



Documentation for Crowd 4.4

Contents

Crowd documentation	7
Getting started with Crowd	8
Installation and Upgrade Guide	13
Installing Crowd	14
Supported Platforms	15
Setting JAVA_HOME	18
End of support announcements for Crowd	19
Installing Crowd and CrowdID	20
Connecting Crowd to a Database	22
Connecting CrowdID to a Database	29
Specifying your Crowd Home Directory	39
Running the Setup Wizard	40
Troubleshooting your Configuration on Setup	49
Configuring Crowd	51
Important directories and files	52
Changing the Port that Crowd uses	64
Configuring Crowd to Work with SSL	66
Installing Crowd as a Windows Service	69
Setting Crowd to Start Automatically on Mac OS X	76
Setting Crowd to Run Automatically and Use an Unprivileged System User on UNIX	79
Upgrading Crowd	81
Upgrading Crowd via Automatic Database Upgrade	82
Upgrading Crowd via XML Data Transfer	84
Crowd 4.4 Upgrade Notes	87
Migrate to Another Database	88
Migrating Crowd Between Servers	89
Migrating from OnDemand to a Crowd installed site	90
Installing Crowd Data Center	91
Migrate from Server to Data Center	95
Crowd 4.4 Release Notes	99
Administration Guide	102
Getting Started	105
Concepts	106
Supported Applications and Directories	108
About the Crowd Administration Console	109
Managing Directories	110
Using the Directory Browser	111
Adding a Directory	112
Configuring an Internal Directory	114
Configuring an LDAP Directory Connector	116
Configuring a Remote Crowd Directory	129
Configuring a Custom Directory Connector	134
Configuring a Delegated Authentication Directory	136
Configuring Azure Active Directory	139
Configuring Caching for an LDAP Directory	143
Using Naive DN Matching	148
Specifying Directory Permissions	150
Importing Users and Groups into a Directory	153
Importing Users from Atlassian Confluence	154
Importing Users from Atlassian Jira	156
Importing Users from Atlassian Bamboo	158
Importing Users from Jive Forums	160
Importing Users from CSV Files	162
Importing Users from One Crowd Directory into Another	171
Configuring directories for failover authentication	173
Pruning delegated directories	175
Managing Applications	176

Using the Application Browser	177
Adding an Application	181
Integrating Crowd with Atlassian Bamboo	185
Integrating Crowd with Atlassian Confluence	190
Integrating Crowd with Atlassian CrowdID	198
Integrating Crowd with Atlassian Crucible	201
Integrating Crowd with Atlassian FishEye	202
Integrating Crowd with Atlassian Jira	209
Integrating Crowd with Atlassian Bitbucket Server	218
Integrating Crowd with Acegi Security	220
Integrating Crowd with Jive Forums	225
Integrating Crowd with Spring Security	232
Integrating Crowd with a Custom Application	244
Integrating Crowd with Atlassian HipChat	246
Configuring the Google Apps Connector	252
Mapping a Directory to an Application	258
Specifying the Directory Order for an Application	260
Specifying an Application's Directory Permissions	262
Viewing Users in Directories Mapped to an Application	266
Specifying which Groups can access an Application	267
Syncing users based on their access rights	268
Effective memberships with multiple directories	270
Specifying an Application's Address or Hostname	273
Testing a User's Login to an Application	275
Enforcing Lower-Case Usernames and Groups for an Application	277
Managing an Application's Session	279
Deleting or Deactivating an Application	280
Configuring Caching for an Application	283
Overview of SSO	285
Configuring Options for an Application	289
Enabling OpenID client app	290
Allowing applications to create user tokens	291
Disabling the OpenID client app	292
Configuring how users log in	293
Managing Users and Groups	294
Using the User Browser	295
Adding a User	296
Editing a User's Details and Password	297
Deleting or Deactivating a User	298
Case Sensitivity of Usernames and Groups	299
Specifying a User's Aliases	300
Editing a User's Group Membership	302
Managing Groups	303
Deleting a Group	304
Adding a Group	305
Managing Group Members	306
Automatically Assigning Users to Groups	308
Adding Users to a Group	313
Removing Users from a Group	315
Nested Groups in Crowd	317
Adding a Sub-Group	321
Group-level administration	323
Removing a Sub-Group	326
Specifying a User's Attributes	327
Granting Crowd Administration Rights to a User	328
Granting Crowd User Rights to a User	329
Managing a User's Session	330
System Administration	331
Configuring Server Settings	332
Deployment Title	333
Domain	334
Session configuration	336
Authorization Caching	338
Licensing	339

Crowd SSO 2.0	341
Finding your SEN	346
SSO Cookie	347
Configuring your Mail Server	348
Creating an Email Notification Template	352
Configuring Trusted Proxy Servers	354
Viewing Crowd's System Information	355
Backing Up and Restoring Data	356
Logging and Profiling	358
Performance Profiling	362
Draft - Troubleshooting and Requesting Technical Support	363
Configuring the LDAP Connection Pool	366
Browsing the audit log	369
Look and feel	371
Overview of Caching	372
Crowd Security Advisories and Fixes	374
Crowd Security Advisory 2010-07-05	375
Crowd Security Advisory 2010-05-04	377
Crowd Security Advisory 2008-10-14 - Parameter Injection Vulnerability	379
Crowd Security Advisory 2012-05-17	380
Crowd Security Notice 2013-07-01	382
Crowd Security Advisory 2013-07-16	383
Crowd Security Advisory 2014-05-21	385
Crowd Security Advisory 2016-10-19	387
Crowd Security Advisory 2017-03-10	391
Crowd Security Advisory 2019-05-22	394
Constructing cron expressions in Crowd	400
User Guide	402
Introduction to Crowd	404
Logging in to Crowd	406
Logging out of Crowd	408
Changing or Resetting your Password	409
Changing your Password	410
Resetting Forgotten Passwords	411
Requesting usernames	412
Updating your User Profile	413
Viewing your Group Membership	414
Viewing your Applications	415
Crowd User's Glossary	416
Alias	418
Authorization to Use Crowd	419
Crowd Administrator	420
Crowd-Connected Application	421
Directory	422
Group	423
Role	424
Self-Service Console	425
Single Sign-On	426
Monitoring license usage	427
Password encryption	429
CrowdID Administration Guide	431
1. About CrowdID	432
1.1 How CrowdID works with Crowd	433
1.1.1 Determining the name of the CrowdID application	434
1.1.2 Locating the Crowd Server that CrowdID is using	436
1.2 How OpenID sites interact with CrowdID	438
1.3 Lightweight OpenID server	439
2. Allowing users to access CrowdID	440
2.1 Granting CrowdID access rights to a user	441
2.2 Granting CrowdID Administration Rights to a User	443
3. Specifying the sites to which users can log in	444
3.1 Allowing all hosts	445
3.2 Allowing all except specified hosts ('Blacklist')	446
3.3 Allowing specified hosts only ('Whitelist')	447

3.4 Approval Whitelist	449
4. Configuring CrowdID system settings	450
4.1 Specifying the CrowdID URL	451
4.2 Enabling localhost authentication	452
4.3 Enabling immediate authentication requests	454
4.4 Enabling communication with stateless clients	456
CrowdID User Guide	458
1. Getting started with CrowdID	459
1.1 What is OpenID?	460
1.2 What is CrowdID?	461
1.3 What is an OpenID URL or identifier?	462
1.4 Viewing the CrowdID page	463
2. Logging in to a website using OpenID	465
2.1 Does the website support OpenID?	466
2.2 Entering your OpenID URL	467
2.3 Logging in to CrowdID	468
2.4 Allowing or denying a login	469
2.5 Providing additional profile information to a website	471
3. Viewing your always-approved websites	472
4. Viewing your login history	474
5. Updating your profile	476
6. Using more than one profile	478
6.1 Adding a profile	479
6.2 Choosing a profile for a website	481
6.3 Setting a default profile	482
6.4 Deleting a profile	484
7. Changing or resetting your password	486
7.1 Changing your password	487
7.2 Resetting your password	489
8. Requesting Forgotten Usernames	490
Crowd FAQ	491
Crowd Resources	495
Deployment FAQ	496
Deploying Multiple Atlassian Applications in a Single Tomcat Container	497
Finding the atlassian-crowd.log File	498
Finding your Crowd home and shared directories	499
Removing the 'crowd' Context from the Application URL	500
Resetting the Domain Cookie Value	501
Restarting the Setup Wizard from Scratch	502
Self Signed Certificate	503
Guides, Hints and Tips	504
How to Print Only Tomcat Logs into Crowd's catalina.out	505
Principals and Users	507
Using Apache Directory Studio for LDAP Configuration	508
Creating a Connection to your LDAP Directory	509
Getting an LDIF Export of a User or Group	514
Restricting LDAP Scope for User and Group Search	515
Integration FAQ	519
All Integrations	520
If I delete a user from Crowd, how will this affect integrated applications?	521
Passing the crowd.properties File as an Environment Variable	522
Atlassian Product Integration	523
Application Caching	524
Jira integration	525
Public Signup Setup	526
IBM Lotus Domino Integration	527
IBM Websphere Integration	528
Support Policies	529
Bug Fixing Policy	530
How to Report a Security Issue	535
New Features Policy	536
Security Advisory Publishing Policy	539
Security Bugfix Policy	540
Security Patch Policy	541

Severity Levels for Security Issues	542
Troubleshooting	543
Finding Known Issues	544
Characters in User or Group DN's that will cause problems when using Crowd	545
Problems when Importing Users into MySQL	547
Troubleshooting LDAP Error Codes	548
Active Directory LDAP Errors	549
Troubleshooting LDAP User Management	550
Troubleshooting SSL certificates and Crowd	559
How to Optimize Crowd Client Caching	560
Troubleshooting Crowd Performance	561
Troubleshooting SSO with Crowd	563
Debugging SSO in environments with Proxy Servers	564
Troubleshooting CrowdID	566
Provide Crowd Information to Atlassian Support	567
Contributing to the Crowd Documentation	568
Tips of the Trade	569
Crowd Documentation in Other Languages	572
Crowd Data Center	573
Crowd user management	575

Crowd documentation

Manage users from multiple directories - Active Directory, LDAP, Crowd - via a single admin console, and control application permissions from the same place.

Get started

New to Crowd? Check out our guides for new administrators and users.

[View guides](#)

Whats new

Time to upgrade? Get the low down on the latest and greatest in **Crowd 4.4**.

[View latest changes](#)

Getting started with Crowd

Thank you for choosing Crowd. To help you get up and running quickly, we have compiled some quick-start instructions on configuring and using Crowd with your [Jira](#) and [Confluence](#) applications.

i This quick-start guide assumes that you have installed and set up a Jira application and/or Confluence and now wish to set up Crowd for user management in one or both of them.

- If you want to use a Jira application or Confluence but have not yet installed them, please follow the instructions in [Installing Jira applications](#) and/or [Confluence Installation Guide](#) before configuring Crowd.
- If you want to use Crowd with other applications but not a Jira application or Confluence, please follow the detailed Crowd [installation and setup guide](#) rather than this getting started guide.

Getting Started

1. Installing Crowd

1. Go to the [Crowd download page](#).
2. Download the ZIP archive file for the Crowd distribution (not EAR-WAR).
3. Unzip the zip archive into a directory of your choice, avoiding spaces in the directory name.
4. Tell Crowd where to find its Crowd Home directory, by editing the `crowd-init.properties` file as described in the [installation guide](#).
5. Set up your database as described in the [database configuration guide](#).
i This quick-start page assumes that you have an existing Jira application or Confluence application. So we recommend that you connect Crowd to a production-ready database and not HSQLDB. But if you are evaluating Crowd, it is fine to use HSQLDB and then move to a different database later. In that case, you do not need to do anything in this step, because Crowd contains everything you need.
6. Start your Crowd server by going to the directory where you unzipped Crowd and running `start_crowd.bat`.
7. To access Crowd, go to your web browser and type this address: `http://localhost:8095/crowd`.
8. Follow the [Setup Wizard](#). This will guide you through the process of setting up your Crowd server and creating an admin user.

For more help on the technical procedures in this section, please refer to the [Crowd installation guide](#).

If you need assistance, please [create a support ticket](#).

1. Go to the [Crowd download page](#).
2. Click the 'Mac OS X' tab and download the TAR.GZ archive file for the Crowd distribution (not EAR-WAR).
3. Unzip the archive into a directory of your choice, avoiding spaces in the directory name.
4. Tell Crowd where to find its Crowd Home directory, by editing the `crowd-init.properties` file as described in the [installation guide](#).
5. Set up your database as described in the [database configuration guide](#).
i This quick-start page assumes that you have an existing Jira or Confluence application. So we recommend that you connect Crowd to a production-ready database and not HSQLDB. But if you are evaluating Crowd, it is fine to use HSQLDB and then move to a different database later. In that case, you do not need to do anything in this step, because Crowd contains everything you need.
6. Start your Crowd server by going to the directory where you unzipped Crowd and double-clicking `start_crowd.sh`.
7. To access Crowd, go to your web browser and type this address: `http://localhost:8095/crowd`.
8. Follow the [Setup Wizard](#). This will guide you through the process of setting up your Crowd server and creating an admin user.

For more help on the technical procedures in this section, please refer to the [Crowd installation guide](#).

If you need assistance, please [create a support ticket](#).

1. Go to the [Crowd download page](#).
2. Click the 'Linux' tab and download the TAR.GZ Archive file for the Crowd distribution (not EAR-WAR).
3. Unzip the archive into a directory of your choice, avoiding spaces in the directory name.

4. Tell Crowd where to find its Crowd Home directory, by editing the `crowd-init.properties` file as described in the [installation guide](#).
5. Set up your database as described in the [database configuration guide](#).
 - 📘 This quick-start page assumes that you have an existing Jira or Confluence application. So we recommend that you connect Crowd to a production-ready database and not HSQLDB. But if you are evaluating Crowd, it is fine to use HSQLDB and then move to a different database later. In that case, you do not need to do anything in this step, because Crowd contains everything you need.
6. Start your Crowd server by going to the directory where you unzipped Crowd and double-clicking `start_crowd.sh`.
7. To access Crowd, go to your web browser and type this address: `http://localhost:8095/crowd`.
8. Follow the [Setup Wizard](#). This will guide you through the process of setting up your Crowd server and creating an admin user.

For more help on the technical procedures in this section, please refer to the [Crowd installation guide](#).

If you need assistance, please [create a support ticket](#).

2. Adding Users and Groups

Crowd is designed to help you manage users and groups across multiple applications. Your next step is to configure a user directory in Crowd to contain your Jira application and/or Confluence users and groups.

1. [Add a Crowd directory](#) Add a Crowd Internal directory to contain all your Jira and Confluence users.
2. [Add the Confluence groups](#) Add the 'confluence-users' and 'confluence-administrators' groups to your new directory.
3. [Add the JIRA groups](#) Add the 'jira-users', 'jira-developers' and 'jira-administrators' groups to your new directory.
4. [Import your users from a CSV file](#) or [add them manually](#).
5. [Add the users to the groups](#) Use Crowd's bulk user management to add all the users to the 'confluence-users' and 'jira-users' groups. Also add any administrators to the administration groups and add the developers to the 'jira-developers' group. For more details about the permissions applicable to each group, refer to the [Confluence](#) and [Jira](#) documentation.

If your Jira users are currently managed via Jira's internal management and your Confluence users are managed separately via Confluence's internal management, you can use Crowd to simplify and centralize your user and group management:

1. [Add a Crowd directory](#) Use the Crowd Administration Console to add a Crowd Internal directory to contain all your Jira and Confluence users.
2. [Import the users and groups from Confluence](#) Use the Crowd importer to copy your users and groups from Confluence into the new Crowd directory. This process will also copy the group memberships into Crowd.
3. [Import the users and groups from JIRA](#) Use the Crowd importer to copy your users and groups from Jira into the same Crowd directory as the Confluence users. This process will add any additional users and groups from Jira and update the existing Confluence users with their Jira group memberships.
4. [Check your users and groups in Crowd](#) Use Crowd's group browser to check that your users, groups and group memberships are available as expected in Crowd.

If your Jira and Confluence users are currently all managed via Jira's internal management, you can use Crowd to simplify and centralize your user and group management:

1. [Add a Crowd directory](#) Use the Crowd Administration Console to add a Crowd Internal directory to contain all your Jira and Confluence users.
2. [Import the users and groups from JIRA](#) Use the Crowd importer to copy your users and groups from Jira into the new Crowd directory. This process will also copy the group memberships into Crowd.
3. [Check your users and groups in Crowd](#) Use Crowd's group browser to check that your users, groups and group memberships are available as expected in Crowd.

If your users are in a corporate LDAP directory, you can choose whether you want to store your groups in LDAP or in Crowd.

- If you want to store your users and groups in LDAP:
 1. [Add a Crowd LDAP directory connector](#) Choose the options for your specific version of LDAP, such as Microsoft Active Directory or Novell eDirectory. Crowd supports a number of LDAP flavors, as listed in the [documentation](#).
 2. [Check your users and groups in Crowd](#) Use Crowd's group browser to check that your users, groups and group memberships are available as expected in Crowd.
- If you want to store your users in LDAP and your groups in Crowd:
 1. [Add a Crowd Delegated Authentication directory](#) Choose the options for your specific version of LDAP, such as Microsoft Active Directory or Novell eDirectory. Crowd supports a number of LDAP flavors, as listed in the [documentation](#).
 2. [Add the Confluence groups](#) Add the 'confluence-users' and 'confluence-administrators' groups to your new Crowd Delegated Authentication directory.
 3. [Add the JIRA groups](#) Add the 'jira-users', 'jira-developers' and 'jira-administrators' groups to your new Crowd Delegated Authentication directory.
 4. [Add the users to the groups](#) Use Crowd's bulk user management to add all the users to the 'confluence-users' and 'jira-users' groups. Also add any administrators to the administration groups and add the developers to the 'jira-developers' group. For more details about the permissions applicable to each group, refer to the [Confluence](#) and [JIRA](#) documentation.

Take the following steps, choosing your directory and other options as indicated in the linked documentation:

1. [Add a Crowd directory](#) Choose the directory type you need to contain all your JIRA and Confluence users.
2. Add your users and groups either via Crowd's importer or manually:
 - [Import your users and groups](#) into Crowd.
 - Or do it manually:
 - a. [Add the users](#).
 - b. [Add the Confluence groups](#) Add the 'confluence-users' and 'confluence-administrators' groups to your new directory.
 - c. [Add the JIRA groups](#) Add the 'jira-users', 'jira-developers' and 'jira-administrators' groups to your new directory.
 - d. [Add the users to the groups](#) Use Crowd's bulk user management to add all the users to the 'confluence-users' and 'jira-users' groups. Also add any administrators to the administration groups and add the developers to the 'jira-developers' group. For more details about the permissions applicable to each group, refer to the [Confluence](#) and [JIRA](#) documentation.

i If you have Confluence or JIRA, but not both, pick the scenario above that best matches your requirements, then just skip the steps for the application that you do not need.

3. Connecting the Applications

Crowd manages your users' access to your applications and makes single sign-on (SSO) possible. (More about SSO [below](#).) For this to happen, you need to tell Crowd about the applications and to copy some Crowd libraries into the applications' installation folders.

1. [Add Confluence](#) Add the Confluence application to Crowd, following the instructions in the [Add Application Wizard](#).
 - Choose 'Confluence' as the application type.
 - In the 'Directories' step, choose the user directory you added for Confluence.
 - In the 'Authorization' step, allow all users to authenticate.
2. [Configure the Crowd libraries in Confluence](#) Copy the Crowd client libraries into your Confluence folders and configure the properties files as described on the [Confluence integration page](#).
3. Now [add JIRA](#) Add the JIRA application to Crowd, following the instructions in the [Add Application Wizard](#).
 - Choose 'JIRA' as the application type.
 - In the 'Directories' step, choose the user directory you added for JIRA.
 - In the 'Authorization' step, allow all users to authenticate.
4. [Configure the Crowd libraries in JIRA](#) Copy the Crowd client libraries into your JIRA folders and configure the properties files as described on the [JIRA integration page](#).

i We will call these your 'Crowd-connected applications'.

Mastering the Basics

4. Examining your Crowd Server Setup

Go to the [System Information](#) screen in Crowd's Administration Console to find useful information about your Crowd server, such as the location of your Crowd Home directory, information about your database and JVM, and your license server ID.

5. Managing SSO

If you have configured single sign-on (SSO) when setting up your Crowd-connected applications (JIRA and Confluence) in step 3 above, your users will only need to log in or log out once, to Crowd or any Crowd-connected application. When they start another Crowd-connected application, they will be logged in automatically. Similarly, when they log out of Crowd or one of the Crowd-connected applications, they will be logged out of Crowd and the other application(s) at the same time.

- [Overview of SSO](#) An overview of Crowd's SSO capabilities, plus links to detailed information.
- [Configuring Trusted Proxy Servers](#) If you are running applications behind one or more proxy servers, you may find it useful to configure Crowd to trust the proxies' IP addresses.

Managing your Users' Experience of Crowd

i Your users will need to access Crowd at `http://<Crowd machine name>:8095/crowd` (not `http://localhost:8095/crowd`).

6. Testing a User's Login

You may want to test a user's login to a specific application if the user has reported problems with logging in, or if you have just set up the first user to access a new application. The test verifies whether a user will be able to log in to a given application, based on the application, directory and group associations in Crowd.

Go to the application's 'Authentication Test' tab in the Crowd Administration Console, as described in the [documentation](#). The documentation also describes the possible error messages and the steps you can take to resolve any problems.

7. Changing or Resetting a User's Password

You may need to change or reset someone's password, if they have forgotten their password or if someone else has come to know the password.

i Crowd users can change or reset their own passwords too. See the [user documentation](#). To allow this, you need to grant them Crowd user rights, as described [below](#).

Go to the 'User Details' screen in the Crowd Administration Console, as described in the [documentation](#).

If you have configured an [email server](#) and a [notification template](#), Crowd will send the user an email about their new password.

8. Setting Up User Aliases

Aliases are useful if the same person has different usernames in JIRA and Confluence. You can define the user just once in Crowd, and allocate one or more aliases for the different applications that the user can access.

The [documentation](#) has the full details. In summary:

1. Make sure that aliasing is enabled for JIRA and Confluence, on the application's 'Options' screen.
2. Add the appropriate alias for each user, on the user's 'Applications' screen.

9. Granting Crowd User Rights to Someone

You can give your users access to Crowd's Self-Service Console, where they can edit their own profile, change their password and see the applications they are allowed to access. They can read the [User Guide](#) for guidance.

Make sure that the person's username is in a user directory where all users are authorized to use Crowd. Please refer to the [documentation](#) for details.

10. Granting Crowd Administrator Rights to Someone

When you first set up Crowd, you will define a single Crowd administrator. It is advisable to give other people administration rights, so that you do not run into problems when the single administrator is unavailable.

Make sure that the person is a member of the 'crowd-administrators' group. Please refer to the [documentation](#).

Important Next Steps

11. Setting Up your Applications' Host Names

When you set up your applications in step 3 above, you will have specified an IP address for each application. If JIRA, Confluence or any Crowd-connected application resides on a server that passes Crowd a host name instead of an IP address, you will need to tell Crowd the host name. Please refer to the [documentation](#).

12. Connecting to an External Database

If you decided to use the default HSQLDB database when you set up Crowd, you need to switch to a production-ready database before using Crowd as a production system. HSQLDB is provided for evaluation purposes only. Please refer to the [documentation](#).

13. Backing Up your Crowd Data

To back up your Crowd data and establish processes for regular backups, please refer to the [documentation](#).

Thank you for choosing Crowd.

We are always happy to help. Feel free to [email](#) or [call us](#) with any questions you may have.

Installation and Upgrade Guide

- [Crowd Release Notes](#)
- [Installing Crowd](#)
- [Upgrading Crowd](#)
- [Migrate to Another Database](#)
- [Migrating Crowd Between Servers](#)
- [Migrating from OnDemand to a Crowd installed site](#)
- [Installing Crowd Data Center](#)
- [Migrate from Server to Data Center](#)

Installing Crowd

Installing Crowd

You can download Crowd [here](#).

Warning: Some unzip programs cause errors

Some archive-extract programs cause errors when unzipping the Crowd archive file.

- **Linux** or **Unix** users can use any unzip program.
- **Solaris** users must use [GNU Tar](#) instead of Solaris Tar.
- **Windows** users should use a third-party unzip program like 7Zip or Winzip. If you do not have one, please download and install one before continuing:
 - [7Zip](#) Recommended. If in doubt, download the '32-bit .exe' version
 - [Winzip](#)

- [Supported Platforms](#)
- [Installing Crowd and CrowdID](#)
- [Running the Setup Wizard](#)
- [Configuring Crowd](#)

Supported Platforms

Before installing Crowd, make sure you have the right software and infrastructure to run it. If a platform and version is not listed on this page, it means we don't test it, fix bugs or provide assistance for it. All platforms are shared between Crowd Server and Crowd Data Center, unless they're clearly marked as Data Center only.

i This page is for Crowd 4.4. If you're looking for a different version, select it at the top-right.

Definitions:

✔ Supported - you can use Crowd 4.4 with this platform.

ⓘ Limited - you can evaluate Crowd 4.4 on this platform, but you can't run a production site on it.

⚠ Deprecated - you can use Crowd 4.4 with this platform, but we're planning to end support in an upcoming release.

Java

Oracle JRE / JDK

✔ Java 8

✔ Java 11

OpenJDK

✔ Java 8

✔ Java 11

Oracle JRE / JDK:

- JDK versions earlier than 8u65 might have problems connecting to LDAP servers over SSL. For more details, see [CWD-4444](#) - Secure LDAP connections are broken when using Java 1.8u51, 1.8u60, 1.7.0_85+ and 1.6.0_101+ RESOLVED
- JDK 1.8u151 might have problems with gzip compression of static resources. For more details, see [CWD-5001](#).

OpenJDK:

- Our Support and Engineering teams use AdoptOpenJDK to replicate any issues raised with OpenJDK. If you're using a different distribution of OpenJDK (e.g. Zulu), we'll still provide support for our products. However, if the bug is caused by a problem in Java distribution, we'll ask you to reach out to the Java distributor for help.

Operating systems

Operating systems

✔ Microsoft Windows

✔ Linux

ⓘ Mac OS X

Browsers

Browsers

✔ Chrome (latest stable version)

✔ Microsoft Edge (Chromium)

Good to know:

- Crowd is a pure Java application and should run on any platform provided the Java runtime platform requirements are satisfied.

✔ Mozilla Firefox(all platforms)

✔ Safari (latest stable version)

Databases

Embedded database

📌 HSQLDB

Good to know:

- Crowd ships with a built-in HSQL database, which is fine for evaluation purposes but is somewhat susceptible to data loss during system crashes. For production environments we recommend that you configure Crowd to use an [external database](#).

MySQL

✔ MySQL 8.0

✔ MySQL 5.7

Good to know:

- Please ensure that you set transaction isolation to 'read-committed' instead of the default 'repeatable-read', as described in the [database configuration guide](#).

Oracle

✔ Oracle 19c

✔ Oracle 12c R2

PostgreSQL

✔ PostgreSQL 12

✔ PostgreSQL 11

✔ PostgreSQL 10

⚠ PostgreSQL 9.6

Microsoft SQL Server

✔ SQL Server 2019

✔ SQL Server 2017

✔ SQL Server 2016

Infrastructure

JDK:

- It is not enough to have the JRE only. Please ensure that you have the full JDK. You can download the Java SE Development Kit (JDK) from the [Oracle website](#).
- Once the JDK is installed, you will need to set the **JAVA_HOME** environment variable, pointing to the root directory of the JDK. Some JDK installers set this automatically (check by typing 'echo %JAVA_HOME%' in a DOS prompt, or 'echo \$JAVA_HOME' in a shell). If it is not set, please see [Setting JAVA_HOME](#).


Hardware:

The hardware required to run Crowd depends significantly on the number of applications and users that your installation will have, as well as the maximum number of concurrent requests that the system will experience during peak hours.

During evaluation Crowd will run well on any reasonably fast workstation computer (eg. 1.5+Ghz processor). Memory requirements depend on how many applications and users you will store, but 256MB is enough for most evaluation purposes.

Most users start by downloading Crowd, and running it on their local computer. It is easy to migrate Crowd to your enterprise infrastructure later.

We would appreciate if you let us know what hardware configuration works for you. Please create a support request in [Jira](#) with your hardware specification and mention the number of applications and users in your Crowd installation.

 While some of our customers run Crowd on SPARC-based hardware, Atlassian only officially supports Crowd running on x86 hardware and 64-bit derivatives of x86 hardware.

Setting JAVA_HOME

Once you have installed the JDK (see [Supported Platforms](#)), you need to set the JAVA_HOME environment variable.

1. Right click on the **'My Computer'** icon on your desktop and select **'Properties'**.
2. Click the **'Advanced'** tab.
3. Click the **'Environment Variables'** button.
4. Click **'New'**.
5. In the **'Variable name'** field, enter 'JAVA_HOME'.
6. In the **'Variable value'** field, enter the directory (including its full path) where you installed the JDK.
7. Restart the computer.

For your current user:

1. Open up a shell / terminal window
2. `vi ~/.profile`(replace vi with your favorite text editor)
3. `Addexport JAVA_HOME=/path/to/java/home/dir` on its own line at the end of the file
4. `Addexport PATH=$JAVA_HOME/bin:$PATH` on its own line immediately after
5. Save, and restart your shell
6. Running `java -version` should give you the desired results

For all users in the system:

1. Open up a shell / terminal window
2. `vi /etc/profile`(replace vi with your favorite text editor)
3. `Addexport JAVA_HOME=/path/to/java/home/dir` on its own line at the end of the file
4. `Addexport PATH=$JAVA_HOME/bin:$PATH` on its own line immediately after
5. Save, and restart your shell
6. Running `java -version` should give you the desired results

If you are using a GUI, you may not need to open up the shell. Instead, you might be able to open the file directly in a graphical text editor.

End of support announcements for Crowd

This page contains announcements of the end of support for various platforms and browsers used with Crowd.

The table below summarizes the end of support announcements for **upcoming** Crowd releases.

Platform/Functionality	Crowd end of support
Internet Explorer 11	With Crowd 4.2 (announcement)

Why is Atlassian ending support for these platforms?

Atlassian is committed to delivering improvements and bug fixes as fast as possible. We are also committed to providing world class support for all the platforms our customers run our software on. However, as new versions of databases, web browsers, etc, are released, the cost of supporting multiple platforms grows exponentially, making it harder to provide the level of support our customers have come to expect from us. Therefore, we no longer support platform versions marked as end-of-life by the vendor, or very old versions that are no longer widely used.

Deprecated browsers for Crowd

Announced 24 September 2019

In 2015 Microsoft released Edge as the browser to supersede Internet Explorer, and in recent times [Microsoft has discouraged the use of Internet Explorer as a default browser](#). To allow us to continue to take advantage of modern web standards to deliver improved functionality and the best possible user experience across all of our products, we have decided to end support for Internet Explorer 11.

End of support means we will not fix bugs specific to Internet Explorer 11, and will begin to introduce features that aren't compatible with this browser.

When is this happening?

- The last Crowd version to support Internet Explorer 11 will be confirmed soon.
- Subsequent versions will not support Internet Explorer 11.

What this means for you


We recommend switching to one of our [supported browsers](#), such as Google Chrome, or Mozilla Firefox.

If you have questions or concerns regarding this announcement, please email eol-announcement@atlassian.com.

Installing Crowd and CrowdID

The instructions below tell you how to install the **Crowd distribution**, which includes Apache Tomcat.


Crowd versions 1.1 and later include **CrowdID**. Installing Crowd, as described below, will also install CrowdID.

 **Hint: If you are evaluating Crowd or you are unsure which version to install, just follow the simple instructions on this page.**

1. Prerequisites

- **Java.** You will need to install a Java Development Kit (JDK) on your operating system before proceeding with a Crowd installation. Please note that Crowd requires the full installation of a JDK. It is not enough to run Crowd on a Java Runtime Environment (JRE) alone. For instructions on installing the Sun JDK and setting `JAVA_HOME`, please refer to [Supported Platforms](#).

2. Install Crowd

1. [Download Crowd](#).
2. Please check your unzip program before extracting the downloaded archive see the note on the [Crowd installation front page](#).
3. Unzip the download archive into a directory of your choice. Note: Do not specify directory names that contain spaces.
 We'll refer to this installation directory as **{CROWD_INSTALL}**.
4. Specify your Crowd Home directory by editing the configuration file at: `{CROWD_INSTALL}\crowd-webapp\WEB-INF\classes\crowd-init.properties`. The **Crowd Home** directory is where Crowd will store its configuration information. If you are using the embedded HSQL database, supplied for evaluation purposes, Crowd will also store its database in this directory. (Note however that the CrowdID database will be in the installation directory, not the Home directory.) To specify the Crowd Home directory:
 - Open the `crowd-init.properties` file. This is found at `<crowd_install_directory>/crowd-webapp/WEB-INF/classes/crowd-init.properties`
 - Choose the appropriate line in the file, depending upon your operating system (see below).
 - Remove the # at the beginning of the line.
 - Enter the name of the directory you want Crowd to use as its Home directory. For example,
 - On Windows:

```
crowd.home=c:/data/crowd-home
```

Note: On Windows, make sure you use forward slashes as shown above, not backward slashes.

- On Mac and UNIX-based systems:

```
crowd.home=/var/crowd-home
```



Important

Please, ensure that the Crowd Home directory will not match the Crowd installation directory **AND** it is writable by the user executing the initialization script.

- Save the `crowd-init.properties` file.

Installing Crowd on Windows with 64-bit JVM

If you intend to run Crowd on a Windows system with a 64-bit JVM, be aware that Crowd bundles both 32 and 64 bit Tomcat binaries and uses the 32-bit binaries by default. The 32-bit binaries and their 64-bit counterparts are listed below:

32-bit	64-bit
{CROWD_INSTALL}/apache-tomcat/bin/tomcat.exe	{CROWD_INSTALL}/apache-tomcat/bin/tomcat.exe.x64
{CROWD_INSTALL}/apache-tomcat/bin/tcnative-1.dll	{CROWD_INSTALL}/apache-tomcat/bin/tcnative-1.dll.x64

In order to use the 64-bit binaries, they must be renamed to the names used by 32-bit binaries, while the 32-bit binaries must be either renamed or deleted. This can be accomplished with a simple script:

```
cd {CROWD_INSTALL}/apache-tomcat/bin
rename tomcat8.exe tomcat8.exe.x86
rename tcnative-1.dll tcnative-1.dll.x86
rename tomcat8.exe.x64 tomcat8.exe
rename tcnative-1.dll.x64 tcnative-1.dll
```

The script above adds the .x86 suffix to the 32-bit binaries and removes the .x64 suffix from the 64-bit binaries, making them usable.

3. Prepare your Database

For evaluators

This step applies to **production** installations. If you are **evaluating** Crowd and are happy to use the database supplied, you can skip this step.

If you wish to set up Crowd and/or CrowdID with an external database, see:

- [Connecting Crowd to a Database](#)
- [Connecting CrowdID to a Database](#)

4. Start Crowd and Complete the Setup Wizard

1. Run the start-up script, found in your {CROWD_INSTALL} directory:
 - start_crowd.bat for Windows.
 - start_crowd.sh for Mac and Unix-based systems.
2. Point a web browser at <http://localhost:8095/crowd> where you will see the **Crowd Setup Wizard**. Follow the instructions in the Wizard. You can also read [more information about the Setup Wizard](#).

Next Steps

- If you are running Crowd on UNIX/Linux, consider setting Crowd to [run automatically on startup and use an unprivileged system user](#).
- If you are running Crowd on Windows, consider setting Crowd to [run automatically on startup](#).

Connecting Crowd to a Database

You can configure your database connection as part of the [Crowd Setup Wizard](#). It will make things easier if you have created the database and deployed the database driver before you start.

HSQLDB database is supplied for evaluation purposes

The Crowd distribution (not EAR-WAR) is shipped with an embedded [HSQLDB](#) database. You can choose this embedded database during the Crowd setup process. The embedded database is fine for evaluation purposes, but for production installations you should connect Crowd to an enterprise database. This also lets you take advantage of existing database backup and recovery procedures.


Select the page corresponding to your database, for help on setting up an external database:

- [HSQLDB](#)
- [MS SQL Server](#)
- [MySQL](#)
- [Oracle](#)
- [PostgreSQL](#)

HSQLDB

The Crowd distribution (not EAR-WAR) is shipped with an embedded [HSQLDB](#) database. When you run the [Crowd Setup Wizard](#), you will be asked to choose a database. If you choose the embedded database, the data files will be stored in the Crowd Home directory, as configured during [installation](#).

Also see <http://hsqldb.sourceforge.net/doc/guide/ch01.html#N101C2>.

 HSQLDB should not be used as a production database. It is included for evaluation purposes only.

HSQLDB periodically must update its files to represent changes made in the database. In doing so, it must delete the current `crowddb.data` file on the file system (beneath the `/database` folder in your Crowd home directory) and replace it with a new one.

If an administrator issues a shutdown on Crowd while this update is happening, data can be lost and typically all configuration data for your Crowd server will be lost.

MS SQL Server

Supported Versions

Crowd supports the versions of MS SQL Server listed on the [Supported Platforms](#) page.

When you run the [Crowd Setup Wizard](#), you will be asked to choose a database and provide configuration settings for that database. It will make things easier if you have created the database and deployed the database driver before you start the Setup Wizard.

Follow the instructions below to set up MS SQL Server for Crowd.

1. Configure SQL Server

1. Create a database user which Crowd will connect as (e.g. **crowduser**).



In SQL Server, the database user (**crowduser** above) should not be the database owner, but should be in the `db_owner` role. Additionally, you should **create the database with case sensitive collation**.

2. Create a database for Crowd to store data in (e.g. **crowddb**).
3. Ensure that the user has permission to connect to the database, and create and populate tables
4. Ensure that the new database was set to use Read Committed with Row Versioning as its isolation level. You can apply the new isolation by executing the following query:

```
ALTER DATABASE <database name>  
SET READ_COMMITTED_SNAPSHOT ON  
WITH ROLLBACK IMMEDIATE;
```

To verify the changes, use this query which should result in '1':

```
SELECT sd.is_read_committed_snapshot_on  
FROM sys.databases AS sd  
WHERE sd.[name] = '<database name>';
```

Next Steps

Complete the Crowd installation, then start Crowd and run the Setup Wizard as described in the [Installation Guide](#).

Configuring Unicode Support in MS SQL Server

To configure Crowd to support [Unicode](#) in **MS SQL Server**, enter the following in the '**Hibernate Dialect**' field on the Crowd Setup Wizard's Database Configuration screen:
`com.atlassian.crowd.util.persistence.hibernate.SQLServerIntlDialect`

MySQL

Supported Versions

Crowd supports the versions of MySQL listed on the [Supported Platforms](#) page.

When you run the [Crowd Setup Wizard](#), you will be asked to choose a database and provide configuration settings for that database. It will make things easier if you have created the database and deployed the database driver before you start the Setup Wizard.

Follow the instructions below to set up MySQL for Crowd.

1. Configure MySQL

1. Create a database user which Crowd will connect as (e.g. **crowduser**).
2. Create a database for Crowd to store data in (e.g. **crowd**). For a UTF-8 encoded database:

```
create database crowd character set utf8 collate utf8_bin;
```

3. Ensure that the user has permission to connect to the database, and create and populate tables:

```
GRANT ALL PRIVILEGES ON crowd.* TO 'crowduser'@'localhost' IDENTIFIED BY 'crowdpass';
```

4. Edit the `my.cnf` file (often named `my.ini` on Windows operating systems) in your MySQL server. Locate the `[mysqld]` section in the file, and add or modify the following parameters (Refer to [MySQL Option Files](#) or detailed instructions on editing `my.cnf` and `my.ini`):

- Specify the default character set to be UTF-8:

```
[mysqld]
...
character-set-server=utf8
collation-server=utf8_bin
...
```

- Set the default storage engine to InnoDB:

```
[mysqld]
...
default-storage-engine=INNODB
...
```

- set to `transaction-isolation= READ-COMMITTED`.

```
[mysqld]
...
transaction-isolation = READ-COMMITTED
...
```

Notes:

- The above configuration will prevent errors when you import directory information into Crowd. See [CWD-1505](#).
5. Restart your MySQL server for the configuration change to take effect.

2. Copy the MySQL Driver to your Application Server

1. Download the [MySQL Connector/J JDBC driver](#) driver.
2. Add the MySQL JDBC driver jar (`mysql-connector-java-5.x.x-bin.jar`) to the following directory:
 - Crowd 2.0.2 or later: `{CROWD_INSTALL}/apache-tomcat/lib/`.

- Crowd 2.0.1 or earlier: {CROWD_INSTALL}/apache-tomcat/common/lib/.

**Do not place Debug Driver on CLASSPATH**

Do not place the Debug Driver (`mysql-connector-java-5.x.x-bin-g.jar`) on the CLASSPATH as this can cause issues. See ([JRA-8674](#)).

Next Steps

Complete the Crowd installation, then start Crowd and run the Setup Wizard as described in the [Installation Guide](#).

Oracle

When you run the [Crowd Setup Wizard](#), you will be asked to choose a database and provide configuration settings for that database. For smoother setup process, we recommend that you create the database and deploy the database driver before you start the Crowd Setup Wizard.

To prepare Oracle for Crowd:

1. Install the Oracle database server.

If you don't already have an operational Oracle server, [download](#) and install it now. See the [Oracle documentation](#) for instructions.

2. Create an Oracle database user.



It is recommended to create a separate database user for Crowd to use. Connecting as the SYS user is not supported.

- a. Create a database user which Crowd will connect as (e.g. crowduser):

```
create user <user> identified by <password> default tablespace <tablespace_name> quota
unlimited on <tablespace_name>;
```

- b. Grant user permission to connect to the database, create and populate tables:

```
grant connect, resource to <user>;
```

Next Steps

Once you've completed the Crowd installation, start Crowd and run the Setup Wizard as described in the [Installation Guide](#).

PostgreSQL

Supported Versions

Crowd supports the versions of PostgreSQL listed on the [Supported Platforms](#) page.

When you run the [Crowd Setup Wizard](#), you will be asked to choose a database and provide configuration settings for that database. It will make things easier if you have created the database and deployed the database driver before you start the Setup Wizard.

Follow the instructions below to set up PostgreSQL for Crowd.

1. Configure PostgreSQL

1. Create a database user which Crowd will connect as (for example, **crowduser**).
2. Create a database for Crowd to store data in (for example, **crowddb**).
3. Ensure that the user has permission to login to the database and can create database objects.

Next Steps

Complete the Crowd installation, then start Crowd and run the Setup Wizard as described in the [Installation Guide](#).

Connecting CrowdID to a Database

CrowdID is a free add-on that ships with Crowd versions 1.1 and later.

By default, CrowdID in the Crowd distribution (not EAR-WAR) is shipped preconfigured with [HSQL](#). This is fine for evaluation purposes, but for production installations, you should connect CrowdID to an enterprise database. This also lets you take advantage of existing database backup and recovery procedures.

CrowdID database connection is not yet part of Setup Wizard

This page describes the procedure for connecting **CrowdID** to an external database. You'll notice that the procedure for connecting **Crowd itself** to a database is simpler, because the Crowd database connection is configured by the [Crowd Setup Wizard](#). The CrowdID database configuration cannot be done as part of the Setup Wizard. We hope to improve the CrowdID integration soon. In the meantime, please follow the steps below.

The following instructions will allow you to configure CrowdID to an external database:

- [HSQLDB for CrowdID](#)
- [MS SQL Server for CrowdID](#)
- [MySQL for CrowdID](#)
- [Oracle for CrowdID](#)
- [PostgreSQL for CrowdID](#)

Database Overview


CrowdID in the Crowd distribution (not EAR-WAR) includes the Apache Tomcat application server and an in-memory HSQL database engine. This JNDI reference (`CrowdIDDS`) can be adjusted to use your custom database and driver by editing the `crowd.xml` deployment description.

HSQLDB for CrowdID

The default version of CrowdID uses an embedded HSQLDB database.

HSQLDB periodically must update its files to represent changes made in the database. To do so, it must delete the current `crowddb.data` file on the filesystem, located in the `/database` folder, and replace it with a new one.

If an administrator issues a shutdown on CrowdID in this period, data can be lost, and typically all configuration data for your CrowdID server will be lost.


 HSQLDB should not be used as a production database. It is included for evaluation purposes only.


MS SQL Server for CrowdID

Follow the steps below to connect CrowdID to MS SQL Server.

1. Configure SQL Server


1. Create a database user which CrowdID will connect as (e.g. **crowduser**).

 In SQL Server, the database user (**crowduser** above) should not be the database owner, but should be in the `db_owner` role.

2. Create a database for CrowdID to store data in (e.g. **crowdiddb**).  This must be a different database to the one used by Crowd.
3. Ensure that the user has permission to connect to the database, and create and populate tables.

2. Copy the SQL Server Driver to your Application Server

1. Download the SQL Server JDBC driver from [JTDS](#) (recommended, assumed below), or [I-net software](#)(commercial).

 Microsoft have their own JDBC driver but we strongly recommend avoiding it after our Jira customers have reported various connection errors ([JIRA-5760](#), [JIRA-6872](#)[|http://jira.atlassian.com/browse/JIRA-6872](http://jira.atlassian.com/browse/JIRA-6872)), workflow problems ([JIRA-8443](#)) and Chinese character problems ([JIRA-5054](#)).

2. Add the SQL Server JDBC driver JAR (`jtds-[version].jar`) to the following directory:
 - Crowd 2.0.2 or later: `{CROWD_INSTALL}/apache-tomcat/lib/`.
 - Crowd 2.0.1 or earlier: `{CROWD_INSTALL}/apache-tomcat/common/lib/`.

3. Configure your Application Server to Connect to SQL Server

1. Edit the `conf/Catalina/localhost/openidserver.xml` file and customize the **username**, **password**, **driverClassName** and **urlparameters** for the Datasource.

```
<Context path="/openidserver" docBase="../../crowd-openidserver-webapp" debug="0">

<Resource name="jdbc/CrowdIDDS" auth="Container" type="javax.sql.DataSource"
username="[enter db username here]"
password="[enter db password here]"
driverClassName="net.sourceforge.jtds.jdbc.Driver"
url="jdbc:jtds:sqlserver://localhost:1433/crowdiddb"
[ delete the minEvictableIdleTimeMillis, timeBetweenEvictionRunsMillis and maxActive params here ]
/>

<Manager className="org.apache.catalina.session.PersistentManager" saveOnRestart="false"/>

</Context>
```

2. Delete the **minEvictableIdleTimeMillis**, **timeBetweenEvictionRunsMillis** and **maxActive** attributes (which are only needed for HSQL, and degrade performance otherwise).

4. Configure CrowdID to use MS SQL Server

1. Edit the `build.properties` file (located in the root of the Crowd distribution) and modify the **hibernate.dialect** to the following:

```
hibernate.dialect=org.hibernate.dialect.SQLServerDialect
```

2. Then run `./build.sh` or `build.bat`. This will configure CrowdID to use the MS SQL Server dialect.

If you do not wish to edit this file and run the build script, you can edit the `jdbc.properties` file (which the above script modifies) directly. The `jdbc.properties` file is located here: `crowd-openidserver-webapp\WEB-INF\classes\jdbc.properties`. Modify the file to the following:

```
# - Crowd Configuration Options

hibernate.connection.datasource=java\:comp/env/jdbc/CrowdIDDS
hibernate.dialect=org.hibernate.dialect.SQLServerDialect
hibernate.transaction.factory_class=org.hibernate.transaction.JDBCTransactionFactory

...
```


Next Steps

You should now have an application server configured to connect to a database, and CrowdID configured to use the correct database. Now start up CrowdID and watch the logs for any errors.

MySQL for CrowdID

Follow the steps below to connect CrowdID to MySQL.

1. Configure MySQL

1. Create a database user which CrowdID will connect as (e.g. **crowduser**).
2. Create a database for CrowdID to store data in (e.g. **crowdidb**).
 This must be a different database from the one used by Crowd.
For a UTF-8 encoded database:

```
create database crowdiddb character set utf8;
```

3. Ensure that the user has permission to connect to the database, and create and populate tables.

2. Copy the MySQL Driver to your Application Server

1. Download the latest [MySQL Connector/J JDBC driver](#).
2. Add the MySQL JDBC driver jar (`mysql-connector-java-3.x.x-bin.jar`) to the following directory:
 - Crowd 2.0.2 or later: `{CROWD_INSTALL}/apache-tomcat/lib/`.
 - Crowd 2.0.1 or earlier: `{CROWD_INSTALL}/apache-tomcat/common/lib/`.

 Do not place the Debug Driver (`mysql-connector-java-3.x.x-bin-g.jar`) on the CLASSPATH as this can cause issues. ([JRA-8674](#)).

3. Configure your Application Server to Connect to MySQL

1. Edit the file `apache-tomcat-X.X.XX/conf/Catalina/localhost/openidserver.xml` and customize the **username**, **password**, **driverClassName** and **urlparameters** for the Datasource.


```
<Context path="/openidserver" docBase="../../crowd-openidserver-webapp" debug="0">

<Resource name="jdbc/CrowdIDDS" auth="Container" type="javax.sql.DataSource"
username="[enter db username here]"
password="[enter db password here]"
driverClassName="com.mysql.jdbc.Driver"
url="jdbc:mysql://localhost/crowdidb?autoReconnect=true&useUnicode=true&
characterEncoding=utf8"
[ delete the minEvictableIdleTimeMillis, timeBetweenEvictionRunsMillis and maxActive params here ]
/>

<Manager className="org.apache.catalina.session.PersistentManager" saveOnRestart="false"/>

</Context>
```

The URL above assumes a UTF-8 database i.e. created with `create database crowdiddb character set utf8;`.

 MySQL closes idle connections after 8 hours, so the `autoReconnect=true` is necessary to tell the driver to reconnect.

2. Delete the **minEvictableIdleTimeMillis**, **timeBetweenEvictionRunsMillis** and **maxActive** attributes (which are only needed for HSQL, and degrade performance otherwise).

4. Configure CrowdID to use MySQL

1. Edit the `build.properties` file (located in the root of the Crowd distribution) and modify the **hibernate.dialect** to the following.

```
hibernate.dialect=org.hibernate.dialect.MySQL5InnoDBDialect
```

2. Then run `./build.sh` or `build.bat`. This will configure CrowdID to use the MySQL dialect.

If you do not wish to edit this file and run the build script, you can edit the **jdbc.properties** (which the above script modifies) directly. The **jdbc.properties** file is located here: `crowd-openidserver-webapp\WEB-INF\classes\jdbc.properties`. Modify the file to the following:

```
# - Crowd Configuration Options

hibernate.connection.datasource=java\:comp/env/jdbc/CrowdIDDS
hibernate.dialect=org.hibernate.dialect.MySQL5InnoDBDialect
hibernate.transaction.factory_class=org.hibernate.transaction.JDBCTransactionFactory

...

```

Next steps

You should now have an application server configured to connect to a database, and CrowdID configured to use the correct database. Now start up CrowdID and watch the logs for any errors.

RELATED TOPICS

- [Supported Platforms](#)
 - [Setting JAVA_HOME](#)
 - [End of support announcements for Crowd](#)
- [Installing Crowd and CrowdID](#)
 - [Connecting Crowd to a Database](#)
 - [HSQLDB](#)
 - [MS SQL Server](#)
 - [MySQL](#)
 - [Oracle](#)
 - [PostgreSQL](#)
 - [Connecting CrowdID to a Database](#)
 - [HSQLDB for CrowdID](#)
 - [MS SQL Server for CrowdID](#)
 - [MySQL for CrowdID](#)
 - [Oracle for CrowdID](#)
 - [PostgreSQL for CrowdID](#)
 - [Specifying your Crowd Home Directory](#)
- [Running the Setup Wizard](#)
 - [Troubleshooting your Configuration on Setup](#)
- [Configuring Crowd](#)
 - [Important directories and files](#)
 - [DRAFT - .Important Directories and Files vCROWD_3.0](#)
 - [The crowd.properties file](#)
 - [Changing the Port that Crowd uses](#)
 - [Configuring Crowd to Work with SSL](#)
 - [Installing Crowd as a Windows Service](#)
 - [Specifying Startup Order of Windows Services](#)
 - [Changing the User for the Crowd Windows Service](#)
 - [Removing the Crowd Windows Service](#)
 - [Troubleshooting Crowd as a Windows Service](#)
 - [Setting Crowd to Start Automatically on Mac OS X](#)
 - [Setting Crowd to Run Automatically and Use an Unprivileged System User on UNIX](#)

Oracle for CrowdID

Follow the steps below to connect CrowdID to Oracle.

1. Configure Oracle

1. Create a database user which CrowdID will connect as (e.g. **crowduser**).
2. Create a database for CrowdID to store data in (e.g. **crowdiddb**). ⚠️ This must be a different database to the one used by Crowd.
3. Ensure that the user has permission to connect to the database, and create and populate tables.

2. Copy the Oracle Driver to your Application Server

1. Download the Oracle JDBC driver from http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html.
2. Add the Oracle JDBC driver jar to the following directory:
 - Crowd 2.0.2 or later: {CROWD_INSTALL}/apache-tomcat/lib/.
 - Crowd 2.0.1 or earlier: {CROWD_INSTALL}/apache-tomcat/common/lib/.

3. Configure your Application Server to Connect to Oracle

1. Edit the file `apache-tomcat-X.X.XX/conf/Catalina/localhost/openidserver.xml` and customize the **username**, **password**, **driverClassName** and **urlparameters** for the Datasource.

```
<Context path="/openidserver" docBase="../../crowd-openidserver-webapp" debug="0">

<Resource name="jdbc/CrowdIDDS" auth="Container" type="javax.sql.DataSource"
username="[enter db username here]"
password="[enter db password here]"
driverClassName="oracle.jdbc.driver.OracleDriver"
url="jdbc:oracle:thin:@localhost:1521:crowdiddb"
[ delete the minEvictableIdleTimeMillis, timeBetweenEvictionRunsMillis and maxActive params here ]
/>

<Manager className="org.apache.catalina.session.PersistentManager" saveOnRestart="false"/>

</Context>
```

2. Delete the **minEvictableIdleTimeMillis**, **timeBetweenEvictionRunsMillis** and **maxActive** attributes (which are only needed for HSQL, and degrade performance otherwise).

4. Configure CrowdID to use Oracle

1. Edit the `build.properties` file (located in the root of the Crowd distribution) and modify the **hibernate.dialect** to the following

```
hibernate.dialect=org.hibernate.dialect.Oracle10gDialect
```

2. Then run `./build.sh` or `build.bat`. This will configure CrowdID to use the Oracle dialect. ⚠️ There is a problem with `build.bat` in Crowd version 1.2.0. To fix the problem, please apply the patch described in [CWD-638](#).

If you do not wish to edit this file and run the build script, you can edit the **jdbc.properties** (which the above script modifies) directly. The **jdbc.properties** file is located here: `crowd-openidserver-webapp\WEB-INF\classes\jdbc.properties`. Modify the file to the following:

```
# - Crowd Configuration Options

hibernate.connection.datasource=java\:comp/env/jdbc/CrowdIDDS
hibernate.dialect=org.hibernate.dialect.OracleDialect
hibernate.transaction.factory_class=org.hibernate.transaction.JDBCTransactionFactory

...
```

Next Steps

You should now have an application server configured to connect to a database, and CrowdID configured to use the correct database. Now start up CrowdID and watch the logs for any errors.

RELATED TOPICS

- [Supported Platforms](#)
 - [Setting JAVA_HOME](#)
 - [End of support announcements for Crowd](#)
- [Installing Crowd and CrowdID](#)
 - [Connecting Crowd to a Database](#)
 - [HSQLDB](#)
 - [MS SQL Server](#)
 - [MySQL](#)
 - [Oracle](#)
 - [PostgreSQL](#)
 - [Connecting CrowdID to a Database](#)
 - [HSQLDB for CrowdID](#)
 - [MS SQL Server for CrowdID](#)
 - [MySQL for CrowdID](#)
 - [Oracle for CrowdID](#)
 - [PostgreSQL for CrowdID](#)
 - [Specifying your Crowd Home Directory](#)
- [Running the Setup Wizard](#)
 - [Troubleshooting your Configuration on Setup](#)
- [Configuring Crowd](#)
 - [Important directories and files](#)
 - [DRAFT - .Important Directories and Files vCROWD_3.0](#)
 - [The crowd.properties file](#)
 - [Changing the Port that Crowd uses](#)
 - [Configuring Crowd to Work with SSL](#)
 - [Installing Crowd as a Windows Service](#)
 - [Specifying Startup Order of Windows Services](#)
 - [Changing the User for the Crowd Windows Service](#)
 - [Removing the Crowd Windows Service](#)
 - [Troubleshooting Crowd as a Windows Service](#)
 - [Setting Crowd to Start Automatically on Mac OS X](#)
 - [Setting Crowd to Run Automatically and Use an Unprivileged System User on UNIX](#)

PostgreSQL for CrowdID

Follow the steps below to connect CrowdID to PostgreSQL.

1. Configure PostgreSQL

1. Create a database user which CrowdID will connect as (for example, **crowduser**).
2. Create a database for CrowdID to store data in (for example, **crowdiddb**). ⚠ This must be a different database to the one used by Crowd.
3. Ensure that the user has permission to connect to the database and to create and populate tables.

3. Configure your Application Server to Connect to PostgreSQL

1. Edit the file `apache-tomcat-X.X.XX/conf/Catalina/localhost/openidserver.xml` and customize the **username**, **password**, **driverClassName** and **urlparameters** for the datasource.

```
<Context path="/openidserver" docBase="../../crowd-openidserver-webapp" debug="0">

<Resource name="jdbc/CrowdIDDS" auth="Container" type="javax.sql.DataSource"
username="[enter db username here]"
password="[enter db password here]"
driverClassName="org.postgresql.Driver"
url="jdbc:postgresql://host:port/crowdidb" [ see also http://jdbc.postgresql.org/doc.html ]"
[ delete the minEvictableIdleTimeMillis, timeBetweenEvictionRunsMillis and maxActive params here ]
/>

<Manager className="org.apache.catalina.session.PersistentManager" saveOnRestart="false"/>

</Context>
```

2. Delete the **minEvictableIdleTimeMillis**, **timeBetweenEvictionRunsMillis** and **maxActive** attributes. (These are only needed for HSQL database, and degrade performance otherwise.)

4. Configure CrowdID to use PostgreSQL

1. Edit the `build.properties` file located in the root of the Crowd distribution, and modify the **hibernate.dialect** to the following

```
hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect
```

2. Run `./build.sh` or `build.bat`. This will configure Crowd to use the PostgreSQL dialect.

If you do not wish to edit this file and run the build script, you can edit the **jdbc.properties** (which the above script modifies) directly. The **jdbc.properties** file is located here: `crowd-openidserver-webapp\WEB-INF\classes\jdbc.properties`. Modify the file to the following:

```
# - Crowd Configuration Options

hibernate.connection.datasource=java\:comp/env/jdbc/CrowdIDDS
hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect
hibernate.transaction.factory_class=org.hibernate.transaction.JDBCTransactionFactory

...
```

Next Steps

You should now have an application server configured to connect to a database, and CrowdID configured to use the correct database. Start up CrowdID and watch the logs for any errors.

RELATED TOPICS

- [Supported Platforms](#)
 - [Setting JAVA_HOME](#)

- End of support announcements for Crowd
- Installing Crowd and CrowdID
 - Connecting Crowd to a Database
 - HSQLDB
 - MS SQL Server
 - MySQL
 - Oracle
 - PostgreSQL
 - Connecting CrowdID to a Database
 - HSQLDB for CrowdID
 - MS SQL Server for CrowdID
 - MySQL for CrowdID
 - Oracle for CrowdID
 - PostgreSQL for CrowdID
 - Specifying your Crowd Home Directory
- Running the Setup Wizard
 - Troubleshooting your Configuration on Setup
- Configuring Crowd
 - Important directories and files
 - DRAFT - .Important Directories and Files vCROWD_3.0
 - The crowd.properties file
 - Changing the Port that Crowd uses
 - Configuring Crowd to Work with SSL
 - Installing Crowd as a Windows Service
 - Specifying Startup Order of Windows Services
 - Changing the User for the Crowd Windows Service
 - Removing the Crowd Windows Service
 - Troubleshooting Crowd as a Windows Service
 - Setting Crowd to Start Automatically on Mac OS X
 - Setting Crowd to Run Automatically and Use an Unprivileged System User on UNIX

Specifying your Crowd Home Directory

The **Crowd Home** directory is where Crowd will store its configuration information. If you are using the embedded HSQL database, supplied for evaluation purposes, Crowd will also store its database in this directory. (Note however that the CrowdID database will be in the installation directory, not the Home directory.) To specify the Crowd Home directory:

- Open the `crowd-init.properties` file. This is found at `<crowd_install_directory>/crowd-webapp/WEB-INF/classes/crowd-init.properties`
- Choose the appropriate line in the file, depending upon your operating system (see below).
- Remove the # at the beginning of the line.
- Enter the name of the directory you want Crowd to use as its Home directory. For example,
 - On Windows:

```
crowd.home=c:/data/crowd-home
```

Note: On Windows, make sure you use forward slashes as shown above, not backward slashes.

- On Mac and UNIX-based systems:

```
crowd.home=/var/crowd-home
```



Important

Please, ensure that the Crowd Home directory will not match the Crowd installation directory **AND** it is writable by the user executing the initialization script.

- Save the `crowd-init.properties` file.

Advanced Usage

It is also possible to define the `crowd.home` property as a Java system or [Servlet Context](#) parameter.

Java System Parameter

Use the following format for your Java parameter:

```
-Dcrowd.home=/var/crowd-home
```

Where should you put this value?

You could add it to the `setenv.sh` or `setenv.bat` file supplied with the Crowd distribution (not Crowd EAR-WAR).

Servlet Context Parameter

The following configuration XML can be added to the `crowd-standalone-install/apache-tomcat/conf/Catalina/localhost/crowd.xml` context file to set the `crowd.home` property:

```
<Parameter name="crowd.home" value="/var/crowd-home" override="false"/>
```

Running the Setup Wizard

Before running the Setup Wizard described below, please follow the instructions on [installing Crowd](#).

When you access the Crowd Administration Console for the first time, you will see the **Crowd Setup Wizard**. This is a series of screens which will prompt you to configure your database connection and to supply some default values (which you can change later if necessary).

On this page:

- [Step 1. Starting the Setup Wizard](#)
- [Step 2. Licensing](#)
- [Step 3. Installation Type](#)
- [Step 4. Database Configuration](#)
- [Step 5. \(Optional\) Import Existing Crowd Data](#)
- [Step 6. Options](#)
- [Step 7. Default Directory](#)
- [Step 8. Default Administrator](#)
- [Step 9. Integrated Applications](#)
- [Step 10. Setup Complete](#)



Do you need to restart the Setup Wizard from the beginning?

Read this [hint in the Crowd Knowledge Base](#).

Step 1. Starting the Setup Wizard

Go to the following URL in your web browser: `http://localhost:8095/crowd` or `http://localhost:8095/crowd/console`.

- If there are no errors, you should see the **'License'** screen described [below](#).
- If there is an error in your configuration, you will see the **'Crowd Checklist'** screen. Read more about [troubleshooting your installation](#).

Step 2. Licensing

License

It appears this is the first time that you have run Crowd. This setup wizard will take you through your initial configuration:

Server ID: A6QZ-A6QZ-A6QZ-A6QZ

License: *

An evaluation license key is available from the [Atlassian website](#).

Crowd licenses are based on the number of end-users who will log in to the applications that are integrated with Crowd.

You can obtain an evaluation license from the [Atlassian](#) website. When you obtain an evaluation license or purchase, renew or upgrade your license you will receive a license key via email or on the Atlassian website.

Type or paste your license key into the '**License**' field, shown on the screenshot above.

Step 3. Installation Type

Crowd Installation

Please select type of installation you would like to perform.

New Installation
Setup a fresh installation of Crowd.

Import data from an XML Backup
Import data using an XML export from an existing Crowd installation.

In this step, you will choose whether to set up a new Crowd database or restore an existing database. Choose an option as follows:

- '**New Installation**' Set up a new Crowd database.
 - ✔ Hint: Choose this option if you are evaluating Crowd.
- '**Import data from an XML Backup**' Import your Crowd data from an XML backup file, which has been exported from your existing Crowd installation.

Step 4. Database Configuration

The '**Database Configuration**' screen allows you to choose the type of database connection, as described below.

✔ If in any doubt, choose the default **'Embedded'** option for evaluation purposes.

ℹ When you click 'Continue' after choosing your database options, there may be a short wait while Crowd writes the information to the database tables. Please be patient.

Database Option 1: Embedded HSQLDB (For Evaluation Purposes Only)

Database Configuration

Select the type of database you would like to use with Crowd.

Embedded
The embedded database will allow Crowd to operate without an external database. This is useful when evaluating Crowd and **not recommended** for production systems.

JDBC Connection
Connect to an external database using a JDBC connection.

JNDI Datasource
Connect to an external database through a datasource managed by the application server.

The Crowd distribution (not EAR-WAR) is shipped with an embedded [HSQLDB](#) database. If you choose the '**Embedded**' option, the data files are stored in the Crowd Home directory, as configured on [installation](#).

The HSQLDB database is fine for evaluation purposes, but for production installations you should connect Crowd to an enterprise database using the JDBC or JNDI datasource connections described below. This also lets you take advantage of your existing database backup and recovery procedures.

Database Option 2: JDBC Connection

Database Configuration

Select the type of database you would like to use with Crowd.

Embedded
 The embedded database will allow Crowd to operate without an external database. This is useful when evaluating Crowd and **not recommended** for production systems.

JDBC Connection
 Connect to an external database using a JDBC connection.

Database: * ▼
Select a database preconfiguration.

Driver Class Name: *
The class name of the database driver. Ensure that this class is in your application servers class path.

JDBC URL: *
The JDBC URL to access the database.

Username: *
The username to access the database.

Password:
The password to access the database.

Hibernate Dialect: *
Only modify the Hibernate dialect if you require a variant dialect for your database type.

Overwrite Existing Data:
Overwrite any existing data in the database for a clean installation of Crowd.

JNDI Datasource
 Connect to an external database through a datasource managed by the application server.

Select the '**JDBC Connection**' if you want to connect to an external database via a JDBC connection. (If you have not yet created your database for Crowd, follow the [database setup instructions](#).)

Supply the details for your database:

Field	Description
Database	Select your database server type.
Driver Class Name	Enter the class name for your database driver. Make sure that the class is in the class path on your application server. See guidelines on creating your specific database .
JDBC URL	Enter the URL at which Crowd can access the database JDBC connection. Do note that if you wish to change the default database name, you can specify it in the URL as well. E.g. for SQL Server, the default URL is <code>jdbc:jtds:sqlserver://localhost:1433/crowd</code> , and if you wish to connect to crowddb instead, modify the URL so that it looks like this: <code>jdbc:jtds:sqlserver://localhost:1433/crowddb</code>
Username	Enter the username which Crowd will use to access the database.

Pass word	Enter the password corresponding to the above username.
Hibernate Dialect	<p>This is the Hibernate configuration for the selected database type. The Crowd installation will supply a default dialect for the database type you have chosen. You should only alter this dialect if you need an alternative for the database type or are using an unsupported database type.</p> <ul style="list-style-type: none"> To configure Crowd to support Unicode in MS SQL Server, enter the following in the 'Hibernate Dialect' field on the Crowd Setup Wizard's Database Configuration screen: <code>com.atlassian.crowd.util.persistence.hibernate.SQLServerIntlDialect</code>
Over write Existing Data	<p>Crowd will ask you to confirm that existing data should be overwritten, if both of the following are true:</p> <ul style="list-style-type: none"> You chose 'New Installation' or 'Import data from an XML Backup' in <i>Step 3 above</i>, and The database configured on the above screen already exists and contains Crowd data.

Database Option 3: JNDI Datasource

Database Configuration

Select the type of database you would like to use with Crowd.

Embedded
The embedded database will allow Crowd to operate without an external database. This is useful when evaluating Crowd and not recommended for production systems.

JDBC Connection
Connect to an external database using a JDBC connection.

JNDI Datasource
Connect to an external database through a datasource managed by the application server.

Database: * ▼
Select a database preconfiguration.

JNDI Name: *
If java:comp/env/jdbc/DataSourceName doesnt work, try jdbc/DataSourceName (or vice versa).

Hibernate Dialect: *
Only modify the Hibernate dialect if you require a variant dialect for your database type.

Overwrite Existing Data:
Overwrite any existing data in the database for a clean installation of Crowd.

Select the '**JNDI Datasource**' if you want to connect to an external database via a datasource managed by your application server.

Supply the details for your database:

Field	Description
Database	Select your database server type.
JNDI Name	Enter the datasource name, e.g. jdbc/CrowdDS or java:comp/env/jdbc/CrowdDS.

Hibernate Dialect	<p>This is the Hibernate configuration for the selected database type. The Crowd installation will supply a default dialect for the database type you have chosen. You should only alter this dialect if you need an alternative for the database type or you have selected an unsupported database type.</p> <ul style="list-style-type: none"> To configure Crowd to support Unicode in MS SQL Server, enter the following in the 'Hibernate Dialect' field on the Crowd Setup Wizard's Database Configuration screen: <code>com.atlassian.crowd.util.persistence.hibernate.SQLServerIntlDialect</code>
Overwrite Existing Data	<p>Crowd will prompt you to confirm that existing data should be overwritten, if both of the following are true:</p> <ul style="list-style-type: none"> You chose 'New Installation' or 'Import data from an XML Backup' in Step 3 above, and The database configured on the above screen already exists and contains Crowd data.

Step 5. (Optional) Import Existing Crowd Data

Import Existing Crowd Data

Enter the Crowd XML backup file to upgrade from.

File Location:

The full file path to your existing data (e.g. C:\crowd\data.xml)

This screen will appear only if you selected '**Import data from an XML Backup**' in [Step 3 above](#).

In '**File Location**', enter the full path to your XML backup file including the name of the XML file.

Upgrading from an existing Crowd installation?

If you have connected to an existing database or imported your data from XML, the setup will be complete once you have clicked 'Continue' on the above screen. See [Step 11 below](#) and read more about [upgrading Crowd](#).

Step 6. Options

Options

Deployment Title: *
The name of this Crowd instance.

Session Timeout: *
The number of minutes a session lasts before expiring. Must be greater than 0.

Base URL: *
The base URL for this installation of Crowd.

This part of the setup process allows you to specify general options for the Crowd server.

- The deployment title is a unique name for your Crowd instance. The deployment title is used by default in the subject line of [email notifications](#).
You can change this value later, via the [Crowd Administration Console](#).
- The session timeout determines how long a session will be considered valid during any period of inactivity. This value is specified in minutes and must be greater than 0.
You can change this value later, via the [Crowd Administration Console](#).
- The base URL is the website address of the Crowd server. You can change this value later, via the [Crowd Administration Console](#).

Step 7. Default Directory

Internal Directory

Name: *
A short, recognisable name that characterises this user directory. For example: "Chicago Employees" or "Web Customers".

Description:
More information about this directory.

Password Regex:
Regular expression pattern which new passwords will be validated against. Leave blank to disable this feature.

Maximum Invalid Password Attempts:
The maximum number of invalid password attempts before the authenticating account will be disabled. Enter 0 to disable this feature.

Maximum Unchanged Password Days:
The number of days until the password must be changed. Enter 0 to disable password expiry.

Password History Count:
The number of previous passwords to check when disallowing repeated passwords on password change. Enter 0 to allow password repeats.

Password Encryption: *
For compatibility between Atlassian products you must use ATLASSIAN-SHA1.

Please configure a default user directory. For information about configuring different types of directories (Internal, LDAP, Delegated Authentication or Custom) refer to [Adding a Directory](#).

i Crowd administrators group is in default directory

The default group `crowd-administrators` will be automatically created in the default directory. Members of this group have rights to [administer Crowd](#).

Step 8. Default Administrator

Default Administrator

To configure the security server, a default administrator needs to be created. Additional administrators may be added later.

Email: *
Email address in standard format (RFC2822).

Username: *
Enter administrator user name.

Password: *

Confirm Password: *

First Name: *

Last Name: *

Please specify a default Crowd administrator. The default administrator will be automatically added to the default group `crowd-administrators`, thereby giving them rights to access the Crowd Administration Console.

Step 9. Integrated Applications

Integrated Applications

i The integrated applications use a default password when communicating with the Crowd server. When deploying to a production environment, it is critical to change the integrated applications default passwords.

Would you like to configure the integrated applications?

OpenID Server: True False
The Crowd OpenID Server will allow you to authenticate using your standard Crowd logins with OpenID enabled websites.

Demo Application: True False
The demo web application highlights best practices when using the Crowd framework. The Crowd download archive contains the entire source to the demo application, which can be used as an example when integrating your web applications

You have the option to auto-configure two applications.

- **OpenID Server** This is the **CrowdID** application, which allows you to provide [OpenID](#) services for your end-users. For details please see the [CrowdID Administration Guide](#) and the [CrowdID User Guide](#).
- **Demo Application** The 'demo' application is an example of an [application integrated with Crowd](#). It highlights best practices for using the Crowd framework, and is provided to assist you with quickly setting up and configuring Crowd. The Crowd download zip file (archive) contains the entire source for the 'demo' application, which you can use as an example when [integrating your custom web applications](#).

Step 10. Setup Complete

You are now ready to log in with the default administrator account you have just created, and use the [Crowd Administration Console](#). For details, please see the [Administration Guide](#).

RELATED TOPICS


- [Supported Platforms](#)
- [Installing Crowd and CrowdID](#)
- [Running the Setup Wizard](#)
- [Configuring Crowd](#)

Troubleshooting your Configuration on Setup

This page describes the 'Crowd Checklist' screen and tells you how to use the screen to troubleshoot your initial Crowd configuration. The 'Crowd Checklist' screen may appear when you start the [Setup Wizard](#) after [installing Crowd](#).




i The 'Crowd Checklist' appears only if there is an error in your environment configuration, preventing you from completing the Setup Wizard.

Troubleshooting your Configuration Problems

The 'Crowd Checklist' shows a list of environmental requirements on the left and a 'Status' for each setting on the right. A red exclamation mark () in the 'Status' column indicates a problem with one of the settings.

Environmental Requirement	Possible Error Message	Solution
Java Development Kit 1.5 or higher	<i>(The screen will show the version of JDK detected in your system, with a red exclamation mark in the 'Status' column if insufficient.)</i>	Refer to the System Requirements page for information about the JDK required and where you can get it.
Servlet 2.3 API or higher	<i>(The screen will show the application server and version detected in your system, with a red exclamation mark in the 'Status' column if insufficient.)</i>	Make sure that the servlet container on your application server supports the Servlet 2.3 specification . Note: Crowd ships with Apache Tomcat (5.5.x) which is compliant.
Crowd Home directory	Invalid home directory specified in {CROWD-INSTALL}/crowd-webapp/WEB-INF/classes/crowd-init.properties. Please edit this file and set the crowd.home value to a directory of your choice. Crowd will use this directory to store its configuration files.	Define the directory which you want Crowd to use as its 'home'. Read all about it in the installation guide .

Screenshot: 'Crowd Checklist'

Crowd Checklist	
<p>Welcome to Crowd.</p> <p>Your environment is not configured correctly. Please fix the problems below and restart Crowd. For more information please consult the Crowd installation documentation.</p>	
	Status
<p>Java Development Kit 1.5 or higher</p> <p><i>Found: Sun Microsystems Inc. - 1.6.0_04</i></p>	
<p>Servlet 2.3 API or higher</p> <p><i>Found: Apache Tomcat/5.5.25</i></p>	
<p>Crowd Home directory</p> <p>Invalid home directory specified in: <code>/C:/Atlassian/atlassian-crowd-1.3-SNAPSHOT/apache-tomcat/webapps/././crowd-webapp/WEB-INF/classes/crowd-init.properties</code>.</p> <p>Please edit this file and set the crowd.home value to a directory of your choice. Crowd will use this directory to store its configuration files.</p> <p>Upgrading from another 1.3.x instance? Set your crowd.home value to point to the old crowd.home directory.</p>	

The above screenshot shows a problem with the setting of the Crowd home directory.

RELATED TOPICS

- [Supported Platforms](#)
- [Installing Crowd and CrowdID](#)
- [Running the Setup Wizard](#)
- [Configuring Crowd](#)

Configuring Crowd

You can configure Crowd to suit your environment, as described on the following pages:

- [Important directories and files](#)
- [Changing the Port that Crowd uses](#)
- [Configuring Crowd to Work with SSL](#)
- [Installing Crowd as a Windows Service](#)
- [Setting Crowd to Start Automatically on Mac OS X](#)
- [Setting Crowd to Run Automatically and Use an Unprivileged System User on UNIX](#)

RELATED TOPICS

- [Specifying your Crowd Home Directory](#)
- [Configuring an SSL Certificate for Microsoft Active Directory](#)
- [Troubleshooting your Configuration on Setup](#)

- [Supported Platforms](#)
- [Installing Crowd and CrowdID](#)
- [Running the Setup Wizard](#)
- [Configuring Crowd](#)

Important directories and files

This page contains information about the important directories and files to be aware of when configuring Crowd.

On this page:

- [Home directory](#)
- [Shared directory](#)
 - [crowd.cfg.xmlfile](#)
 - [plugins directory](#)
 - [backupsdirectory](#)
- [Installation directory](#)

Home directory

The Crowd home directory is where Crowd stores its configuration information. If you're using the embedded HSQL database, supplied for evaluation purposes, Crowd will also store its database in this directory (note however that the CrowdID database will be in the Crowd installation directory, not the home directory.)

The location of this directory is specified in the `crowd-init.properties` file described [below](#). You can set the location during [installation](#).

You can check the location of your Crowd home directory on the [System Information](#) screen.

Important files and directories in the Crowd home directory:

- [bundled-plugins directory](#)
- [caches directory](#)
- [database directory](#)
- [plugin-data directory](#)

bundled-plugins directory

The `bundled-plugins` directory is a sub-directory of your Crowd home directory. It contains plugins which are shipped with your Crowd installation, such as:

- The SAML integration plugin which provides the [Google Apps SSO feature](#).
- The [Shared Access Layer \(SAL\)](#) plugins.
- The [REST module](#) plugin.
- And more.

The plugins are a collection of jars generated when you install the Crowd web application. The jars are obtained by unzipping `atlassian-bundled-plugins.zip` from `{CROWD_INSTALL}\crowd-webapp\WEB-INF\classes`.

caches directory

The `caches` directory is a sub-directory of your Crowd home directory. It contains various files that Crowd caches to improve performance. The files in sub-directories of this directory are either created or updated when you install or restart the Crowd web application.

Do not modify or remove these files while Crowd is running. It should be safe for you to delete these files between application restarts.

It may improve Crowd's performance if you link this sub-directory to a fast disk.

database directory

If you are using the embedded HSQL database, supplied for evaluation purposes, Crowd will store its database in this directory (note however that the CrowdID database will be in the Crowd Installation directory, not the Crowd home directory.)

plugin-data directory

The `plugin-data` directory is a sub-directory of your Crowd home directory. Plugins developed for Crowd 2.12 and older will store their data here. The directory will be created the first time a plugin needs it.

crowd.properties

Crowd 3.0.0 and newer versions don't use the `crowd.properties` file any more. When upgrading from an older version, the required settings from your `crowd.properties` file will be migrated to the database, and the file will be renamed to `crowd.properties.old`. The `crowd.properties` file is still used by CrowdID, and other external integrations.

Shared directory

This directory contains common data for all nodes in your Crowd installation. If you are using Crowd Data Center, this directory is expected to be a network share accessible from every node. If you are not using Crowd Data Center, this will be an ordinary local directory. By default it is located in the `shared` sub-directory of your Crowd home directory.

crowd.cfg.xml file

This file stores configuration information for the Crowd Administration Console application, including:

- License information
- Server ID
- Database configuration properties
- Setup phase reached

The contents of this file is automatically generated when you run the [Crowd Setup Wizard](#).

Here's an example of the content of `crowd.cfg.xml`, when the embedded HSQL database was specified at setup:

```
<?xml version="1.0" encoding="UTF-8"?>

<application-configuration>
  <setupStep>complete</setupStep>
  <setupType>install.new</setupType>
  <buildNumber>320</buildNumber>
  <properties>
    <property name="crowd.server.id">B9AN-B9AN-B9AN-B9AN</property>
    <property name="hibernate.c3p0.acquire_increment">1</property>
    <property name="hibernate.c3p0.idle_test_period">100</property>
    <property name="hibernate.c3p0.max_size">15</property>
    <property name="hibernate.c3p0.max_statements">0</property>
    <property name="hibernate.c3p0.min_size">0</property>
    <property name="hibernate.c3p0.timeout">30</property>
    <property name="hibernate.connection.driver_class">org.hsqldb.jdbcDriver</property>
    <property name="hibernate.connection.password"></property>
    <property name="hibernate.connection.url">jdbc:hsqldb:C:/data/crowd-home-15/database/defaulttdb<
  /property>
    <property name="hibernate.connection.username">sa</property>
    <property name="hibernate.dialect">org.hibernate.dialect.HSQLDialect</property>
    <property name="hibernate.setup">true</property>
    <property name="license">AAABGQ00DAoPeNpdkF1LwzAUhu/plus-some-more-stuff</property>
  </properties>
</application-configuration>
```

plugins directory

The `plugins` directory is a sub-directory of your Crowd shared directory. This directory will contain plugins that are not shipped with Crowd and that you have installed separately onto your Crowd instance.

backups directory

The `backups` directory is a sub-directory of your Crowd shared directory. This is the default location of Crowd backups.

Installation directory

This is the directory into which the downloaded Crowd application has been unzipped during [installation](#).

Important files in the Crowd installation directory:

- [crowd-init.properties file](#)
- [build.properties file](#)
- [build.xml file](#)
- [database directory](#)

crowd-init.properties file

This is where you specify your Crowd home directory (described [above](#)). You can set the location during [installation](#).

The `crowd-init.properties` file is located in the Crowd installation directory at `{CROWD_INSTALL}\crowd-webapp\WEB-INF\classes\crowd-init.properties`.

The file content looks something like this before it has been customized:

```
## You can specify your crowd.home property here or in your system environment variables.

# On Windows-based operating systems, uncomment the following
# line and set crowd.home to a directory Crowd should use to
# store its configuration.
# NOTE: use forward slashes instead of backward slashes

#crowd.home=c:/data/crowd-home

# On Unix-based operating systems, uncomment the following
# line and set crowd.home to a directory Crowd should use to
# store its configuration.

#crowd.home=/var/crowd-home
```

build.properties file

This configuration file stores various deployment properties of Crowd and the ['demo' application](#).

The file is located at the root of your Crowd installation directory (described [above](#)).

The default `build.properties` file will look similar to the following:

```
# Modify the attributes of this file to quickly adjust the deployment values of Crowd.

# The Hibernate database dialect to use.
hibernate.dialect=org.hibernate.dialect.HSQLDialect

# The Hibernate transaction factory to use.
hibernate.transaction.factory_class=org.hibernate.transaction.JDBCTransactionFactory

# The http port you wish to run crowd from, ie: http://localhost:8095/crowd
crowd.tomcat.connector.port=8095

# Tomcat requires a unique port for shutdown
crowd.tomcat.shutdown.port=8020

# Crowd context root
crowd.url=http://localhost:8095/crowd

# Demo context root
demo.url=http://localhost:8095/demo

# OpenID server context root
openidserver.url=http://localhost:8095/openidserver
```

Parameter	Description
hibernate.dialect	This parameter controls the database dialect that the Hibernate persistence system will use when executing commands against your database server.
hibernate.transaction.factory_class	This parameter controls the transaction factory to use when executing transactions at run-time: Hibernate provides two generic options, additional application server specific options are available: <ul style="list-style-type: none"> org.hibernate.transaction.JDBCTransactionFactory delegates to database (JDBC) transactions (default). org.hibernate.transaction.JTATransactionFactory delegates to JTA (if an existing transaction is under way, the work performed is done in that context. Otherwise a new transaction is started).
crowd.url	The path and port for the root of the Crowd Administration Console web-application.
demo.url	The path and port for the root of the Crowd demo web-application
openidserver.url	The path and port for the root of the CrowdID web-application

build.xml file

This is an Ant script that loads properties from the `build.properties` configuration file.

The file is located at the root of your Crowd installation directory (described [above](#)).

If configuring Crowd and/or the demo application to run on a port and context path other than the default, you will need to run the command `build.sh` (or `build.bat`) against the `build.xml` configuration file. This process will then edit all of the necessary Crowd configuration files for your deployment.

The sample output from running `build.xml` will look similar to the following:

```
shamid@mocha:~/atlassian-crowd-1.1.0$ ./build.sh
Buildfile: build.xml

init:

assistant:
  Changing Tomcat's connector port to 8095
  Changing Tomcat's shutdown port to 8020
  Configuring the Crowd Console
  Copying crowd.properties to: crowd-webapp/WEB-INF/classes
  Copying 1 file to /home/shamid/atlassian-crowd-1.1.0/crowd-webapp/WEB-INF/classes
  Configuring the Crowd hibernate configuration
  Updating the HibernateDialect and TransactionFactory in crowd-webapp/WEB-INF/classes/jdbc.properties
  Updating property file: /home/shamid/atlassian-crowd-1.1.0/crowd-webapp/WEB-INF/classes/jdbc.properties
  Configuring the demo application
  Renaming and copying demo.properties to: demo-webapp/WEB-INF/classes/crowd.properties
  Copying 1 file to /home/shamid/atlassian-crowd-1.1.0/demo-webapp/WEB-INF/classes
  Configuring the OpenID server application
  Renaming and copying openidserver.properties to: crowd-openidserver-webapp/WEB-INF/classes/crowd.properties
  Copying 1 file to /home/shamid/atlassian-crowd-1.1.0/crowd-openidserver-webapp/WEB-INF/classes
  Configuring the OpenID hibernate configuration
  Updating the HibernateDialect and TransactionFactory in crowd-openidserver-webapp/WEB-INF/classes/jdbc.
properties
  Updating property file: /home/shamid/atlassian-crowd-1.1.0/crowd-openidserver-webapp/WEB-INF/classes/jdbc.
properties

BUILD SUCCESSFUL
Total time: 2 seconds
```

database directory

If you are using the embedded HSQL database, supplied for evaluation purposes, CrowdID will store its database in this directory (note however that the Crowd database will be in the Crowd home directory, not the installation directory.)

RELATED TOPICS

- [Finding the atlassian-crowd.log File](#)
- [Supported Platforms](#)
- [Installing Crowd and CrowdID](#)
- [Running the Setup Wizard](#)
- [Configuring Crowd](#)

DRAFT - .Important Directories and Files vCROWD_3.0

This page contains information about the important directories and files to be aware of when configuring Crowd.

On this page:

- [Crowd Home Directory](#)
- [Crowd Shared Directory](#)
 - [crowd.cfg.xmlfile](#)
 - [plugins directory](#)
 - [backups directory](#)
- [Crowd Installation Directory](#)

i When configuring an application to work with Crowd, you will be interested in the [crowd.properties file](#).

Crowd Home Directory

The Crowd Home directory is where Crowd will store its configuration information. If you are using the embedded HSQL database, supplied for evaluation purposes, Crowd will also store its database in this directory. (Note however that the CrowdID database will be in the Crowd Installation directory, not the Home directory.)

The location of this directory is specified in the `crowd-init.properties` file described [below](#). You can set the location during [installation](#).

Crowd's [System Information](#) screen shows the location of your Crowd Home directory.

Important files and directories in the Crowd Home directory, listed here and described below:

- [crowd.properties file](#)
- [bundled-plugins directory](#)
- [caches directory](#)
- [database directory](#)
- [plugin-data directory](#)

crowd.properties file

As of Crowd 3.0.0 the `crowd.properties` file is not used by the Crowd application itself. During the initial run the required setting from your `crowd.properties` file will be migrated to the database. The `crowd.properties` file is still used by CrowdID, and other external integrations

For more information, refer to the page about [the crowd.properties File](#).

bundled-plugins directory

The `bundled-plugins` directory is a sub-directory of your Crowd Home directory. It contains plugins which are shipped with your Crowd installation, such as:

- The SAML integration plugin which provides the [Google Apps SSO feature](#).
- The [Shared Access Layer \(SAL\)](#) plugins.
- The [REST module](#) plugin.
- And more.

The plugins are a collection of jars generated when you install the Crowd web application. The jars are obtained by unzipping `atlassian-bundled-plugins.zip` from `{CROWD_INSTALL}\crowd-webapp\WEB-INF\classes`.

caches directory

The `caches` directory is a sub-directory of your Crowd Home directory. It contains various files that Crowd caches to improve performance. The files in sub-directories of this directory are either created or updated generated when you install or restart the Crowd web application.

Do not modify or remove these files while Crowd is running. It should be safe for you to delete these files between application restarts.

It may improve Crowd's performance if you link this sub-directory to a fast disk.

database directory

If you are using the embedded HSQL database, supplied for evaluation purposes, Crowd will store its database in this directory. (Note however that the CrowdID database will be in the Crowd Installation directory, not the Crowd Home directory.)

plugin-data directory

The `plugin-data` directory is a sub-directory of your Crowd Home directory. Plugins developed for Crowd 2.12 and older will store their data here. The directory will be created the first time a plugin needs it.

Crowd Shared Directory

This directory contains common data for all nodes in your Crowd installation. If you are using Crowd Data Center, this directory is expected to be a network share accessible from every node. If you are not using Crowd Data Center, this will be an ordinary local directory. By default it is located in the `shared` sub-directory of your Crowd Home directory.

crowd.cfg.xml file

This file stores configuration information for the Crowd Administration Console application, including:

- License information
- Server ID
- Database configuration properties
- Setup phase reached.

The contents of this file is automatically generated when you run the [Crowd Setup Wizard](#).

Here's an example of the content of `crowd.cfg.xml`, when the embedded HSQL database was specified at setup:

```
<?xml version="1.0" encoding="UTF-8"?>

<application-configuration>
  <setupStep>complete</setupStep>
  <setupType>install.new</setupType>
  <buildNumber>320</buildNumber>
  <properties>
    <property name="crowd.server.id">B9AN-B9AN-B9AN-B9AN</property>
    <property name="hibernate.c3p0.acquire_increment">1</property>
    <property name="hibernate.c3p0.idle_test_period">100</property>
    <property name="hibernate.c3p0.max_size">15</property>
    <property name="hibernate.c3p0.max_statements">0</property>
    <property name="hibernate.c3p0.min_size">0</property>
    <property name="hibernate.c3p0.timeout">30</property>
    <property name="hibernate.connection.driver_class">org.hibernate.jdbcDriver</property>
    <property name="hibernate.connection.password"></property>
    <property name="hibernate.connection.url">jdbc:hsqldb:C:/data/crowd-home-15/database/defaultdb<
  /property>
    <property name="hibernate.connection.username">sa</property>
    <property name="hibernate.dialect">org.hibernate.dialect.HSQLDialect</property>
    <property name="hibernate.setup">true</property>
    <property name="license">AAABGQ00DAoPeNpdkFlLwzAUhu/plus-some-more-stuff</property>
  </properties>
</application-configuration>
```

plugins directory

The `plugins` directory is a sub-directory of your shared Crowd directory. This directory will contain plugins that are not shipped with Crowd and that you have installed separately onto your Crowd instance.

backups directory

The `backups` directory is a sub-directory of your shared Crowd directory. This is the default location of Crowd backups.

Crowd Installation Directory

This is the directory into which the downloaded Crowd application has been unzipped during [installation](#).

Important files in the Crowd Installation directory, listed here and described below:

- [crowd-init.properties file](#)
- [build.properties file](#)
- [build.xml file](#)
- [database directory](#)

crowd-init.properties file

This is where you specify your Crowd Home directory (described [above](#)). You can set the location during [installation](#).

The `crowd-init.properties` file is located in the Crowd Installation directory at `{CROWD_INSTALL}\crowd-webapp\WEB-INF\classes\crowd-init.properties`

The file content looks something like this before it has been customized:

```
## You can specify your crowd.home property here or in your system environment variables.

# On Windows-based operating systems, uncomment the following
# line and set crowd.home to a directory Crowd should use to
# store its configuration.
# NOTE: use forward slashes instead of backward slashes

#crowd.home=c:/data/crowd-home

# On Unix-based operating systems, uncomment the following
# line and set crowd.home to a directory Crowd should use to
# store its configuration.

#crowd.home=/var/crowd-home
```

build.properties file

This configuration file stores various deployment properties of Crowd and the ['demo' application](#).

The file is located at the root of your Crowd Installation directory (described [above](#)).

The default build.properties file will look similar to the following:

```
# Modify the attributes of this file to quickly adjust the deployment values of Crowd.

# The Hibernate database dialect to use.
hibernate.dialect=org.hibernate.dialect.HSQLDialect

# The Hibernate transaction factory to use.
hibernate.transaction.factory_class=org.hibernate.transaction.JDBCTransactionFactory

# The http port you wish to run crowd from, ie: http://localhost:8095/crowd
crowd.tomcat.connector.port=8095

# Tomcat requires a unique port for shutdown
crowd.tomcat.shutdown.port=8020

# Crowd context root
crowd.url=http://localhost:8095/crowd

# Demo context root
demo.url=http://localhost:8095/demo

# OpenID server context root
openidserver.url=http://localhost:8095/openidserver
```

Parameter	Description
hibernate.dialect	This parameter controls the database dialect the Hibernate persistence system will use when executing commands versus your database server.
hibernate.transaction.factory_class	This parameter controls the transaction factory to use when executing transactions at run-time: Hibernate provides two generic options, additional application server specific options are available: <ul style="list-style-type: none"> org.hibernate.transaction.JDBCTransactionFactory delegates to database (JDBC) transactions (default). org.hibernate.transaction.JTATransactionFactory delegates to JTA (if an existing transaction is under way, the work performed is done in that context. Otherwise a new transaction is started).
crowd.url	The path and port for the root of the Crowd Administration Console web-application.
demo.url	The path and port for the root of the Crowd demo web-application

openidserver.url	The path and port for the root of the CrowdID web-application	
------------------	---	--

build.xml file

This is an Ant script that loads properties from the `build.properties` configuration file.

The file is located at the root of your Crowd Installation directory (described [above](#)).

If configuring Crowd and/or the demo application to run on a port and context path other than the default, you will need to run the command `build.sh` (or `build.bat`) against the `build.xml` configuration file. This process will then edit all of the necessary Crowd configuration files for your deployment.

The sample output from running `build.xml` will look similar to the following:

```
shamid@mocha:~/atlassian-crowd-1.1.0$ ./build.sh
Buildfile: build.xml

init:

assistant:
  Changing Tomcat's connector port to 8095
  Changing Tomcat's shutdown port to 8020
  Configuring the Crowd Console
  Copying crowd.properties to: crowd-webapp/WEB-INF/classes
  Copying 1 file to /home/shamid/atlassian-crowd-1.1.0/crowd-webapp/WEB-INF/classes
  Configuring the Crowd hibernate configuration
  Updating the HibernateDialect and TransactionFactory in crowd-webapp/WEB-INF/classes/jdbc.properties
  Updating property file: /home/shamid/atlassian-crowd-1.1.0/crowd-webapp/WEB-INF/classes/jdbc.properties
  Configuring the demo application
  Renaming and copying demo.properties to: demo-webapp/WEB-INF/classes/crowd.properties
  Copying 1 file to /home/shamid/atlassian-crowd-1.1.0/demo-webapp/WEB-INF/classes
  Configuring the OpenID server application
  Renaming and copying openidserver.properties to: crowd-openidserver-webapp/WEB-INF/classes/crowd.properties
  Copying 1 file to /home/shamid/atlassian-crowd-1.1.0/crowd-openidserver-webapp/WEB-INF/classes
  Configuring the OpenID hibernate configuration
  Updating the HibernateDialect and TransactionFactory in crowd-openidserver-webapp/WEB-INF/classes/jdbc.properties
  Updating property file: /home/shamid/atlassian-crowd-1.1.0/crowd-openidserver-webapp/WEB-INF/classes/jdbc.properties

BUILD SUCCESSFUL
Total time: 2 seconds
```

database directory

If you are using the embedded HSQL database, supplied for evaluation purposes, CrowdID will store its database in this directory. (Note however that the Crowd database will be in the Crowd Home directory, not the Installation directory.)

RELATED TOPICS

- [Finding the atlassian-crowd.log File](#)
- [Supported Platforms](#)
- [Installing Crowd and CrowdID](#)
- [Running the Setup Wizard](#)
- [Configuring Crowd](#)

The crowd.properties file

The crowd.properties file is no longer used

Starting from Crowd 3.0, the `crowd.properties` file is no longer used by Crowd. When upgrading, the required settings are migrated to the database, and the file is renamed to `crowd.properties.old`. The file is still used by CrowdID and other integrated applications.

When integrating an application with Crowd, you will copy Crowd's client library and the `crowd.properties` configuration file into the application's library. For details of the procedure, refer to [Adding an Application](#).

Attributes of the crowd.properties File

Attribute	Description
application.name	The name that the application will use when authenticating with the Crowd server. This needs to match the name you specified in Adding an Application .
application.password	The password that the application will use when authenticating with the Crowd server. This needs to match the password you specified in Adding an Application .
application.login.url	Crowd will redirect the user to this URL if their authentication token expires or is invalid due to security restrictions.
crowd.server.url	The URL to use when connecting with the integration libraries to communicate with the Crowd server.
crowd.base.url	The URL used by Crowd to create the full URL to be sent to users that reset their passwords.
session.isauthenticated	The session key to use when storing a <code>Boolean</code> value indicating whether the user is authenticated or not.
session.tokenkey	The session key to use when storing a <code>String</code> value of the user's authentication token.
session.validationinterval	The number of minutes to cache authentication validation in the session. If this value is set to 0, each HTTP request will be authenticated with the Crowd server.
session.lastvalidation	The session key to use when storing a <code>Date</code> value of the user's last authentication.

The following **optional** attributes in the `crowd.properties` file allow further customization of the client:

Attribute	Description	Default Value (ms)
http.proxy.host	The name of the proxy server used to transport SOAP traffic to the Crowd server.	(none)
http.proxy.port	The connection port of the proxy server (must be specified if a proxy host is specified).	(none)
http.proxy.username	The username used to authenticate with the proxy server (if the proxy server requires authentication).	(none)

http.proxy.password	The password used to authenticate with the proxy server (if the proxy server requires authentication).	(none)
http.max.connections	The maximum number of HTTP connections in the connection pool for communication with the Crowd server.	20
http.timeout	The HTTP connection timeout (milliseconds) used for communication with the Crowd server. A value of zero indicates that there is no connection timeout.	5000
cookie.domain	<p>A domain to use when setting cookies, overriding the SSO Domain set in Crowd (since Crowd 2.5.2).</p> <p>When an SSO Domain is set in Crowd, all client applications must be in the same domain so cookies can be shared.</p> <p>A Crowd deployment may have hosts with no common domain suffix, for example 'domain.example.com' and 'domain.internal'. Even though a user has already logged in to 'domain.example.com' and has a cookie set, applications running under 'domain.internal' will not receive this cookie since the domains differ, and users will be unable to log in.</p> <p>Set this property in the crowd client application to override the domain. Applications within the same domain will then be able to share SSO sessions.</p>	(none)
cookie.tokenkey	<p>When using Crowd for single sign-on (SSO), you can specify the SSO cookie name for each application. Under the standard configuration, Crowd will use a single, default cookie name for all Crowd-connected applications. You can override the default with your own cookie name.</p> <p>As well as allowing you to define the SSO cookie name, this feature also allows you to divide your applications into different SSO groups. For example, you might use one SSO token for your public websites and another for your internal websites.</p>	crowd.token_key
socket.timeout	The socket timeout in milliseconds. You may wish to override the default value if the latency to the Crowd server is high.	20000

Passing crowd.properties as an Environment Variable

You can pass the location of a client application's `crowd.properties` file to the client application as an environment variable when starting the client application. This means that you can choose a suitable location for the `crowd.properties` file, instead of putting it in the client application's `WEB-INF/classes` directory.

Example:

```
-Dcrowd.properties={FILE-PATH}/crowd.properties
```

RELATED TOPICS

[Passing the crowd.properties File as an Environment Variable](#)
[Important directories and files](#)
[Adding an Application](#)

Changing the Port that Crowd uses

By default, Crowd is configured to use port 8095. If this port is already in use within your network, you will need to change the port that Crowd uses.

Follow these steps:

1. Change the Ports which Crowd Listens On

Default <Crowd-Installation>/apache-tomcat/conf/server.xml file

```
<Server port="8020" shutdown="SHUTDOWN">

  <Service name="Catalina">

    <Connector acceptCount="100" connectionTimeout="20000" disableUploadTimeout="true"
enableLookups="false" maxHttpHeaderSize="8192" maxThreads="150" minSpareThreads="25" port="8095"
redirectPort="8443" useBodyEncodingForURI="true" URIEncoding="UTF-8" />
...

```

Modified <Crowd-Installation>/apache-tomcat/conf/server.xml file - using server port 8021 and connector port 8093

```
<Server port="8021" shutdown="SHUTDOWN">

  <Service name="Catalina">

    <Connector acceptCount="100" connectionTimeout="20000" disableUploadTimeout="true"
enableLookups="false" maxHttpHeaderSize="8192" maxThreads="150" minSpareThreads="25" port="8093"
redirectPort="8443" useBodyEncodingForURI="true" URIEncoding="UTF-8" />
...

