **keytool** operations if a GUI is preferred over a command line interface.

1.2.2.1. Creating a custom CA Key and Certificate

The following steps demonstrate creating a root CA to sign certificates.

1. Create a key pair.

```
$> openssl genrsa -aes128 -out root-ca.key 1024
```

2. Use the key to sign the CA certificate.

```
$> openssl req -new -x509 -days 3650 -key root-ca.key -out root-ca.crt
```

1.2.2.2. Sign Certificates Using the custom CA

The following steps demonstrate signing a certificate for the **tokenissuer** user by a CA.

1. Generate a private key and a Certificate Signing Request (CSR).

```
$> openssl req -newkey rsa:1024 -keyout tokenissuer.key -out tokenissuer.req
```

2. Sign the certificate by the CA.

```
$> openssl ca -out tokenissuer.crt -infiles tokenissuer.req
```

These certificates are used during system configuration to replace the default certificates.

1.2.3. Updating Settings After Changing Certificates

After changing the certificates, it is necessary to update the system user and the `org.codice.ddf.system.hostname` property with the value of either the FQDN or the IP.

FQDNs should be used wherever possible. In the absence of DNS, however, IP addresses can be used.

Replace `localhost` with the FQDN or the IP in `<DDF_HOME>/etc/users.properties`, `<DDF_HOME>/etc/users.attributes`, and `<DDF_HOME>/etc/custom.system.properties`.

TIP

On linux this can be accomplished with a single command: `sed -i 's/localhost/<FQDN|IP>/g' <DDF_HOME>/etc/users.* <DDF_HOME>/etc/custom.system.properties`

Finally, restart the DDF instance. Navigate to the Admin Console to test changes.

1.3. Configuring (Quick Start)

Set the configurations needed to run DDF.

1. In a browser, navigate to the Admin Console at `https://{FQDN}:{PORT}/admin`.
 - a. The Admin Console may take a few minutes to start up.
2. Enter the default username of `admin` and the password of `admin`.
3. Follow the installer prompts for a standard installation.
 - a. Click start to begin the setup process.
 - b. Configure `guest claims attributes` or use defaults.
 - i. See [Configuring Guest Access](#) for more information about the Guest user.
 - ii. **All users are automatically granted these permissions.**
 - iii. **Guest users are not able to ingest data with more restrictive markings than the guest claims.**
 - iv. **Any data ingested that has more restrictive markings than these guest claims is not visible to Guest users.**
 - c. Select **Standard Installation**.
 - i. This step may take several minutes to complete.
 - d. On the System Configuration page, configure any port or protocol changes desired and add any keystores/truststores needed.
 - i. See [Certificates \(Quick Start\)](#) for more details.
 - e. Click **Next**

f. Click **Finish**

1.4. Adding New User Roles

Users or user groups can be customized to enable or restrict access to specific certain system functions. Define a custom role and assign it to users who need specific access.

For an example, limit the replication admin view to users with a `sr_analyst` role.

Define a new role

1. Navigate to the Admin Console.
2. Navigate to the Security app.
3. Navigate to the **Configuration** tab.
4. Select the **Web Context Policy Manager** Configuration.
5. Select **Required Attributes**.
6. Define the contexts to be restricted.
 - a. Enter `<CONTEXT>={http://schemas.xmlsoap.org/ws/2005/05/identity/claims/role=<ROLE_NAME>`
 - i. For example, enter `/admin/replication={http://schemas.xmlsoap.org/ws/2005/05/identity/claims/role=sr_analyst}`
 - ii. Select the + sign to add another required attribute.
 - iii. Enter `/admin/hub/graphql={http://schemas.xmlsoap.org/ws/2005/05/identity/claims/role=sr_analyst}`

Assign the new role to users

1. Add the new role (`sr_analyst`) to selected users in the `users.properties` file. Add the role to the line of each chosen user.
 - a. For example, `user1=password,sr_analyst`
2. Restart Alliance to apply the changes.
 - a. `/app/bin/stop`
 - b. `/app/bin/start`

In the example given, the `/admin/replication` context is only accessible to users with the `sr_analyst` role. Other roles can be designated as needed by substituting the `<CONTEXT>` and the `<ROLE_NAME>` as needed.

Index

A

Adding New User Roles, [10](#)

D

Demo Certificate Authority, [5](#)

Q

Quick Install of DDF, [3](#)

Quick Install of DDF on a remote headless server,
[4](#)

Quick Install of SolrCloud, [2](#)

Quick Install Prerequisites, [2](#)

Quick Start Certificates, [5](#)

Quick start configuration, [9](#)

Quick start guide, [2](#)

S

Self-Signed Certificates, [8](#)