```

2. Edit the `build.properties` file, as described in [Important Directories and Files](#).
 - a. Change the `crowd.tomcat.connector.port` to the new port on which the [Crowd Administration Console](#) will be accessed.
 - b. Change the `crowd.url` property to the new port on which the [Crowd Administration Console](#) will be accessed.
 - c. Change the `demo.url` property to the new port on which the [Crowd 'demo' application](#) will be accessed.
 - d. Change the `openidserver.url` property to the new port on which the [CrowdID Server](#) will be accessed.
3. Run the `build.sh`(or `build.bat`) script, as described in [Important Directories and Files](#).
4. Change your Crowd Server Base URL to reflect the port changes as per outlined in our [How To Change The Crowd Base URL](#) KB.




e.g. in this case, the updated Crowd's Server Base URL would be `http://localhost:8093`

RELATED TOPICS

- [Supported Platforms](#)
 - [Setting JAVA_HOME](#)
 - [End of support announcements for Crowd](#)
- [Installing Crowd and CrowdID](#)
 - [Connecting Crowd to a Database](#)
 - [HSQLDB](#)
 - [MS SQL Server](#)
 - [MySQL](#)
 - [Oracle](#)
 - [PostgreSQL](#)
 - [Connecting CrowdID to a Database](#)
 - [HSQLDB for CrowdID](#)
 - [MS SQL Server for CrowdID](#)
 - [MySQL for CrowdID](#)

- [Oracle for CrowdID](#)
 - [PostgreSQL for CrowdID](#)
 - [Specifying your Crowd Home Directory](#)
- [Running the Setup Wizard](#)
 - [Troubleshooting your Configuration on Setup](#)
- [Configuring Crowd](#)
 - [Important directories and files](#)
 - [DRAFT - .Important Directories and Files vCROWD_3.0](#)
 - [The crowd.properties file](#)
 - [Changing the Port that Crowd uses](#)
 - [Configuring Crowd to Work with SSL](#)
 - [Installing Crowd as a Windows Service](#)
 - [Specifying Startup Order of Windows Services](#)
 - [Changing the User for the Crowd Windows Service](#)
 - [Removing the Crowd Windows Service](#)
 - [Troubleshooting Crowd as a Windows Service](#)
 - [Setting Crowd to Start Automatically on Mac OS X](#)
 - [Setting Crowd to Run Automatically and Use an Unprivileged System User on UNIX](#)

Configuring Crowd to Work with SSL

 Atlassian applications allow the use of SSL within our products, however Atlassian Support does not provide assistance for configuring it. Consequently, Atlassian **cannot guarantee providing any support for it.**

- If assistance with conversions of certificates is required, please consult with the vendor who provided the certificate.
- If assistance with configuration is required, please raise a question on [Atlassian Answers](#).

Why should you enable HTTPS access to Crowd?

When web applications are accessed across the internet, there is always the possibility of usernames and passwords being intercepted by intermediaries. HTTPS is a good way to safeguard your Crowd data and user logins from being intercepted and read by outsiders.

On this page:

- [Using Crowd over HTTPS](#)
 - [Step 1: Enable Tomcat HTTPS Access](#)
 - [Step 2: Create or Import your SSL Key \(Self-Signed or CA-Issued\)](#)
 - [Creating a Self-Signed SSL Key](#)
 - [Importing a CA-Issued Certificate](#)
 - [Troubleshooting](#)
- [Using SSL between an LDAP Server and Crowd](#)
 - [Microsoft Active Directory Connector using SSL Certificate](#)
 - [Other LDAP Servers](#)

Using Crowd over HTTPS

The process of enabling [HTTPS](#) access is specific to each application server, but specifying which pages require protection is generic. Below we describe the process for Tomcat, the application server bundled with Crowd.

Step 1: Enable Tomcat HTTPS Access

Edit `<crowd installation>/apache-tomcat/conf/server.xml`, and at the bottom before the `</Service>` tag (not to be confused with the `</Server>` tag!), add this section (or uncomment it if it's already there):

```
<Connector port="8443" protocol="org.apache.coyote.http11.Http11Protocol"
    maxThreads="150" SSLEnabled="true" scheme="https" secure="true"
    clientAuth="false" sslProtocol="TLS"
    keystoreFile="${user.home}/.keystore" keystorePass="changeit"
    keyAlias="tomcat" keyPass="changeit"/>
```

This enables SSL access on port 8443. (The default for HTTPS is 443, but just as Tomcat uses 8080 instead of 80 to avoid conflicts, 8443 is used instead of 443 here). You may need to change the values of `keystoreFile`, `keystorePass` and `keyPass` as appropriate for your certificates and set-up.

Step 2: Create or Import your SSL Key (Self-Signed or CA-Issued)

You can either create a self-signed SSL key or import a certificate issued by a Certificate Authority (CA). We describe both methods below.

Creating a Self-Signed SSL Key

You can create a self-signed key for testing purposes with one of the following commands:

```
%JAVA_HOME%\bin\keytool -genkey -alias tomcat -keyalg RSA (Windows)
$JAVA_HOME/bin/keytool -genkey -alias tomcat -keyalg RSA (Unix / Mac OS)
```

When you are asked for your "first and last name", instead supply the hostname for the Crowd server, e.g.:

```
What is your first and last name?
[Unknown]: localhost
```

The keytool utility will prompt you for two passwords: the keystore password and the key password for Tomcat. You can use either of:

1. 'changeit' (this is the default value Tomcat expects), or
2. Any value other than 'changeit', and you must also specify it as the value of `keystorePass` in `conf/server.xml`.

You will then need to import your certificate into the truststore:

1. First, export the key you generated to a file:

```
$JAVA_HOME/bin/keytool -export -alias tomcat -file tomcat.cert
```

2. Import the key into the JRE keystore (you will need permission to write to the keystore specified, and may need elevated privileges):

```
$JAVA_HOME/bin/keytool -import -alias tomcat -file tomcat.cert -keystore $JAVA_HOME/jre/lib/security/cacerts
```

For information on adding a key pair issued by a Certificate Authority (CA), refer to the [Apache Tomcat documentation](#).

Importing a CA-Issued Certificate

When using certificates issued by a Certificate Authority, you also need to import the certificate using the `keytool` command, rather than generating a self-signed key.

Here is an example of the command:

```
keytool -import -alias tomcat -file certificate.cer -keystore some/path/to/file -storepass something.secure
```

The `-file` is your certificate and the `-keystore` is an optional destination, but it will guarantee that you know where your keystore is. By default, the keystore is placed in your user home directory. You can refer to the following Oracle documentation for more information on the `keytool`:

- [Solaris and Linux](#)
- [Windows](#)

Now edit the `server.xml` file as described in section 'Edit the Tomcat Configuration File' in the [Apache Tomcat documentation](#). Basically, you'll need to add the `keystoreFile` and `keystorePass` to the SSL Connector definition to match your keystore settings.

Now start (or restart) your Crowd instance. You should be able to access Crowd at this URL:

```
https://localhost:8443/crowd/console
```

Troubleshooting

Here are some troubleshooting tips if you are using a self-signed key created by keytool, as described above.

When you enter 'https://localhost:8443' in your browser, if you get a message such as 'Cannot establish a connection to the server at localhost:8443', look for error messages in your `logs/catalina.out` log file. Here are some possible errors with explanations:

Can't Find the Keystore

```
java.io.FileNotFoundException: /home/<username>/.keystore (No such file or directory)
```

This indicates that Tomcat cannot find the keystore. The keytool utility creates the keystore as a file called `.keystore` in the current user's home directory. For Unix/Linux the home directory is likely to be `/home/<username>`. For Windows it is likely to be `C:\Documents And Settings\<UserName>`.

Make sure you are running Crowd as the same user who created the keystore. If this is not the case, or if you are running Crowd on Windows as a service, you will need to specify where the keystore file is in `conf/server.xml`. Add the following attribute to the connector tag you uncommented: `keystoreFile="<location of keystore file>"`

Incorrect Password

```
java.io.IOException: Keystore was tampered with, or password was incorrect
```

You used a different password than 'changeit'. You must either use 'changeit' for both the keystore password and for the key password for Tomcat, or if you want to use a different password, you must specify it using the `keystorePass` attribute of the Connector tag, as described above.

Passwords don't Match

```
java.io.IOException: Cannot recover key
```

You specified a different value for the keystore password and the key password for Tomcat. Both passwords must be the same.

To find out more about the options that Tomcat offers, please take a look at the [Apache Tomcat documentation](#).

Using SSL between an LDAP Server and Crowd

Microsoft Active Directory Connector using SSL Certificate

Please refer to [Configuring an SSL Certificate for Microsoft Active Directory](#).

Other LDAP Servers

For other LDAP servers, please consult your LDAP server documentation.

On the Crowd side, when [configuring the connector properties](#), you will have to simply check the 'Secure SSL' box and make sure you use the correct port in the 'URL' field (usually 636).

RELATED TOPICS

[Configuring an SSL Certificate for Microsoft Active Directory](#)
[Configuring Crowd](#)

Installing Crowd as a Windows Service



If you are trying to set up Crowd as a Windows Service on a 64 bit machine, you should ensure that Crowd uses 64-bit Tomcat binaries. See the [Crowd installation guide](#) for more details.

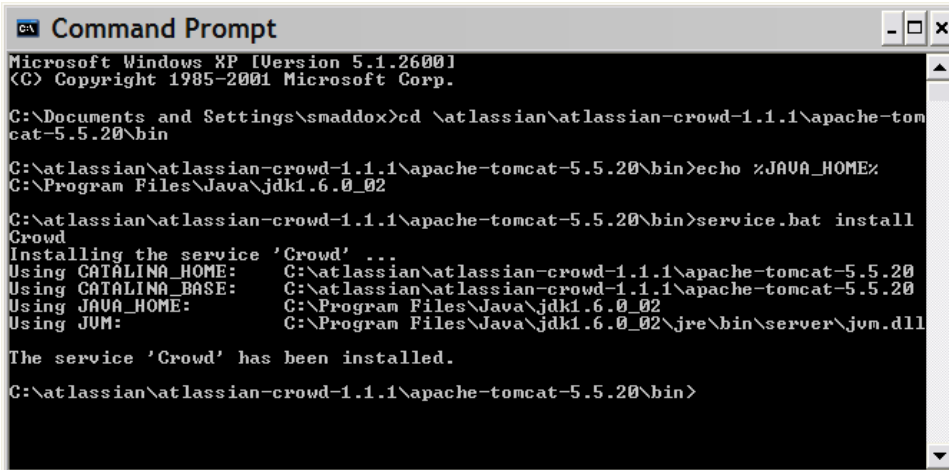
For long-term use, you should configure Crowd to restart automatically when the operating system restarts. For Windows servers, this means configuring Crowd to run as a Windows service.

Running Crowd as a Windows service has other advantages. When Crowd is started manually, a console window opens - there is a risk that someone may accidentally shut down Crowd by closing the window. Also, the Crowd logs are properly managed by the Windows service (reliably found in `\atlassian-crowd.log` in the root Crowd directory, and rotated by file size).

Installing Crowd as a Windows Service

1. Open a DOS Command prompt as Administrator (search for "cmd" in the start menu/screen, right click and "Run as Administrator").
2. 'cd' to your Crowd directory, and then the Tomcat bin subdirectory, e.g. `{CROWD_INSTALL}\apache-tomcat\bin`
3. If a directory in the path has spaces (e.g. `C:\Program Files\..`), please convert it to its eight-character equivalent (e.g. `c:\Progra~1\..`).
4. Ensure the `JAVA_HOME` variable is set to the JDK base directory. Use `echo %JAVA_HOME%` to confirm this.
5. Run the following command:

```
service.bat install Crowd
```

Screenshot: Installing Crowd as a Windows Service


```

Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\smaddox>cd \atlassian\atlassian-crowd-1.1.1\apache-tomcat-5.5.20\bin

C:\atlassian\atlassian-crowd-1.1.1\apache-tomcat-5.5.20\bin>echo %JAVA_HOME%
C:\Program Files\Java\jdk1.6.0_02

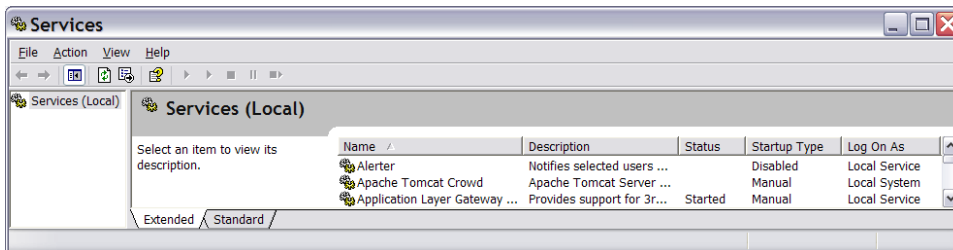
C:\atlassian\atlassian-crowd-1.1.1\apache-tomcat-5.5.20\bin>service.bat install
Crowd
Installing the service 'Crowd' ...
Using CATALINA_HOME: C:\atlassian\atlassian-crowd-1.1.1\apache-tomcat-5.5.20
Using CATALINA_BASE: C:\atlassian\atlassian-crowd-1.1.1\apache-tomcat-5.5.20
Using JAVA_HOME: C:\Program Files\Java\jdk1.6.0_02
Using JUM: C:\Program Files\Java\jdk1.6.0_02\jre\bin\server\jvm.dll

The service 'Crowd' has been installed.

C:\atlassian\atlassian-crowd-1.1.1\apache-tomcat-5.5.20\bin>

```

Crowd should now have been installed as a service, and will be visible in the Windows Services console.

Screenshot: Windows Services Console

- Run the following command, to have the Crowd service start automatically when the server starts:

```
tomcat8w //US//Crowd --Startup auto
```

The Crowd service will automatically start up the next time the server reboots.



- You can manually start the Crowd service with the command `net start Crowd`, and stop it with `net stop Crowd`.
- To see what parameters the Crowd service is starting with, go to **Start -> Run** and run `reg edt32.exe`. There should be an entry at `HKEY_LOCAL_MACHINE -> SOFTWARE -> Apache Software Foundation -> Procrun 2.0 -> Crowd`.

Changing the user running the service

If you wish to run the service as a non-administrator user for security, or if you are using network drives for backups, attachments, or indexes, you can run the service as another user.

To run the service as another user:

- Open the Apache Tomcat Crowd properties.
- Select the **Log On** tab.
- Enter the required username and password.
- Go to Windows **Control Panel > User Accounts** and confirm that the user has write permissions for the {`CROWD_INSTALL`} and {`CROWD_HOME`} directories and all subfolders.



Note that any network drives must be specified by UNC and not letter mappings (eg. `\\backups server\crowd` not `z:\crowd`).

Additional Crowd Setup Options (Optional)

- To increase the maximum memory Crowd can use (the default will be 256MB), run:

```
tomcat8w //US//Crowd --JvmMx 512
```

- If you are running Crowd with Jira and/or Confluence in the same JVM, increase the MaxPermSize to 512 MB:

```
tomcat8w //US//Crowd ++JvmOptions="-XX:MaxPermSize=512m"
```

- Occasionally, it may be useful to view Crowd's Garbage Collection information. This is especially true when investigating memory issues.
 - To turn on the Verbose GC (garbage collection) logging, execute the following command in the command prompt

```
tomcat8w //US//Crowd ++JvmOptions="-Xloggc:path\to\logs\atlassian-gc.log"
```

- The path (denoted by \path\to) refers to the directory in which Crowd is currently installed. For example:

```
tomcat8w //US//Crowd ++JvmOptions="-Xloggc:c:\crowdinstall\logs\atlassian-gc.log"
```

- In order to check all the configurations for the service, you can access:

```
HKEY_LOCAL_MACHINE -> SOFTWARE -> Apache Software Foundation -> Procrun 2.0 -> Crowd -> Parameters -> Java -> Options
```

- If you are using HSQL as your database server: after installing Crowd as a Windows service, you will need to copy your database files.
 - Create a folder called c:\windows\system32\database
 - Copy over the database files from your atlassian-crowd-1.1.2/database.
 - ⚠** We recommend strongly that you use an external database server rather than the HSQL database supplied with Crowd for evaluation purposes.

 Refer to the [Tomcat documentation](#) for further service options.

RELATED TOPICS

- [Specifying Startup Order of Windows Services](#)
- [Changing the User for the Crowd Windows Service](#)
- [Removing the Crowd Windows Service](#)
- [Troubleshooting Crowd as a Windows Service](#)

Specifying Startup Order of Windows Services

This page is relevant if you have [installed Crowd as a Windows service](#).

If you have multiple Windows services that depend on each other, it is important that they are started in the correct order. For example, if you are running both [Jira](#) and Crowd, it is important to start Crowd first, so that Crowd is running before people try to login to JIRA.

For information about specifying the startup order for multiple services, please refer to <http://support.microsoft.com/kb/193888>.

Related Topics

- [Specifying Startup Order of Windows Services](#)
- [Changing the User for the Crowd Windows Service](#)
- [Removing the Crowd Windows Service](#)
- [Troubleshooting Crowd as a Windows Service](#)

- [Installing Crowd as a Windows Service](#)

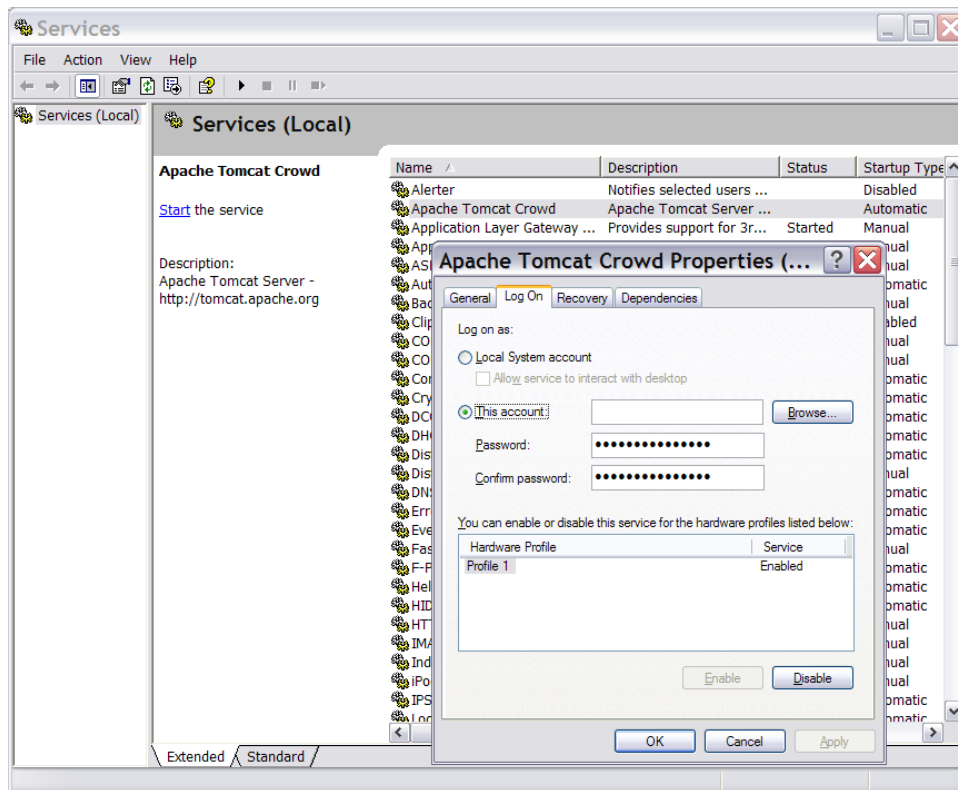
Changing the User for the Crowd Windows Service

This page is relevant if you have [installed Crowd as a Windows service](#). You may want to change the user under which the Crowd Windows service is running, for security reasons.

Changing the Windows User for the Crowd Service

1. Navigate to the service: **Control Panel -> Administrative Tools -> Services**.
2. Locate the '**Apache Tomcat Crowd**' service, right-click and view the '**Properties**'.
3. Go to the '**Log On**' tab and change the user as desired.

Screenshot: [Changing the User for the Windows Service](#)



RELATED TOPICS

- [Specifying Startup Order of Windows Services](#)
- [Changing the User for the Crowd Windows Service](#)
- [Removing the Crowd Windows Service](#)
- [Troubleshooting Crowd as a Windows Service](#)

- [Installing Crowd as a Windows Service](#)

Removing the Crowd Windows Service

This page is relevant if you have [installed Crowd as a Windows service](#)

To remove the Crowd Windows service:

1. Open a DOS prompt.
2. 'cd' to your Crowd directory, and then the Tomcat bin subdirectory, e.g. {CROWD_INSTALL}\apache-tomcat-5.5.20\bin
3. Run one of the following commands:

- Either:

```
service.bat remove Crowd
```

- Or if the above does not work, use

```
tomcat5 //DS//Crowd
```

RELATED TOPICS

- [Specifying Startup Order of Windows Services](#)
- [Changing the User for the Crowd Windows Service](#)
- [Removing the Crowd Windows Service](#)
- [Troubleshooting Crowd as a Windows Service](#)

- [Installing Crowd as a Windows Service](#)

Troubleshooting Crowd as a Windows Service

This page is relevant if you have [installed Crowd as a Windows service](#).

Problem with JDK 6

Problems may occur when trying to set up Crowd to run as a Windows service with JDK 1.6. The problem is caused by a failure to locate `MSVCR71.DLL`, which can be found in your `%JAVA_HOME%/bin`. There are two options to resolve this problem:

- Add `%JAVA_HOME%/bin` to `PATH`, then restart the server.
- Or copy `MSVCR71.DLL` to system path: either `C:\WINDOWS\SYSTEM32` or `C:\WINNT\SYSTEM32`

Please refer to our [Knowledge Base article](#) if you need more details of this issue.

Notes for Windows 64-bit Operating Systems

If you are running 64-bit Windows, please note that Apache Tomcat cannot run as a Windows service if you are using a 64-bit JDK. **Please ensure that you are using a 32-bit JDK.** This is the recommended solution.

Alternatively, you can install a [64-bit JDK](#) and set `JAVA_HOME` to its location. Then follow the same steps above for [Installing Crowd as a Windows Service](#). You'll need to replace `{CROWD_INSTALL}\apache-tomcat-5.5.20\bin\tomcat.exe` with one compiled for 64-bit from these locations:

- http://svn.apache.org/viewvc/tomcat/tc5.5.x/tags/TOMCAT_5_5_24/connectors/procrun/bin/
- http://svn.apache.org/viewvc/tomcat/tc6.0.x/tags/TOMCAT_6_0_16/res/procrun/

RELATED TOPICS

- [Specifying Startup Order of Windows Services](#)
- [Changing the User for the Crowd Windows Service](#)
- [Removing the Crowd Windows Service](#)
- [Troubleshooting Crowd as a Windows Service](#)

- [Installing Crowd as a Windows Service](#)

Setting Crowd to Start Automatically on Mac OS X

For long-term use, you should configure Crowd to restart automatically when the operating system restarts. On Mac OS X, the system startup program called [launchd](#) manages long running processes daemons or services.

Apple provides an [introduction to launchd](#). Below we tell you how to use launchd to start Crowd automatically on Mac OS X when running Tomcat.

On this page:

- [Using launchd with Tomcat](#)
 - [Step 1. Add a Wrapper Shell Script](#)
 - [Step 2. Add a launchd Property List](#)
 - [Starting and Stopping Crowd Manually](#)
 - [Troubleshooting](#)

Using launchd with Tomcat

The Crowd distribution (not EAR-WAR) ships with Tomcat. There is a mismatch between how launchd expects a daemon to behave, and how the default startup scripts for Tomcat operate:

- OS X's launchd expects the process it starts to run forever, but 'catalina.sh start' starts the JVM to run Tomcat and then exits.
- Tomcat provides 'catalina.sh stop' to shut down Tomcat cleanly by connecting to a socket which Tomcat listens on, but launchd stops daemons by sending them a signal that kills the process immediately if no specific handling is included.

You will need a wrapper shell script and properties list to make launchd work with Tomcat.

Step 1. Add a Wrapper Shell Script

Add the following wrapper shell script to `$(CATALINA_HOME)/bin`:

launchd_wrapper.sh

```
#!/bin/bash

function shutdown()
{
    date
    echo "Shutting down Crowd"
    $CATALINA_HOME/bin/catalina.sh stop
}

date
echo "Starting Crowd"
export CATALINA_PID=/tmp/$$

# Uncomment to increase Tomcat's maximum heap allocation
# export JAVA_OPTS=-Xmx512M $JAVA_OPTS

. $CATALINA_HOME/bin/catalina.sh start

# Allow any signal that would kill a process to stop Tomcat
trap shutdown HUP INT QUIT ABRT KILL ALRM TERM TSTP

echo "Waiting for `cat $CATALINA_PID`"
wait `cat $CATALINA_PID`
```

The above shell script starts Tomcat and then waits for the process to complete, so launchd is happy that Tomcat is still running. The script also installs a signal handler, which calls the `shutdown()` function to cleanly shut down Tomcat when launchd signals the script.

You can try this script manually: Start the script, watch Crowd start, and then type **ctrl-C** and see Crowd shut down cleanly. (Note that it will **not** shut down cleanly if Tomcat has not started yet. It takes a few seconds for Tomcat to start listening on the shutdown socket.)

Step 2. Add a launchd Property List

The launchd property list (`.plist`) tells launchd how to start Tomcat.

Add the following plist file to `/Library/LaunchDaemons`, which is the location for system-wide services which are not part of base OS X:

crowd.plist

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>Disabled</key>
  <false/>
  <key>EnvironmentVariables</key>
  <dict>
    <key>CATALINA_HOME</key>
    <string>/Users/myname/conf/crowd-x.x.x</string>
    <key>JAVA_HOME</key>
    <string>/Library/Java/Home</string>
  </dict>
  <key>Label</key>
  <string>com.atlassian.crowd</string>
  <key>OnDemand</key>
  <false/>
  <key>ProgramArguments</key>
  <array>
    <string>/Users/myname/conf/crowd-x.x.x/bin/launchd_wrapper.sh</string>
  </array>
  <key>RunAtLoad</key>
  <true/>
  <key>ServiceDescription</key>
  <string>Crowd</string>
  <key>StandardErrorPath</key>
  <string>/Users/myname/conf/crowd-x.x.x/logs/launchd.stderr</string>
  <key>StandardOutPath</key>
  <string>/Users/myname/conf/crowd-x.x.x/logs/launchd.stdout</string>
  <key>UserName</key>
  <string>root</string>
</dict>
</plist>
```

Notes:

1. Replace `/Users/myname/conf/crowd-x.x.x` with the path to your Crowd installation. The string occurs four times in the above script.
2. `JAVA_HOME` is set to use the default JDK. On OS X version 10.4.4, the default JDK is 1.4.2. You will need to change this value if you want to use a different version of Java. For example, if you want to use JDK 1.5, you will need to change `JAVA_HOME` to `/System/Library/Frameworks/JavaVM.framework/Versions/1.5`.
3. In the above script, we have specified `'root'` as the `UserName`. If necessary, change the `UserName` to the user you want Tomcat to run as.

Starting and Stopping Crowd Manually

To start and stop Crowd manually, use the following commands:

- **Start:**

```
cd /Library/LaunchDaemons
sudo launchctl load -w crowd.plist
```

- Stop:

```
cd /Library/LaunchDaemons  
sudo launchctl unload -w crowd.plist
```

Troubleshooting

- Make sure both files `launch_wrapper.sh` and `crowd.plist` have the necessary file privileges.
- Check the console logging and log file for any abnormalities.

RELATED TOPICS

[Configuring Crowd](#)

Setting Crowd to Run Automatically and Use an Unprivileged System User on UNIX

This page contains some useful information about running Crowd under Linux/UNIX:

- **Dedicated system user.** For security reasons, and to keep your system administrator happy, you should probably create a dedicated non-root user to run Crowd.
- **Automatic startup.** It is useful to set up Crowd to run automatically on UNIX startup.

Running Crowd as an Unprivileged User

Here is an example of some of the changes you can make to *harden up* the directory and file permissions for Crowd to run as a non-root user.

You will need to update the environment variables to suit your installation. This is also for use in BASH. If you are using a different shell, you might need to tweak some things.

```
#!/bin/bash
CROWD_USER="crowd"
CROWD_GROUP="crowd"
INSTALL_BASE="/opt/crowd/atlassian-crowd-3.5.1"
CROWD_HOME="/var/crowd-home"
sudo chgrp ${CROWD_GROUP} ${INSTALL_BASE}/{*.sh,apache-tomcat/bin/*.sh}
sudo chmod g+x ${INSTALL_BASE}/{*.sh,apache-tomcat/bin/*.sh}
sudo chown -R ${CROWD_USER} ${CROWD_HOME} ${INSTALL_BASE}/apache-tomcat/{logs,work,temp}
sudo touch -a ${INSTALL_BASE}/atlassian-crowd-openid-server.log
sudo mkdir ${INSTALL_BASE}/database
sudo chown -R ${CROWD_USER} ${INSTALL_BASE}/{database,atlassian-crowd-openid-server.log}
```

Getting Crowd to Start Automatically

1. Create an `init.d` file (for example, 'crowd.init.d') inside your `{CROWD_INSTALL}` directory:

```
#!/bin/sh -e
# Crowd startup script
#chkconfig: 2345 80 05
#description: Crowd

# Define some variables
# Name of app ( JIRA, Confluence, etc )
APP=crowd
# Name of the user to run as
USER=crowd
# Location of Crowd install directory
CATALINA_HOME=/usr/local/crowd/atlassian-crowd-3.5.1
# Location of Java JDK
export JAVA_HOME=/usr/lib/jvm/adoptopenjdk-8-hotspot-amd64

case "$1" in
  # Start command
  start)
    echo "Starting $APP"
    /bin/su -m $USER -c "$CATALINA_HOME/start_crowd.sh &> /dev/null"
    ;;
  # Stop command
  stop)
    echo "Stopping $APP"
    /bin/su -m $USER -c "$CATALINA_HOME/stop_crowd.sh &> /dev/null"
    echo "$APP stopped successfully"
    ;;
  # Restart command
  restart)
    $0 stop
    sleep 5
    $0 start
    ;;
  *)
    echo "Usage: /etc/init.d/$APP {start|restart|stop}"
    exit 1
    ;;
esac

exit 0
```

2. Create a symbolic link from `/etc/init.d/crowd` to the `init.d` file file.

Hint for Red Hat systems

On Red Hat and Red Hat-based systems such as CentOS, if you put the above script in `/etc/init.d`, you can create the necessary symbolic links with the `chkconfig` script, since all the required information is in the script header.

```
sudo /sbin/chkconfig --add SCRIPT_NAME
```

Replace "SCRIPT_NAME" with whatever the real name of the script is.

Thank you for this information

Thank you to [Matthew Block](#) and [Pete Toscano](#) for the original comments that we based this information on.

Upgrading Crowd

Below are instructions on upgrading an existing Crowd installation to the latest version of Crowd. There are two upgrade procedures to choose from:

- **Method 1: Automatic database upgrade.** Install the new version of Crowd and simply point it at your existing home directory. The upgrade procedure automatically updates your Crowd database.
- **Method 2: Data transfer via XML backup.** Back up your Crowd database to XML before starting the upgrade, install the new version of Crowd and then import the data into your new Crowd installation.

Recommended Upgrade Procedure

Upgrading from Crowd 2.0 or later

If you are upgrading from Crowd 2.0 or later, you can use [method 1, automatic database upgrade](#) or [method 2, data transfer via XML backup](#).

Upgrading from Crowd 1.6 or earlier

If you are upgrading from a version of Crowd prior to 2.0, please make your choice based on your database server, the version of Crowd you are upgrading from and the version you are upgrading to.

- If you are using [PostgreSQL](#), [MySQL](#) or [Microsoft SQL Server](#) and:
 - Upgrading from Crowd 1.3 or later, to Crowd 2.0.4 or later use [method 1, automatic database upgrade](#).
 - Upgrading from Crowd 1.2 or earlier, to Crowd 2.0.4 or later use [method 2, data transfer via XML backup](#).
- If you are using **any other database server** use [method 2, data transfer via XML backup](#)

Alternatives

These are some options you may like to consider:

- If you prefer [method 2, data transfer via XML backup](#), you can choose that option for any database server and no matter which version of Crowd you are upgrading from or to.
- If you are upgrading from Crowd 1.2 or earlier, are using PostgreSQL, MySQL or Microsoft SQL Server, and cannot perform an XML backup:
 1. Upgrade to Crowd 1.6 first, following the instructions in the [Crowd 1.6 upgrade guide](#).
 2. Then upgrade from Crowd 1.6 to Crowd 2.0.4 or later, using the automatic database upgrade as described in the [Crowd 2.0 upgrade guide](#).
- If for some reason you must upgrade to Crowd 2.0.0, 2.0.1, 2.0.2 or 2.0.3 (and cannot upgrade to Crowd 2.0.4), follow [method 2, data transfer via XML backup](#).

Upgrading Crowd via Automatic Database Upgrade

Below are instructions on upgrading an existing Crowd installation to the latest version of Crowd, using the automatic database upgrade.


Before you begin

- Decided on a [recommended upgrade procedure for your database server and Crowd version](#).
- The [Release Notes](#) for the version you are upgrading to, and
- The [Upgrade Notes](#) for any versions you are skipping as well as the version you are upgrading to.

Step 1. Shut Down Crowd and All Integrated Applications

Shut down Crowd and all Crowd-connected applications.

Step 2. Back Up your Crowd Files

1. Use your database backup tools to back up your [Crowd database](#) and your [CrowdID database](#). We **highly recommend** this step, in case something goes wrong during the upgrade process and you need to restore your data from backup.
2. Make backup copies of the following files:
 - Back up your [Crowd Home directory](#), in the location specified in the `crowd-init.properties` file recommended in case something goes wrong during the upgrade process.
 - If your existing Crowd installation is version 1.3.x or 1.4.x: Back up the [crowd.properties](#) file for the Crowd Administration Console application, located at `{CROWD_INSTALL}\crowd-webapp\WEB-INF\classes\crowd.properties` you will need to copy this file to your new Crowd installation.
 -  This step is not required if your current Crowd installation is 1.5 or later.
 - Back up the [crowd.properties](#) file for the **CrowdID application**, located at `{CROWD_INSTALL}/crowd-openidserver-webapp/WEB-INF/classes/crowd.properties` you will need to copy this file to your new Crowd installation.
 - Back up your Crowd JDBC Driver if you have configured [Crowd with a database](#).
3. We recommend that you rename your existing `{CROWD_INSTALL}` directory, because legacy files may cause problems if you unzip the new Crowd installation into an existing directory.

Step 3. Re-Install Crowd

1. [Download Crowd](#).
2. Unzip the download archive into a directory of your choice, taking note of the following:
 - Please make sure that your new `{CROWD_INSTALL}` directory has a different name from your old `{CROWD_INSTALL}` directory.
 - Please check your unzip program before extracting the downloaded archive see the note on the [Crowd installation front page](#).
 - Do not specify directory names that contain spaces.
 - We will refer to this installation directory, where you unzipped the archive, as **{CROWD_INSTALL}**.
3. Point the new Crowd installation at your existing **Crowd Home** directory by editing the configuration file at `{CROWD_INSTALL}\crowd-webapp\WEB-INF\classes\crowd-init.properties`.
The **Crowd Home** directory is where Crowd will store its configuration information. If you are using the embedded HSQL database, supplied for evaluation purposes, Crowd will also store its database in this directory. (Note however that the CrowdID database will be in the installation directory, not the Home directory.) To specify the Crowd Home directory:
 - Open the `crowd-init.properties` file. This is found at `<crowd_install_directory>/crowd-webapp/WEB-INF/classes/crowd-init.properties`
 - Choose the appropriate line in the file, depending upon your operating system (see below).
 - Remove the # at the beginning of the line.
 - Enter the name of the directory you want Crowd to use as its Home directory. For example,
 - On Windows:

```
crowd.home=c:/data/crowd-home
```

Note: On Windows, make sure you use forward slashes as shown above, not backward slashes.

- On Mac and UNIX-based systems:

```
crowd.home=/var/crowd-home
```


Important

Please, ensure that the Crowd Home directory will not match the Crowd installation directory **AND** it is writable by the user executing the initialization script.

- Save the `crowd-init.properties` file.

Use the same Crowd Home directory as used in your previous Crowd installation

Make sure you point the new Crowd installation at your **existing** Crowd Home directory so that the new Crowd can use your existing configuration.

4. Copy the following files, saved in [Step 2 above](#), to your new Crowd installation:
 - If your existing Crowd installation is version 1.3.x or 1.4.x: Copy the `crowd.properties` file for the Crowd Administration Console to the root of your Crowd Home directory.
 -  As from Crowd 1.5, the `crowd.properties` file is located in the Home directory and not the Installation directory. This step is not required if your current Crowd installation is 1.5 or later.
 - Copy the `crowd.properties` file for the **CrowdID application** to your new `{CROWD_INSTALL}/crowd-openidserver-webapp/WEBINF/classes` directory.
 - Copy your Crowd JDBC Driver if you have configured [Crowd with a database](#).

Step 4. Update your Integrated Applications

1. If you have installed Crowd on a new server, or changed Crowd's URL or port number, you will need to update Crowd's Base URL in the General settings section of the admin menu
2. If you are using [CrowdID](#) with an external database, you will need to use the manual JNDI datasource configuration method to [configure an external database connection](#).
3. If you are using CrowdID with the default HSQL database, copy the `thedatabase/` directory from your old installation directory into your new installation directory. Please note that the HSQL database is not suitable for production environments. [Connecting CrowdID to a Database](#) describes how to migration to an enterprise database.

Step 5. Start Crowd

1. Run the start-up script, found in your `{CROWD_INSTALL}` directory:
 - `start_crowd.bat` for Windows.
 - `start_crowd.sh` for Mac and Unix-based systems.
2. Point a web browser at `http://localhost:8095/crowd`. You should now be able to use the [Crowd Administration Console](#).

Troubleshooting

If you have any problems during upgrade, please raise a support request at <https://support.atlassian.com/> and attach your `atlassian-crowd.log` file so that we can help you find out what's gone wrong.


Upgrading Crowd via XML Data Transfer

Below are instructions on upgrading an existing Crowd installation to the latest version of Crowd, using the procedure that transfers your Crowd data via XML backup.

Before you begin

- Decided on [a recommended upgrade procedure for your database server and Crowd version](#)
- Read the [Release Notes](#) for the version you are upgrading to.
- Read the [Upgrade Notes](#) for any versions you are skipping as well as the version you are upgrading to.

Step 1. Export your Crowd Database to XML

1. In the top-right corner, click  > **Backup**.
2. Follow the screen prompts to back up your Crowd database to an XML file. For full instructions, see our guide on [backing up data](#).

Step 2. Shut down Crowd and All Integrated Applications

1. Shut down Crowd and all Crowd-connected applications.

Step 3. Back Up your Crowd Files

1. Use your database backup tools to back up your [Crowd database](#) and your [CrowdID database](#). We **highly recommend** this step, in case something goes wrong during the upgrade process and you need to restore your data from backup.
2. Make backup copies of the following files:
 - The [crowd.properties](#) file for the **CrowdID** application, located at `{CROWD_INSTALL}/crowd-openidserver-webapp/WEB-INF/classes/crowd.properties` You will need to copy this file to your new Crowd installation.
 - Your Crowd JDBC Driver if you have configured [Crowd with a database](#) You will need to copy this file to your new Crowd installation.
 - Your [Crowd Home directory](#), in the location specified in the `crowd-init.properties` file Recommended in case something goes wrong during the upgrade process.
3. Rename your existing `{CROWD_INSTALL}` directory, because legacy files may cause problems if you unzip the new Crowd installation into an existing directory.

Step 4. Download and Re-Install Crowd

1. [Download Crowd](#).
2. Unzip the downloaded archive into a directory of your choice.



- Make sure that your new `{CROWD_INSTALL}` directory has a different name from your old `{CROWD_INSTALL}` directory.
- Check your unzip program before extracting the downloaded archive see the note on the [Crowd installation front page](#).
- Do not use empty spaces in directory names.
- We will refer to this installation directory, where you unzipped the archive, as `{CROWD_INSTALL}`.

- Specify a **new Crowd Home** directory for your new Crowd installation by editing the configuration file at {CROWD_INSTALL}\crowd-webapp\WEB-INF\classes\crowd-init.properties. The **Crowd Home** directory is where Crowd will store its configuration information. If you are using the embedded HSQL database, supplied for evaluation purposes, Crowd will also store its database in this directory. (Note however that the CrowdID database will be in the installation directory, not the Home directory.) To specify the Crowd Home directory:

- Open the crowd-init.properties file. This is found at <crowd_install_directory>/crowd-webapp/WEB-INF/classes/crowd-init.properties
- Choose the appropriate line in the file, depending upon your operating system (see below).
- Remove the # at the beginning of the line.
- Enter the name of the directory you want Crowd to use as its Home directory. For example,
 - On Windows:

```
crowd.home=c:/data/crowd-home
```

Note: On Windows, make sure you use forward slashes as shown above, not backward slashes.

- On Mac and UNIX-based systems:

```
crowd.home=/var/crowd-home
```



Important

Please, ensure that the Crowd Home directory will not match the Crowd installation directory **AND** it is writable by the user executing the initialization script.

- Save the crowd-init.properties file.



Make sure you point the new Crowd installation to a **new** Crowd Home directory, so that Crowd will do a clean installation. Do not point it at your existing Crowd Home directory.

- Copy the following files, saved in [Step 3 above](#), to your new Crowd installation folder:
 - Copy the crowd.properties file for the **CrowdID** application to your new {CROWD_INSTALL}/crowd-openidserver-webapp/WEB-INF/classes directory.
 - Copy your Crowd JDBC Driver if you have configured [Crowd with a database](#).

Step 5. Start Crowd and Run the Setup Wizard

- Run the start-up script, found in your {CROWD_INSTALL} directory:
 - start_crowd.bat for Windows.
 - start_crowd.sh for Mac and Unix-based systems.
- Point a web browser at <http://localhost:8095/crowd> where you will see the **Crowd Setup Wizard**
- Enter your license key on the '**License**' screen, as described in the instructions on the [Setup Wizard](#).
- When asked for your [Installation Type](#), choose '**Import data from an XML Backup**'. This step is required, to import your Crowd data from the XML file which you created in [Step 1 above](#).
- The Setup Wizard will now ask you to [configure your database](#). Supply the JNDI datasource or JDBC connection details of a [new database](#).
- The [Import Existing Crowd Data](#) screen will appear. Enter the location of your XML backup file and click '**Continue**'.
- The Setup Wizard is now complete. You are now ready to log in to the [Crowd Administration Console](#), using your administrator account from your earlier Crowd installation.

Step 6. Update your Integrated Applications

- If you have installed Crowd on a new server, or changed Crowd's URL or port number, you will also need to edit the crowd.properties file in each integrated application accordingly.
- If you are using [CrowdID](#) with an external database, you will need to use the manual JNDI datasource configuration method to [configure an external database connection](#).

3. If you are using CrowdID with the default HSQL database, copy the `database/` directory from your old installation directory into your new installation directory. Please note that the HSQL database is not suitable for production environments. [Connecting CrowdID to a Database](#) describes how to migration to an enterprise database.

Troubleshooting

If you have any problems during upgrade, please raise a support request at <https://support.atlassian.com/> and attach your `atlassian-crowd.log` file so that we can help you find out what's gone wrong.

Crowd 4.4 Upgrade Notes

Here are some important notes on upgrading to **Crowd 4.4**. To learn about new features, see the [release notes](#).



Upgrade notes

Here's some important information you should know about:

Applications are no longer allowed to change the email addresses of Crowd users

Crowd 4.4 improves security by requiring that new email addresses are verified by their users. To avoid working around this requirement, Crowd will reject any requests for changing email addresses coming from connected applications.



Supported platforms

We're ending support for the following platforms:


- PostgreSQL 9.6

Migrate to Another Database

This guide applies to situations when you may need to migrate Crowd to another database.

Before you begin

- Select a new database from one of our [Supported Platforms](#)
- Perform an [XML backup](#) of your existing Crowd server. Make sure that you check the 'Reset Domain' checkbox, otherwise you may be prevented from logging in to the new Crowd Administration Console.
- If you are also migrating the instance to a new server, please also refer to [Migrating Crowd Between Servers](#)

 From this point on, we will call your existing Crowd server the 'original' server.

Migration

1. Copy the XML backup over to the target server.
2. Install Crowd on the target server using our [installation guide](#).
 - The Crowd version can be the same or higher than the version on the original Crowd server.
 - When specifying your [Crowd Home directory](#), make sure you choose a new location and *not* your original Crowd Home directory.
3. Run the [Setup Wizard](#).
 - When asked for the [type of installation](#), choose 'Import data from an XML backup'. Provide the full path to your XML backup file and import the data.
 - When given the option of configuring Crowd to target a database, make sure you choose a new one and *not* your original Crowd database.
4. When the import finishes, shut down Crowd.

Post Migration Verification

1. Start Crowd on the new server. You should be able to authenticate and access Crowd using the same credentials as on your original Crowd server.

Applications and Customizations

For any application you are going to test against this new Crowd server, you will need to modify the application's `crowd.properties` file to point to this new server.

If you have installed any [Crowd plugins](#) or added other customizations, you will need to re-apply them on the new server.

Migrating Crowd Between Servers

This guide applies to situations when you may need to migrate Crowd to a new server, because:

- Your Crowd server hardware is changing.
- You are cloning your production server for a staging, test or development instance.

Preparation

1. Make sure you have a Crowd license for the new server you are targeting. Developer/staging licenses are available for any commercial or academic license. [Create a developer license](#) or [contact us](#) for help.
2. Perform a file system backup of the Crowd Home and Crowd Installation directories.
3. Perform a full database backup.

i From this point on, we will call your existing Crowd server the 'original' server.

Migration

1. Restore the file system and database backups on to the target server.
2. Locate the `shared/crowd-cfg.xml` file in the target server's Crowd Home directory. Modify it to point to the new database, if the name/location of it has changed, in this property: **<property name="hibernate.connection.url">**
3. Locate the `crowd-init.properties` file in your target server's Crowd Installation directory. Modify it to point to the new Crowd Home if the location of it has changed.

Post Migration Verification

1. If your original server serves using HTTPS and the migrated server is on HTTP, you would need to turn off secure cookie with the following SQL query :

```
UPDATE CWD_PROPERTY SET PROPERTY_VALUE = 'false' WHERE PROPERTY_NAME = 'secure.cookie' AND  
PROPERTY_KEY = 'crowd';
```

2. Start Crowd on the new server. You should be able to authenticate and access Crowd using the same credentials as on your original Crowd server.

Applications and Customizations

1. For any application you are going to test against this new Crowd server, you will need to modify the application's `crowd.properties` file to point to this new server, if the URL is different from the original server's URL.
2. If you have installed any [Crowd plugins](#) or added other customizations, you will need to re-apply them on the new server.

If you encounter any difficulties, please feel free to [contact support](#) and let us know which step you are having problems with.

Migrating from OnDemand to a Crowd installed site

You can extract your user data from a Confluence or Jira application installed site instance. Use these queries to export your users and memberships to CSV, and then import them into Crowd.

Migrate Jira Cloud or Confluence Cloud

First you'll need to migrate Jira/Confluence Cloud to your server versions:

[Migrate from Confluence Cloud to Server](#)

[Migrating from Jira Cloud to Server applications](#)

Exporting user data


If you are using Postgres as your database with Jira or Confluence, you can generate the required CSV files from either of those systems, with the following queries. If you are using a different database, you may need to tweak these.

To generate users.csv

```
// Run on JIRA/Confluence
copy (SELECT user_name AS "Username", first_name AS "First Name", last_name AS "Last Name", email_address
AS "Email Address", credential AS "Password" FROM cwd_user) to '/tmp/users.csv' csv header;
```

To generate group_memberships.csv

```
// Run on JIRA/Confluence
copy (SELECT DISTINCT child_name AS "Username", parent_name AS "Groupname" FROM cwd_membership) to '/tmp
/group_memberships.csv' csv header;
```

 Make sure that CSV files are encoded in UTF-8 (Unicode) to ensure compatibility for user data containing special or accented characters.

Retrieving the backup data

The files will be generated in /tmp, or in the folder specified in the query above.

What data is backed up

The backup includes the following data:

- User accounts
- Groups
- Memberships

Importing the data into Crowd

1. Follow the instructions in [Importing Users from CSV Files](#)
2. Connect Crowd to your applications: [Adding an Application](#)
3. Ensure that the Crowd Directory is placed above the Internal Directory in your User Directories in all downstream applications: [Configuring User Directories](#). This is important because you will have the same users in both directories, and the users will authenticate against the highest directory in the hierarchy.

Installing Crowd Data Center

To install Crowd Data Center, you'll create a cluster of Crowd instances that will make sure your users have uninterrupted access both to Crowd, and all other systems that are connected to it. We recommend that you start with the prerequisite information, listed on this page, to understand what Data Center is, and to know exactly what you'll need to complete the installation.

Before you begin

Before you install Crowd Data Center, you need to answer a few questions.


What is Crowd Data Center?	<p>You should first understand what Crowd Data Center is, and how it works. We've gathered some resources that will help you get to know the high-level overview of Data Center, and its architecture.</p> <p>Take a look at Crowd Data Center.</p>
How do I get it?	<p>You'll need two things to get started - an installer, and a license. There's no special installer for Data Center - you just install Crowd, and then enable Data Center features in it.</p> <p>As for the license, get your Crowd Data Center license, or try it out for free.</p>
What are the prerequisites?	<p>Supported platforms</p> <p>Supported operating systems, databases, etc., are the same as for the Server installation, and you can see them here: Supported platforms. You need to use an external database - HSQLDB is not supported.</p> <p>Node requirements</p> <p>Those specific to Data Center include requirements for nodes that create the cluster:</p> <ul style="list-style-type: none">• Each node is a separate machine (physical or virtual). They don't need to be identical, but should be as similar as possible for consistent performance.• All nodes are running the same version of Crowd. You'll be copying Crowd from one node to another, so this shouldn't be a problem.• They use the same timezone, and have the current time synced. You can use <code>ntpd</code> to set this up.• All nodes share a common database, also installed on a separate machine.• All nodes can access the shared home directory. You can set it up using NFS, or a similar solution. We'll mention it in this guide.

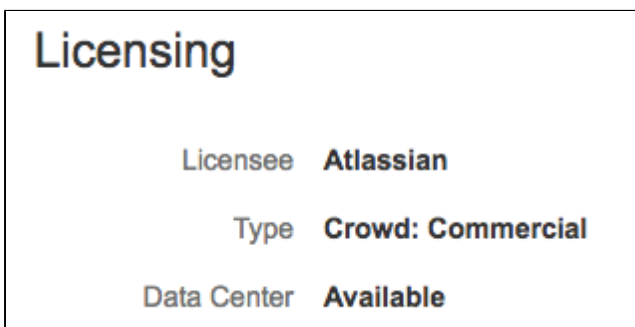
Do I need a load balancer?	<p>Yes. Crowd Data Center relies on a load balancer to balance the traffic between the nodes, and this guide assumes that you already have one set up. You can use a load balancer of your choice, just make sure it meets these requirements:</p> <ul style="list-style-type: none"> • Supports "cookie based session affinity", also known as "sticky sessions". • Can route HTTP/HTTPS traffic to one of the available nodes. • Can determine whether a node is available or not, and route requests to other nodes if needed. • All Atlassian applications and other REST clients must access your nodes through the load balancer. <p>Or you can just turn your proxy into a load balancer.</p> <p>Many bigger installations of Crowd already have a reverse proxy configured, and many reverse proxies can do load balancing as well. We've provided some examples on how to use your proxy as a load balancer. See Load balancer examples.</p>
----------------------------	--

1. Install Crowd

Crowd Data Center is available for Crowd 3.0, or later. If you're not on this version yet, install or upgrade your Crowd instance.

[Crowd installation and upgrade guide](#)

After you've installed Crowd and applied your license, you can verify that Data Center is available by going to  **Licensing**.



2. Set up the shared directory

You'll need to create a remote directory that is readable and writable by all nodes in the cluster. There are multiple ways to do this, but the simplest is to use an NFS share.

1. Stop Crowd.
2. Go to Crowd's home directory, and check whether it already has the `shared` sub-directory, which might have been created after starting Crowd. If it's there, you'll need to copy its contents to the new shared directory that you'll create in the next steps. If you can't find it, just omit this step.
3. Create a remote directory, accessible by all nodes in the cluster, and name it `shared`.
4. Mount `shared` as a sub-directory of the Crowd home directory.


```
<home-directory>/shared
```

3. Add the first Crowd node to your load balancer

Before you begin


You must enable clustering in Crowd first. To do that, in your shared directory edit the `crowd.cfg.xml` file and set the `crowd.clustering.enabled` property to `true` and restart Crowd.




The load balancer distributes the traffic between the nodes. If a node stops working, the remaining nodes will take over its workload, and your users won't even notice it.

1. First, add your load balancer as a trusted proxy server in Crowd. See [Configuring Trusted Proxy Servers](#).
2. Add the first node to the load balancer.
3. Restart the node, and then try opening different pages in Crowd. If the load balancer is working properly, you should have no problems with accessing Crowd.
4. In Crowd, go to  > **Clustering**. The node should be listed as part of the cluster.


Clustering

A cluster of multiple Crowd nodes provides high availability and performance at scale. [Learn more about clustering.](#)

Node ID	Node name	Cluster address	Hostname	Load	Memory	Uptime
 89eb4146-a...	Not config...	192.168.0.199	lpater-dev-2.local	52.64%	31.50%	31 seconds

   New nodes can join the cluster without downtime. [Learn how to add a node.](#)

✔ If your load balancer supports health checks, configure it to perform a check on `http://<crowd-node>:8095/<context-path>/status`, where `<crowd-node>` is the node's hostname or IP address, and `<context-path>` is the Crowd's context path (e.g. `/crowd`). If the node doesn't respond with a 200 OK response within a reasonable time, the load balancer shouldn't direct any traffic to this node.

5. After you've added the node to the load balancer, configure the Crowd's base URL to also point to the load balancer. Go to  > **General**, and enter the URL of your load balancer as *Base URL*.

General options

Deployment title*
The name of this Crowd instance.


Base URL*
The base URL for this installation of Crowd.

4. Add the remaining nodes to the cluster

1. Copy the Crowd installation directory to the new node.
2. Create a home directory, like you did for the first node, and mount `shared` as its sub-directory.
3. Edit `crowd-init.properties`, and enter the path to the home directory that you just created.



The `crowd-init.properties` file is in `<installation-directory>\crowd-webapp\WEB-INF\classes\`.



4. Go to `<installation-directory>/apache-tomcat/conf/Catalina/localhost`, and delete the `openidserver.xml` file. This is needed because currently the **CrowdID** component doesn't support clustering, and it must be enabled only on the first node. The component will work as usual.
5. Start Crowd. It will read the configuration from the shared directory, and start without any extra setup.
6. Take a look around the new Crowd instance. Verify that user and group management, directory synchronization, and any custom integrations all work as expected.

7. Again, verify that the node was added to the cluster. In Crowd, go to  **Clustering**.

Clustering

A cluster of multiple Crowd nodes provides high availability and performance at scale. [Learn more about clustering.](#)


Node ID	Node name	Cluster address	Hostname	Load	Memory	Uptime
 1d31a1fa-5...	node1	10.125.96.5	cdcdog-app1	0.00%	23.30%	4 weeks, 2 hours, 3...
 e6f7a3c8-1...	node2	10.125.96.27	cdcdog-app2	0.00%	27.03%	4 weeks, 2 hours, 36...

  New nodes can join the cluster without downtime. [Learn how to add a node.](#)

8. If everything looks fine, you can configure your load balancer to start routing traffic to the new node. Once you do this, you can make a couple of changes in one Crowd instance to see if they're visible in other instances as well.

What else?

Adding node names

When displaying information about your nodes in the Crowd footer or on the  **Clustering** page, Crowd Data Center uses random IDs that were generated for your nodes. Instead, you can give them more persistent and readable names by setting the `cluster.node.namesystem` property, like in the following example:

```
CATALINA_OPTS=-Dcluster.node.name=node-1
```

Well done! Crowd Data Center is now at your service.

Interested in learning more about what Crowd Data Center provides? [Click here](#) for an overview.

Migrate from Server to Data Center

Below is the process for migrating from Crowd Server to Crowd Data Center.

If you're installing Crowd for the first time and you don't have any existing Crowd data to migrate, see [Installing Crowd Data Center](#).

Before you begin

Before you install Crowd Data Center, you need to answer a few questions.


What is Crowd Data Center?	<p>You should first understand what Crowd Data Center is, and how it works. We've gathered some resources that will help you get to know the high-level overview of Data Center, and its architecture.</p> <p>Take a look at Crowd Data Center.</p>
How do I get it?	<p>You'll need two things to get started - an installer, and a license. There's no special installer for Data Center - you just install Crowd, and then enable Data Center features in it.</p> <p>As for the license, get your Crowd Data Center license, or try it out for free.</p>
What are the prerequisites?	<p>Supported platforms</p> <p>Supported operating systems, databases, etc., are the same as for the Server installation, and you can see them here: Supported platforms. You need to use an external database - HSQLDB is not supported.</p> <p>Node requirements</p> <p>Those specific to Data Center include requirements for nodes that create the cluster:</p> <ul style="list-style-type: none">• Each node is a separate machine (physical or virtual). They don't need to be identical, but should be as similar as possible for consistent performance.• All nodes are running the same version of Crowd. You'll be copying Crowd from one node to another, so this shouldn't be a problem.• They use the same timezone, and have the current time synced. You can use <code>ntpd</code> to set this up.• All nodes share a common database, also installed on a separate machine.• All nodes can access the shared home directory. You can set it up using NFS, or a similar solution. We'll mention it in this guide.

Do I need a load balancer?	<p>Yes. Crowd Data Center relies on a load balancer to balance the traffic between the nodes, and this guide assumes that you already have one set up. You can use a load balancer of your choice, just make sure it meets these requirements:</p> <ul style="list-style-type: none"> • Supports "cookie based session affinity", also known as "sticky sessions". • Can route HTTP/HTTPS traffic to one of the available nodes. • Can determine whether a node is available or not, and route requests to other nodes if needed. • All Atlassian applications and other REST clients must access your nodes through the load balancer. <p>Or you can just turn your proxy into a load balancer.</p> <p>Many bigger installations of Crowd already have a reverse proxy configured, and many reverse proxies can do load balancing as well. We've provided some examples on how to use your proxy as a load balancer. See Load balancer examples.</p>
----------------------------	--

1. Enable Crowd Data Center on your existing Crowd Server instance

Crowd Data Center is available for Crowd 3.0 and later. If you're not on this version yet, install or upgrade your Crowd instance. See [Crowd installation and upgrade guide](#).

Your Crowd license will determine whether you're running Crowd Data Center or Crowd Server. To run Crowd Data Center you need a Data Center license. You can purchase your Crowd Data Center license [here](#), or [try out free evaluation license](#).

1. Go to  > **Licensing** to enter your license key. Once successful you should see that Data Center is now available, but you need to restart before you can start using it.
2. Stop Crowd now. Before restarting it you will need to set up the shared home directory.

2. Set up the shared directory

Crowd Data Center requires that the `<CROWD_HOME>/shared` directory can be read and written by all the machines running Crowd Data Center.

When installing Crowd Server `<CROWD_HOME>/shared` is created as a normal directory. To use Crowd Data Center:

1. Stop Crowd.
2. Backup your Crowd home directory before making any changes.
3. Prepare a shared, network-accessible directory.

For this example we will assume the you are using Linux, and the shared directory is available at `/mnt/nfs/crowd`

4. Move `<CROWD_HOME>/shared` to the shared directory you've prepared:

```
mv <CROWD_HOME>/shared /mnt/nfs/crowd
```

5. Mount or create a symbolic link at `<CROWD_HOME>/shared` that points to the copied directory:


```
ln -s /mnt/nfs/crowd/shared <CROWD_HOME>/shared
```


6. Check if `<CROWD_HOME>/shared/crowd.cfg.xml` exists and is accessible from the machine running Crowd to verify you have configured the directory correctly.
7. Start Crowd again.

3. Add the first Crowd node to your load balancer

Before you begin


You must enable clustering in Crowd first. To do that, in your shared directory edit `thecrowd.cfg.xml` file and set the `crowd.clustering.enabled` property to `true` and restart Crowd.


The load balancer distributes the traffic between the nodes. If a node stops working, the remaining nodes will take over its workload, and your users won't even notice it.


1. First, add your load balancer as a trusted proxy server in Crowd. See [Configuring Trusted Proxy Servers](#).
2. Add the first node to the load balancer.
3. Restart the node, and then try opening different pages in Crowd. If the load balancer is working properly, you should have no problems with accessing Crowd.
4. In Crowd, go to  **Clustering**. The node should be listed as part of the cluster.


Clustering

A cluster of multiple Crowd nodes provides high availability and performance at scale. [Learn more about clustering.](#)

Node ID	Node name	Cluster address	Hostname	Load	Memory	Uptime
 89eb4146-a...	Not config...	192.168.0.199	lpater-dev-2.local	52.64%	31.50%	31 seconds

 New nodes can join the cluster without downtime. [Learn how to add a node.](#)

 If your load balancer supports health checks, configure it to perform a check on `http://<crowd-node>:8095/<context-path>/status`, where `<crowd-node>` is the node's hostname or IP address, and `<context-path>` is the Crowd's context path (e.g. `/crowd`). If the node doesn't respond with a 200 OK response within a reasonable time, the load balancer shouldn't direct any traffic to this node.

5. After you've added the node to the load balancer, configure the Crowd's base URL to also point to the load balancer. Go to  **General**, and enter the URL of your load balancer as *Base URL*.

General options


Deployment title*
The name of this Crowd instance.

Base URL*
The base URL for this installation of Crowd.

4. Add the remaining nodes to the cluster



1. Copy the Crowd installation directory to the new node.
2. Create a home directory, like you did for the first node, and mount `shared` as its sub-directory.
3. Edit `crowd-init.properties`, and enter the path to the home directory that you just created.



The `crowd-init.properties` file is in `<installation-directory>\crowd-webapp\WEB-INF\classes\`.

4. Go to `<installation-directory>/apache-tomcat/conf/Catalina/localhost`, and delete the `openidserver.xml` file. This is needed because currently the `CrowdID` component doesn't support clustering, and it must be enabled only on the first node. The component will work as usual.
5. Start Crowd. It will read the configuration from the shared directory, and start without any extra setup.
6. Take a look around the new Crowd instance. Verify that user and group management, directory synchronization, and any custom integrations all work as expected.
7. Again, verify that the node was added to the cluster. In Crowd, go to  **Clustering**.

Clustering

A cluster of multiple Crowd nodes provides high availability and performance at scale. [Learn more about clustering.](#)


Node ID	Node name	Cluster address	Hostname	Load	Memory	Uptime
 1d31a1fa-5...	node1	10.125.96.5	cdcdog-app1	0.00%	23.30%	4 weeks, 2 hours, 3...
 e6f7a3c8-1...	node2	10.125.96.27	cdcdog-app2	0.00%	27.03%	4 weeks, 2 hours, 36...

  New nodes can join the cluster without downtime. [Learn how to add a node.](#)

8. If everything looks fine, you can configure your load balancer to start routing traffic to the new node. Once you do this, you can make a couple of changes in one Crowd instance to see if they're visible in other instances as well.

What else?

Adding node names

When displaying information about your nodes in the Crowd footer or on the  **Clustering** page, Crowd Data Center uses random IDs that were generated for your nodes. Instead, you can give them more persistent and readable names by setting the `cluster.node.namesystem` property, like in the following example:

```
CATALINA_OPTS=-Dcluster.node.name=node-1
```

Crowd 4.4 Release Notes

15 October 2021

The Crowd team is proud to bring you **Crowd 4.4**.

Highlights

- [Sync users based on their access rights \(Data Center\)](#)
- [Log in with your email address \(Data Center\)](#)
- [Verify new email addresses](#)

More

Read the [upgrade notes](#) for important info about this release and see the [full list of issues](#) resolved.



[Get the latest version](#)

Sync users based on their access rights DATA CENTER

You can choose which users are synced with an application based on their access rights to it. This helps you limit the synced users to only those who can actually access the application, as syncing anyone else is redundant in most cases.

Directory	Who can authenticate	Automatically assigned to	Action
Atlassian Crowd server	1 GROUP	NO GROUPS	...
Azure AD	Add		

Access-based synchronization

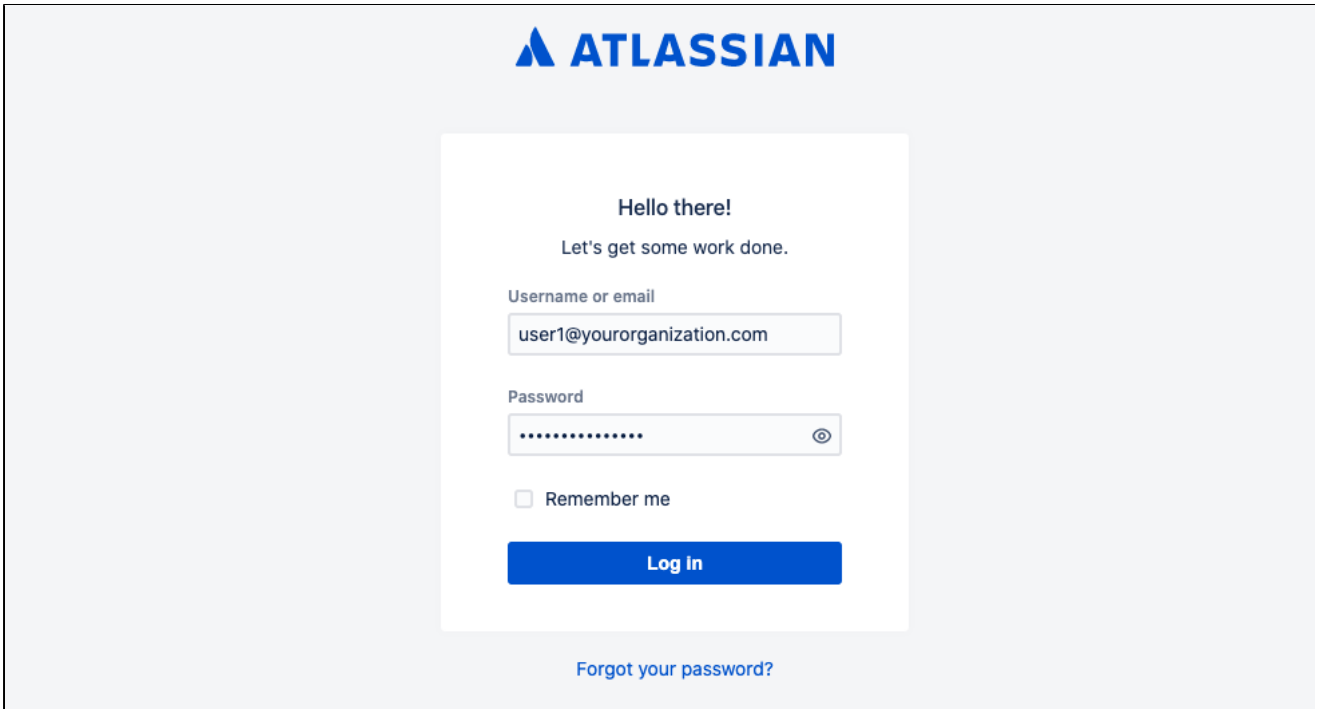
Choose which users and groups should be synchronized with this application:

- All users and groups
Sync all users and groups, regardless of their access rights, to keep the structure of your user directory. Choose this option if you're not sure about the remaining ones.
- All groups, but only users with access rights
Sync all groups so users with access rights can keep all of their group memberships, but filter out users without access rights.
- Only users and groups with access rights
Sync only users and groups that can access this application. These users will lose their memberships in groups that haven't been synced.

The options for access-based synchronization are available in the **Directories & groups** tab for each application. [Learn more](#)

Log in with your email address DATA CENTER

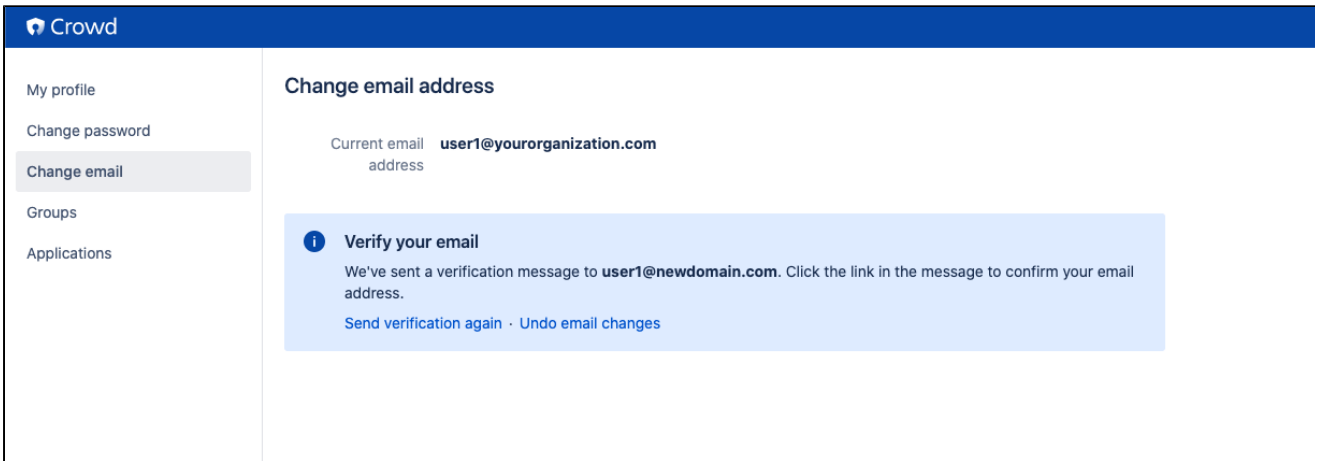
You can now log in to Crowd using your email address. This is to make things simpler for users who switch between different systems and are never sure whether they should use their login or email.



We've enabled the new option by default, but you can disable it in your Crowd settings. Also, make sure your user profile has a valid and unique email address configured. [Learn more](#)

Verify new email addresses

To improve security, we now send verification emails to users whenever they change their email address. We also send a confirmation message to their old address, just to keep them informed and make sure it's actually them making the change.



Complete list of changes and improvements

Here's a full list of issues resolved in this release:

Crowd 4.4.0 - 15 October 2021

T	Key	Summary
	CWD-5702	class java.time.LocalDateTime cannot be cast to class java.lang.String configuring MySQL 8 with Crowd
	CWD-5709	Crowd directory does not timeout when using SSL tunnel via forward proxy unless Connection Timeout is explicitly configured

[2 issues](#)

Administration Guide

The *Crowd Administration Guide* is for people who have [Crowd administration rights](#).

Table of Contents

- [Getting Started](#)
 - [Concepts](#)
 - [Supported Applications and Directories](#)
 - [About the Crowd Administration Console](#)
- [Managing Directories](#)
 - [Using the Directory Browser](#)
 - [Adding a Directory](#)
 - [Configuring an Internal Directory](#)
 - [Configuring an LDAP Directory Connector](#)
 - [LDAP Object Structures](#)
 - [Configuring an SSL Certificate for Microsoft Active Directory](#)
 - [Configuring a Remote Crowd Directory](#)
 - [Configuring a Custom Directory Connector](#)
 - [Configuring a Delegated Authentication Directory](#)
 - [Configuring Azure Active Directory](#)
 - [Configuring Caching for an LDAP Directory](#)
 - [Using Naive DN Matching](#)
 - [Specifying Directory Permissions](#)
 - [Importing Users and Groups into a Directory](#)
 - [Importing Users from Atlassian Confluence](#)
 - [Importing Users from Atlassian Jira](#)
 - [Importing Users from Atlassian Bamboo](#)
 - [Importing Users from Jive Forums](#)
 - [Importing Users from CSV Files](#)
 - [Configuring the CSV Importer](#)
 - [Mapping CSV Fields to Crowd Fields](#)
 - [Confirming the CSV Importer Configuration](#)
 - [Viewing the Results of the Import](#)
 - [Importing Users from One Crowd Directory into Another](#)
 - [Configuring directories for failover authentication](#)
 - [Pruning delegated directories](#)
- [Managing Applications](#)
 - [Using the Application Browser](#)
 - [Adding an Application](#)
 - [Integrating Crowd with Atlassian Bamboo](#)
 - [Integrating Crowd with Atlassian Confluence](#)
 - [Integrating Crowd with Atlassian Confluence 3.4 or earlier](#)
 - [Updating Files in a Confluence Evaluation Distribution](#)
 - [Integrating Crowd with Atlassian CrowdID](#)
 - [Integrating Crowd with Atlassian Crucible](#)
 - [Integrating Crowd with Atlassian FishEye](#)
 - [Configuring FishEye earlier than 4.0 with Crowd](#)
 - [Integrating Crowd with Atlassian Jira](#)
 - [Integrating Crowd with Atlassian Jira 4.2 or earlier](#)
 - [Integrating Crowd with Atlassian Bitbucket Server](#)
 - [Integrating Crowd with Acegi Security](#)
 - [Integrating AppFuse - a Crowd-Acegi Integration Tutorial](#)
 - [Integrating Crowd with Jive Forums](#)
 - [Jive SSO](#)
 - [Integrating Crowd with Spring Security](#)
 - [Integrating AppFuse - a Crowd-Spring Security Integration Tutorial](#)
 - [Integrating Crowd with a Custom Application](#)
 - [Integrating Crowd with Atlassian HipChat](#)
 - [Configuring the Google Apps Connector](#)
 - [Mapping a Directory to an Application](#)
 - [Specifying the Directory Order for an Application](#)
 - [Specifying an Application's Directory Permissions](#)

- Example of Directory Permissions
 - Viewing Users in Directories Mapped to an Application
 - Specifying which Groups can access an Application
 - Syncing users based on their access rights
- Effective memberships with multiple directories
- Specifying an Application's Address or Hostname
- Testing a User's Login to an Application
- Enforcing Lower-Case Usernames and Groups for an Application
- Managing an Application's Session
- Deleting or Deactivating an Application
- Configuring Caching for an Application
- Overview of SSO
- Configuring Options for an Application
- Enabling OpenID client app
- Allowing applications to create user tokens
- Disabling the OpenID client app
- Configuring how users log in
- Managing Users and Groups
 - Using the User Browser
 - Adding a User
 - Editing a User's Details and Password
 - Deleting or Deactivating a User
 - Case Sensitivity of Usernames and Groups
 - Specifying a User's Aliases
 - Editing a User's Group Membership
 - Managing Groups
 - Deleting a Group
 - Adding a Group
 - Managing Group Members
 - Automatically Assigning Users to Groups
 - Adding Users to a Group
 - Removing Users from a Group
 - Nested Groups in Crowd
 - Adding a Sub-Group
 - Group-level administration
 - Adding Group Level Admins
 - Removing Group Level Admins
 - Removing a Sub-Group
 - Specifying a User's Attributes
 - Granting Crowd Administration Rights to a User
 - Granting Crowd User Rights to a User
 - Managing a User's Session
- System Administration
 - Configuring Server Settings
 - Deployment Title
 - Domain
 - Session configuration
 - Authorization Caching
 - Licensing
 - Crowd SSO 2.0
 - Finding your SEN
 - SSO Cookie
 - Configuring your Mail Server
 - Creating an Email Notification Template
 - Configuring Trusted Proxy Servers
 - Viewing Crowd's System Information
 - Backing Up and Restoring Data
 - Logging and Profiling
 - Performance Profiling
 - Draft - Troubleshooting and Requesting Technical Support
 - Configuring the LDAP Connection Pool
 - Browsing the audit log
 - Look and feel
 - Overview of Caching

- [Crowd Security Advisories and Fixes](#)
 - [Crowd Security Advisory 2010-07-05](#)
 - [Crowd Security Advisory 2010-05-04](#)
 - [Crowd Security Advisory 2008-10-14 - Parameter Injection Vulnerability](#)
 - [Crowd Security Advisory 2012-05-17](#)
 - [Crowd Security Notice 2013-07-01](#)
 - [Crowd Security Advisory 2013-07-16](#)
 - [Crowd Security Advisory 2014-05-21](#)
 - [Crowd Security Advisory 2016-10-19](#)
 - [Crowd Security Advisory 2017-03-10](#)
 - [Crowd Security Advisory 2019-05-22](#)
- [Constructing cron expressions in Crowd](#)

Getting Started

- [Concepts](#)
- [Supported Applications and Directories](#)
- [About the Crowd Administration Console](#)

Concepts

Crowd is an application security framework that handles authentication and authorization for your web-based applications. With Crowd you can quickly integrate multiple web applications into a single security architecture that supports single sign-on (SSO) and centralized identity management.

Crowd has the following components:

- The **Crowd Administration Console** is a clean and powerful web-interface for managing directories, users (known in Crowd as 'principals') and their security rights ('permissions'). Refer to the [Crowd Administration Guide](#) for details.
- The **Crowd Self-Service Console** allows authorized users to maintain their user profiles and passwords and to view their usernames, groups, roles and applications. Refer to the [Crowd User Guide](#) for details.
- The **Crowd integration API** provides a platform-neutral way to integrate web applications into a single security architecture. With the [integration API](#), applications can quickly access user information and perform security checks.

Designed for ease of use, Crowd can be deployed with your existing infrastructure. Crowd supports:

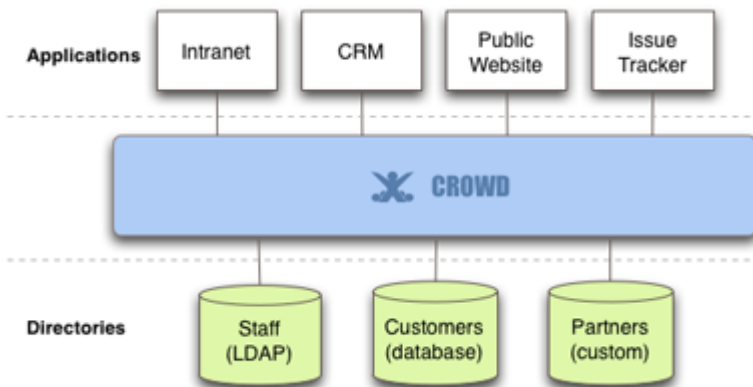
- Java, .NET and PHP [applications](#).
- Popular [directory servers](#) such as Microsoft Active Directory, Sun ONE and OpenLDAP. Additionally, [custom directory connectors](#) may be developed using the Crowd integration API.

See the [list](#) of supported applications and directories.

Architectural Overview

Crowd is a middleware application that integrates web applications into a single security architecture, supporting single sign-on and centralized identity management. Crowd works by dispatching authentication and authorization calls from configured applications to configured directories.

A typical deployment may be similar to the following:



When an application needs to validate a security or authentication request (e.g. when a user attempts to log in to the application) the application will make a simple API call to the Crowd framework, which will then forward the call to the appropriate directory.

About Applications

Crowd integrates and provisions applications. Once [defined](#), an application is [mapped](#) to a directory(s), whose users are then [granted access](#) to the application. Note that an application can only communicate with Crowd when the application uses a known [host address](#).

About Directories

Crowd supports an unlimited number of user directories. A directory can be one of the following types:

- Internal to Crowd.
- Connected to Crowd via an LDAP connector (e.g. for Active Directory), with all authentication and user /group management in LDAP.

- A Crowd internal directory for user/group management but with authentication delegated to LDAP (e.g. Active Directory).
- Connected via a custom directory connector (e.g. for a legacy database).

Once you have [defined](#) a directory in Crowd, you can [map](#) it to applications. Crowd will then pass authentication and authorization requests to the directory, for all applications that are mapped to that directory. Modification of directory entities ([users and groups](#)) can be done via the Crowd Administration Console or via the application, depending on the application's capabilities.

You can even map multiple directories to an application, providing the application with a single view of multiple directories in a specified [order](#).

Supported Applications and Directories

Crowd integrates and provisions applications. Once defined, an application is mapped to one or more directories, whose users are then granted access to the application. This page lists the supported application and directory connectors.

Application Connectors

- [Atlassian Jira](#)
- [Atlassian Confluence](#)
- [Atlassian Bitbucket Server](#)
- [Atlassian Bamboo](#)
- [Atlassian FishEye](#)
- [Atlassian Crucible](#)
- [Google Apps](#)
- [Jive Forums](#)
- [Atlassian CrowdID](#)
- [Acegi](#)
- [Spring Security](#)

You can also add your own [custom applications](#).

Directory Connectors

[Connecting to LDAP directories](#)

Using Crowd's internal directories:

- [Internal Crowd Directory](#)
- [Delegated Authentication Directory](#), combining the features of an internal Crowd directory with delegated LDAP authentication.

You can also add a connector to your own [custom directory](#).

RELATED TOPICS

[Concepts](#)
[Adding an Application](#)
[Adding a Directory](#)
[Crowd documentation](#)

About the Crowd Administration Console

The **Crowd Administration Console** presents the full range of Crowd administration functionality to authorized [Crowd administrators](#).

[Authorized Crowd users](#) who are **not** administrators can also access the Crowd Console. They will see a subset of functionality, which we call the '**Self-Service Console**'. Refer to the [Crowd User Guide](#) for details.

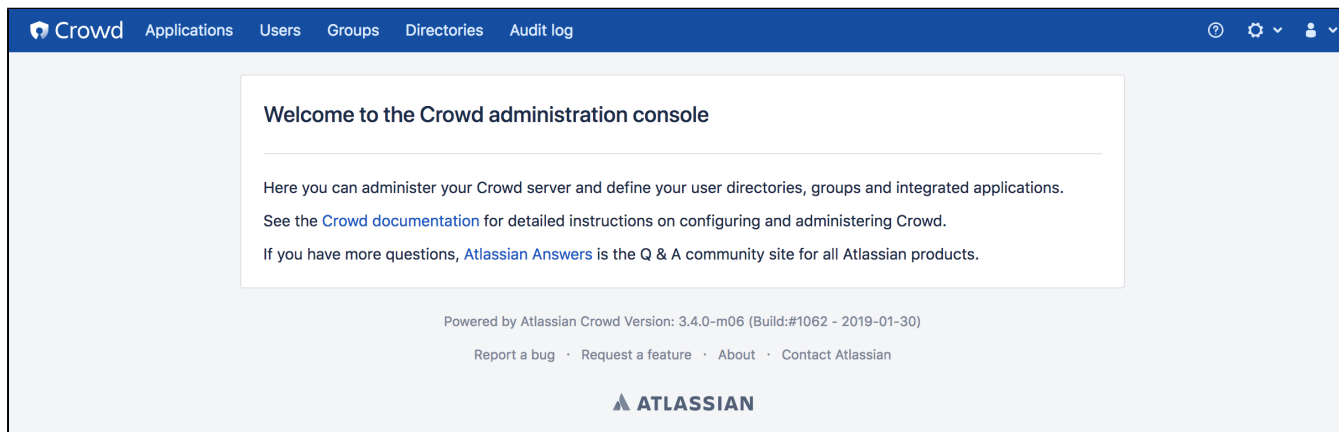
If you are a [Crowd administrator](#), the Crowd Administration Console allows you to perform the following functions:


- Configure [applications](#) to access the Crowd framework.
- Create and manage [users](#) and adjust their group membership.
- Map [directories](#) to allow users to access integrated applications.
- Adjust [server deployment properties](#), including those configured during the setup process.
- [Back up and restore](#) your Crowd data.
- View active [sessions](#) and manually expire sessions.
- View Crowd [system information](#).
- Update your user profile and password and view the groups and applications associated with your username. Refer to the [User Guide](#) for details.

To access the Crowd Administration Console:

1. Go to the URL <http://localhost:8095/crowd> or <http://localhost:8095/crowd/console>.

The welcome screen will appear, looking something like this:



 The Crowd Administration Console is a web application provisioned by Crowd you can see it in the list of applications shown in the [Application Browser](#).

Please refer to the link below in order to grant administrators rights to Crowd user(s):

[Granting Crowd Administration Rights to a User](#)

Managing Directories

Crowd supports an unlimited number of user directories. A directory can be one of the following types:

- Internal to Crowd.
- Connected to Crowd via an LDAP connector (e.g. for Active Directory), with all authentication and user /group management in LDAP.
- A Crowd internal directory for user/group management but with authentication delegated to LDAP (e.g. Active Directory).
- Connected via a custom directory connector (e.g. for a legacy database).

Once you have [defined](#) a directory in Crowd, you can [map](#) it to applications. Crowd will then pass authentication and authorization requests to the directory, for all applications that are mapped to that directory. Modification of directory entities ([users and groups](#)) can be done via the Crowd Administration Console or via the application, depending on the application's capabilities.

You can even map multiple directories to an application, providing the application with a single view of multiple directories in a specified [order](#).

Using the Directory Browser

About Directories

Crowd supports an unlimited number of user directories. A directory can be one of the following types:

- Internal to Crowd.
- Connected to Crowd via an LDAP connector (e.g. for Active Directory), with all authentication and user /group management in LDAP.
- A Crowd internal directory for user/group management but with authentication delegated to LDAP (e.g. Active Directory).
- Connected via a custom directory connector (e.g. for a legacy database).

Once you have [defined](#) a directory in Crowd, you can [map](#) it to applications. Crowd will then pass authentication and authorization requests to the directory, for all applications that are mapped to that directory. Modification of directory entities ([users and groups](#)) can be done via the Crowd Administration Console or via the application, depending on the application's capabilities.

You can even map multiple directories to an application, providing the application with a single view of multiple directories in a specified [order](#).

About the Directory Browser

The Directory Browser allows you to view and search for configured directories.

To use the Directory Browser

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click **Directories**.
This displays the Directory Browser which shows all the directories that exist in your Crowd system. You can refine your search by specifying a **Name** (note that this is case-sensitive), or **Active/Inactive** directories.
i An Inactive directory cannot be used by any applications, regardless of whether or not they are [mapped](#) to it.
3. To view or edit a directory's details, click on the directory's name

You created one default directory when you [set up Crowd](#). To add more directories, see [Adding a Directory](#)

Screenshot: 'Directory Browser'

Directories				
Search		Active	Results per page	
<input type="text" value="Name"/>	<input type="text" value="All"/>	<input type="text" value="100"/>	<input type="button" value="Search"/>	<input type="button" value="Reset"/>
Name	Active	Type	Action	
Crowd	true	Crowd Internal Directory		
Atlassian AD	true	Microsoft Active Directory	Synchronise	
Staff ID	true	Crowd Internal Directory		
Atlassian Staff ID	false	OpenLDAP (Read-Only Posix Schema)	Synchronise	

Adding a Directory

Directories contain authentication and authorization information about users, groups and roles. Crowd supports an unlimited number of directories. Administrators can use different directories to create silos of users. For example, you might store your customers in one directory and your employees in another.

Crowd supports the following types of directory:

- [Crowd Internal Directory](#)
Internal directories use the Crowd database to store user, group and role information. Internal directories are stored in Crowd's [database server](#).
- [Delegated Authentication Directory](#)
A Delegated Authentication directory combines the features of an internal Crowd directory with delegated LDAP authentication. This means that you can have your users authenticated via an external LDAP directory while managing the users and groups in Crowd. You can use Crowd's flexible and simple group management when the LDAP groups do not suit your requirements. Alternatively, you can have Crowd import users' group memberships from LDAP each time they authenticate.
- [LDAP Directory Connector](#)
- [Remote Crowd Directory Connector](#)
Remote Crowd directories allow Crowd to Crowd connections. In other words, one Crowd server can obtain users and groups from another Crowd server.
- [Custom Directory Connector](#)
Custom directory connectors allow developers to connect Crowd to custom user-stores, such as existing databases or legacy systems.

You can add as many directories of each type as you need.

To add a directory,

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click **Directories**.
3. Click **Add Directory**.
This will display the **Select Directory Type** screen (*screenshot below*).
4. Select the type of directory you want to add:
 - **'Internal'** see [Configuring an Internal Directory](#)
 - **'Delegated Authentication'** see [Configuring a Delegated Authentication Directory](#)
 - **'Connector'** see [Configuring an LDAP Directory Connector](#) (e.g. Microsoft Active Directory)
 - **'Remote Crowd'** see [Configuring a Remote Crowd Directory](#)
 - **'Custom'** see [Configuring a Custom Directory Connector](#)

i Once a directory has been configured, you will need to specify [permissions](#) for its users. You can then [map](#) the directory to appropriate applications.

Screenshot: 'Select Directory Type'

Select directory type

Internal

Internal directories store authentication and authorisation information in the Crowd database.

Delegated authentication

Delegated authentication directories store users and groups within Crowd and delegate authentication to an external LDAP directory.

Connector

Crowd ships with several LDAP connectors, such as Active Directory, Apache Directory Server, Sun ONE/DSEE and OpenLDAP.

Remote Crowd

Crowd can connect to remote Crowd directories.

Azure Active Directory

With Azure Active Directory, you can integrate Crowd with your users and groups from Cloud.

Custom

Custom directories allow developers to implement an interface to connect custom user stores such as existing databases.

Next

Configuring an Internal Directory

Internal directories use the Crowd database to store user, group and role information. Internal directories are stored in Crowd's [database server](#).

To configure an internal directory,

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click **Directories**.
This will display the [Directory Browser](#).
3. In the left-hand menu, click **Add directory**.
4. Select **internal** as the directory type and click **Next**.
5. Complete the fields as described in the table below.
6. Click the **Continue** button to configure the directory's [permissions](#).

i Once you have configured the directory's permissions, you will have finished configuring your new directory. You can then [map](#) the directory to appropriate applications.

Screenshot: Create internal directory

Create internal directory

[Details](#) [Permissions](#)

Name
A short, recognisable name that characterises this user directory. For example: "Chicago employees" or "Web customers".

Description
More information about this directory.

Active
If the directory is marked inactive, users in that directory will be unable to access Crowd or any Crowd-connected applications

Password regex
Regular expression pattern which new passwords will be validated against. Leave blank to disable this feature.

Password complexity requirement message
Message explaining the password complexity requirements for the directory.

Maximum password attempts
The maximum number of invalid password attempts before the authenticating account will be disabled. Enter 0 to disable this feature.

Password history count
The number of previous passwords to check when disallowing repeated passwords on password change. Enter 0 to allow password repeats.

Days until password expiry
The number of days until the password must be changed. Enter 0 to disable password expiry.

Notify the user of password expiry in
The number of days left before password expiration when you want send a password change reminder to a user. You can use multiple comma-separated values e.g. 14,7; Leave the field empty to disable notifications.

Password encryption
For compatibility between Atlassian products you must use ATLASSIAN-SECURITY.

Use nested groups
This will enable nested group support for the directory.

[Continue](#) [Cancel](#)

Internal Directory Attributes	Description
Name	The name used to identify the directory within Crowd. This is useful when there are multiple directories configured, e.g. Chicago Employees or Web Customers.

Description	Details about this specific directory.
Active	<p>Only deselect this if you wish to prevent all users within the directory from accessing all mapped applications. If a directory is not marked as 'Active', it is inactive. Inactive directories:</p> <ul style="list-style-type: none"> • are not included when searching for users, groups or memberships. • are still displayed in the Crowd Administration Console screens.
Password Regex	<p>Regex pattern which new passwords will be validated against. The regular expression format used is the java.util.regex.Pattern. For example, for an alphanumeric password of at least 8 characters, you could use the pattern:</p> <pre>[A-Za-z0-9]{8,}</pre> <p>Leave blank to disable this feature.</p>
Password Complexity Message	A message shown when a user is resetting a password to explain custom complexity requirements set with Password Regex (since Crowd 2.5.2).
Maximum Invalid Password Attempts	The maximum number of invalid password attempts before the authenticating account will be locked. Enter 0 to disable this feature.
Maximum Unchanged Password Days	The number of days until the password must be changed. This value is in days, enter 0 to disable this feature.
Password History Count	The number of previous passwords to prevent the user from using. Enter 0 to disable this feature.
Password Encryption	If you wish to import users into this directory from another Atlassian product, specify ' ATLAS SIAN-SECURITY ' in order to ensure password compatibility.
Use Nested Groups	Enable or disable support for nested groups on the internal user directory.

Next Step

See [Specifying Directory Permissions](#).

Configuring an LDAP Directory Connector

Crowd provides built-in connectors for the most popular LDAP directory servers:

- Apache Directory Server (ApacheDS)
- Apple Open Directory
- Fedora Directory Server
- Generic LDAP Directories
- Microsoft Active Directory
- Novell eDirectory
- OpenDS
- OpenLDAP
- OpenLDAP Using Posix Schema
- Posix Schema for LDAP
- Sun Directory Server Enterprise Edition (DSEE)

Before you begin configuring the directory, check for any [directory-specific notes](#) that affect the directory type you're using.

Configuring an LDAP directory connector

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click **Directories**.
The [Directory Browser](#) opens.
3. Click **Add Directory**.
4. Select **Connector**.
5. Complete the configuration information required on each of the tabs to finish setting up the connector and click

General configuration notes

- By default, the **Cache Enabled** setting on the 'Details' tab is selected. We recommend you leave this setting selected. For more information, see [Configuring Caching for an LDAP Directory](#).
- If you select the **Manage Groups Locally** setting on the 'Connector' tab (available only if you've selected the **Cache Enabled** check box), new groups are created and updated in the Crowd database and not propagated to the LDAP server. Memberships of local groups are also stored locally. This makes it possible to augment the group structure with new groups even with a read-only LDAP server. When this option is enabled, only local groups can be created and updated, while groups synchronized from the remote directory cannot be locally modified.
- If you select the **Use the User Membership** setting on the 'Connector' tab, Crowd will use the group membership attribute on the user when it retrieves the members of a given group, which will result in a more efficient retrieval.
- If you select the **Use 'memberOf' for Group Membership** setting on the 'Connector' tab, Crowd will use the 'memberOf' attribute when it retrieves the list of groups a users belongs to, which will result in a more efficient retrieval. If you don't select this setting, Crowd will use the members attribute on the group ('member' by default) for the search.
- Crowd will synchronize user renames made in the LDAP server, provided that the **User Unique Identifier Attribute** is set in the 'Configuration' tab. If this attribute is not set and a user is renamed in the LDAP server, Crowd will not be able to track the user's identity, and will delete the user with the old name and create a new user with the new name. Crowd does not support group renames.
- If the directory type you're using guarantees the format of DNs, we recommend selecting the **Use Naive DN Matching** setting on the 'Connector' tab to allow Crowd to do a direct, case-insensitive, string comparison when it compares DNs. This setting can significantly improve performance.
- Specify the **Username** on the 'Connector' tab in the following format: `cn-administrator, cn=users, dc=ad, dc=acmecorp, dc=com`.
- If you specify the **User Name RDN attribute**, the DN for each LDAP entry is composed of two parts: the RDN and the location within the RDN directory where the record resides. The RDN is the portion of your DN that is not related to the directory tree structure.

- By default the **Synchronise group memberships when logging in** option is set to *For newly added users only*. This will synchronize group memberships for users who have been created in the LDAP directory, but not yet synchronized to Crowd. This is recommended for convenience, without sacrificing performance. Other options are to synchronize the memberships *Every time a user logs in*, which was the behaviour in Crowd 2.7, 2.8 and 2.9, and to *Never* synchronise the memberships, which was how Crowd behaved before version 2.7.
- If you are connecting to the LDAP directory as a user affected by query limits (for example using a DN that is not a RootDN in OpenLDAP, with `olcSizeLimit` set) some operations might not return all results. Currently it is recommended to connect as a user that is unaffected by limits.
- If you have successfully added your connector, but aren't able to see any data when you browse the LDAP directory, make sure that any non-standard object types and filters are [configured correctly](#).



By default, the **Active** setting on the 'Details' tab is selected. Only clear this setting if you want to prevent all users within the directory from accessing [mapped applications](#). Inactive directories:

- Are not included when Crowd searches for users, groups, or memberships
- Still appear in the Crowd Administration Console screens

You can also configure site-wide LDAP connection pool settings. See [Configuring the LDAP Connection Pool](#).

Directory-specific configuration notes

Apache Directory Server (ApacheDS)

- There are two known issues with ApacheDS and Crowd:
 - ApacheDS 1.0.2 does not support password resets without a restart. This is an ApacheDS limitation.
 - ApacheDS does not support paged results. [CWD-1109: Cannot browse users or groups if Use Paged Results is enabled](#). Again, this is an ApacheDS limitation.

Apple Open Directory

- Crowd's Apple Open Directory support is read-only. You cannot add or update user details or group details in a Crowd-connected OS X Open Directory server. Users will not be able to change their passwords from Crowd or from Crowd-connected applications.
- Crowd will check both the `gidNumber` and the `memberUid` attributes to determine if a user is a member of a group. The name of the `gidNumber` attribute is not configurable. Crowd will always use this attribute to determine membership.
- The [RFC 2307 schema](#) does not support nesting of groups, so Crowd does not support nested groups in Apple Open Directory.

Fedora Directory Server

- Crowd supports read-only connections to Fedora DS using the Posix/NIS schema [RFC 2307](#). You cannot add or update user details or group details in a Crowd-connected Fedora Directory server. Users will not be able to change their passwords from Crowd or from Crowd-connected applications.
- Crowd will check both the `gidNumber` and the `memberUid` attributes to determine if a user is a member of a group. The name of the `gidNumber` attribute is not configurable. Crowd will always use this attribute to determine membership.
- The [RFC 2307 schema](#) does not support nesting of groups, so Crowd does not support nested groups in Fedora DS.

Microsoft Active Directory

- If you want to use a secure SSL connection, make sure you [configure an SSL Certificate](#) before enabling this setting.
- We recommend selecting the Enable Incremental Sync setting to allow Crowd to retrieve changes made after the last synchronization when possible.

- Specify the **Base DN** in the following format: `dc=domain1,dc=local`. You will need to replace the `domain1` and `local` for your specific configuration. Microsoft Server provides a tool called `ldp.exe` which is useful for finding out and configuring the the LDAP structure of your server.
- If you want to use Crowd to add users or change passwords in Microsoft Active Directory, you will need to install an SSL certificated generated by your Active Directory server and then install the certificate into your JVM keystore. Please read the instructions: [Configuring an SSL Certificate for Microsoft Active Directory](#).
- Crowd will synchronize the user status with Active Directory. If a user account is disabled in Active Directory, the user will be deactivated in Crowd, and reciprocally, if a user is deactivated in Crowd, the user account will be disabled in Active Directory. To prevent this synchronization, use **Manage User Status Locally** in the 'Connector' tab.
- Users' primary groups in Active Directory will be displayed as regular memberships in Crowd. However, you will not be able to change or remove the user's primary group through Crowd's user interface.
- If you are using a single Active Directory domain, you should disable "Use node referrals" in the directory configuration. If you have a forest, you should read [User lookup fails with PartialResultException in Jira server](#) and ensure your DNS server is configured appropriately.
- We have not tested Crowd integration with Active Directory Application Mode ([ADAM](#)). However, ADAM and Active Directory share the same code base, LDAP interface and API. So ADAM should work with Crowd, following the same integration instructions as above. If you try it, we'd be interested to hear of your experiences.
- Crowd's **Filter out expired users** feature requires an LDAP connection that exposes the `accountExpires` attribute. Care should be taken when connecting to the Active Directory Global Catalog as it does not replicate the aforementioned attribute by default. This may cause inconsistent user status in Crowd.

Posix Schema for LDAP or Open LDAP

- Currently, Crowd supports read-only access to the directory based on the Posix schema. You cannot add or update user details. Crowd supports read-only connections to an LDAP directory using the Posix/NIS schema. This is useful if you have a Unix installation and want to integrate with an LDAP directory. The Posix/NIS schema allows integration between an LDAP directory and the Unix NIS (Network Information Service).
- Crowd will check both the `gidNumber` and the `memberUid` attributes to determine if a user is a member of a group. The name of the `gidNumber` attribute is not configurable. Crowd will always use this attribute to determine membership.
- The [RFC 2307 schema](#) does not support nesting of groups, so Crowd does not support nested groups in the Posix schema.

LDAP Object Structures

The Crowd LDAP connectors assume that all container objects (groups) have the full DN to the associated member. Currently the membership attributes on a User object are not used by Crowd.

Supported Object Types

- groupOfUniqueNames
- inetorgperson
- posixGroup
- posixUser
- zimbraAccount



Microsoft Active Directory

The Active Directory LDAP connector assumes that all LDAP object types are of the default structure. Any changes to the default object structure of the `User` and `Group` objects will require a [custom connector](#) to be coded.

Supported Attributes

Crowd's LDAP connectors support the adding and updating of the following user attributes when integrating with an LDAP server via an LDAP directory connector:

- surname
- given name
- email
- password

If you need support for additional LDAP attributes, the Crowd LDAP connector can be extended. With a license purchase, full source is available and the LDAP connectors can be modified to support any number of attributes.



To help you identify your LDAP structure, you may find an LDAP browser useful. Take a look at our guide on [using Apache Directory Studio](#).

Configuring an SSL Certificate for Microsoft Active Directory

You can configure Crowd to work with Microsoft Active Directory by setting up an [LDAP connector](#) in Crowd. If you wish to use Crowd to add users or change passwords in Active Directory, you will need to install an SSL certificate generated by your Active Directory server and then install the certificate into your JVM keystore.

On this page:

- [Prerequisites](#)
- [Step 1. Install the Active Directory Certificate Services](#)
- [Step 2. Obtain the Server Certificate](#)
- [Step 3. Import the Server Certificate](#)

Updating user, group, and membership details in Active Directory requires that your Atlassian application be running in a JVM that trusts the AD server. To do this, we generate a certificate on the Active Directory server, then import it into Java's `keystore`.

Prerequisites

To generate a certificate, you need the following components installed on the Windows Domain Controller to which you're connecting.

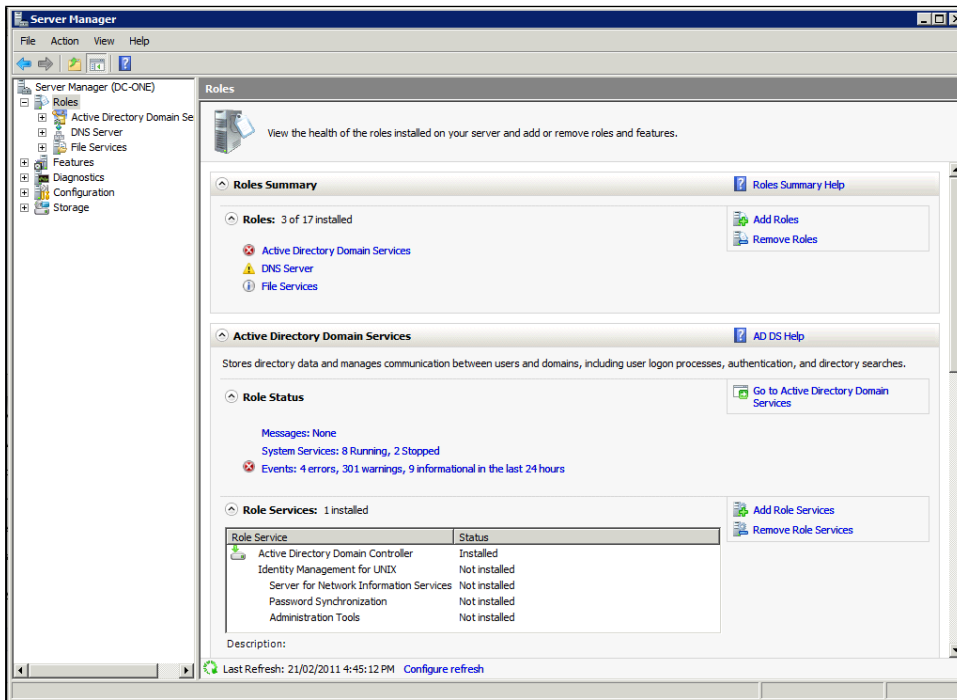
Required Component	Description
Internet Information Services (IIS)	This is required before you can install Windows Certificate Services.
Windows Certificate Services	This installs a certification authority (CA) which is used to issue certificates. Step 1, below, explains this process.
Windows 2000 Service Pack 2	Required if you are using Windows 2000
Windows 2000 High Encryption Pack (128-bit)	Required if you are using Windows 2000. Provides the highest available encryption level (128-bit).

Step 1. Install the Active Directory Certificate Services

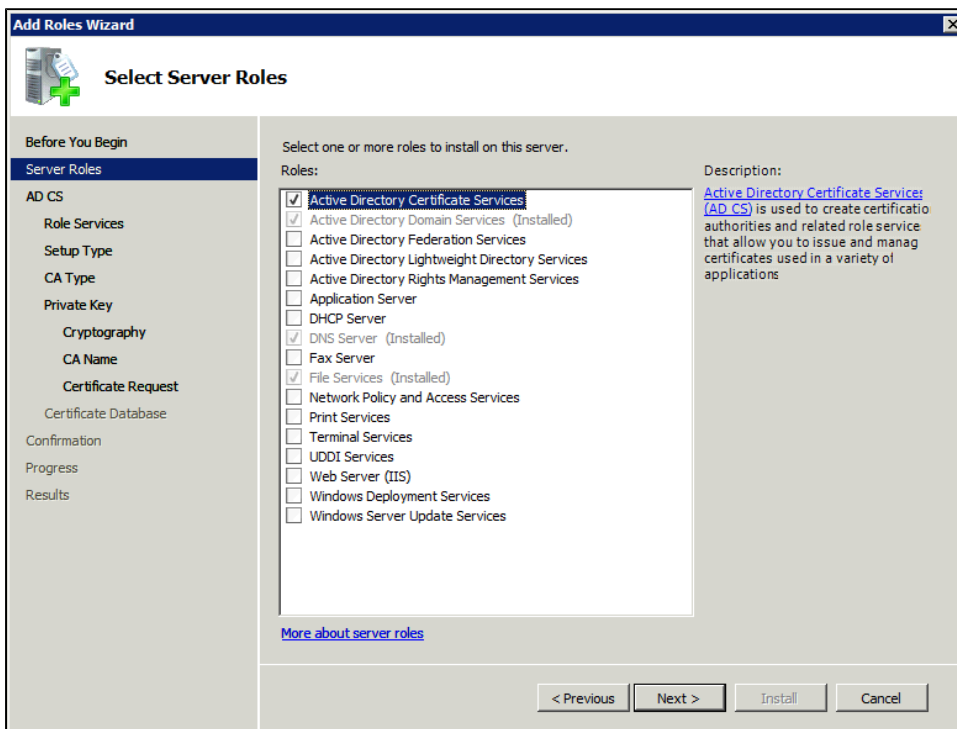
If Certificate Services are already installed, skip to step 2, below. The screenshots below are from Server 2008, but the process is similar for Server 2000 and 2003.

1. Log in to your Active Directory server as an administrator.
2. Click **Start**, point to **Administrative Tools**, and then click **Server Manager**.

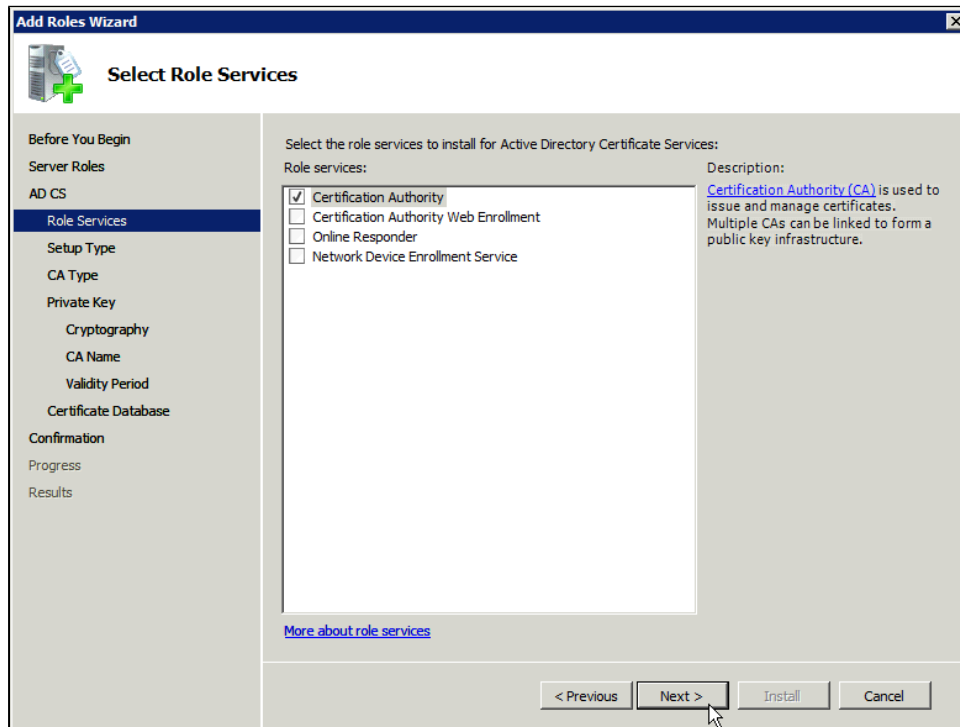
3. In the **Roles Summary** section, click **Add Roles**.



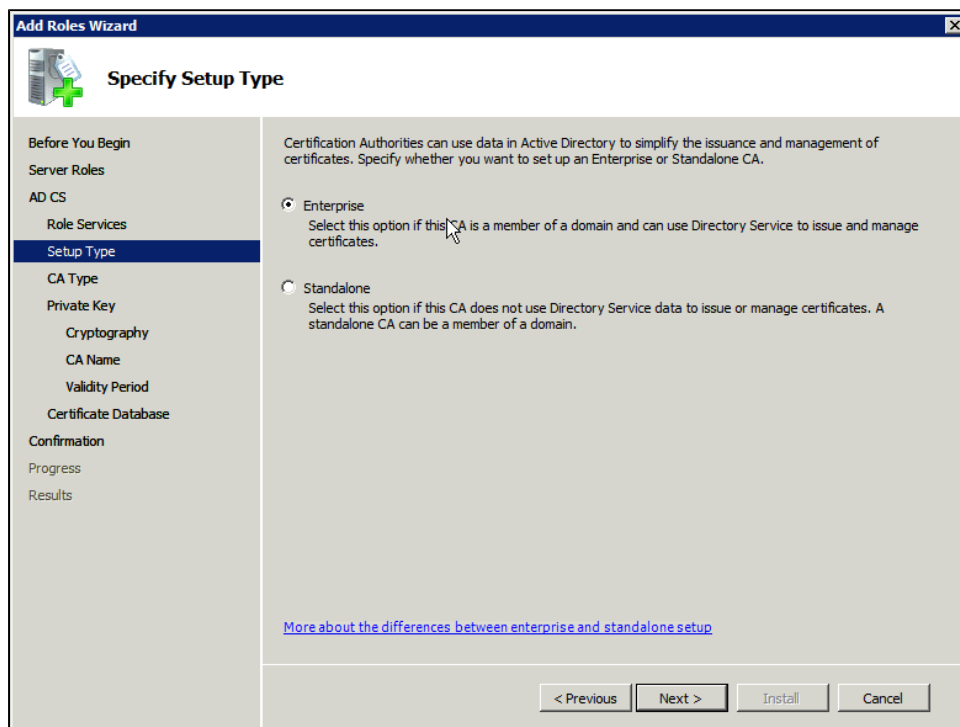
4. On the **Select Server Roles** page, select the **Active Directory Certificate Services** check box. Click **Next** twice.



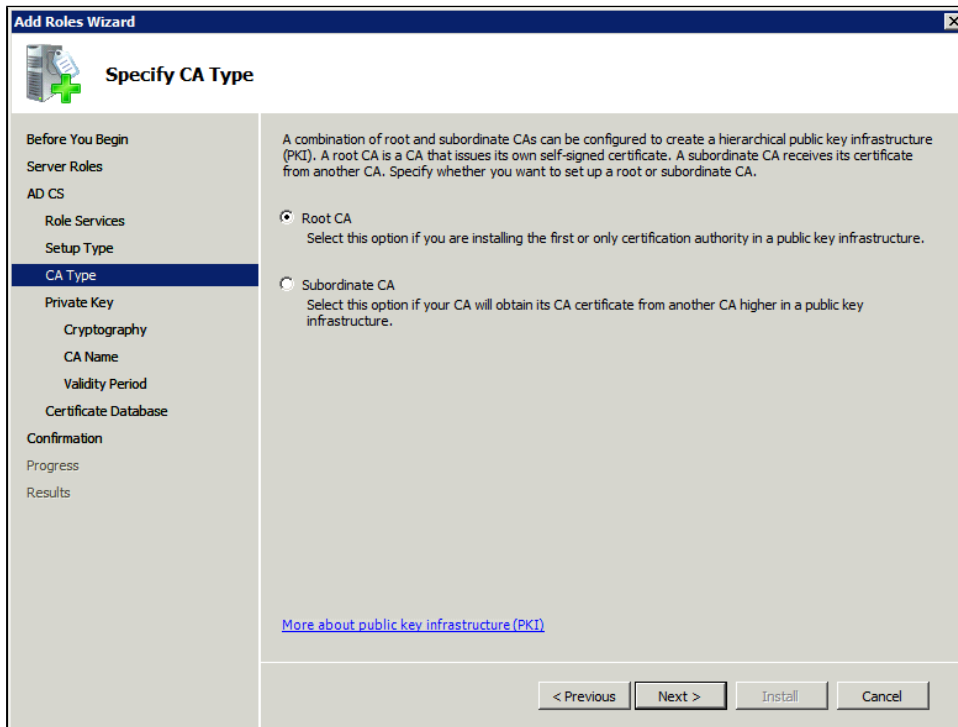
5. On the **Select Role Services** page, select the **Certification Authority** check box, and then click **Next**.



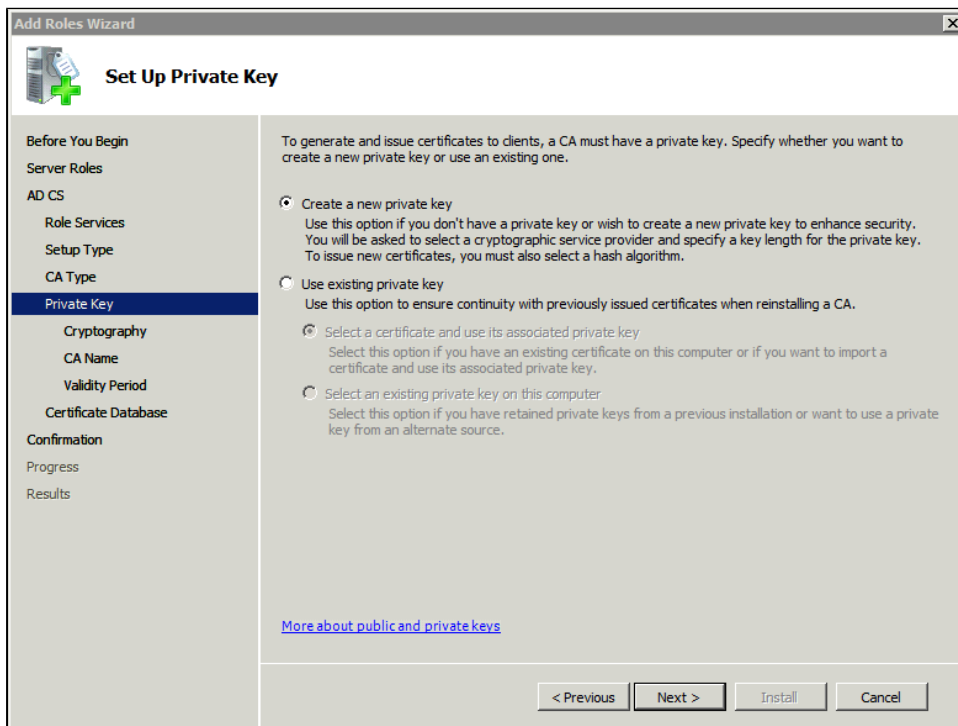
6. On the **Specify Setup Type** page, click **Enterprise**, and then click **Next**.



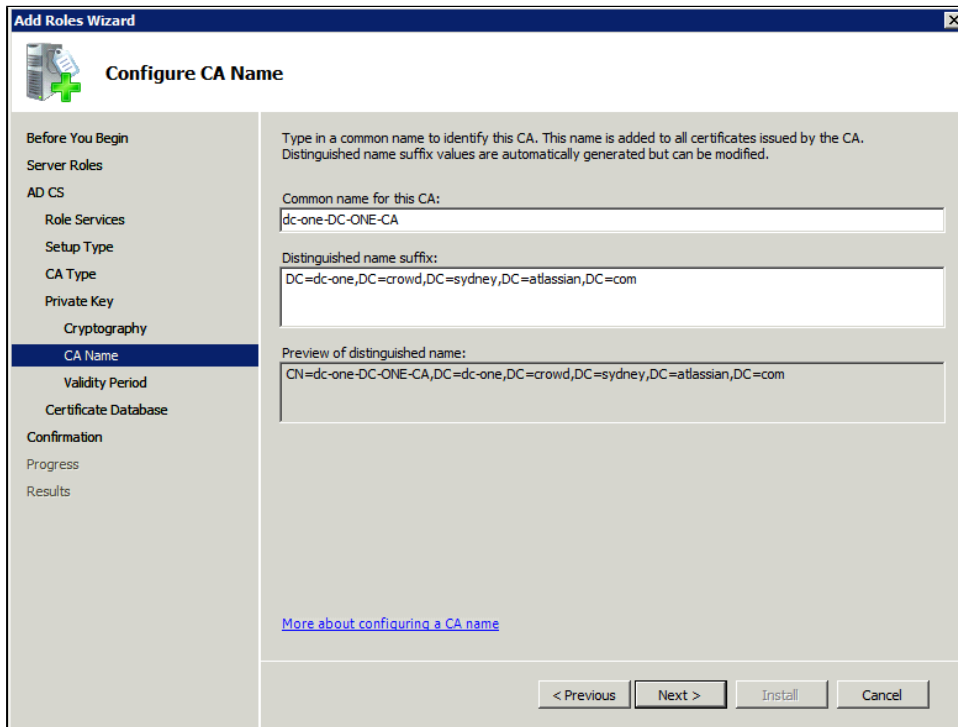
7. On the **Specify CA Type** page, click **Root CA**, and then click **Next**.



8. On the **Set Up Private Key** and **Configure Cryptography for CA** pages, you can configure optional configuration settings, including cryptographic service providers. However, the default values should be fine. Click **Next** twice.

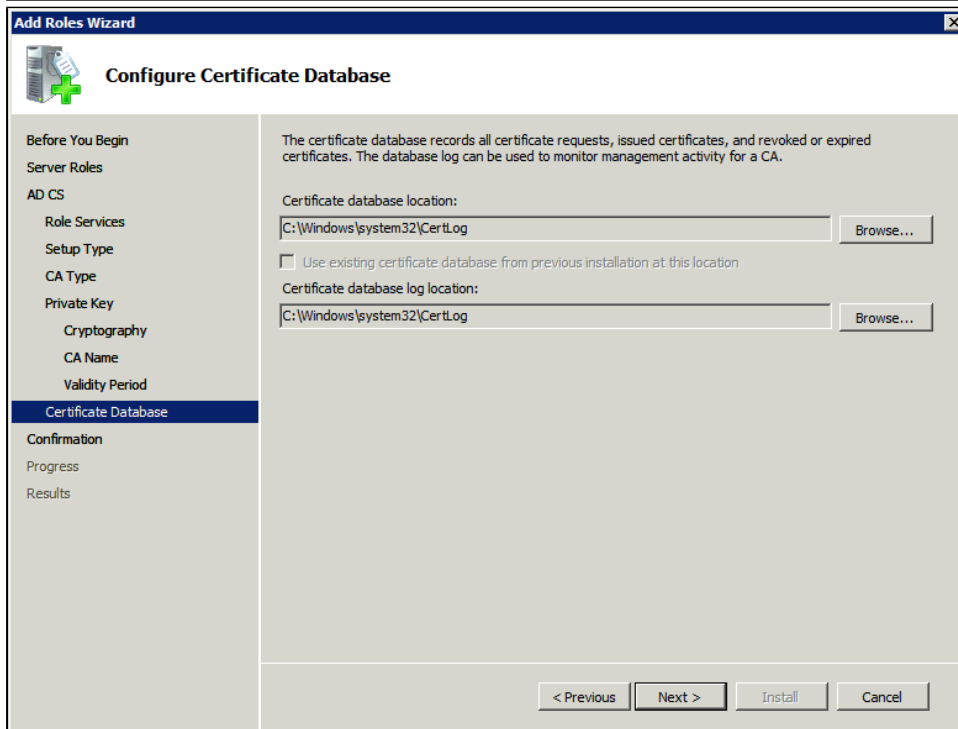
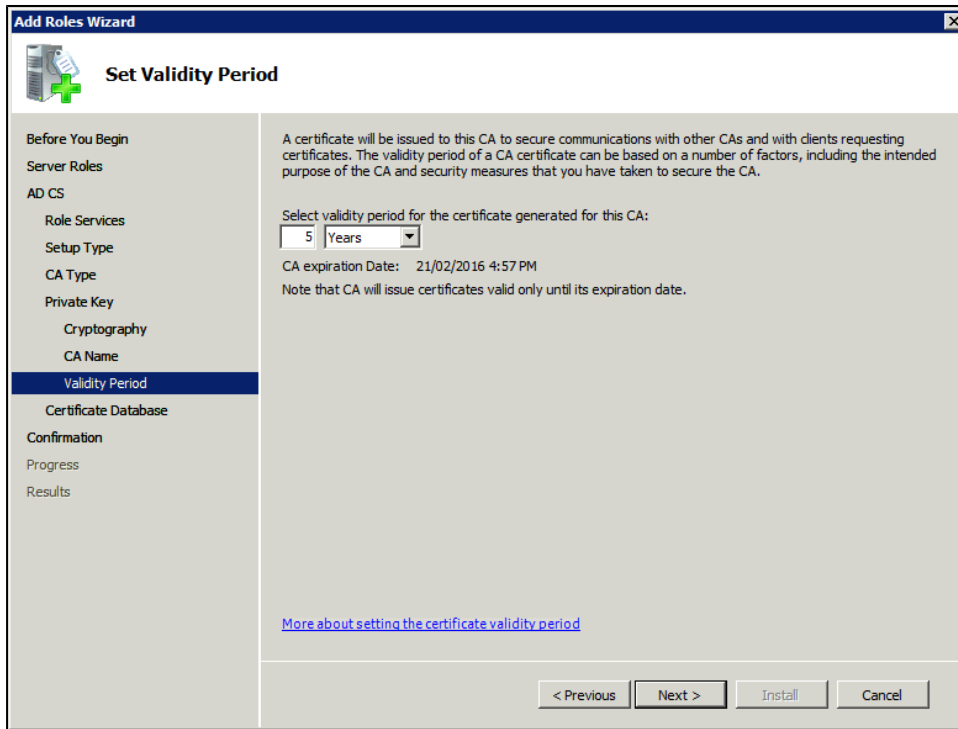


- In the **Common name for this CA** box, type the common name of the CA, and then click **Next**.

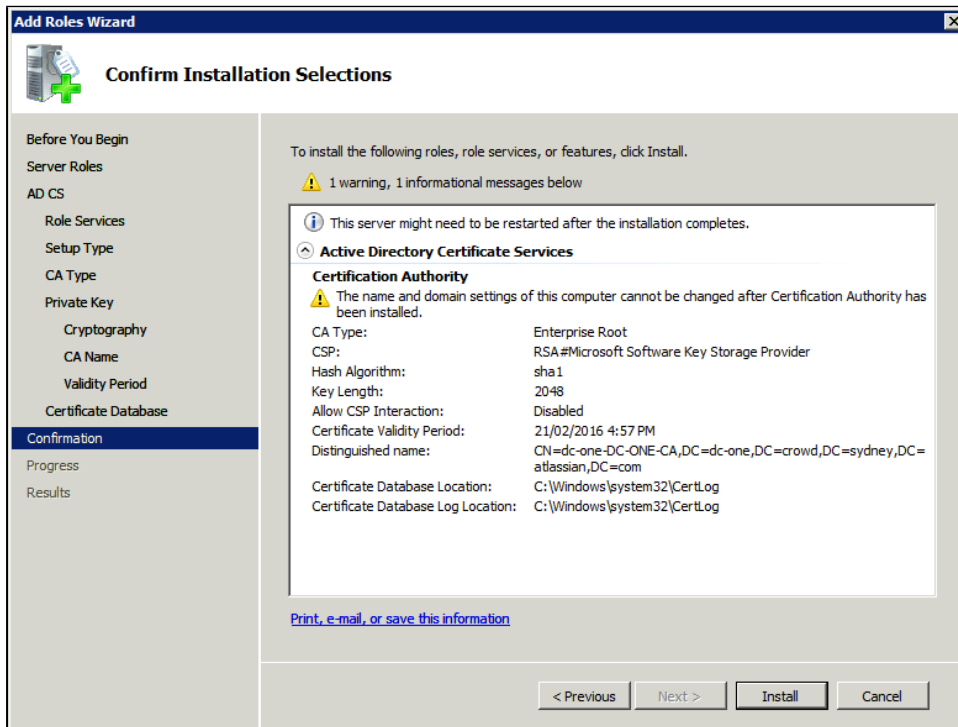


The screenshot shows the 'Add Roles Wizard' dialog box, specifically the 'Configure CA Name' step. The window title is 'Add Roles Wizard' and the main title is 'Configure CA Name'. On the left, there is a navigation pane with the following items: 'Before You Begin', 'Server Roles', 'AD CS', 'Role Services', 'Setup Type', 'CA Type', 'Private Key', 'Cryptography', 'CA Name' (highlighted), 'Validity Period', 'Certificate Database', 'Confirmation', 'Progress', and 'Results'. The main area contains the following text: 'Type in a common name to identify this CA. This name is added to all certificates issued by the CA. Distinguished name suffix values are automatically generated but can be modified.' Below this text are three text boxes: 'Common name for this CA:' with the value 'dc-one-DC-ONE-CA', 'Distinguished name suffix:' with the value 'DC=dc-one,DC=crowd,DC=sydney,DC=atlassian,DC=com', and 'Preview of distinguished name:' with the value 'CN=dc-one-DC-ONE-CA,DC=dc-one,DC=crowd,DC=sydney,DC=atlassian,DC=com'. At the bottom right, there are four buttons: '< Previous', 'Next >', 'Install', and 'Cancel'. A link 'More about configuring a CA name' is located at the bottom left of the main area.

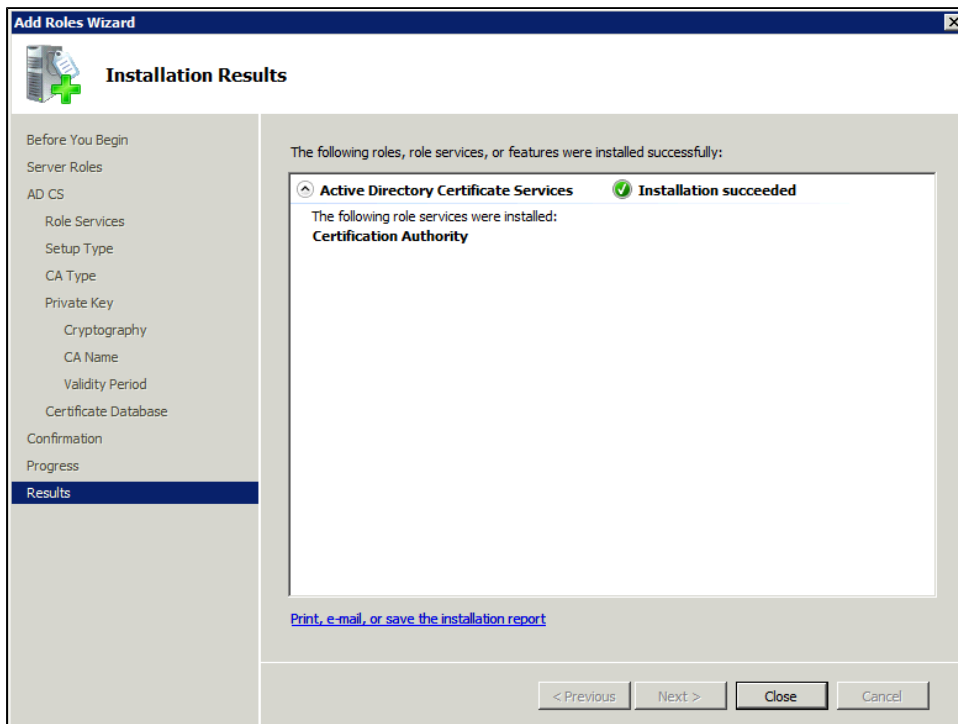
- On the **Set Validity Period** page, accept the default values or specify other storage locations for the certificate database and the certificate database log, and then click **Next**.



11. After verifying the information on the **Confirm Installation Selections** page, click **Install**.



12. Review the information on the results screen to verify that the installation was successful.



Step 2. Obtain the Server Certificate

The steps above describe how to install the certification authority (CA) on your Microsoft Active Directory server. Next, you will need to add the Microsoft Active Directory server's SSL certificate to the list of accepted certificates used by the JDK that runs your application server.

The Active Directory certificate is automatically generated and placed in root of the C:\ drive, matching a file format similar to the tree structure of your Active Directory server. For example: c:\ad2008.ad01.atlassian.com_ad01.crt.

You can also export the certificate by executing this command on the Active Directory server:

```
certutil -ca.cert client.crt
```

You might still fail to be authenticated using the certificate file above. In this case, Microsoft's [LDAP over SSL \(LDAPS\) Certificate](#) page might help. Note that you need to:

1. Choose "No, do not export the private key" in step-10 of [Exporting the LDAPS Certificate and Importing for use with AD DS](#) section
2. Choose "DER encoded binary X.509 (.CER)" in step-11 of [Exporting the LDAPS Certificate and Importing for use with AD DS](#) section. This file will be used in the following step.

Step 3. Import the Server Certificate

For an application server to trust your directory's certificate, the certificate must be imported into your Java runtime environment. The JDK stores trusted certificates in a file called a keystore. The default keystore file is called `cacerts` and it lives in the `jre\lib\security` sub-directory of your Java installation.

In the following examples, we use `server-certificate.crt` to represent the certificate file exported by your directory server. You will need to alter the instructions below to match the name actually generated.

Once the certificate has been imported as per the below instructions, you will need to restart the application to pick up the changes.

Windows

1. Navigate to the directory in which Java is installed. It's probably called something like `C:\Program Files\Java\jdk1.5.0_12`.

```
cd /d C:\Program Files\Java\jdk1.5.0_12
```

2. Run the command below, where `server-certificate.crt` is the name of the file from your directory server:

```
keytool -importcert -keystore .\jre\lib\security\cacerts -file server-certificate.crt
```

3. `keytool` will prompt you for a password. The default keystore password is `changeit`.
4. When prompted `Trust this certificate? [no]:` enter `yes` to confirm the key import:

```
Enter keystore password:  changeit
Owner: CN=ad01, C=US
Issuer: CN=ad01, C=US
Serial number: 15563d6677a4e9e4582d8a84be683f9
Valid from: Tue Aug 21 01:10:46 ACT 2007 until: Tue Aug 21 01:13:59 ACT 2012
Certificate fingerprints:
    MD5:  D6:56:F0:23:16:E3:62:2C:6F:8A:0A:37:30:A1:84:BE
    SHA1: 73:73:4E:A6:A0:D1:4E:F4:F3:CD:CE:BE:96:80:35:D2:B4:7C:79:C1
Trust this certificate? [no]:  yes
Certificate was added to keystore
```

5. Restart the application to take up the `cacerts` changes.
6. You may now change '`URL`' to use LDAP over SSL (i.e. `ldaps://<HOSTNAME>:636/`) and use the '`Secure SSL`' option when connecting your application to your directory server.

UNIX

1. Navigate to the directory in which the Java used by JIRA is installed. If the default JAVA installation is used, then it would be

```
cd $JAVA_HOME
```

2. Run the command below, where `server-certificate.crt` is the name of the file from your directory server:

```
sudo keytool -importcert -keystore ./jre/lib/security/cacerts -file server-certificate.crt
```

3. `keytool` will prompt you for a password. The default keystore password is `changeit`.
4. When prompted `Trust this certificate? [no]:` enter `yes` to confirm the key import:

```
Password:
Enter keystore password:  changeit
Owner: CN=ad01, C=US
Issuer: CN=ad01, C=US
Serial number: 15563d6677a4e9e4582d8a84be683f9
Valid from: Tue Aug 21 01:10:46 ACT 2007 until: Tue Aug 21 01:13:59 ACT 2012
Certificate fingerprints:
    MD5:  D6:56:F0:23:16:E3:62:2C:6F:8A:0A:37:30:A1:84:BE
    SHA1: 73:73:4E:A6:A0:D1:4E:F4:F3:CD:CE:BE:96:80:35:D2:B4:7C:79:C1
Trust this certificate? [no]:  yes
Certificate was added to keystore
```

5. Restart the application to take up the `cacerts` changes.
6. You may now change '`URL`' to use LDAP over SSL (i.e. `ldaps://<HOSTNAME>:636/`) and use the '`Secure SSL`' option when connecting your application to your directory server.

Mac OS X

1. Navigate to the directory in which Java is installed. This is usually

```
cd /Library/Java/Home
```

2. Run the command below, where `server-certificate.crt` is the name of the file from your directory server:

```
sudo keytool -importcert -keystore ./jre/lib/security/cacerts -file server-certificate.crt
```

3. `keytool` will prompt you for a password. The default keystore password is `changeit`.
4. When prompted `Trust this certificate? [no]:` enter `yes` to confirm the key import:

```
Password:
Enter keystore password:  changeit
Owner: CN=ad01, C=US
Issuer: CN=ad01, C=US
Serial number: 15563d6677a4e9e4582d8a84be683f9
Valid from: Tue Aug 21 01:10:46 ACT 2007 until: Tue Aug 21 01:13:59 ACT 2012
Certificate fingerprints:
    MD5:  D6:56:F0:23:16:E3:62:2C:6F:8A:0A:37:30:A1:84:BE
    SHA1: 73:73:4E:A6:A0:D1:4E:F4:F3:CD:CE:BE:96:80:35:D2:B4:7C:79:C1
Trust this certificate? [no]:  yes
Certificate was added to keystore
```

5. Restart the application to take up the `cacerts` changes.
6. You may now change '`URL`' to use LDAP over SSL (i.e. `ldaps://<HOSTNAME>:636/`) and use the '`Secure SSL`' option when connecting your application to your directory server.

RELATED TOPICS

[Configuring Crowd to Work with SSL](#)

Configuring a Remote Crowd Directory

Remote Crowd directories allow Crowd to Crowd connections. In other words, one Crowd server can obtain users and groups from another Crowd server.

Two things need to be done in order to configure the local Crowd server to obtain users and groups from a remote Crowd server:

1. The local Crowd server needs to be given access to the remote Crowd server. This is achieved by adding a new application on the remote Crowd server.
2. The new Remote Crowd directory needs to be added to the local Crowd server with details on how to connect to the remote Crowd server.

In our local testing, we found that it took about 4 minutes to sync to external Crowd with 10 000 users, 1 000 groups, and 200 000 memberships.

 Roles are not supported in Remote Crowd Directories.

To configure a Remote Crowd directory:

1. Log in to the [Crowd Administration Console](#) on the remote Crowd server.
2. [Add a new application](#) which will be the local Crowd server. The application name and password entered here will be used later by the local Crowd server to connect to the remote Crowd server.
3. Log in to the [Crowd Administration Console](#) on the local Crowd server.
4. In the top navigation bar, click **Directories**.
5. In the left-hand menu, click **Add Directory**.
6. Select **Remote Crowd**, and click **Next**.
7. Enter the name and the description.
We recommend that you leave '**Cache Enabled**' at its default setting (enabled). Remote Crowd directory caching works the same way as it does for LDAP directories. For more information, see [Configuring Caching for an LDAP Directory](#).
8. Click the **Connection** tab. See [screenshot 2](#) below.
9. Fill in the basic connection information for your remote Crowd server.
10. Click **Test Connection** to verify that Crowd can successfully connect to the directory.
11. Click the **Permissions** tab to configure the directory's [permissions](#).

Configuring Directory Details

[Screenshot 1: Directory details](#)

Create remote Crowd directory

[Details](#) [Connection](#) [Permissions](#)

Name*

A short, recognisable name that characterises this user directory. For example: "Chicago employees" or "Web customers".

Description

More information about this directory.

Active
If the directory is marked inactive, users in that directory will be unable to access Crowd or any Crowd-connected applications

Cache enabled
Enable caching to keep an up-to-date cache of directory information in the Crowd database. All queries are run against the cache instead of the directory

Use nested groups
This will enable nested group support for the directory.

[Continue](#) [Cancel](#)

Attribute	Description
Name	The name used to identify the directory within Crowd. This is useful when there are multiple directories configured, e.g. 'Chicago Employees' or 'Web Customers'.
Description	Details about this specific directory.
Active	Only deselect this if you wish to prevent all users within the directory from accessing all mapped applications . If a directory is not marked as 'Active', it is inactive . Inactive directories are: <ul style="list-style-type: none"> not included when searching for users, groups or memberships. still displayed in the Crowd Administration Console screens.
Cache Enabled	We recommend that you turn on caching. Remote Crowd directory caching works the same way as it does for LDAP directories. For more information, see Configuring Caching for an LDAP Directory .
Use Nested Groups	Enable or disable support for nested groups on the Remote Crowd directory.

Configuring Connection Details

[Screenshot 2: Connection details](#)

Create remote Crowd directory

Details **Connection** Permissions

URL*

The connection URL to use when connecting to the remote Crowd server. For example `http://hostname:8095/crowd`

Application name*

Connect to the remote Crowd server using the supplied application name.

Application password*

Connect to the remote Crowd server using the supplied application password.

Connection timeout (seconds)

Time to wait before timeout when opening a new server connection.

Max connections

Maximum number of concurrent connections to the remote crowd server.

Proxy host

Hostname of the http proxy server.

Proxy port

Port of the http proxy server.

Proxy username

Username for connecting to the http proxy server.

Proxy password

Password for connecting to the http proxy server.

Enable incremental sync

Enabling incremental synchronisation causes only changes since the last synchronisation to be queried when synchronising a directory.

Polling interval (minutes)

The directory will be periodically polled to detect changes.

Synchronise group memberships when logging in

Update group memberships from the remote directory each time a user authenticates. This ensures the group list is up to date, but can slow down authentication.

Attribute	Description
URL	The connection URL to use when connecting to the directory server. The URL should be in the following format: <code>http://domainname:port/crowd</code> .
Application Name	Application name used to authenticate to the remote Crowd server.

Application Password	Application password used to authenticate to the remote Crowd server.
Connection Timeout	The time, in seconds, to wait for a connection to be established. If there is no connection within the specified time period, the connection attempt will be aborted. A value of 0 (zero) means there is no limit.
Max Connections	The maximum number of simultaneous connections to remote Crowd server.
Proxy Host	HTTP proxy server domain name. This field is required if the remote Crowd server is behind a HTTP proxy.
Proxy Port	HTTP proxy server port number. This field is required if the remote Crowd server is behind a HTTP proxy.
Proxy Username	HTTP proxy server username. This field is required if the HTTP proxy server requires authentication.
Proxy Password	HTTP proxy server password. This field is required if the HTTP proxy server requires authentication.
Enable Incremental Sync	We recommend that you turn on incremental synchronization. It will cause only changes since the last synchronization to be queried when synchronizing, thus significantly reducing synchronization resource usage.
Polling Interval	Crowd will synchronize with the remote Crowd server every x minutes, where 'x' is the number specified here. Remote Crowd directory caching works the same way as it does for LDAP directories, for which there is more information in Configuring Caching for an LDAP Directory .

Next Step

Specify the directory permissions, which allow you to restrict the way in which applications can use the directories. See [Specifying Directory Permissions](#).

Once you have configured the directory's permissions, you have finished configuring your new directory. You can then [map](#) the directory to the appropriate applications.

RELATED TOPICS

- [Using the Directory Browser](#)
- [Adding a Directory](#)
 - [Configuring an Internal Directory](#)
 - [Configuring an LDAP Directory Connector](#)
 - [LDAP Object Structures](#)
 - [Configuring an SSL Certificate for Microsoft Active Directory](#)
 - [Configuring a Remote Crowd Directory](#)
 - [Configuring a Custom Directory Connector](#)
 - [Configuring a Delegated Authentication Directory](#)
 - [Configuring Azure Active Directory](#)
- [Configuring Caching for an LDAP Directory](#)
- [Using Naive DN Matching](#)
- [Specifying Directory Permissions](#)
- [Importing Users and Groups into a Directory](#)
 - [Importing Users from Atlassian Confluence](#)
 - [Importing Users from Atlassian Jira](#)
 - [Importing Users from Atlassian Bamboo](#)
 - [Importing Users from Jive Forums](#)
 - [Importing Users from CSV Files](#)
 - [Configuring the CSV Importer](#)
 - [Mapping CSV Fields to Crowd Fields](#)
 - [Confirming the CSV Importer Configuration](#)

- [Viewing the Results of the Import](#)
- [Importing Users from One Crowd Directory into Another](#)
- [Configuring directories for failover authentication](#)
- [Pruning delegated directories](#)

[Crowd documentation](#)

Configuring a Custom Directory Connector

Custom directory connectors allow developers to connect Crowd to custom user-stores, such as existing databases or legacy systems.

First you need to create a custom directory connector. The simplest way to accomplish this is to add a JAR file with the necessary classes to the Crowd `WEB-INF/lib` folder. For details, please see [Creating a Custom Directory Connector](#).

Once you have added your JAR file to the Crowd `WEB-INF/lib` folder, you are ready to configure a Custom Directory Connector, as described below.

To configure a Custom Directory Connector,

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click **Directories**.
3. Click **Add Directory**.
4. Select the **Custom** directory type and click **Next**.
5. Complete the fields as described in the table below.
6. Click the **Continue** button to configure the directory's [permissions](#).

i Once you have configured the directory's permissions, you will have finished configuring your new directory. You can then [map](#) the directory to appropriate applications.

Screenshot: 'Create Custom Directory'

Create custom connector

[Details](#) [Permissions](#)

Name*

A short, recognisable name that characterises this user directory. For example: "Chicago employees" or "Web customers".

Description

More information about this directory.

Active

If the directory is marked inactive, users in that directory will be unable to access Crowd or any Crowd-connected applications

Implementation*

class

Your implementation of the Java interface `com.atlassian.crowd.directory.RemoteDirectory`. Must be on the Crowd class path.

[Continue](#) [Cancel](#)

Custom Directory Store Attributes	Description
Name	The name used to identify the directory within Crowd. This is useful when there are multiple directories configured, e.g. Chicago Employees or Web Customers.
Description	Details about this specific directory.
Active	Only deselect this if you wish to prevent all users within the directory from accessing all mapped applications . If a directory is not marked as 'Active', it is inactive . Inactive directories: <ul style="list-style-type: none">• are not included when searching for users, groups or memberships.• are still displayed in the Crowd Administration Console screens.

Implementation Class	Implementation of <code>com.atlassian.crowd.directory.RemoteDirectory</code> Java interface. Must be in the Crowd CLASSPATH.
----------------------	--

Next Step:

See [Specifying Directory Permissions](#)

Configuring a Delegated Authentication Directory

A Delegated Authentication directory combines the features of an internal Crowd directory with delegated LDAP authentication. This means that you can have your users authenticated via an external LDAP directory while managing the users and groups in Crowd. You can use Crowd's flexible and simple group management when the LDAP groups do not suit your requirements. Alternatively, you can have Crowd import users' group memberships from LDAP each time they authenticate.

The username must be the same in the Crowd Delegated Authentication directory and in the LDAP directory. If a user is renamed in LDAP, Crowd will automatically rename the user in the Delegated Authentication directory.

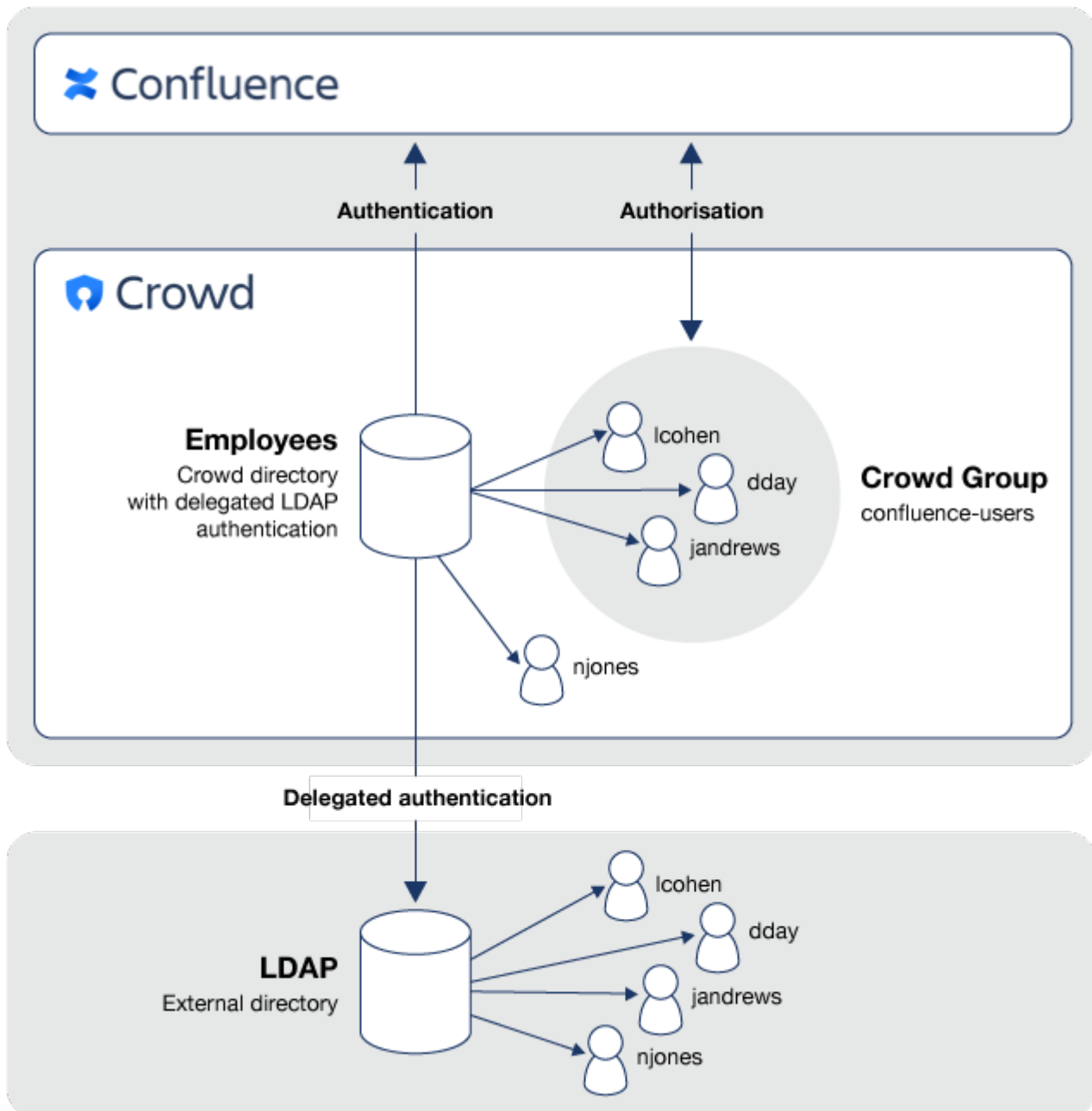
Delegated Authentication directories do not allow you to browse the LDAP data. The directory delegates user authentication to LDAP, but to be able to list users and groups, you will need to add them to the directory. See more details in the [Next Steps](#) section of this page.

Example of using a Delegated Authentication Directory

You can set up a simple group configuration in Crowd for use with [Confluence](#) and other [Atlassian](#) products, while authenticating your users against the corporate LDAP directory. You can also avoid the performance issues which might result from downloading large numbers of groups from LDAP.

The diagram below gives a conceptual overview of delegated LDAP authentication. This example assumes that you have:

- The [Confluence](#) application [integrated with Crowd](#).
- A Crowd Delegated Authentication directory called 'Employees' which contains the group 'confluence-users'.
- An LDAP directory containing all your employees and their authentication details (e.g. username and password).



Configuration Steps

Before setting up a new Delegated Authentication Directory, please review to the notes on LDAP object structures in the page about [LDAP connectors](#).

To configure a Delegated Authentication directory:

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click **Directories**.
The [Directory Browser](#) will open.
3. Click **Add Director**.
4. Select **Delegated Authentication** button.
5. Complete the configuration information required on each of the tabs to finish setting up the directory.

i If a user logs in successfully via LDAP authentication but does not yet exist in Crowd, Crowd will automatically add them to the Delegated Authentication directory. You will then need to add the user to any necessary groups, to allow them to access applications where group membership is required. If you have enabled the "Synchronize Group Memberships" option, groups and group memberships from LDAP will be automatically imported each time a user authenticates.

Next Steps

After configuring your new directory:

1. [Map](#) the directory to the appropriate applications.
2. Consider how you would like to add your users to Crowd's Delegated Authentication directory. There are a few options:
 - Manually [add the users](#) to the Crowd directory.
 - Use Crowd's [Directory importer](#) to copy your LDAP users into your Delegated Authentication directory.
 - Let Crowd do it for you, at login time by enabling the **Synchronize User Details** option when you configure the directory.

Configuring Azure Active Directory

You can configure your Microsoft Azure Active Directory (Azure AD) as a directory in Crowd. All changes to your users, groups, and memberships will be synced between Azure AD and Crowd periodically, or whenever you request it. You'll be able to view information about your users directly in Crowd by using the User Browser and Group Browser.

Before you begin

Before you configure your Azure AD, you should know about the following restrictions:

- In Azure AD, you can have multiple groups with the same name (*displayName*), but it's not supported in Crowd and results in a failing synchronization. Make sure you change your Azure AD group names to unique ones.
- Crowd doesn't support multi-factor authentication. You'll need to disable it for your users in Azure AD, or they will not be able to log in to Crowd or any integrated applications.
- If you need to make any changes to your users, make them directly in Azure AD. You can't edit your Azure AD users in Crowd.


Configuring Azure Active Directory

To configure Azure AD, you'll need to create two applications in your Azure Portal, and then use them to add Azure AD to Crowd.

1. In Azure web application.

1. Create a web application to allow Crowd to communicate with Azure AD:

1. Log in to your Azure Portal.
2. Go to **Azure Active Directory > App registrations**.
3. Create a new application registration with the following details:
 - Application type: **Web** (Option is available under Redirect URI sub-section)
 - Sign-on URL: *<Crowd's base URL>*

In Crowd, go to  **General**, and check the value of `Base URL`.

After the application is created, note down the **Application (client) ID** assigned to it. You will need it later on to configure the integration in Crowd.

2. Configure permissions for the web application to allow Crowd to read data from Azure AD:

1. In your web application, click **API permissions**.
2. In the `API permissions` section, click **Add a permission**.
3. Under **Microsoft APIs** select **Microsoft Graph**, and select **Application permissions** for the type of permissions required for this application
4. Add the following permission from:
 - `Directory.Read.All`
5. Click **Add Permissions** and then, under Grant consent section, click **Grant admin consent** button.
6. Click **Yes** and confirm.

3. Create a key for the web application. Crowd will use this key to authenticate to Azure AD:

1. Click your web application.
2. In the `Certificates & secrets` section, click **New client secret**.
3. Choose a description and an expiry date for your key then save it.

i Keep in mind that when the key expires and you don't replace it, Crowd will not be able to communicate with Azure AD.

4. Copy and store the key value.

! You will not be able to view it after navigating away from the key settings.

2. In Azure native application

4. Create a native application that will be used by Crowd to validate user credentials:

1. Go to App registrations, and create a new application registration with the following details:

- Type: *Native* (Option is available under Redirect URI sub-section)
- Redirect URL: *<Crowd's base URL>*

Note down the **Application ID** assigned to it. You will need it later on to configure the integration in Crowd.

5. Configure permissions for the native application to allow Crowd to validate user credentials:

1. Click your native application.
2. Click **API Permissions**
3. Under Grant consent section, click **Grant admin consent** button.
4. Click **Yes** and confirm.

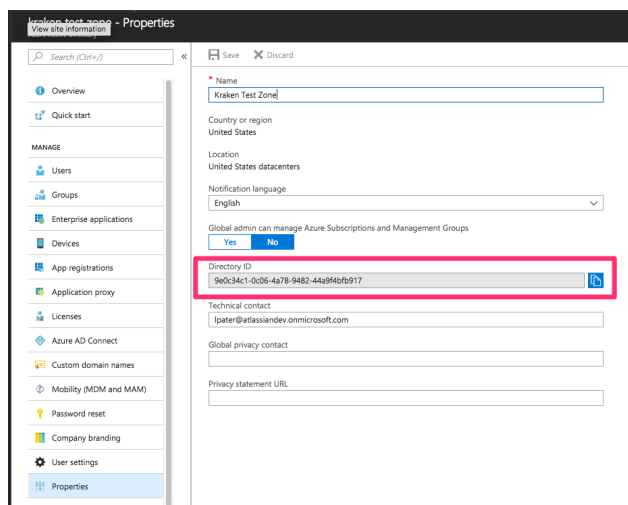
6. Configure manifest for the native application to allow Crowd to validate user credentials:

1. Click your native application.
2. Click **Manifest**
3. In the manifest editor, set the `allowPublicClient` property to `true`
4. In the bar above the manifest editor, click **Save**

7. Get the Tenant ID to configure the integration in Crowd:

1. Go to the main Azure Active Directory blade.
2. Click **Properties**.

Note down the **Directory ID**- this is the **Tenant ID** you will need later on to configure the integration in Crowd.




3. Steps in Crowd

8. Add Azure AD to Crowd.


1. Log in to the [Crowd Administration Console](#).

2. In the top navigation bar, click **Directories**.
3. Click **Add Directory**, and then select **Azure Active Directory** as type.
4. Fill out the required fields.
You will need to specify the **Tenant ID**, **Web application ID**, **Web application key** and **Native application ID** that you received when you configured Azure Active directory.
5. If you are integrating with an Azure Active Directory region that uses alternative API URLs (for example [Azure Germany](#)), you can pick the region from the **Region** drop-down.
If your region is not listed, you can pick **Custom**, and enter the appropriate API URLs manually.
6. (optional) **DATA CENTER ONLY** In the **Group filtering** section, instead of adding the whole user directory to Crowd, you can choose specific groups from Azure AD. Only members of these groups will be added to Crowd.
7. (optional) Modify the default synchronization settings to match your needs.

 **DATA CENTER ONLY** If you check **Enable group filtering** and **Enable nested groups** checkboxes, the **Synchronize group memberships when logging** setting is automatically set to *Never* and can't be changed.

8. (optional) Click **Test Connection** to verify if data you entered is correct.

You've added your Azure AD to Crowd. You should now see a brief summary of your directory, and details about the synchronization.

 In some cases, the synchronization might be failing at first because the new permission wasn't yet propagated in Azure AD. Just wait a few minutes, the problem will fix itself.

Crowd will automatically pull data from Azure AD. If that doesn't happen, you can click **Synchronise now**. Once the synchronization is complete, you can check your users and groups from Azure AD by going to **Users/Groups** in the top navigation bar.

Field mapping

The following tables show how fields in Azure AD are mapped to those in Crowd. We're comparing Azure AD's **API** fields with Crowd's **UI** fields.

Users

Azure AD	Crowd
userPrincipalName	Username
displayName	Display name
givenName	First name
familyName	Last name
accountEnabled	Active
id	External ID
Mail	E-mail address

Groups

Azure AD field	Crowd field
displayName	Name
description	Description

id	External ID
----	-------------

Configuring Caching for an LDAP Directory

Crowd manages a cache of LDAP directory information stored in the Crowd database, to ensure fast recurrent access to user and group data. We call this 'database-backed LDAP caching'.

This page describes the caching of user and group information in the Crowd database. For a description of the other types of caching offered by Crowd, please refer to [Overview of Caching](#).

i Passwords are not cached

The Crowd cache does not store user passwords. All authentication is performed by calls to the LDAP directory itself.

On this page:

- [Features of LDAP Caching in Crowd](#)
- [Supported LDAP Directories](#)
- [Configuring the Cache](#)
- [Finding the time taken to synchronize](#)
- [Manually Synchronizing the Cache](#)
- [Notes](#)

Features of LDAP Caching in Crowd

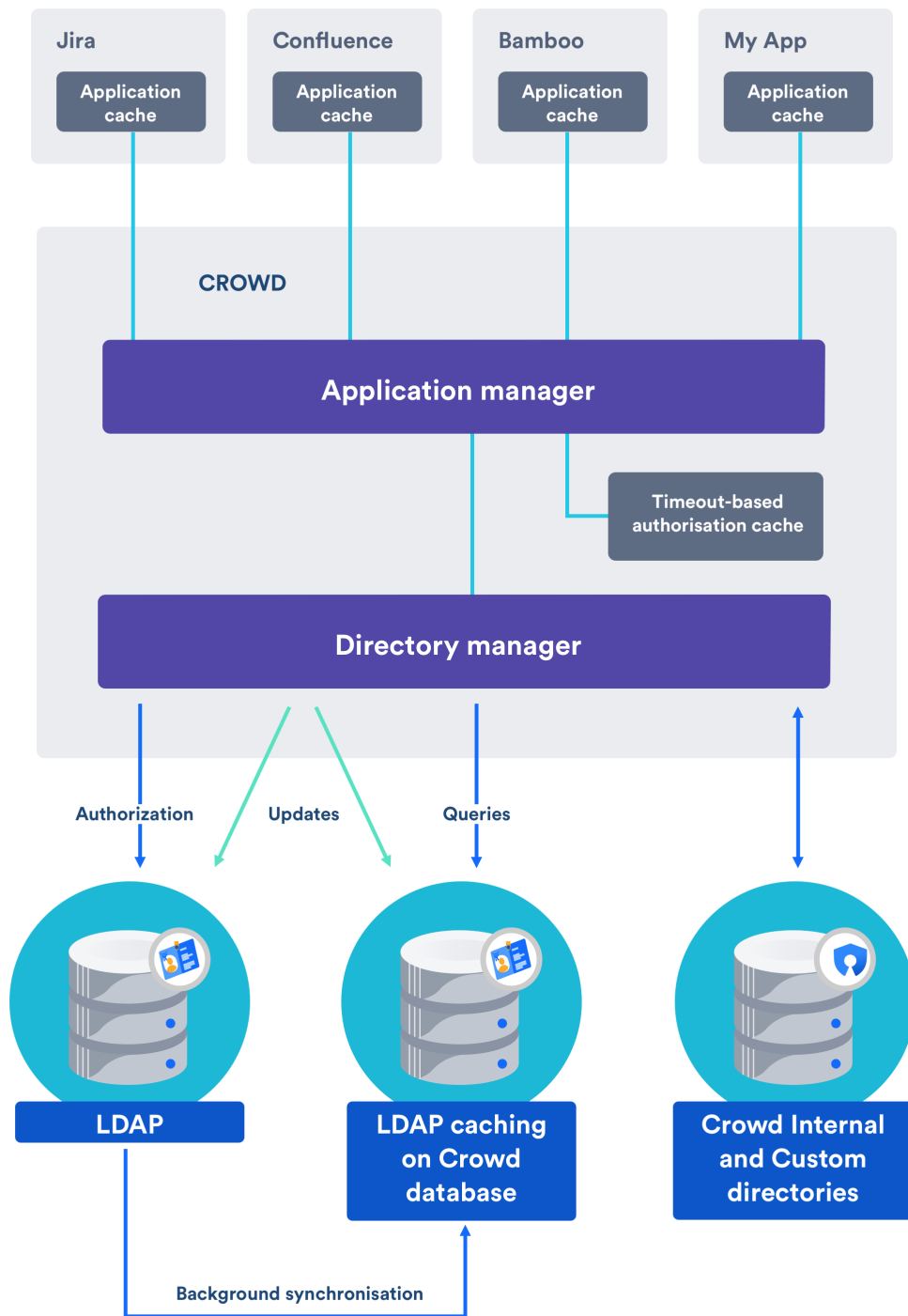
For all LDAP directories with caching enabled, Crowd will keep an up-to-date cache of user and group information retrieved from the LDAP directory. Use of the cache should improve performance particularly in directories which are slow or off site.

i Please refer to the [notes](#) below, especially regarding the **number of users** for which the caching is optimized.

Summary of the caching functionality:

- The caches are held in the Crowd database.
- When you add the directory connector to Crowd, Crowd will start a synchronization task in the background to copy all the required users, groups and membership information from LDAP to the Crowd database. This task may take a while to complete, depending on the size and complexity of your user base.
- Crowd will perform a periodic synchronization to update the database with any changes made to LDAP. The default sync interval, or polling interval, is one hour (60 minutes). You can change the polling interval on the directory connector configuration screen.
- You can manually synchronize the database-backed cache if necessary.
- Whenever an update is made to the users, groups or membership information via Crowd, Crowd will update both the database-backed cache and the LDAP directory immediately.
- For all authentication requests, Crowd performs calls to the LDAP directory itself. The Crowd database-backed cache does not store user passwords.
- Crowd performs all other queries against the database-backed cache.

The diagram below gives a conceptual overview of the caches supported by Crowd, including the LDAP database-backed caching discussed on this page. For a description of the other types of caching offered by Crowd, please refer to the [overview of caching](#).



Supported LDAP Directories

Crowd's database-backed caching is available for all the LDAP directories that Crowd supports. See [Configuring an LDAP Directory Connector](#) for the list of supported directories.

Configuring the Cache

1. In the top navigation, click **Directories**.
2. Click on the directory for which you want to configure cache.
3. In the **Detailstab**, select **Cache Enabled**.
4. In the **Connector** tab, set the polling interval option
The polling interval is the period of time, in minutes, that Crowd will wait between its requests for updates from LDAP.
 - The length of your polling interval depends on the length of time you can tolerate stale data, the amount of load you want to put on Crowd and the LDAP server, and the size of your user base. If you poll more frequently, then your data will be more up to date. The downside of polling more frequently is that you may overload your LDAP server with requests.
 - If in doubt, we recommend that you start with an interval of 60 minutes (this is the default setting) and reduce the value incrementally. You will need to experiment with your setup.

Finding the time taken to synchronize

You can find the time take to synchronize on the **Details** tab of your directory.

Last synchronisation

Started **February 8, 2019 10:35:44 AM UTC**

Time taken **5 seconds**

Status **Incremental synchronisation completed successfully**

Synchronise now

The directory connector's **Details** tab shows information about the last sync operation, including the length of time it took.

Manually Synchronizing the Cache

Screenshot: Manually syncing the cache

View directory - Atlassian AD

[Details](#) [Connector](#) [Configuration](#) [Permissions](#) [Options](#)

Name*
A short, recognisable name that characterises this user directory. For example: "Chicago employees" or "Web customers".

Description
More information about this directory.

Type **Microsoft Active Directory**

Active
If the directory is marked inactive, users in that directory will be unable to access Crowd or any Crowd-connected applications

Cache enabled
Enable caching to keep an up-to-date cache of directory information in the Crowd database. All queries are run against the cache instead of the directory

Last synchronisation

Started **February 8, 2019 10:35:44 AM UTC**

Time taken **5 seconds**

Status **Incremental synchronisation completed successfully**

You can manually synchronize the cache by clicking the **Synchronise Now** button on the the directory connector's **Details** tab. If a sync operation is already in progress, you cannot start another until the first has finished.

Notes

General Notes

- Be aware of the optimal number of users.** We have optimized the database caching for directories containing approximately 10 000 (ten thousand) users. If your directory is significantly larger, the new caching may not be as beneficial. For really large user bases, we recommend that you leave the caching disabled.
- You can reduce the number of LDAP users visible to Crowd.** You can narrow the LDAP user/group filter to control the size of the userbase visible to Crowd.
- Delegated Authentication directories are not cached.** [Delegated Authentication directories](#) are not cached, because only the authentication is delegated to the directory, and authentication itself is not cached.
- Synchronization errors are shown in the logs.** If there are any errors during the synchronization process, they will appear in the logs (not the UI). If one user fails to sync for some reason, the process will write the error to the logs, skip that user and continue with the remaining users.

Additional Notes for Active Directory

When Crowd synchronizes with Active Directory, Crowd requests only the changes from the LDAP server rather than the entire user base. This optimizes the synchronization process and gives much faster performance on the second and subsequent requests.

On the other hand, this synchronization method results in a few limitations:

- Externally moving objects out of scope or renaming objects causes problems in AD.** If you move objects out of scope, this will result in an inconsistent cache. We recommend that you do not use the

external LDAP directory interface to move objects out of the scope of the sub-tree, as defined on Crowd's Directory Connector screen. If you do need to make structural changes to your LDAP directory, manually synchronize the directory cache after you have made the changes to ensure cache consistency.

2. **Syncing between AD servers is not supported.** Microsoft Active Directory does not replicate the `uSNChanged` attribute across instances. For that reason, Crowd does not support connecting a single directory to different AD servers for syncing.
3. **You must restart Crowd after restoring AD from backup.** On restoring from backup of an AD server, the `uSNChanged` timestamps are reverted to the backup time. To avoid the resulting confusion, you will need to flush the directory cache after a Active Directory restore operation.
4. **Obtaining AD object deletions requires administrator access.** Active Directory stores deleted objects in a special container called `cn=Deleted Objects`. By default, to access this container you need to connect as an administrator and so, for Crowd to be aware of deletions, you must use administrator credentials. Alternatively, it's possible to change the permissions on the `cn=Deleted Objects` container. If you wish to do so, please see [this Microsoft KB Article](#).

Our Test Results

We performed internal testing of synchronization with an AD server on our local network consisting of 10 000 users, 1000 groups and 200 000 memberships.

We found that the initial synchronization took about 5 minutes. Subsequent synchronizations with 100 modifications on the AD server took a couple of seconds to complete.

Please keep in mind that a number of factors come into play when trying to tune the performance of the synchronization process, including:

- **Size of userbase.** Use LDAP filters to keep this to the minimum that suits your requirements.
- **Type of LDAP server.** We currently support change detection in AD, so subsequent synchronizations are much faster for AD than for other LDAP servers.
- **Network topology.** The further away your LDAP server is from your application server, the more latent LDAP queries will be.
- **Database performance.** As the synchronization process caches data in the database, the performance of your database will affect the performance of the synchronization.
- **JVM heap size.** If your heap size is too small for your userbase, you may experience heavy garbage collection during the synchronization process which could in turn slow down the synchronization.

Using Naive DN Matching

When configuring an [LDAP directory connector](#) in Crowd, you can turn 'naive DN matching' on or off. A 'DN' is a distinguished name. Naive DN matching is also known as 'relaxed DN standardization'. This page gives some background to the setting of this option.

Crowd needs to compare DNs (distinguished names) to check a number of things, such as whether a user is a member of a group. Some directories guarantee that DNs will always be in a standard format, and some return slight variants with changes such as extra whitespace. If we know that, in a specific directory, DNs are case insensitive and are always returned in a compact format (that is, the separators are commas without spaces) then we can convert both the attribute names and values to lower case and just do a direct string comparison.

i Using naive DN matching provides significant performance benefits. For that reason, we recommend enabling it where possible.

Effect of Turning Naive DN Matching On or Off

Naive DN Matching in Crowd	Processing in Crowd	Comments
Off	Crowd will perform the full DN parsing and compare the parsed version.	See below for default settings for each directory type.
On	Crowd will perform a <code>toLowerCase</code> operation and then do a direct comparison of the two DN strings.	If this setting is 'off' by default for your directory type (see below) then you may be able to turn it on. Both of the following two statements need to be true: <ol style="list-style-type: none">1. The directory server always returns memberDNs in a compact format i.e. the separators are commas without spaces. For example:<ul style="list-style-type: none">• Compact format: 'cn=bob,dc=example,dc=com'• Not compact: 'cn=bob, dc=example, dc=com'2. The attribute names in the RDN are always lower case, or all searches for DNs and memberDN attributes are case insensitive.

Default Settings in Crowd

Crowd ships with the following default settings, as determined by the characteristics of each directory type.

Directory Type	Naive DN Matching
ApacheDS 1.0.x	Off
ApacheDS 1.5.x	Off
Apple Open Directory	On
FedoraDS	On
Generic LDAP	Off
Microsoft Active Directory	On
Novell eDirectory	Off
OpenDS	Off
OpenLDAP	On

OpenLDAP Posix	On
Generic Posix	On
Sun Directory Server DSEE	Off

Specifying Directory Permissions

Directory permissions allow you to restrict the way in which directories can be used by [mapped applications](#). Often, administrators need to limit applications to only being able to read not modify directory entity data, i.e. the users and groups contained within the directory. You can achieve this by disabling the relevant directory permissions.

Directory permissions are defined at two levels:

1. **Directory-level permissions** are defined on the 'Permissions' tab of the 'View Directory' screen. These permissions apply to each application mapped to the directory, unless the application has its own application-level permissions.
2. **Application-level directory permissions** are defined on the 'Permissions' tab of the 'View Application' screen. If a permission is enabled at directory level, you can enable it for a specific application. For example, you could enable the 'Add User' permission on the 'Customers' directory in Jira but disable the permission for Confluence.

Take a look at an [example](#).

Disabling a directory-level permission will override any permissions enabled at application level. If a permission is enabled at application level and then subsequently disabled at directory level, the directory-level permission will apply. (The application-level permissions will be 'remembered' and will apply again if re-enabled at directory level.)

How do directory permissions affect the Crowd application (Crowd Administration Console)?

- If a particular permission is turned off at directory level, then **no** application can perform the related function - not even the Crowd application. So, for example, if you disable the 'Remove User' permission for a directory, then the Crowd Administration Console will not allow you to delete a user from that directory.
- The Crowd application is not bound by application-level permissions, because any user who could log into the Crowd application could change the application-level permissions for the Crowd application anyway.

You can also read more about [application-level directory permissions](#).

When you [add a new directory](#), all of its permissions are enabled by default.

To specify directory permissions

1. Configure a new directory as described in [Adding a Directory](#) **or** select an existing directory from the [Directory Browser](#).
2. In the directory, click the **Permissions** tab.
This will display a list of permissions as shown in the screenshot below.

Need to grant users permission to access an application?

To control which users within a directory may access a [mapped application](#), see [Specifying which Groups can access an Application](#).

Screenshot: Directory permissions

View directory - Atlassian AD

Details Connector Configuration Permissions Options

Permissions

- Add group
Allow groups to be added to the directory.
- Add user
Allow users to be added to the directory.
- Modify group
Allow groups to be modified in the directory.
- Modify user
Allow users to be modified in the directory.
- Modify group attributes
Allow group attributes to be modified in the directory.
- Modify user attributes
Allow user attributes to be modified in the directory.
- Remove group
Allow groups to be removed from the directory.
- Remove user
Allow users to be removed from the directory.

Permission	Description
Add Group	Allows applications to add groups to the directory.
Add User	Allows applications to add users to the directory.
Modify Group	Allows applications to modify groups in the directory.
Modify User	Allows applications to modify users in the directory.
Modify Group Attributes	Allows applications to modify group attributes in the directory.
Modify User Attributes	Allows applications to modify user attributes in the directory including the active option.
Remove Group	Allows applications to delete groups from the directory.
Remove User	Allows applications to delete users from the directory. ⚠ Consider carefully whether you allow the deletion of users, as some applications contain historical data, e.g. documents that the user has created. Read more .

Importing Users and Groups into a Directory

Once you have [added a directory](#), you can import groups and users into it from external user-stores or from another directory defined in Crowd. This can reduce the number of user-stores within your organization, and give you a consolidated, centralized point of user management. Once you have imported users into a Crowd directory, you can manage them via the Crowd Administration Console (assuming the directory's [permissions](#) allow this). For example, your organization might currently have user IDs for Atlassian Jira users stored within Jira's database, and user IDs for Jive Forums users stored within Jive's database. You could use Crowd to import all the user IDs from both places into Microsoft Active Directory.

You can import from different user-stores into a single Crowd directory, or into different Crowd directories, depending on your needs.

To import users into a directory:

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click **Users**.
3. Click **Import Users**.
This displays the **Import Type** screen (see screenshot below).
4. Select the type of user-store or file from which you want to import external users into Crowd:
 - **Atlassian Importer** see [Importing Users from Atlassian Confluence](#), [Importing Users from Atlassian Jira](#) and [Importing Users from Atlassian Bamboo](#)
 - **Directory Importer** see [Importing Users from One Crowd Directory into Another](#)
 - **CSV Importer** see [Importing Users from CSV Files](#)
 - **JIVE** see [Importing Users from Jive Forums](#)
5. Click **Next**.
6. Finish the import configuration.

Screenshot: 'Select Import Type'

The screenshot shows a web interface titled "External user importer". At the top right, there are three tabs: "Import type" (which is active and highlighted with a blue dot), "Options", and "Results". Below the tabs, there are four radio button options, each with a brief description:

- Atlassian importer**
Use the Atlassian importer to import users from Atlassian products, e.g. JIRA, Confluence, Bamboo.
- Directory importer**
Import your users and groups from another directory defined in Crowd.
- CSV importer**
Import your users and groups from a CSV file. You can supply one or two files, the first (mandatory) containing your users and another (optional) containing their group memberships (e.g. "jsmith","administrators").
- Jive Forums importer**
Import your users and groups from your Jive Forums installation.

At the bottom left of the form, there is a blue button labeled "Next".

Importing Users from Atlassian Confluence

If you have already been using Atlassian Confluence, and are now [configuring Confluence as a Crowd application](#), you will probably want to import your existing Confluence users and groups into a Crowd directory.

It is recommended that you import your Confluence users into an [Internal Directory](#) that has its '**Password Encryption**' set to '**ATLASSIAN-SHA1**'. Otherwise, users' passwords will not be copied across to Crowd.

To import users and groups from Atlassian Confluence into a Crowd directory:

1. Ensure that the database driver for the Confluence database is on Crowd's classpath. Copy the JDBC driver jar for your particular Confluence database across to `apache-tomcat/common/lib` in your Crowd installation directory.
2. Restart Crowd.
3. Log in to the [Crowd Administration Console](#).
4. From the top navigation bar, click **Users**.
The [User Browser](#) displays.
5. In the left-hand side, click **Import Users**.
6. Select the **Atlassian Importer** type.
7. Complete the fields as follows:
 - **Atlassian Product** Select 'Confluence'.
 - **Directory** Select the directory that you have created for your Confluence users.
 - **Import Passwords** Select this checkbox if you wish to import the users' passwords from Confluence. You can only import passwords if the Crowd directory is using the 'Atlassian SHA1' encryption method.
 - **Product Database URL** Type the URL of your Confluence instance's database. The exact syntax will depend on which database you are using; see [Database Configuration](#) in the *Confluence Configuration Guide*.
 - **Database Driver** type the name of your Confluence instance's database JDBC driver (e.g. for MySQL, type `com.mysql.jdbc.Driver`).
 - **Username** Type the username of the database user that Crowd will use to login to your Confluence instance's database.
 - **Password** Type the password of the database user Crowd will use to login to your Confluence instance's database.
8. Click **Continue** button to import the users from your Confluence instance into your Crowd directory.

The results screen will be displayed, showing how many users and groups have been imported into your Crowd directory.

Click the '**Users**' button to [view and manage](#) the imported users and groups via the Crowd Administration Console (assuming the directory's [permissions](#) allow this).

Screenshot: 'Import Confluence Users'

Atlassian product importer

Import type **Options** Results

Which Atlassian product are you importing from?

Atlassian product*

Select the Atlassian product to import users and groups from.

Directory*

Select the directory to import your users and groups into.

Import passwords

Password can only be imported to an internal directory that is using the Atlassian Security encryption method.

Product database URL*

Database driver*

Username*

Password

Continue Cancel

Next Step

To give the imported groups access to the [Confluence application](#), see [Specifying which Groups can access an Application](#).


Importing Users from Atlassian Jira

If you have already been using Atlassian Jira, and are now [configuring Jira as a Crowd application](#), you will probably want to import your existing Jira users and groups into a Crowd directory.

It is recommended that you import your Jira users into an [Internal Directory](#) that has its **Password Encryption** set to **'ATLASSIAN-SECURITY'**. Otherwise, users' passwords will not be copied across to Crowd.

To import users and groups from Atlassian Jira into a Crowd directory

1. Ensure that the database drivers for the Jira database are on Crowd's classpath. To do this, simply copy the JDBC driver jar for your particular Jira database across to `apache-tomcat/common/lib` in your Crowd installation directory. Then restart Crowd.
2. Log in to the [Crowd Administration Console](#).
3. In the top navigation bar, click **Users**.
The [User Browser](#) displays.
4. From the left-hand side, click **Import Users**.
5. Select the **Atlassian Importer** type and click **Next**.
6. Complete the fields as follows:
 - **Atlassian Product** Select 'Jira'.
 - **Directory** Select the directory that you have created for your Jira users.
 - **Import Passwords** Select this checkbox if you wish to import the users' passwords from Jira. You can only import passwords if the Crowd directory is using the 'Atlassian SHA1' encryption method.
 - **Product Database URL** Type the URL of your Jira instance's database. The exact syntax will depend on which database you are using; see [Connecting Jira to a Database](#) in the *Jira Installation Guide*.
 - **Database Driver** Type the name of your Jira instance's database JDBC driver, e.g.
 - For MySQL, type `com.mysql.jdbc.Driver`
 - For PostgreSQL type `org.postgresql.Driver`
 - For Oracle type `oracle.jdbc.OracleDriver`
 - For MS SQL Server type `net.sourceforge.jtds.jdbc.Driver`
 - For HSQLDB type `org.hsqldb.jdbcDriver`
 - **Username** Type the username of the database user that Crowd will use to log in to your Jira instance's database.
 - **Password** Type the password of the database user Crowd will use to log in to your Jira instance's database.

 The import process will log in to the database, not into Jira.
7. **Continue** to import the users from your Jira instance into your Crowd directory.

The results screen displays, showing how many users and groups have been imported into your Crowd directory.

Click the **'Users'** button to [view and manage](#) the imported users and groups via the Crowd Administration Console (assuming the directory's [permissions](#) allow this).

Screenshot: 'Import Jira Users'

Atlassian product importer

Import type **Options** Results

Which Atlassian product are you importing from?

Atlassian product*

Select the Atlassian product to import users and groups from.

Directory*

Select the directory to import your users and groups into.

Import passwords

Password can only be imported to an internal directory that is using the Atlassian Security encryption method.

Product database*
URL

Database driver*

Username*

Password

Next Step

To give the imported groups access to the [Jiraapplication](#), see [Specifying which Groups can access an Application](#).

Importing Users from Atlassian Bamboo

If you have already been using Atlassian [Bamboo](#), and are now [configuring Bamboo as a Crowd application](#), you will probably want to import your existing Bamboo users and groups into a Crowd directory.

We recommend that you import your Bamboo users into an [internal Crowd directory](#) that has its '**Password Encryption**' set to '**ATLASSIAN-SHA1**'. Otherwise, users' passwords will not be copied across to Crowd.

To import users and groups from Atlassian Bamboo into a Crowd directory:

1. Ensure that the database driver for the Confluence database is on Crowd's classpath. Copy the JDBC driver jar for your particular Confluence database across to `apache-tomcat/common/lib` in your Crowd installation directory.
2. Restart Crowd.
3. Log in to the [Crowd Administration Console](#).
4. From the top navigation bar, click **Users**.
5. The [User Browser](#) displays.
6. In the left-hand side, click **Import Users**.
7. Select the **Atlassian Importer** type.
8. Complete the fields as follows:
 - **Atlassian Product** Select 'Bamboo'.
 - **Directory** Select the directory that you have created for your Bamboo users.
 - **Import Passwords** Select this checkbox if you wish to import the users' passwords from Bamboo. You can only import passwords if the Crowd directory is using the 'Atlassian SHA1' encryption method.
 - **Product Database URL** Type the URL of your Bamboo instance's database. The exact syntax will depend on which database you are using. See [Database Configuration](#) in the *Bamboo Installation Guide*.
 - **Database Driver** Type the name of your Bamboo instance's database JDBC driver (e.g. for MySQL, type `com.mysql.jdbc.Driver`).
 - **Username** Type the username of the database user that Crowd will use to log in to your Bamboo instance's database.
 - **Password** Type the password of the database user Crowd will use to log in to your Bamboo instance's database.
9. Click **Continue** button to import the users from your Confluence instance into your Crowd directory.

The results screen will be displayed, showing how many users and groups have been imported into your Crowd directory.


Click the '**Users**' button to [view and manage](#) the imported users and groups via the Crowd Administration Console (assuming the directory's [permissions](#) allow this).

[Screenshot: 'Import Bamboo Users'](#)


Atlassian product importer

Import type **Options** Results

Which Atlassian product are you importing from?

Atlassian product* 

Select the Atlassian product to import users and groups from.

Directory* 

Select the directory to import your users and groups into.

Import passwords

Password can only be imported to an internal directory that is using the Atlassian Security encryption method.

Product database URL*

Database driver*

Username*

Password

Next Step

To give the imported groups access to the [Bamboo application](#), see [Specifying which Groups can access an Application](#).

Importing Users from Jive Forums

If you have already been using Jive Forums, and are now [configuring Jive Forms as a Crowd application](#), you will probably want to import your existing Jive users and groups into a Crowd directory.

Before you begin:

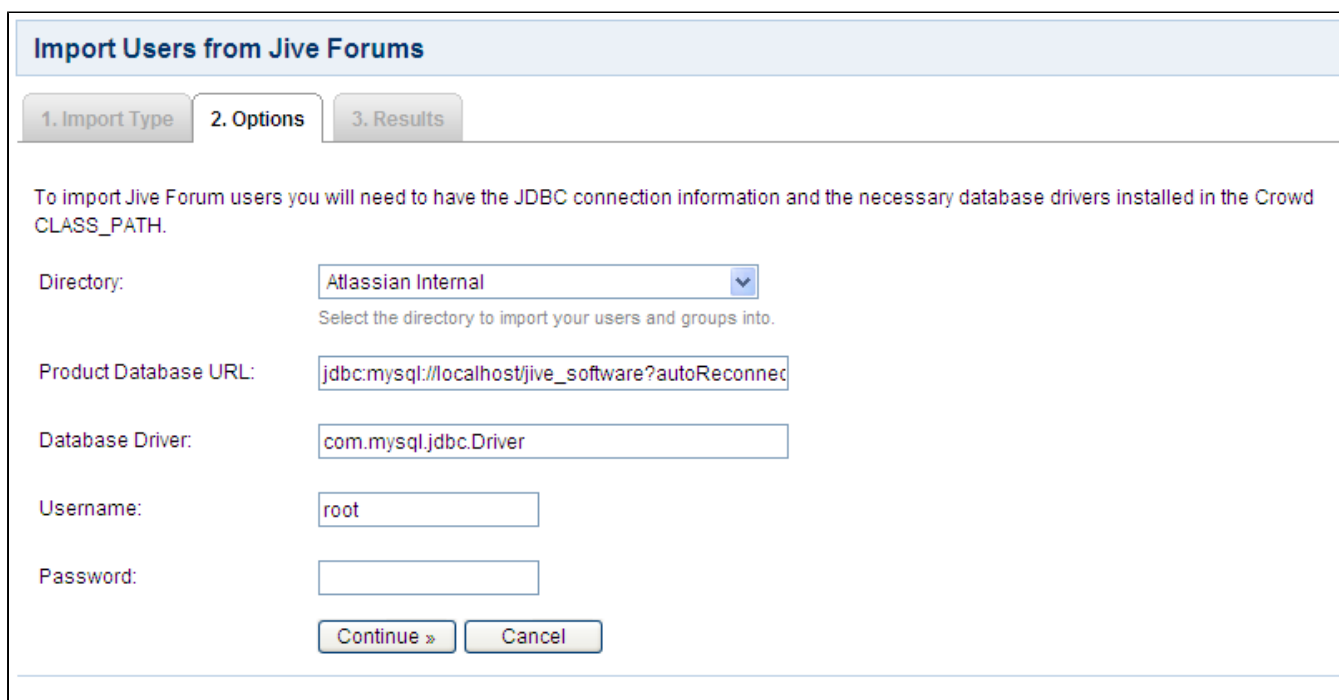
The database drivers for the Jive Forums database will need to be on Crowd's classpath. To do this, simply copy the database driver JAR for your particular Jive database across to `CROWD/apache-tomcat/common/lib` and restart Crowd.

Note: the passwords for users in Jive will not be copied across to Crowd as they are stored as hashes in Jive's internal database.

To import users and groups from Jive Forums into a Crowd directory,

1. Login to the [Crowd Administration Console](#).
2. Click the **'Users'** link in the top navigation bar.
3. This will display the [User Browser](#). Click the **'Import Users'** link.
4. This will display the **'Import Type'** screen. Click the **'JIVE'** button.
5. This will display the **'Options'** screen. Complete the fields as follows:
 - **'Directory'** select the directory that is [mapped](#) to the [Jive Forums application](#).
 - **'DB URL'** type the URL of Jive's database.
 - **'DB Driver'** type the name of Jive's database JDBC driver.
 - **'Username'** type the username of the database user that Crowd will use to login to Jive's database.
 - **'Password'** type the password of the database user Crowd will use to login to Jive's database.
6. Click the **'Continue'** button to import the users from Jive Forums into your Crowd directory.
7. The **'Status'** screen will be displayed, showing how many users and groups have been imported into your Crowd directory.
8. Click the **'Users'** button to [view and manage](#) the imported users and groups via the Crowd Administration Console (assuming the directory's [permissions](#) allow this).

Screenshot: 'Import Jive Users'



Import Users from Jive Forums

1. Import Type 2. Options 3. Results

To import Jive Forum users you will need to have the JDBC connection information and the necessary database drivers installed in the Crowd CLASS_PATH.

Directory:
Select the directory to import your users and groups into.

Product Database URL:

Database Driver:

Username:

Password:

Next Step

To give the imported groups access to the [Jive Forums application](#), see [Specifying which Groups can access an Application](#).

Related Topics

- [Using the Directory Browser](#)
- [Adding a Directory](#)
 - [Configuring an Internal Directory](#)
 - [Configuring an LDAP Directory Connector](#)
 - [LDAP Object Structures](#)
 - [Configuring an SSL Certificate for Microsoft Active Directory](#)
 - [Configuring a Remote Crowd Directory](#)
 - [Configuring a Custom Directory Connector](#)
 - [Configuring a Delegated Authentication Directory](#)
 - [Configuring Azure Active Directory](#)
- [Configuring Caching for an LDAP Directory](#)
- [Using Naive DN Matching](#)
- [Specifying Directory Permissions](#)
- [Importing Users and Groups into a Directory](#)
 - [Importing Users from Atlassian Confluence](#)
 - [Importing Users from Atlassian Jira](#)
 - [Importing Users from Atlassian Bamboo](#)
 - [Importing Users from Jive Forums](#)
 - [Importing Users from CSV Files](#)
 - [Configuring the CSV Importer](#)
 - [Mapping CSV Fields to Crowd Fields](#)
 - [Confirming the CSV Importer Configuration](#)
 - [Viewing the Results of the Import](#)
 - [Importing Users from One Crowd Directory into Another](#)
- [Configuring directories for failover authentication](#)
- [Pruning delegated directories](#)

[Crowd documentation](#)

Importing Users from CSV Files

You can copy users from an external directory or user base into Crowd via a CSV (comma-separated values) file. There are two phases involved:

1. Export your existing users and their group memberships from your external directory into a CSV file or files.
2. Import the users, groups and group memberships into a Crowd directory from the CSV files.

The CSV importer is available with Crowd 1.1.1 and later.

How the CSV Importer Handles Data

The CSV Importer **adds** to the Crowd directory, but does not update or delete existing information:

- If the Username already exists in Crowd, the CSV Importer does not overwrite the information for that user even if the Username exists in the CSV file with different user information.
- The CSV Importer does not remove users from Crowd.
- If your 'Group Membership' CSV file contains additional group(s) for a user, the additional group(s) and group membership(s) will be imported.
- Existing group memberships will not be changed or removed.

Preparing the CSV import directory

Starting from Crowd 4.2.4 and 4.3.5, any CSV files that you want to import to Crowd must be added to the `import` directory. You need to create this directory manually and make it readable to the user that is running Crowd.

To create the `import` directory:

1. Go to your Crowd shared home directory.
2. Create a directory and name it `import`.
3. Add the CSV files to this directory.

Example

Here's an example, just to show you the hierarchy:

```
/var/atlassian/crowd/home/shared/import.
```

Preparing your CSV Files

You will need:

- a CSV file containing user information, and
- optionally, another CSV file containing group memberships.

Attached are simple examples of the CSV files:

- [Example user CSV file](#)
- [Example group membership CSV file](#)

The CSV Importer's '[File Mappings](#)' screen allows you to match the CSV fields to Crowd's User and Group fields.

Formatting and location of the CSV files:

Requirement	Description
Location	The CSV files must be on the local drive (e.g. C:) of the Crowd server. If you're using Crowd 4.2.4, 4.3.5, or newer, CSV files must additionally be placed within the <i>'import'</i> subdirectory of the shared Crowd home.

Supported attributes	The CSV Importer does not support custom attributes. The supported attributes are shown in the drop-down lists on the 'File Mappings' screen .
Header row	The first row in each CSV file must be a header row. The CSV Importer will not import the information in the first row. The information in the first row is displayed in the column labeled 'CSV Header Row' on the 'File Mappings' screen
Delimiter	The fields in the CSV file must be separated by a single-character delimiter. The CSV Importer's 'Configuration' screen lets you tell Crowd which delimiter you have used.
Passwords	You will need to decide whether to import your passwords into Crowd. And if you do import the passwords, you must choose to import them as either encrypted or clear text. ⚠ Check the password encryption in the directory you are exporting users from, and compare it with the encryption method of the Crowd directory you want to import the users into. You can use Crowd's Directory Browser to view the directory's configuration details, including the encryption method. The CSV Importer's 'Configuration' screen lets you tell the CSV Importer whether to encrypt the passwords.

To export information from your user directory into a CSV file:

1. Export the users from your external user directory or database into a CSV file. Your directory or user base should have an option to allow you to do this.
2. If you want to copy your existing group memberships into Crowd, export the groups and group memberships into another CSV file.

Importing the CSV Files into Crowd

Once you have prepared your CSV file(s), you can import the users and groups into a Crowd directory.

To import users and groups from CSV files:

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click **Users**.
This will display the [User Browser](#).
3. Click **Import Users**.
4. In the Import type screen, select **CSV Importer** and click **Next**.
5. Enter the details of the CSV files as described in ['Configuring the CSV Importer'](#) and click **Continue**.

RELATED TOPICS

- [Configuring the CSV Importer](#)
- [Mapping CSV Fields to Crowd Fields](#)
- [Confirming the CSV Importer Configuration](#)
- [Viewing the Results of the Import](#)

[Crowd documentation](#)

Configuring the CSV Importer

Once you have [started the CSV Importer](#), the 'Configuration' screen allows you to specify information about the Crowd directory and CSV file(s) involved in the import.

Refer to information on [preparing your CSV files](#).

To configure the CSV importer

i The CSV files to import (the one that are entered in the 'User File' and 'Group Membership File' fields must be in a path readable by the Crowd server performing the import.

1. Start the [CSV Importer](#).
2. Complete the following fields:
 - **Directory** Select the Crowd user directory into which you want to import the users.
 - **Are your passwords encrypted?**
 - Select **Yes** if the passwords in your CSV file are already encrypted. Crowd will not re-encrypt the passwords during the import.
 - Select **No** if the passwords in your CSV file are not encrypted. Crowd will encrypt the passwords during the import, using the encryption method of the Crowd directory you are importing into.
 - **User state after import** Select the default state for users imported with the CSV file.
 - **Delimiter** Type the single-character delimiter used to separate the fields in your CSV file(s).
 - **User File** Type the location of the CSV file containing the users you wish to import. If you're using Crowd 4.2.4, 4.3.5, or newer, the file path must be relative to the 'import' subdirectory of the shared Crowd home.
 - **Group Membership File** If you want to import groups and group memberships of your users, type the location of the CSV file containing the group membership information. If you're using Crowd 4.2.4, 4.3.5, or newer, the file path must be relative to the 'import' subdirectory of the shared Crowd home.
3. Click the **Continue** button to [map the CSV fields to the Crowd directory fields](#).

Screenshot: 'CSV Importer - Configuration'

CSV importer

Import type
Configuration
File mappings
Confirmation
Results

Import your users and their group memberships

Directory* Atlassian Crowd server ▼
Select the directory to import your users and groups into.

Are your passwords encrypted Yes No
If you are importing passwords, are they already encrypted?

User state after import Active Inactive Same as in the CSV file
Select the default state for users imported with the CSV file.

Delimiter*
The CSV file delimiter used in your file(s)

User file*
The file containing your users (e.g. "John","Smith","jsmith","john@atlassian.com","password").

Group membership file
The file containing your users' group membership information (e.g. "jsmith","administrators").

RELATED TOPICS

- [Using the Directory Browser](#)
- [Adding a Directory](#)
 - [Configuring an Internal Directory](#)
 - [Configuring an LDAP Directory Connector](#)
 - [LDAP Object Structures](#)
 - [Configuring an SSL Certificate for Microsoft Active Directory](#)
 - [Configuring a Remote Crowd Directory](#)
 - [Configuring a Custom Directory Connector](#)
 - [Configuring a Delegated Authentication Directory](#)
 - [Configuring Azure Active Directory](#)
- [Configuring Caching for an LDAP Directory](#)
- [Using Naive DN Matching](#)
- [Specifying Directory Permissions](#)
- [Importing Users and Groups into a Directory](#)
 - [Importing Users from Atlassian Confluence](#)
 - [Importing Users from Atlassian Jira](#)
 - [Importing Users from Atlassian Bamboo](#)
 - [Importing Users from Jive Forums](#)
 - [Importing Users from CSV Files](#)
 - [Configuring the CSV Importer](#)
 - [Mapping CSV Fields to Crowd Fields](#)
 - [Confirming the CSV Importer Configuration](#)
 - [Viewing the Results of the Import](#)
 - [Importing Users from One Crowd Directory into Another](#)
- [Configuring directories for failover authentication](#)
- [Pruning delegated directories](#)

[Crowd documentation](#)

Mapping CSV Fields to Crowd Fields

Once you have entered details on the [Configuration screen of the CSV Importer](#), the 'File Mappings' screen allows you to match the CSV fields to the User and Group fields in Crowd. Crowd will use these mappings to import the information from the CSV file(s) into your Crowd directory.

Refer to information on [preparing your CSV files](#).

The 'File Mappings' screen has two main sections:

- **'User Mappings'** Use this section to map the fields in your 'User' CSV file.
- **'Group Mappings'** Use this section to map the fields in your 'Group Membership' CSV file, if you have one. This section will only appear if you have specified a 'Group Membership File' on the [Configuration screen](#).

Each section has the following columns:

Column	Description
CSV Header Row	This column shows the text from each field in the first row of your CSV file. The CSV Importer assumes that the first row is a header row.
Sample Row	This column shows the text from each field in the second row of your CSV file. This is done to help you with the mapping process.
Mapping	Each row in this column contains a drop-down list of the Crowd field names available for mapping. To map a Crowd field to a CSV field, select the appropriate Crowd field name from the drop-down list to match the CSV field shown in the 'CSV Header Row' column.

In the 'User Mappings' section, the 'Mapping' drop-down lists contain the following Crowd field names:

Crowd field	Description
Username	Required. One of the rows on the screen must map this value to the CSV field containing the usernames.
First Name	Required. One of the rows on the screen must map this value to the CSV field containing the users' first names.
Last Name	Required. One of the rows on the screen must map this value to the CSV field containing the users' last names.
Email Address	Required. One of the rows on the screen must map this value to the CSV field containing the users' email addresses.
Status	Required. You can modify this field only if you selected the import status to be Same as in the CSV file . One of the rows on the screen must map this value to the CSV field containing the status. Crowd recognizes the following status values: <ul style="list-style-type: none">• for active status: "active", "true", "t", "1"• for inactive status: "inactive", "false", "f", "0"
Display Name	Optional. One of the rows on the screen can map this value to the CSV field containing the users' display name.
Password	If your CSV file contains passwords, map this value to the CSV field containing the passwords.
None	Select 'None' if the CSV field displayed under 'CSV Header Row' is not to be mapped to any Crowd fields. These CSV fields will not be imported into Crowd.

In the '**Group Mappings**' section (if present), the 'Mapping' drop-down lists contain the following Crowd field names:

Crowd field	Description
Group Name	Required. One of the rows on the screen must map this value to the CSV field containing the names of the groups.
Username	Required. One of the rows on the screen must map this value to the CSV field containing the usernames.
None	Select 'None' if the CSV field displayed under 'CSV Header Row' is not to be mapped to any Crowd fields. These CSV fields will not be imported into Crowd.

To map the CSV fields to Crowd fields,

1. Start the [CSV Importer](#).
2. Complete the details on the '[Configuration screen](#)' and click the 'Continue' button.
3. This will display the '**File Mappings**' screen. Complete the mappings in the '**User Mappings**' section as follows:
 - In the 'CSV Header Row' column, find the field which contains your users' first names select '**First Name**' from the drop-down list in the 'Mapping' column.
 - In the 'CSV Header Row' column, find the field which contains your users' last names select '**Last Name**' from the drop-down list in the 'Mapping' column.
 - In the 'CSV Header Row' column, find the field which contains your users' email addresses select '**Email Address**' from the drop-down list in the 'Mapping' column.
 - In the 'CSV Header Row' column, find the field which contains the usernames select '**Username**' from the drop-down list in the 'Mapping' column.
 - In the 'CSV Header Row' column, find the field which contains your users' passwords select '**Password**' from the drop-down list in the 'Mapping' column.
 - Select '**None**' from the drop-down lists for all unmatched rows.
4. Complete the mappings in the '**Group Mappings**' section (if present) as follows:
 - In the 'CSV Header Row' column, find the field which contains the group names select '**Group Name**' from the drop-down list in the 'Mapping' column.
 - In the 'CSV Header Row' column, find the field which contains the usernames select '**Username**' from the drop-down list in the 'Mapping' column.
 - Select '**None**' from the drop-down lists for all unmatched rows.
5. Click the '**Continue**' button to [confirm the CSV configuration](#).

[Screenshot: 'CSV Importer - File Mappings'](#)

Crowd Applications Users Groups Directories Audit log

Search users
Add user
Import users

CSV importer

Import type Configuration **File mappings** Confirmation Results

Map the fields in your CSV files to the Crowd user and group attributes.

User mappings

CSV header row	Sample row	Mapping
User Name	user0	<input checked="" type="checkbox"/> None
First Name	First0	<input type="checkbox"/> First name
Last Name	Last0	<input type="checkbox"/> Last name
Email Address	user0@example.com	<input type="checkbox"/> Email address
Status	t	<input type="checkbox"/> Username
Display Name	displayname0	<input type="checkbox"/> Password
Password	secret0	<input type="checkbox"/> Display name
		<input type="checkbox"/> Status

[Continue](#) [Previous](#)

Powered by Atlassian Crowd Version: 4.2.0-m05-SNAPSHOT (Build:#1579 - 2020-09-24) 6529ab7e-4fc5-45be-bba3-004e73e0bc20

[Report a bug](#) · [Request a feature](#) · [About](#) · [Contact Atlassian](#)

ATLASSIAN

Confirming the CSV Importer Configuration

The 'Confirmation' screen allows you to review your [configuration](#) and [mapping](#) before performing the [CSV import](#).

To confirm the CSV configuration and mapping,

1. Review the information shown on the 'Confirmation' screen.
2. Click the **Continue** button to import the users from your CSV file into your Crowd directory.
3. Once the import is complete, Crowd will display the [Results](#) screen.

Screenshot: 'CSV Importer - Confirmation'

The screenshot displays the 'CSV importer' confirmation screen in the Atlassian Crowd application. The interface includes a navigation menu on the left with options like 'Search users', 'Add user', and 'Import users'. The main content area is titled 'CSV importer' and contains the following information:

- Directory: Atlassian Crowd server
- User file: /Users/ckrawczyk/IdeaProjects/crowd/components/crowd-acceptance-test/src/main/resources/com/atlassian/crowd/acceptance/tests/applications/crowd/10UsersWithStatusAndDisplayName.csv
- Are your passwords encrypted? Yes
- User status after import - Same as in the CSV file

A table titled 'User mappings' shows the mapping between CSV header rows and Crowd fields:

CSV header row	Mapping
User Name	Username
First Name	First name
Last Name	Last name
Email Address	Email address
Status	Status
Display Name	Display name
Password	Password

At the bottom of the main content area, there are two buttons: 'Continue' (highlighted in blue) and 'Previous'.

Footer information includes: Powered by Atlassian Crowd Version: 4.2.0-m05-SNAPSHOT (Build:#1579 - 2020-09-24) 6529ab7e-4fc5-45be-bba3-004e73e0bc20. Links for 'Report a bug', 'Request a feature', 'About', and 'Contact Atlassian' are also present.

Viewing the Results of the Import

The 'Results' screen shows the outcome of the [CSV import](#).

i The CSV Importer **adds** to the Crowd directory, but does not update or delete existing information:

- If the Username already exists in Crowd, the CSV Importer does not overwrite the information for that user even if the Username exists in the CSV file with different user information.
- The CSV Importer does not remove users from Crowd.
- If your 'Group Membership' CSV file contains additional group(s) for a user, the additional group(s) and group membership(s) will be imported.
- Existing group memberships will not be changed or removed.
- The 'Results' screen will show number of duplicate usernames in the CSV file which were ignored i.e. not imported.
- The 'Results' screen will show number of duplicate group names in the CSV file which were ignored i.e. not imported.

Screenshot: 'CSV Importer - Results'

The screenshot shows the 'CSV importer' configuration page in the Atlassian Crowd interface. The page has a dark blue header with navigation links: 'Crowd', 'Applications', 'Users', 'Groups', 'Directories', and 'Audit log'. On the left side, there is a sidebar with 'Search users', 'Add user', and 'Import users' options. The main content area is titled 'CSV importer' and includes sub-links for 'Import type', 'Configuration', and 'File mapping'. The configuration details are as follows:

- Confirm the configuration for your import**
- Directory: Atlassian Crowd server
- User file: /Users/ckrawczyk/IdeaProjects/crowd/components/crowd-acceptance-test/src/main/resources/com/atlassian/crowd/acceptance/tests/applications/crowd/10UsersWithStatusAndDisplayName.csv
- Are your passwords encrypted? Yes
- User status after import - Same as in the CSV file

User mappings	
CSV header row	Mapping
User Name	Username
First Name	First name
Last Name	Last name
Email Address	Email address
Status	Status
Display Name	Display name
Password	Password

Importing Users from One Crowd Directory into Another

Warning

Please take in mind that this does not work for Remote Directories e.g. From one Crowd to another Crowd

Once you have [added a directory](#), you can import users and groups into it from an external system or from another directory defined in Crowd. To learn about importing from external systems, refer to [Importing Users and Groups into a Directory](#). Below we tell you how to import from one Crowd directory to another.

You can copy users, groups and memberships:

- From an [LDAP directory](#) to a [Delegated Authentication directory](#).
- From one [internal Crowd directory](#) to another internal Crowd directory.

Things to be aware of:

- The '**Password Encryption**' method must be the same in both directories, otherwise you will not be able to copy the users across.
- The directory importer does not support nested groups when importing users and groups from LDAP into a [delegated authentication](#) directory. See [CWD-1334](#).
- The '**source directory**' is the directory you want to copy users and groups from. The '**destination directory**' is where you want to copy them to. Both directories must be defined in Crowd before you start the import process.

To import users and groups from one Crowd directory into another,

1. Log in to the [Crowd Administration Console](#).
2. If not already defined, [add the source directory](#) to Crowd.
3. If not already defined, [add the destination directory](#) to Crowd.
4. Click the '**Users**' link in the top navigation bar.
5. This will display the [User Browser](#). Click the '**Import Users**' link.
6. This will display the '**Import Type**' screen. Click the '**Directory Importer**' button.
7. This will display the '**Options**' screen, shown below. Complete the fields as follows:
 - '**Source Directory**' Select the directory that contains the users and groups you want to copy.
 - '**Destination Directory**' Select the directory that you want to copy the users and groups into.
 - '**Overwrite Destination Directory**' Tick the box if you want to delete and replace all the details and memberships for any user who exists in both source and destination directories:
 - If the checkbox is empty, Crowd will not update the user details for that username in the destination directory, but will add any new group memberships for that username.
 - If the checkbox is ticked, Crowd will remove all the details and memberships for that username from the destination directory and replace them with the details and memberships from the source directory.
8. Click the '**Continue**' button.
9. The '**Confirmation**' screen will be displayed. Check the details and click the '**Continue**' button.
10. The '**Results**' screen will be displayed, showing how many users and groups have been imported into your Crowd directory.
If the import of any users or groups failed, please check the log files to find out why.

Screenshot: 'Import users from one directory to another'

Directory Importer

1. Import Type
2. Options
3. Confirmation
4. Results

Which directory do you want to copy your users from? And where do you want them to go?

Source Directory: * ▼
The directory to import your users and groups from.

Destination Directory: * ▼
The directory to import your users and groups into.

Overwrite Destination Directory:
Tick this box to delete and replace all details and memberships for users that already exist in the destination directory.

Next Steps

To allow the users to log in to the [integrated application\(s\)](#) via Crowd:

- Map the directory to the application(s), if not already done. See [Mapping a Directory to an Application](#).
- Give the imported groups access to the application(s). See [Specifying which Groups can access an Application](#).

RELATED TOPICS

- [Using the Directory Browser](#)
- [Adding a Directory](#)
 - [Configuring an Internal Directory](#)
 - [Configuring an LDAP Directory Connector](#)
 - [LDAP Object Structures](#)
 - [Configuring an SSL Certificate for Microsoft Active Directory](#)
 - [Configuring a Remote Crowd Directory](#)
 - [Configuring a Custom Directory Connector](#)
 - [Configuring a Delegated Authentication Directory](#)
 - [Configuring Azure Active Directory](#)
- [Configuring Caching for an LDAP Directory](#)
- [Using Naive DN Matching](#)
- [Specifying Directory Permissions](#)
- [Importing Users and Groups into a Directory](#)
 - [Importing Users from Atlassian Confluence](#)
 - [Importing Users from Atlassian Jira](#)
 - [Importing Users from Atlassian Bamboo](#)
 - [Importing Users from Jive Forums](#)
 - [Importing Users from CSV Files](#)
 - [Configuring the CSV Importer](#)
 - [Mapping CSV Fields to Crowd Fields](#)
 - [Confirming the CSV Importer Configuration](#)
 - [Viewing the Results of the Import](#)
 - [Importing Users from One Crowd Directory into Another](#)
- [Configuring directories for failover authentication](#)
- [Pruning delegated directories](#)

[Crowd documentation](#)

Configuring directories for failover authentication

Adding an extra user directory for failover authentication means that when the primary directory is unavailable (e.g. due to a connection timeout), Crowd will authenticate your users by trying the next directory from the list. It works like a backup directory for authentication and ensures that your users can log in even if the primary directory is not working.

To add a failover authentication directory:

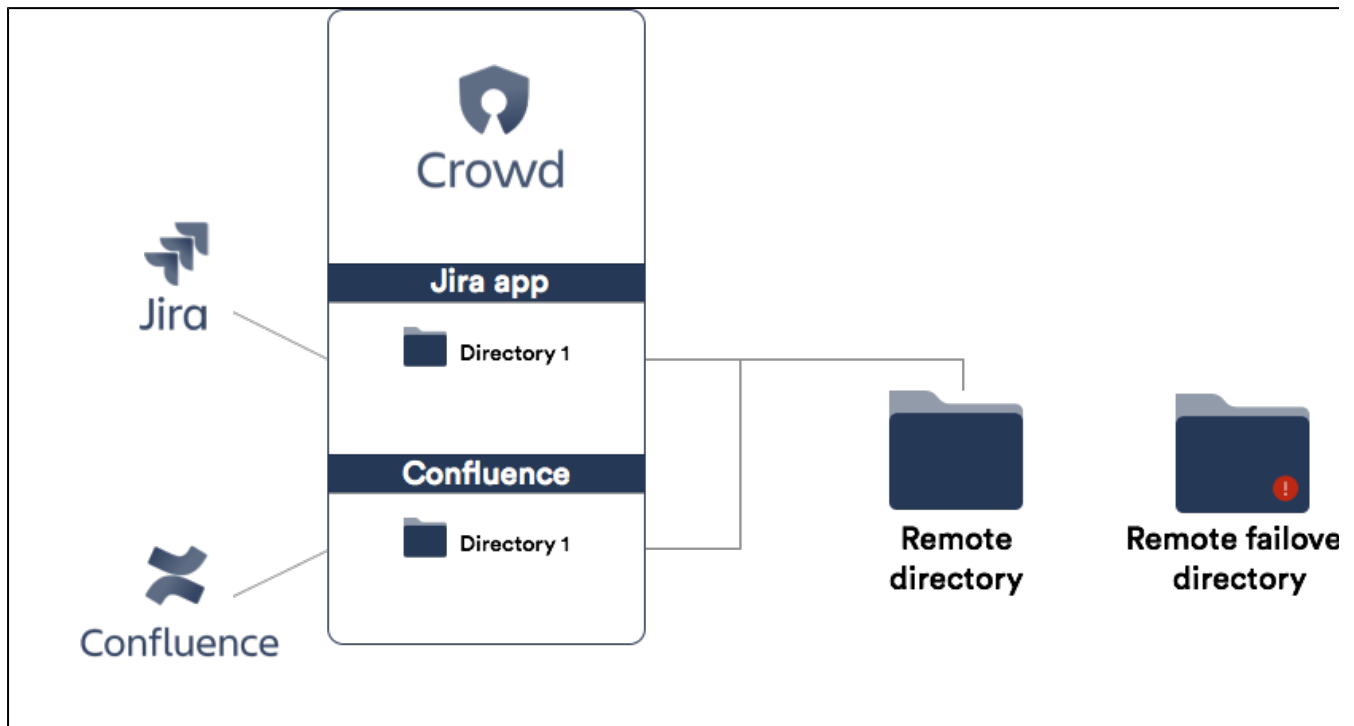
1. Log in to the [Crowd Administration Console](#).
2. Click the **'Directories'** link in the top navigation bar.
3. This will display the [Directory Browser](#). Click the **'Add Directory'** link.
4. This will display the **'Select Directory Type'** screen. Choose the **'Delegated Authentication'** directory type.
For details on how to configure this type of directory, see [Configuring a Delegated Authentication Directory](#).
5. [Map](#) the failover directory to the right application that already uses the primary directory.
6. The failover directory will appear at the bottom of the list. Use the blue up-arrow or down-arrow to move it right after the primary directory.

Note

- [Map](#) the failover directory to each application you'd like to use it for.
- Specify the same user access rights for the primary and failover directories (either all users can log in, or only specific groups.)

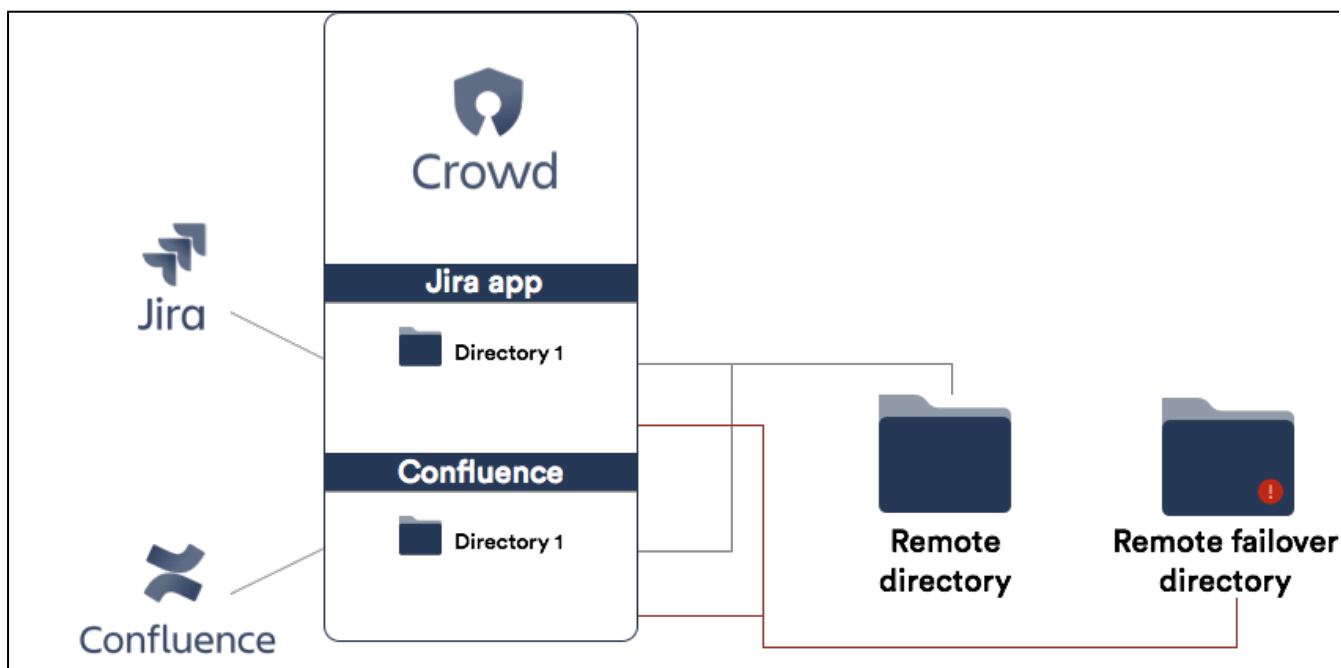
Example

The following example shows a simple scenario, where a failover directory is added to Crowd.



1. A remote directory *Directory 1* is defined in Crowd.
2. Two applications are using this directory Jira and Confluence.
3. A replica of this directory is in your infrastructure, but it hasn't been added to Crowd yet.

You define an extra directory in Crowd that points to the replica. If *Directory 1* is down, Crowd will use the replica to authenticate your users. Your setup should then look like in the image below:



Pruning delegated directories

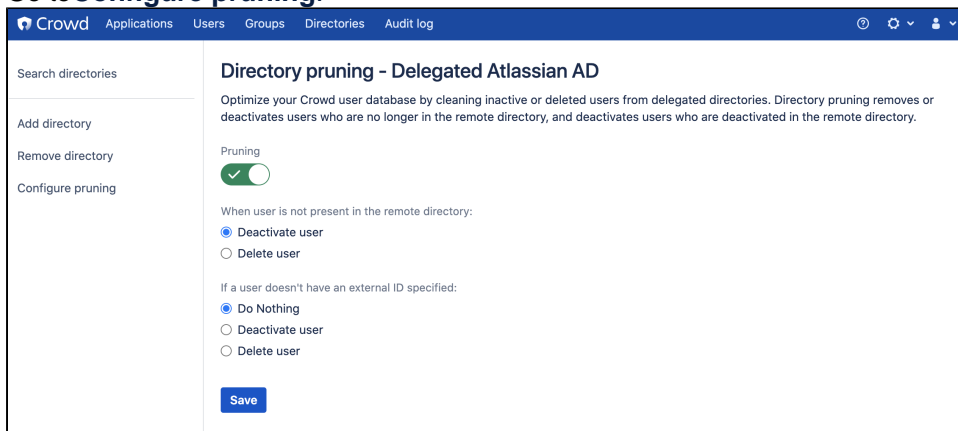
Optimize your user database by cleaning inactive users from your delegated directories. After configuring pruning for a delegated directory, Crowd will periodically check if the directory contains any users who have been deactivated or removed from the remote directory.

By default such users will be deactivated in Crowd. It is also possible to enable "hard delete" mode, in which users who have been deleted in the remote directory will also be deleted in Crowd.

Delegated directory pruning feature is Crowd's native feature starting from version **4.3**. If you use version of Crowd lower than 4.3, you must install the Delegated Directory Pruning plugin from [Atlassian Marketplace](#) to be able to use this functionality.

To enable pruning for your delegated directory

1. Go to a delegated directory for which you want to enable pruning
2. Go to **Configure pruning**.




3. Enable pruning by selecting the **Pruning** toggle
4. Choose the pruning mode:
 - **Deactivate user**- when a user is deactivated or deleted in a remote directory, it is also disabled in Crowd.
 - **Delete user**- when a user is deleted in a remote directory, it is also deleted in Crowd.

 User deactivated in remote directory will be disabled in Crowd.

5. Choose what pruning should do with users that don't have an external ID specified:

This is a user that was manually added to delegated directory by **Add user** in Crowd or using Crowd's REST API. Such user will not have an `externalID` because they have no link to the corresponding user in the remote directory.

- **Do Nothing**-don't do anything with a user that doesn't have an external ID.
 - **Deactivate user**- deactivate a user that doesn't have an external ID.
 - **Delete user** - delete a user that doesn't have an external ID.
6. Click **Save**.

 Changes made to users in delegated directories are reflected in Crowd with a delay. Pruning is being run periodically - everyday at 03:00 AM server time.

Managing Applications

Crowd integrates and provisions applications. Once [defined](#), an application is [mapped](#) to a directory(s), whose users are then [granted access](#) to the application. Note that an application can only communicate with Crowd when the application uses a known [host address](#).

Using the Application Browser

This page describes the Application Browser and gives an overview of the types of application you may find in Crowd.

On this page:

- [About the Application Browser](#)
- [About Applications](#)
 - [Default Applications](#)
 - [Application Types](#)

About the Application Browser

The Application Browser allows you to view and search for integrated applications.





To use the Application Browser,

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click **Applications**.
This will display the Application Browser, showing all the applications that exist in your Crowd system. You can refine your search by specifying a **Name** (note that this is case sensitive), or **Active/Inactive** applications.
3. To view or edit an application's details, click the application name or the **View** link next to the specific application.

Screenshot: Application Browser

Applications

Search Active Results per page [Search](#) [Reset](#)

Name	Description
 bulldog-staging	
 cdcdog-internal-users	Syncs internal user directory to cdcdog
 crowd	Crowdog
 google-apps	Google Applications Connector

Screenshot: Example of an application's details

crowd

[Details](#) [Directories & groups](#) [Administrators](#) [Users](#) [Permissions](#) [Remote addresses](#) [Authentication test](#)

Options

Name
The unique name that the application will use to authenticate against the Crowd framework as a client.

Description
A short description of the application. Often a URL is helpful.

Application type **Crowd**
 Active

Conception **04 Apr 2016, 10:42:20**

Last modified **25 Jan 2019, 14:48:38**

Password [Change password](#)
To set a new password, enter the password and confirm.

Confirm password

[Update](#) [Cancel](#)

About Applications

Crowd integrates and provisions applications. Once [defined](#), an application is [mapped](#) to a directory(s), whose users are then [granted access](#) to the application. Note that an application can only communicate with Crowd when the application uses a known [host address](#).









Default Applications

When you first use the Application Browser, you will see a number of default applications, i.e. applications that are shipped with your Crowd installation:

- **'crowd'** This is the [Crowd Administration Console](#). The Crowd Administration Console is itself a web application that is provisioned by Crowd. The 'crowd' application is mapped to the default directory which you defined during [setup](#), and can be accessed by members of the `crowd-administrators` [group](#).
- **'crowd-openid-server'** This is the CrowdID application which you (optionally) configured during [setup](#). It allows you to provide OpenID services to your users. For details please see the [CrowdID Administration Guide](#) and the [CrowdID User Guide](#). To access CrowdID, go to `http://localhost:8095/openidserver`.
- **'demo'** This is the 'demo' application which you (optionally) configured during [setup](#). Its main purpose is to provide an example of how to integrate [custom applications](#) with Crowd.
- **'google-apps'** This is the Crowd application connector which allows single sign-on (SSO) to [Google Apps](#). To enable SSO between Crowd-connected applications and Google Apps, you will need to configure the Google Apps connector as described in [Configuring the Google Apps Connector](#).

Application Types

Crowd supports the following application types, as indicated by the application-type icons on the Application Browser:

	<p>This icon marks the Crowd application.</p> <ul style="list-style-type: none"> • There will be one, and only one, application of this type. • You cannot rename, deactivate or delete this application.
	This marks a Bamboo server connected to Crowd.
	This marks a Confluence server connected to Crowd.
	This marks a Crucible server connected to Crowd.
	This marks a Fisheye server connected to Crowd.
	This marks a Jira server connected to Crowd.
	<p>These are the 'remote' applications, which you can add to Crowd as described in Adding an Application. This application type does not include plugin applications. You can rename, deactivate or delete remote applications.</p>
	<p>The 'plugin' applications are implemented as plugins to Crowd.</p> <ul style="list-style-type: none"> • An example of a plugin application is the Google Apps connector, which is shipped with your Crowd installation. To activate the Google Apps connector, you need to configure it. • In future, other plugin applications may become available. You will then be able to install them by copying the relevant jars to your Crowd installation. See Important directories and files. • All installed plugin applications are created automatically when the Crowd server starts up, by loading them from the relevant folders in your Crowd Home directory. • You cannot rename or delete plugin applications. You can deactivate them.

RELATED TOPICS

- [Using the Application Browser](#)
- [Adding an Application](#)
- [Configuring the Google Apps Connector](#)
- [Mapping a Directory to an Application](#)
- [Effective memberships with multiple directories](#)
- [Specifying an Application's Address or Hostname](#)
- [Testing a User's Login to an Application](#)

- [Enforcing Lower-Case Usernames and Groups for an Application](#)
- [Managing an Application's Session](#)
- [Deleting or Deactivating an Application](#)
- [Configuring Caching for an Application](#)
- [Overview of SSO](#)
- [Configuring Options for an Application](#)
- [Enabling OpenID client app](#)
- [Allowing applications to create user tokens](#)
- [Disabling the OpenID client app](#)
- [Configuring how users log in](#)

[Crowd documentation](#)

Adding an Application

This page gives an overview of the process for adding an application to Crowd. It refers to the application-specific pages for detailed instructions.

Overview

There are two main steps to integrating an application with Crowd:

- **Step 1. Configure Crowd to talk to your application** that is, set up a directory in Crowd containing your users and groups, and then add the application to Crowd using the 'Add Application' wizard, as described [below](#). The application will now be allowed to authenticate against Crowd.
- **Step 2. Configure the application to talk to Crowd** that is, install the Crowd client into the application and configure the application to forward users' authentication and security requests to Crowd.

Detailed Instructions

Please refer to the details for your specific application:

- [Integrating Crowd with Atlassian Bamboo](#)
- [Integrating Crowd with Atlassian Confluence](#)
- [Integrating Crowd with Atlassian CrowdID](#)
- [Integrating Crowd with Atlassian Crucible](#)
- [Integrating Crowd with Atlassian FishEye](#)
- [Integrating Crowd with Atlassian Jira](#)
- [Integrating Crowd with Atlassian Bitbucket Server](#)
- [Integrating Crowd with Acegi Security](#)
- [Integrating Crowd with Jive Forums](#)
- [Integrating Crowd with Spring Security](#)
- [Integrating Crowd with a Custom Application](#)
- [Integrating Crowd with Atlassian HipChat](#)

Using Crowd's 'Add Application' Wizard

1. Log in to the [Crowd Administration Console](#).
2. Consider whether you need to add any directories, users and groups. If so, see the [detailed instructions](#) for your application.
3. In the top navigation bar, click **Applications**.
4. Click **Add Application**.
5. Complete the form:

Attribute	Description
Application Type	This is used to define the type of application you are adding to Crowd. If you cannot see a matching application type, please choose the 'Generic Application' option.
Name	The username which the application will use when it authenticates against the Crowd framework as a client. This value must be unique, i.e. it cannot be used by more than one application client.
Description	A short description of the application. Note: A URL is often helpful.
Password	The password which the application will use when it authenticates against the Crowd framework as a client.
Confirm Password	Retype the same password as above, to confirm it.

Add application

Details Connection Directories Authorisation Confirmation

Application type* Confluence

Are you connecting JIRA to Crowd, or perhaps Confluence or Bamboo?

Name* MyConfluence

The unique name that the application will use to authenticate against the Crowd framework as a client.

Description

A short description of the application. Often a URL is helpful.

Password*

Confirm password*

Next Cancel

6. Click **Next**.
7. Enter the connection details for your application, as described in the table below.

Attribute	Description
URL	The URL of your application. For example this may be http://jira.atlassian.com . Remember to include the port, if you are not using a proxy. After entering the URL for the application, you can click Resolve IP Address . Crowd will attempt to resolve the IP address for your application.
Remote IP Address	This is the IP address of the server where your application exists. To help you work this out, you can click Resolve IP Address once you have entered a URL.

Add application - confluence

Details Connection Directories Authorisation Confirmation

URL* [input] Resolve IP Address

The URL where this application resides, e.g. https://jira.atlassian.com

Remote IP address* [input]

The IP address for the application, e.g. 127.0.0.1

Next Cancel

8. Click **Next**.
9. Now select one or more directories that this application can use for authentication and authorization:

Add application - myconfluence

Details Connection Directories Authorisation Confirmation

Please select the directories you are going to let this application use for authentication and authorisation.

- AtlassianDir
Crowd Internal Directory
- Atlassian AD
Microsoft Active Directory
- Atlassian JIRA Server Crowd Server
Crowd Internal Directory
- Atlassian Staff ID
OpenLDAP (Read-Only Posix Schema)

10. Click **Next**.

11. In the 'Authorization' step you specify the users who are authorized to access the application.

For each directory, you should do *one* of the following:

- Select **Allow all users to authenticate**, to grant application access to all users defined in the directory, or
- Select one or more groups you wish to have access, and click **Add Group** to add each group to the list. The **Add Group** button appears when you select a group.
 - To remove a group from the list after adding it, click **remove** next to the authorized groups' names.

12. Click **Next**.

13. Now confirm the details for your application.

Check the details of your application.

- If you need to change anything, you can click the tabs to go back to one of the steps in the 'Add Application' wizard.
- When you are happy with the details, click **Add Application**.

14. After completing the 'Add Application' wizard, remember to configure the application as described in the detailed instructions:

- [Integrating Crowd with Atlassian Bamboo](#)
- [Integrating Crowd with Atlassian Confluence](#)
- [Integrating Crowd with Atlassian CrowdID](#)
- [Integrating Crowd with Atlassian Crucible](#)
- [Integrating Crowd with Atlassian FishEye](#)
- [Integrating Crowd with Atlassian Jira](#)
- [Integrating Crowd with Atlassian Bitbucket Server](#)
- [Integrating Crowd with Acegi Security](#)
- [Integrating Crowd with Jive Forums](#)
- [Integrating Crowd with Spring Security](#)

- [Integrating Crowd with a Custom Application](#)
- [Integrating Crowd with Atlassian HipChat](#)

Community application connectors

You may also be interested in the [Crowd plugins](#) created by community developers. (Please check under 'Plugin Details' for each plugin to see if the plugin is supported by Atlassian.)

RELATED TOPICS

- [Using the Application Browser](#)
- [Adding an Application](#)
- [Configuring the Google Apps Connector](#)
- [Mapping a Directory to an Application](#)
- [Effective memberships with multiple directories](#)
- [Specifying an Application's Address or Hostname](#)
- [Testing a User's Login to an Application](#)
- [Enforcing Lower-Case Usernames and Groups for an Application](#)
- [Managing an Application's Session](#)
- [Deleting or Deactivating an Application](#)
- [Configuring Caching for an Application](#)
- [Overview of SSO](#)
- [Configuring Options for an Application](#)
- [Enabling OpenID client app](#)
- [Allowing applications to create user tokens](#)
- [Disabling the OpenID client app](#)
- [Configuring how users log in](#)

[Crowd documentation](#)

Integrating Crowd with Atlassian Bamboo

This page tells you how to connect Atlassian's [Bamboo integration server](#) to one or more directory servers through Crowd.


Currently Crowd supports centralized authentication and single sign-on for Bamboo versions 1.2.2 and later.

Please check that this documentation applies to your version of Crowd

Please check the Crowd release number in this documentation against your version of Crowd. If you are using a different version of Crowd, you can find the appropriate documentation under 'Previous Versions' on the [Crowd documentation homepage](#).

- [Prerequisites](#)
- [Step 1. Configuring Crowd to Talk to Bamboo](#)
 - [1.1 Prepare Crowd's Directories/Groups/Users for Bamboo](#)
 - [1.2 Define the Bamboo Application in Crowd](#)
 - [1.3 Specify which Users can Log In to Bamboo](#)
 - [1.4 Specify the Address from which Bamboo can Log In to Crowd](#)
- [Step 2. Configuring Bamboo to Talk to Crowd](#)
 - [2.1 Install the Crowd Client Libraries into Bamboo](#)
 - [2.2 Edit Bamboo's crowd.properties file](#)
 - [2.3 Configure Bamboo to use Crowd's Authenticator](#)
 - [2.4 Configure External User Management in Bamboo](#)
 - [2.5 \(Optional\) Enable Single Sign-On](#)
 - [2.6 \(Optional\) Tune the Cache](#)
- [See Crowd in Action](#)

Prerequisites

 Due to incompatible atlassian-user libraries, Bamboo releases prior to 1.2.2 are not compatible with latest version of Crowd. Please upgrade to the latest version of Bamboo before attempting to integrate Crowd.

 Do not deploy multiple Atlassian applications in a single Tomcat container.

Deploying multiple Atlassian applications in a single Tomcat container is **not supported**. We do not test this configuration and upgrading any of the applications (even for point releases) is likely to break it. There are also a number of known issues with this configuration. See [this FAQ](#) for more information.

There are also a number of practical reasons why we do not support deploying multiple Atlassian applications in a single Tomcat container. Firstly, you must shut down Tomcat to upgrade any application and secondly, if one application crashes, the other applications running in that Tomcat container will be inaccessible.

Finally, we recommend not deploying *any other applications* to the same Tomcat container that runs Crowd, especially if these other applications have large memory requirements or require additional libraries in Tomcat's `lib` subdirectory.

1. Download and install Crowd. Refer to the [Crowd installation guide](#) for instructions. We will refer to the Crowd root folder as `CROWD`.
2. Download and install Bamboo (version 1.2.2 or later). Refer to the [Bamboo installation guide](#) for instructions. We will refer to the Bamboo root folder as `BAMBOO`. For the purposes of this document, we will assume that you have used the Bamboo distribution (not EAR-WAR) (ie. the easier) installation method of Bamboo. If you need to install Bamboo as an EAR/WAR, simply explode the EAR/WAR and make the necessary changes as described below, then repackage the EAR/WAR.
3. Run the Bamboo Setup Wizard, as described in the [Bamboo documentation](#). During this setup process, you will define the Bamboo administrator's username and password. It is easier to do this before you integrate Bamboo with Crowd.

4. After you have installed and set up Bamboo, shut Bamboo down before you begin the integration process described below.

Step 1. Configuring Crowd to Talk to Bamboo

1.1 Prepare Crowd's Directories/Groups/Users for Bamboo

1. **Create a Crowd directory:** The Bamboo application will need to authenticate users against a directory configured in Crowd. You will need to set up a directory in Crowd for Bamboo. For more information on how to do this, see [Adding a Directory](#). We will assume that the directory is called *Crowd Bamboo Directory* for the rest of this document. It is possible to assign more than one directory for an application, but for the purposes of this example, we will use *Crowd Bamboo Directory* to house Bamboo users.
2. **Add users and groups:** You can either import them from your Bamboo deployment or add them manually.
 - **Importing users and groups from Bamboo:** If you have an existing Bamboo deployment and would like to import existing users and groups into Crowd, use the Bamboo Importer tool by navigating to **Users > Import Users > Atlassian Importer**. Select 'Bamboo' as the Atlassian Product and the *Crowd Bamboo Directory* as the directory into which Bamboo users will be imported. For details please see [Importing Users from Atlassian Bamboo](#). ⓘ If you are going to import users into Crowd, you need to do this now, before you proceed any further.
 - **Adding users and groups manually:** Bamboo needs an administrative group to exist in the directory in order to access the administration features. You can also create an optional additional group for other users. Create the groups in the *Crowd Bamboo Directory*:
 - `bamboo-admin`
 - `bamboo-user` (*optional*)
See the documentation on [Creating Groups](#) for more information on how to define these groups.
 - Create at least one user in the *Crowd Bamboo Directory* and assign the user(s) to both the `bamboo-user` and the `bamboo-admin` groups. The Crowd documentation has more information on [creating groups](#), [creating users](#) and [assigning users to groups](#).

1.2 Define the Bamboo Application in Crowd

Crowd needs to be aware that the Bamboo application will be making authentication requests to Crowd. We need to add the Bamboo application to Crowd and map it to the *Crowd Bamboo Directory*:

1. Log in to the [Crowd Administration Console](#) and navigate to **Applications > Add Application**.
2. Complete the '**Add Application**' wizard for the Bamboo application. See the [instructions](#). ⓘ The **Name** and **Password** values you specify in the 'Add Application' wizard must match the **application.name** and **application.password** that you will set in the `crowd.properties` file. You can find the `crowd.properties` file in either `Bamboo/webapp/WEB-INF/classes/` (Bamboo 3.1 and earlier) or `$BAMBOO_HOME/xml-data/configuration` (Bamboo 3.2 or later). See Step 2 below.

1.3 Specify which Users can Log In to Bamboo

Once Crowd is aware of the Bamboo application, Crowd needs to know which users can authenticate (log in) to Bamboo via Crowd. As part of the 'Add Application' wizard, you will set up your directories and group authorizations for the application. If necessary, you can adjust these settings after completing the wizard. Below are some examples.

You can either allow entire directories to authenticate, or just particular groups within the directories. In our example, we will allow the `bamboo-user` and `bamboo-admin` groups within the *Crowd Bamboo Directory* to authenticate.

If you are not using a `bamboo-user` group as a security restriction, you will need to set '**Allow all to authenticate**' to 'true' when [mapping the directory](#), otherwise only `bamboo-admin` group members will be able to log in to Bamboo.

1.4 Specify the Address from which Bamboo can Log In to Crowd

As part of the 'Add Application' wizard, you will set up Bamboo's IP address. This is the address which Bamboo will use to authenticate to Crowd. If necessary you can add a hostname, in addition to the IP address, after completing the wizard. See [Specifying an Application's Address or Hostname](#).

Step 2. Configuring Bamboo to Talk to Crowd

Before you begin Step 2

- If you are using **Bamboo 3.2 or later**, please refer to the Bamboo instructions on [Integrating Bamboo with Crowd](#) and skip Step 2 (and all sub-steps) on this page.
- If your Bamboo version is **earlier than 1.2.2**, please upgrade to the latest stable version of Bamboo.

2.1 Install the Crowd Client Libraries into Bamboo

Bamboo needs Crowd's client libraries in order to be able to delegate user authentication to the Crowd application. In some cases, you will need to modify the Bamboo application, which is stored in `BAMBOO/webapp`.

1. Please check your versions of Crowd and Bamboo:
 - If you are using **Bamboo 1.2.2 to 1.2.4**, you will need to update the Bamboo libraries as described in this step below.
 - If you are using **Bamboo 2.0** or later, the Crowd client libraries and `crowd.properties` file are included in the Bamboo 2.0 installation download. Please check if your version of Crowd is the same version as the Crowd client library included in the Bamboo 2.x.x installation download (e.g. Bamboo 2.0 currently includes the client library for Crowd 1.3).
 - If the Crowd library versions are different, you will need to update the Bamboo libraries as described in this step below.
 - If the Crowd library versions are the same, you can skip this step.
 - Please, copy `CROWD/client/crowd-integration-client-X.X.X.jar` into `BAMBOO/atlassian-bamboo/WEB-INF/libdirectory`.
 - If you are using the Crowd WAR distribution, then you will need to get the CROWD client libraries from the Crowd distribution, available on our [download site](#).
 - Copy the Crowd client libraries and configuration files to Bamboo:

Copy From	Copy To
CROWD/client/crowd-integration-client-X.X.X.jar	BAMBOO/webapp/WEB-INF/lib (No need to copy any crowd .jar files over for the latest Bamboo versions (after 4.0) as they already contain the needed jar files to work with Crowd)
CROWD/client/conf/crowd.properties	BAMBOO/webapp/WEB-INF/classes (Bamboo 3.1 and earlier) or \$BAMBOO_HOME/xml-data/configuration (Bamboo 3.2 or later)
CROWD/client/conf/crowd-ehcache.xml	BAMBOO/webapp/WEB-INF/classes (Bamboo 3.1 and earlier) or \$BAMBOO_HOME/xml-data/configuration (Bamboo 3.2 or later)

2. For **Bamboo 1.2.4** only: You will need to remove the `seraph-0.7.23.jar` file from Bamboo's `WEB-INF/lib/` directory and replace it with the following file: <http://repository.atlassian.com/maven2/com/atlassian/seraph/atlassian-seraph/0.10/atlassian-seraph-0.10.jar>
(Note: the 0.10 version of the Seraph JAR is newer than 0.7.23.)

2.2 Edit Bamboo's crowd.properties file

Configure the Bamboo application's properties to determine how Crowd will interact with Bamboo.

1. Edit `crowd.properties` found in `BAMBOO/webapp/WEB-INF/classes` (Bamboo 3.1 and earlier) or `$BAMBOO_HOME/xml-data/configuration`(Bamboo 3.2 or later). Change the following properties:

Key	Value
-----	-------

application.name	bamboo The application.name and application.password must match the Name and Password that you specified when defining the application in Crowd (see Step 1 above).
application.password	The application.name and application.password must match the Name and Password that you specified when defining the application in Crowd (see Step 1 above).
crowd.server.url	http://localhost:8095/crowd/services/ If your Crowd server's port is configured differently from the default (8095), set it accordingly.
session.validationinterval	Set to 0, if you want authentication checks to occur on each request. Otherwise set to the number of minutes between requests to validate if the user is logged in or out of the Crowd SSO server. Setting this value to 1 or higher will increase the performance of Crowd's integration.

You can read more about optional settings in [the crowd.properties file](#).

2.3 Configure Bamboo to use Crowd's Authenticator

Now that the Crowd client libraries exist, we need to configure Bamboo to use them.

1. Edit the `atlassian-user.xml` file (found in `BAMBOO/webapp/WEB-INF/classes` (Bamboo 3.1 and earlier) or `$BAMBOO_HOME/xml-data/configuration` (Bamboo 3.2 or later)) so that the contents of the file is:

```
<atlassian-user>
  <repositories>

    <crowd key="crowd" name="Crowd Repository"/>

  </repositories>
</atlassian-user>
```

2. At this stage, Bamboo is set up for **centralized authentication**. If you wish to enable **single sign-on (SSO)** to Bamboo, refer to [section 2.5](#) of this document.

2.4 Configure External User Management in Bamboo

For Bamboo to integrate successfully with Crowd, Bamboo's '**External User Management**' option needs to be:

- **Checked** if you are using an LDAP directory with Crowd and you don't have write-access in LDAP.
- **Unchecked** if you are using internal Crowd directories, or Crowd with LDAP where you do have write-access.
- **Unchecked** if you are using a [Delegated Authentication](#) directory.

More information:

- Please ignore the wording on some versions of the Bamboo screens, which may imply that you should check this option.
- In later versions of Bamboo, the option will be called '**Read-Only External User Management**'.
- Refer to the [Bamboo documentation](#) for full details of Bamboo's external management configuration.

2.5 (Optional) Enable Single Sign-On

i SSO is optional

Single sign-on (SSO) is optional when integrating Bamboo and other Atlassian products with Crowd. To use centralized authentication *without* SSO, skip the steps below.

To enable single sign-on (SSO), you will configure Bamboo's authentication and access request calls to use Seraph. To configure Seraph-based authentication:

1. Edit the `\BAMBOO\webapp\WEB-INF\classes\seraph-config.xml`
2. Comment out the `authenticator` node :

```
<!--<authenticator class="com.atlassian.bamboo.user.authentication.BambooAuthenticator"/>-->
```

Please, uncomment the `authenticator "com.atlassian.crowd.integration.seraph.v25.BambooAuthenticator"`:

```
<!--
If you're authenticating against a Crowd server you can use this authenticator for single sign-on.
Enable it after configuring your Crowd properties through user management and restart Bamboo.
It does not support Crowd property changes at runtime. If you need to switch back to local users,
revert the change and restart Bamboo again.
-->
<authenticator class="com.atlassian.crowd.integration.seraph.v25.BambooAuthenticator"/>
```

Bamboo's authentication and access request calls will now be performed using Seraph.

2.6 (Optional) Tune the Cache

When using the atlassian-user and Crowd framework together with Bamboo, it is highly recommended that caching be enabled. Multiple redundant calls to the atlassian-user framework are made on any given request. These results can be stored locally between calls by enabling caching via the [Crowd Options menu](#). (Note that this caching in the Crowd application is enabled by default.)

Bamboo will obtain all necessary information for the period specified by the cache configuration - see [Configuring Caching for an Application](#). If a change or addition occurs in Crowd to users, groups and roles, these changes will not be visible in Bamboo until the cache expires for that specific item (i.e. for the particular user, group or role).

i The default value for the application cache is 5 minutes (300 seconds). To increase the performance of your application, consider changing the cache value to one or two hours (3600 or 7200 seconds).

See Crowd in Action

Welcome to Bamboo with Crowd!

- Users belonging to the `bamboo-user` group should now be able to log in to Bamboo. Try adding a user to the group using Crowd you should be able to log in to Bamboo using this newly created user. That's **centralized authentication** in action!
- If you have enabled SSO, you can try adding the *Crowd Bamboo Directory* and `bamboo-admin` group to the *crowd* application (see [Mapping a Directory to an Application](#) and [Specifying which Groups can access an Application](#)). This will allow Bamboo administrators to log in to the [Crowd Administration Console](#). Try logging in to Crowd as a Bamboo administrator, and then point your browser at Bamboo. You should be logged in as the same user in Bamboo. That's **single sign-on** in action!

Integrating Crowd with Atlassian Confluence

Atlassian's popular [Confluence wiki](#) can quickly be configured to use [Crowd](#) for user and group management.

Compatibility of Confluence and Crowd Versions

Please ensure that your Crowd and Confluence versions are compatible:

Confluence Version	Supported Crowd Version	Notes
Lower than 2.6.2	N/A	Confluence does not support Crowd - please upgrade Confluence .
Between 2.6.2 and 2.7.4	1.2 and later	Confluence 2.6.1 is not supported - the earliest supported version is 2.6.2
Between 2.8 and 3.4.8	Between 1.3.2 and 2.2.7	In Confluence 2.8, the interface for <code>atlassian-user</code> changed. Crowd 1.3.2 is the earliest version to support this change. Note: As per CWD-2542 - Remove atlassian-user-1 implementation CLOSED , <code>atlassian-user</code> support was removed as per Crowd 2.3. Thus, Crowd 2.2.7 is the latest version that will run with Confluence 3.4.8.
Confluence 3.5 and above	Crowd 2.1 or Later	In Confluence 3.5 and above, the communication between Confluence and Crowd has been changed from SOAP to REST.

Prerequisites

 Do not deploy multiple Atlassian applications in a single Tomcat container.

Deploying multiple Atlassian applications in a single Tomcat container is **not supported**. We do not test this configuration and upgrading any of the applications (even for point releases) is likely to break it. There are also a number of known issues with this configuration. See [this FAQ](#) for more information.

There are also a number of practical reasons why we do not support deploying multiple Atlassian applications in a single Tomcat container. Firstly, you must shut down Tomcat to upgrade any application and secondly, if one application crashes, the other applications running in that Tomcat container will be inaccessible.

Finally, we recommend not deploying *any other applications* to the same Tomcat container that runs Crowd, especially if these other applications have large memory requirements or require additional libraries in Tomcat's `lib` subdirectory.

1. Download and install Crowd. Refer to the [Crowd installation guide](#) for instructions. We will refer to the Crowd root folder as `CROWD`.
2. Download and install Confluence (version 2.6.2 or later). Refer to the [Confluence installation guide](#) for instructions. We will refer to the Confluence root folder as `CONFLUENCE`. For the purposes of this document, we will assume that you have used the Crowd distribution (not EAR-WAR) (i.e. the easier) installation method of Confluence. If you need to install Confluence as an EAR/WAR, simply explode the EAR/WAR and make the necessary changes as described below, then repackage the EAR/WAR.
3. Run the Confluence Setup Wizard, as described in the [Confluence documentation](#). During this setup process, you will define the Confluence administrator's username and password. It is easier to do this before you integrate Confluence with Crowd.
4. After setting up Confluence, shut down Confluence before you begin the integration process described below.

Step 1. Configuring Crowd to Talk to Confluence

1.1 Prepare Crowd's Directories/Groups/Users for Confluence

The Confluence application will need to authenticate users against a directory configured in Crowd. You will need to set up a directory in Crowd for Confluence. For more information on how to do this, see [Adding a Directory](#). We will assume that the directory is called *Confluence Directory* for the rest of this document. It is possible to assign more than one directory for an application, but for the purposes of this example, we will use *Confluence Directory* to house Confluence users.


Confluence also requires particular groups to exist in the directory in order to authenticate users. You will need to create two groups in the *Confluence Directory*:

1. `confluence-users`
2. `confluence-administrators`

See the documentation on [Creating Groups](#) for more information on how to define these groups.

You also need to ensure that the *Confluence Directory* contains at least one user who is a member of both groups. Choose one of the two options below:

- If you have an existing Confluence deployment and would like to import existing users and groups into Crowd, use the Confluence Importer tool by navigating to **Users > Import Users > Atlassian Importer**. Select '**Confluence**' as the Atlassian product, and the *Confluence Directory* as the directory into which Confluence users will be imported. For details please see [Importing Users from Atlassian Confluence](#).

 If you are going to import users into Crowd, you need to do this now before you proceed any further


OR:

- If you don't wish to import your Confluence users, make sure you use Crowd to create at least one user in the *Confluence Directory* and assign them to both the `confluence-users` and the `confluence-administrators` group. The Crowd documentation has more information on [creating groups](#), [creating users](#) and [assigning users to groups](#).

1.2 Define the Confluence Application in Crowd

Crowd needs to be aware that the Confluence application will be making authentication requests to Crowd. We need to add the Confluence application to Crowd and map it to the *Confluence Directory*:

1. Log in to the [Crowd Administration Console](#) and navigate to **Applications > Add Application**.
2. Complete the '**Add Application**' wizard for the Confluence application. See the [instructions](#).

 The **Name** and **Password** values you specify in the 'Add Application' wizard must match the **application.name** and **application.password** that you will set in the `CONFLUENCE/confluence/WEB-INF/classes/crowd.properties` file. (See Step 2 below.)

1.3 Specify which Users can Log In to Confluence

Once Crowd is aware of the Confluence application, Crowd needs to know which users can authenticate (log in) to Confluence via Crowd. As part of the 'Add Application' wizard, you will set up your directories and group authorizations for the application. If necessary, you can adjust these settings after completing the wizard. Below are some examples.


You can either allow entire directories to authenticate, or just particular groups within the directories. In our example, we will allow the `confluence-users` and `confluence-administrators` groups within the *Confluence Directory* to authenticate.

For details please see [Specifying which Groups can access an Application](#).

1.4 Specify the Address from which Confluence can Log In to Crowd

As part of the 'Add Application' wizard, you will set up Confluence's IP address. This is the address which Confluence will use to authenticate to Crowd. If necessary you can add a hostname, in addition to the IP address, after completing the wizard. See [Specifying an Application's Address or Hostname](#).

Step 2. Configuring Confluence to talk to Crowd

 The instructions for step 2 below apply to Confluence 3.5 or newer. If you use Confluence 3.4 or older, please follow "Step 2" on [Integrating Crowd with Atlassian Confluence 3.4 or earlier](#) instead.

2.1 Add a Crowd Directory in Confluence

Confluence can use Crowd for user authentication simply by adding the '**Atlassian Crowd**' user directory.

1. Log in to Confluence Admin as '**confluence-administrator**'.
2. Click on the '**User Directories**' label of the left bar under the '**Security**' tab.
3. Click '**Add Directory**'. Then select '**Atlassian Crowd**' from the dropdown list. Click '**Next**'.
4. Enter connection parameters and save. Now a new Crowd directory should appear on the user directory list.

For more information on configuring a Crowd remote directory in Confluence, check out the Confluence documentation on [Connecting to Crowd or Jira for User Management](#).

2.2 Enable SSO integration with Crowd (Optional)

1. If Confluence is running, shut it down first.
2. Now, edit the file `CONFLUENCE/confluence/WEB-INF/classes/seraph-config.xml`
Comment out the line:-


```
<!-- <authenticator class="com.atlassian.confluence.user.ConfluenceAuthenticator"/> -->
```

Uncomment the line:-


```
<authenticator class="com.atlassian.confluence.user.ConfluenceCrowdSSOAuthenticator"/>
```

3. Copy the `crowd.properties` file from `CROWD/client/conf/` to `CONFLUENCE/confluence/WEB-INF/classes`.
4. Edit `CONFLUENCE/confluence/WEB-INF/classes/crowd.properties`. Change the following properties:

Key	Value
application.name	confluence The application.name and application.password must match the Name and Password that you specified when defining the application in Crowd (see Step 1 above).
application.password	The application.name and application.password must match the Name and Password that you specified when defining the application in Crowd (see Step 1 above).
crowd.base.url	<code>http://localhost:8095/crowd/</code> If your Crowd server's port is configured differently from the default (i.e. 8095), set it accordingly.

session validation interval	<p>This is the number of minutes between validation requests, when Crowd validates whether the user is logged in to or out of the Crowd SSO server. Set to the required number of minutes between validation requests. The recommended default is 2 minutes. Setting this value to 1 or higher will increase the performance of Crowd's integration.</p> <div style="border: 1px solid #f0e68c; padding: 5px; margin-top: 10px;"> <p> Setting this value to 0 will cause the application to perform authentication checks on each request but can cause poor performance, especially with Crowd 2.1 - Crowd 2.3.2 using REST due to CWD-2646.</p> </div>
-----------------------------	---

You can read more about optional settings in [the crowd.properties file](#).

 It is possible to define multiple user directories in Confluence. However, if you enable Crowd SSO integration, you will only be able to authenticate as users from the Crowd server defined in the `crowd.properties` file.

To log in using a user from another directory, such as the Confluence Internal Directory, you will need to either:

- in the Confluence admin console, go to **User Directories** and disable the **Crowd Server** directory, or
- disable Crowd SSO by reverting back to the default Confluence authenticator.

Steps to disable Crowd SSO:

1. Shut down Confluence if it is currently running
2. Edit the file `CONFLUENCE/confluence/WEB-INF/classes/seraph-config.xml`
3. Uncomment the default Confluence authenticator:

```
<authenticator class="com.atlassian.confluence.user.ConfluenceAuthenticator"/>
```

Comment out the Crowd SSO Authenticator:


```
<!-- <authenticator class="com.atlassian.confluence.user.ConfluenceCrowdSSOAuthenticator"/>
-->
```

Restart Confluence

See Crowd in Action

- Users belonging to the `confluence-users` group should now be able to log in to Confluence.
- Try adding a user to the `confluence-users` group using Crowd you should be able to log in to Confluence using this newly created user. That's **centralized authentication** in action!
- If you have enabled SSO, you can try adding the *Confluence Directory* and `confluence-administrators` group to the `crowd` application (see [Mapping a Directory to an Application](#) and [Specifying which Groups can access an Application](#)). This will allow Confluence administrators to log in to the [Crowd Administration Console](#). Try logging in to Crowd as a Confluence administrator, and then point your browser at Confluence. You should be logged in as the same user in Confluence. That's **single sign-on** in action!

Integrating Crowd with Atlassian Confluence 3.4 or earlier

 This is an alternate step to "Step 2" defined in [Integrating Crowd with Atlassian Confluence](#) for users wanting to integrate Crowd with Confluence 3.4 or earlier.

- If you are using Confluence 3.5 or later, please follow the guide on [Integrating Crowd with Atlassian Confluence](#).
- If you are using Confluence 3.4 or earlier, please complete "Step 1" from [Integrating Crowd with Atlassian Confluence](#) before attempting the alternate "Step 2" below.
- Crowd 2.3.3 or later requires Confluence 3.2.1 at minimum. See also [CWD-2680](#).

Step 2. Configuring Confluence to talk to Crowd


2.1 Install the Crowd Client Library into Confluence

Confluence needs Crowd's client library and configuration file in order to be able to delegate user authentication to the Crowd application. As stated earlier, we will modify the Confluence application by editing the application, which is an exploded WAR stored in `CONFLUENCE/confluence`.

1. If you are using the Crowd WAR distribution, then you will need to get the CROWD client libraries from the Crowd distribution, available on our [download site](#).
2. If you are using the Windows Evaluation distribution of Confluence, please see this page on [how to update the crowd.properties file in Confluence](#).
3. Copy the Crowd client library and configuration file to Confluence:

Copy From	Copy To
CROWD/client/crowd-integration-client-X.X.X.jar	CONFLUENCE/confluence/WEB-INF/lib
CROWD/client/conf/crowd.properties	CONFLUENCE/confluence/WEB-INF/classes

There is no need to copy across anything from `CROWD/client/lib`. All the required libraries from that directory already exist in Confluence versions 2.3 and later.

 Be sure that there is **only one** `crowd-integration-client-x.x.x.jar` file in the lib directory. Otherwise, it would cause library incompatibilities.

A note about older Confluence versions:


Confluence **2.5.6 to 2.6.1** are not compatible with Crowd 1.2 and later. We recommend that you upgrade to Confluence **2.6.2 or later**. If you can not upgrade your Confluence instance, you will need to remove the `seraph-X.X.X.jar` file from Confluence's `<CONFLUENCE-INSTALLATION>/confluence/WEB-INF/lib/seraph-X.X.X.jar` and replace it with the following file:

<http://repository.atlassian.com/maven2/com/atlassian/seraph/atlassian-seraph/0.10/atlassian-seraph-0.10.jar>.

4. Replace Confluence's cache configuration file:

Copy From	Replace File
CROWD/client/conf/crowd-ehcache.xml	CONFLUENCE/confluence/WEB-INF/classes/crowd-ehcache.xml

5. Edit `CONFLUENCE/confluence/WEB-INF/classes/crowd.properties`. Change the following properties:

Key	Value
application.name	confluence The application.name and application.password must match the Name and Password that you specified when defining the application in Crowd (see Step 1 above).
application.password	The application.name and application.password must match the Name and Password that you specified when defining the application in Crowd (see Step 1 above).
crowd.server.url	http://localhost:8095/crowd/services/ If your Crowd server's port is configured differently from the default (i.e. 8095), set it accordingly.
session.validation.interval	This is the number of minutes between validation requests, when Crowd validates whether the user is logged in to or out of the Crowd SSO server. Set to the required number of minutes between validation requests. The recommended default is 2 minutes. Setting this value to 1 or higher will increase the performance of Crowd's integration.  Setting this value to 0 will cause the application to perform authentication checks on each request but can cause poor performance, especially with Crowd 2.1 - Crowd 2.3.2 using REST due to CWD-2646 .

You can read more about optional settings in [the crowd.properties file](#).

2.2 Configure Confluence to use Crowd's Authenticator


Now that the Crowd client libraries exist, we need to configure Confluence to use them.

1. Edit the `CONFLUENCE/confluence/WEB-INF/classes/atlassian-user.xml` file so that the content of the file is:

```
<atlassian-user>
  <repositories>

    <crowd key="crowd" name="Crowd Repository"/>

  </repositories>
</atlassian-user>
```

 Make sure the content of the file is only what is indicated above, otherwise you may get [this error](#)

2. At this stage, Confluence is set up for **centralized authentication**. If you wish to enable **single sign-on (SSO)** or if you are using **Confluence 3.2.1 or later**, take the following steps to ensure that Confluence's authentication and access request calls will be performed using Seraph:

Edit the `CONFLUENCE/confluence/WEB-INF/classes/seraph-config.xml` file. Comment out the authenticator node:

```
<!--<authenticator class="com.atlassian.confluence.user.ConfluenceAuthenticator"/>-->
```

Add a new authenticator, choosing the one relevant to your version of Confluence:

- If you are using Confluence 3.4 or later:

```
<authenticator class="com.atlassian.crowd.integration.seraph.v22.ConfluenceAuthenticator" />
```

- If you are using Confluence 3.3.3 or earlier:

```
<authenticator class="com.atlassian.crowd.integration.seraph.ConfluenceAuthenticator" />
```

2.3 Enable Confluence's External User Management

Once the setup is complete, you may wish to turn 'External User Management' **on** in Confluence. This will prevent Confluence administrators from being able to add or update users. For more information please see the Confluence documentation regarding [External User Management](#).


Note:

- If you are using Confluence **2.6.2 or earlier**, this step is required i.e. you must turn on external user management in Confluence.
- If your [Crowd directory permissions](#) are configured so that Confluence cannot update the Crowd directories, this step is required i.e. you must turn on external user management in Confluence. Otherwise, an error will occur when Confluence attempts to write data into Crowd.
- If you have [imported Confluence users into Crowd](#), you may want to delay turning on 'External User Management' for a week or two, to give users time to reset their passwords. (Because users' passwords are encrypted in Confluence's database, they will not be copied across to Crowd.)

2.4 (Optional) Tune the Cache

Enabling caching on the Crowd server: When using the Atlassian-User and Crowd framework together with Confluence, it is highly recommended that caching be enabled on the Crowd server. Multiple redundant calls to the Atlassian-User framework are made on any given request. These results can be stored locally between calls by enabling caching via the [Crowd Options menu](#). Note that this caching on the Crowd server is enabled by default.

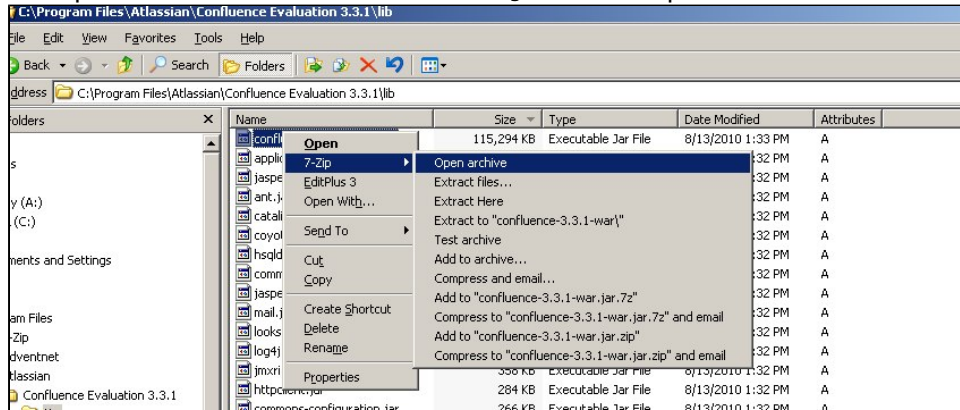
Enabling application caching for Confluence: If application caching is enabled for Confluence, Confluence will obtain all necessary information for the period specified by the cache configuration. See [Configuring Caching for an Application](#). If a change or addition occurs to Crowd users, groups and roles, these changes will not be visible in Confluence until the cache expires for that specific item, i.e. for the particular user, group or role.

 The default period for the application cache is 5 minutes (300 seconds). To increase the performance of your application, consider changing the cache value to one or two hours (3600 or 7200 seconds).

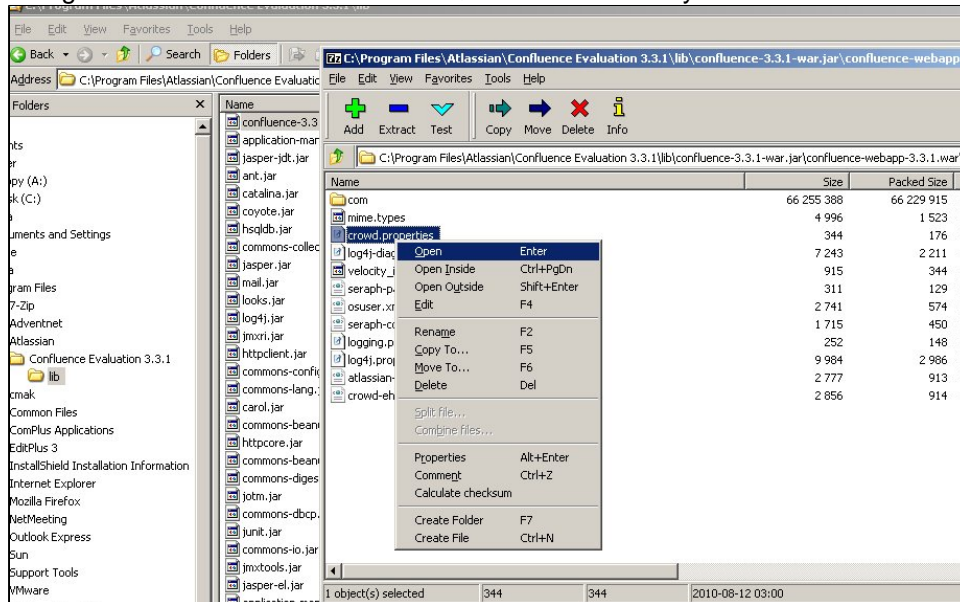
Updating Files in a Confluence Evaluation Distribution

This page tells you how to update the `crowd.properties` file in Confluence, if you are using the Windows Evaluation distribution of Confluence.

1. Download [7-zip](#), a program that you can use to unzip a JAR file.
2. Navigate to your `C:\Program Files\Atlassian\Confluence Evaluation 3.3.1\lib` directory and open the `confluence-3.3.1-war.jar` file in 7zip.



3. Navigate to the relevant `./WEB-INF/classes` directory.



4. Edit the `crowd.properties` file and save the changes to the zip archive.

Integrating Crowd with Atlassian CrowdID

[Atlassian CrowdID](#) is a free add-on to Crowd. It gives administrators a secure way to provide [OpenID](#) accounts for their users.

 When installing Crowd 1.1+ the [Crowd Setup Wizard](#) allows you to install CrowdID with Crowd. If you chose to install CrowdID as part of the Setup Wizard, there is no need for further configuration. The CrowdID server will be up and running at `http://localhost:8095/openidserver`

If you have not already installed CrowdID, follow the instructions below to install it now.

Prerequisites

1. Download and install Crowd. Refer to the [Crowd installation guide](#) for detailed information on how to do this. We will refer to the Crowd root folder as `CROWD`.
2. This guide assumes that CrowdID was NOT installed with the installation of Crowd. If CrowdID was installed using the [Crowd Setup Wizard](#), there is no need for further configuration.

Step 1. Configuring Crowd to Talk to CrowdID

1.1 Prepare Crowd's Directories/Groups/Users for CrowdID

The CrowdID application will need to locate users from a directory configured in Crowd. You will need to set up a directory in Crowd for CrowdID. For information on how to do this, see [Adding a Directory](#). We will assume that the directory is called *CrowdID Directory* for the rest of this document. It is possible to assign more than one directory for an application, but for the purposes of this example, we will use *CrowdID Directory* to house CrowdID users.

CrowdID also requires an administrator group to exist in the directory. You need to ensure that a `crowd-administrators` groups exist in the *CrowdID Directory*. Any user in this group will have CrowdID administrator access.

The Crowd documentation has more information on [creating groups](#), [creating users](#) and [assigning users to groups](#).

1.2 Define the CrowdID Application in Crowd

Crowd needs to be aware that the CrowdID application will be making authentication requests to Crowd. We need to add the CrowdID application to Crowd and map it to the *CrowdID Directory*.

1. Log in to the [Crowd Administration Console](#) and navigate to **Applications > Add Application**.
2. Complete the **'Add Application' wizard** for the CrowdID application.
 - For the **Application type** select **'Generic Application'**
 - For **Name** and **Password**, the values you specify must match the `application.name` and `application.password` that you will set in the `CROWD/crowd-openidserver-webapp/WEB-INF/classes/crowd.properties` file. (See Step 2 below.)

Need more help? See the [full instructions](#) for the Add application wizard.

1.3 Specify which Users can Log In to CrowdID

Once Crowd is aware of the CrowdID application, Crowd needs to know which users can authenticate (log in) to CrowdID via Crowd. As part of the 'Add Application' wizard, you will set up your directories and group authorizations for the application. If necessary, you can adjust these settings after completing the wizard. Below are some examples.

You can either allow entire directories to authenticate, or just particular groups within the directories. In our example, we will allow the entire *CrowdID Directory* to authenticate:

crowd-openid-server

Details

Directories

Groups

Permissions

Remote Addresses

Config Test

Map your user directories to the application. When a user accesses the application, Crowd searches the mapped directories to authenticate the user and determine their group/role membership. To access the application, the user must belong to a directory that allows all to authenticate, or to a group that is mapped to the application.

Directory	Allow All to Authenticate	Action
CrowdID Directory	True <input type="button" value="v"/>	Remove

For details please see [Specifying which Groups can access an Application](#).

1.4 Specify the Address from which CrowdID can Log In to Crowd

As part of the 'Add Application' wizard, you will set up CrowdID's IP address. This is the address which CrowdID will use to authenticate to Crowd. If necessary you can add a hostname, in addition to the IP address, after completing the wizard. See [Specifying an Application's Address or Hostname](#).

Step 2. Configuring CrowdID to Talk to Crowd

Edit `CROWD/crowd-openidserver-webapp/WEB-INF/classes/crowd.properties`. Change the following properties:

Key	Value
application.name	crowd-openid-server The application.name and application.password must match the Name and Password that you specified when you defined the application in Crowd (see Step 1 above).
application.password	The application.name and application.password must match the Name and Password that you specified when you defined the application in Crowd (see Step 1 above).
application.login.url	<code>http://localhost:8095/openidserver</code> The application.login.url should point to the correct host and port of the CrowdID application.
crowd.server.url	<code>http://localhost:8095/crowd/services/</code> If your Crowd server's port is configured differently from the default (i.e. 8095), set it accordingly.
session.validation.interval	This is the number of minutes between validation requests, when Crowd validates whether the user is logged in to or out of the Crowd SSO server. Set this value to 0 if you want authentication checks to occur on each request. Otherwise set to the required number of minutes between validation requests. Setting this value to 1 or higher will increase the performance of Crowd's integration.

You can read more about optional settings in [the crowd.properties file](#).

After editing these properties, you must restart your CrowdID container before the changes will take effect.

See CrowdID in Action

- Go to <http://localhost:8095/openidserver> and log in with any user in the *CrowdID Directory*.

RELATED TOPICS

- [Using the Application Browser](#)
- [Adding an Application](#)
 - [Integrating Crowd with Atlassian Bamboo](#)
 - [Integrating Crowd with Atlassian Confluence](#)
 - [Integrating Crowd with Atlassian Confluence 3.4 or earlier](#)
 - [Updating Files in a Confluence Evaluation Distribution](#)
 - [Integrating Crowd with Atlassian CrowdID](#)
 - [Integrating Crowd with Atlassian Crucible](#)
 - [Integrating Crowd with Atlassian FishEye](#)
 - [Configuring FishEye earlier than 4.0 with Crowd](#)
 - [Integrating Crowd with Atlassian Jira](#)
 - [Integrating Crowd with Atlassian Jira 4.2 or earlier](#)
 - [Integrating Crowd with Atlassian Bitbucket Server](#)
 - [Integrating Crowd with Acegi Security](#)
 - [Integrating AppFuse - a Crowd-Acegi Integration Tutorial](#)
 - [Integrating Crowd with Jive Forums](#)
 - [Jive SSO](#)
 - [Integrating Crowd with Spring Security](#)
 - [Integrating AppFuse - a Crowd-Spring Security Integration Tutorial](#)
 - [Integrating Crowd with a Custom Application](#)
 - [Integrating Crowd with Atlassian HipChat](#)
- [Configuring the Google Apps Connector](#)
- [Mapping a Directory to an Application](#)
 - [Specifying the Directory Order for an Application](#)
 - [Specifying an Application's Directory Permissions](#)
 - [Example of Directory Permissions](#)
 - [Viewing Users in Directories Mapped to an Application](#)
 - [Specifying which Groups can access an Application](#)
 - [Syncing users based on their access rights](#)
- [Effective memberships with multiple directories](#)
- [Specifying an Application's Address or Hostname](#)
- [Testing a User's Login to an Application](#)
- [Enforcing Lower-Case Usernames and Groups for an Application](#)
- [Managing an Application's Session](#)
- [Deleting or Deactivating an Application](#)
- [Configuring Caching for an Application](#)
- [Overview of SSO](#)
- [Configuring Options for an Application](#)
- [Enabling OpenID client app](#)
- [Allowing applications to create user tokens](#)
- [Disabling the OpenID client app](#)
- [Configuring how users log in](#)

[Crowd documentation](#)

Integrating Crowd with Atlassian Crucible

You can use Crowd to provide external authentication and authorization for Atlassian's [Crucible](#) code review tool.

Crucible and FishEye

When you purchase and install Crucible, you may also purchase Atlassian's [FishEye](#) source-repository viewer. If you have both FishEye and Crucible, they will share a common authentication mechanism and integration with Crowd. Crucible and FishEye will authenticate to Crowd using the same application name and password. See [Integrating Crowd with Atlassian FishEye](#). If you have Crucible only (available from **Crucible 1.6**), you will need to set up the Crowd directory and application in the same way, following the instructions in [Integrating Crowd with Atlassian FishEye](#).

Prerequisites

1. Download and install Crowd. Refer to the [Crowd installation guide](#) for detailed information on how to do this. We will refer to the Crowd root folder as CROWD.
2. Download and install Crucible. Refer to the [Crucible Installation Guide](#) for detailed information on how to do this.
3. Follow the instructions on [integrating Crowd with FishEye](#).
For **Crucible versions 1.2.x and later**, refer to the instructions for FishEye 1.4. For **Crucible 1.1.x and earlier**, refer to the the instructions for FishEye 1.3.

Configure Authorization in Crucible Projects (If Required)

Optionally, you can now use the Crowd users and/or groups in the permission schemes for your Crucible projects. If you have created groups in the Crowd directory which is mapped to your FishEye application (see [Integrating Crowd with Atlassian FishEye](#)), the Crowd groups can be seen in Crucible.

Please refer to the Crucible documentation for instructions on:

- Creating projects in Crucible ([here](#)).
- Creating permission schemes and assigning them to users and/or groups ([here](#)).
- Linking the permission scheme to a Crucible project ([here](#)).

RELATED TOPICS

- [Using the Application Browser](#)
- [Adding an Application](#)
- [Configuring the Google Apps Connector](#)
- [Mapping a Directory to an Application](#)
- [Effective memberships with multiple directories](#)
- [Specifying an Application's Address or Hostname](#)
- [Testing a User's Login to an Application](#)
- [Enforcing Lower-Case Usernames and Groups for an Application](#)
- [Managing an Application's Session](#)
- [Deleting or Deactivating an Application](#)
- [Configuring Caching for an Application](#)
- [Overview of SSO](#)
- [Configuring Options for an Application](#)
- [Enabling OpenID client app](#)
- [Allowing applications to create user tokens](#)
- [Disabling the OpenID client app](#)
- [Configuring how users log in](#)

[Crowd documentation](#)

Integrating Crowd with Atlassian FishEye

You can use Crowd to provide external authentication and authorization for Atlassian's [FishEye](#) source-repository viewer.

Crowd supports centralized authentication and single sign-on (SSO) for **FishEye versions 1.3.1 and later**.

Crucible and FishEye

If you are using Atlassian's [Crucible](#) code review tool, you will need to follow the instructions below on integrating Crowd with FishEye. If you have the standalone version of Crucible without FishEye (available from **Crucible 1.6**), please follow the instructions below to set up the Crowd directory and application for Crucible instead of FishEye. If preferred, you can change the name of your Crowd application and directory to 'Crucible' rather than 'FishEye'. Then follow the further instructions to [integrate Crowd with Crucible](#).

On this page:

- [Prerequisites](#)
- [Step 1. Configuring Crowd to talk to FishEye](#)
 - [1.1 Prepare Crowd's directories/groups/users for FishEye](#)
 - [1.2 Define the FishEye application in Crowd](#)
 - [1.3 Specify which users can log in to FishEye](#)
 - [1.4 Specify the address from which FishEye can log in to Crowd](#)
- [Step 2. Configuring FishEye to talk to Crowd](#)
- [Next step for Crucible users](#)

Prerequisites

1. Download and install Crowd. Refer to the [Crowd installation guide](#) for detailed information on how to do this. We will refer to the Crowd root folder as `CROWD`.
2. Download and install FishEye. Refer to the [FishEye Installation Guide](#) for detailed information on how to do this. We will refer to the FishEye root folder as `FISHEYE`.
If you have the standalone version of Crucible (available from **Crucible 1.6**), there is no need to download or install FishEye.
3. After FishEye is set up, make sure FishEye is not running when you begin the integration process described below.

Step 1. Configuring Crowd to talk to FishEye

1.1 Prepare Crowd's directories/groups/users for FishEye

The FishEye application will need to authenticate users against a directory configured in Crowd. You will need to set up a directory in Crowd for FishEye. For more information on how to do this, see [Adding a Directory](#). We will assume that the directory is called *FishEye Directory* for the rest of this document. It is possible to assign more than one directory for an application, but for the purposes of this example, we will use *FishEye Directory* to house FishEye users.

If you wish to use Crowd groups to control access to your FishEye repositories, you should set up your groups in Crowd. See the documentation on [Creating Groups](#) for more information on how to define these groups.

Use Crowd to create at least one user in the *FishEye Directory*. If you are using groups, assign your user(s) to the appropriate groups. The Crowd documentation has more information on [creating users](#) and [assigning users to groups](#).

1.2 Define the FishEye application in Crowd

Crowd needs to be aware that the FishEye application will be making authentication requests to Crowd. We need to add the FishEye application to Crowd and map it to the *FishEye Directory*:

1. Log in to the [Crowd Administration Console](#) and navigate to **Applications > Add Application**.

- 2. Complete the 'Add Application' wizard for the FishEye application. See the [instructions](#). The **Name** and **Password** values you specify in the 'Add Application' wizard must match the **Application name** and **Application password** that you will set in FishEye's 'Crowd Authentication Settings' screen. (See Step 2 below.)

1.3 Specify which users can log in to FishEye

Once Crowd is aware of the FishEye application, Crowd needs to know which users can authenticate (log in) to FishEye via Crowd. As part of the 'Add Application' wizard, you will set up your directories and group authorizations for the application. If necessary, you can adjust these settings after completing the wizard. Below are some examples.

You can either allow entire directories to authenticate, or just particular groups within the directories. In our example, we will allow the entire *FishEye Directory* to authenticate:

The screenshot shows a web interface for configuring the 'fisheye' application. It has a header with the name 'fisheye' and a navigation bar with tabs: 'Details', 'Directories', 'Groups', 'Permissions', 'Remote Addresses', and 'Config Test'. Below the tabs is a descriptive text: 'Map your user directories to the application. When a user accesses the application, Crowd searches the mapped directories to authenticate the user and determine their group/role membership. To access the application, the user must belong to a directory that allows all to authenticate, or to a group that is mapped to the application.' Below this is a table with columns 'Directory', 'Allow All to Authenticate', and 'Action'. The table contains one row for 'FishEye Directory' with 'True' selected in the dropdown and a 'Remove' link in the action column. At the bottom, there is a search box containing 'Employees', and three buttons: 'Add »', 'Update »', and 'Cancel'.

Directory	Allow All to Authenticate	Action
FishEye Directory	True	Remove

If you wish to authorize specific groups only, please see [Mapping a Directory to an Application](#) and [Specifying which Groups can access an Application](#).

1.4 Specify the address from which FishEye can log in to Crowd

As part of the 'Add Application' wizard, you will set up FishEye's IP address. This is the address which FishEye will use to authenticate to Crowd. If necessary you can add a hostname, in addition to the IP address, after completing the wizard. See [Specifying an Application's Address or Hostname](#).

Step 2. Configuring FishEye to talk to Crowd

[Click here for instructions for older versions of FishEye \(before 4.0.0\).](#)

To set up FishEye to use Crowd authentication, follow the instructions in the [FishEye documentation](#).

If you have groups in the Crowd directory that is mapped to your FishEye application (see Step 1 above), the Crowd groups can be seen in FishEye. You can use those groups to control access to your FishEye repositories.

See [Permissions](#) in the FishEye documentation for details.

Next step for Crucible users

If you are using Atlassian's [Crucible](#) code review tool, please take a look at the further instructions on [integrating Crowd with Crucible](#).

RELATED TOPICS

- [Using the Application Browser](#)
- [Adding an Application](#)
- [Configuring the Google Apps Connector](#)
- [Mapping a Directory to an Application](#)
- [Effective memberships with multiple directories](#)

- [Specifying an Application's Address or Hostname](#)
- [Testing a User's Login to an Application](#)
- [Enforcing Lower-Case Usernames and Groups for an Application](#)
- [Managing an Application's Session](#)
- [Deleting or Deactivating an Application](#)
- [Configuring Caching for an Application](#)
- [Overview of SSO](#)
- [Configuring Options for an Application](#)
- [Enabling OpenID client app](#)
- [Allowing applications to create user tokens](#)
- [Disabling the OpenID client app](#)
- [Configuring how users log in](#)

[Crowd documentation](#)

Configuring FishEye earlier than 4.0 with Crowd

You can use Crowd to provide external authentication and authorization for Atlassian's [FishEye](#) source-repository viewer.

Crowd supports centralized authentication and single sign-on (SSO) for **FishEye versions 1.3.1 up to 3.10**. If you are using FishEye 4.0 or later, see [Integrating Crowd with Atlassian FishEye](#).


Crucible and FishEye

If you are using Atlassian's [Crucible](#) code review tool, you will need to follow the instructions below on integrating Crowd with FishEye. If you have the standalone version of Crucible without FishEye (available from **Crucible 1.6**), please follow the instructions below to set up the Crowd directory and application for Crucible instead of FishEye. If preferred, you can change the name of your Crowd application and directory to 'Crucible' rather than 'FishEye'. Then follow the further instructions to [integrate Crowd with Crucible](#).

On this page:

- [Prerequisites](#)
- [Step 1. Configuring Crowd to talk to FishEye](#)
 - [1.1 Prepare Crowd's directories/groups/users for FishEye](#)
 - [1.2 Define the FishEye application in Crowd](#)
 - [1.3 Specify which users can log in to FishEye](#)
 - [1.4 Specify the address from which FishEye can log in to Crowd](#)
- [Step 2. Configuring FishEye to talk to Crowd](#)
 - [2.1 Change the details of your existing FishEye users](#)
 - [2.2 Configure FishEye to use Crowd's authenticator](#)
 - [2.3 Configure group authorization in FishEye \(if required\)](#)
- [Step 3. Override Crowd default properties \(optional\)](#)
- [Next step for Crucible users](#)

Prerequisites

1. Download and install Crowd. Refer to the [Crowd installation guide](#) for detailed information on how to do this. We will refer to the Crowd root folder as `CROWD`.
2. Download and install FishEye. Refer to the [FishEye Installation Guide](#) for detailed information on how to do this. We will refer to the FishEye root folder as `FISHEYE`.
 -  If you have the standalone version of Crucible (available from **Crucible 1.6**), there is no need to download or install FishEye.
3. After FishEye is set up, make sure FishEye is not running when you begin the integration process described below.

Crowd Client JAR

Please make sure you use the default Crowd client JAR that ships with FishEye. In particular, FishEye is not compatible with the `crowd-integration-client-2.0.7.jar` that is bundled with Crowd 2.0.7. See the [Crowd 2.0.7 Release Notes](#).

Step 1. Configuring Crowd to talk to FishEye

1.1 Prepare Crowd's directories/groups/users for FishEye


The FishEye application will need to authenticate users against a directory configured in Crowd. You will need to set up a directory in Crowd for FishEye. For more information on how to do this, see [Adding a Directory](#). We will assume that the directory is called *FishEye Directory* for the rest of this document. It is possible to assign more than one directory for an application, but for the purposes of this example, we will use *FishEye Directory* to house FishEye users.

If you wish to use Crowd groups to control access to your FishEye repositories, you should set up your groups in Crowd. See the documentation on [Creating Groups](#) for more information on how to define these groups.

Use Crowd to create at least one user in the *FishEye Directory*. If you are using groups, assign your user(s) to the appropriate groups. The Crowd documentation has more information on [creating users](#) and [assigning users to groups](#).

1.2 Define the FishEye application in Crowd

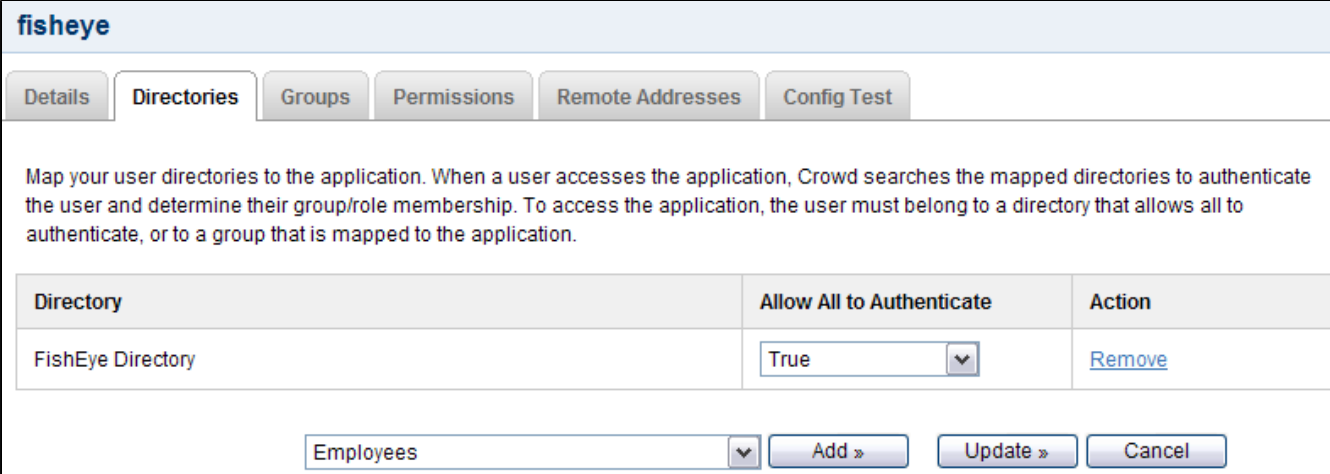
Crowd needs to be aware that the FishEye application will be making authentication requests to Crowd. We need to add the FishEye application to Crowd and map it to the *FishEye Directory*:

1. Log in to the [Crowd Administration Console](#) and navigate to **Applications > Add Application**.
2. Complete the 'Add Application' wizard for the FishEye application. See the [instructions](#).  The **Name** and **Password** values you specify in the 'Add Application' wizard must match the **Application name** and **Application password** that you will set in FishEye's 'Crowd Authentication Settings' screen. (See Step 2 below.)

1.3 Specify which users can log in to FishEye

Once Crowd is aware of the FishEye application, Crowd needs to know which users can authenticate (log in) to FishEye via Crowd. As part of the 'Add Application' wizard, you will set up your directories and group authorizations for the application. If necessary, you can adjust these settings after completing the wizard. Below are some examples.

You can either allow entire directories to authenticate, or just particular groups within the directories. In our example, we will allow the entire *FishEye Directory* to authenticate:



fisheye

Details Directories Groups Permissions Remote Addresses Config Test

Map your user directories to the application. When a user accesses the application, Crowd searches the mapped directories to authenticate the user and determine their group/role membership. To access the application, the user must belong to a directory that allows all to authenticate, or to a group that is mapped to the application.

Directory	Allow All to Authenticate	Action
FishEye Directory	True <input type="button" value="v"/>	Remove


Employees

If you wish to authorize specific groups only, please see [Mapping a Directory to an Application](#) and [Specifying which Groups can access an Application](#).

1.4 Specify the address from which FishEye can log in to Crowd

As part of the 'Add Application' wizard, you will set up FishEye's IP address. This is the address which FishEye will use to authenticate to Crowd. If necessary you can add a hostname, in addition to the IP address, after completing the wizard. See [Specifying an Application's Address or Hostname](#).

Step 2. Configuring FishEye to talk to Crowd

 The instructions below are for **FishEye 1.4.x 3.10**.

2.1 Change the details of your existing FishEye users

If you have an existing FishEye installation with existing built-in users, please do the following for each username in FishEye:

- Change the account type from **built-into crowd**. This is required for the new authorization through Crowd to work properly. For details please see the [FishEye documentation](#).
- Ensure that the username in FishEye is the same as in Crowd. If necessary, rename the user in FishEye. See the [FishEye documentation](#) for details.

2.2 Configure FishEye to use Crowd's authenticator

1. Log in to the FishEye Administration area and click **Authentication** (under 'Security Settings').
2. Click **Edit** under 'JIRA/Crowd Authentication'.
 - ❗ FishEye allows only one authentication method to be configured at any one time. If you have already configured a different authentication source, click **Remove** to remove that authentication method. You will then be presented with the options for different authentication methods one will be the option to set up Crowd authentication.
3. The 'Crowd Authentication Settings' screen will appear, as shown below. Enter the following information:
 - **Application name** The name for the FishEye application you specified in Step 1 above.
 - **Application password** The password you specified in Step 1 above.
 - **Crowd URL** `http://localhost:8095/crowd/services/`
(i) The trailing slash is required.
 - **Auto-add** Select **Create a FishEye user on successful login**(default) to ensure that your Crowd users will be automatically enrolled into FishEye when they first log in via Crowd.
 - **Single sign on (SSO)** Controls whether FishEye should attempt to participate in a single sign on (SSO) environment.
 - ❗ This SSO option is available only with **FishEye 1.5.1** and later.
 - Select **Enabled**(default) if you want FishEye to use Crowd's SSO capability.
 - Select **Disabled** if you want FishEye to use Crowd to check username/passwords and group membership, without participating in SSO. In this mode, FishEye will not read or set `crowd.token` cookies. This is useful in environments where you want FishEye to ignore `crowd.token` cookies set by other Crowd-enabled applications.

Crowd Authentication Settings

Application name:

Application password:

Crowd URL:

Auto-add:

- Create a FishEye user on successful login
- Users must be added to FishEye manually

Single sign on (SSO):

- Enabled
- Disabled

For more information, please see the [FishEye documentation](#) on configuring external authentication sources.

2.3 Configure group authorization in FishEye (if required)

If you have groups in the Crowd directory that is mapped to your FishEye application (see Step 1 above), the Crowd groups can be seen in FishEye. You can use those groups to control access to your FishEye repositories.

See [Permissions](#) in the FishEye documentation for details.

Step 3. Override Crowd default properties (optional)

You set the basic Crowd properties, such as the application name, password and URL, using the FishEye administration screens (described above). You can also fine tune your Crowd integration by overriding the default Crowd properties, such as the session validation interval and SSO cookie name, by manually editing the `config.xml` file in your FishEye installation directory.

To override the default Crowd properties:

1. Shutdown the application.
2. Backup and then open the `config.xml` file in your `<FishEye home directory>` (the folder where you installed FishEye).
3. Add a new `<crowd-properties>` element to the file.
4. Override the default values for any of the Crowd properties (described in [the crowd.properties file](#)) by adding the property in the `<crowd-properties>` section with the desired value.
For example, your `config.xml` file should look like this, if you want to set the `session.validationinterval` to 20 minutes:

```
<config control-bind="127.0.0.1:8059" version="1.0">
  <crowd-properties>
    session.validationinterval=20
  </crowd-properties>
</config>
```

Note that FishEye 2.8, and later, overrides the Crowd defaults with these values:

Property	Crowd Default	FishEye 2.8+
<code>http.timeout</code>	5000 (millisecs)	5000 (millisecs)
<code>socket.timeout</code>	600000 (milliseconds)	20000 (millisecs)

5. Save the file and restart FishEye.

Next step for Crucible users

If you are using Atlassian's [Crucible](#) code review tool, please take a look at the further instructions on [integrating Crowd with Crucible](#).

RELATED TOPICS

- [Using the Application Browser](#)
- [Adding an Application](#)
- [Configuring the Google Apps Connector](#)
- [Mapping a Directory to an Application](#)
- [Effective memberships with multiple directories](#)
- [Specifying an Application's Address or Hostname](#)
- [Testing a User's Login to an Application](#)
- [Enforcing Lower-Case Usernames and Groups for an Application](#)
- [Managing an Application's Session](#)
- [Deleting or Deactivating an Application](#)
- [Configuring Caching for an Application](#)
- [Overview of SSO](#)
- [Configuring Options for an Application](#)
- [Enabling OpenID client app](#)
- [Allowing applications to create user tokens](#)
- [Disabling the OpenID client app](#)
- [Configuring how users log in](#)

[Crowd documentation](#)

Integrating Crowd with Atlassian Jira

Currently Crowd supports centralized authentication and single sign-on for Jira versions 3.7.4 and later.

Please check that this documentation applies to your version of Crowd

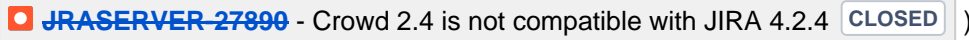
Please check the Crowd release number in this documentation against your version of Crowd. If you are using a different version of Crowd, you can find the appropriate documentation under 'Previous Versions' on the [Crowd documentation homepage](#).

On this page:

- [Compatibility of Jira and Crowd Versions](#)
- [Prerequisites](#)
- [Step 1. Configuring Crowd to talk to Jira](#)
 - [1.1 Prepare Crowd's Directories/Groups/Users for Jira](#)
 - [1.2 Define the Jira Application in Crowd](#)
 - [1.3 Specify which users can log in to Jira](#)
 - [1.4 Specify the address from which Jira can log in to Crowd](#)
- [Step 2. Configuring Jira to talk to Crowd](#)
 - [2.1 Add a Crowd Directory in Jira](#)
 - [2.2 Configure Jira to use Crowd's Authenticator to enable SSO \(Optional\)](#)
 - [2.3 \(Optional\) Disable the Auto-Complete Function in Jira's User Picker](#)
- [See Crowd in Action](#)
- [Known Limitations](#)

Compatibility of Jira and Crowd Versions

Please ensure that your Crowd and Jira versions are compatible:

- If you are using **Jira 4.2** please upgrade to Crowd 2.0.7 or later. (watch out for Crowd 2.4 though: )
- If you are using **Jira 4.3 or later**, please upgrade to Crowd 2.1 or later.
Explanation: With Jira 4.3 and higher, the communication between Jira and Crowd has been changed from SOAP to REST.

Prerequisites

 Do not deploy multiple Atlassian applications in a single Tomcat container.


Deploying multiple Atlassian applications in a single Tomcat container is **not supported**. We do not test this configuration and upgrading any of the applications (even for point releases) is likely to break it. There are also a number of known issues with this configuration. See [this FAQ](#) for more information.

There are also a number of practical reasons why we do not support deploying multiple Atlassian applications in a single Tomcat container. Firstly, you must shut down Tomcat to upgrade any application and secondly, if one application crashes, the other applications running in that Tomcat container will be inaccessible.

Finally, we recommend not deploying *any other applications* to the same Tomcat container that runs Crowd, especially if these other applications have large memory requirements or require additional libraries in Tomcat's `lib` subdirectory.

1. Download and install Crowd. Refer to the [Crowd installation guide](#) for instructions. We will refer to the Crowd root folder as `CROWD`.
2. Download and install Jira (version 3.7.4 or later). Refer to the [Jira installation guide](#) for instructions. We will refer to the Jira root folder as `JIRA`. For the purposes of this document, we will assume that you have used the 'Crowd distribution (not EAR-WAR)' (i.e. the easier and recommended) installation method of Jira. If you need to install Jira as an EAR/WAR, simply explode the EAR/WAR and make the necessary changes as described below, then repackage the EAR/WAR.

3. Run the Jira Setup Wizard, as described in the [Jira documentation](#). During this setup process, you will define the JIRA administrator's username and password. It is easier to do this before you integrate Jira with Crowd.
4. For Jira 4.2 or earlier: after setting up Jira, shut down Jira before you begin the integration process described below.


 If you are using Jira as a User Directory in any other applications such as Fisheye or Confluence these will be inaccessible while Jira is shut down. You can avoid this by configuring these applications to use Crowd prior to integrating Crowd with Jira.

Step 1. Configuring Crowd to talk to Jira

1.1 Prepare Crowd's Directories/Groups/Users for Jira

1. The Jira application will need to locate users from a directory configured in Crowd. You will need to set up a directory in Crowd for Jira. This directory may be any Crowd-configured directory, such as an LDAP directory hooked up to Crowd or a Crowd internal directory. For information on how to do this, see [Adding a Directory](#).


We will assume that the directory is called *Jira Directory in Crowd* for the rest of this document. It is possible to assign more than one directory for an application, but for the purposes of this example, we will use *Jira Directory in Crowd* to house Jira users.

2. Jira also requires particular groups to exist in the directory in order to authenticate users. You need to ensure that these three groups exist in the *Jira Directory in Crowd*:
 - jira-users
 - jira-developers
 - jira-administrators
3. You also need to ensure that the *Jira Directory in Crowd* contains at least one user who is a member of all three groups. You can either:
 - If you have an existing Jira deployment and would like to import existing groups and users into Crowd, use the Jira Importer tool by navigating to **Users > Import Users > Atlassian Importer**. Select 'Jira' as the Atlassian Product and the *Jira Directory in Crowd* as the directory into which Jira users will be imported. For details please see [Importing Users from Atlassian Jira](#).  If you are going to import users into Crowd, you need to do this now before you proceed any further. OR:
 - If you don't wish to import your Jira users, use the Crowd Administration Console to create the three groups, then create at least one user in the *Jira Directory in Crowd* and add them to the three Jira-specific groups (above). The Crowd documentation has more information on [creating groups](#), [creating users](#) and [assigning users to groups](#).


Error will occur in JIRA if the required groups do not exist

JIRA expects that the group names mentioned above will exist. If you need to use different group names, you may want to remove the above pre-existing groups from [Jira's Global Permissions](#). If the above groups do not exist somewhere in Crowd, you will receive an error when you try to remove the groups from Jira's Global Permissions.

1.2 Define the Jira Application in Crowd

 If multiple versions of Jira are being connected to Crowd, ensure you define an application in Crowd for each one

Crowd needs to be aware that the Jira application will be making authentication requests to Crowd. We need to add the Jira application to Crowd and map it to the *Jira Directory in Crowd*.

1. Log in to the [Crowd Administration Console](#) and navigate to **Applications > Add Application**.
2. Complete the '**Add Application**' wizard for the Jira application. See the [instructions](#).  The **Name** and **Password** values you specify in the 'Add Application' wizard must match the **application.name** and **applic**

ation.password that you will set in the `JIRA/atlassian-jira/WEB-INF/classes/crowd.properties` file. (See Step 2 below.)

1.3 Specify which users can log in to Jira

Once Crowd is aware of the Jira application, Crowd needs to know which users can authenticate (log in) to Jira via Crowd. As part of the 'Add Application' wizard, you will set up your directories and group authorizations for the application. If necessary, you can adjust these settings after completing the wizard. Below are some examples.

You can either allow entire directories to authenticate, or just particular groups within the directories. In our example, we will allow the `jira-users`, `jira-developers` and `jira-administrators` groups within the *Jira Directory in Crowd* to authenticate:

The screenshot shows the 'jira' configuration page in the Crowd administration interface. The 'Groups' tab is selected, displaying a table of group mappings. The table has three columns: 'Directory - Group', 'Status', and 'Action'. Three rows are shown, each representing a group within the 'JIRA Directory': 'jira-administrators', 'jira-developers', and 'jira-users'. Each row has a dropdown menu set to 'Active' and a 'Remove' link. Below the table are 'Update »' and 'Cancel' buttons.

Directory - Group	Status	Action
JIRA Directory - jira-administrators	Active	Remove
JIRA Directory - jira-developers	Active	Remove
JIRA Directory - jira-users	Active	Remove

i With this example, only users who are members of the `jira-users`, `jira-developers` and `jira-administrators` groups will be able to authenticate against Jira.

For details please see [Specifying which Groups can access an Application](#).

1.4 Specify the address from which Jira can log in to Crowd

As part of the 'Add Application' wizard, you will set up Jira's IP address. This is the address which Jira will use to authenticate to Crowd. If necessary you can add a hostname, in addition to the IP address, after completing the wizard. See [Specifying an Application's Address or Hostname](#).

Step 2. Configuring Jira to talk to Crowd

! The instructions for step 2 below apply to Jira 4.3 or newer. If you use Jira 4.2 or older, please follow "Step 2" on [Integrating Crowd with Atlassian Jira 4.2 or earlier](#) instead.

2.1 Add a Crowd Directory in Jira

Jira can use Crowd for user authentication simply by adding '**Atlassian Crowd**' as user directory.

1. Login to the administration section of Jira
2. Click on the '**User Directories**' label of the left bar under the '**User management**'.tab.
3. Click '**Add Directory**'. Then select '**Atlassian Crowd**' from the dropdown list. Click '**Next**'.
4. Enter connection parameters and save. If you configure Server URL to use HTTPS, by replacing `http://` with `https://`, communications between Jira and Crowd will be encrypted.
5. Now a new Crowd directory should appear on the user directory list.

Configure Atlassian Crowd Server

JIRA can use Crowd for user management: users, groups and authentication.

Server Settings

Name *

Server URL *
for example, http://www.example.com:8095/crowd/

Application Name *
Used to authenticate to the user management server.

Application Password *
Used to authenticate to the user management server.

Crowd Permissions

Read-Only
Users, groups and memberships are retrieved from the Crowd server and cannot be modified in JIRA.

Read-Write
Modifying users, groups and memberships in JIRA will cause the changes to be applied directly to your Crowd server. Your configured Crowd application will need to have modification permissions on your Crowd server.

Advanced Settings

Enable Nested Groups
If true, groups can contain other groups. Enabling this option may degrade performance.

For more information on configuring a Crowd directory in Jira, check out the Jira documentation on [Connecting to Crowd or Another Jira Server for User Management](#).

2.2 Configure Jira to use Crowd's Authenticator to enable SSO (Optional)

At this stage, Jira is set up for **centralized authentication**. If you wish, you can now enable **single sign-on (SSO)** to Jira. This will ensure that Jira's authentication and access request calls will be performed using Seraph.

Note: if you are migrating/upgrading a Jira instance that already uses Crowd, you will need to merge these files (not overwrite them).

1. If Jira is running, shut it down first.
2. Edit the `JIRA/atlassian-jira/WEB-INF/classes/seraph-config.xml` file. Comment out the `authenticator`node:

```
<!--<authenticator class="com.atlassian.jira.security.login.JiraSeraphAuthenticator" />-->
```

Uncomment the line that contains the new authenticator:

```
<authenticator class="com.atlassian.jira.security.login.SSOSeraphAuthenticator" />
```

3. Copy the `crowd.properties` file from `CROWD/client/conf/` to `JIRA/atlassian-jira/WEB-INF/classes`.
4. Edit `JIRA/atlassian-jira/WEB-INF/classes/crowd.properties`. Change the following properties:

Key	Value
application.name	jira The application name must match the name that you specified when you defined the application in Crowd (see <i>Step 1</i> above).

application.password	The password must match the one that you specified when you defined the application in Crowd (see <i>Step 1</i> above).
crowd.base.url	eg. (http://localhost:8095/crowd/) If your Crowd server's port is configured differently from the default (i.e. 8095), set it accordingly. crowd.base.url must be the same URL used to access Crowd in your Browser.
session.validationinterval	Set to 0, if you want authentication checks to occur on each request. Otherwise set to the number of minutes between request to validate if the user is logged in or out of the Crowd SSO server. Setting this value to 1 or higher will increase the performance of Crowd's integration.



It is possible to define multiple user directories in Jira. However, if you enable SSO integration, you will only be able to authenticate as users from the Crowd server defined in the `crowd.properties` file.

You can read more about optional settings in [the crowd.properties file](#).

2.3 (Optional) Disable the Auto-Complete Function in Jira's User Picker

To improve performance on page-loading in Jira, we recommend that you disable the auto-complete function in Jira's 'User Picker' popup screens. Follow the instructions in the [Jira documentation](#).

More information: In our experience, disabling this feature in Jira helps performance for customers with extremely large user bases. If you leave this feature enabled and have adequate performance results in Jira, feel free to leave it enabled.

See Crowd in Action

- You should now be able to login using users belonging to the `jira-users` group. Try adding a user to the group using Crowd you should be able to login to Jira using this newly created user. That's **centralized authentication** in action!
- If you have enabled SSO, you can try adding the *JIRA Directory in Crowd* and `jira-administrators` group to the `crowd` application (see [Mapping a Directory to an Application](#) and [Specifying which Groups can access an Application](#)). This will allow Jira administrators to log in to the [Crowd Administration Console](#). Try logging in to Crowd as a Jira administrator, and then point your browser at Jira. You should be logged in as the same user in Jira. That's **single sign-on** in action!

Known Limitations

If you are using **Jira 4.2**, a problem occurs in Jira if a user is removed after that user has participated in an issue i.e. if Jira refers to the user. If the user is internally managed by Jira, Jira will prevent the removal of the user but if the user is managed by an external system such as Crowd, Jira will throw a `DataAccessException`. We recommend upgrading Jira or deactivating the user's account by removing them from the `jira-usersgroup`.

If you are using **Jira 4.3 or later**, this problem has been [resolved](#), allowing the removal of users that are externally managed, despite existing data associations. When a user managed by an external system such as Crowd is removed, any user associations in Jira will continue to be associated, with the username acting as a placeholder. This username will not be listed in the User Browser and no profile exists for that user.


RELATED TOPICS

- [Using the Application Browser](#)
- [Adding an Application](#)
 - [Integrating Crowd with Atlassian Bamboo](#)
 - [Integrating Crowd with Atlassian Confluence](#)
 - [Integrating Crowd with Atlassian Confluence 3.4 or earlier](#)


- [Updating Files in a Confluence Evaluation Distribution](#)
- [Integrating Crowd with Atlassian CrowdID](#)
- [Integrating Crowd with Atlassian Crucible](#)
- [Integrating Crowd with Atlassian FishEye](#)
 - [Configuring FishEye earlier than 4.0 with Crowd](#)
- [Integrating Crowd with Atlassian Jira](#)
 - [Integrating Crowd with Atlassian Jira 4.2 or earlier](#)
- [Integrating Crowd with Atlassian Bitbucket Server](#)
- [Integrating Crowd with Acegi Security](#)
 - [Integrating AppFuse - a Crowd-Acegi Integration Tutorial](#)
- [Integrating Crowd with Jive Forums](#)
 - [Jive SSO](#)
- [Integrating Crowd with Spring Security](#)
 - [Integrating AppFuse - a Crowd-Spring Security Integration Tutorial](#)
- [Integrating Crowd with a Custom Application](#)
- [Integrating Crowd with Atlassian HipChat](#)
- [Configuring the Google Apps Connector](#)
- [Mapping a Directory to an Application](#)
 - [Specifying the Directory Order for an Application](#)
 - [Specifying an Application's Directory Permissions](#)
 - [Example of Directory Permissions](#)
 - [Viewing Users in Directories Mapped to an Application](#)
 - [Specifying which Groups can access an Application](#)
 - [Syncing users based on their access rights](#)
- [Effective memberships with multiple directories](#)
- [Specifying an Application's Address or Hostname](#)
- [Testing a User's Login to an Application](#)
- [Enforcing Lower-Case Usernames and Groups for an Application](#)
- [Managing an Application's Session](#)
- [Deleting or Deactivating an Application](#)
- [Configuring Caching for an Application](#)
- [Overview of SSO](#)
- [Configuring Options for an Application](#)
- [Enabling OpenID client app](#)
- [Allowing applications to create user tokens](#)
- [Disabling the OpenID client app](#)
- [Configuring how users log in](#)

[Crowd documentation](#)

Integrating Crowd with Atlassian Jira 4.2 or earlier

 This is an alternate step to "Step 2" defined in [Integrating Crowd with Atlassian Jira](#) for users wanting to integrate Crowd with Jira 4.2 or earlier.

- If you are using Jira 4.3 or later, please follow the guide on [Integrating Crowd with Atlassian Jira](#).
- If you are using Jira 4.2 or earlier, please complete "Step 1" from [Integrating Crowd with Atlassian Jira](#) before attempting the alternate "Step 2" below.

 Use the client libraries from **Crowd 2.2.7** to integrate with Jira 4.2 or earlier even when the Crowd server is more recent. The client libraries from Crowd 2.2.7 remain compatible with later releases of the Crowd server.

Step 2. Configuring Jira to talk to Crowd

2.1 Install the Crowd Client Libraries into Jira

Jira needs Crowd's client libraries in order to be able to delegate user authentication to the Crowd application. As stated earlier, we are going to be modifying the Jira application by editing the application, which is an exploded WAR stored in `Jira/atlassian-jira`.

1. If you are using the Crowd WAR distribution, then you will need to get the CROWD client libraries from the Crowd distribution, available on our [download site](#).
2. Copy the Crowd client libraries and configuration files to Jira:

Copy From	Copy To
CROWD/client/crowd-integration-client-X.X.X.jar	JIRA/atlassian-jira/WEB-INF/lib
CROWD/client/conf/crowd.properties	JIRA/atlassian-jira/WEB-INF/classes

Duplicate Crowd Client libraries in your classpath

Jira should only have a single copy of crowd-integration-client installed. Therefore you need to delete the existing crowd-integration-client-X.X.X.jar file from Jira's WEB-INF/lib directory and replace it with CROWD/client/crowd-integration-client-X.X.X.jar instead of just copying it over. Also, renaming the existing crowd-integration-client jar will not work as Jira will start with duplicate Crowd Client libraries in its classpath.

3. If you are using **Jira 3.11 or earlier**, you will need to remove the `seraph-0.7.12.jar` file from Jira's `WEB-INF/lib/` directory and replace it with the following file:
<http://repository.atlassian.com/maven2/com/atlassian/seraph/atlassian-seraph/0.10/atlassian-seraph-0.10.jar>
4. If you are using **Jira 3.12.2 or earlier**, you will need to update Jira's xfire libraries:
 - Remove the `xfire-all-1.2.1.jar` file from Jira's `WEB-INF/lib/` directory.
 - Copy the following two files from Crowd's `client/lib/` directory to Jira's `WEB-INF/lib/` directory:
 - `xfire-aegis-1.2.6.jar`
 - `xfire-core-1.2.6.jar`
5. Replace Jira's cache configuration file:

Copy From	Replace File
CROWD/client/conf/crowd-ehcache.xml	JIRA/atlassian-jira/WEB-INF/classes/crowd-ehcache.xml

6. Edit `JIRA/atlassian-jira/WEB-INF/classes/crowd.properties`. Change the following properties:

Key	Value
application.name	jira The application name must match the name that you specified when you defined the application in Crowd (see <i>Step 1</i> above).
application.password	The password must match the one that you specified when you defined the application in Crowd (see <i>Step 1</i> above).
crowd.server.url	http://localhost:8095/crowd/services/ If your Crowd server's port is configured differently from the default (i.e. 8095), set it accordingly.
session.validationinterval	Set to 0, if you want authentication checks to occur on each request. Otherwise set to the number of minutes between request to validate if the user is logged in or out of the Crowd SSO server. Setting this value to 1 or higher will increase the performance of Crowd's integration.

You can read more about optional settings in [the crowd.properties file](#).

2.2 Configure Jira to use Crowd's Authenticator

Now that the Crowd client libraries exist, we need to configure Jira to use them.

Note: if you are migrating/upgrading a Jira instance that already uses Crowd, you will need to merge these files (not overwrite them).

1. Edit the Jira config file `JIRA/atlassian-jira/WEB-INF/classes/osuser.xml`. Comment out any existing authentication providers and uncomment/insert the Crowd providers:

```
<!-- This is where JIRA's credentials checking can be configured. For instance, see
http://www.atlassian.com/software/jira/docs/latest/ldap.html -->
<opensymphony-user>
  <authenticator class="com.opensymphony.user.authenticator.SmartAuthenticator" />

  <!-- You will need to uncomment the Crowd providers below to enable Crowd integration -->
  <provider class="com.atlassian.crowd.integration.osuser.CrowdCredentialsProvider"/>
  <provider class="com.atlassian.crowd.integration.osuser.CrowdAccessProvider"/>
  <provider class="com.atlassian.crowd.integration.osuser.DelegatingProfileProvider">
    <property name="provider-1">com.atlassian.crowd.integration.osuser.CrowdProfileProvider<
  /property>
    <property name="provider-2">com.atlassian.jira.user.ExternalEntityJiraProfileProvider</property>
    <property name="provider-2-exclusive-access">true</property>
  </provider>

  <!-- CROWD:START - The providers below here will need to be commented out for Crowd integration -->
  <!--
  <provider class="com.atlassian.core.ofbiz.osuser.CoreOFBizCredentialsProvider">
    <property name="exclusive-access">true</property>
  </provider>

  <provider class="com.opensymphony.user.provider.ofbiz.OFBizProfileProvider">
    <property name="exclusive-access">true</property>
  </provider>

  <provider class="com.opensymphony.user.provider.ofbiz.OFBizAccessProvider">
    <property name="exclusive-access">true</property>
  </provider>
  -->
  <!-- CROWD:END -->

</opensymphony-user>
```


- View `Jira/atlassian-jira/WEB-INF/classes/propertyset.xml`. If there is no entry for the `CrowdPropertySet`, add the following `<propertyset>` item at the end of the file as the last `<propertyset>` item:

```
<propertyset name="crowd" class="com.atlassian.crowd.integration.osuser.CrowdPropertySet" />
```

- At this stage, Jira is set up for **centralized authentication**. If you wish, you can now enable **single sign-on (SSO)** to Jira. This will ensure that Jira's authentication and access request calls will be performed using Seraph. When authentication or access request calls are performed versus the OSUser framework, the Jira stack will call the Crowd providers and `propertyset` implementations.

Edit the `JIRA/atlassian-jira/WEB-INF/classes/seraph-config.xml` file. Comment out the authenticator node:

```
<!--<authenticator class="com.atlassian.jira.security.login.JiraOsUserAuthenticator" />-->
```

Add a new authenticator, choosing the one relevant to your version of Jira:

- If you are using Jira 4.2.x:

```
<authenticator class="com.atlassian.crowd.integration.seraph.v22.JIRAAuthenticator" />
```

- If you are using Jira 4.1.2 or earlier:

```
<authenticator class="com.atlassian.crowd.integration.seraph.JIRAAuthenticator" />
```

2.3 (Optional) Tune the Cache

Enabling caching on the Crowd server: When using the Atlassian-User and Crowd framework together with Jira, it is highly recommended that caching be enabled on the Crowd server. Multiple redundant calls to the Atlassian-User framework are made on any given request. These results can be stored locally between calls by enabling caching via the [Crowd Options menu](#). Note that this caching on the Crowd server is enabled by default.

Enabling application caching for Jira: If application caching is enabled for Jira, Jira will obtain all necessary information for the period specified by the cache configuration. See [Configuring Caching for an Application](#). If a change or addition occurs to Crowd users, groups and roles, these changes will not be visible in Jira until the cache expires for that specific item, i.e. for the particular user, group or role.

i From Jira 3.13, the default cache is two hours. In earlier versions, the default value for the application cache is 5 minutes (300 seconds) increasing this to one or two hours (3600 or 7200 seconds) will improve the performance of your Jira site.

2.4 (Optional) Disable the Auto-Complete Function in Jira's User Picker

To improve performance on page-loading in Jira, we recommend that you disable the auto-complete function in Jira's 'User Picker' popup screens. Follow the instructions in the [Jira documentation](#).

More information: In our experience, disabling this feature in Jira helps performance for customers with extremely large user bases. If you leave this feature enabled and have adequate performance results in Jira, feel free to leave it enabled.

Integrating Crowd with Atlassian Bitbucket Server

You can use Crowd to provide external authentication, and to determine group memberships for authorization, for Atlassian's [Bitbucket Server](#).

On this page:

- [Prerequisites](#)
- [Step 1. Configuring Crowd to talk to Bitbucket Server](#)
- [Step 2. Configuring Bitbucket Server to talk to Crowd](#)

Prerequisites

1. Download and install Crowd. Refer to the [Crowd installation guide](#) for detailed information on how to do this. We will refer to the Crowd root folder as `CROWD`.
2. Download and install Bitbucket Server. Refer to either [Getting started](#) (if running the Bitbucket Server installer) or [Install Bitbucket Server from an archive file](#), for detailed information on how to do this. We will refer to the Bitbucket Server root folder as `Bitbucket`.



Crowd Client JAR

Please make sure you use the default Crowd client JAR that ships with Bitbucket Server.

Step 1. Configuring Crowd to talk to Bitbucket Server

1.1 Prepare Crowd's directories/groups/users for Bitbucket Server

The Bitbucket Server application will need to authenticate users against a directory configured in Crowd. You will need to set up a directory in Crowd for Bitbucket Server. For more information on how to do this, see [Adding a Directory](#). We will assume that the directory is called *Bitbucket Directory* for the rest of this document. It is possible to assign more than one directory for an application, but for the purposes of this example, we will use *Bitbucket Directory* to house Bitbucket Server users.

If you wish to use Crowd groups to control access to your Bitbucket Server projects, you should set up your groups in Crowd. See the documentation on [Creating Groups](#) for more information on how to define these groups.

Use Crowd to create at least one user in the *Bitbucket Directory*. If you are using groups, assign your user(s) to the appropriate groups. The Crowd documentation has more information on [creating users](#) and [assigning users to groups](#).

1.2 Define the Bitbucket Server application in Crowd

Crowd needs to be aware that the Bitbucket Server application will be making authentication requests to Crowd. We need to add the Bitbucket Server application to Crowd and map it to the *Bitbucket Directory*:

1. Log in to the [Crowd Administration Console](#) and navigate to **Applications > Add Application**.
2. Complete the 'Add Application' wizard for the Bitbucket Server application. See the [instructions](#). When prompted for an **Application Type**, choose **Generic Application** if the **Bitbucket Server** option is not available. Note that the **Name** and **Password** values you specify in the 'Add Application' wizard must match those for **Application Name** and **Application Password** that you will set in Bitbucket Server's 'Configure Atlassian Crowd Server' screen (see Step 2 below).

1.3 Specify which users can log in to Bitbucket Server

Once Crowd is aware of the Bitbucket Server application, Crowd needs to know which users can authenticate (log in) to Bitbucket Server via Crowd. As part of the 'Add Application' wizard, you will set up your directories and the group memberships that Bitbucket Server will use for authorization. If necessary, you can adjust these settings after completing the wizard.

You can either allow entire directories to authenticate, or just particular groups within the directories. If you only specific groups to be able to log in, please see [Mapping a Directory to an Application](#) and [Specifying which Groups can access an Application](#).

1.4 Specify the address from which Bitbucket Server can log in to Crowd

As part of the 'Add Application' wizard, you will need to tell Crowd the IP address and/or hostname of the server that Bitbucket Server is connecting from. See [Specifying an Application's Address or Hostname](#).

Step 2. Configuring Bitbucket Server to talk to Crowd

2.1 Connecting Bitbucket Server to Crowd

To set up Bitbucket Server to use Crowd authentication, follow the instructions in [Connect Bitbucket to Crowd](#).


2.2 Configure group permissions in Bitbucket Server (if required)

If you have created groups in the Crowd directory which is mapped to your Bitbucket Server application (see Step 1 above), the Crowd groups can be seen in Bitbucket Server. Now you can set up group permissions for your Bitbucket Server projects. See [Creating projects](#).

2.3 Configure Bitbucket Server to enable SSO with Crowd (optional)

Once the Crowd directory has been set up, you can enable Crowd SSO integration in Bitbucket Server. See [Connecting to Crowd](#) for details.

Integrating Crowd with Acegi Security

 Crowd 2.6 [removed support for Acegi Security \(Upgrade Notes\)](#). Please upgrade to [Spring Security](#) or use an [older release of Crowd](#).

- [Using the Application Browser](#)
- [Adding an Application](#)
- [Configuring the Google Apps Connector](#)
- [Mapping a Directory to an Application](#)
- [Effective memberships with multiple directories](#)
- [Specifying an Application's Address or Hostname](#)
- [Testing a User's Login to an Application](#)
- [Enforcing Lower-Case Usernames and Groups for an Application](#)
- [Managing an Application's Session](#)
- [Deleting or Deactivating an Application](#)
- [Configuring Caching for an Application](#)
- [Overview of SSO](#)
- [Configuring Options for an Application](#)
- [Enabling OpenID client app](#)
- [Allowing applications to create user tokens](#)
- [Disabling the OpenID client app](#)
- [Configuring how users log in](#)

[Crowd documentation](#)

Integrating AppFuse - a Crowd-Acegi Integration Tutorial

[AppFuse](#) provides a sweet starting point for developing web applications. You choose the frameworks, AppFuse generates the skeleton application.

At its core, the web security of AppFuse 2.0.1 and earlier applications relies on the modular and extensible Acegi authentication framework. In this tutorial, we look at a basic integration of Crowd with Acegi, using an application generated by AppFuse.

i If you're working with AppFuse 2.0.2 or later, it uses Spring Security instead of Acegi. Please see our [separate tutorial](#).

i This tutorial assumes you have installed Crowd 1.5.1 or later.

Step 1. Get AppFuse

In this tutorial, we will be using the Struts2-basic archetype to create the project, but the other types should be similar. For more information, consult the AppFuse [quickstart guide](#). In particular, it outlines the database requirements for AppFuse.

1. Create the project.

```
mvn archetype:create -DarchetypeGroupId=org.appfuse.archetypes \
-DarchetypeArtifactId=appfuse-basic-struts \
-DremoteRepositories=http://static.appfuse.org/releases -DarchetypeVersion=2.0 \
-DgroupId=com.mycompany.app -DartifactId=myproject
```

2. Since we will be editing the core Acegi configuration, we will need the full source code of the application.

```
cd myproject
mvn appfuse:full-source
```

3. Build it.

```
mvn clean install
```

4. Run it.

```
mvn jetty:run-war -Dmaven.test.skip
```

5. Play with it.

```
http://localhost:8080/
```

6. Shut it down.

```
ctrl+c
```

Step 2. Let Crowd Know about AppFuse

Add `appfuse` as an application via the Crowd Console. See [Adding an Application](#) for more information.

Step 3. Add the Crowd Acegi Connector to AppFuse

Open up the `pom.xml` and add the Crowd client libraries as a project dependency:

```
<dependencies>
  <dependency>
    <groupId>com.atlassian.crowd</groupId>
    <artifactId>crowd-integration-client</artifactId>
    <version>1.5.1</version>
  </dependency>
  ...
</dependencies>
```

You will also need to create the file `myproject/src/main/resources/crowd.properties`:

```
application.name           appfuse
application.password       password
application.login.url      http://localhost:8095/crowd/
crowd.server.url          http://localhost:8095/crowd/services/
session.isauthenticated    session.isauthenticated
session.tokenkey           session.tokenkey
session.validationinterval 0
session.lastvalidation     session.lastvalidation
```

In particular, the application name and password must match the values defined for the application added in Step 2.

Finally, copy the `STANDALONE/client/conf/crowd-ehcache.xml` to `myproject/src/main/resources/crowd-ehcache.xml`. This file defines the cache properties, such as cache timeouts, used when accessing data from the Crowd server.

Step 4. Hook Up Centralized Authentication

Before modifying the security configuration, you will need to add the Spring configuration file to wire up the Crowd client beans. Add the `applicationContext-CrowdClient.xml` configuration file to the list of `contextConfigLocations` in `WEB-INF/web.xml`:

```
<context-param>
  <param-name>contextConfigLocation</param-name>
  <param-value>
    classpath:/applicationContext-resources.xml
    classpath:/applicationContext-dao.xml
    classpath:/applicationContext-service.xml
    classpath*/applicationContext.xml
    classpath:/applicationContext-CrowdClient.xml
    /WEB-INF/applicationContext*.xml
    /WEB-INF/xfire-servlet.xml
    /WEB-INF/security.xml
  </param-value>
</context-param>
```

AppFuse neatly stores all the Acegi configuration in `myproject/src/main/webapp/WEB-INF/security.xml`. In order to get centralized authentication, we will need to set up Acegi to use the wrapped authenticator class we just created. Edit the Acegi beans in `security.xml`:

1. Add the definition of the CrowdUserDetailsService:

```
<bean id="crowdUserDetailsService" class="com.atlassian.crowd.integration.acegi.user.CrowdUserDetailsServiceImpl">
  <property name="authenticationManager" ref="crowdAuthenticationManager"/>
  <property name="groupMembershipManager" ref="crowdGroupMembershipManager"/>
  <property name="userManager" ref="crowdUserManager"/>
  <property name="authorityPrefix" value="ROLE_"/>
</bean>
```

2. Add the definition of the RemoteCrowdAuthenticationProvider which will delegate Acegi's authentication requests to Crowd:

```
<bean id="crowdAuthenticationProvider" class="com.atlassian.crowd.integration.acegi.
RemoteCrowdAuthenticationProvider">
  <constructor-arg ref="crowdAuthenticationManager"/>
  <constructor-arg ref="httpAuthenticator"/>
  <constructor-arg ref="crowdUserDetailsService"/>
</bean>
```

3. Replace the DaoAuthenticationProvider with our authenticator in the authentication manager:

```
<bean id="authenticationManager" class="org.acegisecurity.providers.ProviderManager">
  <property name="providers">
    <list>
      <ref local="crowdAuthenticationProvider"/>
      <!--ref local="daoAuthenticationProvider"-->
      <ref local="anonymousAuthenticationProvider"/>
      <ref local="rememberMeAuthenticationProvider"/>
    </list>
  </property>
</bean>
```

4. Now do a:

```
mvn jetty:run-war -Dmaven.test.skip
```

5. Head over to <http://localhost:8080/>.

You should now be able to authenticate the users in your Crowd repository that **meet all of the following conditions**:

- They are in a Crowd directory assigned to the AppFuse application in Crowd. See [more information](#).
- They are in Crowd groups named `USER` and `ADMIN`. You will need to [add these groups](#) and assign the user as a [member of the groups](#). These Crowd group names map to the Acegi authorization roles defined in the AppFuse application.
- They are allowed to authenticate with the AppFuse application because EITHER they are in a group allowed to authenticate with Crowd [see more](#) OR their container directory allows all users to authenticate [see more](#).

Congratulations. You have **centralized authentication** 😊

Application-level centralized user management

One quirk you may notice is that you can't view the profile details of users who exist in Crowd, but did not exist in AppFuse prior to the Crowd integration. Although it's possible to authenticate a Crowd user 'dude' and still run AppFuse as 'dude', 'dude' will not be in AppFuse's local database. AppFuse makes use of a database-backed user management system. In order to achieve application-level **centralized user management**, AppFuse will need to delegate its calls to create, retrieve, update and delete users to Crowd using [Crowd's remote API](#). This will prevent data redundancy and eliminate the hassle of data synchronization. This is beyond the scope of this short tutorial.

Step 5. Hook Up Single Sign-On

Enabling single sign-on (SSO) requires a little more tweaking of the `security.xml`:

1. Change the default processing filter to Crowd's SSO filter:

```
<bean id="authenticationProcessingFilter" class="com.atlassian.crowd.integration.acegi.CrowdSSOAuthenticationProcessingFilter">
  <property name="httpAuthenticator" ref="httpAuthenticator"/>
  <property name="authenticationManager" ref="authenticationManager"/>
  <property name="authenticationFailureUrl" value="/login.jsp?error=true"/>
  <property name="defaultTargetUrl" value="/" />
  <property name="filterProcessesUrl" value="/j_security_check"/>
  <property name="rememberMeServices" ref="rememberMeServices"/>
</bean>
```

2. Add the definition of the CrowdLogoutHandler:

```
<bean id="crowdLogoutHandler" class="com.atlassian.crowd.integration.acegi.CrowdLogoutHandler">
  <property name="httpAuthenticator" ref="httpAuthenticator"/>
</bean>
```

3. Update the definition of the LogoutFilter to use the CrowdLogoutHandler. You may need to uncomment the logout filter.

```
<bean id="logoutFilter" class="org.acegisecurity.ui.logout.LogoutFilter">
  <constructor-arg value="/index.jsp"/>
  <constructor-arg>
    <list>
      <ref bean="rememberMeServices"/>
      <ref bean="crowdLogoutHandler"/>
      <bean class="org.acegisecurity.ui.logout.SecurityContextLogoutHandler"/>
    </list>
  </constructor-arg>
  <property name="filterProcessesUrl" value="/logout.jsp"/>
</bean>
```

4. If the logout filter is not defined in the filter invocation list, you will need to add it:

```
<bean id="filterChainProxy" class="org.acegisecurity.util.FilterChainProxy">
  <property name="filterInvocationDefinitionSource">
    <value>
      ...
      /**=httpSessionContextIntegrationFilter,logoutFilter,authenticationProcessingFilter,
securityContextHolderAwareRequestFilter,rememberMeProcessingFilter,anonymousProcessingFilter,
exceptionTranslationFilter,filterInvocationInterceptor
    </value>
    ...
  </property>
</bean>
```

5. Now repeat:

```
mvn jetty:run-war -Dmaven.test.skip=true
```

SSO will only work for users that are able to **authenticate** with both applications and are **authorized** to use both applications. Try out the following:

- Log in to Crowd you should be logged in to AppFuse.
- Log out of AppFuse you should be logged out of Crowd.
- Log in to AppFuse; log out of Crowd; log in to Crowd as another user; refresh AppFuse you should be logged in as the new user.

Congratulations, you have **SSO** 😊

Integrating Crowd with Jive Forums

Jive Forums allows you to specify an implementation that provides authentication and authorization external to the application. This document outlines how to integrate Crowd's authenticator with Jive Forums.

Support for Jive Forums version 5.5.13 only

Crowd provides centralized authentication and single sign-on (SSO) for Jive Forums version **5.5.13** only. Jive have announced that Jive Forums has evolved into a new product, [Jive Social Business Software \(SBS\)](#). We have no plans to update Crowd to support later versions of Jive Forums.


Prerequisites

1. Download and configure Crowd. Refer to the [Crowd installation guide](#) for detailed information on how to do this. We will refer to the Crowd root folder as `CROWD`.
2. Install/configure Jive Forums. Refer to the relevant Jive Forums documentation for information regarding this installation process. The documentation is usually supplied with the software distribution. Do not attempt to use Crowd as the authentication system during the installation process (use the default authentication system for the installation process).

Step 1. Tell Crowd about Jive Forums


1.1 Prepare Crowd's Directory/Users for Jive Forums

The Jive Forums application will need to locate users from a directory configured in Crowd. You will need to set up a directory in Crowd for Jive. For more information on how to do this, see [Adding a Directory](#). We will assume that the directory is called *Jive Forum Directory* for the rest of this document. It is possible to assign more than one directory for an application, but for the purposes of this example, we will use *Jive Forum Directory* to house Jive Forum users.

If you have an existing Jive Forums deployment and would like to import existing users into Crowd, use the Jive Importer tool by navigating **Users > Import Users > JIVE**. Select the *Jive Forum Directory* as the directory into which Jive Forum users will be imported. For details please see [Importing Users from Jive Forums](#).  If you are going to import users into Crowd, you need to do this now before you proceed any further.

1.2 Define the Jive Forums Application in Crowd

Crowd needs to be aware that the Jive Forums application will be making authentication requests to Crowd. We need to add the Jive Forums application to Crowd and map it to the *Jive Forums Directory*:

1. Log in to the [Crowd Administration Console](#) and navigate to **Applications > Add Application**.
2. Complete the '**Add Application**' wizard for the Jive Forums application. See the [instructions](#).  The **Name** and **Password** values you specify in the 'Add Application' wizard must match the **application.name** and **application.password** that you will set in the `JIVEFORUMS/WEB-INF/classes/crowd.properties` file. (See Step 2 below.)

1.3 Specify which Users can Log In to Jive Forums

Once Crowd is aware of the Jive Forums application, Crowd needs to know which users can authenticate (log in) to Jive Forums via Crowd. As part of the 'Add Application' wizard, you will set up your directories and group authorizations for the application. If necessary, you can adjust these settings after completing the wizard. Below are some examples.

You can either configure entire directories to authenticate or allow particular groups. In our example, we can simply allow the entire directory to authenticate:

jiveforums

Details Directories Groups Permissions Remote Addresses Config Test

Map your user directories to the application. When a user accesses the application, Crowd searches the mapped directories to authenticate the user and determine their group/role membership. To access the application, the user must belong to a directory that allows all to authenticate, or to a group that is mapped to the application.

Directory	Allow All to Authenticate	Action
Jive Forums Directory	True <input type="button" value="v"/>	Remove

Alternatively, we can use the **Groups** tab to restrict the application to only authenticate particular groups of users. For details please see [Specifying which Groups can access an Application](#).

1.4 Specify the Address from which Jive Forums can Log In to Crowd

As part of the 'Add Application' wizard, you will set up Jive Forums's IP address. This is the address which Jive Forums will use to authenticate to Crowd. If necessary you can add a hostname, in addition to the IP address, after completing the wizard. See [Specifying an Application's Address or Hostname](#).

Step 2. Tell Jive Forums about Crowd

2.1 Install the Crowd Client Libraries into the Jive Forums WebApp

Jive Forums may be deployed on an application server as a single WAR file or a an exploded WAR folder. For the rest of the installation process, we will assume that Jive Forums has been set up as an exploded war file. If you need Jive Forums to be installed as a single WAR file, simply expand the WAR to a directory, make the changes as described below, and zip up the directory to form the WAR file. We will refer to the root folder of the Jive Forums web-app as `JIVEFORUMS`.

1. Copy the Crowd integration libraries and configuration files (this is described in the [Client Configuration](#) documentation). This is summarized below:

Copy From	Copy To
CROWD/client/crowd-integration-client-X.X.X.jar	JIVEFORUMS/WEB-INF/lib
CROWD/client/lib/log4j-1.2.8.jar	JIVEFORUMS/WEB-INF/lib/
CROWD/client/lib/ehcache-1.2.3.jar	JIVEFORUMS/WEB-INF/lib/
CROWD/client/conf/crowd.properties	JIVEFORUMS/WEB-INF/classes/
CROWD/client/conf/crowd-ehcache.xml	JIVEFORUMS/WEB-INF/classes/

2. Replace the XFire libraries in your Jive Forums installation with the later version shipped with Crowd:
 - Remove all `xfire*.jar` files from your `JIVEFORUMS/WEB-INF/lib` folder.
 - Copy the XFire libraries from Crowd:

Copy From	Copy To
CROWD/client/xfire*.jar	JIVEFORUMS/WEB-INF/lib/

3. Examine the `JIVEFORUMS/WEB-INF/lib` folder and delete any duplicate JARs. Duplicate JARs represent common libraries used by both the Crowd client and Jive Forums.
4. Edit `JIVEFORUMS/WEB-INF/classes/crowd.properties`. Change the following properties:

Key	Value
-----	-------

application.name	jiveforums
application.password	<i>set a password</i>

The **name** and **password** values must match those set when defining the application in Crowd (see Step 1 above).

You can read more about [the crowd.properties file](#).

2.2 Configure Jive Forums to use Crowd's Authenticator

Crowd is now set up to provide authentication services to Jive. Now Jive needs to be set up to use Crowd's authenticator. There are a few ways of doing this. The most user-friendly method is outlined below:

1. In your `jiveHome` directory, edit a file named `jive_startup.xml`. Modify the `<setup>` node to be `false`:

```
<jive>
  <!-- When setup is false, you can access the setup tool. -->
  <setup>false</setup>
  ...
  <!-- Allow SSO login for admins -->
  <admin>
    <tryAlternativeLogin>true</tryAlternativeLogin>
  </admin>
</jive>
```

As the XML comment states, this lets us re-run Jive's setup.

2. Restart Jive Forums so that it picks up the changes.
3. View the Jive Forums site with a web browser - usually under the `/jiveforums` context-root. Jive will run the "Jive Forums Setup".
4. In the '**Install Checklist**' screen, click '**Continue**' to navigate through the setup process.
5. In the '**Datasource Settings**' screen, re-enter your database configuration details and click '**Continue**'.
6. In the '**User System**' screen, select '**Custom**' authentication system and click '**Continue**':

The screenshot shows the 'Jive Forums Setup' interface. At the top, there's a blue header with 'Jive Forums Setup' on the left and 'Jive Forums' on the right. Below the header is a progress bar with five steps: 'Install Checklist' (green), 'Datasource Settings' (green), 'User System' (yellow), 'Email Settings' (red), and 'Admin Account' (red). The main content area is titled 'User, Group and Authentication Systems' and contains the following text: 'Choose a user, group and authentication system below. Most installations should use the default implementation. The other options can be used when you need to integrate Jive Forums with an existing user database or authentication system.' There are three radio button options: 'Default - Use the Jive Forums default user, group and authentication implementations.', 'LDAP - Use LDAP for authentication and storing user data.', and 'Custom - Specify a custom user, group or authentication implementation.' The 'Custom' option is selected. A 'Continue' button is located at the bottom right of the form.

7. You should be at the '**Custom User System**' screen. Enter the following details which specify Crowd as the custom authenticator:

UserManager implementation:

```
com.atlassian.crowd.integration.jive.CrowdUserManager
```

GroupManager implementation:

If you would like Crowd to manage your user groups, add the following group manager:

```
com.atlassian.crowd.integration.jive.CrowdGroupManager
```

i You can safely leave this field empty if you do not want Crowd to manage your groups.

AuthFactory implementation:

```
com.atlassian.crowd.integration.jive.CrowdAuthFactory
```

Click '**Continue**'.

If you have any errors at this stage, it is very likely that there is a classpath issue (eg. the Crowd client libraries aren't being properly loaded by Jive). Please read the documentation regarding [Crowd Client Libraries](#) for help identifying the problem.

8. In the '**Email Settings**' screen, re-enter your email configuration details and click '**Continue**'.
 9. In the '**Admin Account Setup**' screen, *do not enter any details*. Click '**Skip this step**'.

i Warning

The default administrator for Jive Forums is the user `admin`. This user will need to exist in your mapped directory (i.e. the *Jive Forums Directory*) in Crowd. Without this user, you will not be able to access the administration console of Jive Forums.

10. Bounce the server and test that Crowd is authenticating users for Jive. You can do this by creating users (users) via the Crowd Administration Console and verifying that they are able to log in to Jive Forums.

i Jive Forums Documentation

For further information regarding Jive Forums Authentication Integration, check out the Jive Forums Documentation at <http://www.jivesoftware.com/builds/docs/latest/documentation/developer-guide.html#userintegration>

Check out the [Jive SSO](#) page for more details on Jive SSO Integration and corresponding use cases.

RELATED TOPICS

- [Using the Application Browser](#)
- [Adding an Application](#)
 - [Integrating Crowd with Atlassian Bamboo](#)
 - [Integrating Crowd with Atlassian Confluence](#)
 - [Integrating Crowd with Atlassian Confluence 3.4 or earlier](#)
 - [Updating Files in a Confluence Evaluation Distribution](#)
 - [Integrating Crowd with Atlassian CrowdID](#)
 - [Integrating Crowd with Atlassian Crucible](#)
 - [Integrating Crowd with Atlassian FishEye](#)
 - [Configuring FishEye earlier than 4.0 with Crowd](#)
 - [Integrating Crowd with Atlassian Jira](#)
 - [Integrating Crowd with Atlassian Jira 4.2 or earlier](#)
 - [Integrating Crowd with Atlassian Bitbucket Server](#)
 - [Integrating Crowd with Acegi Security](#)
 - [Integrating AppFuse - a Crowd-Acegi Integration Tutorial](#)
 - [Integrating Crowd with Jive Forums](#)
 - [Jive SSO](#)
 - [Integrating Crowd with Spring Security](#)
 - [Integrating AppFuse - a Crowd-Spring Security Integration Tutorial](#)
 - [Integrating Crowd with a Custom Application](#)
 - [Integrating Crowd with Atlassian HipChat](#)
- [Configuring the Google Apps Connector](#)
- [Mapping a Directory to an Application](#)
 - [Specifying the Directory Order for an Application](#)
 - [Specifying an Application's Directory Permissions](#)
 - [Example of Directory Permissions](#)
 - [Viewing Users in Directories Mapped to an Application](#)
 - [Specifying which Groups can access an Application](#)
 - [Syncing users based on their access rights](#)
- [Effective memberships with multiple directories](#)
- [Specifying an Application's Address or Hostname](#)
- [Testing a User's Login to an Application](#)
- [Enforcing Lower-Case Usernames and Groups for an Application](#)
- [Managing an Application's Session](#)
- [Deleting or Deactivating an Application](#)
- [Configuring Caching for an Application](#)
- [Overview of SSO](#)
- [Configuring Options for an Application](#)
- [Enabling OpenID client app](#)
- [Allowing applications to create user tokens](#)
- [Disabling the OpenID client app](#)
- [Configuring how users log in](#)

[Crowd documentation](#)

Jive SSO

This page details the nuts and bolts of Jive SSO. If you are having issues with Jive SSO, this page should be able to give you a better idea of what's going on behind the scenes and help you diagnose any common problems.

For Crowd-Jive integration, the incoming request must:

1. be authenticated with Crowd (have a Crowd SSO token in session or as a cookie)
2. be authenticated with Jive (have a CrowdAuthToken stored in HttpSession for Jive)

To authenticate with Crowd: simply log in to Crowd via any Crowd-SSO enabled application. This includes Jive's login page.

To authenticate with Jive: you need to be authenticated with Crowd as a user "allowed to be authenticated" by Jive. This means, the user must belong to a group or directory which Jive is authorized to authenticate. This user also needs to NOT be on any user/IP ban lists within the Jive application. The Crowd integration will honor the ban list. See note below.

Enumeration of Use Cases

User views Jive Forums and:

1. request is not authenticated with Crowd -> appears as guest user in Jive.
2. request is authenticated with Crowd, but user is not in directory/group allowed to authenticate with Jive -> appears as guest user in Jive.
3. request is authenticated with Crowd, user allowed to authenticate with Jive, user not on any ban list -> appears as logged-in user in Jive.
4. authenticated Jive user clicks logout from Jive -> user is logged out of Jive and Crowd.
5. authenticated Jive user logs out of Crowd using another SSO app -> user eventually times out of Jive.
6. request is authenticated with Crowd, user banned from logging into Crowd -> user appears as guest in Jive.
7. admin authenticated with Crowd and attempts to access Jive admin console -> admin appears logged in to Jive admin console.
8. authenticated Jive admin attempts to log out from Jive's admin console -> **admin is still logged in** (support issue filed with Jive Forums).
9. authenticated Jive admin attempts to log out from Jive Forums -> admin is logged out of Jive and Crowd.
10. request is authenticated with Crowd but user is banned from Jive Forums -> user is still authenticated with Crowd, but not allowed to log in to Jive Forums

Special Cases

- It is known that the "remember me" functionality of Jive will cease to function. This has been intentionally disabled. Jive's "remember me" functionality will need to be replaced by a more general "remember me" from within Crowd. Once this is implemented in Crowd, the Jive integration libraries can utilize Crowd's "remember me", so that "remember me" is centralized.
- It is recommended that admins do not use ban lists. Rather, you should manage access control based on Crowd's groups. So it's best to disable Ban Users from within Ban Settings inside the Jive admin console. There is nothing wrong with using ban lists, as they will be honored by the Crowd-Jive integration libraries. So they will make it hard for a banned user to switch to a non-banned user. The only way a banned Jive user, authenticated with Crowd for Jive, will be able to switch to a different user that Jive will pick up, is when the Jive's Crowd authentication cache clears, so that Jive recognizes a new user is signing in. This is because there is no way to log out a banned user from Jive, as they will always appear to be "guest". So basically, if you have users with multiple identities, if one is banned and attempts to log in, the user will have to wait until the client cache is cleared before he/she can log in with a different identity. Note: it's easy for non-banned users to switch identities as the client authentication cache is cleared when they click "logout" from within Jive.


Related Topics

- [Using the Application Browser](#)
- [Adding an Application](#)
 - [Integrating Crowd with Atlassian Bamboo](#)

- [Integrating Crowd with Atlassian Confluence](#)
 - [Integrating Crowd with Atlassian Confluence 3.4 or earlier](#)
 - [Updating Files in a Confluence Evaluation Distribution](#)
- [Integrating Crowd with Atlassian CrowdID](#)
- [Integrating Crowd with Atlassian Crucible](#)
- [Integrating Crowd with Atlassian FishEye](#)
 - [Configuring FishEye earlier than 4.0 with Crowd](#)
- [Integrating Crowd with Atlassian Jira](#)
 - [Integrating Crowd with Atlassian Jira 4.2 or earlier](#)
- [Integrating Crowd with Atlassian Bitbucket Server](#)
- [Integrating Crowd with Acegi Security](#)
 - [Integrating AppFuse - a Crowd-Acegi Integration Tutorial](#)
- [Integrating Crowd with Jive Forums](#)
 - [Jive SSO](#)
- [Integrating Crowd with Spring Security](#)
 - [Integrating AppFuse - a Crowd-Spring Security Integration Tutorial](#)
- [Integrating Crowd with a Custom Application](#)
- [Integrating Crowd with Atlassian HipChat](#)
- [Configuring the Google Apps Connector](#)
- [Mapping a Directory to an Application](#)
 - [Specifying the Directory Order for an Application](#)
 - [Specifying an Application's Directory Permissions](#)
 - [Example of Directory Permissions](#)
 - [Viewing Users in Directories Mapped to an Application](#)
 - [Specifying which Groups can access an Application](#)
 - [Syncing users based on their access rights](#)
- [Effective memberships with multiple directories](#)
- [Specifying an Application's Address or Hostname](#)
- [Testing a User's Login to an Application](#)
- [Enforcing Lower-Case Usernames and Groups for an Application](#)
- [Managing an Application's Session](#)
- [Deleting or Deactivating an Application](#)
- [Configuring Caching for an Application](#)
- [Overview of SSO](#)
- [Configuring Options for an Application](#)
- [Enabling OpenID client app](#)
- [Allowing applications to create user tokens](#)
- [Disabling the OpenID client app](#)
- [Configuring how users log in](#)

[Crowd documentation](#)

Integrating Crowd with Spring Security

 The content on this page relates to platforms which are not supported for Crowd. Consequently, Atlassian **can not guarantee providing any support for the steps described on this page**. Please be aware that this material is provided for your information only and that you use it at your own risk.

Crowd provides **centralized authentication** and **single sign-on** connectors for the web security framework [Spring Security](#). Spring Security provides a modular and highly configurable approach to authentication and authorization for J2EE applications.

If your web application already makes use of the Spring Security framework for authentication and authorization, you can use the Crowd Spring Security connector to allow your application to easily delegate authentication and authorization requests to Crowd.

Spring, Acegi and Crowd versions

Spring Security was formerly known as Acegi. There is a [separate tutorial for integrating Acegi with Crowd](#). The connector is developed and tested with **Spring Security 3.1** from **Crowd 2.5 and later**. Please use a previous supported release of Crowd if you require compatibility with **Spring Security 2.0.4**.

Please consult the Spring Security [suggested steps](#) or [reference guide](#) for a thorough insight into the Spring Security framework. You might also find useful information in our [Appfuse integration tutorial](#).

This guide assumes developer-level knowledge and a Spring Security-based web application

This guide is for developers rather than administrators. This guide assumes you have **Crowd 2.5** or later installed and that you want to integrate your Spring Security-based web application with Crowd's security server. The documentation below describes how to integrate Crowd with your own application that uses the Spring Security framework. It assumes you already use Spring Security in your application. If you need help integrating the Spring Security framework with your web application, have look at some of the [Spring Security documentation](#).

Prerequisites

1. Download and configure Crowd. Refer to the [Crowd Installation Guide](#) for detailed information on how to do this. We will refer to the Crowd root folder as `CROWD`.
2. Have your Spring Security-based custom application ready for tweaking. We will refer to your custom application as '**SpringSecApp**'.

Step 1. Configuring Crowd to Talk to your Spring Security Application

Crowd needs to be aware that SpringSecApp will be making authentication requests to Crowd. In brief, you will need to do the following:

1. [Add the SpringSecApp application](#) to Crowd.
2. [Add and configure](#) the directories visible to SpringSecApp.
3. [Add and map](#) the groups which are allowed to authenticate with SpringSecApp.

Please see [Adding an Application](#) for a detailed guide.

Step 2. Installing the Crowd Spring Security Connector

2.1 Adding the Crowd Spring Security Connector to your Spring Security Application

You will need to add the Crowd Spring Security connector library and its associated dependencies to your Spring Security application. You can do this manually by copying over the JAR files to your Spring Security application or, if your Spring Security application is a [Maven](#) project, you can add the Crowd Spring Security connector as a project dependency. Both methods are described below.

2.1.1 Manually Adding the Crowd Spring Security Connector Libraries

Follow either 2.1.1 or 2.1.2 (not both).

Copy the Crowd integration libraries and configuration files. This is described in the [Client Configuration](#) documentation. You will need to copy at least the following file to your Spring Security application:

Copy From	Copy To
crowd-integration-springsecurity-X.X.X (this file is not included in the crowd distribution, you need to download it from packages.atlassian.com)	SpringSecApp/WEB-INF/lib
CROWD/client/lib/*.jar	SpringSecApp/WEB-INF/lib

2.1.2 Adding the Crowd Spring Security Connector as a Maven Dependency

Follow either 2.1.1 or 2.1.2 (not both).

Add to your `pom.xml`:

```
<properties>
  <crowd.version>2.5.0</crowd.version>
  <spring.version>3.1.0.RELEASE</spring.version>
</properties>

<dependencies>
  ...

  <dependency>
    <groupId>com.atlassian.crowd</groupId>
    <artifactId>crowd-integration-springsecurity</artifactId>
    <version>${crowd.version}</version>
    <scope>runtime</scope>
  </dependency>

  <!-- Crowd needs at runtime -->
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context</artifactId>
    <version>${spring.version}</version>
    <scope>runtime</scope>
  </dependency>
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-core</artifactId>
    <version>${spring.version}</version>
    <scope>runtime</scope>
  </dependency>
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-beans</artifactId>
    <version>${spring.version}</version>
    <scope>runtime</scope>
  </dependency>
  ...
</dependencies>
```

Ensure you have dependencies on the `spring`-modules to pick up the versions of Spring required by Crowd rather than the possibly lower version specified by Spring Security.

2.2 Adding the Cache Configuration File

Copy the following file into your application's classpath:

Copy From	Copy To
-----------	---------

CROWD/client/conf/crowd-ehcache.xml	SpringSecApp/WEB-INF/classes/crowd-ehcache.xml
-------------------------------------	--

This file can be tweaked to change the cache behavior.

2.3 Configuring the Crowd Spring Security Connector Properties

The Crowd Spring Security connector needs to be configured with the details of the Crowd server.

1. Copy the defaultcrowd.propertiesfile to the classpath of your Spring Security application:

Copy From	Copy To
CROWD/client/conf/crowd.properties	SpringSecApp/WEB-INF/classes

2. Editcrowd.propertiesand populate the following fields appropriately:

Key	Value
application.name	Same as application name defined when adding the application to Crowd in Step 1.
application.password	Same as application password defined when adding the application to Crowd in Step 1.
crowd.server.url	http://localhost:8095/crowd/services/
session.validation.interval	This is the time interval between requests which validate whether the user is logged in or out of the Crowd SSO server. Set to 0, if you want authentication checks to occur on each request. Otherwise set to the number of minutes you wish to wait between requests. Setting this value to 1 or higher will increase the performance of Crowd's integration.

You can read more about[the crowd.properties file](#).

Step 3. Configuring your Spring Security Application to Use the Crowd Spring Security Connector

There are two ways you can integrate your application with Crowd:

- **Centralized user management:** The user repository available to your application will be the user repository allocated to your application via Crowd. This means that your application will use the centralized user repository for retrieving user details as well as performing authentication.
- **Single sign-on:** In addition to centralized authentication, SSO will be available to your application. If any other SSO-enabled applications (such as [Jira](#), [Confluence](#), or your own custom applications) are integrated with Crowd, then SSO behavior will be established across these applications. If you sign in to one application, you are signed in to all applications. If you sign out of one application, you are signed out of all applications.

First, you will need to add the Crowd client application context to wire up the Crowd beans that manage the communication to Crowd. You can do this by including theapplicationContext-CrowdRestClient.xmlSpring configuration file, found incrowd-integration-client-rest.jar. For example, if you are configuring Spring using a context listener, you can add the following parameter in your your Spring Security application'sWEB-INF/web.xml:

```
<context-param>
  <param-name>contextConfigLocation</param-name>
  <param-value>
    ...
    classpath:/applicationContext-CrowdRestClient.xml
    ...
  </param-value>
</context-param>
```

3.1 Configuring Centralized User Management

The following sections assume that you have the Spring Security schema mapped to the `security` namespace. Perform the following updates to your Spring Security configuration:

1. Add the definition of the `CrowdUserDetailsService`:

```
<bean id="crowdUserDetailsService" class="com.atlassian.crowd.integration.springsecurity.user.CrowdUserDetailsServiceImpl">
  <property name="crowdClient" ref="crowdClient" />
  <property name="authorityPrefix" value="ROLE_" />
</bean>
```

2. Add the definition of the `RemoteCrowdAuthenticationProvider`:

```
<bean id="crowdAuthenticationProvider" class="com.atlassian.crowd.integration.springsecurity.RemoteCrowdAuthenticationProvider">
  <constructor-arg ref="crowdClient" />
  <constructor-arg ref="crowdHttpAuthenticator" />
  <constructor-arg ref="crowdUserDetailsService" />
</bean>
```

✔ Controlling granted authority names

Rather than taking the group name and setting a prefix, you can define a mapping to grant specific authorities when a user belongs to Crowd groups:

```
<util:map id="groupToAuthorityMappings">
  <beans:entry key="crowd-administrators" value="ROLE_crowd-administrators" />
  <beans:entry key="some-other-group" value="specific-authority-for-other-group" />
</util:map>
```

and then set it on the `crowdUserDetailsService`:

```
<beans:bean id="crowdUserDetailsService" class="com.atlassian.crowd.integration.springsecurity.user.CrowdUserDetailsServiceImpl">
  ...
  <beans:property name="groupToAuthorityMappings">
    <beans:bean factory-bean="groupToAuthorityMappings" factory-method="entrySet" />
  </beans:property>
```

✔ Further extensions

- If you have an existing user data model, then you can extend or wrap the `CrowdDetailsService` to cater for user objects within your application domain.
- If you require users within Crowd to be created in your application's persistence model so that you can store application-specific user data, you can extend the `CrowdAuthenticationProvider` to create records for successfully authenticated Crowd users.

Crowd's remote API

We recommend that applications do not store the Crowd users locally. Rather, applications should query users via Crowd's [remote API](#).

3.2 Configuring Single Sign-On (SSO)

SSO is optional and requires centralized user management

Single sign-on is optional. If you wish to configure SSO you must first configure centralized user management as described in step 3.1 above.

Perform the following additional updates to your Spring Security configuration:

1. Remove defaults from the `<http/>` element:
 - a. Remove the `auto-config` attribute and add an `entry-point-ref="crowdAuthenticationProcessingFilterEntryPoint"` attribute to the `http` element.
 - b. Remove the `<form-login>` element.
 - c. Include `custom-filters` for login and logout.
You should end up with `http` elements similar to this:

```
<http pattern='/styles/*' security='none' />
<http pattern='/scripts/*' security='none' />

<http auto-config="false"
      entry-point-ref="crowdAuthenticationProcessingFilterEntryPoint">

  <custom-filter position="FORM_LOGIN_FILTER" ref='authenticationProcessingFilter' />
  <custom-filter position="LOGOUT_FILTER" ref='logoutFilter' />

  <intercept-url pattern="/admin/*" access="ROLE_application-administrators" />
  <intercept-url pattern="/passwordHint.html" access="ROLE_ANONYMOUS,ROLE_ADMIN,ROLE_USER" />
  <intercept-url pattern="/**/*.html*" access="IS_AUTHENTICATED_FULLY" />
</http>
```

2. Change the default processing filter to Crowd's SSO filter by adding the following bean definitions:

```

<authentication-manager alias="authenticationManager">
  <authentication-provider ref='crowdAuthenticationProvider' />
</authentication-manager>

<beans:bean id="crowdAuthenticationProcessingFilterEntryPoint" class="org.springframework.security.
web.authentication.LoginUrlAuthenticationEntryPoint">
  <beans:constructor-arg value="/login.jsp"/>
</beans:bean>

<beans:bean id="authenticationProcessingFilter" class="com.atlassian.crowd.integration.
springsecurity.CrowdSSOAuthenticationProcessingFilter">
  <beans:constructor-arg ref="tokenHelper"/>
  <beans:constructor-arg ref="crowdClient"/>
  <beans:constructor-arg ref="clientProperties"/>
  <beans:property name="httpAuthenticator" ref="crowdHttpAuthenticator"/>
  <beans:property name="authenticationManager" ref="authenticationManager"/>
  <beans:property name="filterProcessesUrl" value="/j_security_check"/>
  <beans:property name="authenticationFailureHandler">
    <beans:bean class="com.atlassian.crowd.integration.springsecurity.
UsernameStoringAuthenticationFailureHandler">
      <beans:property name="defaultFailureUrl" value="/login.jsp?error=true"/>
    </beans:bean>
  </beans:property>

  <beans:property name="authenticationSuccessHandler">
    <beans:bean class="org.springframework.security.web.authentication.
SavedRequestAwareAuthenticationSuccessHandler">
      <beans:property name="defaultTargetUrl" value="/"/>
    </beans:bean>
  </beans:property>
</beans:bean>

```

3. Add the definition of the CrowdLogoutHandler and add in a LogoutFilter that references it:

```

<beans:bean id="crowdLogoutHandler" class="com.atlassian.crowd.integration.springsecurity.
CrowdLogoutHandler">
  <beans:property name="httpAuthenticator" ref="crowdHttpAuthenticator"/>
</beans:bean>

<beans:bean id="logoutFilter" class="org.springframework.security.web.authentication.logout.
LogoutFilter">
  <beans:constructor-arg value="/index.jsp"/>
  <beans:constructor-arg>
    <beans:list>
      <beans:ref bean="crowdLogoutHandler"/>
      <beans:bean class="org.springframework.security.web.authentication.logout.
SecurityContextLogoutHandler"/>
    </beans:list>
  </beans:constructor-arg>
  <beans:property name="filterProcessesUrl" value="/logout.jsp"/>
</beans:bean>

```

Step 4. Restarting your Spring Security Application

Bounce your application. You should now have centralized authentication and single sign-on with Crowd.

Authorization

For the purposes of Crowd integration with Spring Security, you should map Spring Security's roles to Crowd's groups. To put it another way: in order to use Spring Security's authorization features, users in Crowd will have their Spring Security roles specified by their group names.

The authorities granted will use the `authorityPrefix` specified on `crowdUserDetailsService`. If no suffix is specified, the authorities will append the Crowd group name.

For example if user 'admin' is in the 'crowd-admin' group, then the user 'admin' will be authorized to view pages restricted to the 'ROLE_crowd-admin' role in Spring Security.

```

<http>
  ...
  <intercept-url pattern="/console/secure/**" access="ROLE_crowd-administrators"/>
  <intercept-url pattern="/console/info/**" access="ROLE_crowd-users"/>
  <intercept-url pattern="/console/user/**" access="IS_AUTHENTICATED_FULLY"/>
  ...
</http>

```

If `authoritySuffix` is also specified, any user in the mapped groups configured in crowd will be granted "`authorityPrefix+authoritySuffix`" (for example, `ROLE_ADMIN`).

```

<beans:bean id="crowdUserDetailsService" ...>
  ...
  <beans:property name="authorityPrefix" value="ROLE_" />
  <beans:property name="authorityPrefix" value="ADMIN" />
</beans:bean>

<http>
  ...
  <intercept-url pattern="/console/secure/**" access="ROLE_ADMIN"/>
  <intercept-url pattern="/console/user/**" access="IS_AUTHENTICATED_FULLY"/>
  ...
</http>

```

RELATED TOPICS

- [Integrating AppFuse - a Crowd-Spring Security Integration Tutorial](#)
- [Integrating Crowd with Acegi Security](#)
- [Using the Application Browser](#)
- [Adding an Application](#)
- [Configuring the Google Apps Connector](#)
- [Mapping a Directory to an Application](#)
- [Effective memberships with multiple directories](#)
- [Specifying an Application's Address or Hostname](#)
- [Testing a User's Login to an Application](#)
- [Enforcing Lower-Case Usernames and Groups for an Application](#)
- [Managing an Application's Session](#)
- [Deleting or Deactivating an Application](#)
- [Configuring Caching for an Application](#)
- [Overview of SSO](#)
- [Configuring Options for an Application](#)
- [Enabling OpenID client app](#)
- [Allowing applications to create user tokens](#)
- [Disabling the OpenID client app](#)
- [Configuring how users log in](#)

[Crowd documentation](#)

Integrating AppFuse - a Crowd-Spring Security Integration Tutorial

[AppFuse](#) provides a sweet starting point for developing web applications. You choose the frameworks, AppFuse generates the skeleton application.

At its core, the web security of AppFuse 2.0.2+ applications relies on the modular and extensible [Spring Security](#) authentication framework. In this tutorial, we look at a basic integration of Crowd with Spring Security, using an application generated by AppFuse.

Spring Security was formerly known as Acegi

- The Acegi security framework changed its name to Spring Security with its 2.0 release.
- Appfuse 2.0.2 changed from Acegi to Spring Security for authentication. Earlier versions of Appfuse use Acegi.
- If you are working with Acegi in an earlier version of Appfuse, we have a [separate tutorial](#).
- Crowd 1.6 and above provide support for both Spring Security and Acegi. Earlier versions of Crowd only supported Acegi.
- We recommend all new projects use Spring Security as it is being actively maintained.

Prerequisites

This tutorial assumes you have installed Crowd 1.6 or later and are using Appfuse 2.0.2 or later.

Step 1. Get AppFuse

In this tutorial, we will be using the Struts2-basic archetype to create the project, but the other types should be similar. For more information, consult the AppFuse [quickstart guide](#). In particular, it outlines the database requirements for AppFuse.

1. Create the project.

```
mvn archetype:create -DarchetypeGroupId=org.appfuse.archetypes \
-DarchetypeArtifactId=appfuse-basic-struts \
-DremoteRepositories=http://static.appfuse.org/releases -DarchetypeVersion=2.0.2 \
-DgroupId=com.mycompany.app -DartifactId=myproject
```

2. Since we will be editing the core Spring Security configuration, we will need the full source code of the application.

```
cd myproject
mvn appfuse:full-source
```

3. Build it.

```
mvn clean install
```

4. Run it.

```
mvn jetty:run-war -Dmaven.test.skip
```

5. Play with it.

```
http://localhost:8080/
```

6. Shut it down.

```
ctrl+c
```

Step 2. Let Crowd Know about AppFuse

Add `appfuse` as an application via the Crowd Console. See [Adding an Application](#) for more information.

Step 3. Add the Crowd Spring Security Connector to AppFuse

Open up the `pom.xml` and add the Crowd client libraries as a project dependency:

```
<dependencies>
  <dependency>
    <groupId>com.atlassian.crowd</groupId>
    <artifactId>crowd-integration-client</artifactId>
    <version>1.6</version>
  </dependency>
  ...
</dependencies>
```

You will also need to create the file `myproject/src/main/resources/crowd.properties`:

```
application.name           appfuse
application.password       password
application.login.url      http://localhost:8095/crowd/
crowd.server.url          http://localhost:8095/crowd/services/
session.isauthenticated    session.isauthenticated
session.tokenkey          session.tokenkey
session.validationinterval 0
session.lastvalidation    session.lastvalidation
```

In particular, the application name and password must match the values defined for the application added in Step 2.

Finally, copy the `STANDALONE/client/conf/crowd-ehcache.xml` to `myproject/src/main/resources/crowd-ehcache.xml`. This file defines the cache properties, such as cache timeouts, used when accessing data from the Crowd server.

Step 4. Hook Up Centralized Authentication

Before modifying the security configuration, you will need to add the Spring configuration file to wire up the Crowd client beans. Add the `applicationContext-CrowdClient.xml` configuration file to the list of `contextConfigLocations` in `myproject/src/main/webapp/WEB-INF/web.xml`:

```
<context-param>
  <param-name>contextConfigLocation</param-name>
  <param-value>
    classpath:/applicationContext-resources.xml
    classpath:/applicationContext-dao.xml
    classpath:/applicationContext-service.xml
    classpath*/applicationContext.xml
    classpath:/applicationContext-CrowdClient.xml
    /WEB-INF/applicationContext*.xml
    /WEB-INF/xfire-servlet.xml
    /WEB-INF/security.xml
  </param-value>
</context-param>
```

AppFuse neatly stores all the Spring Security configuration in `myproject/src/main/webapp/WEB-INF/security.xml`. In order to get centralized authentication, we will need to set up Spring Security to use Crowd components for user information. Edit the beans in `security.xml`:

1. Add the definition of the `CrowdUserDetailsService`:


```
<beans:bean id="crowdUserDetailsService" class="com.atlassian.crowd.integration.springsecurity.user.CrowdUserDetailsServiceImpl">
  <beans:property name="authenticationManager" ref="crowdAuthenticationManager"/>
  <beans:property name="groupMembershipManager" ref="crowdGroupMembershipManager"/>
  <beans:property name="userManager" ref="crowdUserManager"/>
  <beans:property name="authorityPrefix" value="ROLE_" />
</beans:bean>
```

2. Add the definition of the RemoteCrowdAuthenticationProvider that delegates Spring Security authentication requests to Crowd:

```
<beans:bean id="crowdAuthenticationProvider" class="com.atlassian.crowd.integration.springsecurity.RemoteCrowdAuthenticationProvider">
  <custom-authentication-provider />
  <beans:constructor-arg ref="crowdAuthenticationManager"/>
  <beans:constructor-arg ref="httpAuthenticator"/>
  <beans:constructor-arg ref="crowdUserDetailsService"/>
</beans:bean>
```

3. Comment out the default authentication provider, as we've replaced it with Crowd:

```
<!--
  <authentication-provider user-service-ref="userDao">
    <password-encoder ref="passwordEncoder"/>
  </authentication-provider>
-->
```

4. Now do a:

```
mvn clean install
```

This will pick up the configuration changes and add the Crowd client library into your app. Then run:

```
mvn jetty:run-war -Dmaven.test.skip
```

5. Head over to `http://localhost:8080/`. You should now be able to authenticate the users in your Crowd repository that **meet all of the following conditions**:

- They are in a Crowd directory assigned to the AppFuse application in Crowd. See [more information](#).
- They are in Crowd groups named `USER` and `ADMIN`. You will need to [add these groups](#) and assign the user as a [member of the groups](#). These Crowd group names map to the Spring Security authorization roles defined in the AppFuse application.
- They are allowed to authenticate with the AppFuse application because EITHER they are in a group allowed to authenticate with Crowd ([click for details](#)) OR their container directory allows all users to authenticate ([click for details](#)).

Congratulations. You have **centralized authentication** 😊

Application-level centralized user management

One quirk you may notice is that you can't view the profile details of users who exist in Crowd, but did not exist in AppFuse prior to the Crowd integration. Although it's possible to authenticate a Crowd user 'dude' and still run AppFuse as 'dude', 'dude' will not be in AppFuse's local database. AppFuse makes use of a database-backed user management system. In order to achieve application-level **centralized user management**, AppFuse will need to delegate its calls to create, retrieve, update and delete users to Crowd using [Crowd's remote API](#). This will prevent data redundancy and eliminate the hassle of data synchronization. This is beyond the scope of this short tutorial.

Step 5. Hook Up Single Sign-On

Enabling single sign-on (SSO) requires quite a bit more tweaking of the `security.xml`:

1. Remove defaults from the `<http/>` element:

- a. Remove the `auto-config` attribute and add an `entry-point-ref="crowdAuthenticationProcessingFilterEntryPoint"` attribute to the `http` element.
- b. Remove the `<form-login>` element.

You should end up with an `http` element similar to this:

```
<http lowercase-comparisons="false" entry-point-ref="
crowdAuthenticationProcessingFilterEntryPoint"> <!-- note: no auto-config attribute! -->
<!--intercept-url pattern="/images/*" filters="none"/>
<intercept-url pattern="/styles/*" filters="none"/>
<intercept-url pattern="/scripts/*" filters="none"/-->
<intercept-url pattern="/admin/*" access="ROLE_ADMIN"/>
<intercept-url pattern="/passwordHint.html*" access="ROLE_ANONYMOUS,ROLE_ADMIN,ROLE_USER"/>
<intercept-url pattern="/signup.html*" access="ROLE_ANONYMOUS,ROLE_ADMIN,ROLE_USER"/>
<intercept-url pattern="/a4j.res/*.html*" access="ROLE_ANONYMOUS,ROLE_ADMIN,ROLE_USER"/>
<!-- APF-737, OK to remove line below if you're not using JSF -->
<intercept-url pattern="/**/*.html*" access="ROLE_ADMIN,ROLE_USER"/>
<!-- <form-login login-page="/login.jsp" authentication-failure-url="/login.jsp?error=true"
login-processing-url="/j_security_check"/> -->
<remember-me user-service-ref="userDao" key="e37f4b31-0c45-11dd-bd0b-0800200c9a66"/>
</http>
```

2. Change the default processing filter to Crowd's SSO filter by adding the following bean definitions:

```
<authentication-manager alias="authenticationManager"/>

<beans:bean id="crowdAuthenticationProcessingFilterEntryPoint" class="org.springframework.security.
ui.webapp.AuthenticationProcessingFilterEntryPoint">
  <beans:property name="loginFormUrl" value="/login.jsp"/>
</beans:bean>

<beans:bean id="crowdAuthenticationProcessingFilter" class="com.atlassian.crowd.integration.
springsecurity.CrowdSSOAuthenticationProcessingFilter">
  <custom-filter position="AUTHENTICATION_PROCESSING_FILTER"/>
  <beans:property name="httpAuthenticator" ref="httpAuthenticator"/>
  <beans:property name="authenticationManager" ref="authenticationManager"/>
  <beans:property name="authenticationFailureUrl" value="/login.jsp?error=true"/>
  <beans:property name="defaultTargetUrl" value="/"/>
  <beans:property name="filterProcessesUrl" value="/j_security_check"/>
</beans:bean>
```

3. Add the definition of the `CrowdLogoutHandler` and add in a `LogoutFilter` that references it:

```
<beans:bean id="crowdLogoutHandler" class="com.atlassian.crowd.integration.springsecurity.
CrowdLogoutHandler">
  <beans:property name="httpAuthenticator" ref="httpAuthenticator"/>
</beans:bean>

<beans:bean id="logoutFilter" class="org.springframework.security.ui.logout.LogoutFilter">
  <custom-filter position="LOGOUT_FILTER"/>
  <beans:constructor-arg value="/index.jsp"/>
  <beans:constructor-arg>
    <beans:list>
      <beans:ref bean="crowdLogoutHandler"/>
      <beans:bean class="org.springframework.security.ui.logout.SecurityContextLogoutHandler"/>
    </beans:list>
  </beans:constructor-arg>
  <beans:property name="filterProcessesUrl" value="/logout.jsp"/>
</beans:bean>
```

4. Now repeat:

```
mvn jetty:run-war -Dmaven.test.skip=true
```

SSO will only work for users that are able to **authenticate** with both applications and are **authorized** to use both applications. Try out the following:

- Log in to Crowd you should be logged in to AppFuse.
- Log out of AppFuse you should be logged out of Crowd.
- Log in to AppFuse; log out of Crowd; log in to Crowd as another user; refresh AppFuse you should be logged in as the new user.

Congratulations, you have **SSO** 😊

Integrating Crowd with a Custom Application

Crowd ships with out-of-the-box support for a number of [applications](#). You can also integrate Crowd with other applications as follows:

Step 1. Configuring Crowd to talk to your Application

Please see [Adding an Application](#).

Step 2. Configuring your Application to talk to Crowd

2.1 Developing a Crowd Client

If your application is not listed in [Supported Applications and Directories](#) then you will need to create your own Crowd client for your application, using the Crowd REST APIs.

For assistance, please see the developer's guide to [creating a Crowd client for your custom application](#).

2.2 Configuring your Application

The integration libraries and configuration files are included in the Crowd download, in the `client` folder. You will find the Crowd integration library, and the client libraries on which the framework depends, in the `lib` folder. An example client properties file `crowd.properties` is located in the `conf` folder.

To configure your application, perform the following:

1. Copy the Crowd client and supporting libraries to your application's classpath, typically `WEB-INF/lib`. These files will be in Crowd's `client` folder, with a name similar to `crowd-integration-client-X.X.X.jar` and all supporting JARs in the `client/lib` folder.
2. Copy the client properties file `crowd.properties` to your application's deployment directory, typically `WEB-INF/classes`.
3. Edit the `crowd.properties` file to reflect the values of your deployment parameters. Refer to the description of the [attributes in the crowd.properties file](#).

Passing crowd.properties as an environment variable

You can pass the location of a client application's `crowd.properties` file to the client application as an environment variable when starting the client application. This means that you can choose a suitable location for the `crowd.properties` file, instead of putting it in the client application's `WEB-INF/classes` directory.

Example:

```
-Dcrowd.properties={FILE-PATH}/crowd.properties
```

RELATED TOPICS

- [Using the Application Browser](#)
- [Adding an Application](#)
 - [Integrating Crowd with Atlassian Bamboo](#)
 - [Integrating Crowd with Atlassian Confluence](#)
 - [Integrating Crowd with Atlassian Confluence 3.4 or earlier](#)
 - [Updating Files in a Confluence Evaluation Distribution](#)
 - [Integrating Crowd with Atlassian CrowdID](#)
 - [Integrating Crowd with Atlassian Crucible](#)
 - [Integrating Crowd with Atlassian FishEye](#)
 - [Configuring FishEye earlier than 4.0 with Crowd](#)
 - [Integrating Crowd with Atlassian Jira](#)
 - [Integrating Crowd with Atlassian Jira 4.2 or earlier](#)
 - [Integrating Crowd with Atlassian Bitbucket Server](#)
 - [Integrating Crowd with Acegi Security](#)

- [Integrating AppFuse - a Crowd-Acegi Integration Tutorial](#)
- [Integrating Crowd with Jive Forums](#)
 - [Jive SSO](#)
- [Integrating Crowd with Spring Security](#)
 - [Integrating AppFuse - a Crowd-Spring Security Integration Tutorial](#)
- [Integrating Crowd with a Custom Application](#)
- [Integrating Crowd with Atlassian HipChat](#)
- [Configuring the Google Apps Connector](#)
- [Mapping a Directory to an Application](#)
 - [Specifying the Directory Order for an Application](#)
 - [Specifying an Application's Directory Permissions](#)
 - [Example of Directory Permissions](#)
 - [Viewing Users in Directories Mapped to an Application](#)
 - [Specifying which Groups can access an Application](#)
 - [Syncing users based on their access rights](#)
- [Effective memberships with multiple directories](#)
- [Specifying an Application's Address or Hostname](#)
- [Testing a User's Login to an Application](#)
- [Enforcing Lower-Case Usernames and Groups for an Application](#)
- [Managing an Application's Session](#)
- [Deleting or Deactivating an Application](#)
- [Configuring Caching for an Application](#)
- [Overview of SSO](#)
- [Configuring Options for an Application](#)
- [Enabling OpenID client app](#)
- [Allowing applications to create user tokens](#)
- [Disabling the OpenID client app](#)
- [Configuring how users log in](#)

[Crowd documentation](#)

Integrating Crowd with Atlassian HipChat

You can connect Hipchat Server to Atlassian Crowd or to a Jira application (version 4.3 or later) for management of users and for authentication (verification of a user's login).


Connecting Hipchat Server to Crowd


You can connect Hipchat Server to Atlassian Crowd for user management and authentication. For more information on Crowd, see the [Crowd Administration Guide](#).


When to use this option: Connect to Crowd if you want to import and synchronize users from and authenticate those users against Crowd.

On this page:

- [Connecting Hipchat Server to Crowd](#)
- [Connecting Hipchat Server to Jira applications](#)
- [Diagrams of some possible configurations](#)

 You can't filter users. When you sync users, you'll sync all users in Atlassian Crowd. ([Learn the workaround.](#)) Hipchat Server marks inactive users from Crowd as deactivated users and doesn't count them towards your license.

 Crowd does not provide SSO functionality for Hipchat. A feature request to support SSO is available to be watched and up-voted:

 [HCPUB-304](#) - Jira issue doesn't exist or you don't have permission to view it.

To connect Hipchat Server to Crowd:

1. Go to your **Crowd Administration Console** and define the Hipchat Server to Crowd. See the Crowd documentation: [Adding an Application](#).
2. Browse to your server's fully qualified domain name, for example <https://hipchat.yourcompany.com/>.
3. Log into the Hipchat Server web user interface (UI) using your administrator email and password.
4. Click **Group admin > Authentication > External directory**.
5. Choose **Add Directory** then choose **Atlassian Crowd**. Enter the settings as described below.
6. Save the directory settings.
7. Define the **directory order** by clicking the blue up and down arrows next to each directory in the **User Directories** list. Here is a summary of how the directory order affects the processing:
 - The order of the directories is the order in which they will be searched for users and groups.
 - Changes to users and groups will be made only in the first directory where the application has permission to make changes.

For details see [Managing Multiple Directories](#).

Settings in Hipchat Server for the Crowd directory type

Setting	Description
Name	A meaningful name that will help you to identify this Crowd server amongst your list of directory servers. Examples: <ul style="list-style-type: none">• Crowd Server• Example Company Crowd

Server URL	The web address of your Crowd console server. Examples: <ul style="list-style-type: none"> • http://www.example.com:8095/crowd/ • http://crowd.example.com
Application Name	The name of your application, as recognized by your Crowd server. Note that you will need to define the application in Crowd too, using the Crowd administration Console. See the Crowd documentation on adding an application .
Application Password	The password which the application will use when it authenticates against the Crowd framework as a client. This must be the same as the password you have registered in Crowd for this application. See the Crowd documentation on adding an application .

Crowd permissions

Setting	Description
Read Only	The users, groups and memberships in this directory are retrieved from Crowd and can only be modified via Crowd. You cannot modify Crowd users, groups or memberships via the application administration screens. Note: There are no user group structures within Hipchat Server. Group data is synchronized, but not used.
Read /Write	Not applicable to Hipchat Server.

Advanced Crowd settings

Setting	Description
Enable Nested Groups	Not applicable to Hipchat Server. When you sync your users, you'll sync every user that exists in the directory.
Synchronization Interval (minutes)	Synchronization is the process by which the application updates its internal store of user data to agree with the data on the directory server. The application will send a request to your directory server every x minutes, where 'x' is the number specified here. The default value is 60 minutes.

Connecting Hipchat Server to Jira applications


Subject to certain limitations, you can connect a number of Atlassian applications to a single JIRA application for centralized user management.

When to use this option: You can connect to a server running **Jira 4.3** or later, **Jira Software 7.0** or later, **Jira Core 7.0** or later, or **Jira Service Management (formerly Jira Service Desk) 3.0** or later. Choose this option as an alternative to Atlassian Crowd, for simple configurations with a limited number of users.



You can't filter users. When you sync users, you'll sync all users in the Jira application. Hipchat Server marks inactive users from the Jira application as deactivated users and doesn't count them towards your license.

To connect Hipchat Server to a Jira application:

1. Configure the Jira application to recognize Hipchat Server:
 - a. Log in to the Jira application as a user with the **Jira Administrators** global permission.
 - b. Choose  > **User Management** > **User Server**.
 ✓ **Keyboard shortcut: 'g' + 'g' + start typing 'jira user'**.
 - c. **Add** an application.
 - d. Enter the **application name** and **password** that Hipchat Server will use when accessing the Jira application.
 - e. Enter the **IP address** or addresses of Hipchat Server.
 - f. **Save** the new application.
2. Configure Hipchat Server to connect to a Jira application:
 - a. Browse to your server's fully qualified domain name, for example `https://hipchat.yourcompany.com/`.
 - b. Log into the Hipchat Server web user interface (UI) using your administrator email and password.
 - c. Click **Group admin** > **Authentication** > **External directory**.
 - d. Choose **Add Directory** then choose **Atlassian Jira**. Enter the settings as described below.
 - e. Enter the settings as described below. When asked for the **application name** and **password**, enter the values that you defined in the settings on the Jira application.
 - f. Save the directory settings.
 - g. Define the **directory order** by clicking the blue up- and down-arrows next to each directory in the **User Directories** list.

Here is a summary of how the directory order affects the processing:

- The order of the directories is the order in which they will be searched for users.
- Changes to users will be made only in the first directory where the application has permission to make changes.

For details see [Managing Multiple Directories](#).

Settings for the Jira directory type

Setting	Description
Name	A meaningful name that will help you to identify this Jira server in the list of directory servers. Examples: <ul style="list-style-type: none"> • JIRA Service Desk Server • My Company JIRA
Server URL	The web address of your Jira server. Examples: <ul style="list-style-type: none"> • <code>http://www.example.com:8080</code> • <code>http://jira.example.com</code>
Application Name	The application name that Hipchat Server will use when accessing the Jira application.
Application Password	The password that Hipchat Server will use when accessing the Jira server that acts as user manager.

Permissions for the Jira directory type

Setting	Description
---------	-------------

Read Only	<p>The users, groups and memberships in this directory are retrieved from the Jira server that is acting as user manager. They can only be modified via that Jira server.</p> <p>Note: There are no user group structures within Hipchat Server. Group data is synchronized, but not used.</p>
-----------	---

Advanced Settings for the Jira directory type

Setting	Description
Enable Nested Groups	Not applicable to Hipchat Server.
Synchronization Interval (minutes)	Synchronization is the process by which the application updates its internal store of user data to agree with the data on the directory server. The application will send a request to your directory server every x minutes, where 'x' is the number specified here. The default value is 60 minutes.

Diagrams of some possible configurations

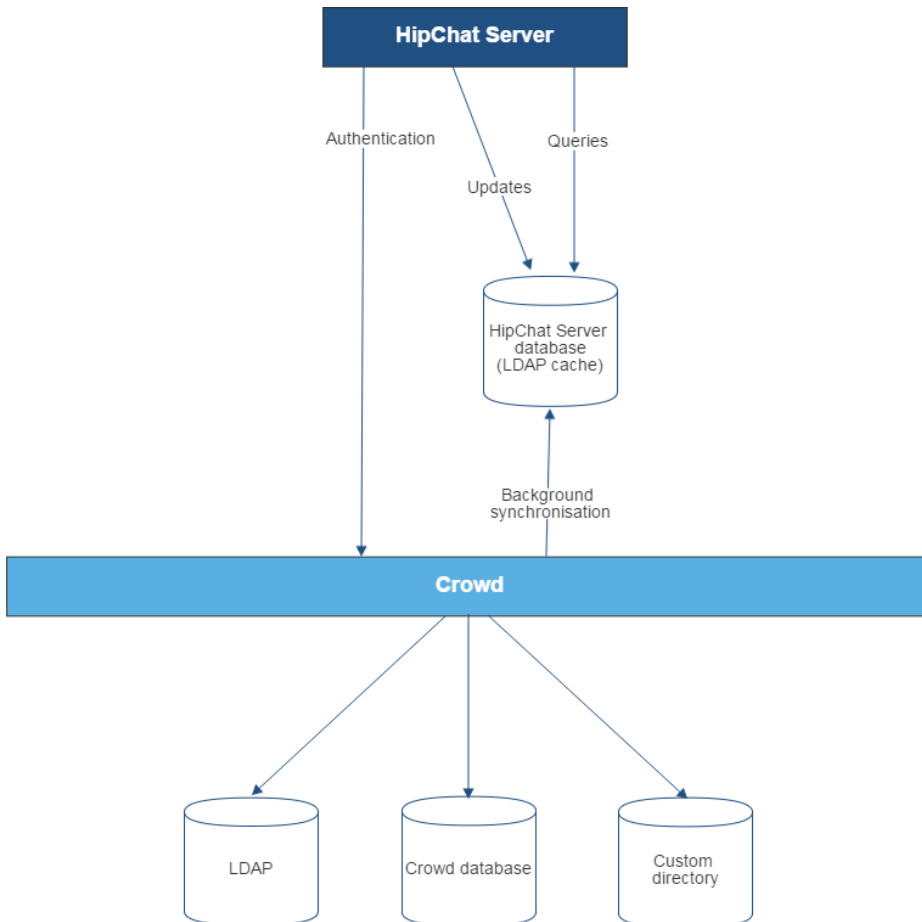


Diagram above: Hipchat Server connecting to Crowd for user management.

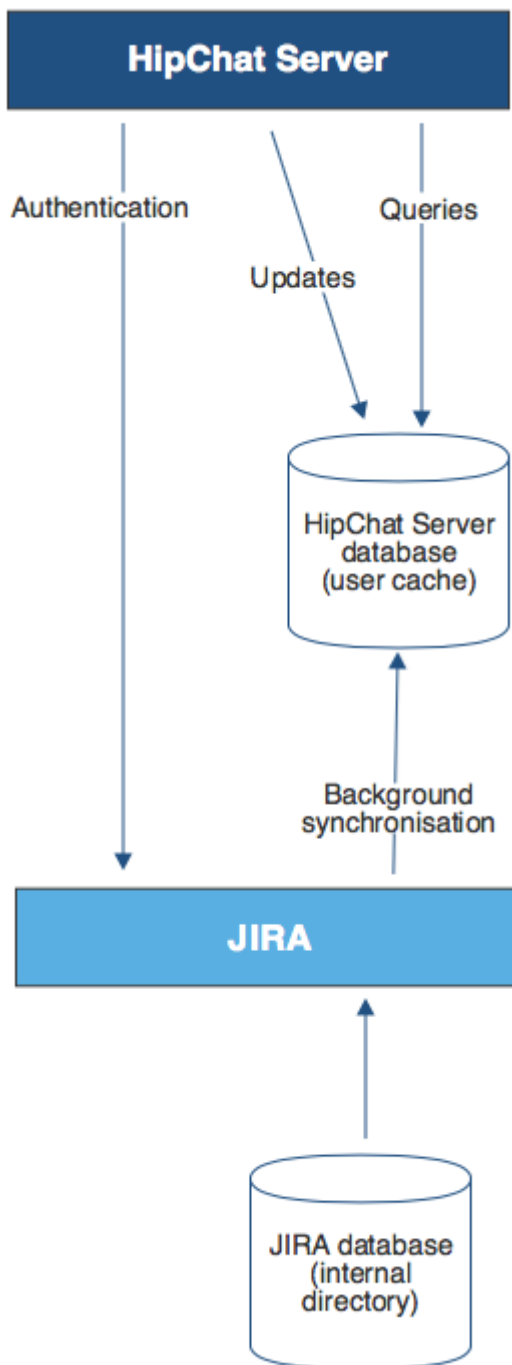


Diagram above: Hipchat Server connecting to a Jira application for user management. The Jira application does the user management, storing the user data in its internal directory.

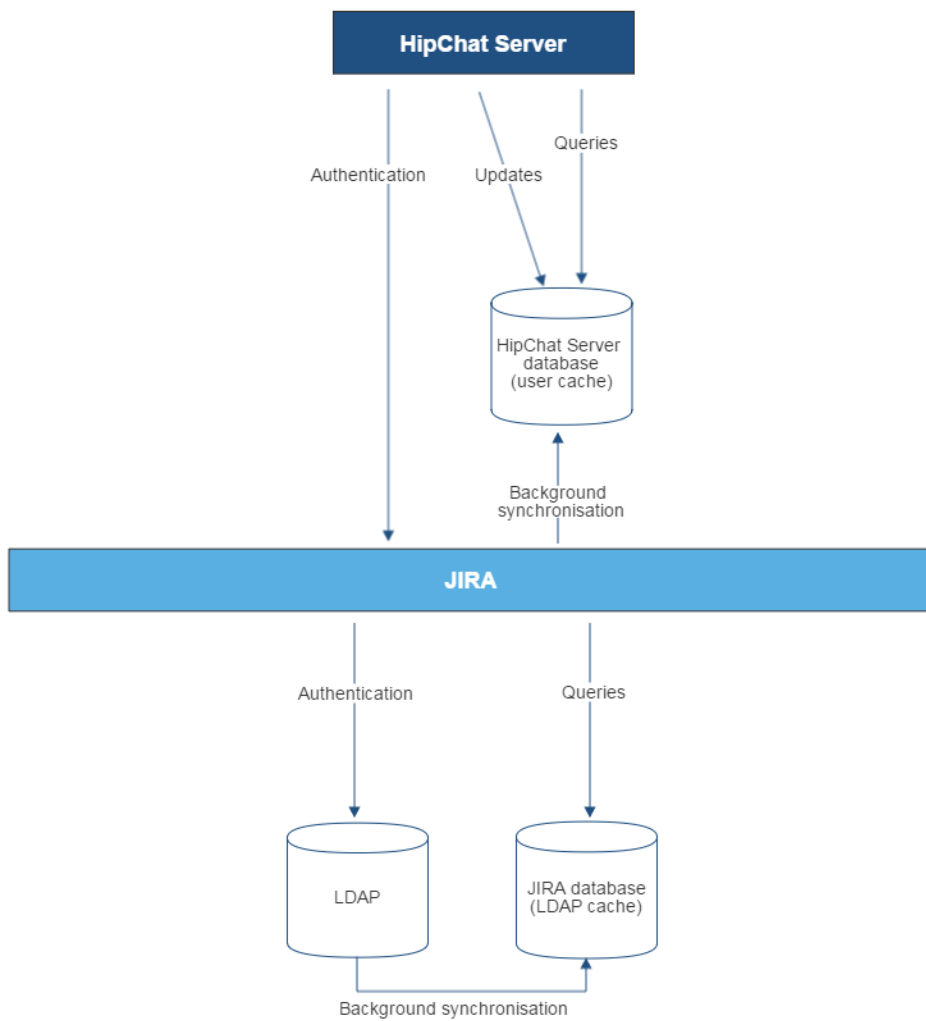


Diagram above: Hipchat Server connecting to a Jira application for user management, with the Jira application in turn connecting to an LDAP server.

Configuring the Google Apps Connector

The Google Apps connector is shipped with your Crowd installation. This is a Crowd application connector which allows single sign-on (SSO) to [Google Apps](#). If you wish to activate SSO between Crowd-connected applications and Google Apps, you will need to configure the Google Apps connector as described below.

On this page:

- [Background](#)
- [Prerequisites](#)
- [Step 1. Configuring the Crowd Application, Directory and Group Details](#)
- [Step 2. Generating your SSO Keys](#)
- [Step 3. Configuring Google Apps to Recognize Crowd](#)
- [Step 4. Verifying that a User can Log in to Google Apps](#)
- [More Information about the Google Apps Connector](#)
 - [Deleting the Keys](#)
 - [The Ins and Outs of SSO with Google Apps](#)
 - [Usernames must be the Same in Google Apps and Crowd](#)
 - [Other Authentication Frameworks and SAML Support](#)
- [An Example of Google Apps SSO in Action](#)

Background

When people refer to [Single sign-on](#) (SSO) they are usually referring to two things.

- Authentication - which userid and which password is used to confirm that a user is who they say they are
- SSO - authenticating for one application means that you don't have to authenticate to access another application (at least for a while and if you are using the same browser)

So in the case of Google Apps, the authentication is your Google Apps userid and password. Google Apps handles the SSO part with its [Google Accounts](#) site. When you log into Gmail in the morning you are actually being asked to enter your authentication information at the Google Accounts site, which then redirects your browser back to Gmail if you successfully authenticate. The Google Accounts remembers that you successfully authenticated this morning so that when you go to a Google Docs page later on that day, the redirect happens again without needing to reauthenticate. This all happens unseen by most Google Apps users.

Now Google Apps has the ability to change where it goes for its SSO functionality. A Google Apps administrator can [configure](#) just their Google Apps instance to use a different SSO. This could be Crowd, or any other SSO service. Crowd then becomes the master SSO service instead of Google Accounts. This means that logging into Gmail in the morning will take you to a Crowd authentication screen, *not* the Google Accounts. The redirection back to Gmail after a successful authentication happens just as before.

However this is not how OnDemand [integrates with Google Apps](#). In that case the SSO functionality remains with Google Accounts.

Prerequisites

Please note the following before you start:

- **Google Apps support for SSO:** To enable single sign-on in Google Apps, you will need the Premier, Education, or Partners edition of Google Apps. The free Standard Edition of Google Apps does not support SSO. See the [Google Apps documentation](#).

Step 1. Configuring the Crowd Application, Directory and Group Details

In this step, you will enter the application details for the Google Apps application connector in Crowd. You will manage access to Google Apps by associating Crowd directories and/or groups with the Google Apps application.

To define the Google Apps application details in Crowd:

1. Log in to the [Crowd Administration Console](#).
2. Click the **Applications** tab in the top navigation bar.
3. Click the link on the 'google-apps' application name.
4. If required, you can change the description. Please ensure that the **Active** checkbox remains ticked.
5. Click the **Directories** tab and select one or more user [directories](#) that contain the users who should have access to Google Apps.
6. To choose which users within the directory may authenticate against the application, either:
 - On the **Directories** tab, change **Allow all to authenticate** to **True**. This will allow all users in that directory to log in to Google Apps. (The default is **False**.)
 - OR**
 - On the **Groups** tab, use the **Add** button to select one or more [groups](#) of users.
7. Click the **Permissions** tab and set the [directory permissions for the application](#).
8. If required, you can change the application options on the **Options** tab:
 - **Lower Case Output** See [Enforcing Lower-Case Usernames and Groups for an Application](#).
 - **Enable Aliasing** See [Specifying a User's Aliases](#).
9. Click the **Configuration** tab and generate your SSO keys as described in [Step 2](#) below.

Screenshot: Google Apps application details in Crowd

🔗 google-apps

Details Directories Groups Users Permissions Authentication test Options Configuration

Name
The unique name that the application will use to authenticate against the Crowd framework as a client.

Description
A short description of the application. Often a URL is helpful.

Application type **Plugin**
 Active

Conception **05 Jun 2017, 09:43:47**

Last modified **05 Jun 2017, 09:44:08**

Step 2. Generating your SSO Keys

Now you will ask Crowd to generate a public and a private key for use in authenticating Crowd to Google Apps. (Google Apps calls the public key a 'verification certificate'.)

To generate your SSO keys:

1. In the Crowd Application Browser, as described in [Step 1](#) above, click the **Configuration** tab for the Google Apps application.
2. Click **Generate New Keys**.

Crowd will generate a public key and a private key, placing them in the `database`. When the keys have been generated, you will see a message *'DSA keys successfully generated and stored.'*

Screenshot: Configuring the Google Apps connector in Crowd

🌐 google-apps

Details Directories Groups Users Permissions Authentication test Options Configuration

Generate your Google Apps keys here. Then use the public key and the information below to set up SSO in your Google Apps control panel.

Sign-in page URL **http://localhost:8095/crowd/console/plugin/secure/saml/samlauth.action**

Sign-out page URL **http://localhost:8095/crowd/console/logoff.action**

Change password URL **http://localhost:8095/crowd/console/user/viewchangepassword.action**

Public key

Download

Generate new keys

Delete keys

Step 3. Configuring Google Apps to Recognize Crowd

In this step, you will log in to Google Apps as an administrator and enter the information required for Crowd to authenticate to Google Apps. This information consists of some Crowd URLs and the public key which you generated from Crowd.

To configure Google Apps to recognize Crowd:

1. Log in to your **Google Apps Dashboard** as a **Google Apps administrator**.
2. In Google Apps, click **Security**.
3. Click **Advanced Settings**.
4. Click **Set up single sign-on (SSO)**.
5. Copy the URLs from the Crowd configuration screen (see [above](#)) and paste them into the Google Apps screen.
6. Now you will upload the public key which Crowd generated for you in [Step 2](#) above:
 - Still in Google Apps, click **Browse** under 'Verification certificate'.
 - Navigate in Crowd to google-apps configuration and download the public key clicking on button Download next to the Public Key label.
 - Select the public key certificate (file name `DSAPublic.key`) and upload it to Google Apps.
7. If necessary for your network configuration, check **Use a domain specific issuer** and enter any required network masks in Google Apps. Please refer to the Google Apps documentation for guidance on these settings.
8. Save your changes in Google Apps.

Screenshot: Setting up SSO in Google Apps

Google Apps for thanksforcomingin.com - admin@thanksforcomingin.com [Inbox](#) [Calendar](#) [Help](#) [Sign out](#)

Google Premier Edition

Search accounts Search Help Center

Dashboard User accounts Domain settings **Advanced tools** Service settings-

[« Back to Advanced tools](#)

Set up single sign-on (SSO)

To set up SSO, please provide the information below. [SSO Reference](#)

Enable Single Sign-on

Sign-in page URL *
 URL for signing in to your system and Google Apps

Sign-out page URL *
 URL to redirect users to when they sign out

Change password URL *
 URL to let users change their password in your system

Verification certificate *

The certificate file must contain the public key for Google to verify sign-in requests. [Learn more](#)

Use a domain specific issuer

This must be checked if your domain uses an IDP Aggregator to handle SAML requests. If enabled, the issuer value sent in the SAML request will be `google.com/a/thanksforcomingin.com` instead of simply `google.com` [Learn more](#)

Network masks

Network masks determine which addresses will be affected by single sign-on. If no masks are specified, SSO functionality will be applied to the entire network.
 Use a semicolon to separate the masks. Example: (64.233.187.99/8; 72.14.0.0/16)
 For ranges, use a dash. Example: (64.233.167-204.99/32)
 All network masks must end with a CIDR. [Learn more](#)

[Terms of Service](#) - [Privacy policy](#) - [Suggest a feature](#) - [Google Home](#)
 ©2008 Google Inc.

Step 4. Verifying that a User can Log in to Google Apps

It is a good idea now to check that your users can log in to Google Apps.

To test a user's authentication to Google Apps:

1. In the Crowd Application Browser, as described in [Step 2](#) above, click the **Authentication Test** tab for the Google Apps application.
2. Enter a user's login details and verify the login. For more details, you can refer to [Testing a User's Login to an Application](#).


Congratulations! You have now configured Crowd for SSO with Google Apps.

More Information about the Google Apps Connector

Deleting the Keys

Once you have generated the keys, a **Delete Keys** button will appear on Crowd's configuration screen. Click this button to remove the keys from the database. This will disable SSO with Google Apps.

The Ins and Outs of SSO with Google Apps

- Single sign-on (SSO) applies only to the applications within Google Apps. The Google Apps administration section (control panel) does not support SSO.
- When you sign out of Google Apps, you will also be signed out of Crowd and all Crowd-connected applications. This is the usual SSO behavior.
- But when you sign out of Crowd, you will remain logged in to Google Apps even though you will be logged out of other Crowd-connected applications. (Reason: Google does not rely on a cookie, so there is no easy way for Crowd to tell Google you have signed out.)
 -  It would take some additional development to support single sign-out from Google Apps. If you would like to see this work undertaken, please vote for issue [CWD-1238](#).
- If you go directly to a Google Apps application without logging in to Crowd, Google Apps direct you to a Crowd login screen.
- The Crowd login screen for Google Apps will not offer a 'Forgotten your password' link. You cannot change your Crowd password via Google Apps. Instead, if you need to change your password please log in to Crowd directly, by going to this URL: <http://YOUR-CROWD-LOCATION:8095/crowd/>

Username must be the Same in Google Apps and Crowd

Username must exist in Google Apps as well as Crowd and a person's username must be the same in both Google Apps and Crowd. The Crowd Google Apps connector does not support the automatic adding of users. If a user exists in Crowd but not in Google Apps, then the user will not be able to log in to Google Apps.

Other Authentication Frameworks and SAML Support

Crowd currently supports SSO via SAML with Google Apps only. The following information is relevant to developers who may want to use Crowd's classes to develop a plugin that supports SAML authentication with other frameworks.

Crowd's SAML implementation meets the requirements for Google Apps SSO. As Google Apps supports a subset of the SAML 2.0 spec, any authentication framework that relies on the same subset should also be compatible. The Crowd implementation is capable of servicing SAML 2.0 authentication requests using the HTTP-Redirect binding. For more information on the Google Apps authentication protocol, check out [their SSO documentation](#).

An Example of Google Apps SSO in Action

Here's one example of how it might work:

- John raises an issue in Jira. In the issue description, he adds a link to a Google Apps document containing more details.
- He assigns the issue to Sarah.
- Sarah clicks the link and opens the document directly in Google Apps. No need to log in again, no need to remember a different password.

The screenshot shows the JIRA interface for an issue titled "Dynamic Menu Builder" in the "MYPROJECT-2" project. The issue is of type "New Feature", status "Open", and priority "Major". It is assigned to Sarah Maddox and reported by John Pumpkin. The issue was created and updated today at 11:43 AM. The environment is "The App 2.2". The description reads: "Please develop the Dynamic Menu Builder, as specified in [this Google Apps document](#)".

The screenshot shows a Google Docs document titled "Technical Specification" saved on August 30, 2008, at 1:37 PM by Sarah Maddox. The document content is as follows:

Technical Specification
Application: The App
Feature: Dynamic Menu Builder

Summary

This feature will dynamically build the menu structures, based on the user's previous selections.

RELATED TOPICS

- [Using the Application Browser](#)
- [Adding an Application](#)
- [Configuring the Google Apps Connector](#)
- [Mapping a Directory to an Application](#)
- [Effective memberships with multiple directories](#)
- [Specifying an Application's Address or Hostname](#)
- [Testing a User's Login to an Application](#)
- [Enforcing Lower-Case Usernames and Groups for an Application](#)
- [Managing an Application's Session](#)
- [Deleting or Deactivating an Application](#)
- [Configuring Caching for an Application](#)
- [Overview of SSO](#)
- [Configuring Options for an Application](#)
- [Enabling OpenID client app](#)
- [Allowing applications to create user tokens](#)
- [Disabling the OpenID client app](#)
- [Configuring how users log in](#)

[Crowd documentation](#)

Mapping a Directory to an Application

Mapping a [directory](#) to an application defines the user-base for an application. Directory mappings determine which user directories will be used when authenticating and authorizing a user's access request. Read more about [users and groups](#).

When you [defined an application](#), you chose a default directory for that application to use. Crowd also allows you to map multiple directories to each application. This allows each of your applications to view multiple user directories as a single repository.

To map a directory to an application:

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click **Applications**.
3. Click the application you wish to map.
4. Select the **Directories & groups** tab.
5. Select the new directory from the drop-down list and click **Add**.
The new directory will be added to the bottom of the list of mapped directories. You can drag & drop the newly created row to move a directory higher or lower in the order. [Why is directory order important?](#)

Want to speed up authentication?

You can optimize the authentication process by selecting **Optimize priority of cached directories**. Once you select this option, Crowd will go through its cache for all the directories listed in this screen looking for information on the authenticating user. If there's no such information in Crowd's cache, Crowd will automatically lower the priority of such directory on the list.

If a user was added recently, these directories might stay at the end of the list until Crowd's cache is updated.

6. In the **Directories & groups** tab, check **Determine the users' group membership using all directories**. [Learn more](#).
7. Choose which users within the directory may authenticate against the application. Select one of the following:
 - *Allow all users* within the directory to authenticate against the application:
 - i. In the **Actions** column, click the meatballs icon and select **Configure authentication**.
 - ii. Enable the **Allow all users from this directory to authenticate** check box, and click **Save**.
 - *Allow only specific groups of users* within the directory to authenticate against the application, see [Specifying which Groups can access an Application](#).
8. Click the **Permissions** tab and set the [directory permissions for the application](#).
Note that updates are handled differently, depending on the membership aggregation scheme you selected. See [Directory update operations](#).

Screenshot: 'Application Map Directories'

The screenshot shows the 'your jira' application configuration page in Crowd, specifically the 'Directories & Groups' tab. The page is divided into several sections:

- Search applications:** Includes links for 'Add application' and 'Remove application'.
- Directories & Groups:** The main section, containing:
 - Directory mappings:** A table listing mapped directories with their authentication status and group assignments.

Directory	Who can authenticate	Automatically assigned to	Actions
Crowd Internal	ALL	1 GROUP	...
My Active Directory	NO GROUPS	2 GROUPS	...
Company Open LDAP	NO ONE	NO GROUPS	...
 - Test Internal Directory:** A dropdown menu with an 'Add' button.
 - Actions:** A context menu for the 'My Active Directory' row, containing:
 - Configure authentication
 - Configure automatically assigned groups (highlighted)
 - Remove directory
- Directory aggregation:** A section titled 'CHOOSE HOW GROUP MEMBERSHIPS WORK FOR USERS IN MULTIPLE DIRECTORIES' with a checkbox for 'Aggregate group memberships across directories'.

For each directory mapped to the application you can also define groups that users will be automatically assigned to, when they first log in to the application. See [Automatically Assigning Users to Groups](#) for more details.

Specifying the Directory Order for an Application

When you map multiple directories to an application, you also need to define the directory priority order. The directory order is used for the following:

Authentication

Authentication only relies on the groups you mapped to the application. Users are authenticated if they belong to a group mapped to the application in the first directory where they exist, or if that directory is mapped to the application using the **Allow all users from this directory to authenticate** option.

Authorization

When multiple directories are mapped to an integrated application, and duplicated usernames and group names are used across those directories, the effective group memberships for *authorization* are determined on the basis of the membership aggregation scheme that has been applied.

In particular, the non-aggregating membership scheme depends on the directory order to determine access permissions for a user.

See [Effective memberships with multiple directories](#) for more information.

Directory updates

When a user is added to a group, Crowd adds them to the first directory it has access to in priority order. This applies to both the aggregating and non-aggregating membership schemes.

When a user is removed from a group, the behavior depends on the membership scheme:

- With non-aggregating membership, the user is only removed from the group in the first directory the user is found in.
- With aggregating membership, the user is removed from the group in *all* directories the user is found in.

See [Directory update operations](#) for an explanation of the membership aggregation schemes.

Specify the directory order

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click the **Application** tab.
3. Click **View** for the application.
4. Click the **Directories & groups** tab to display a list of directories that are currently mapped to the application.
5. Drag & drop rows to move a directory higher or lower in the order:

demo

- Details
- Directories & groups
- Users
- Permissions
- Remote addresses
- Authentication test
- Options

Directory mappings

When authenticating a user, Crowd checks directories based on their priority on the list. If a given directory is unavailable or doesn't contain the user, Crowd will move on to the next one, and so on. In case of duplicated users, the first one found will be used. [Learn more](#)

You can also select default groups to which users will be automatically assigned once authenticated.

If you've made any changes to automatically assigned groups, make sure to change directory configuration on your application side.

[Tell me how to do that](#)

Directory	Who can authenticate	Default groups	Actions
Customers	1 GROUP	UNASSIGNED	...
Employees	ALL GROUPS	1 ASSIGNED	...

Crowd

Directory aggregation

- Determine the users' group memberships using all directories.

If not selected, users' group memberships are determined using only the directory they logged in from. [Learn more](#)

Specifying an Application's Directory Permissions

When you [map a directory to an application](#), you can also define the application's ability to add/update/delete users and groups in the directory. To do this, use the 'Permissions' tab in the 'View Application' screen.

Directory permissions are defined at two levels:

1. **Directory-level permissions** are defined on the 'Permissions' tab of the 'View Directory' screen. These permissions apply to each application mapped to the directory, unless the application has its own application-level permissions.
2. **Application-level directory permissions** are defined on the 'Permissions' tab of the 'View Application' screen. If a permission is enabled at directory level, you can enable it for a specific application. For example, you could enable the 'Add User' permission on the 'Customers' directory in Jira but disable the permission for Confluence.


Take a look at an [example](#).

Disabling a directory-level permission will override any permissions enabled at application level. If a permission is enabled at application level and then subsequently disabled at directory level, the directory-level permission will apply. (The application-level permissions will be 'remembered' and will apply again if re-enabled at directory level.)

How do directory permissions affect the Crowd application (Crowd Administration Console)?

- If a particular permission is turned off at directory level, then **no** application can perform the related function - not even the Crowd application. So, for example, if you disable the 'Remove User' permission for a directory, then the Crowd Administration Console will not allow you to delete a user from that directory.
- The Crowd application is not bound by application-level permissions, because any user who could log into the Crowd application could change the application-level permissions for the Crowd application anyway.

For details on directory-level permissions, refer to the instructions on [specifying directory permissions](#). Below are instructions on setting the application-level directory permissions.

Permission	Description
Add Group	Allows the application to add groups to the selected directory.
Add User	Allows the application to add users to the selected directory.
Modify Group	Allows the application to modify groups in the selected directory.
Modify User	Allows the application to modify users in the selected directory.
Remove Group	Allows the application to delete groups from the selected directory.
Remove User	Allows the application to delete users from the selected directory.  Consider carefully whether you allow the deletion of users, as some applications contain historical data, e.g. documents that the user has created. Read more .

When you initially [map a directory to an application](#), all of the application's permissions are enabled by default. But note that disabling a directory-level permission will override any permissions enabled at application level.

To set the directory permissions for an application,

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click **Applications**.
3. Click the application you want to edit.
4. Click the **Permissions** tab.
This displays a list of directories that are currently mapped to the application, and a set of permission check-boxes.
5. From the drop-down list, select a **directory**.
6. Select permissions you wish to allow this application to perform on the selected directory.

Screenshot: Setting directory permissions for an application

Jira

Details Directories & groups Users Permissions Remote addresses Authentication test Options

SSO

Please select a directory and then choose the permissions you wish to allow this application to perform on the selected directory.

Directories

Permissions

- Add group**
Allow groups to be added to the directory.
- Add user**
Allow users to be added to the directory.
- Modify group**
Allow groups to be modified in the directory.
- Modify user**
Allow users to be modified in the directory.
- Modify group attributes**
Allow group attributes to be modified in the directory.
- Modify user attributes**
Allow user attributes to be modified in the directory.
- Remove group**
Allow groups to be removed from the directory.
- Remove user**
Allow users to be removed from the directory.

i On the application permissions screen, the words '**(disabled globally)**' will appear next to any permission that is disabled at directory level.

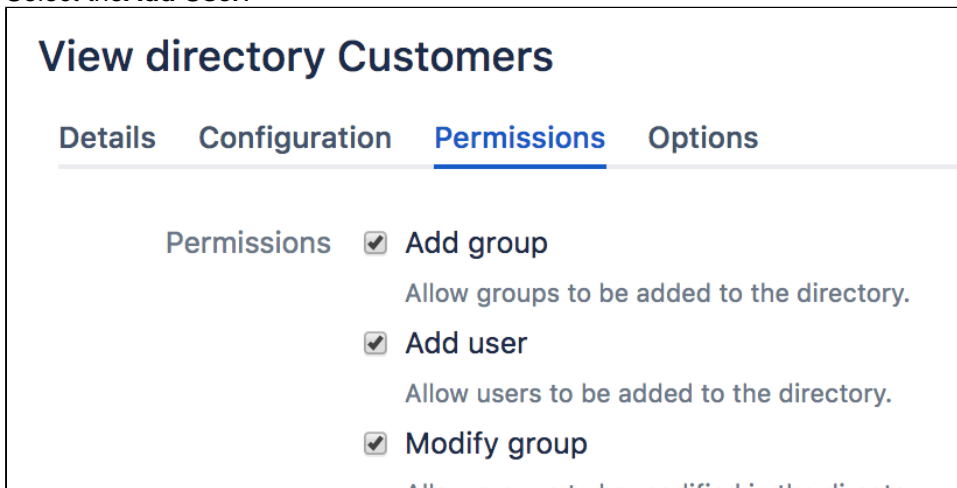
Example of Directory Permissions

For the purpose of this example, let's assume that you want to:

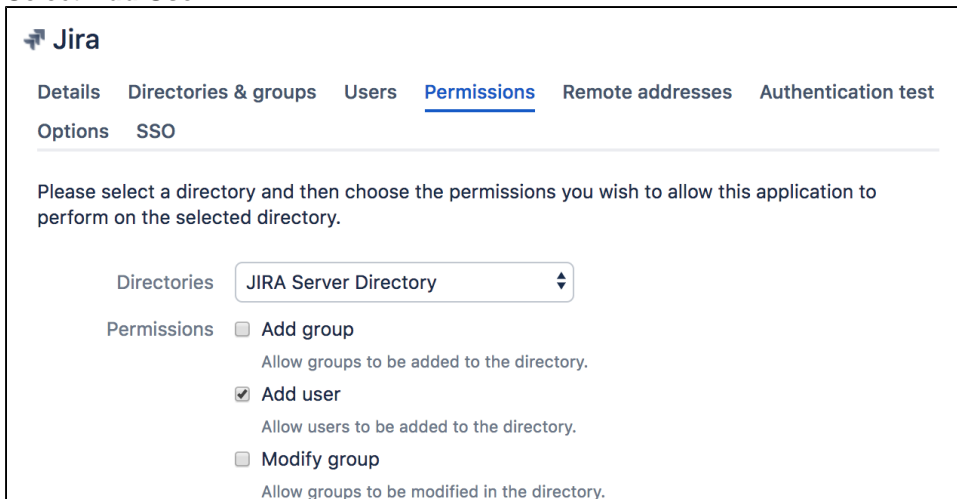
- Allow self-registration (automatic signup) of new users in your 'Customers' directory via [Jira](#), and
- Disable self-registration via [Confluence](#).

Here's how you would set the directory-level and application-level permissions in Crowd.

1. At directory level, enable the 'Add User' permission (and any other permissions you want):
 - a. Log on to the [Crowd Administration Console](#).
 - b. In the top navigation bar, click **Directories**.
 - c. From the drop-down list, select the Customers directory.
 - d. Click the **Permissionstab**.
 - e. Select the **Add User**.



2. At application level, make sure the 'Add User' permission is enabled for the Jira application:
 - a. In the top navigation bar, click **Applications**.
 - b. Click on your Jira application.
 - c. In the application screen, click the **Permissionstab**.
 - d. Select the 'Customers' directory.
 - e. Select **Add User**.



3. At application level, disable the 'Add User' permission the Confluence application:
 - a. In the top navigation bar, click **Applications**.
 - b. Click on your Confluence application.
 - c. In the application screen, click the **Permissionstab**.
 - d. Select the 'Customers' directory.

e. Deselect **Add User**.

Confluence

Details Directories & groups Users Permissions Remote addresses Authentication test

Options SSO

Please select a directory and then choose the permissions you wish to allow this application to perform on the selected directory.

Directories

Permissions

- Add group
Allow groups to be added to the directory.
- Add user
Allow users to be added to the directory.
- Modify group
Allow groups to be modified in the directory.

In summary:

With the above application permissions, a person will be able to sign up for a user account via JIRA and this user will be created in the 'Customers' directory, but they will not be able to sign up for an account via Confluence.

Viewing Users in Directories Mapped to an Application

The application **Users** tab shows all the users in all the directories mapped to the selected application. You will also see basic information for each user, including the user's full name, username and email address. If the user has an **alias** for the selected application, the alias will appear too.

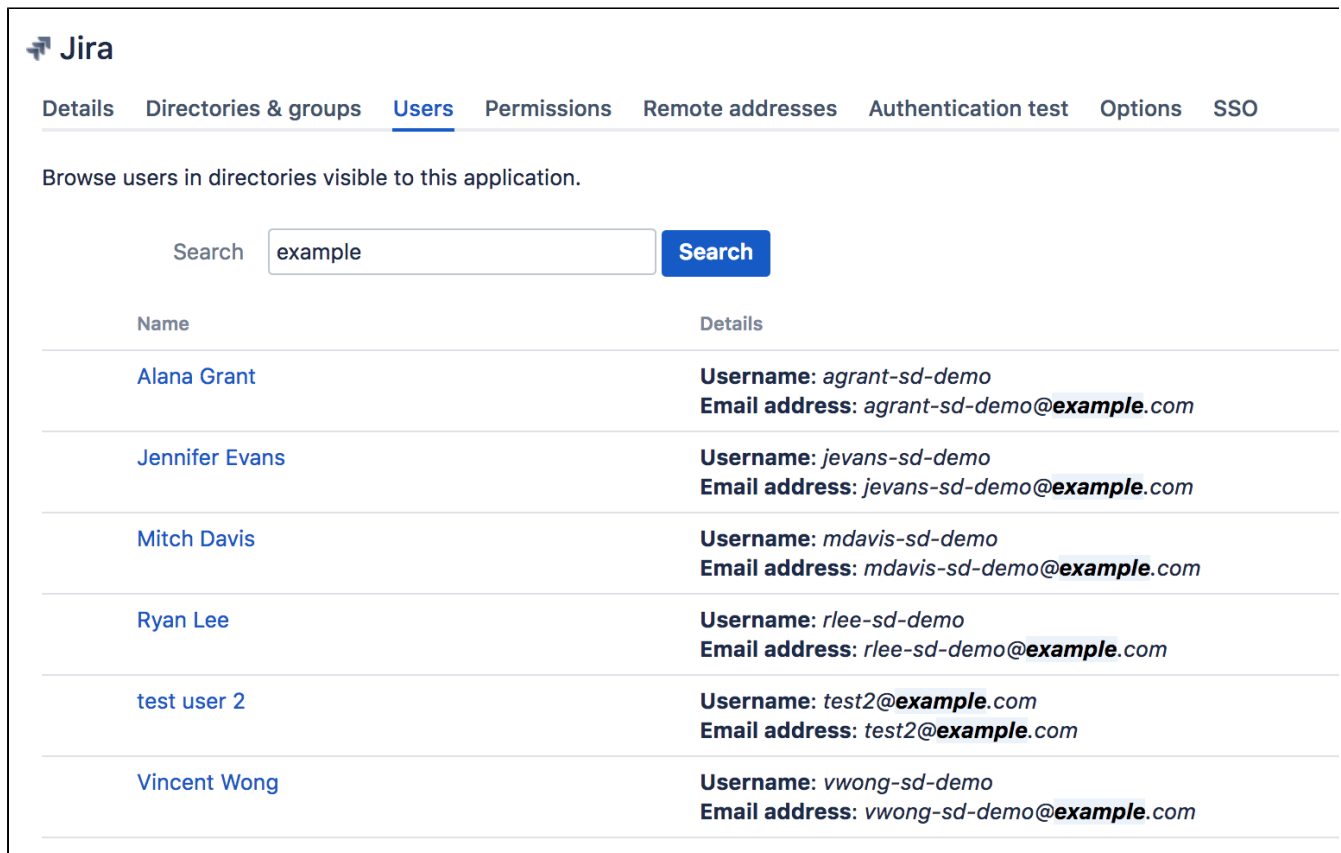
Effect of access-based synchronization on displayed users

The data displayed in the Users tab depends on whether you use access-based synchronization. If this feature is enabled, then the tab will include only users who have access to the particular application. If it's disabled (which means that all users are synced from the mapped directories), then we'll show all users, even if your application only allows access for some of them.

To see the users visible to an application,

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click **Applications**.
3. Click the name of the application you want to view.
The **View Application** screen appears.
4. Click the **Users** tab.
5. Enter your search criteria in the **Search** textbox. You can enter all or part of the user's name, email address or username. Leave the search box empty to match all users.
6. Click **Search** button.

Screenshot: Viewing users for an application



The screenshot shows the Jira application interface. At the top, there is a navigation bar with tabs: Details, Directories & groups, **Users**, Permissions, Remote addresses, Authentication test, Options, and SSO. Below the navigation bar, there is a heading "Browse users in directories visible to this application." and a search bar with the text "example" and a "Search" button. Below the search bar, there is a table with two columns: "Name" and "Details". The table contains the following data:

Name	Details
Alana Grant	Username: agrant-sd-demo Email address: agrant-sd-demo@example.com
Jennifer Evans	Username: jevans-sd-demo Email address: jevans-sd-demo@example.com
Mitch Davis	Username: mdavis-sd-demo Email address: mdavis-sd-demo@example.com
Ryan Lee	Username: rlee-sd-demo Email address: rlee-sd-demo@example.com
test user 2	Username: test2@example.com Email address: test2@example.com
Vincent Wong	Username: vwong-sd-demo Email address: vwong-sd-demo@example.com

Specifying which Groups can access an Application


You can specify which users are allowed to authenticate against each application. For each [mapped directory](#), you can either allow *all* users within the directory to authenticate with the application, or just particular [groups](#) within the directory. You can then [assign group membership](#) to each user.

For example, the default group `crowd-administrators`, which is automatically created in the default directory that you specified [during setup](#), is allowed to access the [Crowd Administration Console](#). This means that users who belong to the group `crowd-administrators` are allowed to log in to the Crowd Administration Console (assuming they supply a valid password).

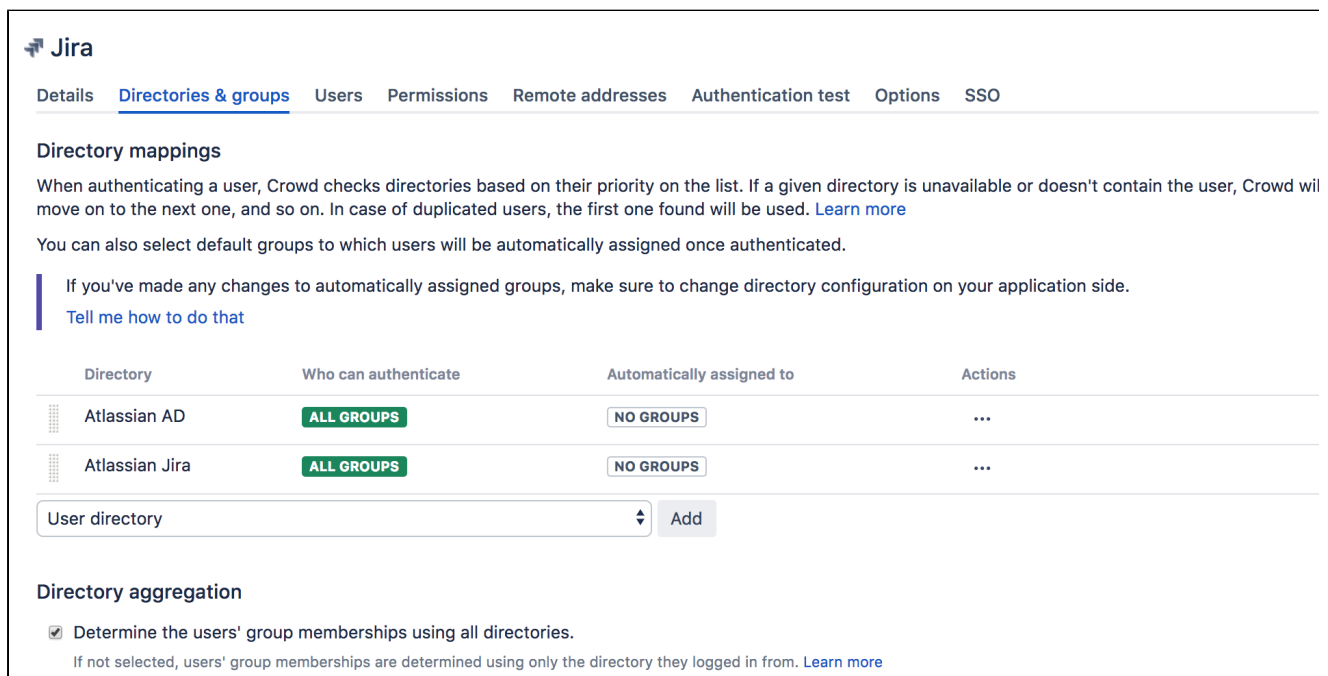
This setting will override any permissions configured in a client application. For example, even if the `test-users` group is given the `Can Use` permission in Confluence, if they aren't a mapped group as specified on this page, they will be unable to authenticate. This does not prevent usernames and groups from appearing in the client application.

To allow a group to access an application,

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click **Applications**.
3. Click the name of the application you want to link that corresponds to the application you wish to map.
4. In the application screen, click the **Directories & groups** tab.
This displays a list of groups that currently have access to the application.
5. From the drop-down list, select the the group and click **Add**.

 Alternatively, you can allow *all* users from a particular directory to authenticate against the application. See [Mapping a Directory to an Application](#).

Screenshot: Application Specify Groups



Jira

Details **Directories & groups** Users Permissions Remote addresses Authentication test Options SSO

Directory mappings

When authenticating a user, Crowd checks directories based on their priority on the list. If a given directory is unavailable or doesn't contain the user, Crowd will move on to the next one, and so on. In case of duplicated users, the first one found will be used. [Learn more](#)

You can also select default groups to which users will be automatically assigned once authenticated.

If you've made any changes to automatically assigned groups, make sure to change directory configuration on your application side.
[Tell me how to do that](#)

Directory	Who can authenticate	Automatically assigned to	Actions
Atlassian AD	ALL GROUPS	NO GROUPS	...
Atlassian Jira	ALL GROUPS	NO GROUPS	...

User directory Add

Directory aggregation

Determine the users' group memberships using all directories.
If not selected, users' group memberships are determined using only the directory they logged in from. [Learn more](#)

Specifying which users are synced based on their access rights

After you allow only some groups to access the application, you can add the same restrictions to users and groups that are synced with the application. For more info, see [Syncing users based on their access rights](#).

Syncing users based on their access rights

When you map a user directory to an application in Crowd, you can choose which users are synced with the application based on their access rights to it. It might be useful to limit the synced users to only those who can actually access the application, as syncing anyone else is redundant in most cases.

To choose which users will be synced with your application:

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click **Applications**, and choose your application.
3. Select the **Directories & groupstab**.
4. Scroll down to **Access-based synchronization**, and choose one of the options.

The screenshot shows the 'Directories & groupstab' configuration page. At the top, there are columns for 'Directory', 'Who can authenticate', 'Automatically assigned to', and 'Actions'. The first row shows 'Atlassian Crowd server' with '1 GROUP' under authentication and 'NO GROUPS' under assignment. Below this is a search bar with 'Azure AD' and an 'Add' button. The 'Access-based synchronization' section is active, with the following options:

- All users and groups
Sync all users and groups, regardless of their access rights, to keep the structure of your user directory. Choose this option if you're not sure about the remaining ones.
- All groups, but only users with access rights
Sync all groups so users with access rights can keep all of their group memberships, but filter out users without access rights.
- Only users and groups with access rights
Sync only users and groups that can access this application. These users will lose their memberships in groups that haven't been synced.

Good to know

Here are some additional details:

- Your settings will apply to all Crowd APIs used by your applications
- Membership aggregation and nested groups are supported.
- If a user exists in multiple directories, their access rights in the first one will decide whether they're synced or not.
- You can only use full synchronization, the incremental one isn't supported.
- When **All groups, but only users with access rights** is enabled, applications will not be able to create users in Crowd.
- When **Only users and groups with access rights** is enabled, applications will not be able to create users and groups in Crowd.

How syncing works with aggregated group memberships

You might encounter some confusing cases if you're using aggregated group memberships. If something isn't synced the way you expect it, have a look at the use cases we've described below.

Sample scenario

You have two directories mapped to an application. In Directory 1, the user *john* belongs to *group A*, while in Directory 2 *group B*. You also have the **Determine the users' group memberships using all directories** option enabled.

Here are some use cases to show you which groups the user will belong to after syncing. Note that we're changing syncing and authentication options for each case:

1. Syncing: **All users and groups**
Who can authenticate: N/A
In this case, *john* will be a member of *group A* and *group B*, as everything is synchronized.
2. Syncing: **All groups, but only users with access rights**
Who can authenticate: Directory 1: *group A*; Directory 2: *group B*.
In this case, *john* will be a member of *group A* and *group B*. Both of these groups are allowed to authenticate, so he's treated as a user with access rights in both directories, keeping all of his group memberships.

3. Syncing: **All groups, but only users with access rights**

Who can authenticate: Directory 1: *group A*; Directory 2: *group C* (some other group *john* doesn't belong to)

In this case, *group A* and *group B* will be synchronized, but *john* will be a member of *group A* only. That's because *group B* isn't allowed to authenticate so *john* from Directory 2 is treated as a user without access rights. He's a member of *group B*, but not *group C* that's allowed for this directory.

4. Syncing: **Only users and groups with access rights**

Who can authenticate: Directory 1: *group A*; Directory 2: *group B*

In this case, *john* will be a member of *group A* and *group B*. Both of these groups are allowed to authenticate, so he's treated as a user with access rights in both directories, keeping all of his group memberships.

5. Syncing: **Only users and groups with access rights**

Who can authenticate: Directory 1: *group A*; Directory 2: *group C* (some other group *john* doesn't belong to)

In this case, *group B* won't be synchronized at all, because it doesn't have access rights. Likewise, *john* from Directory 2 is treated as a user without access rights, similarly to Case 3 above. *Group A* will be synchronized and *john* will be a member of it.

We want to bring your attention to Case 3 that might appear confusing. In this case, the users should be treated as separate – one with access rights, and one without them. As *john* from Directory 2 doesn't have access rights (*group B* can't authenticate for this directory), he isn't synced and his group memberships aren't taken into account.

Troubleshooting

Having problems? Check the details below:

This might be because of the following reasons:

- You have a Server license. This feature is only in Data Center.
- You allow all groups from all directories to authenticate. In this case, there's no reason to limit synced users as all of them need access.

Missing users:

- Make sure the user belongs to at least one of the groups with access rights.
- Make sure it's not a shadowed user (see Limitations)

Missing groups and group memberships:

- If you chose **Only users and groups with access rights**, make sure the group actually has access rights.

Effective memberships with multiple directories

This page describes how Crowd determines group memberships for an integrated application that is mapped to multiple directories, where duplicate user names and group names are used across those directories.



Information gathered in this page do not apply to application which use only one directory. With more than one directory, this page applies both to user memberships in groups, and group memberships in other groups (nested groups).

In Crowd 2.8, and later versions, there are two different schemes that Crowd can use with multiple directories. We have called these '[aggregating membership](#)' and '[non-aggregating membership](#)', and have described them below.

These schemes are only used to determine the group memberships that the integrated application uses for authorization purposes. Authentication is determined on the basis of the group mappings for an integrated application. See [Specifying which Groups can access an Application](#).

When you map multiple directories to an integrated application (with duplicate user names), non-aggregating membership is applied by default. You can easily configure the application to use aggregating membership, as [described below](#).

Non-aggregating membership

This is the 'masking' case Crowd sees the directories mapped to an application from the top down, and will determine the effective group memberships based on the *first instance* of a user that is detected, working from the highest priority directory down to the lowest. Occurrences of the user in a lower priority directory are 'masked', and are not effective.

For example, Crowd will only consider Sam to be a member of the Admin group if that membership exists in the first directory in which Sam is found. If Sam is not a member of the Admin group in the first directory in which Sam is found, Crowd will not consider that membership *even if Sam is a member of Admin in a lower directory*.

Note that the priority order of the mapped directories is essential to this scheme. See [Specifying the Directory Order for an Application](#).

In the diagram of non-aggregating membership below, users or groups in a directory 'mask' themselves in lower priority directories.

For non-aggregating memberships:

A user effectively belongs to only those groups where the user is a group member in the highest directory that contains the user.

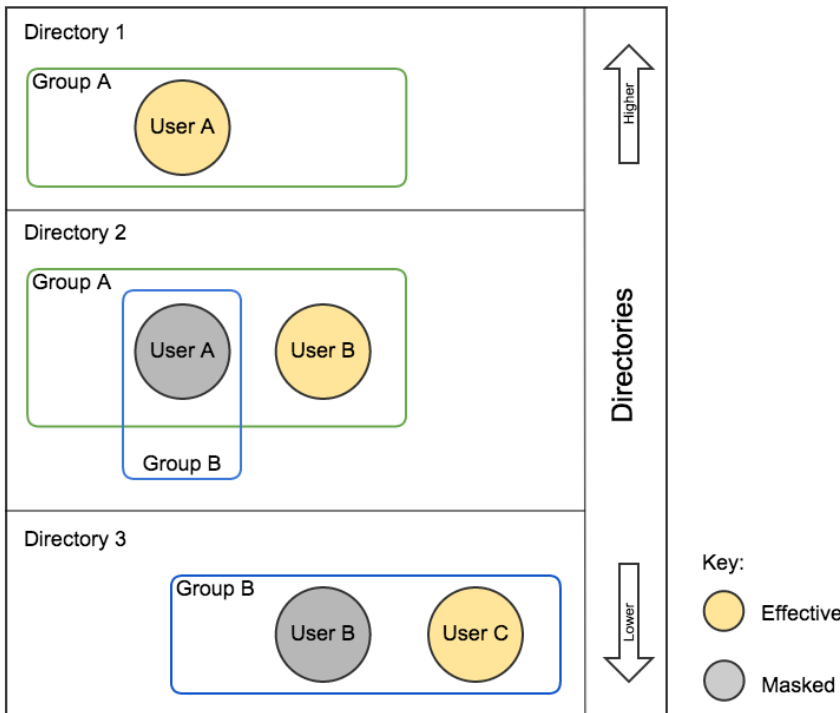
So, in the diagram:

- User A is only a member of Group A.
- User B is only a member of Group A.
- User C is only a member of Group B.

A group's members are all the 'top-most' users that belong to the group. If a group contains the user in a 'lower' directory, that membership is 'masked out' and is not effective.

So, in the diagram:

- Group A contains Users A and B.
- Group B only contains User C.

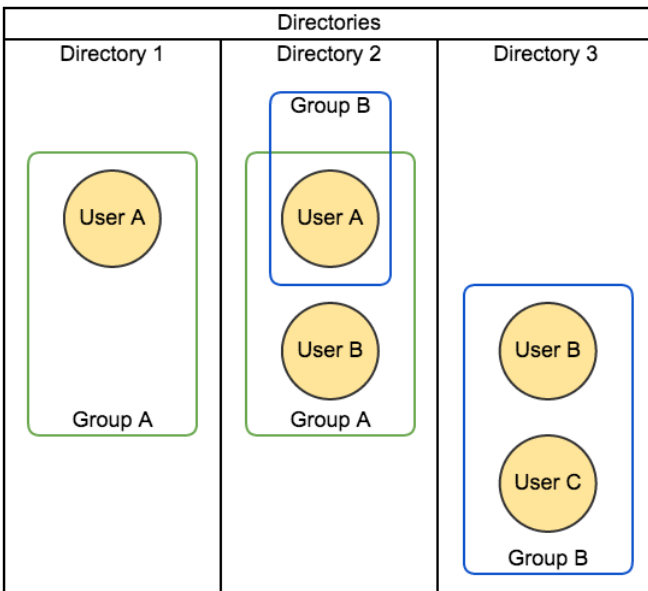


Aggregating membership

This is the 'blending' case - Crowd sees across all the directories mapped to an application, and will determine the effective group memberships based on a union of the directories.

For example, if Sam is a member of the Admin group in any directory, Crowd will consider Sam to be a member of the Admin group.

In the diagram of aggregating membership below, effective membership is a blend of the memberships in all mapped directories.



For aggregating memberships:

A user effectively belongs to all the groups of which the user is a member, in any directory.

So, in the diagram:

- User A is a member of Groups A and B.
- User B is a member of Groups A and B.
- User C is a member of Group B.

A group effectively contains all the users that belong to that group in any directory.

So, in the diagram:

- Group A contains Users A and B.
- Group B contains Users A, B and C.

Understand how access-based synchronization affects membership aggregation

If you're using aggregated group memberships and also sync users based on their access rights to the application, we recommend that you have a look at [Syncing users based on their access rights](#), where we've described and explained a few cases that might appear confusing.

Configure the aggregation scheme for an application

When you map multiple directories to an integrated application in Crowd, non-aggregating membership is applied by default.

You can change the aggregation scheme for each integrated application by checking, or clearing, the **Aggregate group memberships...** checkbox on the **Directories** tab for the application:

Aggregate group memberships across directories

If checked, group memberships will be effective if they are present in any active directory.

Update

The aggregation scheme applies across all the directories mapped to the application.

Directory update operations

When a user is added to a group, they are only added to the first writeable directory available, in priority order. This applies for both the aggregating and non-aggregating membership schemes.

When a user is removed from a group, the behavior depends on the membership scheme:

- With non-aggregating membership, the user is only removed from the group in the first directory the user is found in.
- With aggregating membership, the user is removed from the group in *all* directories the user is found in.

Inactive users

The membership schemes described above are not used when Crowd determines if a user should be able to authenticate.

Crowd only checks if the user is active in the first (highest priority) directory in which they are found when determining authentication.

For example, an application in Crowd is mapped to two directories: Crowd Internal Directory (primary) and an AD Delegated Authentication Directory (secondary).

- User A is inactive in the primary directory
- User A is active in secondary directory

Result: Crowd rejects access (authentication), because user A is first found in the primary directory, and the user is inactive there.

Specifying an Application's Address or Hostname

To ensure that your Crowd server can be used by legitimate applications only, Crowd will allow applications to log in only from known addresses. This means that you need to specify the IP address(es) and/or hostname(s) of each application.


When you [add a new application](#), you will specify the application's IP address. After adding the application, you can update the IP address if necessary, as described below. In some cases, you may need to add the applicable host name as well as the IP address.

IP address and/or host name?

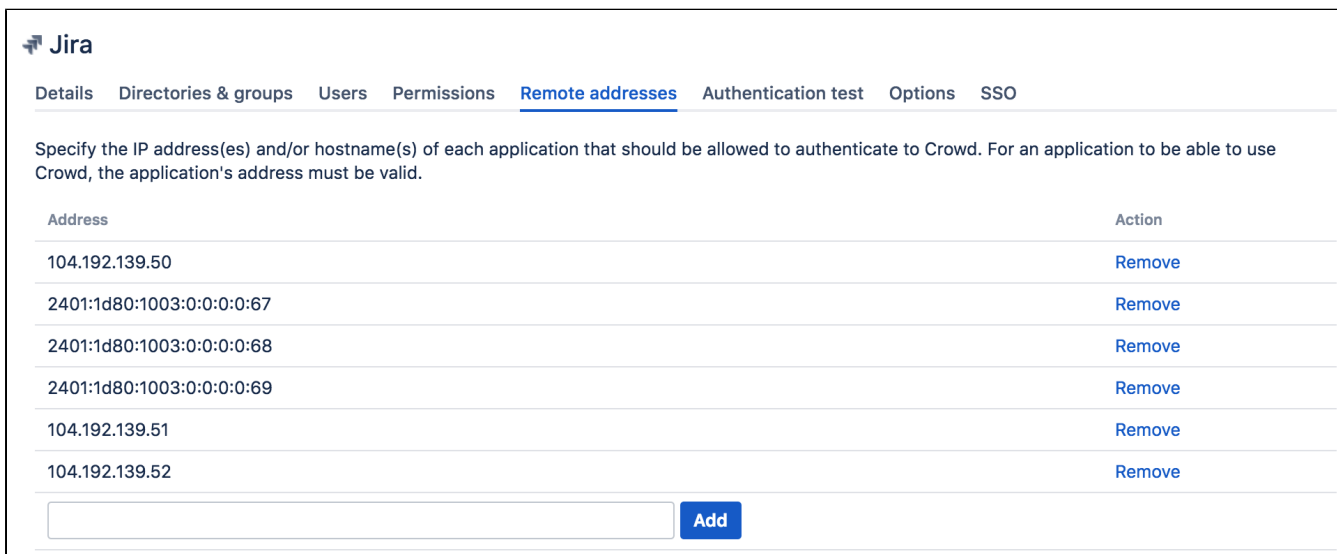
You should always specify the application's IP address. In addition, you may need to give a host name as well as the IP address. Some application servers may pass the host name to Crowd, instead of the IP address. If this happens, Crowd will not grant the application's authorization request unless Crowd recognizes the host name.

To specify an application's IP address or hostname:

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click **Applications**.
3. In the [Application Browser](#), click the name of the application you wish to update
4. In the **View Application** screen, click the **Remote Addresses** tab.
You will see a list of IP addresses and hostnames that are currently mapped to the application.
5. Type the new IP address or hostname in the Address field.
Possible values are:
 - A full IP address, e.g. 192.168.10.12.
 - A wildcard IP range, using CIDR notation, e.g. 192.168.10.1/16. For more information, see the introduction to [CIDR notation on Wikipedia](#) and [RFC 4632](#).
 - A host name, e.g. myhost.com.

 Note: If an application running on the same server needs to access Crowd, you may need to add 'localhost' as well as '127.0.0.1' to the list of IP addresses and hostnames.
6. Click **Add**.
The new address will be added to the bottom of the list.

Screenshot: Application addresses



Jira

Details Directories & groups Users Permissions Remote addresses Authentication test Options SSO

Specify the IP address(es) and/or hostname(s) of each application that should be allowed to authenticate to Crowd. For an application to be able to use Crowd, the application's address must be valid.

Address	Action
104.192.139.50	Remove
2401:1d80:1003:0:0:0:67	Remove
2401:1d80:1003:0:0:0:68	Remove
2401:1d80:1003:0:0:0:69	Remove
104.192.139.51	Remove
104.192.139.52	Remove

[Add](#)

Troubleshooting

- **A common problem: Application not connecting with Crowd.** For an application to be able to use Crowd, the application's address must be valid and the application must be active. Ensure the 'Active' check box is ticked on the application 'Details' tab.

Testing a User's Login to an Application

You can use an application's **Authentication Test** tab to verify that a user will be able to log in to a given application, based on the user, directory and group associations in Crowd.

How does the test work?

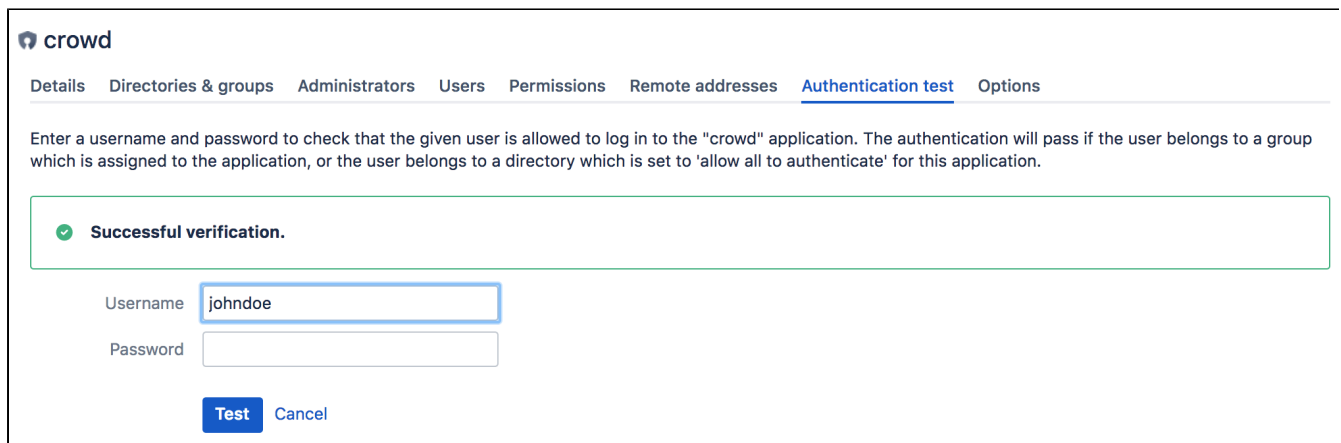
1. You enter the username and password of the user you wish to verify has access to a given application.
2. Crowd searches for the user with that username in the application's [mapped directories](#), and verifies the password.
3. If the user is not found or the password is invalid, the authentication fails the test.
4. Crowd checks whether the directory is set to [allow all to authenticate](#).
5. If all can authenticate, the test passes.
6. Else, Crowd checks the group(s) to which the [user belongs](#) and verifies whether those groups have [access to the application](#).
7. If the user belongs to an allowed group, the test passes, otherwise it fails.

How do I do it?

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click **Applications**.
The [Application Browser](#) displays.
3. Click the name of the application you want to verify.
4. In the **View Application** screen, click **Authentication Test** tab.
5. Enter the **Username** and **Password** that you wish to verify.
6. Click the **'Update'** button.
7. A message appears above the 'Username', displaying one of the following:
 - **'Successful verification'** The authentication has passed the test.
 - **'Invalid verification'** The authentication has failed the test.

Below are some suggestions for the next steps you can take in each case.

Screenshot: Authentication test showing successful verification



The screenshot shows the 'crowd' application interface. At the top, there is a navigation bar with 'crowd' and several menu items: 'Details', 'Directories & groups', 'Administrators', 'Users', 'Permissions', 'Remote addresses', 'Authentication test' (which is highlighted), and 'Options'. Below the navigation bar, there is a text box containing the instruction: 'Enter a username and password to check that the given user is allowed to log in to the "crowd" application. The authentication will pass if the user belongs to a group which is assigned to the application, or the user belongs to a directory which is set to 'allow all to authenticate' for this application.' Below this text box, there is a green-bordered box containing a green checkmark icon and the text 'Successful verification.'. Below this box, there are two input fields: 'Username' with the value 'johndoe' and 'Password' which is empty. At the bottom of the form, there are two buttons: 'Test' (highlighted in blue) and 'Cancel'.

Successful Verification

If this test is successful, but the user is having trouble authenticating to an application, then the problem is caused by the connection between the application and Crowd rather than by user authentication.

Next step: Check the **'Application Sessions'** tab in the [Session Browser](#) to see if the application is connected to Crowd.

Failed Verification

If the test declares the login to be invalid, this means that the configuration is incorrect within Crowd.

Next steps:

Check the following - all must be true to allow successful verification.

- The user must belong to a directory which is [mapped to this application](#).
- The password you used must be valid. In particular, check that the password is the one specified in the **first** directory in which the user appears. (If the user belongs to more than one directory, Crowd uses the first directory in which the user appears, as determined by the [directory order](#).)
- Either:
 - The directory must be set to [allow all to authenticate](#).
 - OR:
 - The user must belong to a [group](#) which has [access to the application](#).

Enforcing Lower-Case Usernames and Groups for an Application

In some cases you may wish to convert usernames and group names to lower case when passing them to an application. You can set an option for each application, as described below. When the option is set, Crowd will convert upper-case and mixed-case information obtained from your user directory to lower case before passing the information to the application. The conversion is applied to the following information:

- Usernames
- Group names
- Group memberships

If you set this option for an application, the conversion will apply to **all** directories [mapped to the application](#).

This option is useful in the following situations:

1. First situation: Existing application-to-directory integration:
 - You have previously integrated an application that enforces lower-case usernames (e.g. `jsmith`) with a corporate directory which allows mixed-case usernames (e.g. `JSmith`). Examples of such applications are [Jira](#) and [Confluence](#).
 - You have existing usernames in the application, which are therefore all lower case.
 - Now you want to integrate the application with Crowd.
2. Second situation: You have a custom application which demands lower-case usernames and cannot do the conversion itself.



Check your options carefully

You should only enforce lower-case conversion if you are in a situation as described above. There is no need to enforce lower-case conversion if you are starting out afresh with a Crowd-to-Jira or Crowd-to-Confluence integration. When lower-case conversion is not enforced, Crowd's behavior is **case-insensitive but case-preserving** it will ignore case when comparing usernames etc ('`JSmith`' = '`jsmith`') and it will preserve case when passing information between applications and directories ('`JSmith`' remains '`JSmith`'). This results in the expected behavior in the Crowd-integrated directories as well as the Crowd-integrated applications such as Jira and Confluence.

To enforce lower-case conversion for an application,

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click **Applications**.
The [Application Browser](#) appears.
3. Click the name of the application you want to configure.
4. In **View Application**, click the **Options** tab.
5. Select **Lower Case Output**.
6. Click **Update**.

Screenshot: Application Options

Jira

Details Directories & groups Users Permissions Remote addresses Authentication test Options SSO

Lower case output
Convert all users and groups to lower case when passing the data to the application. This can be used to achieve case insensitivity for applications when the underlying directories contain mixed-cased data.


Enable aliasing
With aliasing enabled, users can have a different username (alias) for the "test jira" application. This is useful if "test jira" does not support user renaming, but may impact the performance of incremental directory synchronisation for this application. See the [documentation](#) for more information.

Managing an Application's Session


Crowd allows you to see a list of all applications currently logged in to the [Crowd framework](#). This is effectively a list of the applications which currently have users logged in to them, since an application will only ever log in to the Crowd framework when it needs to authenticate a user.


You can also force any session to expire, that is, you can log the application out of Crowd.

To see which applications are currently logged in to Crowd:

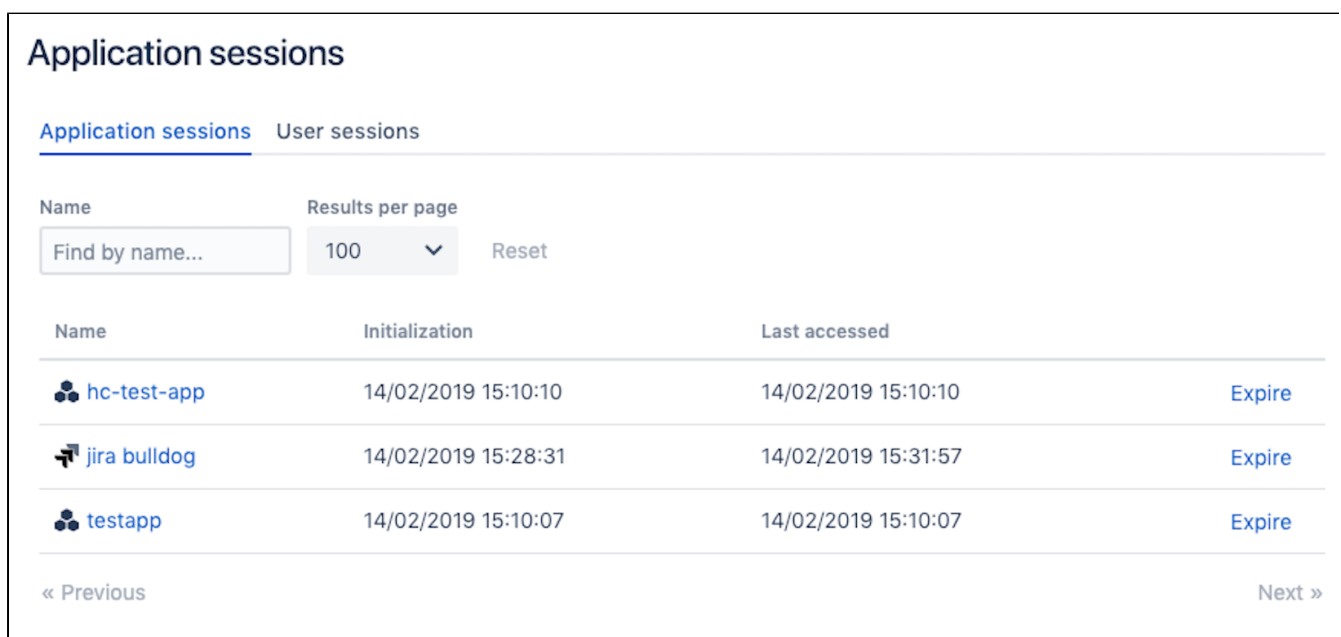
1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click **Administration**.
3. Click '**Current Sessions**' in the left-hand menu.
4. This will display the '**Application Sessions**' screen, showing a list of all applications which are currently logged in to the Crowd framework. For example, the screenshot below shows that the **crowd** application (i.e. the [Crowd Administration Console](#)) is currently logged in to the Crowd framework.
 You can refine your search by specifying an application's '**Name**'. (Note that this is case sensitive.)




To force an application to log out of Crowd:

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click .
3. In the left-hand menu, click **Current Sessions**.
You can see a list of all applications which are currently logged in to the Crowd framework.
4. Next to the application you want to log out of Crowd, click **Expire**.

 If you want to *permanently* prevent an application from logging in to Crowd, please see [Deleting or Deactivating an Application](#).

Screenshot: 'Sessions Applications'



Name	Initialization	Last accessed	
 hc-test-app	14/02/2019 15:10:10	14/02/2019 15:10:10	Expire
 jira bulldog	14/02/2019 15:28:31	14/02/2019 15:31:57	Expire
 testapp	14/02/2019 15:10:07	14/02/2019 15:10:07	Expire

« Previous Next »


Deleting or Deactivating an Application

Deactivating an application prevents users from logging in to the application. You might do this if you are making changes to an application and need to temporarily keep users out of it.

Deleting an application removes the application's [details](#) and its [directory mappings](#). You would typically only do this if the application is no longer required.

To deactivate an application,

1. Log in to the [Crowd Administration Console](#).
2. Click the '**Applications**' tab in the top navigation bar.
3. This will display the [Application Browser](#). Click the '**View**' link that corresponds to the application you wish to deactivate.
4. This will display the '**Application Details**' screen. Deselect the '**Active**' check-box, then click the '**Update**' button. No users will now be able to log in to the application.

 To reactivate the application, follow the same steps but *select* the '**Active**' check-box.

To delete an application,

1. Log in to the [Crowd Administration Console](#).
2. Click the '**Applications**' tab in the top navigation bar.
3. This will display the [Application Browser](#). Click the '**View**' link that corresponds to the application you wish to deactivate.
4. This will display the '**Application Details**' screen. Click '**Remove Application**' in the left-hand menu.

The application will be removed from Crowd and will no longer appear in the [Application Browser](#).



You cannot delete or deactivate the '**crowd**' application (i.e. the [Crowd Administration Console](#)).

Screenshot: 'Deleting or Deactivating an Application'

The screenshot shows the 'Applications' management page in the Crowd interface. The top navigation bar includes 'Applications', 'Users', 'Groups', 'Roles', 'Directories', and 'Administration'. The 'Applications' tab is active, and the application 'demo' is selected. The left sidebar contains 'Search Applications', 'Add Application', and 'Remove Application'. The main content area shows the 'demo' application details with tabs for 'Details', 'Directories', 'Groups', 'Permissions', 'Remote Addresses', and 'Config Test'. The 'Details' tab is active, displaying the following fields:

- Name:** demo (The unique name that the application will use to authenticate against the Crowd framework as a client.)
- Description:** Crowd demo applicatio (A short description of the application. Often a web URL is helpful.)
- Active:**
- Conception:** 05 Feb 2008, 08:43:41
- Last Modified:** 29 Feb 2008, 11:46:36
- Password:** (To set a new password, enter the password and confirm. Leave blank to leave the password unchanged)
- Confirm Password:**

At the bottom of the form are 'Update »' and 'Cancel' buttons.

RELATED TOPICS

- [Using the Application Browser](#)
- [Adding an Application](#)
 - [Integrating Crowd with Atlassian Bamboo](#)
 - [Integrating Crowd with Atlassian Confluence](#)
 - [Integrating Crowd with Atlassian Confluence 3.4 or earlier](#)
 - [Updating Files in a Confluence Evaluation Distribution](#)
 - [Integrating Crowd with Atlassian CrowdID](#)
 - [Integrating Crowd with Atlassian Crucible](#)
 - [Integrating Crowd with Atlassian FishEye](#)
 - [Configuring FishEye earlier than 4.0 with Crowd](#)
 - [Integrating Crowd with Atlassian Jira](#)
 - [Integrating Crowd with Atlassian Jira 4.2 or earlier](#)
 - [Integrating Crowd with Atlassian Bitbucket Server](#)
 - [Integrating Crowd with Acegi Security](#)
 - [Integrating AppFuse - a Crowd-Acegi Integration Tutorial](#)
 - [Integrating Crowd with Jive Forums](#)
 - [Jive SSO](#)
 - [Integrating Crowd with Spring Security](#)
 - [Integrating AppFuse - a Crowd-Spring Security Integration Tutorial](#)
 - [Integrating Crowd with a Custom Application](#)
 - [Integrating Crowd with Atlassian HipChat](#)
- [Configuring the Google Apps Connector](#)
- [Mapping a Directory to an Application](#)
 - [Specifying the Directory Order for an Application](#)
 - [Specifying an Application's Directory Permissions](#)
 - [Example of Directory Permissions](#)
 - [Viewing Users in Directories Mapped to an Application](#)
 - [Specifying which Groups can access an Application](#)
 - [Syncing users based on their access rights](#)
- [Effective memberships with multiple directories](#)
- [Specifying an Application's Address or Hostname](#)

- [Testing a User's Login to an Application](#)
- [Enforcing Lower-Case Usernames and Groups for an Application](#)
- [Managing an Application's Session](#)
- [Deleting or Deactivating an Application](#)
- [Configuring Caching for an Application](#)
- [Overview of SSO](#)
- [Configuring Options for an Application](#)
- [Enabling OpenID client app](#)
- [Allowing applications to create user tokens](#)
- [Disabling the OpenID client app](#)
- [Configuring how users log in](#)

[Crowd documentation](#)

Configuring Caching for an Application

Caching is used to store run-time authentication and authorization rules, which can be expensive to calculate.

This page describes the cache that can be configured in each of the Crowd-connected applications, such as Jira, Confluence and Bamboo. For an overview of the other types of caching offered by Crowd, please refer to [Overview of Caching](#).

i Crowd application caching is also referred to as 'client caching'.

Explanation of Crowd Application Caching

Crowd-integrated applications can store user, group and role data in a local cache. This helps improve the performance of Crowd since these applications do not have to repeatedly request information from Crowd. Generally, it is not necessary to configure application caching, although this depends on the size of your application deployments.

Enabling Application Caching

To enable application caching:

- Edit the `crowd-ehcache.xml` file, which is located in the `WEB-INF/classes` directory of your application's [Crowd client](#). The two main properties are:
 - **diskStore:** If you have enabled disk persistence (`diskPersistent="true"`) this is the location on the file system where Ehcache will store its caching information. By default it uses `java.io.tmpdir` which is Java's default temporary file location.
 - **defaultCache:** This property has *many* configurable options. Please read the [documentation provided by Ehcache](#) to fully understand the implications and possibilities with this property. Some basic features are described below.

i Some applications may enable/disable caching based on the Crowd server setting

The Crowd API allows an application to query whether caching is enabled on the Crowd server (`isCacheEnabled`). The Crowd Java client does not make use of this API feature, because it makes more sense to have application caching configured entirely on the application side. If you have a Crowd-integrated custom application which does make use of this API call, then the setting on the [Crowd server](#) will affect your application-side caching as well.

Extract from the ehcache.xml file

Below is a small snippet of the `crowd-ehcache.xml` file.

```
<ehcache>

  <diskStore path="java.io.tmpdir"/>

  <defaultCache
    maxElementsInMemory="4096"
    eternal="false"
    overflowToDisk="false"
    timeToIdleSeconds="300"
    timeToLiveSeconds="300"
    diskPersistent="false"
    diskExpiryThreadIntervalSeconds="120"/>

</ehcache>
```

Basic Cache Attributes

- **eternal:** This indicates that all elements in the cache will live for ever and that any time-outs will be ignored. It is strongly recommended that you set this to false.
- **timeToldleSeconds:** This sets the maximum amount of time between an element being accessed and its expiry. If you set this value to 0, the element will idle indefinitely.
- **timeToLiveSeconds:** This sets the maximum time between creation time of an element and its expiry. If you set this value to 0 it will live indefinitely.
- **maxElementsInMemory:** Sets the maximum number of elements that can be stored in the cache's memory. If this limit is reached, the default caching strategy **LRU** (*Least Recently Used*) will be invoked and those elements will be removed.

An element is anything stored in Crowd's cache: a user, a group, a list of users, a list of groups, a list of user memberships, a list of group memberships.

✔ Hint: If you want to store everything in memory, try this value to start with:
 (Number of users x 2) + (number of groups x 2)

Important Client Caches

The default **maxElementsInMemory** value of 4096 should be sufficient for most Crowd-integrated applications. However, for larger installations please ensure that the **maxElementsInMemory** matches the recommended size calculation listed below:

Name of Cache:	Size Calculation:
com.atlassian.crowd.integration-user	The number of users in your system.
com.atlassian.crowd.integration-group	The number of groups in your system.
com.atlassian.crowd.integration-parentgroup	The number of groups in your system.
com.atlassian.crowd.integration-group-membership	The number of users multiplied by the number of groups (<i>users * groups</i>). This total could be quite large, so you can optimize it by setting it to the number of users that are likely to be active at any one time. The algorithm will fall back to using the <code>com.atlassian.crowd.integration-all-group-members</code> cache (see below) before hitting the server to check.
com.atlassian.crowd.integration-all-memberships	The number of users in your system.
com.atlassian.crowd.integration-all-group-members	The number of groups in your system.

Overview of SSO

Crowd provides single sign-on (SSO) across a number of applications. This means that users can log in just once, then access the applications without having to log in to each one individually.

In Crowd Server the SSO functionality is available for applications within a single domain, such as Jira, Confluence and others. [Crowd Data Center](#) offers SSO 2.0 that is multi-domain.

Looking for a cross-domain SSO solution?

Look no more! Crowd SSO 2.0 offers one solution for Server, Data Center, and Cloud applications and setting it up takes only minutes.

Are you are ready for the change? See [Crowd SSO 2.0](#).

In Crowd Server single domain SSO you can also extend SSO to beyond-the-firewall applications using CrowdID for OpenID and Crowd's Google Apps connector.

This page gives an overview of Crowd's Server single domain SSO capabilities, plus links to detailed information on configuring Crowd and the applications concerned.

If you are looking for multi-domain SSO please go to [Crowd SSO 2.0](#)

SSO within a Single Domain

The core Crowd functionality supports SSO across applications within a single domain, such as `*.mydomain.com`. Crowd uses a browser cookie to manage SSO. Because your browser limits cookie access to hosts in the same domain, this means that all applications participating in SSO must be in the same domain.

Example 1: If you wish to have single sign-on (SSO) support for `*.mydomain.com`, you will need to configure the SSO domain in Crowd as `mydomain.com`. All your Crowd-connected applications must be in the same domain. For example:

Crowd	<code>crowd.mydomain.com</code>	✓
Jira	<code>jira.mydomain.com</code>	✓
Confluence	<code>confluence.mydomain.com</code>	✓
FishEye	<code>fisheye.mydomain.com</code>	✓
FishEye in different domain	<code>fisheye.example.com</code>	✗

Example 2: If you wish to have single sign-on (SSO) support for `mydomain.com/*`, you will need to configure the SSO domain in Crowd as `mydomain.com`. All your Crowd-connected applications must be in the same domain. For example:

Crowd	<code>mydomain.com/crowd</code>	✓
Jira	<code>mydomain.com/jira</code>	✓
Confluence	<code>mydomain.com/confluence</code>	✓
FishEye	<code>mydomain.com/fisheye</code>	✓
FishEye in different domain	<code>example.com/fisheye</code>	✗

You can find information the comparison of host name strings in [RFC 6265](#) (section 5).

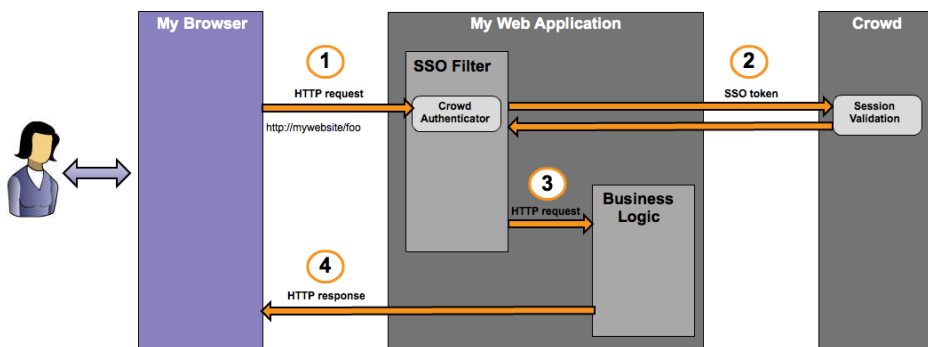
You can configure the SSO domain via the Crowd Administration Console, as described in the [documentation](#).

How It Works

The diagram below gives a conceptual overview of an HTTP request passing through an SSO filter and moving directly through the application business logic to create the response. (Click the link below the diagram to see a larger version.)

The diagram shows the 'happy path' only, assuming that:

- The user has already logged in to an application that is configured to participate in SSO. If the user has already logged in to one application, they will not need to log in again when accessing another application in the same domain.
- The request passes all authentication and authorization checks.



The diagram illustrates the following steps:

- **Step 1:** The HTTP request with an SSO cookie.
 - a. The user has already logged in to an application that is part of the SSO environment.
 - b. The user accesses a new application within the SSO environment, or performs some other action on the website.
 - c. The browser creates an HTTP request, bundles all the cookies for the domain and sends the request to the web application. This includes the SSO cookie, since the user has already logged in.
 - d. The request is trapped by the SSO filter in the web application's security framework. This filter may be provided by [Atlassian Seraph](#), by [Spring Security](#), by another framework or via custom code.
 - e. (If the user has not logged in, the filter re-directs the user to the login screen at this point. But we're assuming the user has logged in.)
 - f. The Crowd authenticator finds the SSO cookie, extracts the SSO token and passes the token to Crowd. The Crowd authenticator is a plugin to the security framework ([Atlassian Seraph](#), [Spring Security](#), or others).
- **Step 2:** Validation of the SSO token.
 - a. Crowd validates the session token. If another application in the same domain has already authenticated the user, Crowd will validate the existing authentication.
 - b. If the session has expired, Crowd re-directs the user to the login screen and re-authenticates the user.
 - c. Crowd checks that the user is authorized to access the application.
 - d. If the user does not have the required permissions, Crowd re-directs the user to the login screen.
 - e. Once validation is successful, Crowd passes the validated token back to the application's SSO filter.
 - If the session is still valid, the user will not need to log in again even if accessing a different application. The authentication and authorization will be transparent to the user.
- **Step 3:** Processing of the HTTP request.
 - a. The application's SSO filter passes the request to the business logic handler. (In a Java application, this is the servlet.)
 - b. The business logic handler processes the request and builds the response.
- **Step 4:** The HTTP response.
 - a. The application sends the response back to the browser.

Here is an [overview of servlet filters](#) from Oracle and a useful [tutorial](#) from O'Reilly.

The SSO filter may be provided by a security framework or by custom code as follows:

Security Framework or Custom Code	Comments
Framework: Atlassian Seraph	Most of the Atlassian applications use Seraph. The Crowd documentation tells you how to integrate SSO into Confluence , Jira , Bamboo , etc. If you are integrating a custom application with Crowd, you may also decide to use Seraph as your security framework.
Framework: Spring Security	You may have a web application that uses the Spring Security framework and that you are now integrating with Crowd. The Crowd documentation tells you how to integrate SSO into a Spring Security-based application . A point of interest: Crowd uses the Spring Security framework, and so does the Crowd 'demo' application .
Framework: Acegi Security (old)	You may have a web application that uses the Acegi Security framework and that you are now integrating with Crowd. The Crowd documentation tells you how to integrate SSO into an Acegi-based application . Note that Acegi Security is an earlier version of Spring Security .
Custom authentication for Atlassian Fisheye and Crucible	Crowd provides a custom integration with Fisheye and/or Crucible, including SSO. See the Crowd documentation .
Crowd API for your custom application	When integrating your own web application with Crowd, you can use the Crowd API to implement SSO. <ul style="list-style-type: none"> We recommend that you use the Crowd REST APIs for long-term compatibility. If you have a Java application, you can use the Java Integration Libraries shipped with Crowd, but please be aware that they may change between releases. You may need to re-compile your source and possibly change a package name. There are a number of third-party language bindings and application connectors developed by Crowd users. You can see them in the Atlassian Marketplace.

Configuring Crowd for SSO

Below are the configuration settings which affect SSO:

Short Description	More Information
Set your SSO domain	Set the domain via the Crowd Administration Console, as described in the documentation .
Optional: Configure Trusted Proxy Servers	Configure Crowd to trust a proxy's IP address, if you are running applications behind one or more proxy servers. See the documentation .
Optional: Enforce a secure connection, such as SSL, for all SSO requests	You can specify that the 'secure' flag is set on the SSO cookie, as described in the documentation . ⚠ Unsecured connections will be rejected, including the Crowd Administration Console if not accessed via SSL.

Configuring the Applications for SSO

When integrating an application with Crowd, you will configure the application to use Crowd as a centralized authentication repository. For most applications, **but not all**, you can also choose to configure SSO. This is described in detail for each application:

- [Integrating Crowd with Atlassian Bamboo](#)
- [Integrating Crowd with Atlassian Confluence](#)
- [Integrating Crowd with Atlassian CrowdID](#)
- [Integrating Crowd with Atlassian Crucible](#)
- [Integrating Crowd with Atlassian FishEye](#)
- [Integrating Crowd with Atlassian Jira](#)
- [Integrating Crowd with Atlassian Bitbucket Server](#)
- [Integrating Crowd with Acegi Security](#)
- [Integrating Crowd with Jive Forums](#)
- [Integrating Crowd with Spring Security](#)
- [Integrating Crowd with a Custom Application](#)
- [Integrating Crowd with Atlassian HipChat](#)

Troubleshooting SSO

See [Troubleshooting SSO with Crowd](#).

SSO Beyond the Firewall

Crowd allows you to extend SSO to beyond-the-firewall applications using CrowdID and Crowd's Google Apps connector.

Using CrowdID as an OpenID Provider

Crowd allows you to host an OpenID provider, called CrowdID, so that your users have a single point of authentication for all OpenID-enabled websites. Refer to the [CrowdID Administration Guide](#) and [CrowdID User Guide](#).

[OpenID](#) is an open, free protocol which allows a user to have a single identifier for logging in to any OpenID-enabled website. The website will communicate with a specific OpenID provider (in this case, your CrowdID server) when attempting to verify the user's login. For example, if your team uses 37signals' CRM tool [Highrize](#), using Crowd's OpenID provider means you can get SSO between Highrize and your behind-the-firewall applications for all your team.

Using SSO with Google Apps

Crowd offers SSO with [Google Apps](#) via the [Google Apps connector](#) shipped with your Crowd installation. This means that your users can log in just once and then move between Google Apps and other applications like Jira, Confluence, etc.

Configuring Options for an Application

Once you have [added an application](#) to Crowd, you can configure various options for that application in the **Options** tab. Click the links below for information about each option:

- [Lower Case Output](#)
- [Enable Aliasing](#)

Screenshot: Application Options

Jira


Details Directories & groups Users Permissions Remote addresses Authentication test **Options** SSO

- Lower case output**
Convert all users and groups to lower case when passing the data to the application. This can be used to achieve case insensitivity for applications when the underlying directories contain mixed-cased data.
- Enable aliasing**
With aliasing enabled, users can have a different username (alias) for the "test jira" application. This is useful if "test jira" does not support user renaming, but may impact the performance of incremental directory synchronisation for this application. See the [documentation](#) for more information.

Update Cancel

Enabling OpenID client app

Starting from Crowd version 3.6.2, OpenID client application will be disabled from deployment by default. However, you can still reenable this app on your staging environment by following steps bellow.

 For security reasons OpenID client app shouldnt be deployed on production environment.

Note regarding DC versions: Following procedure is related to a single node. If youre running cluster with many nodes then to deploy OpenID client app on every of them you need to perform following procedure on each node.

1. Stop Tomcat container (if running) by executing following script:

```
{CROWD_INSTALL}/stop_crowd.sh
```

2. Create following file by running:

```
touch {CROWD_INSTALL}/apache-tomcat/conf/Catalina/localhost/openidclient.xml
```

and fill the newly created file with the following content:

```
<Context docBase="../../crowd-openidclient-webapp" />
```

3. Start Tomcat container by running following script:

```
{CROWD_INSTALL}/start_crowd.sh
```

4. OpenID client app should be now accessible on the following endpoint:


```
http://{CROWD_HOST}:{PORT_NUMBER}/openidclient
```

Allowing applications to create user tokens

All applications connected to Crowd can generate Crowd tokens for any user that can authenticate into that application. This can be useful, for example, for the remember me functionality as the app will not have to ask for credentials upon every login. For security reasons, by default, applications connected to Crowd are not allowed to create user tokens.

To allow applications to create such tokens:

1. In Crowd, go to **Applications** > *<your_application_name>* **Options**.
2. Check **Allow to generate user tokens**.

 There is a possibility for applications connected to Crowd to generate Crowd tokens for users without passing their passwords in a request.

Such token can later be used to impersonate user in other SSO version 1 applications if they have similar directory setup.

User tokens can be used to impersonate user in Crowd web application if Crowd application has similar directory setup.

For this reason, it is important to connect only trusted applications to Crowd. Additionally, it's recommended that you keep the **Allow to generate user tokens** setting disabled unless your application and setup clearly requires this setting to be turned on.

Disabling the OpenID client app

OpenID client is a testing app used as a starting point to develop OpenID-enabled Java applications. Its bundled with standard Crowd distribution and it is deployed by default. However, its not required for Crowd or OpenID server to work correctly. OpenID client/server should not be confused with OpenID Connect.

In version 3.6.2, 3.7.1, 4.0.0 Crowd had the OpenID client app disabled from deploying by default. To prevent the OpenID client app from deploying on versions of Crowd before the mentioned above, follow the procedure below.

This procedure should also be put as part of your upgrade procedure in case you're upgrading to any version of Crowd other than 3.6.2, 3.7.1, 4.0.0 or later.

Note regarding DC versions: If you're running a cluster with more than one node you should perform following procedure on each node.

Before you begin


Check if the OpenID client app is running in your Crowd Tomcat container. With Crowd running, go to:

```
http://{CROWD_HOST}:{PORT_NUMBER}/openidclient
```

If the result is *HTTP Status 404 - Not Found*, it means the OpenID Connect client app is not running in the specified environment.

If you see the OpenID Client app page, proceed with the following steps to disable it.

To disable the OpenID client app

 The following procedure will require scheduling a downtime of Crowd.

1. Stop Tomcat container by running following script:

```
{CROWD_INSTALL}/stop_crowd.sh
```

2. Backup and delete the following file which is responsible for deploying the OpenID client app. Backup of the file will allow you to reenable the app in the future. Remember that you shouldn't enable this application on production environment.

```
{CROWD_INSTALL}/apache-tomcat/conf/Catalina/localhost/openidclient.xml
```

3. Start Tomcat container by running following script:

```
{CROWD_INSTALL}/start_crowd.sh
```

4. Again, verify that the OpenID client app is disabled. Go to:

```
http://{CROWD_HOST}:{PORT_NUMBER}/openidclient
```

You should see *HTTP Status 404 - Not Found*, which means the app is not running anymore.

Configuring how users log in

Users can log in to Crowd using their logins or email addresses. Using email addresses has been introduced in Crowd 4.4 and enabled by default, but you can change these settings.

To change how users log in:

1. Go to your Crowd administration console.
2. Select the Crowd application and switch to the **Options** tab.
3. Select whether users can log in with email addresses.

The screenshot shows the 'Options' tab for the 'crowd' application. The navigation menu includes: Details, Directories & groups, Administrators, Users, Permissions, Remote addresses, Authentication test, and Options. The settings are as follows:

- Lower case output: Convert all users and groups to lower case when passing the data to the application. This can be used to achieve case insensitivity for applications when the underlying directories contain mixed-cased data.
- Enable aliasing: With aliasing enabled, users can have a different username (alias) for the "crowd" application. This is useful if "crowd" does not support user renaming, but may impact the performance of incremental directory synchronization for this application. See the [documentation](#) for more information.
- Allow to generate user tokens: Allow this application to generate user tokens. For security reasons, it's recommended to disable this setting unless there are applications which require this setting to be enabled to work properly. To learn more, [see documentation](#).
- Authenticate with email address: Allow your users to authenticate with username or email address.

Buttons: Save, Cancel

Limitations

In some cases, Crowd is unable to verify that the email belongs to the user and won't allow them to log in by using it. To describe these cases, let's assume that Crowd is configured with two directories **Directory 1** and **Directory 2**, defined in that exact order.

Example 1: Ambiguous email ownership

If there are multiple Crowd users with the same email address, none of them will be able to log in.

Directory	Who can authenticate	Automatically assigned to	Actions
Directory 1	ALL GROUPS	NO GROUPS	...
Directory 2	ALL GROUPS	NO GROUPS	...

Example 2: Shadowed email owner

In this example, each of your two directories has a user called John. John from Directory 2 uses an email address *john@john.com*, while John from Directory 1 uses a different email address. When *john@john.com* is used during authentication, the user won't be able to log in. This is because John 1 is the canonical user in Crowd and he doesn't use this address. Check out [Effective memberships with multiple directories](#) to learn more about canonical and shadowed users.

Example 3: Overlapping emails and usernames

In this example, we have two Crowd users with the same email address. But, for *john one*, the email address is actually a username.

john two	john	john@john.com
john one	john@john.com	anotherjohn@john.c

Crowd first checks the provided value against usernames and only then emails. In this case, Crowd will attempt to log in John 1 as the provided value is his username.

Managing Users and Groups

This section describes how to add and edit users and groups using the [Crowd Administration Console](#). Note that the ability to do this depends on the [permissions](#) of the directory which contains the users and groups.

Managing Users and Groups

- [Using the User Browser](#)
- [Adding a User](#)
- [Editing a User's Details and Password](#)
- [Deleting or Deactivating a User](#)
- [Case Sensitivity of Usernames and Groups](#)
- [Specifying a User's Aliases](#)
- [Editing a User's Group Membership](#)
- [Managing Groups](#)
 - [Deleting a Group](#)
 - [Adding a Group](#)
- [Managing Group Members](#)
 - [Automatically Assigning Users to Groups](#)
 - [Adding Users to a Group](#)
 - [Removing Users from a Group](#)
 - [Nested Groups in Crowd](#)
 - [Adding a Sub-Group](#)
 - [Group-level administration](#)
 - [Adding Group Level Admins](#)
 - [Removing Group Level Admins](#)
 - [Removing a Sub-Group](#)
- [Specifying a User's Attributes](#)
- [Granting Crowd Administration Rights to a User](#)
- [Granting Crowd User Rights to a User](#)
- [Managing a User's Session](#)

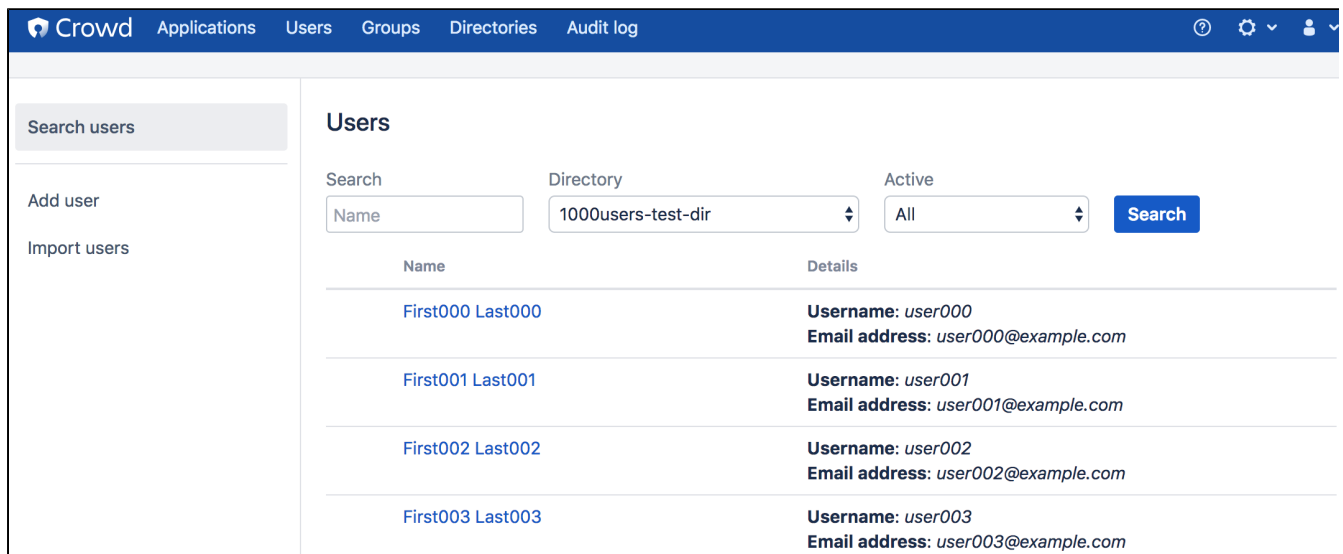
Using the User Browser

The User Browser allows you to search, view, add, and edit users within a specified directory.

To use the User Browser:

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click **Users**.
The User Browser appears.
3. In the **Search** textbox, enter your search criteria.
You can enter all or part of the user's name, email address or username. Leave the search box empty to retrieve all users.
You can refine your search by choosing **Active** or **Inactive** users. (An **Inactive** user is typically someone who has left your organization.)
4. Click **Search**.
Crowd will list all the users in the selected directory who match your search criteria.
 - To [view or edit a user's details](#), click the link on the user's name.
 - A maximum of 100 users will appear on a page.
 - If there are more than 100 users that match the search, the **Next** and **Previous** links will appear at the bottom of the page, so that you can move from one page to the next.

Screenshot: 'User Browser'



The screenshot displays the 'Users' section of the Crowd Administration Console. The top navigation bar includes 'Crowd', 'Applications', 'Users', 'Groups', 'Directories', and 'Audit log'. On the left sidebar, there are options for 'Search users', 'Add user', and 'Import users'. The main content area is titled 'Users' and features search filters: a 'Search' field with 'Name' selected, a 'Directory' dropdown set to '1000Users-test-dir', and an 'Active' dropdown set to 'All'. A blue 'Search' button is positioned to the right of these filters. Below the filters, a table lists four users with their names and details.

Name	Details
First000 Last000	Username: user000 Email address: user000@example.com
First001 Last001	Username: user001 Email address: user001@example.com
First002 Last002	Username: user002 Email address: user002@example.com
First003 Last003	Username: user003 Email address: user003@example.com

Adding a User

To add users in Crowd, you can either import users into Crowd in bulk (see [Importing Users and Groups into a Directory](#)), or add them individually as described below.

To add a user:

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click **Users**.
3. In the left-hand menu, click **Add user**.
4. Fill in user details.

Email The email address of the user. Email addresses must follow the RFC2822 format.

Active Only deselect this if you wish to deny the user access to the Crowd-integrated applications.

Username The user's login name. Within a given directory, the username must be unique. Note that you cannot change the username once the user has been created.

Password The user's password.

Confirm Password Enter the same password again, to ensure that you have typed it correctly.

First Name The user's first name.

Last Name The user's last name.

Directory The directory to which the user will be added. Note that the user cannot be moved to a different directory once the user has been created.

5. Click **Create** to add the user.

Next steps

Once you've added a user to Crowd, you are able to specify their [attributes](#) and [group membership](#). If you wish, you can also [verify that the user can log in to](#) appropriate applications.

Automatically adding users to Jira or other groups

You can configure your directory to automatically add users to one or more groups. Define the default groups on the directory as described in [Automatically Assigning New Users to Groups](#). For example, you can add Jira groups as default groups for your LDAP directory connector. Whenever a new user is added to LDAP, they will automatically get access to Jira.

Editing a User's Details and Password

Crowd administrators can edit a user's details, rename users, mark a user as active or inactive, and change or reset a user's password.

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click **Users**.
3. Select the relevant directory, search for the user you want to update, and click the user's name.
4. Edit the details as required.
5. Click **Update**.

Note: Not all directories support user rename. Crowd will inform you in the event where a user cannot be renamed

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click **Users**.
3. Select the relevant directory, search for the user you want to update, and click the user's name.
4. Edit the **Username**.
5. Click **Update**.

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click **Users**.
3. Select the relevant directory, search for the user you want to update, and click the user's name.
4. You can *either*:
 - Click **Reset Password** in the left-hand menu. Crowd will generate a random, unique URL and email it to the user. The user can then click the link and choose their own new password.
OR
 - Enter a new password then click **Update**. Crowd will *not* email the user in this case.



- You will need to configure an [email server](#) so that Crowd can send the user an email notification when you reset their password.
- You can edit the [email notification template](#) to determine the content of the email sent to the user.
- Users can update their own profiles. Authorized Crowd users can log in to the Self Service Console and update their own user profiles, as described in the [User Guide](#).

Deleting or Deactivating a User

Deactivating a user prevents the user from logging in to any [applications](#) that use the [Crowd framework](#) and also excludes the user from the license count. You would typically do this when a user leaves your organization.

Deleting a user removes the user completely from the relevant [directory](#).

Deactivating instead of Deleting

We recommend that you deactivate a user rather than delete them, in case some applications contain historical data, such as documents that the user has created. Read [more](#).

For Microsoft Active Directory servers, LDAP Connector Directories and Delegated Authentication Directories will synchronize the status of users with the remote server, if the **Synchronize User Details** option has been enabled. In other words, if a user account is disabled in Active Directory, it will be deactivated in Crowd on the next synchronization. Likewise, if a user is deactivated through Crowd, the user account will be disabled in Active Directory. If you want to prevent this synchronization, enable the **Manage User Status Locally** option in the directory configuration.



Deactivating a user that resides in LDAP

For applications that need users to exist for historical data (such as Jira), you should recreate the user and mark it inactive in a Crowd Internal Directory before deleting from your LDAP directory.

To deactivate a user:

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click **Users**.
3. Select the relevant directory, search for the user you wish to deactivate, and click the user's name. The **User Details** screen appears.
4. Deselect the **Active** checkbox.
5. Click **Update**.

The user will now be unable to log in to any applications that use the Crowd framework.

To delete a user:

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click **Users**. Select the relevant directory, search for the user you wish to deactivate, and click the user's name. The **User Details** screen appears.
3. In the left-hand menu, click **Remove User**.
4. Confirm the deletion when prompted.

The user will be removed from the relevant directory and will no longer appear in the [User Browser](#).

Case Sensitivity of Usernames and Groups

This page summarizes the way Crowd handles case sensitivity for usernames and group names when storing, matching and searching data and when passing data between directories and applications.

Terminology:

- **Case insensitive** Upper-case and lower-case letters are assumed to have the same meaning: `JSmith` is the same as `jsmith`.
- **Case preserving** Upper and lower case are retained when passing or storing information: `JSmith` remains `JSmith`.

Outside Crowd

External to Crowd:

- Most LDAP directory schemas specify the user and group names as case insensitive for matching and searching, but case preserving when storing the data and passing it back to the requestor.
- Applications behave in different ways. Some, like [Jira](#) and [Confluence](#), insist on lower-case usernames and groups and store all user-related data in lower case.

The Crowd Solution

Crowd's application caches and LDAP directory caches are case insensitive but case preserving. Crowd will ignore case when comparing usernames, etc (`JSmith = jsmith`) and it will preserve case when passing information between applications and directories (`JSmith` remains `JSmith`).

In addition, Crowd [Internal](#) and [Delegated Authentication](#) directories:

- Are case preserving, i.e. they store usernames and group names in mixed case.
- Support case-insensitive matching and searching.

Importing Users and Groups into Crowd Internal Directories

When you import user information into a Crowd [Internal](#) or [Delegated Authentication](#) directory, the case of usernames and group names will be preserved.

Enforcing Lower-Case Usernames and Groups for an Application

In some cases you may wish to convert user and group names to lower case when passing them to an application. You can set an option for each application, as described in [Enforcing Lower-Case Usernames and Groups for an Application](#). When the option is set, Crowd will convert upper-case and mixed-case information obtained from your user directory to lower case before passing the information to the application.

Specifying a User's Aliases

A single user can have different usernames in different applications. These different usernames are called 'aliases'. As a Crowd administrator, you can manage each user's aliases for the applications the user is authorized to access.

Enabling User Aliasing for an Application:

You can choose to enable or disable aliasing for each application. By default, user aliasing is disabled.

i User aliasing can reduce the performance of your user directory, especially on user searches.

1. Log in to the [Crowd Administration Console](#).
2. Click the '**Applications**' tab in the top navigation bar.
3. The [Application Browser](#) will appear. Click the link on the name of the application you wish to configure.
4. The '**View Application**' screen will appear. Click the '**Options**' tab.
5. Put a tick in the checkbox labeled '**Enable Aliasing**'.
6. Click the '**Update**' button.

Specifying a User's Aliases:

You can add and remove aliases via the user management screens in the Crowd Administration Console.

1. Log in to the [Crowd Administration Console](#).
 2. Click the '**Users**' link in the top navigation bar.
 3. This will display the [User Browser](#). Select the relevant directory, find the user in that you want to update, then click the link on the user's name.
 4. The '**User Details**' screen will appear. Click the '**Applications**' tab.
- **To add an alias for the user,**
 1. Scroll down until you find the application to which the alias applies.
 - ✔ For example, if the user's primary username is 'adent' but he has a username of 'arthur' in Confluence, then you need to find your Confluence application.
 2. Type the value of the new alias (e.g. 'arthur') into the '**Alias**' field next to the application.
 3. Click the '**Update**' button.
 - **To edit an existing alias,** update the corresponding field in the '**Alias**' column, then click the '**Update**' button.
 - **To remove an alias,** click the corresponding '**Remove Alias**' link in the '**Action**' column.

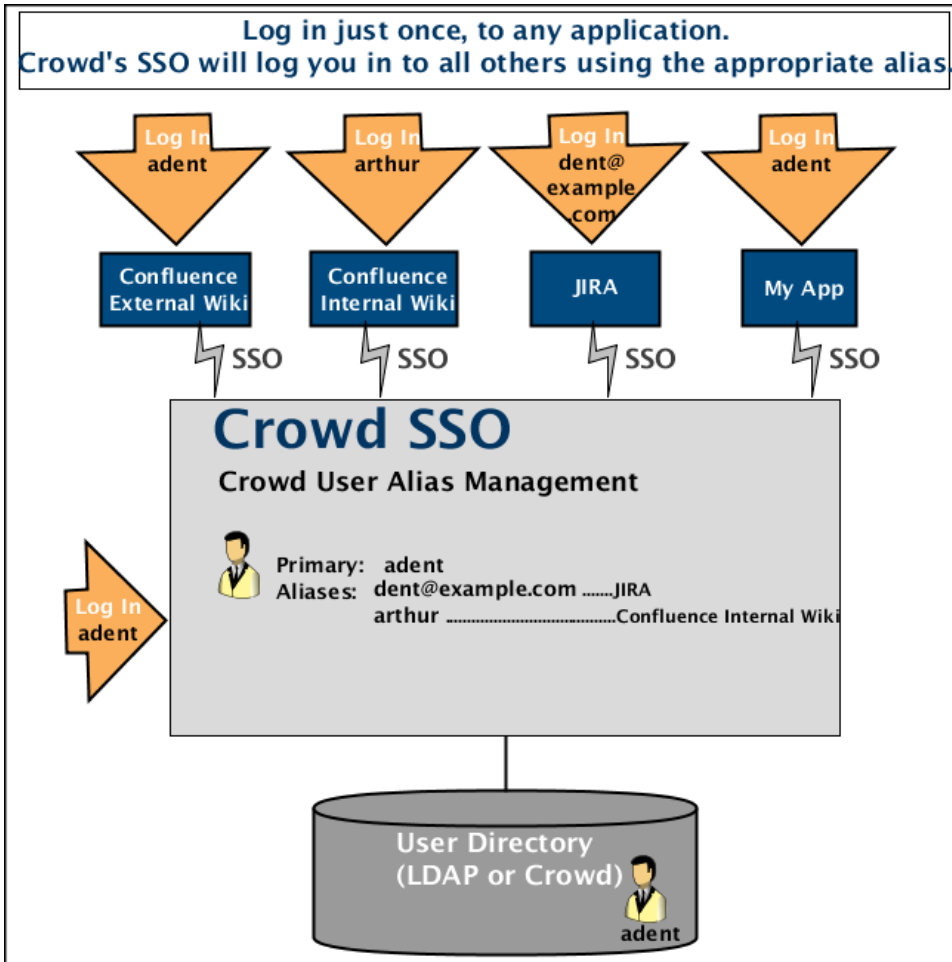
Examples and Use Cases

An example: Arthur Dent might have username 'dent@example.com' in your [Jira](#) issue tracker, 'arthur' in your internal [Confluence](#) wiki and 'adent' in your public-facing [Confluence](#) wiki.

- Using Crowd, you can link a number of usernames as aliases of Arthur's primary login ID.
- Arthur can log in just once, to any Crowd-connected application. He will be automatically logged into the other applications via single sign-on (SSO).
- When logging in to a specific application (e.g. Confluence), Arthur must use the specific username (alias) for that application, e.g. 'arthur'.
- When logging in to Crowd, Arthur must use his primary login i.e. the one in the directory, e.g. 'adent'.

Here are some cases where Crowd's user aliasing may be useful:

- Aliasing allows you to work around the problem that occurs when you want to implement a single user base for a number of existing systems, where users may have different usernames in each system.
- When someone gets married or changes their name, you may wish to rename a user in your LDAP directory, such as Microsoft Active Directory. To avoid problems in applications which do not allow user renaming, you can now link the new LDAP username to an alias in Crowd.
- Some systems may use email addresses as usernames, while in others this may expose users to email spambots. Using Crowd aliasing, you can use different username formats to suit your application requirements.



Editing a User's Group Membership

Within any given [directory](#), you can choose the groups to which each user belongs.

When you add a user to a group, that user will be authorized to use any applications that [use the group to control access](#).

Furthermore, that group membership may be used to determine authorization permissions see [Effective memberships with multiple directories](#).

Groups

The Crowd Administration Console provides two ways of adding users to, or removing them from, a group:

- The group management screen for a specific group here you can add many users at once to the selected group.
- The user management screen for a specific user here you can add a user to one or more groups at a time.

Full instructions are in [Adding Users to a Group](#) and [Removing Users from a Group](#).

Managing Groups

This page introduces you to groups in Crowd.

About Groups

Groups are known as *permission container objects*. Groups are particularly important in Crowd, as they are often used to [control access](#) to applications. Note also that the [crowd-administrators](#) group confers Crowd administration rights to its members. Support for roles, [previously deprecated](#), **has been removed** in Crowd 2.5. The implementation of roles in Crowd was identical to the implementation of groups and did not provide any extra functionality.

Nested Groups

Some user directories allow you to define a group as a member of another group. Groups in such a structure are called '**nested groups**'. In Crowd, you can [map any group to an application](#), including a group which contains other groups. Crowd supports nested groups for LDAP directory connectors, Crowd internal directories, Delegated Authentication directories and custom directories. You can **enable or disable** support for nested groups on each directory individually. For more information, refer to the documentation on [configuring a directory](#). For more details about nested groups, refer to [Nested Groups in Crowd](#).

About the Group Browser

The Group Browser allows you to search, view, add and edit the various groups stored within a specified directory.

To use the Group Browser:

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click **Groups**.
3. Select the directory in which you are interested, and click **Search** button to list all the groups that exist in that directory.
You can refine your search by specifying a '**Name**' or by choosing '**Active**' or '**Inactive**' groups.
4. To view or edit a group's details, click the link on the group name.
5. Click the **Direct Members** tab to view the immediate members of the group, including users and other groups.
6. Click the **Nested Members** tab to view all users who are included in the group and in its sub-groups

You can read more about group members in [Managing Group Members](#).

The screenshot shows the Crowd Administration Console interface. The top navigation bar includes 'Crowd', 'Applications', 'Users', 'Groups', 'Directories', and 'Audit log'. On the left side, there is a sidebar with 'Search groups', 'Add group', and 'Remove group' options. The main content area is titled 'View group - group000' and has three tabs: 'Details', 'Direct members', and 'Nested members'. The 'Details' tab is active, displaying the following information: Name: group000, Directory: Crowd user directory, Description: group000, and a checked 'Active' checkbox. At the bottom of the details section, there are 'Update' and 'Cancel' buttons.

Deleting a Group

Deleting a group removes it completely from the relevant [directory](#).

To delete a group:

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click the **Groups**.
3. Select the relevant directory, and locate the group you wish to deactivate.
4. Click the name of the group you want to delete.
5. In the left-hand side of the **Group Details** screen, click **Remove Group**.
6. Click **Continue**.


Adding a Group

Adding a Group via the Administration Console

To add a group:

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click **Groups**.
3. In the left-hand menu, click **Add Group**.
4. Complete the fields as described in the table below.

Name	The unique name of the group. Within a given directory, the Name must be unique. Note that the Name cannot be changed once the group is created.
Description	A short description of the group.
Directory	The directory to which the group will be added. Note that the group cannot be moved to a different directory after it is created.
Active	Only deselect this if you wish to deny access to all members of the group.

5. Click **Create**.
 You can now [add users](#) to the new group. If your directory supports [nested groups](#), you can now [add sub-groups](#).

Adding a local group with the same name as a group in remote directory

If you add a group to a directory which has the **Manage Groups Locally** option enabled, there's a possibility that Crowd will use a local group that you added instead of the group in the remote directory. This will happen if you the local group you're adding has the same name as a group on the remote server, and the remote group hasn't been synchronized yet.

However, if the remote group has already been synchronized, and exists in Crowds cache, when adding a local group of the same name, you will see an error: **Another group with this name already exists**.

Importing Groups from Other Applications

You can also add groups via Crowd's migration tools. See [Importing Users and Groups into a Directory](#).

Group Authorization

See [Specifying which Groups can access an Application](#).

Roles have been Removed

Support for roles, [previously deprecated](#), **has been removed** in Crowd 2.5. The implementation of roles in Crowd was identical to the implementation of groups and did not provide any extra functionality.

[Crowd documentation](#)

Managing Group Members

Groups are known as *permission container objects*. Groups are particularly important in Crowd, as they are often used to control [access to applications](#). Note also that the 'crowd-administrators' group confers [Crowd administration rights](#) to its members.

This page tells you how to view the members of a group in Crowd. The list of group members may take a while to load, depending upon the size of your user base.

Other things you can do from the group browser:

- [Add users to a group](#)
- [Remove users from a group](#)
- [Add sub-groups](#) (nested groups)
- [Remove sub-groups](#) (nested groups)

About nested groups

Some user directories allow you to define a group as a member of another group. Groups in such a structure are called '**nested groups**'. In Crowd, you can [map any group to an application](#), including a group which contains other groups. Crowd supports nested groups for LDAP directory connectors, Crowd internal directories, Delegated Authentication directories and custom directories. You can **enable or disable** support for nested groups on each directory individually. For more information, refer to the documentation on [configuring a directory](#). For more details about nested groups, refer to [Nested Groups in Crowd](#).

To view the members of a group:

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click **Groups**.
3. In the [Group Browser](#), select the directory and click the **Search** button to list all the groups that exist in that directory.
You can refine your search by specifying a **Name** or by choosing **Active** or **Inactive** groups.
4. Click the link on a specific group name to view the group's details.
5. The '**View Group Details**' screen will appear. Click the '**Direct Members**' tab to view the immediate members of the group, as shown in [screenshot 2 below](#).
 - If your user directory allows [nested groups](#), users and other groups may be members of the selected group. The 'Direct Members' tab shows all the immediate members of the group, including users and other groups.
 - If the group you are viewing does not contain other groups as members, the 'Direct Members' tab will show only users.
6. Click the '**Nested Members**' tab (if present) to view all users who are included in the group and in its sub-groups, as shown in [screenshot 3 below](#).

Screenshot 1: Group Browser

View group – group000

Details [Direct members](#) Nested members

Groups in this group

Group name	Description	Active
group001	group001	true
group002	group002	true
group003	group003	true
group004	group004	true
group005	group005	true

[Add groups](#) [Remove groups](#)

Adding users to groups and sub-groups

The 'Nested Members' tab does not allow you to add or remove members. To edit the membership of the group, please click the '**Direct Members**' tab. To edit the membership of a sub-group, click the '**Direct Members**' tab and then click the name of the sub-group to open the group maintenance screens for that group.

Automatically Assigning Users to Groups

For each directory in the application you may define a set of groups that will be automatically assigned to the user upon first successful login to this application using such directory.

If the directory is shared among different application in Crowd, you may define a different set of auto-assigned groups per application.

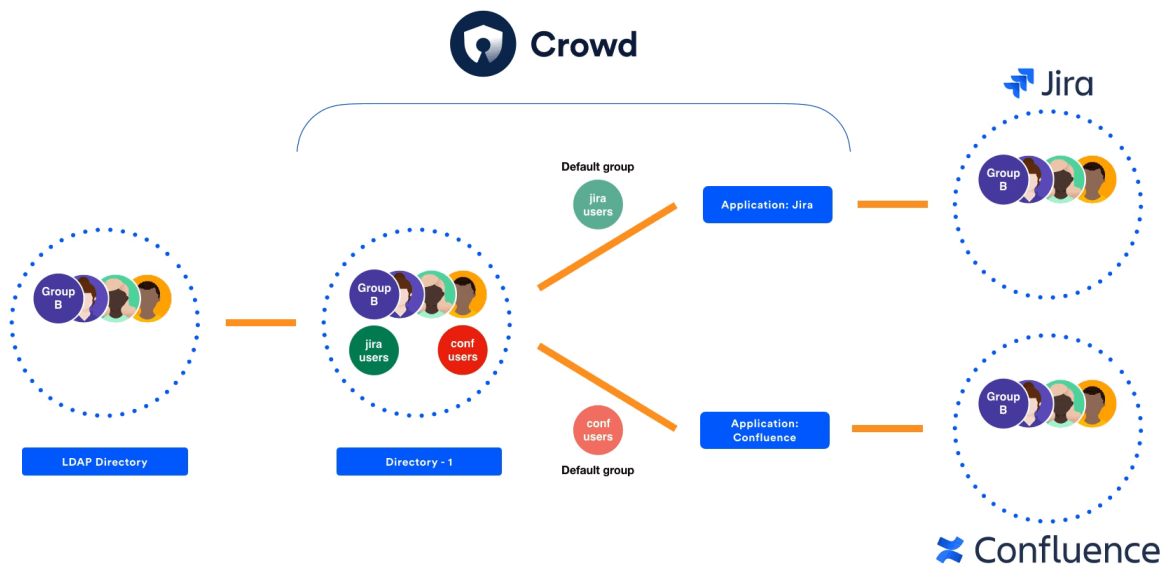
Example

Here's an example that will help you understand the concept of automatically assigned groups better:

There are two products, Jira and Confluence, which are mapped to the same remote directory using Crowd. In Crowd, there is only one definition of this remote directory.

The administrator has defined two local groups called `jira-users` and `conf-users` in the directory and mapped these groups to applications: `jira-users` to Jira and `conf-users` to Confluence.

Now, when users log in to Jira for the first time, they will be assigned to the `jira-users` group upon successfully authenticating with the remote directory.



Use case scenario

Whenever you have multiple applications connected to Crowd and some of these applications have different sets of users that have access to it, you might want to optimize your license usage so that only users actually using these products consume product licenses.

You could achieve that by manually assigning your users to groups per product, however, this solution does not scale with growing number of users.

Using automatically assigned groups per application allows you to optimize your products license usage, as only users that authenticated to an application would be assigned a license.

Defining automatically assigned groups per application

To add new automatically assigned groups per application:

i This instruction assumes that you have already defined directory mappings for your applications. If not, please consider the following [guide](#).

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click **Applications**.
3. Search for the application you wish to update, and click the link on the application name.
4. Select the **Directories & Groups** tab.
5. In the Actions menu, click **Configure automatically assigned groups**.
6. Search for your groups.
7. Select the groups by clicking on it or hitting Enter.
After selecting a group you may select another one by repeating steps 6 to 8
8. Click **Add** and then confirm your configuration by clicking **Save**.

To remove automatically assigned groups per application

1. In the **Automatically assigned groups** screen, find your group/
2. Click 'x' icon next to the group
3. Click **Save**.

Screenshot: Application directory mapping - configuring automatically assigned groups per application

The screenshot displays the 'your jira' application configuration page in the Crowd Administration Console. The 'Directories & Groups' tab is active, showing a table of directory mappings. The table has the following data:

Directory	Who can authenticate	Automatically assigned to	Actions
Crowd Internal	ALL	1 GROUP	...
My Active Directory	NO GROUPS	2 GROUPS	...
Company Open LDAP	NO ONE	NO GROUPS	... Configure authentication Configure automatically assigned groups Remove directory
Test Internal Directory		Add	

Below the table, the 'Directory aggregation' section is visible, with the option 'Aggregate group memberships across directories' selected.

Screenshot: Dialog for adding default groups per application

Assign default groups

×

When a user in My Active Directory directory authenticates successfully to your jira for the first time, they will be automatically added to the following groups:

Name

test-1 ×

test-2 ×

Add

test-3

Save

Cancel



- Groups will be automatically assigned to users only when they first log in to a product connected to Crowd.
- When you remove or add a group to the list, users who have already been added to a default group will not be added to the new groups, or removed from the old ones.

Configuring product

Once automatically assigned groups are defined for application in Crowd, make sure that the **Update group memberships when logging in** advanced option is set to **Every time the user logs in** for Crowd directory in the product represented by this application in Crowd.

This option can be set on the directory configuration screen. See the following documentation with instructions how to configure Crowd directory in products:

- [Integrating Crowd with Atlassian Jira](#)
- [Integrating Crowd with Atlassian Confluence](#)
- [Integrating Crowd with Atlassian Bitbucket Server](#)
- [Integrating Crowd with Atlassian FishEye](#)

Automatically assigning groups per directory

You can configure Crowd to assign new users to specific groups automatically. You can define default groups for each directory so that every user that logs in becomes a member of these groups automatically.

To add new default groups for a directory:

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click **Directories**.
3. Search for the directory you wish to update, and click the link on the directory name.
4. Select the '**Options**' tab.
5. Click the '**Add Groups**' button.
6. Search for your groups.
Crowd will list the groups in the selected directory that match your search criteria but excluding groups that are already defined as default groups for the selected directory.
 ⓘ Crowd will display a maximum number of groups as specified in the '**Maximum Results**' field. If too many groups match the search, you can change the search criteria and click 'Search' again. (There is no way to move to the next page of matching groups.)
7. Select your groups.
8. Click **Add Selected groups**.

To remove a group from the list of default groups for a directory:

1. In the **Options** tab, find the group on the list.
2. Click the 'x' button next to the group name.

ⓘ Once you have removed the group from the list, users will not be added automatically to the group when they log in. Existing users will remain members of the group.

Screenshot: Default groups for a directory

View directory - Atlassian Crowd

Details

Configuration

Permissions

Options

Default group memberships

When a user in this directory authenticates successfully for the first time, they will be automatically added to the following group:

- crowd-administrators ⓘ
- jira-administrators ⓘ
- jira-developers ⓘ

Screenshot: Popup for adding default groups

Add groups

Search

Active

Maximum results

<input type="checkbox"/>	Name	Description
<input checked="" type="checkbox"/>	crowd-administrators	Crowd administrators
<input type="checkbox"/>	jira-administrators	All Jira users
<input type="checkbox"/>	jira-developers	Developers

RELATED TOPICS

- [Managing Groups](#)
- [Managing Group Members](#)
- [Managing Directories](#)
- [Crowd documentation](#)

Adding Users to a Group

When you add a user to a group, that user will be authorized to use any applications that [use this group to control access](#).


Group membership may be used to determine authorization permissions see [Effective memberships with multiple directories](#).


You can add users to a group in two places:

- The group management screen for a specific group Here you can add **many users at once** to the selected group.

Using the group management screen for a specific group, you can add many users at once to the selected group.

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click **Groups**.
3. Locate the group you want to add users to and click the group name.
4. In the **Group Details** screen, click the **Direct member** tab.
This will display a list of the selected group's members, both the groups and the users that are direct members of the group.
5. Click **Add Users**.
6. Enter your search criteria in the **Search** textbox.

 You can enter all or part of the user's email address or username. Leave the search box empty to match all usernames and email addresses.
You can refine your search by choosing **Active** or **Inactive** users. (An 'Inactive' user is typically someone who has left your organization.)
You can also set the **Maximum Results**, i.e. the number of users to be retrieved.

7. Click **Search**.
Crowd will list the users in the selected directory who match your search criteria, but excluding users who are already members of the selected group.
 Crowd will display a maximum number of users as specified in the **Maximum results** field. If too many users match the search, you can change the search criteria and click **Search** again. (There is no way to move to the next page of matching users.)
 8. Select the users by putting a tick in the checkbox next to one or more users. To select all users, you can put a tick in the checkbox at the top of the table.
 9. Click the **Add selected users** button to add the selected users to the group.
- The user management screen for a specific user Here you can add the selected user to **one or more groups** at a time.

[Using the user management screen for a specific user, you can add the selected user to one or more groups at a time.](#)

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click **Users**.
3. Locate the user you wish to add, and click the link on the user's name.
4. In the **User Details** screen, click the **Groups** tab.
A list of the user's current groups (if any) will appear.
5. Click **Add groups**.
6. Enter your search criteria in the **Search** textbox.
Leave the search box empty to match all groups.
You can refine your search by choosing **Active** or **Inactive** groups.
You can also set the **Maximum Results**, i.e. the number of groups to be retrieved.
7. Click **Search**.
Crowd will list the groups in the selected directory that match your search criteria, but excluding groups that the user already belongs to.

- Crowd will display a maximum number of groups as specified in the **Maximum Results** field. If too many groups match the search, you can change the search criteria and click **Search** again. (There is no way to move to the next page of matching groups.)
- 8. Select the groups by putting a tick in the checkbox next to one or more groups. To select all groups, you can put a tick in the checkbox at the top of the table.
- 9. Click **Add selected groups** to add the user to the selected groups.

Removing Users from a Group

If you remove a user from a group, the user will no longer be able to log in to any applications that [use this group to control access](#).


Removing a user from a group does not delete the user from the directory. See [Deleting or Deactivating a User](#).


You can remove users from a group in two places:

- The group management screen for a specific group Here you can remove **many users at once** from the selected group.

Using the group management screen for a specific group, you can remove **many users at once** from the selected group.


1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click **Groups**.
3. Locate the group you want to add users to and click the group name.
4. In the **Group Details** screen, click the **Direct member** tab.
This will display a list of the selected group's members, both the groups and the users that are direct members of the group.
5. Click **Remove users**.
6. Enter your search criteria in the '**Search**' textbox.

 You can enter all or part of the user's email address or username. Leave the search box empty to match all usernames and email addresses. You can refine your search by choosing **Active** or **Inactive** users. (An 'Inactive' user is typically someone who has left your organization.) You can also set the **Maximum Results**, i.e. the number of users to be retrieved.

7. Click **Search**.
Crowd will list the users in the selected directory who match your search criteria and are members of the selected group.
 -  Crowd will display a maximum number of users as specified in the '**Maximum Results**' field. If too many users match the search, you can change the search criteria and click 'Search' again. (There is no way to move to the next page of matching users.)
8. Select the users by putting a tick in the checkbox next to one or more names. To select all users, you can put a tick in the checkbox at the top of the table.
9. Click the **Remove selected users** button to remove the selected users from the group.

- The user management screen for a specific user Here you can remove the selected user from **one or more groups** at a time.

Using the user management screen, you can remove a specific user from the groups that that user belongs to.

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click **Users**.
3. Locate the user you wish to remove, and click the link on the user's name.
4. In the **User details** screen, click the **Groups** tab.
A list of the user's current groups (if any) will appear.
5. Click **Remove groups**.
The **Remove groups** popup screen will appear.
6. Enter all or part of the group name in the '**Search**' textbox, or leave the search box empty to match all groups.
You can refine your search by choosing '**Active**' or '**Inactive**' groups.
You can also set the '**Maximum Results**', i.e. the number of groups to be retrieved.
7. Click **Search**.
Crowd will list the groups that the user belongs to, matching your search criteria in the selected directory.
 -  Crowd will display a maximum number of groups as specified in the **Maximum results** field. If too many groups match the search, you can change the search criteria and click 'Search' again. (There is no way to move to the next page of matching groups.)

8. Select the groups by putting a tick in the checkbox next to one or more groups. To select all groups, you can put a tick in the checkbox at the top of the table.
9. Click **Remove selected groups** to remove the user from the selected groups.

Nested Groups in Crowd

This page describes the way Crowd handles **nested groups**, i.e. groups that contain other groups or that belong to groups.

Summary of Nested Groups in Crowd

Some user directories allow you to define a group as a member of another group. Groups in such a structure are called '**nested groups**'. In Crowd, you can [map any group to an application](#), including a group which contains other groups. Crowd supports nested groups for LDAP directory connectors, Crowd internal directories, Delegated Authentication directories and custom directories. You can **enable or disable** support for nested groups on each directory individually. For more information, refer to the documentation on [configuring a directory](#). Here's the effect on authorization and presentation of group members to integrated applications:

- When verifying a user's login to an integrated application, Crowd will search the [mapped group](#) plus all its sub-groups.
- When an [integrated application](#) requests a list of users, Crowd will present a flat list of users gathered from the requested group and its sub-groups.

The rest of this page describes the above functionality in more detail.

In addition, you can follow the instructions to:

- [Add a sub-group \(nested group\)](#)
- [Remove a sub-group \(nested group\)](#)

Definition of Nested Groups

A 'nested group' is a group which is a member of another group. If you are using groups to manage permissions, you can create nested groups to allow inheritance of permissions from one group to its sub-groups.

In an LDAP directory, a nested group is defined as a child group entry whose DN (Distinguished Name) is referenced by an attribute contained within a parent group entry.

✓ For example, a parent group '**Group One**' might have an `objectClass=group` attribute and one or more `member=DN` attributes, where the DN can be that of a user *or* that of a group elsewhere in the LDAP tree:

```
member=CN=John Smith,OU=Users,OU=OrgUnitA,DC=sub,DC=domain
member=CN=Group Two,OU=OrgUnitBGroups,OU=OrgUnitB,DC=sub,DC=domain
```

Supported Directory Types

Crowd supports nested groups for the following directory types:

- [LDAP directory connectors](#)
- [Internal directories](#)
- [Delegated Authentication directories](#)
- [Custom directories](#), provided that the customization meets the interface requirements of the `RemoteDirectory` API.

The [directory importer](#) does **not** support nested groups when importing users, groups and roles from LDAP into a [delegated authentication](#) directory. See [CWD-1334](#).

Group Management via the Crowd Administration Console

The Crowd administrator can [view group memberships](#), [add](#) a group as a member of another group, and [remove](#) a group's membership of another group.

Verifying a User's Access to an Application

When verifying a user's login to an [integrated application](#), Crowd will search the groups [mapped to the application](#), plus all their sub-groups. If the username exists in one of the groups, Crowd will allow the user access to the application.

Presenting Flattened Lists of Users to Integrated Applications

[Integrated applications](#) may ask Crowd for a list of members in a group. Crowd will present all users who are members of the group and all users belonging its sub-groups, consolidated into one list. We call this list a 'flattened' group. This is necessary because many integrated applications do not understand the concept of nested groups. For that reason, Crowd makes the nesting transparent to integrated applications.

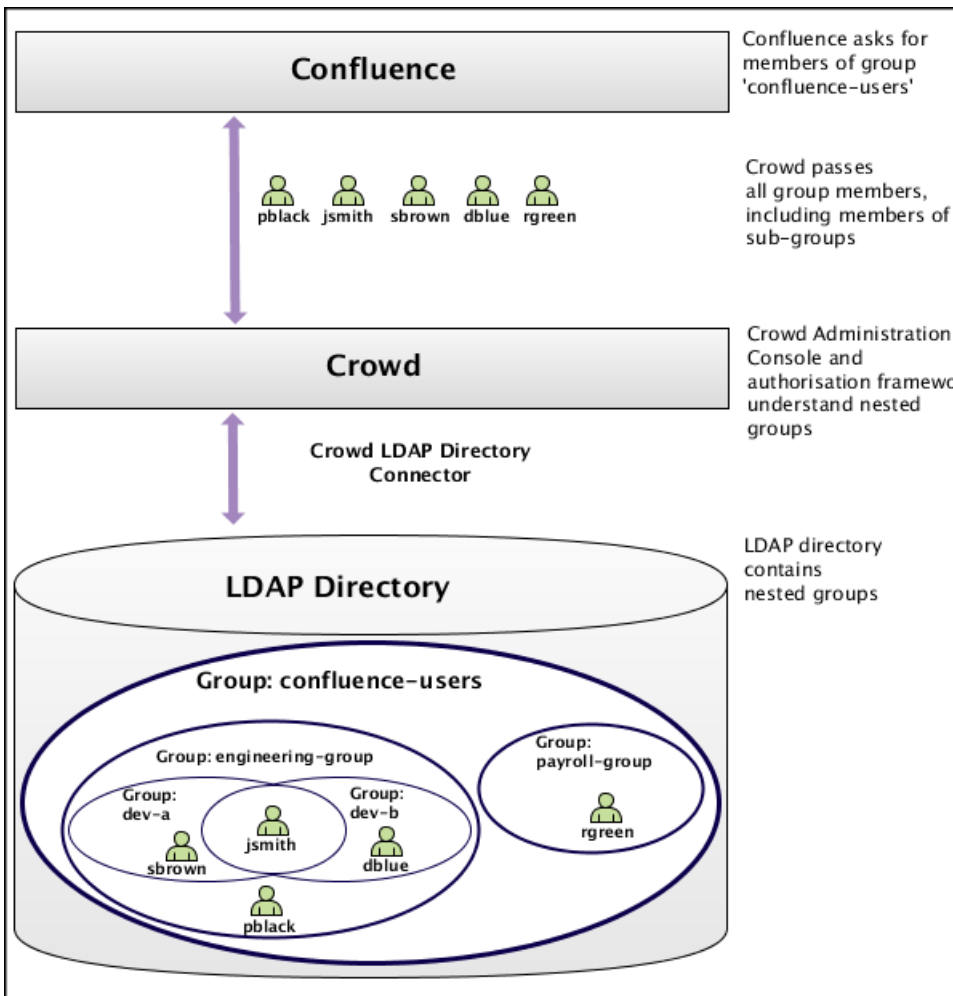
✔ Use Case: Confluence Requests a List of Users in 'confluence-users' group

A [Crowd-integrated Confluence](#) instance will see users in sub-groups as members of the parent group, allowing administrators to use nested groups to manage permissions. (This will not affect Confluence instances that are not Crowd-enabled.)

For example:

- In LDAP we have groups '**engineering-group**' and '**payroll-group**'. We want to grant both groups access to our Confluence site.
 1. Using Crowd, we [add a group](#) called '**confluence-users**' in the LDAP directory.
 2. Add the '**engineering-group**' as a [sub-group](#) of '**confluence-users**'.
 3. Add the '**payroll-group**' as a [sub-group](#) of '**confluence-users**'.
- Group memberships are now:
 - **confluence-users** sub-groups: **engineering-group**, **payroll-group**
 - **engineering-group** sub-groups: **dev-a**, **dev-b**; users: **pblack**
 - **dev-a** users: **jsmith**, **sbrown**
 - **dev-b** users: **jsmith**, **dblue**
 - **payroll-group** users: **rgreen**
- When Confluence requests a list of users in the '**confluence-users**' group, Crowd will present the following list:
 - **pblack**
 - **jsmith**
 - **sbrown**
 - **dblue**
 - **rgreen**

[Diagram: Presenting Flattened Lists of Users to Integrated Applications](#)



User Management via Integrated Applications

i Recommendation: Enable External User Management

If you have [Jira](#), [Confluence](#), [Bitbucket Server](#), [Bamboo](#), [FishEye](#) or [Crucible](#) connected to Crowd, and you have nested groups in your directory, we recommend that you turn **on** external user management, via the administration screen of the integrated application. This will avoid confusion in the user-management screens of the integrated application, since these applications do not understand the concept of nested groups.

✔ Use Case: Application Adds a User to a Group

If an [integrated application](#) adds a user to a [flattened](#) group, the user is added to the named group and not to any of its sub-groups.

✔ Use Case: Application Removes a User from a Group

If an [integrated application](#) attempts to remove a user from a [flattened](#) group, Crowd will do the following:

- If the user is a member of the top group in the hierarchy (tree) of groups contained in the flattened list (e. g. `confluence-users`), Crowd will remove the user.
- Otherwise, Crowd will return an error stating that the user is not a direct member of the group.

Further Notes on Crowd's Processing

- Crowd handles circular/cyclical references. For example, '`group1`' is a member of '`group2`', '`group2`' is a member of '`group3`', and '`group3`' is in turn a member of '`group1`'.

- Crowd ignores members which are not users or groups. Group members might be computers, printers, etc.
- Crowd gracefully handles unreachable groups. There may be references to groups or members that Crowd cannot enumerate. This might be because the referenced group no longer exists, or the LDAP group structure is not entirely consistent. Crowd will ignore such groups and print a warning to the [log file](#).


Adding a Sub-Group


If your directory supports [nested groups](#), you can add a group as a member of another group. This page tells you how to add such a sub-group.


About nested groups

Some user directories allow you to define a group as a member of another group. Groups in such a structure are called '**nested groups**'. In Crowd, you can [map any group to an application](#), including a group which contains other groups. Crowd supports nested groups for LDAP directory connectors, Crowd internal directories, Delegated Authentication directories and custom directories. You can **enable or disable** support for nested groups on each directory individually. For more information, refer to the documentation on [configuring a directory](#). For more details about nested groups, refer to [Nested Groups in Crowd](#).

To add a sub-group,

1. Log in to the [Crowd Administration Console](#). First, **enable nested groups** via the **directory configuration** screen, the 'Add Groups' button will not appear if nested groups are not enabled for your directory.
2. In the top navigation bar, click the **Groups**.
3. Select a directory and click **Search** to list all the groups that exist in that directory. You can refine your search by specifying a **Name** or by choosing **Active** or **Inactive** groups.
4. Now, you need to edit the parent group which will contain the sub-group:
 - If the parent group does not yet exist, [add it now](#).
 - If the parent group already exists, find it in the list of groups and click the link on the group name to view the group details.
5. In the **View Group Details** screen, click the **Direct Members** tab.
6. This will display a list of the selected group's members, both the groups and the users that are direct members of the group. See the [screenshot below](#). Click the '**Add Groups**' button.
 The 'Add Groups' button will not appear if nested groups are not enabled for your directory. You can enable nested groups via the directory configuration screen.
7. Enter your search criteria in the **Search** textbox.

 You can enter all or part of the group name. Leave the search box empty to match all group names.
You can refine your search by choosing '**Active**' or '**Inactive**' groups.
You can also set the '**Maximum Results**', i.e. the number of groups to be retrieved.

8. Click **Search**.
Crowd lists the groups in the selected directory that match your search criteria, but excluding groups that are already sub-groups of the selected group.
 Crowd displays a maximum number of groups as specified in the **Maximum Results** field. If too many groups match the search, you can change the search criteria and click 'Search' again. (There is no way to move to the next page of matching groups.)
9. Select the groups by putting a tick in the checkbox next to one or more group names. To select all groups, you can put a tick in the checkbox at the top of the table.
10. Click **Add Selected groups** button to add the selected groups to the group.

Screenshot: Direct members of a group

View Group – my-team

Details Direct Members Nested Members

Groups in this Group

Group Name	Description	Active
team2	Team 2	true

Add Groups Remove Groups

Users in this Group

Username	Email	Active
adent	adent@example.com	true
admin	smaddox@atlassian.com	true
trillian	trillian@example.com	true

Add Users Remove Users

Screenshot: Popup for adding sub-groups

Add Groups

Search : Search

Active : All Maximum Results : 100

<input type="checkbox"/>	Name	Description
<input type="checkbox"/>	crowd-administrators	
<input checked="" type="checkbox"/>	team3	Team 3

Add Selected Groups Cancel

Group-level administration

Starting from version 3.3, Crowd Data Center allows administrators to delegate some of their responsibilities over groups with Group Level Admins. Group Level Admins have rights to manage members of the groups they were allowed to administer. This also include groups and their members that are not in the same directories as Group Level Admins.

Group Level Administrators can add users to groups and remove them from groups. Group Level Administrators can't create new users nor can they delete users. Changes made by Group Level Administrators might take extra time to be reflected in the applications.

Group-level administration rights are can be given by only by Crowd administrators and they can be assigned both to individual users as well as groups.

Once logged in as a Group Level Administrator, in the **Groups** section you're able to see all the groups you can manage. Click on the group name, to see a list of all group members.

Users in a group in the group-level admin view

View group - company contractors in Company LDAP

If a user has been added to or removed from a group, it might take applications that use Crowd a while to retrieve this information.

Bobbie Brown x James Marstens x Add

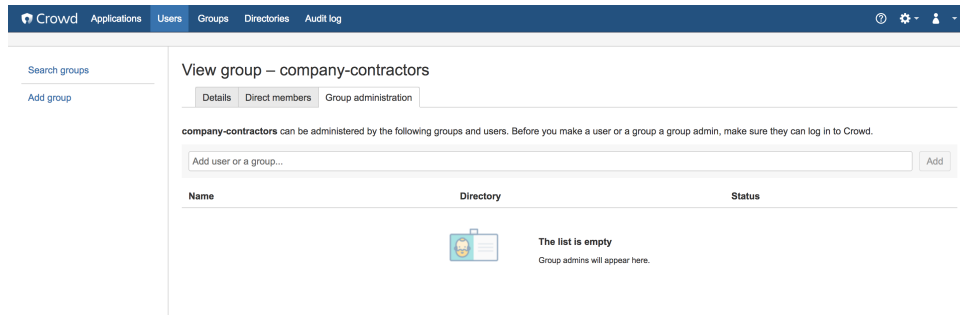
Name	Username	Email address	Status
JohnDoe	jdoe	jdoe@email.com	active x
Alex White	awhite	awhite@email.com	active x
Makie Black	mblack	mblack@email.com	active x
Daren Arofonvosky	darofonovsky	darofonovsky@email.com	active x
Ula Lachovitsch	ulachovitsch	ulachovitsch@email.com	active x
Abelard Sans	asans	asans@email.com	active x
Jon Snow	jsnow	jsnow@email.com	active x
Malin Trooper	mtrooper	mtrooper@email.com	active x
Becky Norman	bnorman	bnorman@email.com	active x
Stan Stanford	sstanford	sstanford@email.com	active x
Ollie King	oking	oking@email.com	active x
Random Name	rname	rname@email.com	active x
Tim Darryl	tdarryl	tdarryl@email.com	active x
Olivia Gates	ogates	ogates@email.com	active x
Eggy Knight	eknight	eknight@email.com	active x
Alana Alana	aalana	aalana@email.com	active x

Adding Group Level Admins

 Group-level administration exists only in Crowd Data Center.

To add Group Level Admins:

1. In the top navigation, click **Users**.
2. Select the **Group administration** tab.
A list of current group administrators is displayed. The list does not include groups from uncached Connector directories, Custom directories, and Azure AD directories which cannot be administered by Group Level Admins.

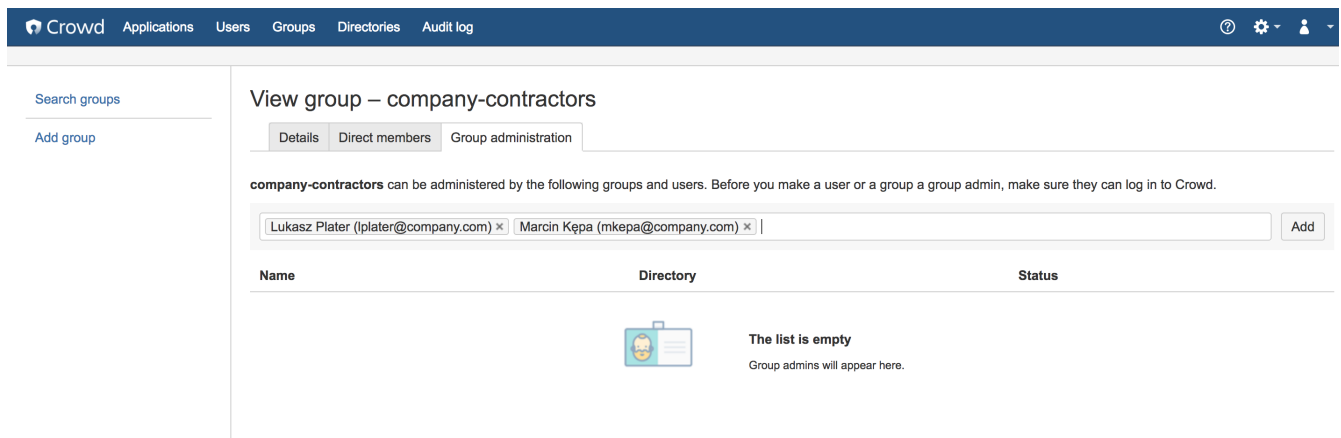


3. Add a new Group Level Admins by using the autocomplete text box.
Remember that group administration rights can be assigned both to users as well as groups so you can save yourself some time selecting multiple users and select the whole group instead.

Group Level Admins and nested groups

When a group in a directory with nested groups becomes a group administrator of another group, all effective members of the administering group will be allowed to administer the group.

However Group Level Admins of a group with child groups will only be able to administer the group they are directly assigned administrative rights to.

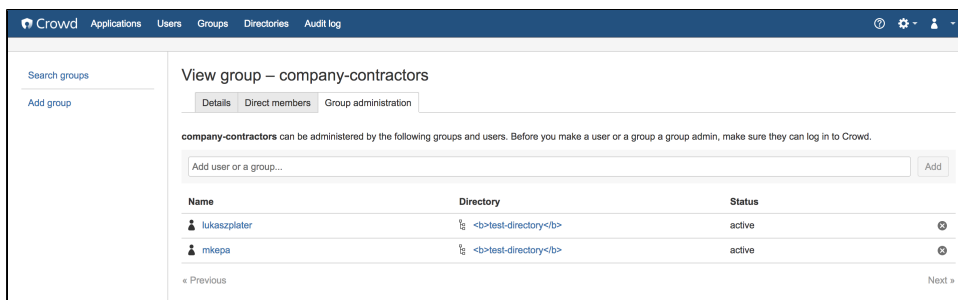


Removing Group Level Admins

 Group-level administration exists only in Crowd Data Center.

To remove Group Level Admins:

1. In the top navigation, click **Users**.
2. Select the **Group administration** tab.
A list of current group administrators is displayed. The list does not include groups from uncached Connector directories, Custom directories, and Azure AD directories which cannot be administered by Group Level Admins.
3. Remove a Group Level Admin by clicking the X button next to their name.
Remember that group administration rights can be removed from both users as well as groups so you can save yourself some time selecting multiple users and select the whole group instead.





Removing a Sub-Group

If your directory supports [nested groups](#), the directory may contain groups which are members of other groups. This page tells you how to remove a group's membership of another group. Note that removing a sub-group does **not delete the group**.

About nested groups

Some user directories allow you to define a group as a member of another group. Groups in such a structure are called '**nested groups**'. In Crowd, you can [map any group to an application](#), including a group which contains other groups. Crowd supports nested groups for LDAP directory connectors, Crowd internal directories, Delegated Authentication directories and custom directories. You can **enable or disable** support for nested groups on each directory individually. For more information, refer to the documentation on [configuring a directory](#). For more details about nested groups, refer to [Nested Groups in Crowd](#).

To remove a sub-group:

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click **Groups**.
3. Select the directory in which you are interested, then click **Search**.
You can refine your search by specifying a **Name** or by choosing **Active** or **Inactive** groups.
4. Find the parent group in the list of groups and click the link on the group name to view the group details.
5. Click the **Direct Members** tab.
This will display a list of the selected group's members, both the groups and the users that are direct members of the group.
6. Click **Remove Groups**.
 The 'Remove Groups' button will not appear if nested groups are not enabled for your directory. You can enable nested groups via the directory configuration screen.
7. Enter your search criteria in the **Search** textbox.
You can enter all or part of the group name. Leave the search box empty to match all group names. You can refine your search by choosing **Active** or **Inactive** groups. You can also set the **Maximum Results**, i. e. the number of groups to be retrieved.
8. Click **Search** button.
Crowd will list the groups in the selected directory that match your search criteria and are sub-groups of the selected group.
 Crowd will display a maximum number of groups as specified in the '**Maximum Results**' field. If too many groups match the search, you can change the search criteria and click 'Search' again. (There is no way to move to the next page of matching groups.)
9. Select the groups by putting a tick in the checkbox next to one or more group names.
10. Click **Remove Selected Groups** to remove the selected sub-groups from the group.

Specifying a User's Attributes

A user's default *attributes* are specific to the [directory](#) to which the user belongs. You can add other attributes (e.g. address, phone number, date of birth) manually as required.


Cannot add attributes to LDAP directories

You cannot add new attributes to directories connected via Crowd's [LDAP connector](#), although you can update the existing supported attributes as described in our [LDAP connector documentation](#). Any new attributes added via the Crowd Administration Console will simply not appear in the directory, even though they are stored locally on the Crowd server.

To edit a user's attributes

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click **Users**.
The [User Browser](#) will appear.
3. Search for the user you want to update, and the the user's name.
4. In **User Details** screen, click the **Attributes** tab.


- **To add a new attribute,**

-  You cannot add an attribute to an LDAP directory see note above.

1. Enter the name of the new attribute (e.g. `phone`) in the **Attribute** field.
2. Enter the value of the new attribute (e.g. `0123456789`) in the **Value** field.
3. Click **Add**.

- **To edit an existing attribute,** edit the corresponding field in the **Values** column, then click **Update**.

- **To delete an attribute,** click the corresponding **Remove** link in the **Action** column.

 Note that some attributes may correspond to particular fields on the [User Details](#) screen. However, attributes are optional whereas the 'Details' fields are all required.

View user – mkowalsky

Details [Attributes](#) Groups Applications

Attribute	Values	Action
invalidPasswordAttempts	<input type="text" value="0"/>	Remove
lastActive	<input type="text" value="1550829086613"/>	Remove
lastAuthenticated	<input type="text" value="1550829086608"/>	Remove
passwordLastChanged	<input type="text" value="1550828643468"/>	Remove
requiresPasswordChange	<input type="text" value="false"/>	Remove
<input type="text" value="Attribute"/>	<input type="text" value="Value"/>	<input type="button" value="Add"/>

Granting Crowd Administration Rights to a User


Members of the '**crowd-administrators**' group have administration privileges that is, they can:

- Access the [Crowd Administration Console](#) and perform the functions described in the [Administration Guide](#).
- Access the CrowdID '**Administration**' menu and perform the functions described in the [CrowdID Administration Guide](#).

The '**crowd-administrators**' group is automatically created in your default directory when you install Crowd. (See [Running the Setup Wizard](#).) If you need to grant Crowd administration rights to users in other directories, you can create a '**crowd-administrators**' group in any or all of your other directories and [map the directories](#) to the '**crowd**' application.

To grant administration privileges to a user

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click **Users**.
The [User Browser](#) will appear.
3. Select the relevant directory, search for the user you want to update, and click the user's name.
The **User Details** screen will appear.
4. Click the **Groups** tab.
A list of the user's current groups (if any) will appear.
5. From the drop-down list, Select the **crowd-administrators** group, and click **Add**.

 If you wish, you can use a different or additional group to contain your Crowd administrators. To do this, map your chosen group(s) to the '**crowd**' application as described in [Specifying which Groups can access an Application](#). Note that CrowdID administrators, however, must always belong to the '**crowd-administrators**' groups.

Granting Crowd User Rights to a User

This page tells you how to authorize users to access Crowd, without giving them Crowd administration rights. Only [Crowd administrators](#) can authorize other users to access Crowd.

Administrators and Non-Administrators

The [Crowd Administration Console](#) presents the full range of Crowd administration functionality to authorized [Crowd administrators](#).

Authorized Crowd users who are **not** administrators can also access the Crowd Console. They will see a subset of functionality, which we call the 'Self-Service Console'. Refer to the [User Guide](#) for details of this functionality.



Non-administrators cannot affect other users or the Crowd installation

Granting Crowd user rights will give your users the power to update their own profiles and passwords and view their authorization details. But they will not be able to view or update other user profiles, nor perform any Crowd administration functions.

Authorizing Non-Administrators to Use the Crowd Self-Service Console

To authorize a non-administrator to use Crowd, you should ensure that both of the following are true:

- The person's username is in a user directory where all users are authorized to use Crowd. See the instructions below.
- The person is **not** a member of a group mapped to the 'crowd' application. (Group members will have [Crowd administration rights](#).)

To grant an entire directory access to Crowd

1. Log in to the [Crowd Administration Console](#).
2. [Map your chosen user directory](#) to the **crowd** application.
3. On the **Directoriestab**, set the **Allow All to Authenticate** option to **True**.
4. [Add the user\(s\)](#) to the directory, if not already added.

Managing a User's Session



Number of Sessions

For Crowd 2.0.4 and newer versions, a single session is allowed for each user in a machine accessing an application integrated to Crowd. So, for instance, if you are accessing Jira and then open a new Browser model and try to login to the same application, two sessions will be created in the issue tracker, however a single session will be created in Crowd. If one of the sessions is terminated in Jira, all the sessions will be terminated.


For any given directory, Crowd allows you to see which users are currently logged in to one or more applications that use the [Crowd framework](#).

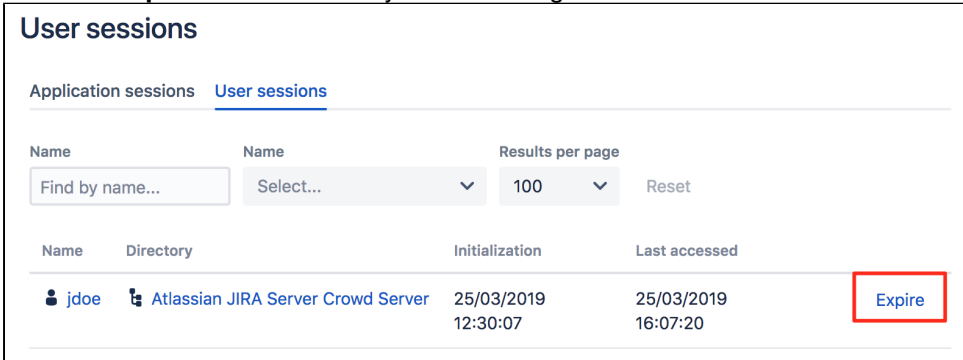
You can also force any session to expire, that is, you can log the user out of Crowd.



To see which users are currently logged in to Crowd


1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, go to **Administration**  **>Current sessions**. This will display the **Session Browser**.
3. Click the **User Session** tab.
4. Select the directory that contains the users in which you are interested, and click **Search**. This will display a list of all users, within your chosen directory, who are currently logged in to the Crowd framework.
 -  You can refine your search by specifying a user's **'Name'** (note that this is case-sensitive).

To log a user out of Crowd

1. Login to the [Crowd Administration Console](#).
2. In the top navigation bar, go to **Administration**  **>Current sessions**.
3. Click the **User Session** tab. This will display a list of all users which are currently logged in to the Crowd framework.
4. Click the **Expire** link for the user you want to log out.



User sessions			
Application sessions		<u>User sessions</u>	
Name	Name	Results per page	
<input type="text" value="Find by name..."/>	<input type="text" value="Select..."/>	<input type="text" value="100"/>	<input type="button" value="Reset"/>
Name	Directory	Initialization	Last accessed
 jdoe	 Atlassian JIRA Server Crowd Server	25/03/2019 12:30:07	25/03/2019 16:07:20
			<input type="button" value="Expire"/>

 If you want to *permanently* prevent a user from logging in to Crowd, please see [Deleting or Deactivating a User](#).

System Administration

- [Configuring Server Settings](#)
 - [Deployment Title](#)
 - [Domain](#)
 - [Session configuration](#)
 - [Authorization Caching](#)
 - [Licensing](#)
 - [Crowd SSO 2.0](#)
 - [Finding your SEN](#)
 - [SSO Cookie](#)
- [Configuring your Mail Server](#)
- [Creating an Email Notification Template](#)
- [Configuring Trusted Proxy Servers](#)
- [Viewing Crowd's System Information](#)
- [Backing Up and Restoring Data](#)
- [Logging and Profiling](#)
 - [Performance Profiling](#)
- [Draft - Troubleshooting and Requesting Technical Support](#)
- [Configuring the LDAP Connection Pool](#)
- [Browsing the audit log](#)
- [Look and feel](#)
- [Overview of Caching](#)

Other Related Security Resources

- [System Administration](#)
- [How to Report a Security Issue](#)
- [Security Advisory Publishing Policy](#)
- [Security Patch Policy](#)
- [Severity Levels for Security Issues](#)
- [Crowd Security Notice 2013-07-01](#)
- [Administration Guide](#)
- [Crowd Security Advisories and Fixes](#)

Configuring Server Settings


You can alter the settings which were specified when your Crowd server was installed:

- [Deployment Title](#)
- [Domain](#)
- [Session configuration](#)
- [Authorization Caching](#)
- [Licensing](#)
- [Crowd SSO 2.0](#)
- [Finding your SEN](#)
- [SSO Cookie](#)

Deployment Title

The deployment title is a unique name for your Crowd instance. The deployment title is used by default in the subject line of [email notifications](#).

To specify the deployment title

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click .
The **General Options** screen appears.
3. Type the new name into the **Deployment Title** field.
4. Click **Save**.

Domain

The **SSO domain** is used when setting HTTP authentication cookies in a user's browser. If this attribute is not correct, single sign-on (SSO) will not work when the user switches between applications.

Overview

The core Crowd functionality supports SSO across applications within a single domain, such as `*.mydomain.com`. Crowd uses a browser cookie to manage SSO. Because your browser limits cookie access to hosts in the same domain, this means that all applications participating in SSO must be in the same domain.


Example 1: If you wish to have single sign-on (SSO) support for `*.mydomain.com`, you will need to configure the SSO domain in Crowd as `mydomain.com`. All your Crowd-connected applications must be in the same domain. For example:

Crowd	<code>crowd.mydomain.com</code>	✓
Jira	<code>jira.mydomain.com</code>	✓
Confluence	<code>confluence.mydomain.com</code>	✓
FishEye	<code>fisheye.mydomain.com</code>	✓
FishEye in different domain	<code>fisheye.example.com</code>	✗

Example 2: If you wish to have single sign-on (SSO) support for `mydomain.com/*`, you will need to configure the SSO domain in Crowd as `mydomain.com`. All your Crowd-connected applications must be in the same domain. For example:


Crowd	<code>mydomain.com/crowd</code>	✓
Jira	<code>mydomain.com/jira</code>	✓
Confluence	<code>mydomain.com/confluence</code>	✓
FishEye	<code>mydomain.com/fisheye</code>	✓
FishEye in different domain	<code>example.com/fisheye</code>	✗

You can find information the comparison of host name strings in [RFC 6265](#) (section 5).

 When developing on your local machine, you should set the domain to `localhost`.

Setting the SSO Domain

To specify the domain:

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click .
3. Type the new domain into the **SSO Domain** field.
4. Click **Save**.

Setting the SSO Domain when Crowd is behind a Proxy Server

If Crowd is being run behind a proxy server, before setting the SSO domain value, make sure that the domain specified in the proxy (that is currently being used to access the Crowd console) was specified in the Tomcat connector **proxyName** attribute. Example:

File: Apache-Tomcat/conf/server.xml

```
<Connector acceptCount="100" connectionTimeout="20000" disableUploadTimeout="true" enableLookups="false"
maxHttpHeaderSize="8192" maxSpareThreads="75" maxThreads="150" minSpareThreads="25"
port="8095" redirectPort="8443" useBodyEncodingForURI="true"
proxyName="mycompany.com" />
```

Notes

- **Avoiding problems with old cookie versions.** In order to avoid problems with hosts or domains defined in old cookie versions, after setting the SSO Domain in Crowd, log out of Crowd and the integrated applications and delete all the web browser cookies.
- **SSO domain.** The 'SSO Domain' field will accept only values based on the domain that is used to access the Crowd console. For instance, if you are using '**www.mycrowd.com/crowd/console**' to access the console in the web browser, this field will accept the following values:
 - Empty
 - mycrowd.com
 - .mycrowd.com

If you enter any other value, Crowd will show an error message: *The supplied domain is invalid.*

- **IP addresses.** SSO will not operate when sites are accessed using IP addresses rather than domain names. This is a limitation of the cookie technology implemented in web browsers.

Session configuration


This page tells you how to set the [timeout period for a session token](#) and how to enable/disable [in-memory token storage](#).

Session Timeout

When a successful authentication occurs, for either an application or a user, a unique token is assigned. Tokens are valid for the period of time specified as the 'Session Timeout' attribute.

The session timeout determines how long a session will be considered valid during any period of inactivity. This value is specified in minutes and must be greater than 0.

To specify the session timeout:

1. Log in to the [Crowd Administration Console](#).
2. In the upper-right corner, click .
3. In the left-hand menu, **Session Configuration**.
The '**Session Config**' screen will appear, as shown [below](#).
4. Type the new value into the '**Session Timeout**' field, then click the '**Update**' button.


Require Consistent Client IP Address

 (Available since Crowd 2.5.2.)

Authenticated sessions can be tied to the IP address they were created from. This means that an attempt to use that session from another machine will fail, which will force mobile clients to reauthenticate when their IP address changes.

This setting can be disabled to relax that requirement, so a session can be used from any IP address. Note that changing this setting will invalidate any existing sessions, so **you will be logged out after making this change**.

To allow sessions to be used from any IP address,

1. Log in to the [Crowd Administration Console](#).
2. In the upper-right corner, click .
3. In the left-hand menu, **Session Configuration**.
The '**Session Config**' screen will appear, as shown [below](#).
4. Check or uncheck **Require Consistent Client IP Address** as required.
5. Click **Update**.

Authentication Token Storage

Authentication tokens are used to validate application and user sessions. A token is stored for each active session. By default, they're kept in the Crowd database. Storing these tokens in memory can benefit performance, but with one significant drawback, that sessions will not be saved across Crowd restarts. If you restart Crowd, all your users will have to log in again.

In-memory token management is not available in Crowd Data Center


Switching from database to in-memory token management does not require a restart of Crowd; nor will sessions be lost or validations failed. However, if you have lots of active sessions, and therefore lots of tokens, it can take some time to copy the token information. During this time, validation requests will be queued and Crowd will appear unresponsive to client applications.

As a guide, below are some benchmarks of time taken to switch from one form of token storage to the other. The measurements were taken on a quad-core Mac Pro, using a lightly-loaded PostgreSQL database:

Number of Tokens:	100	500	1000	5000	10000
Database -> Memory	0.1s	0.7s	1.2s	4.2s	8.2s
Memory -> Database	1.2s	4.8s	9.2s	45s	90s

To switch the token storage location:

1. Log in to the [Crowd Administration Console](#).

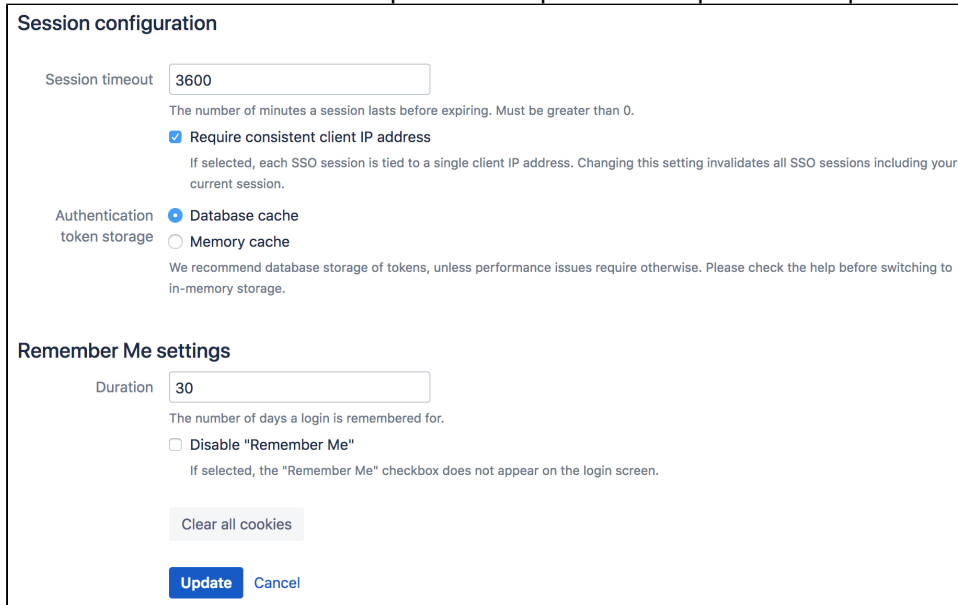
2. In the upper-right corner, click .

3. In the left-hand men, **Session Configuration**.

The '**Session Config**' screen will appear, as shown [below](#).

4. Next to **Authentication Token Storage**, select one of the radio buttons :

- '**Database Cache**' This is the default option. Select it to store your tokens in the Crowd database. We recommend this option unless performance problems require in-memory storage.



Session configuration

Session timeout
The number of minutes a session lasts before expiring. Must be greater than 0.

Require consistent client IP address
If selected, each SSO session is tied to a single client IP address. Changing this setting invalidates all SSO sessions including your current session.

Authentication token storage **Database cache**
 Memory cache
We recommend database storage of tokens, unless performance issues require otherwise. Please check the help before switching to in-memory storage.

Remember Me settings

Duration
The number of days a login is remembered for.

Disable "Remember Me"
If selected, the "Remember Me" checkbox does not appear on the login screen.

- '**Cache**' Select this option to store your tokens in memory.

5. Click **Update**.

Memory

Screenshot: 'Session Config'

In-memory cache size

The size of the in-memory token cache is defined in the `crowd-webapp/WEB-INF/classes/crowd-ehcache.xml` file. The default should be acceptable for most cases. If you require more than 2048 concurrent sessions in memory you may increase the size of the '`-hash-cache`' caches.

Remember Me settings

Use this setting to define the number of days a user's login is remembered for. You can also turn the Remember Me option off to force the need to provide credentials every time a users wants to log in.

Clear cookies

If for any reasons, like for example security breach, you want to force your users to log in again, click **Clear all cookies**. This will force all users to log in again once their session expires. The session expiration time is configured in the session timeout setting in the same screen.

Authorization Caching

Caching is used to store run-time authentication and authorization rules, which can be expensive to calculate.

This page describes the cache that can be configured on the Crowd server, to store users' authentication and per-application permissions for a specified period. For an overview of the other types of caching offered by Crowd, please refer to [Overview of Caching](#).

Caching of Users' Application Permissions on the Crowd Server The Authorization Cache

Crowd can store users' authentication and per-application permissions in a local cache for a specified period after retrieving the information from the directory and application data. The cached data will answer the following questions:


- For a particular user: Is the user authenticated?
- For a particular user and application: Does the user have access to the application?

You might call this the 'has access' cache, or the '**authorization cache**'.

Recommended setting: **Enabled**. For performance reasons, we recommend that the cache be enabled on the Crowd server. This is the default setting.

The effect of caching the data is that users will retain access to applications for a period after their username or permission has been removed, i.e. until the server-side cache expires. You should disable the cache only if you need immediate results when removing users or their permissions.

To enable caching of user-to-application permissions on the Crowd server,

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click .
3. Select **Enable Authorization Caching**.
4. Click **Save**.

Some applications may enable/disable caching based on the Crowd server setting

The Crowd API allows an application to query whether caching is enabled on the Crowd server (`isCacheEnabled`). The Crowd Java client does not make use of this API feature, because it makes more sense to have application caching configured entirely on the application side. If you have a Crowd-integrated custom application which does make use of this API call, then the setting on the [Crowd server](#) will affect your application-side caching as well.

Licensing

Crowd licenses are based on the number of end-users who will log in to the applications that are integrated with Crowd.


You can obtain an evaluation license from the [Atlassian](#) website. When you obtain an evaluation license or purchase, renew or upgrade your license you will receive a license key via email or on the Atlassian website. You will need to enter your license key into your Crowd server as described below.

Important changes to our server and Data Center products

We've ended sales for new server licenses, and will end support for server on February 2, 2024. We're continuing our investment in Data Center with several key improvements. [Learn what this means for you](#)

Entering your License Key

To enter your license key,

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click .
3. Click **Licensing**.
4. Fill in the **License Key** field.
5. Click **Save**.

Warning when Number of Users approaches License Limit

Whenever the number of users reaches 90% of the number allowed by the license, Crowd will send an email informing the administrator about the license limit and the current number of users. The email is sent to the email notification address, as defined on the '**Mail Configuration**' screen in the Crowd Administration Console. (See [Configuring your Mail Server](#).)

It is important to avoid exceeding the license limit, because once the user limit has been exceeded, no one can log in (including administrators).

What to Do if the Number of Users Exceeds your License Limit


If the number of users who are allowed to log in to the Crowd console exceeds the user license limit, no one will be able to log in to any applications. When you try to access Crowd, you'll be redirected to a page where you can update your license key. On this page, you can either:

- Enter a new license key for a [higher user count](#). (Purchase one [here](#).)
- Enter a 30-day evaluation license key. (Create one [here](#).) The 30-day evaluation key lets an unlimited number of users sign in during the trial period. You can use the 30 days to clean up the users in the system or [purchase](#) a license key for a higher user count.

Minimizing your Licensing Cost

If you have more than one directory, ensure that the same user does not exist in multiple directories.


We recommend that you allow only [particular groups](#) to log in to each application, rather than entire directories.

 Note that a mapped application can 'see' all users in a directory, even if not all of them can log in to the application. For example, a Human Resources application might be mapped to your entire Active Directory server, but only the HR group is allowed to log in to the application.

Recalculating your User Total

The Licensing screen shows the number of users who currently count towards your license. This total is updated automatically at regular intervals. If you have recently added or removed users, the total may not be up to date when you view the screen. You can update the count immediately, as described below.

To recalculate your user total:

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click .
3. Click **Licensing**.
4. Click **Recalculate your user total**.
5. The recalculation may take a while, depending on the size of your user base.

List all users who count toward the Crowd license

To list users who consume license in Crowd:

1. Enable ***com.atlassian.crowd.manager.license*** in the DEBUG level
2. Recalculate your User Total.

Once the calculation is ready, you will be able to see the license count, list of users (and the directories those users are from) in log files.

Example:

```
2018-03-28 07:12:15,789 Caesium-2-2 DEBUG [crowd.manager.license.CrowdLicenseManagerImpl] Finished
counting licensed users, 9 total
2018-03-28 07:12:15,790 Caesium-2-2 DEBUG [crowd.manager.license.CrowdLicenseManagerImpl] Licensed users
are: [leela (131074), professor (131074), test (753665), zoidberg (131074), admin (131073), hermes
(131074), bender (131074), amy (131074), fry (131074)]
```

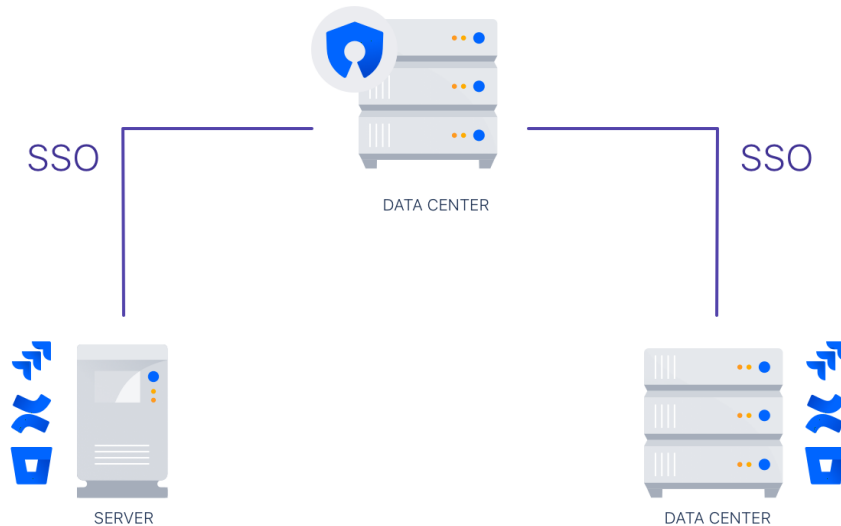
Server ID and Support Entitlement Number

Your **License Server ID** is generated automatically, based on your license key.

The **Support Entitlement Number** will appear only on newer licenses. If your License Server ID starts with a 'B', you should also have a Support Entitlement Number. This number is not currently used, but will be used by Atlassian Support in the future.

Crowd SSO 2.0

Single sign-on (SSO) authentication allows you to use a single set of credentials to access multiple applications. The SSO service authenticates you for all the application youve been given rights to and eliminates any further prompts for authentication during the same session. Crowds SSO 2.0 allows you to access Jira, Jira Service Desk, Bitbucket, and Confluence across different domains both Server and Data Center with one common login page.



The SSO 2.0 functionality is available with **Crowd Data Center**



Data Center

Once you configure SSO 2.0 in Crowd Data Center, you can use it to access Server and Data Center applications. If you are not a Crowd Data Center license holder, you can create your evaluation license for Crowd Data Center and take Crowd for a spin. You can get your free Crowd Data Center evaluation license from [Atlassian license evaluation page](#).

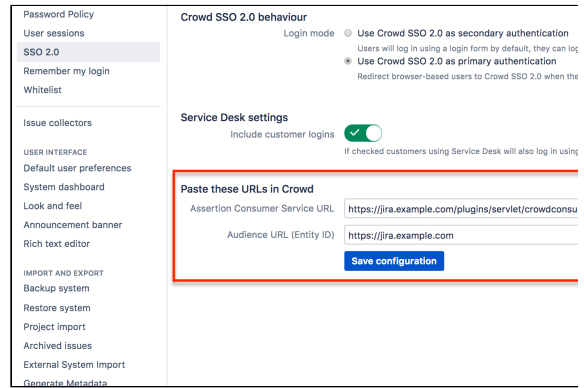
Before you begin

- To enable SSO 2.0 for your application, you must perform configuration on the Crowd side and on the application side.
- Make sure you're using Crowd version 3.4 or later. You can download the latest version of Crowd [here](#).
- You must upgrade the Crowd and SAML Single Sign-On 2.0 plugin for every application for which you want to use SSO 2.0. Go to [Atlassian Marketplace](#).
- Only users existing in Crowd user directory can use SSO 2.0. Make sure that users from your application's individual user directories also exist in Crowd user directory. To avoid any potential conflicts, mapping of user directories on the application side and Crowd must be identical.
- To access their applications using SSO 2.0 users must have permission to access Crowd as well. Its enough to give your users basic Crowd log in rights. For security reasons, we advise to check if your users dont fall into the Crowd admin group.
- From the Crowd and SAML Single Sign-On 2.0 plugin configuration section in your application, copy the application details.

1. In your application, go to **Configuration > System > SSO 2.0**.

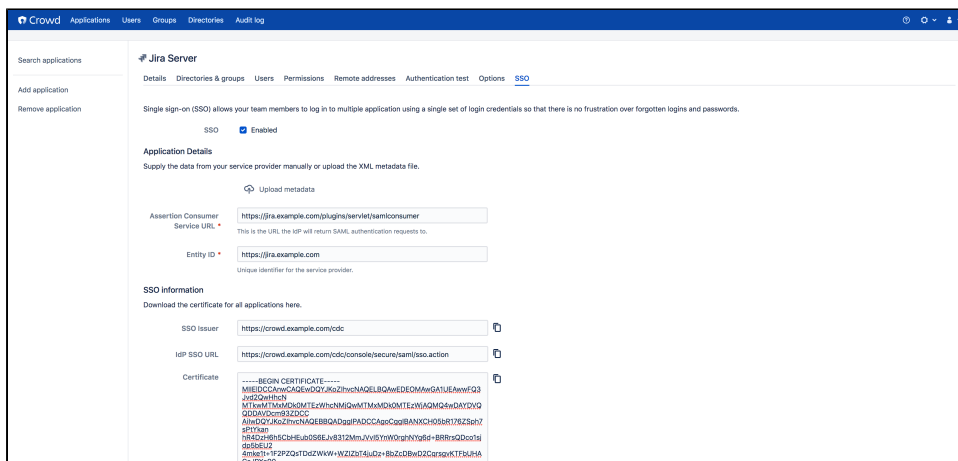
2. Copy the following data to clipboard. You will need to copy this in Crowd SSO 2.0 configuration later.

- Assertion Consumer Service URL
- Entity ID



To enable SSO 2.0 in Crowd:

1. In Crowd top navigation, click **Applications**.
2. Select the application you want to perform the SSO configuration for.
3. In the application settings, click the **SSO** tab.

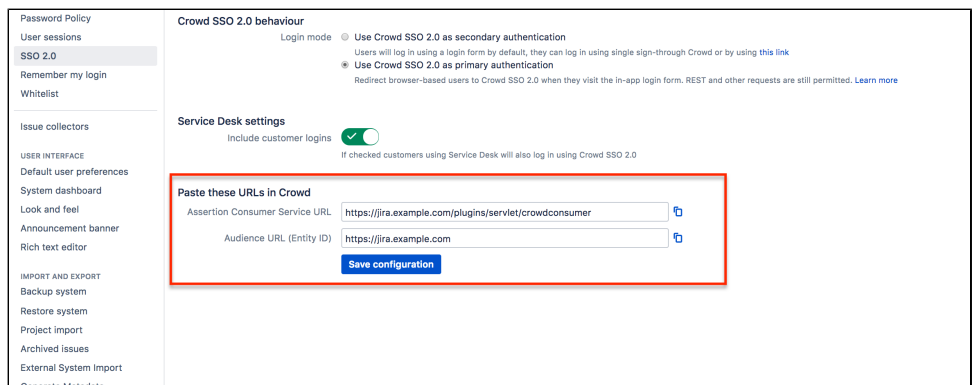


4. Select the **SSO Enabled** checkbox.
5. Copy the Application Details from your Crowd and SAML Single Sign-On 2.0 configuration section in your application.

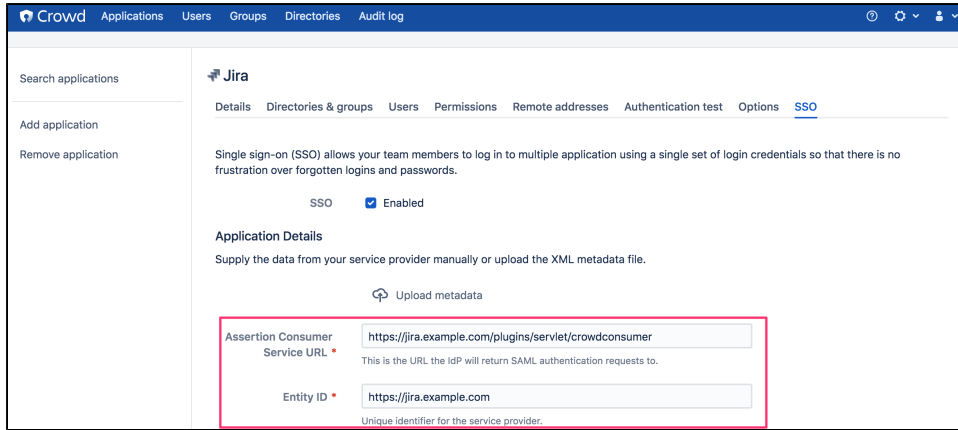
1. In your application, go to **Configuration > System > SSO 2.0**.

2. Copy the following data to clipboard.

- Assertion Consumer Service URL
- Entity ID



3. Paste copied data to the Application details section in Crowd SSO tab for your application.



6. From the Crowd SSO tab, copy the following SSO information:

- SSO issuer
- IdP SSO URL
- Certificate



7. Paste the copied SSO information to your SSO 2.0 plugin for your application.

In Jira:


1. Go to  > **System** > **SSO 2.0**.

2. Paste the copied SSO information to the relevant fields:

Crowd SSO 2.0 settings

Crowd URL *
The URL your Crowd is at, e.g https://crowd.example.com/cdc

Certificate *
Copy and paste the entire certificate from your Crowd Data Center.

 Crowd's certificate is by default valid for 5 years. After that time, you'll have to regenerate the certificate and manually copy it over to individual applications for which you want to use SSO 2.0. In case of a security breach, for safety reasons we suggest that you regenerate the certificate and copy it to your applications immediately. For information on how to reset the certificate, see [Crowd REST API Reference](#).

Remember that a regenerated certificate needs to be again provided in all application you want to use SSO with.

8. Click **Save**.

Next steps

- To access their applications using SSO 2.0 users must also have the permission to access Crowd. Its enough to give your users basic Crowd login rights. We advice taking extra care when copying permissions from your application to Crowd so that the group does unauthorized users don't get admin permissions.
- You can test SSO 2.0 before enabling it for all users.

1. In you application configuration, go to the SSO 2.0 plugin settings.

2. For **Crowd SSO 2.0 behaviour** logging in mode, select **Use Crowd SSO 2.0 as secondary authentication**.

This will allow you to test if the configuration works properly before redirecting all your users to the new flow.


3. Save configuration.

Once the settings are saved, you can try to log in as one of the users using the link provided in the Crowd SSO 2.0 behavior configuration under the secondary authentication option. See the screenshot.

Crowd SSO 2.0 behaviour

Login mode Use Crowd SSO 2.0 as secondary authentication
Users will log in using a login form by default, they can log in using single sign-through Crowd or by using [this link](#)

Use Crowd SSO 2.0 as primary authentication
Redirect browser-based users to Crowd SSO 2.0 when they visit the in-app login form. REST and other requests are still permitted. [Learn more](#)




The link redirects you through <https://jira.yourdomain.com/plugins/servlet/external-login> to the common login page provided by Crowd SSO 2.0 that will be used for every user to access Jira, Confluence, Bitbucket or any other application connected and configured in CrowdSSO 2.0 once Crowd SSO 2.0 is selected as primary authentication.

Once you can log in successful through that new common login page from Crowd, you can go back to your application configuration in section SSO 2.0 and change the settings for CrowdSSO 2.0 behavior to: **Use Crowd SSO 2.0 as primary authentication** and save your configuration. From now on everyone using this application will log in through Crowd SSO 2.0 common login screen.

Finding your SEN

There are three ways to find your Support Entitlement Number (SEN).

Method 1: View the license details in the Crowd Administration Console

Go to  > **Licensing**. Sen is displayed in the Licensing screen.

Method 2: Log in to my.atlassian.com as the account holder or technical contact

Your Support Entitlement Number is available at <http://my.atlassian.com>.

Method 3: Look at your Atlassian Invoice

Your Support Entitlement Number (SEN) appears on the third page of your Atlassian invoice.

More information

See [Finding Your Support Entitlement Number](#) in the support space for more general information about how Atlassian Support uses this number.


SSO Cookie

When using Crowd for single sign-on (SSO), you can specify that the 'secure' flag is set on the SSO cookie. This will enforce a secured connection, such as SSL, for all SSO requests.

Unsecured connections will be rejected

If you set this flag, any applications not using a secure connection will not be able to participate in SSO and users will not be able to log in. Potentially, this may make it impossible to log in to Crowd, if your Crowd Administration Console application is not accessed via SSL.

To specify the secure flag on the SSO cookie

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click .
3. Tick or untick the **Secure SSO Cookie** checkbox as required:
 - Ticked The 'secure' attribute will be included on the SSO cookie. A secured connection, such as SSL or TLS, is required for all SSO requests. Unsecured connections will be refused.
 - Not ticked This is the default. The 'secure' attribute will not be included on the SSO cookie. This means that the SSO cookie may be transmitted over an unsecured connection.
4. Click **Save**.

Configuring your Mail Server


Once you have configured your mail server as described below, Crowd can send email notifications to users at specific events, such as when a user requests a [password reset](#) or a server event occurs.

On this page:

- [Accessing the Mail Configuration Screen](#)
- [Mail Server Option 1: SMTP](#)
- [Mail Server Option 2: JNDI Location](#)

Accessing the Mail Configuration Screen

To configure SMTP email,

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar click,  > **Mail configuration**.
3. Enter the details of your mail server as described below.
4. Click **Update**.

Mail Server Option 1: SMTP

Mail configuration

Server alert email*
 Notification emails will be sent to this address regarding critical server messages, such as when a license is reaching its resource limits.

From email address*
 The sender (or FROM) email address to use when sending email notifications.

Subject prefix
 The subject prefix to use when sending email notifications. This is useful for mail client filtering rules. For example: [ACME CORP - Crowd].

Mail server details

Mail server type SMTP server
 JNDI location
 Choose if you want to use SMTP or JNDI for your mail configuration

SMTP server

SMTP host*
 The host address. For example: localhost or smtp.acmecorp.com.

SMTP port*
 SMTP port number to use (default: 25).

Username
 The username to use when connecting to the mail server.

Password
 The password to use when connecting to the mail server.

Connection timeout*
 (seconds) Maximum time Crowd tries to send messages (default: 60).

Use Secure Sockets Layer (SSL)
 SMTP server requires encryption

Enter the details as follows:

- **Server alert email** The email address which will receive notifications about server events. For example, Crowd will send an email message to this address when the number of users approaches the license limit.
- **From Email Address** Crowd will add this email address as the 'sender' on the emails generated by Crowd and sent to users.
- **Subject Prefix** The prefix which will appear at the start of the email subject, for all emails generated by Crowd. This can be useful for email client programs that offer filtering rules.
- **Mail Server Type** Select the '**SMTP Server**' radio button.
- **SMTP Host** The hostname of the SMTP mail server, e.g. 'localhost' or 'smtp.acme.com'.
- **SMTP Port** The port on which the SMTP mail server listens. The default is '25'.
- **Username** The username that your Crowd server will use when it logs in to your mail server.
- **Password** The password that your Crowd server will use when it logs in to your mail server.
- **Timeout** Maximum time Crowd tries to send messages. The default is '60' seconds.
- **Use Secure Sockets Layer (SSL)** Select this check-box if you want to access your mail server over SSL (Secure Sockets Layer). This ensures that all email communications between Crowd and your mail server are encrypted, provided your mail server supports SSL.

Additionally, as you are connecting to an SSL service, you will need to import the SMTP server certificate into a Java keystore. The process is described in [Configuring Crowd to Work with SSL](#).

Mail Server Option 2: JNDI Location

Mail configuration

Server alert email*
Notification emails will be sent to this address regarding critical server messages, such as when a license is reaching its resource limits.

From email address*
The sender (or FROM) email address to use when sending email notifications.

Subject prefix
The subject prefix to use when sending email notifications. This is useful for mail client filtering rules. For example: [ACME CORP - Crowd].

Mail server details

Mail server type SMTP server
 JNDI location
Choose if you want to use SMTP or JNDI for your mail configuration

JNDI location

JNDI location*
The JNDI location of a javax.mail.Session object, setup by your application server.

Select the '**JNDI Location**' if you want to connect to a mail server via a datasource managed by your application server.

Enter the details as follows:

- **Server alert email** The email address which will receive notifications about server events.
- **From Email Address** Crowd will add this email address as the 'sender' on the emails generated by Crowd and sent to users.
- **Subject Prefix** The prefix which will appear at the start of the email subject, for all emails generated by Crowd. This can be useful for email client programs that offer filtering rules.
- **Mail Server Type** Select the '**JNDI Location**' radio button.
- **JNDI Location** The datasource name of a `javax.mail.Session` object which has been set up by your application server.

Configuring the JNDI Resource

For example, in Tomcat 5.5 (the default application server that is bundled with the Crowd distribution (not EAR-WAR)), your JNDI location would be `java:comp/env/mail/CrowdMailServer`, and you would add the following section in `conf/server.xml` or `conf/Catalina/localhost/crowd.xml`, inside the `<Context>` node:

```
<Context path="/crowd" docBase="${CATALINA_HOME}/crowd-webapp" reloadable="false">
  <Resource name="mail/CrowdMailServer"
    auth="Container"
    type="javax.mail.Session"
    mail.smtp.host="yourmailserver.example.com"
    mail.smtp.port="25"
    mail.transport.protocol="smtp"
    mail.smtp.auth="true"
    mail.smtp.user="your_userid"
    password="your_password"
  />
</Context>
```

If you have problems connecting, add a `mail.debug="true"` parameter, which will let you see SMTP-level details when testing the connection.

You will also need to ensure that the [JavaMail classes](#) and [Java Beans Activation Framework](#) are present in your application server's classpath.

If JavaMail is not present in your application server installation, you will receive the following error in your log file:

```
java.lang.NoClassDefFoundError: javax/mail/Authenticator
```

If the Activation Framework is not present in your application server installation, you will receive the following error in your log file:

```
java.lang.NoClassDefFoundError: javax/activation/DataSource
```

Notes

- To customize the password notification message, see the page about [email notification templates](#).

Creating an Email Notification Template

Crowd uses an email template to build the content of an email message that Crowd sends to a user. Crowd provides the following email templates:

- **Password Resets:** A template for the email sent when an administrator [asks a user to reset their password](#) and when a user asks to [reset their own forgotten password](#).
- **Forgotten usernames:** A template for the email sent when a user [requests their forgotten username](#).

Email Template for Password Resets (Forgotten Passwords)

This is a template for the email sent when an administrator [asks a user to reset their password](#) and when a user asks to [reset their own forgotten password](#).

To edit the email template for password resets,

1. Log in to the [Crowd Administration Console](#).
2. Click the '**Administration**' tab in the top navigation bar.
3. Click '**Mail Template**' in the left-hand menu.
4. In the '**Forgotten Password Template**' text box, enter the text and macros that will form the body of the email message. Use a macro when you want to include a variable into the email text. Crowd will replace the macro with the relevant value when it sends the email. Below are the available macros and their replacement values:
 - **\$username** The username of the person who will receive the email.
 - **\$firstname** The user's first name.
 - **\$lastname** The user's last name.
 - **\$deploymenttitle** The title of your Crowd site, as defined in [Deployment Title](#).
 - **\$date** The date/time of the message event.
 - **\$resetlink** The automatically-generated URL that the user can click, allowing them to choose a new password.
Note: To ensure that the password reset URL will include the correct domain for your Crowd server, please make sure that Crowd's base url is set correctly in the [Server Settings](#) screen.
5. Click '**Update**'.

i Earlier releases of Crowd supplied the '**\$password**' macro to represent the user's new password, automatically generated by Crowd. Crowd no longer generates a new password, but instead generates a link that the user can click to choose their own new password. For backwards compatibility, if your email template contains the '\$password' macro, Crowd will now replace it with the text '**available at (link)**'. The '(link)' will be the same as now available in the '**\$resetlink**' macro.

Email Template for Forgotten Usernames

This is a template for the email sent when a user [requests their forgotten username](#).

To edit the email template for forgotten usernames



1. Log in to the [Crowd Administration Console](#).
2. Click the '**Administration**' tab in the top navigation bar.
3. Click '**Mail Template**' in the left-hand menu.
4. In the '**Forgotten Username(s) Template**' text box, enter the text and macros that will form the body of the email message. Use a macro when you want to include a variable into the email text. Crowd will replace the macro with the relevant value when it sends the email. Below are the available macros and their replacement values:
 - **\$username** The username of the person who will receive the email.
 - **\$firstname** The user's first name.
 - **\$lastname** The user's last name.
 - **\$deploymenttitle** The title of your Crowd site, as defined in [Deployment Title](#).
 - **\$date** The date/time of the message event.
 - **\$email** The email address that the user entered when requesting forgotten usernames. This is the address to which the email message is sent.
 - **\$admincontact** The email address of the Crowd administrator.
5. Click '**Update**'.

Configuring Trusted Proxy Servers

If you are running applications behind one or more proxy servers then you may find it useful to configure Crowd to trust the proxies' addresses. When a proxy server forwards an HTTP request, Crowd will recognize the request as coming from the request's originator, not from the proxy server. This is particularly useful if you want single sign-on amongst several applications running behind different proxy servers.

Configuring a trusted proxy server means that Crowd will iterate through client IP address and IP addresses in the `X-Forwarded-For` header from right to left and pick the first IP address that is not a trusted proxy. The address is then used as the client's IP address.

To configure Crowd to trust a proxy server,


1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click  > **Trusted proxy servers**.
3. Type the IP address or the host name of the proxy server. Possible values are:
 - A full IP address, e.g. 192.168.10.12 (IPv4) or 2001:db8:85a3:0:0:8a2e:370:7334 (IPv6).
 - An IPv4 subnet using wildcard notation, e.g. 192.168.*.*.
 - An IPv4 or IPv6 subnet, using CIDR notation, e.g. 192.168.10.1/16 (IPv4) or 2001:db8:85a3::/64 (IPv6). For more information, see the introduction to [CIDR notation on Wikipedia](#) and [RFC 4632](#).
 - A host name, e.g. proxy.example.org. All IP addresses bound to the given host name will be trusted.
 -  Using host names will cause DNS requests to be sent, which might affect Crowd performance.
4. Click **Add**.

Viewing Crowd's System Information

Crowd provides a useful summary of your server's system information, including:

- Time and date information
- Java version
- Location of your [Crowd Home](#) directory
- Memory usage
- Application server details
- Database information
- Server ID (see [Licensing](#) for more details)

To view your Crowd server's system information,

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click  > **System Info**.

Backing Up and Restoring Data


You can back up your Crowd data by exporting it to an XML file in the [Crowd shared directory](#). The data includes:

- Your Crowd server configuration details, including connection details for all your directories and applications.
- Any [internal directories](#) that exist.

Important Note about Crowd Backup Functionality


We recommend that you back up your data regularly, especially before any significant configuration changes and before upgrading Crowd. You should also perform regular backups of your [database](#) and your [Crowd Home directory](#).

To schedule daily backups of your Crowd data,


1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click  > **Backup**.
3. In the Schedule Backup panel, select the **Enable scheduled backups to XML** and choose the time.
4. Click **Submit**.

When scheduled backups are enabled, Crowd will create a daily backup in the `/backups` directory under your [Crowd shared directory](#). Backup files will be rotated to retain the most recent fifty daily backups.

To manually back up your Crowd data,

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click  > **Backup**.
3. In the **Manual Backup** panel, select the **Reset Domain** checkbox if the backup file will be restored onto a different server. Selecting **Reset Domain** will reset the domain to blank. (After you restore the data, you can change the domain as described in [Domain](#).)
4. Enter an appropriate **Backup File Name**. This will be the name of the XML file that Crowd will create. When the backup process has finished, you will find the backup file in the `/backups` directory under your [Crowd shared directory](#).
5. Click **Submit**.

To restore your Crowd data,

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click  > **Restore**.
3. In the **Restore File Path** field, type the path to the backup file, including the name of the XML file.
4. Click **Submit**.

Active status and backups

While Crowd allows you to locally set the active status of users in directories without proper active status support (eg. OpenLDAP), those changes will not be stored in the backup XML. When restoring from a backup, all of those users will be subsequently reactivated. Crowd versions 3.4.6, 3.5.1, and 3.6.0 come with the possibility to export connector users to the backup file, allowing you to preserve the active flag status after a restore. This option however will increase the size of the backup file and may require increasing Crowds memory limits.



For this feature to work, new backup files must be generated with the Backup connector directories option enabled.

Logging and Profiling

When troubleshooting problems with your Crowd installation, it is often useful to change the level of information provided by your Crowd server so that more information, messages and warnings are shown than usual. This page describes how to:

- Adjust the settings which affect Crowd's logging.
- Enable performance profiling. With performance profiling turned on, your system output console will show a record of the time it takes (in milliseconds) to complete each Crowd action. This will help with diagnosing performance problems. The resulting output will be large, so you should not enable it for long periods. You can see an example of performance profiling output [here](#).

Summary of the Logging Levels

Crowd uses Apache's [log4j](#) logging service. The amount of information written to the log file is determined by the logging 'level'. The type of message output at each level is as follows:

Level	Type of Message Written to the Log
DEBUG	Used to troubleshoot SSO problems only. These are low-level details that most people never need to know about. NOTE: This setting may cause user passwords to be logged.
INFO	Informational messages about what Crowd is doing. Usually not interesting.
WARN	Warnings that something may have gone wrong, or other messages a system administrator may wish to know. These are conditions that, while not errors in themselves, may indicate that the system is running sub-optimally.
ERROR	Indications that something has gone wrong in Crowd. The person responsible for configuring Crowd should be notified.
FATAL	Indications that something has gone wrong so badly that the system cannot recover.
ALL	All possible log messages.

Finding the Crowd Log File

When you report a problem to Atlassian Support, we may ask you to send us your `atlassian-crowd.log` file. The location of the log file may vary, depending on your Crowd installation type. Provided that you have not changed the log file location from the default, the Crowd log file is at the location described below.

Installation Type	Location of Log File
Crowd Standalone edition	Crowd 2.0.3 and older versions: In the root directory of your Crowd application, e.g. <code>atlassian-crowd-2.0.0/atlassian-crowd.log</code> Crowd 2.0.4 and newer versions: In the Crowd application Home Directory, e.g. <code>Crowd-Home-Directory/logs/atlassian-crowd.log</code>
Crowd Standalone running as a Windows service	<code>C:\Windows\system32\atlassian-crowd.log</code>

Changing the Log Settings

You can change the log settings in two ways:

- Set the logging levels at runtime via the Administration Console, as described [immediately below](#). Your changes will be in effect only until you next restart Crowd.

- Edit the log configuration file, as described in the [Advanced](#) section below. Your changes will take effect next time you start Crowd, and for all subsequent sessions.

Crowd Data Center

For Crowd Data Center, you'll need to change the log settings on each node.

Configuring the Log Settings and Performance Profiling via the Administration Console

If necessary, you can edit the configuration file directly

If you change the log settings via the Administration Console, the changes are not written to the `log4j.properties` file and are therefore discarded when you next stop Crowd. Also, not all logging behavior can be changed via the Administration Console. For logging configuration not mentioned below, or to change the log settings permanently, you will need to stop Crowd and then [edit the log configuration file](#) instead.

The '**Logging & Profiling**' screen tells you whether performance profiling is currently on or off, and shows a list of all currently defined loggers. On this screen you can:

- Turn **performance profiling** on or off. With performance profiling turned on, your system output console will show a record of the time it takes (in milliseconds) to complete each Crowd action. This will help with diagnosing performance problems. The resulting output will be large, so you should not enable it for long periods. You can see an example of performance profiling output [here](#).
- Set the **logging level** for each class or package name, or reset all logging levels to the default setting. Refer to the section on logging levels [above](#). Any changes made in this way will apply only to the currently-running Crowd lifetime.

To configure profiling and logging,

1. Log in to the [Crowd Administration Console](#).

2. In the top navigation bar, click  > **Logging & Profiling**.

The **Logging and Profiling** screen appears, as shown below. The screen has the following sections:

- **Performance Profiling** Click the **Enable Profiling** button to turn profiling on, or **Disable Profiling** to turn it off. (You will only see one of these buttons.)
 - **Log4j Logging** This section shows the loggers currently in action for your Crowd instance.
 - You can change the logging level by selecting a value from the **New Level** dropdown list. [Above](#) is a definition of each level. You can also read the [Apache documentation](#) for more information.
 - You can click the **Revert to Default** button if you want to reset the logging levels to the values shipped with your Crowd installation.
3. Click the **Update Logging** button to save any changes you have made in the **Log4j Logging** section.

Description of the loggers:

Logger	Description
<code>com.atlassian.crowd</code>	This is the parent of the crowd package loggers. Any children which do not have a level assigned to them will inherit the level from their parent. This logger should be set to DEBUG only if you are investigating SSO issues.

<code>com.atlassian.crowd...XFireFaultLoggingMethodHandler</code>	Can be helpful if a Crowd SOAP service fault is thrown. It is best to enable DEBUG for all three XFire classes simultaneously when troubleshooting Crowd's SOAP service.
<code>com.atlassian.crowd...XFireOutLoggingMethodHandler</code>	The Crowd server outputs the incoming SOAP request method and parameters. This is useful when debugging your applications or monitoring the level of traffic for an integrated application.
<code>com.atlassian.crowd...XFireInLoggingMethodHandler</code>	The Crowd server outputs the outgoing SOAP request method and parameters. This is useful when debugging your applications or monitoring the level of traffic for an integrated application.
<code>com.atlassian.crowd.license</code>	Useful for troubleshooting certain licensing issues in Crowd.
<code>com.atlassian.crowd.startup</code>	Can be helpful for troubleshooting startup errors in Crowd.
<code>root</code>	This is the root of the logger hierarchy, i.e. it is the parent of all loggers. The level assigned to the root will be the default level for any loggers which do not have a specific level and do not inherit from another parent.

Advanced Log Configuration

Terminology: In log4j, a 'logger' is a named entity. Logger names are case sensitive and follow a hierarchical naming standard. For example, the logger named `com.foo` is a parent of the logger named `com.foo.Bar`.

Finding the Log Configuration File

Crowd's logging behavior is defined in the following properties file:

- For [Crowd installations](#): `{CROWD-INSTALL}/crowd-webapp/WEB-INF/classes/log4j.properties`

This file is a standard log4j configuration file, as described in the [Apache log4j documentation](#).

Editing the Log Configuration File

To configure the logging levels and other settings on a permanent basis:

1. Stop Crowd.
2. With a text editor, open the `log4j.properties` file in the location described [above](#).
3. Adjust the output level to the required level of importance listed in the section on levels [above](#).
4. Save the `log4j.properties` file.
5. Restart Crowd to have the new log settings take effect.

When diagnosing a server problem you need to adjust Crowd's package logging to:
`log4j.logger.com.atlassian.crowd=DEBUG`

Changing the Destination of the Crowd Log File

Terminology: In log4j, an output destination is called an 'appender'.

To change the destination of the Crowd log file:

1. Stop Crowd.

2. With a text editor, open the `log4j.properties` file in the location described [above](#).
3. Look for the `org.apache.log4j.RollingFileAppender` entry in the '**Log File Locations**' section of the file. This appender controls the default logging destination described [above](#).
4. Edit the following line, and replace `atlassian-crowd.log` with the full path and file name for the required logging destination:
`log4j.appender.filelog.File=atlassian-crowd.log.`
5. Save the `log4j.properties` file.
6. Restart Crowd to have the new log settings take effect.

Note: If you change the location of your log files, they will no longer be included when you generate a support zip. This means you'll need to attach your logs to any support requests manually.

Adjusting the Log Settings for CrowdID

The Crowd Administration Console does not give access to the **CrowdID** log settings. To adjust the logging levels of the CrowdID OpenID server, you will need to modify the configuration file at this location:

- For [Crowd installations](#) of CrowdID: `{CROWDID-INSTALL}/crowd-openidserver-webapp/WEB-INF/classes/log4j.properties`

Performance Profiling

When troubleshooting problems with your Crowd installation, it is often useful to turn on performance profiling.

To enable profiling, in the top navigation bar, go to  > **Logging & profiling**. Full instructions are in the section on [logging and profiling](#).

With performance profiling turned on, your system output console will show a record of the time it takes (in milliseconds) to complete each Crowd action. This will help with diagnosing performance problems. The resulting output will be large, so you should not enable it for long periods. Here is an example of the performance profiling output, when search for and viewing a user via the Crowd Administration Console:

```
[15ms] - AOP: SecurityServer.findPrincipalByToken()  
  
[15ms] - AOP: SecurityServer.isValidPrincipalToken()  
  
[15ms] - AOP: SecurityServer.isValidPrincipalToken()  
  [15ms] - AOP: SOAPService.validateSOAPService()  
  
[15ms] - AOP: SecurityServer.isValidPrincipalToken()  
  
[16ms] - AOP: SecurityServer.getDomain()  
  [16ms] - AOP: SOAPService.validateSOAPService()
```

Draft - Troubleshooting and Requesting Technical Support

This page is visible to **Atlassian staff only**. It will be published when the functionality is available in Crowd.

⚠ Page must be updated if ever published

If we ever publish this page, we must update it to bring it in line with the new [Support Policies](#). And add a link from [Creating a Support Issue](#)

This document tells you how to troubleshoot problems and obtain technical support for Crowd.

On this page:

- [Troubleshooting a Problem](#)
- [Raising a Support Request](#)
 - [Raising a Support Ticket via the Crowd Support Request Form](#)
 - [Raising a Support Ticket via the Internet](#)
- [Logging a Bug Report](#)

Troubleshooting a Problem

If you have a problem with Crowd, please follow these steps:

1. If you are not a Crowd administrator, report your problem to the person in charge of your Crowd installation and ask them to follow up on the issue.
2. Check Crowd's [system requirements](#).
3. Search our [Knowledge Base](#) for a solution to your problem.
4. If you are having problems configuring Crowd, please take a look at the appropriate guides:
 - [Installation Guide](#)
 - [Setup Guide](#)
 - [Configuration Guide](#)
 - [Administration Guide](#)

If the above documentation does not solve your problem, you should create a support request. If you believe you have found a bug, you may wish to create a bug report instead. This page contains instructions for both:

- [Support requests](#)
- [Bug reports](#)

Raising a Support Request

i Plugin support

If you have a plugin-related issue, please check whether the plugin is supported by Atlassian. Visit the plugin's home page in the [Atlassian Plugin Exchange](#) and check the support details in the '**Plugin Details**' panel. If the plugin is not supported by Atlassian, you will need to contact the plugin author directly.

There are two ways to raise a support request with Atlassian:

- Complete the support request form via your Crowd **Administration Console**. The advantage of this method is that Crowd will create the support ticket and attach the relevant system information and logs for you. See [below](#).
- Raise a support ticket directly via our internet support site. See [below](#).

Raising a Support Ticket via the Crowd Support Request Form

i This method is recommended, provided that [SMTP email](#) is set up on your Crowd instance.

The advantage of this method is that Crowd will create the support ticket and attach the relevant system information and logs for you. You can also use this method to append system information to an existing support ticket.

To raise a support request via the Crowd Administration Console,

1. Log in to the [Crowd Administration Console](#).
2. Click the **'Administration'** tab in the top navigation bar.
3. Click **'Support Request'** in the left-hand menu. The **'Raise Support Request'** form will appear.
4. Please provide as much information as possible, following these guidelines:
 - **'To'** This is an email address, named the 'Site Support Address' and configured on the **'General Configuration'** screen of your Crowd instance. It points to a [JIRA](#) instance (usually the [Atlassian Support System](#)) which is configured to receive and handle support requests by email.
 - **'CC'** Any email address(es) entered here will receive a copy of the support request, including all system information. You can enter more than one email address, separated by commas (e.g. joe@mycompany.com, sally@mycompany.com, jane@myothercompany.com).
 - **'Subject'** Enter a short and meaningful description of the problem.
 - **'Description'** Please enter as much information as possible, including any error messages that are appearing and any steps the support team can take to reproduce the problem.
 - **'Existing Support Request'** If you have previously raised a support request for the problem, please type the issue key here (e.g. CSP-12345). The information on this form will be appended to the existing support ticket.
 - **'Contact Name'** This will default to the name of the logged-in user.
 - **'Contact Email'** This will default to the email address of the logged-in user.
Note: This email address will be used to find your support account on the [Atlassian Support System](#). If no matching account is found, a new account will be created. Crowd will also send all further notifications and updates to this address.
 - **'Contact Phone Number'** Please enter a telephone number where our support staff can reach you. Include international and city codes.
5. Click the **'Send'** button.
6. Crowd will submit your request via email to the JIRA instance referenced by the 'To' email address on the form. If you do not already have a support account, Crowd will automatically request one for you. The submitted request will include all the system and environment information which you see on the support request form. It will also include a zipped copy of your Crowd log file. Refer to [Logging and Profiling](#) for information about the log files. JIRA will create a support ticket including the submitted information.

i Log files can be very big it is possible that your email server may bounce the message if too large. With the default log4j configuration, the log file could be up to 20Mb in size. If you have customized the log settings, the maximum size could be much larger still. Please check whether the email message has been successfully sent, and consult your email administrator if you need special provisions for this email message.
7. Once you have submitted your support request, you will receive email updates about its progress. These emails will give you the support ticket number.

i You can view the status of your support request and add any additional information required by visiting the [Atlassian Support System](#) at any time.

Raising a Support Ticket via the Internet

If your Crowd instance is not configured with SMTP mail or your Crowd instance is not running, you can raise a support ticket via the [Atlassian Support System](#):

1. Create a zip of your Crowd logs to attach to the ticket. Refer to [Logging and Profiling](#) for information about the log files.
2. If you do not already have a free Atlassian support account, [create one here](#).
3. Log in to <https://support.atlassian.com> and select **'Create New Issue'**.
4. Lodge a detailed description of your problem in the new support ticket.
5. Fill in all applicable information about your system, such as application server, database, etc.

6. If Crowd is running, go to the '**System Information**' screen in your **Administration Console** and copy the text of your system information into the ticket.
7. Once your ticket is lodged, wait to be notified by email of updates. If your production instance of Crowd is experiencing a critical problem, jump on [Live Support](#) and ask to have your issue reviewed immediately.

Logging a Bug Report

If you have found a bug, the easiest way to report it is to:

- Create numbered instructions on how to reproduce the bug.
- Log them as a [support request](#).
The Atlassian support team will confirm your bug and lodge a bug report.

Alternatively, you can log a bug report directly by confirming it according to these instructions:

1. Visit the [Crowd bug tracker](#).
2. On the left under '**Text Search**', type keywords for your problem into the '**Query**' field.
3. Click '**View**' and browse the summaries of the unresolved bugs. If any summary appears to describe your problem, check whether the bug matches yours. If it is the same, you may wish to set a watch to be notified of updates or apply your vote towards having it resolved.
4. If your bug has not been reported already, log the new bug [here](#) along with the information you used to duplicate it.

RELATED TOPICS

- [Configuring Server Settings](#)
 - [Deployment Title](#)
 - [Domain](#)
 - [Session configuration](#)
 - [Authorization Caching](#)
 - [Licensing](#)
 - [Crowd SSO 2.0](#)
 - [Finding your SEN](#)
 - [SSO Cookie](#)
- [Configuring your Mail Server](#)
- [Creating an Email Notification Template](#)
- [Configuring Trusted Proxy Servers](#)
- [Viewing Crowd's System Information](#)
- [Backing Up and Restoring Data](#)
- [Logging and Profiling](#)
 - [Performance Profiling](#)
- [Draft - Troubleshooting and Requesting Technical Support](#)
- [Configuring the LDAP Connection Pool](#)
- [Browsing the audit log](#)
- [Look and feel](#)
- [Overview of Caching](#)

[Crowd documentation](#)

Configuring the LDAP Connection Pool

When connection pooling is enabled, the LDAP service provider maintains a pool of connections and assigns them as needed. When a connection is closed, LDAP returns the connection to the pool for future use. This can improve performance significantly.

This page describes the site-wide settings for LDAP connection pooling in Crowd.

To configure the LDAP connection pooling in Crowd,

1. Log in to the [Crowd Administration Console](#).
2. Click the '**Administration**' tab in the top navigation bar.
3. Click '**LDAP Connection Pool**' in the left-hand menu.
4. The '**LDAP Connection Pool**' screen appears. Enter the details for each setting, as described in the table below.
5. Click the '**Update**' button.
6. Restart Crowd to put the changes into effect.

Connection Pool Setting	Description	Default Value
Initial Pool Size	The number of LDAP connections created when initially connecting to the pool.	1
Preferred Pool Size	The optimal pool size. LDAP will remove idle connections when the number of connections grows larger than this value. A value of 0 (zero) means that there is no preferred size, so the number of idle connections is unlimited.	10
Maximum Pool Size	The maximum number of connections. When the number of connections reaches this value, LDAP will refuse further connections. As a result, requests made by an application to the LDAP server will be blocked. A value of 0 (zero) means that the number of connections is unlimited.	0
Pool Timeout	The length of time, in seconds, that a connection may remain idle before being removed from the pool. When the application is finished with a pooled connection, the connection is marked as idle, waiting to be reused. A value of 0 (zero) means that the idle time is unlimited, so connections will never be timed out.	30
Pool Protocol	Only these protocol types are allowed to connect to LDAP. If you want to allow multiple protocols, enter the values separated by a space. Valid values are: <ul style="list-style-type: none">• plain• ssl	plain ssl (Both plain and ssl)
Pool Authentication	Only these authentication types are allowed to connect to LDAP. If you want to allow multiple authentication types, enter the values separated by a space. See RFC 2829 for details of LDAP authentication methods. Valid values are: <ul style="list-style-type: none">• none• simple• DIGEST-MD5	simple

Screenshot: LDAP Connection Pool

LDAP Connection Pool

You can configure the settings used for pooling of LDAP server connections below. These settings are system wide and will be used to create a new connection pool for each configured LDAP server.

Current Settings

Initial Pool Size:	1
Preferred Pool Size:	10
Maximum Pool Size:	0
Pool Timeout (seconds):	30
Pool Protocol:	plain ssl
Pool Authentication:	simple

Update Settings

Changes to these settings will not be active until the server has been restarted.

Initial Pool Size:	<input style="width: 90%;" type="text" value="1"/>	<small>Number of connections to create when initially connecting to the pool.</small>
Preferred Pool Size:	<input style="width: 90%;" type="text" value="10"/>	<small>Idle connections will be removed from the pool if the pool is larger than the preferred size. Value of 0 means there is no preferred pool size.</small>
Maximum Pool Size:	<input style="width: 90%;" type="text" value="0"/>	<small>Maximum number of connections to the LDAP server. Value of 0 means no maximum. Note that requests will block if there is available connection.</small>
Pool Timeout (seconds):	<input style="width: 90%;" type="text" value="30"/>	<small>Idle time for a connection before it is removed from the pool. Value of 0 means there is no timeout.</small>
Pool Protocol:	<input style="width: 90%;" type="text" value="plain ssl"/>	<small>Only connections with the specified protocol types will be allowed. Valid types are: plain, ssl.</small>
Pool Authentication:	<input style="width: 90%;" type="text" value="simple"/>	<small>Only connections with the specified authentication types will be allowed. Valid types are: none, simple, DIGEST-MD5.</small>

RELATED TOPICS

- [Configuring Server Settings](#)
 - [Deployment Title](#)
 - [Domain](#)
 - [Session configuration](#)
 - [Authorization Caching](#)
 - [Licensing](#)
 - [Crowd SSO 2.0](#)
 - [Finding your SEN](#)
 - [SSO Cookie](#)
- [Configuring your Mail Server](#)
- [Creating an Email Notification Template](#)
- [Configuring Trusted Proxy Servers](#)
- [Viewing Crowd's System Information](#)

- [Backing Up and Restoring Data](#)
- [Logging and Profiling](#)
 - [Performance Profiling](#)
- [Draft - Troubleshooting and Requesting Technical Support](#)
- [Configuring the LDAP Connection Pool](#)
- [Browsing the audit log](#)
- [Look and feel](#)
- [Overview of Caching](#)

[Crowd documentation](#)

Browsing the audit log

All key activities in Crowd are tracked and recorded in the audit log. The audit log helps you diagnose problems, and verify important actions for security purposes. Auditing is enabled by default.

About auditing in Crowd

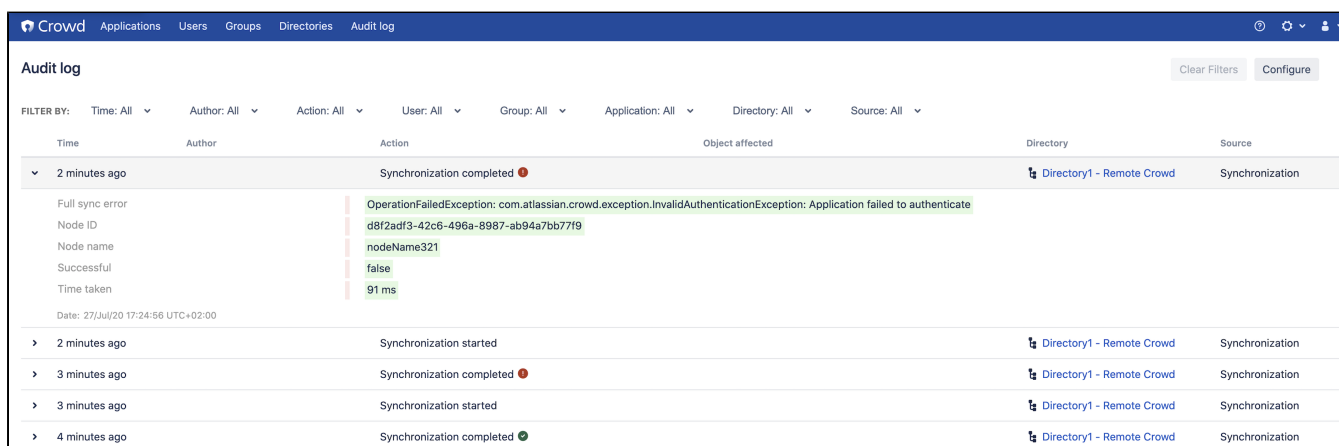
The audit log doesn't record all activities in Crowd. It logs configuration changes that can affect security or the setup of Crowd.

What information Crowd records?	What information is NOT recorded by Crowd?
<ul style="list-style-type: none">• changes to the system configuration of Crowd• adding or removing applications, updating application configuration• adding or removing directories, updating directory configuration• creating, modifying or removing users and groups• adding or removing group memberships• start and end of a synchronization with a remote user directory• start and end of a data import	<ul style="list-style-type: none">• application access• user login failures

Viewing the audit log

To view the audit log:

1. Log in to the [Crowd Administration Console](#).
2. In the top navigation bar, click **Audit log**.



Modifying the audit log's retention period

You can choose how long the recorded activities are stored in Crowd.

1. Log in to the [Crowd Administration Console](#).
2. Click the '**Audit log**' link in the top navigation bar.
3. Click the '**Configure**' button.
4. Choose the retention period.

Auditing and the REST API

The audit log can also be accessed via the REST API. You can use it to:

- Search the recorded activities
- Export the audit log
- Add events from external applications to the audit log

For more information about using the REST API, see [Crowd REST documentation](#).


Look and feel

You can use the configuration panel to customize you login screen.

Before you begin

Only Crowd administrators can make changes to the look and feel of the login screen.

To make changes to the login screen

1. Go to  > **Look and feel**.
2. Make changes the following login screen elements:
 - logo - the image displayed at the top of the login screen; supported file formats: JPG, PNG.
 - welcome text - the text displayed above the credentials.
 - accent color - the color of the button and the the **Remember me** checkbox.
3. Click **Save**.

The login screen with your changes will be displayed to logging users from now on. If you want to go back to the default settings, click the **Restore defaults** button next to the **Save** button.

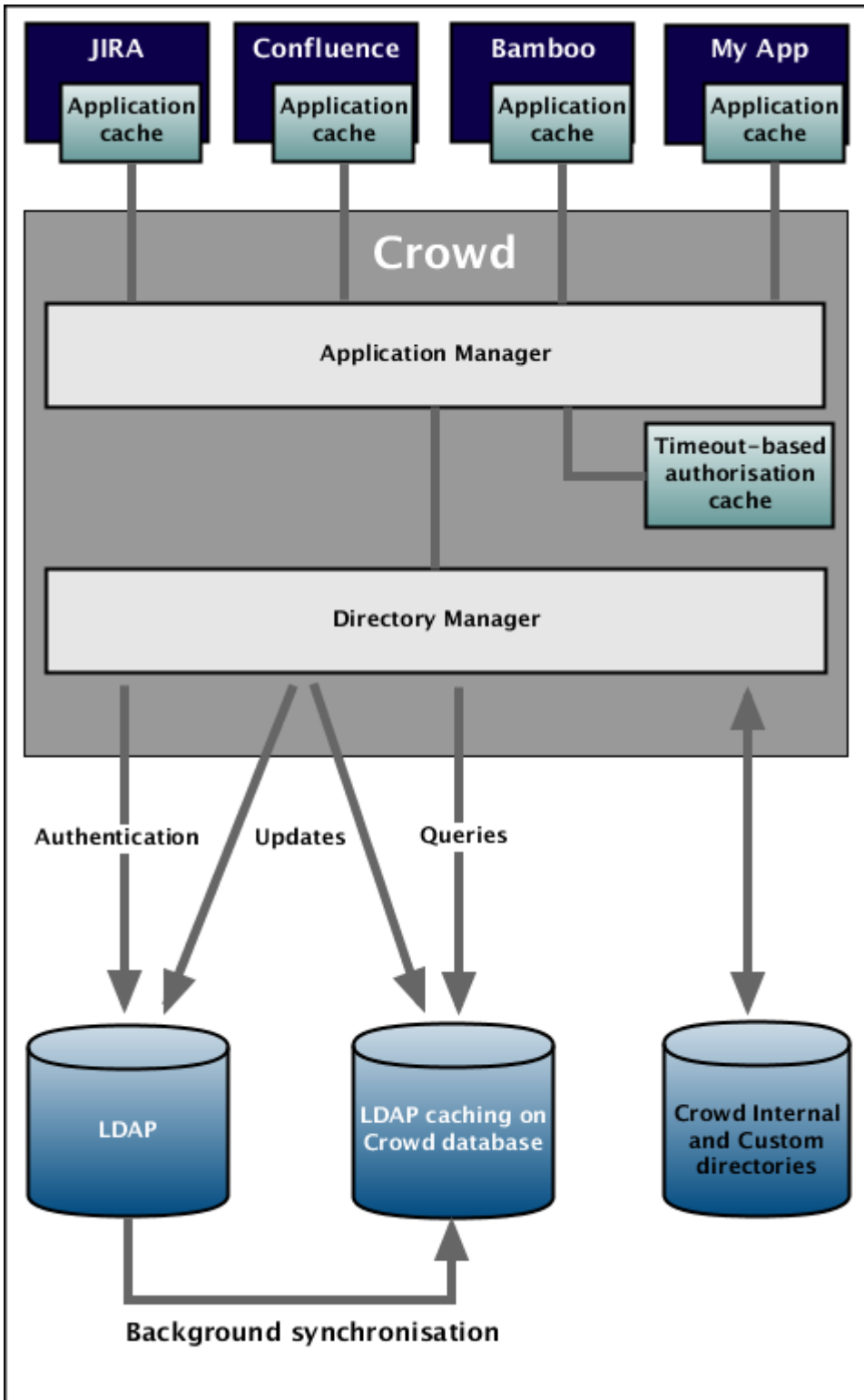
Overview of Caching

Caching is used to store run-time authentication and authorization rules, which can be expensive to calculate.

In Crowd, data caching occurs in three main areas:

- **Application caches in the applications that are connected to Crowd** Applications such as Jira, Confluence and Bamboo can store user, group and role data in a local cache. This helps improve the performance of Crowd, since these applications do not have to repeatedly request information from Crowd. Generally it is not necessary to configure application caching, although this depends on the size of your application deployments. You can set the options for application caching in the cache configuration file for that application. See [Configuring Caching for an Application](#).
- **An authorization cache on the Crowd server** To improve performance, Crowd can store users' authentication and per-application permissions in a local cache for a specified period. You can enable or disable this cache via an option on the 'General Options' screen in the Crowd Administration Console. See [Authorization Caching](#).
- **LDAP directory caches in the Crowd database** The Crowd database keeps an up-to-date cache of all user and group information from the LDAP directory. You can configure this cache on the directory connector screen. See [Configuring Caching for an LDAP Directory](#).

This diagram gives a conceptual overview of the caches described above:



Crowd Security Advisories and Fixes

This page has information on how to report any security bugs you might find in Crowd, and what we will do to fix the problem and announce the solution.

On this page:

- [Finding and Reporting a Security Vulnerability](#)
- [Publication of Security Advisories](#)
- [Severity Levels](#)
- [Patches and Fixes](#)
- [Published Security Advisories](#)

Finding and Reporting a Security Vulnerability

Atlassian's approach to reporting security vulnerabilities is detailed in [How to Report a Security Issue](#).

Publication of Security Advisories

Atlassian's approach to releasing security advisories is detailed in [Security Advisory Publishing Policy](#).

Severity Levels

Atlassian's approach to ranking security issues is detailed in [Severity Levels for Security Issues](#).

Patches and Fixes

Atlassian's approach to releasing patches for security issues is detailed in [Security Patch Policy](#).

Published Security Advisories

- [Crowd Security Advisory 2019-05-22](#)
- [Crowd Security Advisory 2017-03-10](#)
- [Crowd Security Advisory 2016-10-19](#)
- [Crowd Security Advisory 2014-05-21](#)
- [Crowd Security Advisory 2013-07-16](#)
- [Crowd Security Notice 2013-07-01](#)
- [Crowd Security Advisory 2012-05-17](#)
- [Crowd Security Advisory 2010-07-05](#)
- [Crowd Security Advisory 2010-05-04](#)
- [Crowd Security Advisory 2008-10-14 - Parameter Injection Vulnerability](#)

Crowd Security Advisory 2010-07-05

This advisory announces a security vulnerability in earlier versions of Crowd that we have found and fixed in [Crowd 2.0.5](#).

In this advisory:

- [XSS Vulnerability](#)
 - [Severity](#)
 - [Risk Assessment](#)
 - [Vulnerability](#)
 - [Risk Mitigation](#)
 - [Fix](#)

XSS Vulnerability

Severity

Atlassian rates the severity level of this vulnerability as **high**, according to the scale published in [Severity Levels for Security Issues](#). The scale allows us to rank the severity as critical, high, moderate or low.

Risk Assessment

We have identified and fixed a cross-site scripting (XSS) vulnerability that may affect Crowd instances in a public environment. This vulnerability may allow an attacker to embed their own JavaScript into the Crowd login page. An attacker's text and script might be displayed to other people viewing the page. This is potentially damaging to your company's reputation.

You can read more about XSS attacks at [cgisecurity](#), [CERT](#) and other places on the web.

Vulnerability

The Crowd login form may be vulnerable to XSS attacks. This vulnerability is tracked in [CWD-1952](#).

This vulnerability exists in **all versions of Crowd** up to and including Crowd 2.0.4.

Risk Mitigation

To address the issue, we recommend that you upgrade Crowd. If you cannot upgrade immediately, you can fix the XSS vulnerability by editing your configuration to disallow request parameters in generated URLs. Details are below.

Alternatively, if you are not in a position to upgrade or edit your configuration immediately, you should configure your firewall to block Internet access to Crowd.

Fix

Crowd 2.0.5 fixes the security flaw and other bugs. See the [release notes](#). You can download Crowd 2.0.5 from the [download centre](#).

If you cannot upgrade immediately, you can fix this XSS vulnerability by disallowing request parameters in generated URLs. You can globally turn off the inclusion of request parameters in generated URLs by editing your WebWork properties file:

1. Edit the `webwork.properties` file located at `{CROWD-INSTALLATION-DIRECTORY}\crowd-webapp\WEB-INF\classes\webwork.properties`.
2. Add the following property as a new line in the file:

```
webwork.url.includeParams=none
```

3. Save the file.
4. Restart Crowd.

The WebWork documentation has more about the [webwork.properties](#) file.

Crowd Security Advisory 2010-05-04

This advisory announces a number of security vulnerabilities in earlier versions of Crowd that we have found and fixed in [Crowd 2.0.4](#). In addition to releasing Crowd 2.0.4, we also provide point releases for earlier versions of Crowd to fix the vulnerabilities reported here.

In this advisory:

- [XSS Vulnerabilities](#)
 - [Severity](#)
 - [Risk Assessment](#)
 - [Vulnerability](#)
 - [Risk Mitigation](#)
 - [Fix](#)

XSS Vulnerabilities

Severity

Atlassian rates these vulnerabilities as **high**, according to the scale published in [Severity Levels for Security Issues](#). The scale allows us to rank a vulnerability as critical, high, moderate or low.

Risk Assessment

We have identified and fixed a number of cross-site scripting (XSS) vulnerabilities which may affect Crowd instances in a public environment.

- An attacker might take advantage of the vulnerability to steal other users' session cookies or other credentials, by sending the credentials back to such an attacker's own web server.
- An attacker's text and script might be displayed to other people viewing the Crowd page. This is potentially damaging to your company's reputation.

You can read more about XSS attacks at [cgisecurity](#), [CERT](#) and other places on the web.

Vulnerability

The table below lists the affected areas of Crowd. These XSS vulnerabilities exist in **all versions of Crowd**, up to and including Crowd 2.0.3.

Crowd Feature	Issue Tracking
Crowd Administration Console	CWD-1888
Error page	CWD-1889

Risk Mitigation

To address the issues, you should upgrade Crowd as soon as possible. If you cannot upgrade immediately, you should configure your firewall to block Internet access to Crowd.

Fix

Crowd 2.0.4 fixes all of these issues and introduces some nice improvements too. See the [release notes](#). You can download Crowd 2.0.4 from the [download centre](#).

If you cannot upgrade to Crowd 2.0.4, please download the relevant upgrade file for your version of Crowd from the [download centre](#):

- If you have Crowd 1.6.x upgrade to **Crowd 1.6.3** (see the [release notes](#) and [upgrade guide](#)).
- If you have Crowd 1.5.x upgrade to **Crowd 1.5.3** (see the [release notes](#) and [upgrade guide](#)).
- If you have Crowd 1.4.x upgrade to **Crowd 1.4.8** (see the [release notes](#) and [upgrade guide](#)).

Crowd Security Advisory 2008-10-14 - Parameter Injection Vulnerability

In this advisory:

- [Parameter Injection Vulnerability in Crowd](#)
 - [Severity](#)
 - [Risk Assessment](#)
 - [Risk Mitigation](#)
 - [Vulnerability](#)
 - [Fix](#)

Parameter Injection Vulnerability in Crowd

Severity

Atlassian rates this vulnerability as **critical**, according to the scale published in [Crowd Security Advisories and Fixes](#). The scale allows us to rank a vulnerability as critical, high, moderate or low.

Risk Assessment

We have identified and fixed a flaw which would allow a malicious user (hacker) to inject their own values into a Crowd request by adding parameters to the URL string. This would allow a hacker to bypass Crowd's security checks and perform actions that they are not authorised to perform.

Risk Mitigation

To address the issue, you should upgrade Crowd as soon as possible. Please follow the instructions in the 'Fix' section below. If you judge it necessary, you can block all untrusted IP addresses from accessing Crowd.

Vulnerability

A hacker can design a URL string containing parameters which perform specific actions on the Crowd server, bypassing Crowd's security checks. This is because Crowd does not adequately sanitise user input before applying it as an action on the server.

Exploiting this issue could allow an attacker to access or modify data and compromise the Crowd application.

The following Crowd versions are vulnerable: All versions from **1.0 to 1.5.0** inclusive.

Fix

Please download the relevant upgrade file for your version of Crowd from the [download centre](#) as follows:

- If you have Crowd 1.5.0 upgrade to **Crowd 1.5.1** (see the [release notes](#) and [upgrade guide](#)).
- If you have Crowd 1.4.x upgrade to **Crowd 1.4.7** (see the [release notes](#) and [upgrade guide](#)).
- If you have Crowd 1.3.x upgrade to **Crowd 1.3.3** (see the [release notes](#) and [upgrade guide](#)).
- If you have Crowd 1.2.x upgrade to **Crowd 1.2.4** (see the [release notes](#) and [upgrade guide](#)).

Crowd Security Advisory 2012-05-17

This advisory discloses a **critical** security vulnerability that exists in all versions of Crowd up to and including 2.4.0. Customers should upgrade their existing Crowd installations to fix this vulnerability. We also provide a patch that you will be able to apply to existing installations of Crowd to fix this vulnerability. However, we recommend that you upgrade your complete Crowd installation rather than applying the patch.

Our thanks to Will Caput and Trevor Hartman who reported the vulnerability in this advisory. Atlassian is committed to improving product security. We [fully support the reporting of vulnerabilities](#) and we appreciate it when people work with us to identify and solve the problem.

If you have questions or concerns regarding this advisory, please raise a support request at <http://support.atlassian.com/>.

In this advisory:

- [Critical XML Parsing Vulnerability](#)
 - [Severity](#)
 - [Description](#)
 - [Risk Mitigation](#)
 - [Fix](#)

Critical XML Parsing Vulnerability

Severity

Atlassian rates the severity level of this vulnerability as **critical**, according to the scale published in [Severity Levels for Security Issues](#). The scale allows us to rank the severity as critical, high, moderate or low.


This is an independent assessment and you should evaluate its applicability to your own IT environment.

Description

We have identified and fixed a vulnerability in Crowd that results from the way third-party XML parsers are used in Crowd.

This vulnerability allows an attacker to:

- execute denial of service attacks against the Crowd server, or
- read all local files readable to the system user under which Crowd runs.

All versions of Crowd **up to and including 2.4.0** are affected by this vulnerability. This issue can be tracked here:  [GWD-2797](#) - XML Vulnerability in Crowd CLOSED

Risk Mitigation

We recommend that you upgrade your Crowd installation to fix this vulnerability.

Alternatively, if you are not in a position to upgrade or apply patches immediately, you should do **all** of the following until you can upgrade or patch. Please note, these measures will only limit the impact of the vulnerability, they will not mitigate it completely.

- Ensure that Crowd URLs cannot be reached from untrusted sources, e.g. configure appropriate firewall or proxy settings.
- Ensure that the operating system user under which Crowd process runs is restricted.

Fix

Upgrade (recommended)

Upgrade to Crowd 2.4.1 or later which fixes this vulnerability. For a full description of this release, see the [Crowd 2.4.1 Release Notes](#). The following releases have also been made available to fix these issues in older Crowd versions. You can download these versions of Crowd from the [download centre](#).

- 2.3.7 for Crowd 2.3
- 2.2.9 for Crowd 2.2
- 2.1.2 for Crowd 2.1
- 2.0.9 for Crowd 2.0

Patches (not recommended)

We recommend patching only when you can neither upgrade nor apply external security controls. Patches are usually only provided for vulnerabilities of critical severity (as per our [Security Patch Policy](#)), as an interim solution until you can upgrade. You should not expect that you can continue patching your system instead of upgrading. Our patches are often non-cumulative we do not recommend that you apply multiple patches from different advisories on top of each other, but strongly recommend upgrading to the most recent version regularly.

If for some reason you cannot upgrade to the latest version of Crowd, you must apply the patch provided for the relevant version of Crowd below to fix the vulnerability described in this advisory.

1. Download the patch file for your version of Crowd. Note, the patches are only available for the point release indicated. If you are using an earlier point release for a major version, you must upgrade to the latest point release first.

Version	Patch
Crowd 2.4.0	patch-CWD-2797-2.4.0.zip
Crowd 2.3.6	patch-CWD-2797-2.3.6.zip

2. Unzip the patch file to the `atlassian-crowd-x.x.x` (where `x.x.x` is the Crowd version) directory, overwriting the existing files.

Crowd Security Notice 2013-07-01

On 30th of June 2013, an article was uploaded to Slashdot regarding two vulnerabilities in Atlassian Crowd. We had already identified and fixed the first vulnerability, which affects only standalone Crowd servers and which the author had labeled CVE-2013-3925. Patches and updated packages are available at <https://jira.atlassian.com/browse/CWD-3366>.

We have been unable to substantiate the existence of the second alleged vulnerability. The author of the article has not contacted Atlassian and has provided no details to us, making it difficult to validate the claim.

While we have been unable to confirm the existence of the second vulnerability, designated CVE-2013-3926, we are taking it seriously and have reached out to the author directly for more details. If we can confirm that there is a vulnerability, a patch will be issued and all Crowd customers will be emailed details on how to update.

Crowd Security Advisory 2013-07-16

This advisory discloses security vulnerabilities that we have found in standalone Crowd server and fixed in a recent version of Crowd.

- **Customers who have downloaded and installed standalone Crowd servers** should upgrade their existing Crowd installations to fix this vulnerability.
- **Atlassian OnDemand customers** are not affected by any of the issues described in this advisory.
- No type of Crowd deployment other than **standalone servers** is impacted.

Atlassian is committed to improving product security.

The vulnerability listed in this advisory is **a vulnerability in a third party framework - Struts 2 / WebWork 2** that is used by Crowd. The vulnerability has been independently discovered by Atlassian and reported to the Struts maintainers.

More details about the underlying Struts vulnerability CVE-2013-2251 are available at [CVE database](#) and in the [Struts advisory](#).

If you have questions or concerns regarding this advisory, please raise a support request at <http://support.atlassian.com/>.

OGNL injection in WebWork 2

Severity

Atlassian rates the severity level of this vulnerability as **critical**, according to the scale published in [Severity Levels for Security Issues](#). The scale allows us to rank the severity as critical, high, moderate or low.

This is an independent assessment and you should evaluate its applicability to your own IT environment.

Description

We have fixed a vulnerability in WebWork 2, which is a part of the Struts web framework. In specific circumstances, attackers can use this vulnerability to execute Java code of their choice on systems that use these frameworks. In the case of Crowd, the attacker needs to be able to access the Crowd web interface. A valid user account is not required to exploit this vulnerability.

Customers should be advised that this affects all versions of Crowd, except OnDemand, Crowd 2.3.9, Crowd 2.4.10, Crowd 2.5.5 and Crowd 2.6.4 or later. The issue can be tracked here:

 **CWD-3430** - Webwork 2 code injection vulnerability CLOSED

Risk Mitigation

If you are unable to upgrade or patch your Crowd server: as a **temporary workaround**, you can do the following:

- Block access to all URLs on a Web Application Firewall or a reverse proxy that contain any of the following strings: "redirect:", "action:" or "redirect-action:" strings. A partial example for an nginx server is below. Note that the example only covers the "redirect:" prefix and does not account for any URL encoding that may be present.

```
location ~* ^/<path to your Crowd>/ {
    if ($args ~* "redirect:") {
        return 403;
    }
    proxy_pass http://$host.internal$request_uri;
}
```

or

- Block access to your Crowd server's web interface from untrusted networks, such as the Internet.

Fix

This vulnerability can be fixed by upgrading Crowd. There are no patches available for this vulnerability for any questions, please raise a support request at <http://support.atlassian.com/>.

The [Security Patch Policy](#) describes when and how we release security patches and security upgrades for our products.

Upgrade

The vulnerabilities and fix versions are described in the 'Description' section above.

We recommend that you upgrade to the latest version of Crowd, if possible. For a full description of the latest version of Crowd, see the [release notes](#). You can download the latest version of Crowd from the [download centre](#).

Crowd Security Advisory 2014-05-21

This advisory discloses a critical security vulnerability that we have found in Crowd and fixed in a recent version of Crowd.

- **Customers who have downloaded and installed Crowd** should upgrade their existing Crowd installations or apply the patch to fix this vulnerability.
- **Atlassian OnDemand customers** have been upgraded with the fix for the issue described in this advisory.
- No other Atlassian products are affected.

The vulnerability affects all versions of Crowd up to and including 2.7.1.

Atlassian is committed to improving product security. We [fully support the reporting of vulnerabilities](#) and we appreciate it when people work with us to identify and solve the problem.

If you have questions or concerns regarding this advisory, please raise a support request at <http://support.atlassian.com>.

ClassLoader manipulation vulnerability

Severity

Atlassian rates the severity level of this vulnerability as **critical**, according to the scale published in [Severity Levels of Security Issues](#). The scale allows us to rank the severity as critical, high, moderate or low.

This is an independent assessment and you should evaluate its applicability to your own IT environment.

Description

We have fixed a vulnerability in our fork of [Apache Struts](#). Attackers can use this vulnerability to execute Java code of their choice on systems that use these frameworks. The attacker needs to be able to access the Crowd web interface. In cases when anonymous access is enabled, a valid user account is not required to exploit this vulnerability.

We have discovered this vulnerability during our review of the recent Struts security advisories. This vulnerability is specific to Crowd.

The vulnerability affects all versions of Crowd earlier than and including 2.7. Crowd 2.5.7, 2.6.7, 2.7.2 are not vulnerable. The issue is tracked in [GWD-3904 - ClassLoader manipulation vulnerability](#) CLOSED.

Risk Mitigation.

If you are unable to upgrade your Crowd server you can do the following as a **temporary workaround**:

- Block at a reverse proxy or a firewall all requests matching the following regular expression pattern in URI parameters. Note that the example does not account for any URL encoding that may be present.

```
.*[?&](.*\.|.*|\[('|"))(c|C)lass(\.|('|"))|\[).*
```

Fix

This vulnerability can be fixed by upgrading Crowd. There are no patches available for this vulnerability.

The [Security Patch Policy](#) describes when and how we release security patches and security upgrades for our products.

Upgrading Crowd

Upgrade to Crowd 2.5.7, 2.6.7, 2.7.2, or a later version, which fixes this vulnerability. We recommend that you upgrade to the latest version of Crowd, if possible. For a full description of these releases, see the [Crowd Release Notes](#). You can download these versions of Crowd from the [download center](#).

Crowd Security Advisory 2016-10-19


Crowd - LDAP JavaObject Injectionresulting in remote code execution -CVE-2016-6496

Summary	CVE-2016-6496 -Crowd LDAP Java Object Injection
Advisory Release Date	19 Oct 201610 AM PDT (Pacific Time,-7 hours)
Product	Crowd
Affected Crowd Versions	<ul style="list-style-type: none">• 1.4.1 <= version < 2.8.8• 2.9.0 <= version < 2.9.5
Fixed Crowd versions	<ul style="list-style-type: none">• for 2.8.x, Crowd2.8.8has been released with a fix for this issue.• for 2.9.x, Crowd2.9.5has been released with a fix for this issue.• for 2.10.x, Crowd2.10.1has been released with a fix for this issue.

Summary of Vulnerability


This advisory discloses a**critical severity**security vulnerability which was introduced in version 1.4.1 of Crowd.V versions of Crowdstarting with 1.4.1before 2.8.8 (the fixed version for 2.8.x) and from 2.9.0 before 2.9.5(the fixed version for 2.9.x) are affected by this vulnerability.

 **Atlassian Cloud customers**are **not** affected by the issue described in this advisory.

 **Customers**who have downloaded and installed Crowd **>=1.4.1**less than **2.8.8** (the fixed version for 2.8.x)

Customerswho have downloaded and installed Crowd **>= 2.9.0** less than **2.9.5** (the fixed version for 2.9.x)

Please**upgrade**your Crowd installations**immediately**to fix this vulnerability.

 **JIRA Core, JIRA Software, JIRA Service Desk, Confluence, Bitbucket Server, FishEye and Crucible** installations which **do not use SSL/TLS connection** to configured LDAP server or allow users to manipulate specific attributes of an LDAP entry are affected by this issue. Atlassian rates the severity level of this vulnerability in these products as **high**. According to**Security Bug fix Policy**fixes are included in the last maintenance releases.

Customerswho have downloaded and installed JIRA **>=4.3.0**less than **7.2.1** (the fixed version for 7.2.x)

Customerswho have downloaded and installed Confluence **>=3.5.0**less than **5.10.6** (the fixed version for 5.10.x)

Customerswho have downloaded and installed Bitbucket Server **>=1.3.0**less than **4.10.0** (the fixed version for 4.10.x)

Customerswho have downloaded and installed FishEye and Crucible **>=4.0.0**less than **4.2.0** (the fixed version for 4.2.x)

We recommend to **upgrade** your installations to fix this vulnerability.

Crowd LDAP Java Object Injection (CVE-2016-6496)

Severity


Atlassian rates the severity level of this vulnerability as **critical**, according to the scale published in our [Atlassian severity levels](#). The scale allows us to rank the severity as critical, high, moderate or low.

This is an independent assessment and you should evaluate its applicability to your own IT environment.

Description

The Crowd LDAP directory connector allowed an attacker to gain remote code execution in Crowd by injecting malicious attributes in LDAP entries. To exploit this issue, attackers either need to modify an entry in an LDAP directory that Crowd is configured to use or successfully execute a [Man-in-The-Middle attack](#) between an LDAP server and Crowd. Crowd installations configured to communicate with an LDAP server using the **LDAPS** protocol with the **Secure SSL** option enabled are immune to the [Man-in-The-Middle attack](#) vector only (unless an attacker is able to obtain the private key of the SSL/TLS certificate used to secure the communication).

All versions of **Crowd** from 1.4.1 before 2.8.8 (the fixed version for 2.8.x) and from 2.9.0 before 2.9.5 (the fixed version for 2.9.x) are affected by this vulnerability. This issue can be tracked here:

 [GWD-4790](#) - CVE-2016-6496: LDAP Java Object Injection in Crowd CLOSED

Acknowledgements

We would like to credit **Alvaro Munoz** and **Alexander Miros** of HPE SecurityFortify for reporting this issue to us.

Fix

We have taken the following steps to address this issue:

1. Crowd version 2.10.1 has been released with a fix for this issue.
2. Crowd version 2.9.5 has been released with a fix for this issue.
3. Crowd version 2.8.8 has been released with a fix for this issue.
4. JIRA Core version 7.2.1 has been released with a fix for this issue
5. Confluence version 5.10.6 has been released with a fix for this issue
6. Bitbucket Server version 4.10.0 has been released with a fix for this issue
7. FishEye and Crucible version 4.2.0 has been released with a fix for this issue

What You Need to Do

Upgrade (recommended)

The vulnerabilities and fix versions are described in the description section above. Atlassian recommends that you upgrade to the latest version.

Upgrade Crowd to version 2.10.1 or higher.

If you are running **Crowd 2.9.x** and cannot upgrade to **Crowd 2.10.1** then upgrade to version **2.9.5**.

If you are running **Crowd 2.8.x** and cannot upgrade to **Crowd 2.9.5** then upgrade to version **2.8.8**.

For a full description of the latest version of Crowd, see [the release notes](#). You can download the latest version of Crowd from the [download center](#).

Upgrade JIRA Core (this is also required if you are running JIRA Software or JIRA Service Desk) to version 7.2.1 or higher.

For a full description of the latest version of JIRA Core, see [the release notes](#). You can download the latest version of JIRA from the [download center](#).

Upgrade Confluence to version 5.10.6 or higher.

For a full description of the latest version of Confluence, see [the release notes](#). You can download the latest version of Confluence from the [download center](#).

Upgrade Bitbucket Server to version 4.10.0 or higher.

For a full description of the latest version of Bitbucket Server, see [the release notes](#). You can download the latest version of Bitbucket Server from the [download center](#).

Upgrade FishEye and Crucible to version 4.2.0 or higher.

For a full description of the latest version of FishEye and Crucible, see [the release notes](#). You can download the latest version of FishEye and Crucible from the [download center](#).

Mitigation

The issue can be mitigated by restricting modifications of LDAP entries by LDAP users and configuring SSL/TLS connection between an LDAP Server and Crowd.

If you are running Crowd 2.8.x and cannot upgrade to 2.8.8 or 2.9.5 then you can follow these steps to mitigate the issue:

1. Configure the SSL/TLS connection between an LDAP server and Crowd. See [Configuring Crowd to Work with SSL](#) for information on setting up secure transport.
2. Restrict modification of LDAP entries by LDAP users. The following restrictions should be implemented:
 - The users **should not be able to add additional object classes** to their LDAP entries ("javaContainer", "javaObject", "javaNamingReference", "javaSerializedObject", "javaMarshaledObject")
 - The users **should not be able to manipulate attributes** related to storing Java objects in an LDAP directory ("javaSerializedData", "javaClassName", "javaFactory", "javaCodeBase", "javaReferenceAddress", "javaClassNames", "javaRemoteLocation")

Support

If you have questions or concerns regarding this advisory, please raise a support request at <https://support.atlassian.com/>.

References

Security Bug fix Policy	<p>As per our new policy critical security bug fixes will be back ported to major software versions for up to 12 months for JIRA and Confluence. We will release new maintenance releases for the versions covered by the new policy instead of binary patches.</p> <p>Binary patches will no longer be released.</p>
---	--

Severity Levels for security issues	Atlassian security advisories include a severity level and a CVE identifier. This severity level is based on our self-calculated CVSS score for each specific vulnerability. CVSS is an industry standard vulnerability metric. You can also learn more about CVSS at FIRST.org .
End of Life Policy	Our end of life policy varies for different products. Please refer to our EOL Policy for details.

Crowd Security Advisory 2017-03-10

Crowd -CVE-2017-5638

Summary	CVE-2017-5638 - Remote code execution in Crowd
Advisory Release Date	10 Mar 2017 10 AM PDT (Pacific Time,-7 hours)
Product	Crowd. (CrowdID is also affected but EmbeddedCrowd is not affected)
Affected Crowd Versions	<ul style="list-style-type: none">• 2.8.3 <= version < 2.9.7• 2.10.1 <= version < 2.10.3• 2.11.0 <= version < 2.11.1
CVE ID(s)	<ul style="list-style-type: none">• CVE-2017-5638

Summary of vulnerability

This advisory discloses a **critical severity** security vulnerability that was introduced in version 2.8.3 of Crowd. Versions of Crowd from 2.8.3 before 2.9.7 (the fixed version for 2.9.x), from version 2.10.1 before 2.10.3 (the fixed version for 2.10.x) and from version 2.11.0 before 2.11.1 (the fixed version for 2.11.x) are affected by this vulnerability.

 **Atlassian Cloud** instances aren't affected by the issue described on this page.

 If you have **upgraded Crowd to version 2.9.7 or 2.10.3 or 2.11.1**, you are **not affected by this issue**.

 **Customers who have downloaded and installed Crowd >= 2.8.3 and less than 2.9.7 (the fixed version for 2.9.x)**

Customers who have downloaded and installed Crowd >= 2.10.1 and less than 2.10.3 (the fixed version for 2.10.x)

Customers who have downloaded and installed Crowd >= 2.11.0 and less than 2.11.1 (the fixed version for 2.11.x)

Please **upgrade your Crowd installations immediately** to fix this vulnerability.

Remote code execution through Apache Struts 2 (CVE-2017-5638)

Severity

We rate the severity level of this vulnerability as **critical**, according to our [Atlassian severity levels](#). The scale allows us to rank the severity as critical, high, moderate or low.

This is an independent assessment. You should evaluate its applicability to your own IT environment.

Description

Crowd used a version of Apache Struts 2 that was vulnerable to [CVE-2017-5638](#). Attackers can use this vulnerability to execute Java code of their choice without prior authentication on systems that have a vulnerable version of Crowd.

All versions of Crowd from 2.8.3 before 2.9.7 (the fixed version for 2.9.x), from version 2.10.1 before 2.10.3 (the fixed version for 2.10.x) and from version 2.11.0 before 2.11.1 (the fixed version for 2.11.x) are affected by this vulnerability. You can track this issue here:

 [GWD-4879](#) - Apache Struts 2 Remote Code Execution (CVE-2017-5638) CLOSED

Fix

To address this issue, we have released the following versions containing a fix:

- Crowd version 2.9.7
- Crowd version 2.10.3
- Crowd version 2.11.1

What you need to do

Upgrade Crowd to version 2.11.1 or higher

The vulnerabilities and fix versions are described above. If affected, you should upgrade to the latest version immediately.

If you're running **Crowd 2.10.x** and cannot upgrade to 2.11.1, upgrade to version 2.10.3.

If you're running **Crowd 2.9.x** and cannot upgrade to 2.11.1, upgrade to version 2.9.7.

For a full description of the latest version of Crowd, see the [Crowd release notes](#). You can download the latest version of Crowd from the [download centre](#).

Detection

Given that we have confirmed exploitation in the wild, we recommend customers look for signs of compromise even if they upgrade immediately. Detection of such attacks is very environment-specific but below are some indicators that customers may find useful in their investigations.

The same expression can be used across all log types, which is especially useful if you are using a SIEM or log aggregator to analyze the logs all at once:

```
grep -E 'InvalidContentTypeException.+multipart/form-data' *.log
```

If remote code was executed, you will see an OGNL expression being evaluated on the server. These commands will typically start with ``#cmd=`` or ``#cmds=`` such as:


```
(#cmd='ls')
(#cmds=(#iswin?{'cmd.exe','/c',#cmd}:{'/bin/bash','-c',#cmd}))
```

Crowd logs all its application events inside `atlassian-crowd.log`. The default path is: ``<CROWD_HOME>/logs/atlassian-crowd.log``.

```
017-03-10 10:27:29,340 http-bio-8095-exec-22 WARN [struts2.dispatcher.multipart.JakartaMultiPartRequest]
Unable to parse request
org.apache.commons.fileupload.FileUploadBase$InvalidContentTypeException: the request doesn't contain a
multipart/form-data or multipart/mixed stream, content type header is %({#nike='multipart/form-data'}).
(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).(#_memberAccess?(#
_memberAccess=#dm):((#container=#context['com.opensymphony.xwork2.ActionContext.container']).
(#ognlUtil=#container.getInstance(@com.opensymphony.xwork2.ognl.OgnlUtil@class)).(#ognlUtil.
getExcludedPackageNames().clear()).(#ognlUtil.getExcludedClasses().clear()).(#context.setMemberAccess
(#dm))).(#cmd='touch /tmp/pwned').(#iswin=@java.lang.System@getProperty('os.name').toLowerCase().contains
('win'))).(#cmds=(#iswin?{'cmd.exe','/c',#cmd}:{'/bin/bash','-c',#cmd})).(#p=new java.lang.ProcessBuilder
(#cmds)).(#p.redirectErrorStream(true)).(#process=#p.start()).(#ros=@org.apache.struts2.
ServletActionContext@getResponse().getOutputStream()).(@org.apache.commons.io.IOUtils@copy(#process.
getInputStream(),#ros)).(#ros.flush())}
...

```

Support

If you have questions or concerns regarding this advisory, please raise a support request at <https://support.atlassian.com/>.

References

Security Bug fix Policy	<p>As per our new policy, critical security bug fixes will be back-ported to major software versions for up to 12 months for JIRA and Confluence. We'll release new maintenance releases for the versions covered by the new policy instead of binary patches.</p> <p>Binary patches will no longer be released.</p>
Severity Levels for security issues	<p>Atlassian security advisories include a severity level and a CVE identifier. This severity level is based on our self-calculated CVSS score for each specific vulnerability. CVSS is an industry standard vulnerability metric. You can also learn more about CVSS at FIRST.org.</p>
End of Life Policy	<p>Our end-of-life policy varies for each product. Please refer to our EOL Policy for details.</p>

Crowd Security Advisory 2019-05-22

Crowd: pdkinstall development plugin incorrectly enabled (CVE-2019-11580)

Summary	pdkinstall development plugin incorrectly enabled (CVE-2019-11580)
Advisory Release Date	22 May 2019 10 AM PDT (Pacific Time,-7 hours)
Products	<ul style="list-style-type: none">• Crowd• Crowd Data Center
Affected Versions	<ul style="list-style-type: none">• 2.1.0 <= version < 3.0.5• 3.1.0 <= version < 3.1.6• 3.2.0 <= version < 3.2.8• 3.3.0 <= version < 3.3.5• 3.4.0 <= version < 3.4.4 <p>2.1.0 2.1.1 2.1.2 2.2.0 2.2.1 2.2.2 2.2.3 2.2.4 2.2.5 2.2.6 2.2.7 2.2.8 2.2.9 2.3.0 2.3.1 2.3.2 2.3.3 2.3.4 2.3.5 2.3.6 2.3.7 2.3.8 2.3.9 2.3.10 2.4.0 2.4.1 2.4.2 2.4.3 2.4.4 2.4.5 2.4.6 2.4.7 2.4.8 2.4.9 2.4.10 2.4.11 2.5.0 2.5.1 2.5.2 2.5.3 2.5.4</p>

2.5.5
2.5.6
2.5.7
2.6.0
2.6.1
2.6.2
2.6.3
2.6.4
2.6.5
2.6.6
2.6.7
2.7.0
2.7.1
2.7.2
2.8.0
2.8.1
2.8.2
2.8.3
2.8.4
2.8.6
2.8.7
2.8.8
2.9.1
2.9.2
2.9.3
2.9.4
2.9.5
2.9.6
2.9.7
2.10.1
2.10.2
2.10.3
2.10.4
2.11.0
2.11.1
2.11.2
2.12.0
2.12.1
3.0.0
3.0.1
3.0.2
3.0.3
3.1.0
3.1.1
3.1.2
3.1.3
3.1.4
3.1.5
3.2.0
3.2.1
3.2.2
3.2.3
3.2.4
3.2.5
3.2.6
3.2.7
3.3.0
3.3.1
3.3.2
3.3.3
3.3.4
3.4.0
3.4.1
3.4.2

	3.4.3
Fixed Versions	<ul style="list-style-type: none"> • 3.0.5 • 3.1.6 • 3.2.8 • 3.3.5 • 3.4.4
CVE ID(s)	CVE-2019-11580

Summary of Vulnerability

This advisory discloses a **critical severity** security vulnerability which was introduced in version 2.1.0 of Crowd and Crowd Data Center. Versions of Crowd and Crowd Data Center starting with version 2.1.0 before 3.0.5 (the fixed version for 3.0.x), from version 3.1.0 before 3.1.6 (the fixed version for 3.1.x), from version 3.2.0 before 3.2.8 (the fixed version for 3.2.x), from version 3.3.0 before 3.3.5 (the fixed version for 3.3.x), and from version 3.4.0 before 3.4.4 (the fixed version for 3.4.x) are affected by this vulnerability.

i Customers who have upgraded Crowd or Crowd Data Center to version 3.0.5 or 3.1.6 or 3.2.8 or 3.3.5 or 3.4.4 are not affected.

! Customers who are currently running:

- Crowd or Crowd Data Center **from version 2.1.0 before 3.0.5** (the fixed version for 3.0.x)
- Crowd or Crowd Data Center **from version 3.1.0 before 3.1.6** (the fixed version for 3.1.x)
- Crowd or Crowd Data Center **from version 3.2.0 before 3.2.8** (the fixed version for 3.2.x)
- Crowd or Crowd Data Center **from version 3.3.0 before 3.3.5** (the fixed version for 3.3.x)
- Crowd or Crowd Data Center **from version 3.4.0 before 3.4.4** (the fixed version for 3.4.x)

Please **upgrade your Crowd or Crowd Data Center** installations **immediately** to fix this vulnerability.

pdkinstall development plugin incorrectly enabled (CVE-2019-11580)

Severity

Atlassian rates the severity level of this vulnerability as **critical**, according to the scale published in our [Atlassian severity levels](#). The scale allows us to rank the severity as critical, high, moderate or low.


This is our assessment and you should evaluate its applicability to your own IT environment.

Description

Crowd and Crowd Data Center had the pdkinstall development plugin incorrectly enabled in release builds. Attackers who can send unauthenticated or authenticated requests to a Crowd or Crowd Data Center instance can exploit this vulnerability to install arbitrary plugins, which permits remote code execution on systems running a vulnerable version of Crowd or Crowd Data Center.

All versions of Crowd from version **2.1.0 before 3.0.5** (the fixed version for 3.0.x), from version **3.1.0 before 3.1.6** (the fixed version for 3.1.x), from version **3.2.0 before 3.2.8** (the fixed version for 3.2.x), from version **3.3.0 before 3.3.5** (the fixed version for 3.3.x), and from version **3.4.0 before 3.4.4** (the fixed version for 3.4.x) are affected by this vulnerability.

This issue can be tracked here:


 **GWD-5388** - Crowd - pdkinstall development plugin incorrectly enabled - CVE-2019-11580 CLOSED

Fix

We have taken the following steps to address this issue:

- Released Crowd and Crowd Data Center version 3.4.4 that contains a fix for this issue and can be downloaded from <https://www.atlassian.com/software/crowd/download>
- Released Crowd and Crowd Data Center versions 3.0.5, 3.1.6, 3.2.8, and 3.3.5 that contain a fix for this issue and can be downloaded from <https://www.atlassian.com/software/crowd/download-archive>

What You Need to Do

Atlassian recommends customers running a version of Crowd below version 3.3.0 upgrade to version 3.2.8 to avoid  **GWD-5352** CLOSED, for customers running a version above or equal to 3.3.0 Atlassian recommends to upgrade to the latest version. For a full description of the latest version of Crowd, see the [release notes](#). You can download the latest version of Crowd from the [download centre](#).

If you are running version	Then upgrade to bugfix version
2.1.0, 2.1.1, 2.1.2, 2.2.0, 2.2.1, 2.2.2, 2.2.3, 2.2.4, 2.2.5, 2.2.6, 2.2.7, 2.2.8, 2.2.9, 2.3.0, 2.3.1, 2.3.2, 2.3.3, 2.3.4, 2.3.5, 2.3.6, 2.3.7, 2.3.8, 2.3.9, 2.3.10, 2.4.0, 2.4.1, 2.4.2, 2.4.3, 2.4.4, 2.4.5, 2.4.6, 2.4.7, 2.4.8, 2.4.9, 2.4.10, 2.4.11, 2.5.0, 2.5.1, 2.5.2, 2.5.3, 2.5.4, 2.5.5, 2.5.6, 2.5.7, 2.6.0, 2.6.1, 2.6.2, 2.6.3, 2.6.4, 2.6.5, 2.6.6, 2.6.7, 2.7.0, 2.7.1, 2.7.2, 2.8.0, 2.8.1, 2.8.2, 2.8.3, 2.8.4, 2.8.6, 2.8.7, 2.8.8, 2.9.1, 2.9.2, 2.9.3, 2.9.4, 2.9.5, 2.9.6, 2.9.7, 2.10.1, 2.10.2, 2.10.3, 2.10.4, 2.11.0, 2.11.1, 2.11.2, 2.12.0, 2.12.1, 3.0.0, 3.0.1, 3.0.2, 3.0.3, 3.0.4	3.0.5
3.1.0, 3.1.1, 3.1.2, 3.1.3, 3.1.4, 3.1.5	3.1.6
3.2.0, 3.2.1, 3.2.2, 3.2.3, 3.2.4, 3.2.5, 3.2.6, 3.2.7	3.2.8
3.3.0, 3.3.1, 3.3.2, 3.3.3, 3.3.4	3.3.5
3.4.0, 3.4.1, 3.4.2, 3.4.3	3.4.4

Mitigation

This issue can be mitigated by doing the following:

1. Stop Crowd
2. Find and delete any pdkinstall-plugin jar files from the Crowd **installation directory and the data directory**
3. Remove the pdkinstall-plugin jar file from **<Crowd installation directory>/crowd-webapp/WEB-INF/classes/atlassian-bundled-plugins.zip**
 - a. Unzip the **<Crowd installation directory>/crowd-webapp/WEB-INF/classes/atlassian-bundled-plugins.zip** file
 - b. Delete the **pdkinstall-plugin-0.4.jar** file inside the extracted folder
 - c. Zip the contents of the folder back into **<Crowd installation directory>/crowd-webapp/WEB-INF/classes/atlassian-bundled-plugins.zip**
4. Start Crowd
 - a. If you have any issues starting Crowd after these changes - [clear the plugin cache](#).
5. Check that there are no pdkinstall-plugin jar files in the **installation directory or the data directory**

The following bash script can be used to apply the above mitigation on Linux systems:

```
#!/bin/bash
set -u

INSTALLATION_DIRECTORY= # set this to where crowd is installed
DATA_DIRECTORY= # set this to the crowd data directory

if [ -z "$INSTALLATION_DIRECTORY" ]
then
    echo "Please set INSTALLATION_DIRECTORY"
    exit 1
fi

if [ -z "$DATA_DIRECTORY" ]
then
    echo "Please set DATA_DIRECTORY"
    exit 1
fi

if test -f $DATA_DIRECTORY; then
    echo "Please check that DATA_DIRECTORY is correct."
    exit 1
fi

if test -f $INSTALLATION_DIRECTORY/stop_crowd.sh; then
    echo "Stopping Crowd"
    $INSTALLATION_DIRECTORY/stop_crowd.sh > /dev/null
    find $INSTALLATION_DIRECTORY -iname 'atlassian-bundled-plugins.zip' -exec zip -d {} 'pdkinstall-
plugin-*.jar' \;
    # You should see something like deleting: pdkinstall-plugin-0.4.jar after the above find command
    has run

    find $DATA_DIRECTORY -iname 'pdkinstall-plugin*' -exec rm {} \;
    echo "Starting Crowd"
    if test -f $INSTALLATION_DIRECTORY/start_crowd.sh; then
        $INSTALLATION_DIRECTORY/start_crowd.sh
        sleep 60
        find $DATA_DIRECTORY -iname 'pdkinstall-plugin*' -exec "Failed to apply the mitigation - {}
still exists" \;
    else
        echo "Failed to start crowd"
    fi
else
    echo "Unable to stop crowd, please ensure that you have specified the correct installation
directory."
fi
```

Support

If you did not receive an email for this advisory and you wish to receive such emails in the future go to <https://my.atlassian.com/email> and subscribe to Alerts emails.

If you have questions or concerns regarding this advisory, please raise a support request at <https://support.atlassian.com/>.

References

Security Bug fix Policy


As per our new policy critical security bug fixes will be back ported in accordance with <https://www.atlassian.com/trust/security/bug-fix-policy>. We will release new maintenance releases for the versions covered by the policy instead of binary patches.

Binary patches are no longer released.

Severity Levels for security issues	Atlassian security advisories include a severity level and a CVE identifier. This severity level is based on our self-calculated CVSS score for each specific vulnerability. CVSS is an industry standard vulnerability metric. You can also learn more about CVSS at FIRST.org .
End of Life Policy	Our end of life policy varies for different products. Please refer to our EOL Policy for details.

Constructing cron expressions in Crowd

A cron expression gives you more control when scheduling directory synchronization comparing to polling interval. For example, you could define a cron expression to run directory synchronization at 8:15 am on the second Friday of every month. This page describes how to construct a cron expression.

 This functionality is available for Crowd Data Center only

Constructing a cron expression

A cron expression is a string of fields separated by spaces. The following table displays the fields of a cron expression, in the order that they must be specified (from left to right):

	Second	Minute	Hour	Day-of-month	Month	Day-of-week	Year (optional)
Allowed values	0-59	0-59	0-23	1-31	1-12 or JAN-DEC	1-7 or SUN-SAT	1970-2099
Allowed special characters	, - * /	, - * /	, - * /	, - * / ? L W C	, - * /	, - * / ? L C #	, - * /

Note, cron expressions are not case-sensitive.

Here is an example:

```
0 15 8 ? JAN MON 2014
```

This literally translates to 0 second, 15 minute, 8 hour, any day of the month, January, Monday, 2014.

In plain English, this represents 8:15am one every Monday during January of 2014. Note, the ? character means "no particular value". In this example, we've set the Day-of-month to no particular value. We don't need to specify it, as we've specified a Day-of-week value. Read more about special characters in the next section.

More examples of cron expressions are explained in the examples section at the bottom of this page.

Special characters

Special character	Usage
,	Specifies a list of values. For example, in the Day-of-week field, 'MON,WED,FRI' means 'every Monday, Wednesday, and Friday'.
-	Specifies a range of values. For example, in the Day-of-week field, 'MON-FRI' means 'every Monday, Tuesday, Wednesday, Thursday and Friday'.
*	Specifies all possible values. For example, in the Hour field, '*' means 'every hour of the day'.
/	Specifies increments to the given value. For example, in the Minute field, '0/15' means 'every 15 minutes during the hour, starting at minute zero'.
?	Specifies no particular value. This is useful when you need to specify a value for one of the two fields Day-of-month or Day-of-week , but not the other.

L	Specifies the last possible value; this has different meanings depending on context. In the Day-of-week field, 'L' on its own means 'the last day of every week' (i.e. 'every Saturday'), or if used after another value, means 'the last xxx day of the month' (e.g. 'SATL' and '7L' both mean 'the last Saturday of the month'). In the Day-of-month field, 'L' on its own means 'the last day of the month', or 'LW' means 'the last weekday of the month'.
W	Specifies the weekday (Monday-Friday) nearest the given day of the month. For example, '1W' means 'the nearest weekday to the 1st of the month' (note that if the 1st is a Saturday, the email will be sent on the nearest weekday <i>within the same month</i> , i.e. on Monday 3rd). 'W' can only be used when the day-of-month is a single day, not a range or list of days.
#	Specifies the nth occurrence of a given day of the week. For example, 'TUES#2' (or '3#2') means 'the second Tuesday of the month'.

Examples

0 15 20 ? * *	Every day at 8:15 pm.
0 15 8 * * ?	Every day at 8:15 am.
0 * 14 * * ?	Every minute starting at 2:00 pm and ending at 2:59 pm, every day.
0 0/5 14 * * ?	Every 5 minutes starting at 2:00 pm and ending at 2:55 pm, every day.
0 0/5 14,18 * * ?	Every 5 minutes starting at 2:00 pm and ending at 2:55 pm, AND every 5 minutes starting at 6:00 pm and ending at 6:55 pm, every day.
0 0-5 14 * * ?	Every minute starting at 2:00 pm and ending at 2:05 pm, every day.
0 0/10 * * * ? *	Every 10 minutes, forever.
0 10,44 14 ? 3 WED	2:10 pm and 2:44 pm every Wednesday in the month of March.
0 15 8 ? * MON-FRI	8:15 am every Monday, Tuesday, Wednesday, Thursday, and Friday.
0 15 8 15 * ?	8:15 am on the 15th day of every month.
0 15 8 L * ?	8:15 am on the last day of every month.
0 15 8 LW * ?	8:15 am on the last weekday of every month.
0 15 8 ? * 6L	8:15 am on the last Friday of every month.
0 15 8 ? * 6#2	8:15 am on the second Friday of every month.
0 15 8 ? * 6#2 2007-2009	8:15 am on the second Friday of every month during the years 2007, 2008, and 2009.

User Guide

About Crowd

Atlassian's **Crowd** is a software application installed by the system administrator. The administrator will also connect one or more of your organization's applications to Crowd. When you log in to a **Crowd-connected application**, Crowd will verify your password and login permissions.

Using Crowd for single sign-on (SSO), each person needs only one username and password to access all web applications. You can host your own OpenID provider to include external applications.

- You only need to log in once, to Crowd or a Crowd-connected application. When you start another Crowd-connected application, you will be logged in automatically.
- When you log out of Crowd or one of the Crowd-connected applications, you will be logged out of Crowd and the other application (s) at the same time.

Crowd also manages the information held about you as a user of other software applications:

- Your login permissions to various applications.
- The password you use to log in to those applications.
- The groups and roles you belong to, which are used by the applications to decide which functions you can perform within the applications.
- The user directories which hold your information.

Search the User Guide

About the User Guide

The **Crowd User Guide** contains information for people who use Crowd to update their user profiles and passwords and to view their groups, roles and applications.

If you need information about installing Crowd, configuring your Crowd server or using the Crowd Administration Console, please visit the [Crowd documentation home page](#).

If you have a question about using Crowd that hasn't been answered here, please [let us know](#).

Download

You can [download the Crowd documentation](#) in PDF, HTML or XML formats.

Getting Help

[Support](#) | [Feature requests and bug reports](#) | [Answers](#)
| [Knowledge base](#)

Introduction to Crowd

This page gives a brief introduction to Crowd, for people who will view and update their login and user profile information in Crowd.

What is Crowd?

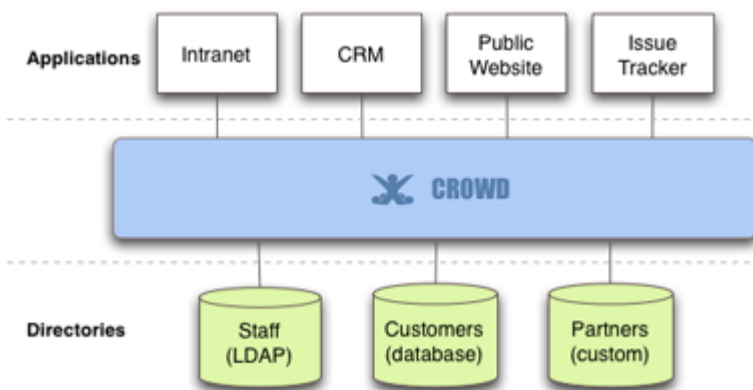
Atlassian's [Crowd](#) is a software application installed by the system administrator. The administrator will also connect one or more of your organization's applications to Crowd. When you log in to a [Crowd-connected application](#), Crowd will verify your password and login permissions.

Using Crowd for single sign-on (SSO), each person needs only one username and password to access all web applications. You can host your own OpenID provider to include external applications.

- You only need to log in once, to Crowd or a Crowd-connected application. When you start another Crowd-connected application, you will be logged in automatically.
- When you log out of Crowd or one of the Crowd-connected applications, you will be logged out of Crowd and the other application(s) at the same time.

Crowd also manages the information held about you as a user of other software applications:

- Your login permissions to various applications.
- The password you use to log in to those applications.
- The groups and roles you belong to, which are used by the applications to decide which functions you can perform within the applications.
- The user directories which hold your information.



Using Crowd

The [Crowd administrator](#) has access to Crowd's Administration Console, which provides the functions described in the [Administration Guide](#).

Every [authorized Crowd user](#) has access to Crowd's Self-Service Console, where you can edit your user profile, change your password and view other information about your Crowd username. The [User Guide](#) describes this functionality.

Some Terminology

Here is a list of all entries in the glossary, plus the first few lines of content. Click a link to see the full text for each entry.

Alias

Crowd allows you to have different usernames in different applications. These different usernames are called 'aliases'. Your Crowd administrator can manage your aliases for the applications you are authorized to access.

Authorization to Use Crowd

If you are authorized to use Crowd, you can log in to Crowd's Self-Service Console to update your user profile and view other information about your username. The Crowd administrator can grant people access to the Self-Service Console, as described in the Crowd Administration Guide. Basically, the administrator should ensure that your username is in a user directory which is mapped to the Crowd application.

Crowd Administrator

A Crowd administrator is a user who has access to the Crowd Administration Console, which provides the functions described in the Administration Guide. The first administrator is defined during the installation of Crowd. A Crowd administrator can grant administration rights to other users, as described in the Crowd Administration Guide.

Crowd-Connected Application

A 'Crowd-connected application' is a software application which has been designed and configured to use Crowd for user logins. These applications pass all login requests to Crowd for authentication. Depending on the integration level, the application may also make use of the groups and roles defined in Crowd for authorization purposes, and allow single sign-on across the Crowd domain. The Crowd Administration Guide tells you how to connect an application to Crowd.

Directory

Crowd uses the term 'directory', or 'user directory', to refer to a store of information about a user. Typically, a directory will hold your username, name, password, email address, and so on. Your Crowd administrator can define one or more directories internally in Crowd or connect one or more external directories to Crowd.

Group

A group is a collection of users. Administrators create groups so that the administrator can assign permissions to a number of people at once. For example, it is quicker to give group 'X' access to Jira, rather than giving every team member access individually. In Crowd, each group belongs to a specific directory. It is possible to have two groups with the same name, such as 'X', in two different directories. A user can be a member of group 'X' in one directory, in both directories or in neither.

Role

Support for roles, previously deprecated, was removed in Crowd 2.5.

Self-Service Console

Authorized Crowd users can access the Crowd Console, even if they are not Crowd administrators. Non-administrators will see a subset of the Crowd Console functionality, which we call the 'Self-Service Console'. The User Guide describes this functionality. The Crowd Administration Console presents the full range of Crowd administration functionality to authorized Crowd administrators.

Single Sign-On

Single sign-on (SSO) is a feature offered by Crowd. Your Crowd administrator can choose to enable this feature for the Crowd-connected applications. If SSO is enabled, you will only need to log in or log out once.

Logging in to Crowd

If you are authorized to use Crowd, you can log in to Crowd's Self-Service Console to update your user profile and view other information about your username. The [Crowd administrator](#) can grant people access to the Self-Service Console, as described in the [Crowd Administration Guide](#). Basically, the administrator should ensure that your username is in a user directory which is mapped to the Crowd application.

If your administrator has configured Crowd to allow [single sign-on](#) (SSO), then you only need to log in once. When you start another [Crowd-connected application](#), you will be logged in automatically.

On this page:

- [How to Log In](#)
- [User Aliases](#)
- [SSO and Google Apps](#)

How to Log In

To log in to Crowd,

1. Open Crowd in your web browser. In most cases, you will do this by typing an address like this one into the browser's address bar:

```
http://YOUR-CROWD-LOCATION:8095/crowd/
```

Replace 'YOUR-CROWD-LOCATION' with the address of your Crowd server. (Ask your Crowd administrator for this address.)

2. The Crowd login screen will appear, as shown in the screenshot below. Enter your username and password.
3. Click the **'Log In'** button.

If you have forgotten your password or your username, you can click the link labeled **'Can't access your account?'**. Read more about [resetting your password](#) or [requesting a forgotten username](#).

User Aliases


Crowd allows you to have different usernames in different applications. These different usernames are called 'aliases'. Your Crowd administrator can manage your aliases for the applications you are authorized to access.

- When you log in to Crowd itself, you must use your primary username i.e. the one registered in Crowd.
- If you choose to log in to another Crowd-connected application directly, such as Confluence or Jira, instead of logging in via Crowd, then you must log in using the alias registered in that application (Confluence, Jira, or whatever.)
- If [SSO](#) is enabled you will only need to log in or log out once, to Crowd or a Crowd-connected application. When you start another Crowd-connected application, you will be logged in automatically.

SSO and Google Apps

These notes are relevant if your Crowd administrator has enabled single sign-on between Crowd and Google Apps:

- Single sign-on (SSO) applies only to the applications within Google Apps. The Google Apps administration section (control panel) does not support SSO.
- When you sign out of Google Apps, you will also be signed out of Crowd and all Crowd-connected applications. This is the usual SSO behavior.

- But when you sign out of Crowd, you will remain logged in to Google Apps even though you will be logged out of other Crowd-connected applications. (Reason: Google does not rely on a cookie, so there is no easy way for Crowd to tell Google you have signed out.)
 -  It would take some additional development to support single sign-out from Google Apps. If you would like to see this work undertaken, please vote for issue [CWD-1238](#).
- If you go directly to a Google Apps application without logging in to Crowd, Google Apps direct you to a Crowd login screen.
- The Crowd login screen for Google Apps will not offer a 'Forgotten your password' link. You cannot change your Crowd password via Google Apps. Instead, if you need to change your password please log in to Crowd directly, by going to this URL: <http://YOUR-CROWD-LOCATION:8095/crowd/>

Logging out of Crowd

Logging out of Crowd is easy just click the **'Log Out'** link at the top of the Crowd screen.

If your administrator has configured Crowd to allow [single sign-on](#) (SSO), then you will be automatically logged out of all [Crowd-connected applications](#) when you log out of Crowd.

i This automatic logout will also happen if you log out of one of the other Crowd-connected applications you will be logged out of Crowd and the other application(s) at the same time.

i SSO and Google Apps

- Single sign-on (SSO) applies only to the applications within Google Apps. The Google Apps administration section (control panel) does not support SSO.
- When you sign out of Google Apps, you will also be signed out of Crowd and all Crowd-connected applications. This is the usual SSO behavior.
- But when you sign out of Crowd, you will remain logged in to Google Apps even though you will be logged out of other Crowd-connected applications. (Reason: Google does not rely on a cookie, so there is no easy way for Crowd to tell Google you have signed out.)
 - i** It would take some additional development to support single sign-out from Google Apps. If you would like to see this work undertaken, please vote for issue [CWD-1238](#).
- If you go directly to a Google Apps application without logging in to Crowd, Google Apps direct you to a Crowd login screen.
- The Crowd login screen for Google Apps will not offer a 'Forgotten your password' link. You cannot change your Crowd password via Google Apps. Instead, if you need to change your password please log in to Crowd directly, by going to this URL: <http://YOUR-CROWD-LOCATION:8095/crowd/>

Changing or Resetting your Password

If you are authorized to use Crowd, you can log in to Crowd's Self-Service Console and [change your password](#).

When attempting to log in to Crowd, you can also ask to [reset your password](#). This is useful if you have forgotten the old one.

Password change applies to one user directory only

In most cases, your username will be defined in one [user directory](#) only. But some organizations may have more than one user directory. For example, your username may be defined in Crowd for Jira use, and also in another Crowd-connected directory (e.g. LDAP) for use in another application. If you change your password, the new password will apply only in one directory: the directory mapped to the '**crowd**' application and defined as **first** in the directory sequence. Your Crowd administrator can define the order of the directories, as described in the [Crowd Administration Guide](#).

Changing your Password

If you are [authorized to use Crowd](#), you can log in to Crowd's Self-Service Console and change your password, as described below. If you have forgotten your password or your username, you can [ask Crowd to email your username](#) and [reset your password](#).

To change your password

1. [Log in](#) to Crowd.
2. If you are not a [Crowd administrator](#), you can skip this step because you will go directly to the Crowd Self-Service Console.
 - If you are a Crowd administrator, the **Crowd Administration Console** will open. Click the **'My Profile'** link in the top navigation bar.
3. The **Crowd Self-Service Console** will open.
4. Click **'Change Password'** in the left-hand menu.
5. The **'Change Password'** screen will appear, as shown in the screenshot below. Enter the following information:
 - **Current Password** Your current password.
 - **New Password** The new password you would like to start using.
 - **Confirm Password** Your new password again, to verify that you typed it correctly the first time.
6. Click the **'Update'** button.
7. If the change is successful, a **'Password updated'** message will appear on the screen.

Password change applies to one user directory only

In most cases, your username will be defined in one [user directory](#) only. But some organizations may have more than one user directory. For example, your username may be defined in Crowd for Jira use, and also in another Crowd-connected directory (e.g. LDAP) for use in another application. If you change your password, the new password will apply only in one directory: the directory mapped to the **'crowd'** application and defined as **first** in the directory sequence. Your Crowd administrator can define the order of the directories, as described in the [Crowd Administration Guide](#).

Resetting Forgotten Passwords

You can request your password to be reset directly from the login screen. Crowd will send you an email message containing a unique, randomly-generated URL. When you click the link on that URL, you will go to a screen where you can choose your own new password.

To reset your password

1. Open Crowd in your web browser. In most cases, you will do this by typing an address like this one into the browser's address bar:

```
http://YOUR-CROWD-LOCATION:8095/crowd/
```

Replace 'YOUR-CROWD-LOCATION' with the address of your Crowd server. (Ask your Crowd administrator for this address.)

2. The Crowd login screen appears.
3. Click **Forgot your password?**
4. Select **I have forgotten my password.**
5. Enter your Crowd username and click the **Continue**.

You will receive an email message that contains a link to a unique, randomly-generated URL of a page where you can create a new password. This link remains available for 24 hours. Click the link in the email message or copy the URL to your browser address bar.

Password change applies to one user directory only

In most cases, your username will be defined in one [user directory](#) only. But some organizations may have more than one user directory. For example, your username may be defined in Crowd for Jira use, and also in another Crowd-connected directory (e.g. LDAP) for use in another application. If you change your password, the new password will apply only in one directory: the directory mapped to the '**crowd**' application and defined as **first** in the directory sequence. Your Crowd administrator can define the order of the directories, as described in the [Crowd Administration Guide](#).

Requesting usernames

You can go to the Crowd **Login** screen and ask Crowd to email you your username(s). This is useful when you have forgotten your username. Crowd will send a message to the email address you specify, containing all the usernames that are registered for that email address.

To request your username(s)

1. Open Crowd in your web browser. In most cases, you will do this by typing an address like this one into the browser's address bar:

```
http://YOUR-CROWD-LOCATION:8095/crowd/
```

Replace 'YOUR-CROWD-LOCATION' with the address of your Crowd server. (Ask your Crowd administrator for this address.)

2. The Crowd login screen appears.
3. Click **Forgot your password?**
4. Select **I have forgotten my username.**
5. Enter your Crowd username and click the **Continue**.

If your email address is recognized, you will receive a message with your username in it.

Updating your User Profile

Provided that you are [authorized to use Crowd](#), you can change the profile information for your username.

To update your user profile

1. [Log in](#) to Crowd.
2. If you are not a [Crowd administrator](#), you can skip this step because you will go directly to the Crowd Self-Service Console.
 - If you are a Crowd administrator, the **Crowd Administration Console** will open. Click the '**My Profile**' link in the top navigation bar.
3. The **My Profile** screen will open, as shown in the screenshot below.
4. Update your profile information where necessary:
 - **First Name** Your first name.
 - **Last Name** Your last name or surname.
 - **Email** Crowd will use this email address when sending you messages, such as a new password if you [reset your password](#).

Which user directories are updated?

In most cases, your username will be defined in one [user directory](#) only. But some organizations may have more than one user directory. For example, your username may be defined in Crowd for Jira use, and also in another Crowd-connected directory (e.g. LDAP) for use in another application. If you change your profile details, the change will be applied to **all** directories which the '**crowd**' application has permission to update. Your Crowd administrator defines the application permissions, as described in the [Crowd Administration Guide](#).

Viewing your Group Membership

Provided that you are [authorized to use Crowd](#), you can see a list of the groups to which your username belongs.

To see which groups you belong to

1. [Log in](#) to Crowd.
2. If you are not a [Crowd administrator](#), you can skip this step because you will go directly to the Crowd Self-Service Console.
 - If you are a Crowd administrator, the **Crowd Administration Console** will open. Click the **My Profile** link in the top navigation bar.
3. The **Crowd Self-Service Console** will open. Click **Groups** in the left-hand menu.
4. The **Groups** screen will appear, as shown in the screenshot below.

What is a Group?

A group is a collection of users. Administrators create groups so that the administrator can assign permissions to a number of people at once. For example, it is quicker to give group 'X' access to Jira, rather than giving every team member access individually. In Crowd, each group belongs to a specific [directory](#). It is possible to have two groups with the same name, such as 'X', in two different directories. A user can be a member of group 'X' in one directory, in both directories or in neither directory. Two groups called 'X' will be presented to an application as a single group with membership lists aggregated. Groups are particularly important in Crowd, as they are used to [control access to applications](#).

Each group appears only once

Even if you are a member of the same group in more than one directory, the group name will appear only once on this screen. *More explanation:* In most cases, your username will be defined in one [user directory](#) only. But some organizations may have more than one user directory. For example, your username may be defined in Crowd as a Crowd administrator, and also in another Crowd-connected directory (e.g. LDAP). In addition, you may then be a member of the same group (e.g. 'confluence-users') in both directories. On the Crowd **Groups** screen, the group 'confluence-users' will appear only once.

Viewing your Applications

Provided that you are [authorized to use Crowd](#), you can see a list of the applications you are authorized to log in to.

More information about the applications listed:

- Crowd verifies all logins to these applications. Your Crowd administrator has defined them as [Crowd-connected applications](#).
- Your username is authorized to log in to these applications. Your Crowd administrator has made you a member of a directory or a group which is mapped to the application.

Crowd allows you to have different usernames in different applications. These different usernames are called 'aliases'. Your Crowd administrator can manage your aliases for the applications you are authorized to access.

- When you log in to Crowd itself, you must use your primary username i.e. the one registered in Crowd.
- If you choose to log in to another Crowd-connected application directly, such as Confluence or Jira, instead of logging in via Crowd, then you must log in using the alias registered in that application (Confluence, Jira, or whatever.)
- If [SSO](#) is enabled you will only need to log in or log out once, to Crowd or a Crowd-connected application. When you start another Crowd-connected application, you will be logged in automatically.

To see the applications which you can log in to

1. [Log in](#) to Crowd.
2. If you are not a [Crowd administrator](#), you can skip this step because you will go directly to the Crowd Self-Service Console.
 - If you are a Crowd administrator, the **Crowd Administration Console** will open. Click the '**My Profile**' link in the top navigation bar.
3. The **Crowd Self-Service Console** will open. Click '**Applications**' in the left-hand menu.
4. The '**Applications**' screen will appear, as shown in the screenshot below.

The 'crowd' application

One of the applications listed will be the '**crowd**' application. This is the Crowd Administration and Self-Service Console. If you can log in to Crowd, that means that you do have access to the 'crowd' application and you should see it in the list.

Crowd User's Glossary

Here is a list of all entries in the glossary, plus the first few lines of content. Click a link to see the full text for each entry.

Alias

Crowd allows you to have different usernames in different applications. These different usernames are called 'aliases'. Your Crowd administrator can manage your aliases for the applications you are authorized to access.

Authorization to Use Crowd

If you are authorized to use Crowd, you can log in to Crowd's Self-Service Console to update your user profile and view other information about your username. The Crowd administrator can grant people access to the Self-Service Console, as described in the Crowd Administration Guide. Basically, the administrator should ensure that your username is in a user directory which is mapped to the Crowd application.

Crowd Administrator

A Crowd administrator is a user who has access to the Crowd Administration Console, which provides the functions described in the Administration Guide. The first administrator is defined during the installation of Crowd. A Crowd administrator can grant administration rights to other users, as described in the Crowd Administration Guide.

Crowd-Connected Application

A 'Crowd-connected application' is a software application which has been designed and configured to use Crowd for user logins. These applications pass all login requests to Crowd for authentication. Depending on the integration level, the application may also make use of the groups and roles defined in Crowd for authorization purposes, and allow single sign-on across the Crowd domain. The Crowd Administration Guide tells you how to connect an application to Crowd.

Directory

Crowd uses the term 'directory', or 'user directory', to refer to a store of information about a user. Typically, a directory will hold your username, name, password, email address, and so on. Your Crowd administrator can define one or more directories internally in Crowd or connect one or more external directories to Crowd.

Group

A group is a collection of users. Administrators create groups so that the administrator can assign permissions to a number of people at once. For example, it is quicker to give group 'X' access to Jira, rather than giving every team member access individually. In Crowd, each group belongs to a specific directory. It is possible to have two groups with the same name, such as 'X', in two different directories. A user can be a member of group 'X' in one directory, in both directories or in neither

Role

Support for roles, previously deprecated, was removed in Crowd 2.5.

Self-Service Console

Authorized Crowd users can access the Crowd Console, even if they are not Crowd administrators. Non-administrators will see a subset of the Crowd Console functionality, which we call the 'Self-Service Console'. The User Guide describes this functionality. The Crowd Administration Console presents the full range of Crowd administration functionality to authorized Crowd administrators.

Single Sign-On

Single sign-on (SSO) is a feature offered by Crowd. Your Crowd administrator can choose to enable this feature for the Crowd-connected applications. If SSO is enabled, you will only need to log in or log out once.

RELATED TOPICS

[Introduction to Crowd
User Guide](#)

Alias

Crowd allows you to have different usernames in different applications. These different usernames are called 'aliases'. Your Crowd administrator can manage your aliases for the applications you are authorized to access.

- When you log in to Crowd itself, you must use your primary username i.e. the one registered in Crowd.
- If you choose to log in to another Crowd-connected application directly, such as Confluence or Jira, instead of logging in via Crowd, then you must log in using the alias registered in that application (Confluence, Jira, or whatever.)
- If [SSO](#) is enabled you will only need to log in or log out once, to Crowd or a Crowd-connected application. When you start another Crowd-connected application, you will be logged in automatically.

Authorization to Use Crowd

If you are authorized to use Crowd, you can log in to Crowd's Self-Service Console to update your user profile and view other information about your username. The [Crowd administrator](#) can grant people access to the Self-Service Console, as described in the [Crowd Administration Guide](#). Basically, the administrator should ensure that your username is in a user directory which is mapped to the Crowd application.

Crowd Administrator

A Crowd administrator is a user who has access to the **Crowd Administration Console**, which provides the functions described in the [Administration Guide](#). The first administrator is defined during the installation of Crowd. A Crowd administrator can grant administration rights to other users, as described in the [Crowd Administration Guide](#).

Crowd-Connected Application

A 'Crowd-connected application' is a software application which has been designed and configured to use Crowd for user logins. These applications pass all login requests to Crowd for authentication. Depending on the integration level, the application may also make use of the groups and roles defined in Crowd for authorization purposes, and allow [single sign-on](#) across the Crowd domain. The [Crowd Administration Guide](#) tells you how to connect an application to Crowd.

Directory

Crowd uses the term 'directory', or 'user directory', to refer to a store of information about a user. Typically, a directory will hold your username, name, password, email address, and so on. Your [Crowd administrator](#) can define one or more directories internally in Crowd or connect one or more external directories to Crowd.

The external directory may be a corporate directory such as Microsoft's Active Directory. To learn more about Crowd's directory management, please refer to the [Crowd Administration Guide](#).

Group

A group is a collection of users. Administrators create groups so that the administrator can assign permissions to a number of people at once. For example, it is quicker to give group 'X' access to Jira, rather than giving every team member access individually. In Crowd, each group belongs to a specific [directory](#). It is possible to have two groups with the same name, such as 'X', in two different directories. A user can be a member of group 'X' in one directory, in both directories or in neither directory. Two groups called 'X' will be presented to an application as a single group with membership lists aggregated. Groups are particularly important in Crowd, as they are used to [control access to applications](#).

Role

Support for roles, [previously deprecated](#), **has been removed** in Crowd 2.5. The implementation of roles in Crowd was identical to the implementation of groups and did not provide any extra functionality.

Self-Service Console

[Authorized Crowd users](#) can access the Crowd Console, even if they are not [Crowd administrators](#). Non-administrators will see a subset of the Crowd Console functionality, which we call the 'Self-Service Console'. The [User Guide](#) describes this functionality. The [Crowd Administration Console](#) presents the full range of Crowd administration functionality to authorized Crowd administrators.

Single Sign-On

Single sign-on (SSO) is a feature offered by Crowd. Your Crowd administrator can choose to enable this feature for the [Crowd-connected applications](#). If SSO is enabled, you will only need to log in or log out once.

Specifically:

- You only need to log in once, to Crowd or a Crowd-connected application. When you start another Crowd-connected application, you will be logged in automatically.
- When you log out of Crowd or one of the Crowd-connected applications, you will be logged out of Crowd and the other application(s) at the same time.

Monitoring license usage

Centralized license visibility in Crowd allows you to verify the actual license consumption in all Atlassian products in your environment.

Before you begin

- To enable license monitoring for your product in Crowd, you must add that product to Crowd first. See [Adding an Application](#).
- This functionality is available for Crowd Data Center only.
- Centralized license monitoring in Crowd supports all currently supported versions of the following products: Jira, Bitbucket, Confluence.


Enable license usage - process overview

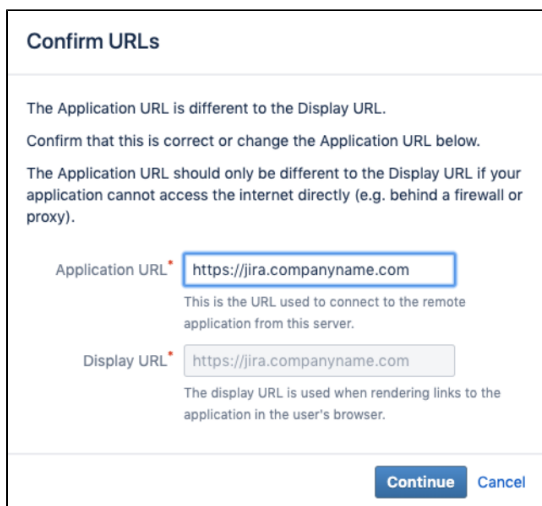
It takes three steps to enable centralized license visibility in Crowd. First, from Atlassian Marketplace, you must install the License license visibility plugin. Once you've done that, you must establish communication between your application and Crowd by creating an application link between the two. Finally, in your application's settings in Crowd, you just have to set the right application link. Then you're ready to receive license data from your application. See steps below for step-by-step instructions on how to do all that.

1. Install the Centralized license visibility plugin

1. Download the plugin from [Atlassian Marketplace](#).
2. Install the plugin in the Atlassian product in which you want to monitor license usage.
You must install the Centralized license visibility plugin in every Atlassian product in which you want to monitor license usage.

2. Create the application link / Link Crowd with your application

1. Go to  > **Application Links**.
2. Enter the URL of the application you want to link.
3. Click **Create new link**.
4. Follow the wizard.



Confirm URLs

The Application URL is different to the Display URL.
Confirm that this is correct or change the Application URL below.

The Application URL should only be different to the Display URL if your application cannot access the internet directly (e.g. behind a firewall or proxy).

Application URL*
This is the URL used to connect to the remote application from this server.

Display URL*
The display URL is used when rendering links to the application in the user's browser.

Continue **Cancel**

3. Add the application link to your application

1. In the top navigation bar, click **Applications**.
2. Click on your application.

3. In the **Details** tab, go to the Application link section.

Application links

Application links allow Crowd to communicate with other Atlassian applications. To start this communication, select an application link for your application from the drop-down menu. You can create application links in the [Admin panel](#).

Application Link

An Application Link associated with this application.

Synchronisation interval

Minutes after which Crowd will synchronize data from the application.

4. From the dropdown list, select the application link you previously created.
5. Set the *Synchronisation interval*.
6. Click **Save**.

Results


You have set up license monitoring for your application. You can view license consumption for the application by clicking the **Licenses** tab in your application details.

Password encryption

All passwords for external systems stored in Crowd are encrypted by default.

 The password encryption functionality is available starting from **CROWD 4.2**.

If you're an upgrading user, your passwords stored in Crowd will be encrypted automatically during upgrade to Crowd 4.2 or later.

 Starting from Crowd 4.2 it's crucial to make sure you backup your encryption keys. Without them you won't be able to properly restore Crowd from backup.

The keys are stored in the shared folder `{crowdHome}/shared/keys`. To backup the keys copy the mentioned directory to the secure place.

To correctly restore from backup with encrypted passwords, corresponding keys must be present in the `{crowdHome}/shared/keys`, otherwise Crowd won't be able to decrypt passwords.

Here's the complete list of sensitive data which Crowd encrypts:

- LDAP directory password
- Remote Crowd directory application password
- Azure AD web application key
- SMTP mail password
- Proxy password

FAQ

Crowd can encrypt your password using one of the following algorithms:


- AES/CBC/PKCS5Padding (default)
- DES/CBC/PKCS5Padding
- DESede/CBC/PKCS5Padding

To change the default encryption algorithm:


1. Issue the admin authenticated PUT request to the following URL:
`{baseUrl}/rest/admin/1.0/encryption/encryptor`.
2. Set content type to `application/json`.
3. Set body to algorithm name.

List of available names can be found through GET request to `{baseUrl}/rest/admin/1.0/encryption`

Yes. Password encryption is enabled by default in Crowd 4.2. To disable it, issue the admin authenticated PUT request to the following URL `{baseUrl}/rest/admin/1.0/encryption/disable`

 The password encryption will be disabled and your existing data will be automatically decrypted.

To reenable your password encryption, follow the [procedure for changing the encryption algorithm](#).

 For security reasons, we recommend rotating the encryption key at least once per year. If the security key is leaked, you must rotate it immediately.


Issue admin authenticated PUT request to the following url `{baseUrl}/rest/admin/1.0/encryption/changeKey` .

Existing data will be automatically re-encrypted using new encryption key.

Rest endpoints support both basic authentication and Crowd token key (usually `crowd.token_key`). Depending on configuration, Crowd might not allow to re-use token key cookie on IP address different than the initial one.

If the encryption key is missing, Crowd wont be able to decrypt passwords. All passwords will have to be restored manually.

If your admin account comes from remote directory, admin wont be able to authenticate. In such case Crowd will need to be started in recovery mode to restore passwords manually. See [Using recovery mode](#).

 You should consider disabling password encryption only if you notice that this functionality is causing you problems with the upgrade.

To disable password encryption during upgrade:

Start Crowd with the following flag `-Dcrowd.encryption.upgrade.disabled=true`

For higher security, please restrict filesystem permission for `{crowdHome}/shared/keys` so that only Crowd user (on all nodes) have access to this directory.

CrowdID Administration Guide

CrowdID is a free add-on that ships with Crowd versions 1.1 and later. It gives administrators a secure way to provide [OpenID](#) accounts for their users.

The *CrowdID Administration Guide* is for people who have [CrowdID administration rights](#). For instructions on using CrowdID to access OpenID-enabled websites, please see the [CrowdID User Guide](#).


 Please note that CrowdID does not support clustering. When using CrowdID with Crowd Data Center you will need to make sure that only one node is running CrowdID. Please see [Installing Crowd Data Center](#) for details

Table of Contents

- 1. About CrowdID
 - 1.1 How CrowdID works with Crowd
 - 1.1.1 Determining the name of the CrowdID application
 - 1.1.2 Locating the Crowd Server that CrowdID is using
 - 1.2 How OpenID sites interact with CrowdID
 - 1.3 Lightweight OpenID server
- 2. Allowing users to access CrowdID
 - 2.1 Granting CrowdID access rights to a user
 - 2.2 Granting CrowdID Administration Rights to a User
- 3. Specifying the sites to which users can log in
 - 3.1 Allowing all hosts
 - 3.2 Allowing all except specified hosts ('Blacklist')
 - 3.3 Allowing specified hosts only ('Whitelist')
 - 3.4 Approval Whitelist
- 4. Configuring CrowdID system settings
 - 4.1 Specifying the CrowdID URL
 - 4.2 Enabling localhost authentication
 - 4.3 Enabling immediate authentication requests
 - 4.4 Enabling communication with stateless clients

1. About CrowdID

CrowdID is a free add-on that ships with Crowd versions 1.1 and later. It gives administrators a secure way to provide [OpenID](#) accounts for their users.

Crowd is a middleware application that connects web applications (such as CrowdID, Jira and Confluence) to specified directories (e.g. Microsoft Active Directory, OpenLDAP). For details please see [Concepts](#) in the *Administration Guide*.

- [1.1 How CrowdID works with Crowd](#)
 - [1.1.1 Determining the name of the CrowdID application](#)
 - [1.1.2 Locating the Crowd Server that CrowdID is using](#)
- [1.2 How OpenID sites interact with CrowdID](#)
- [1.3 Lightweight OpenID server](#)

To access CrowdID, go to `http://localhost:8095/openidserver`.

1.1 How CrowdID works with Crowd

CrowdID is a free add-on that ships with Crowd versions 1.1 and later. It gives administrators a secure way to provide [OpenID](#) accounts for their users.

Crowd is a middleware application that connects web applications (such as CrowdID, Jira and Confluence) to specified directories (e.g. Microsoft Active Directory, OpenLDAP). For details please see [Concepts](#) in the [Administration Guide](#).

This means that:

- CrowdID is a Crowd-connected application.
- CrowdID users are authenticated against Crowd-connected directories.
- If a user has already logged into any other Crowd-connected application (and single sign-on is enabled), they will not be prompted for any further login once they have entered their OpenID URL at an OpenID-enabled website.
- Multiple CrowdID instances can use one Crowd instance. Large organizations often find this useful.

CrowdID is automatically installed when you install Crowd. When you start Crowd for the first time and run the [Setup Wizard](#), you will be offered the option of configuring CrowdID. If you choose not to setup CrowdID at that time, you can always set it up later as described in [4. Configuring CrowdID system settings](#). Note that you will also need to define the CrowdID application in Crowd, and map it to an appropriate directory for details please see the [Administration Guide](#). To access CrowdID, go to `http://localhost:8095/openidserver`.

RELATED TOPICS

- [1.1 How CrowdID works with Crowd](#)
 - [1.1.1 Determining the name of the CrowdID application](#)
 - [1.1.2 Locating the Crowd Server that CrowdID is using](#)
- [1.2 How OpenID sites interact with CrowdID](#)
- [1.3 Lightweight OpenID server](#)

[Crowd documentation](#)

1.1.1 Determining the name of the CrowdID application

CrowdID is a Crowd-connected application (for more information please see [Managing Applications](#) in the *Administration Guide*).

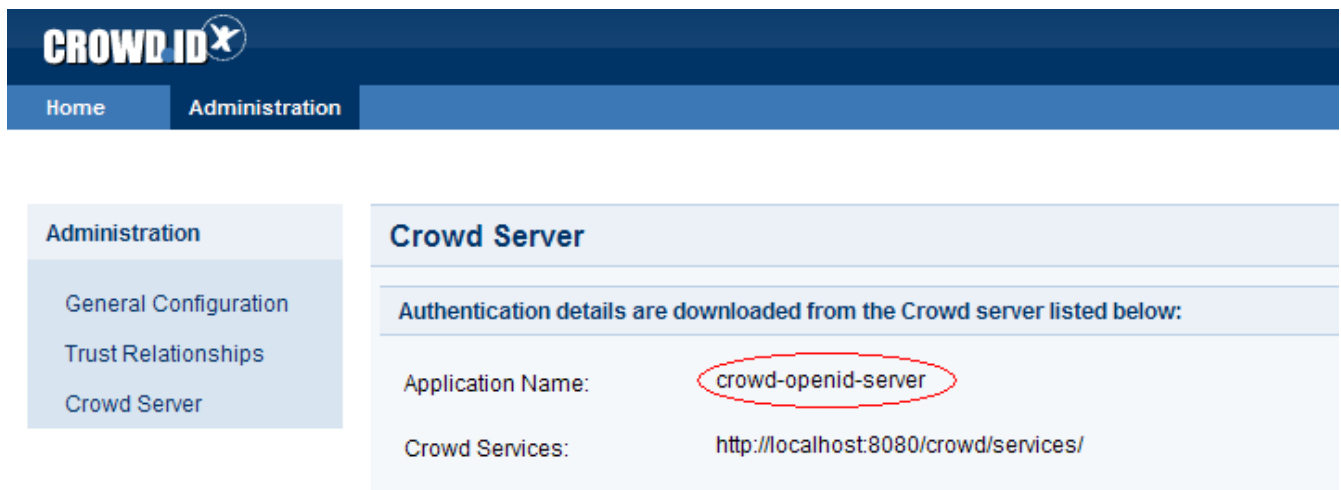
To change the details or users of your CrowdID application within Crowd, you will need to know the name by which your Crowd application is defined in your Crowd server.

To see the name of your CrowdID application,

1. Login to CrowdID.
2. Click the '**Administration**' link in the top navigation bar.
3. Click the '**Crowd Server**' link in the left navigation column.
4. This will display the '**Crowd Server**' details.

The '**Application Name**' field contains the name by which your CrowdID application is known to your Crowd server.

Screenshot: 'Application Name'



The screenshot shows the CrowdID Administration interface. At the top, there is a dark blue header with the 'CROWD ID' logo and a navigation bar with 'Home' and 'Administration' links. Below the navigation bar, there is a left sidebar with 'Administration' selected, containing links for 'General Configuration', 'Trust Relationships', and 'Crowd Server'. The main content area is titled 'Crowd Server' and contains the text 'Authentication details are downloaded from the Crowd server listed below:'. Below this, there are two rows of information: 'Application Name: crowd-openid-server' (where 'crowd-openid-server' is circled in red) and 'Crowd Services: http://localhost:8080/crowd/services/'.

RELATED TOPICS

- [1.1 How CrowdID works with Crowd](#)
 - [1.1.1 Determining the name of the CrowdID application](#)
 - [1.1.2 Locating the Crowd Server that CrowdID is using](#)
- [1.2 How OpenID sites interact with CrowdID](#)
- [1.3 Lightweight OpenID server](#)

[Crowd documentation](#)

1.1.2 Locating the Crowd Server that CrowdID is using

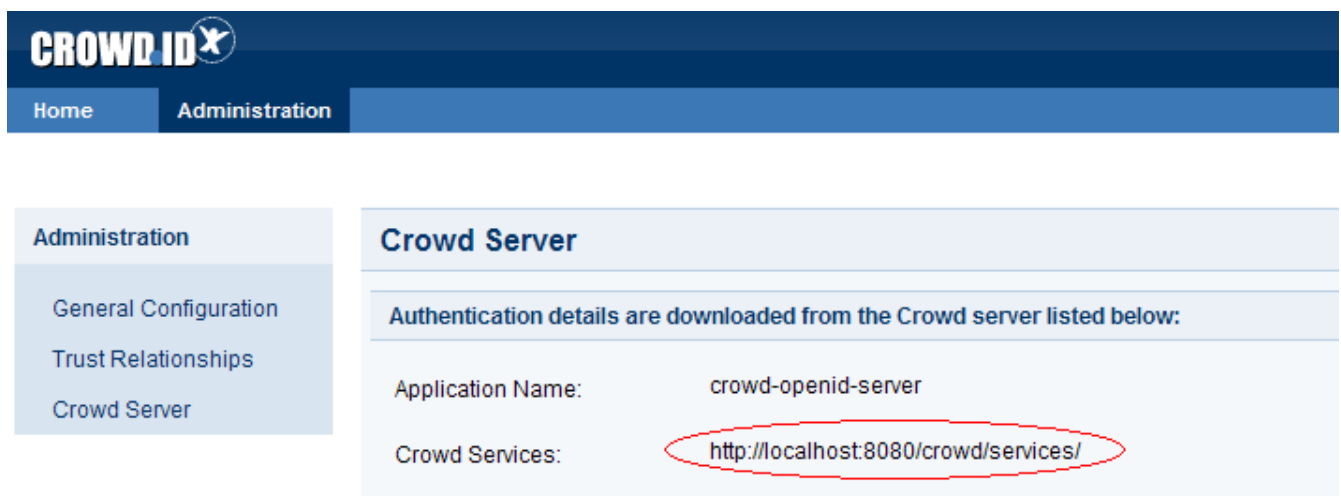
To change the details or users of your CrowdID application within Crowd, you will need to login to your Crowd server.

To determine the location of your Crowd server,

1. Login to CrowdID.
2. Click the '**Administration**' link in the top navigation bar.
3. Click the '**Crowd Server**' link in the left navigation column.
4. This will display the '**Crowd Server**' details.

The '**Crowd Services**' field contains the URL of your Crowd server. Go to this URL to login to Crowd.

Screenshot: 'Crowd Server'



The screenshot shows the CrowdID Administration interface. At the top, there is a dark blue header with the 'CROWD ID' logo and a navigation bar with 'Home' and 'Administration' tabs. Below the navigation bar, there is a left sidebar with 'Administration' selected, containing links for 'General Configuration', 'Trust Relationships', and 'Crowd Server'. The main content area is titled 'Crowd Server' and contains the text 'Authentication details are downloaded from the Crowd server listed below:'. Below this, there are two rows of information: 'Application Name: crowd-openid-server' and 'Crowd Services: http://localhost:8080/crowd/services/'. The URL 'http://localhost:8080/crowd/services/' is circled in red.

RELATED TOPICS

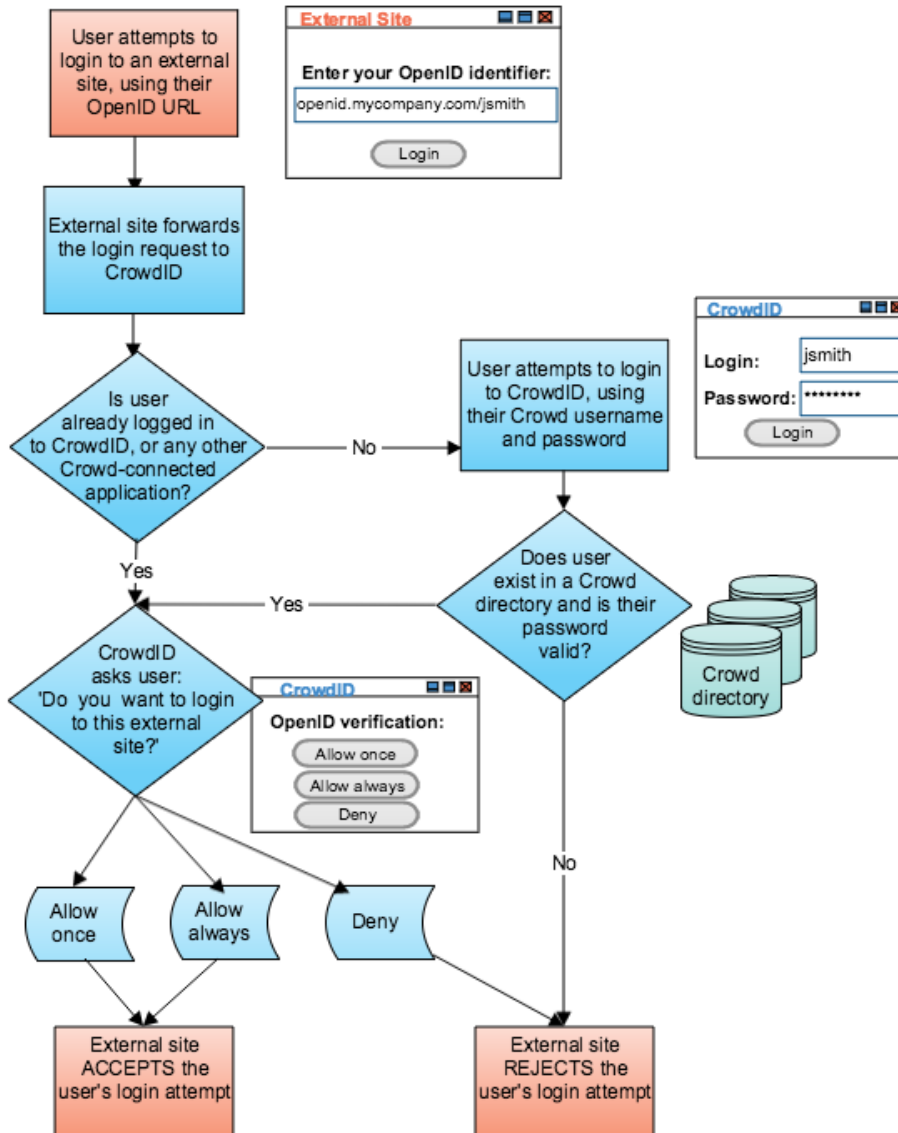
- [1.1 How CrowdID works with Crowd](#)
 - [1.1.1 Determining the name of the CrowdID application](#)
 - [1.1.2 Locating the Crowd Server that CrowdID is using](#)
- [1.2 How OpenID sites interact with CrowdID](#)
- [1.3 Lightweight OpenID server](#)

[Crowd documentation](#)

1.2 How OpenID sites interact with CrowdID

This diagram shows how an OpenID-enabled website (known as a 'Relying Party') interacts with CrowdID (an 'OpenID Provider') to validate an end-user's login attempt.

For more information about the OpenID protocol please see <http://openid.net>.



RELATED TOPICS

- [1.1 How CrowdID works with Crowd](#)
 - [1.1.1 Determining the name of the CrowdID application](#)
 - [1.1.2 Locating the Crowd Server that CrowdID is using](#)
- [1.2 How OpenID sites interact with CrowdID](#)
- [1.3 Lightweight OpenID server](#)

[Crowd documentation](#)

1.3 Lightweight OpenID server

Crowd 2.8 introduces a new lightweight UI-free OpenID server, in addition to the existing [OpenID server](#) that ships with Crowd.

It uses persistent identifiers unaffected by renaming, and can be accessed at `/openidserver/v2/op`. It is automatically installed when you install Crowd, and no database setup is necessary.

The OpenID server is a Crowd-connected application which authenticates against the directories configured in Crowd. If a user has already logged into any other Crowd-connected application (and single sign-on is enabled), they will not be prompted for any further login once they have entered their OpenID URL at an OpenID-enabled website.

You can deploy multiple OpenID servers against a single Crowd instance, which may be useful in larger deployments.

Configuration

The Lightweight OpenID server has no admin UI. You control the server using its [approval whitelist](#) configuration file. By default, the whitelist is empty so **no authentication will succeed** until you add URLs.

For communication with Crowd, see the `crowd.properties` file located by default in `crowd-openidserver-webapp/WEB-INF/classes`. You may also set the system property `crowd.openid.home` to point to another directory holding this configuration file.

2. Allowing users to access CrowdID

Granting access to CrowdID is done through Crowd. You can grant people rights to:

- [use CrowdID](#) Granting CrowdID access rights to a user allows them to use CrowdID to access OpenID websites and perform all the actions described in the [CrowdID User Guide](#).
- [administer CrowdID](#) Granting administration rights to a user allows them to use the '**Administration**' menu within CrowdID, which enables them to perform the actions described in the [CrowdID Administration Guide](#).

2.1 Granting CrowdID access rights to a user

Granting CrowdID access rights to a user allows them to use CrowdID to access OpenID websites and perform all the actions described in the [CrowdID User Guide](#).

Access to CrowdID is managed via Crowd. A user can only access CrowdID if they belong to a directory that is *mapped* to the CrowdID application within Crowd.

To grant CrowdID access rights to a particular user,

1. Login to your Crowd server.
2. View your CrowdID application as described in [Using the Application Browser](#) in the [Administration Guide](#).
3. Click the '**Directories**' tab to see a list of directories that are mapped to your CrowdID application. You will need to add the user to one of these directories.
4. If your directory capabilities permit, add the user to the directory via Crowd as described in [Adding a User](#) in the [Administration Guide](#). (Otherwise you may need to use your specific directory-management tool, instead of Crowd, to add the user to the directory.)

To grant CrowdID access rights to *all* the users in a particular directory,

1. Login to your Crowd server.
2. Map the directory to your CrowdID application as described in [Mapping a Directory to an Application](#) in the [Administration Guide](#).

To grant CrowdID access rights to a particular *group* of users within a directory,

1. Login to your Crowd server.
2. Map the group to your CrowdID application as described in [Specifying which Groups can access an Application](#) in the [Administration Guide](#).

To find your Crowd server's URL, see [1.1.2 Locating the Crowd Server that CrowdID is using](#).

To identify the name by which your CrowdID application is known within Crowd, see [1.1.1 Determining the name of the CrowdID application](#).

RELATED TOPICS

- [2.1 Granting CrowdID access rights to a user](#)
- [2.2 Granting CrowdID Administration Rights to a User](#)

[Crowd documentation](#)

RELATED TOPICS

- [2.1 Granting CrowdID access rights to a user](#)
- [2.2 Granting CrowdID Administration Rights to a User](#)

[Crowd documentation](#)

2.2 Granting CrowdID Administration Rights to a User

Granting administration rights to a user allows them to use the '**Administration**' menu within CrowdID, which enables them to perform the actions described in the [CrowdID Administration Guide](#).

CrowdID administration rights are managed via Crowd. To grant administration rights to a user, you need to add them to the '**crowd-administrators**' group as described below.

Note:

- Adding a user to the '**crowd-administrators**' group will also give them Crowd administration rights (unless you choose to use a different group to contain Crowd administrators). See [Granting Crowd Administration Rights to a User](#) in the [Administration Guide](#).
- The '**crowd-administrators**' group always contains CrowdID administrators, regardless of whether you are using it to contain Crowd administrators.

To grant administration rights to a user,

1. Log in to your Crowd server.
2. Click the '**Users**' tab in the top navigation bar.
3. This will display the [User Browser](#). Select the directory that contains the user to whom you wish to grant administration rights.
4. Use the '**Search**' to locate the user, then click the '**View**' link that corresponds to the user.
5. This will display the '**User Details**' screen. Click the '**Groups**' tab.
6. A list of the user's current groups (if any) will be displayed. Select the '**crowd-administrators**' group from the drop-down box below the list, then click the '**Add**' button.

To find your Crowd server's URL, see [1.1.2 Locating the Crowd Server that CrowdID is using](#).

Screenshot: Granting Crowd administration rights

The screenshot shows the 'View User - jane' interface. At the top, there are four tabs: 'Details', 'Attributes', 'Groups', and 'Roles'. The 'Groups' tab is selected. Below the tabs, the text reads: 'These are the groups the user is a member of.' Below this text is a table with two columns: 'Group' and 'Action'. Below the table, there is a dropdown menu showing 'crowd-administrators' and three buttons: 'Add »', 'Update »', and 'Cancel'.

RELATED TOPICS

- [2.1 Granting CrowdID access rights to a user](#)
- [2.2 Granting CrowdID Administration Rights to a User](#)

[Crowd documentation](#)

3. Specifying the sites to which users can log in

There are three ways to specify which OpenID hosts (i.e. websites or IP addresses) your users can log in to using their CrowdID:

- [No restriction](#) your CrowdID users can log in to any OpenID host
- [Blacklist](#) your CrowdID users can log in to any OpenID host except the one(s) that you specify
- [Whitelist](#) your CrowdID users can log in to only those OpenID host(s) that you specify

In addition, you may configured an [Approval Whitelist](#) for trusted sites,

3.1 Allowing all hosts

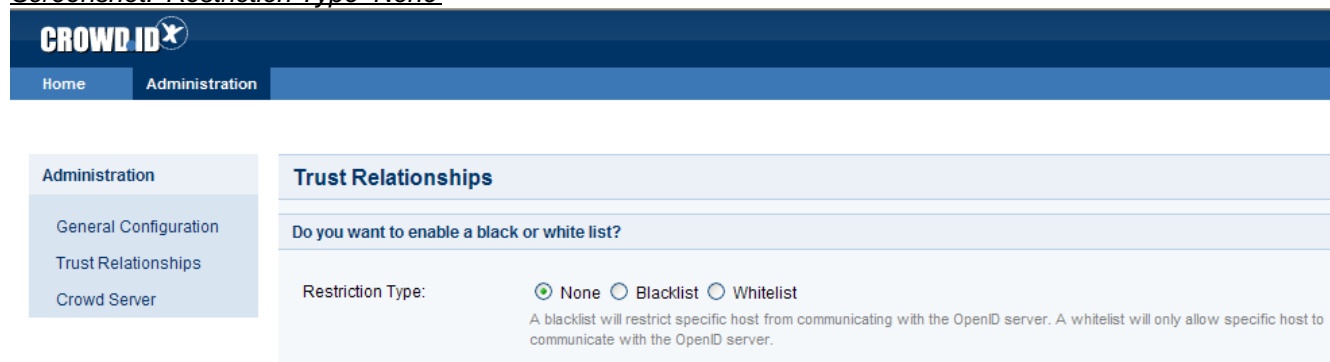
There are three ways to specify which OpenID hosts (i.e. websites or IP addresses) your users can log in to using their CrowdID:

- [No restriction](#) your CrowdID users can log in to any OpenID host
- [Blacklist](#) your CrowdID users can log in to any OpenID host except the one(s) that you specify
- [Whitelist](#) your CrowdID users can log in to only those OpenID host(s) that you specify

To allow users to login to any OpenID host,

1. Login to CrowdID.
2. Click the '**Administration**' link in the top navigation bar.
3. Click the '**Trust Relationships**' link in the left navigation column.
4. For '**Restriction Type**', select '**None**'.

Screenshot: 'Restriction Type None'



RELATED TOPICS

- [3.1 Allowing all hosts](#)
- [3.2 Allowing all except specified hosts \('Blacklist'\)](#)
- [3.3 Allowing specified hosts only \('Whitelist'\)](#)
- [3.4 Approval Whitelist](#)

[Crowd documentation](#)

3.2 Allowing all except specified hosts ('Blacklist')

There are three ways to specify which OpenID hosts (i.e. websites or IP addresses) your users can log in to using their CrowdID:

- [No restriction](#) your CrowdID users can log in to any OpenID host
- [Blacklist](#) your CrowdID users can log in to any OpenID host except the one(s) that you specify
- [Whitelist](#) your CrowdID users can log in to only those OpenID host(s) that you specify

To specify an OpenID blacklist,

1. Login to CrowdID.
2. Click the '**Administration**' link in the top navigation bar.
3. Click the '**Trust Relationships**' link in the left navigation column.
4. For '**Restriction Type**', select '**Blacklist**'.
5. Wait for a section titled '**Blacklist mode: hosts that can not login**' to appear on the screen.
6. For each site to which you want to prevent users logging in,
 - a. Type the URL or IP address in the '**Address**' field.
 - b. Click the '**Add**' button.

Screenshot: 'Restriction Type Blacklist'

The screenshot shows the CrowdID Administration interface. The top navigation bar includes 'Home' and 'Administration'. The left sidebar shows 'Administration' with sub-links for 'General Configuration', 'Trust Relationships', and 'Crowd Server'. The main content area is titled 'Trust Relationships' and contains the following elements:

- A question: 'Do you want to enable a black or white list?'
- 'Restriction Type:' with radio buttons for 'None', 'Blacklist' (selected), and 'Whitelist'.
- A note: 'A blacklist will restrict specific host from communicating with the OpenID server. A whitelist will only allow specific host to communicate with the OpenID server.'
- A section titled 'Blacklist mode: hosts that can not login.'
- A table with columns 'Address' and 'Action':

Address	Action
www.waste-of-space.com	Remove
- An input field for 'Address:' containing 'www.waste-of-time.com' and an 'Add »' button.

RELATED TOPICS

- [3.1 Allowing all hosts](#)
- [3.2 Allowing all except specified hosts \('Blacklist'\)](#)
- [3.3 Allowing specified hosts only \('Whitelist'\)](#)
- [3.4 Approval Whitelist](#)

[Crowd documentation](#)

3.3 Allowing specified hosts only ('Whitelist')

There are three ways to specify which OpenID hosts (i.e. websites or IP addresses) your users can log in to using their CrowdID:

- [No restriction](#) your CrowdID users can log in to any OpenID host
- [Blacklist](#) your CrowdID users can log in to any OpenID host except the one(s) that you specify
- [Whitelist](#) your CrowdID users can log in to only those OpenID host(s) that you specify

To specify an OpenID whitelist,

1. Login to CrowdID.
2. Click the '**Administration**' link in the top navigation bar.
3. Click the '**Trust Relationships**' link in the left navigation column.
4. For '**Restriction Type**', select '**Blacklist**'.
5. Wait for a section titled '**Whitelist mode: hosts that can login**' to appear on the screen.
6. For each site to which you want to allow users to login,
 - a. Type the URL or IP address in the '**Address**' field.
 - b. Click the '**Add**' button.

Screenshot: '*Restriction Type Whitelist*'

The screenshot shows the CrowdID Administration interface. The top navigation bar includes 'Home' and 'Administration'. The left sidebar lists 'Administration', 'General Configuration', 'Trust Relationships', and 'Crowd Server'. The main content area is titled 'Trust Relationships' and contains the following elements:

- A question: 'Do you want to enable a black or white list?'
- 'Restriction Type:' with three radio buttons: 'None', 'Blacklist', and 'Whitelist' (which is selected).
- A note: 'A blacklist will restrict specific host from communicating with the OpenID server. A whitelist will only allow specific host to communicate with the OpenID server.'
- A section titled 'Whitelist mode: hosts that can login.' containing a table with the following data:

Address	Action
www.mycompany.com	Remove

At the bottom of the page, there is an 'Address:' input field containing 'www.trusted-company.com' and an 'Add »' button.

RELATED TOPICS

- [3.1 Allowing all hosts](#)
- [3.2 Allowing all except specified hosts \('Blacklist'\)](#)
- [3.3 Allowing specified hosts only \('Whitelist'\)](#)
- [3.4 Approval Whitelist](#)

[Crowd documentation](#)

3.4 Approval Whitelist

For trusted sites, such as internal services, you may wish to simplify the user experience by automatically approving authentication requests. Users will not be prompted to verify authentication requests from these realms.

The OpenID Verification page ([2.4 Allowing or denying a login](#)) shows the realm that a host is using.

This configuration is stored in a file:

```
crowd-openidserver-webapp/WEB-INF/classes/crowdid.approval-whitelist
```

Each line is a single OpenID realm. If an authentication request is received from a site on that list it will automatically be approved as if the user had selected 'Allow Always'.

Example

In the default configuration, using the demo OpenID client to authenticate, the OpenID server will present an [OpenID verification page](#):

*The following site:
<http://localhost:8095/openidclient>
has requested that you confirm the following address as your personal identity*

Adding:

```
http://localhost:8095/openidclient
```

to the approval whitelist would automatically approve the demo OpenID client for all users.

RELATED TOPICS

- [3.1 Allowing all hosts](#)
- [3.2 Allowing all except specified hosts \('Blacklist'\)](#)
- [3.3 Allowing specified hosts only \('Whitelist'\)](#)
- [3.4 Approval Whitelist](#)

[Crowd documentation](#)

4. Configuring CrowdID system settings

- [4.1 Specifying the CrowdID URL](#)
- [4.2 Enabling localhost authentication](#)
- [4.3 Enabling immediate authentication requests](#)
- [4.4 Enabling communication with stateless clients](#)

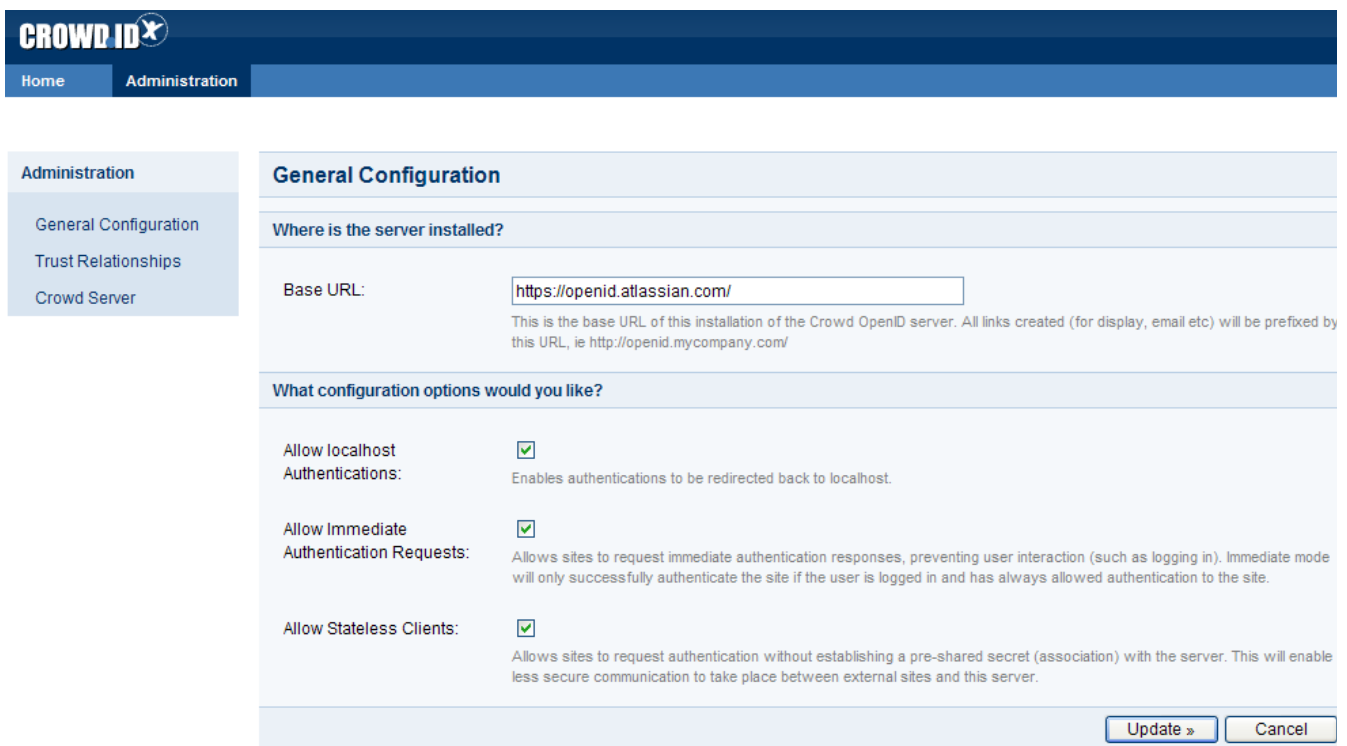
4.1 Specifying the CrowdID URL

The **CrowdID URL** is the URL that your end-users will type when [logging into OpenID-enabled websites](#).

To define the URL of your CrowdID instance,

1. Login to CrowdID.
2. Click the '**Administration**' link in the top navigation bar.
3. Click the '**General Configuration**' link in the left navigation column.
4. Type the URL into the '**Base URL**' field.
5. Click the '**Update**' button.

Screenshot: 'General Configuration'



RELATED TOPICS

- [4.1 Specifying the CrowdID URL](#)
- [4.2 Enabling localhost authentication](#)
- [4.3 Enabling immediate authentication requests](#)
- [4.4 Enabling communication with stateless clients](#)

[Crowd documentation](#)

4.2 Enabling localhost authentication

Enabling **localhost authentication** prevents OpenID-enabled sites from directly accessing your end-users' local machines.

To enable localhost authentication,

1. Login to CrowdID.
2. Click the '**Administration**' link in the top navigation bar.
3. Click the '**General Configuration**' link in the left navigation column.
4. Select the '**Allow localhost authentications**' checkbox.
5. Click the '**Update**' button.

Screenshot: 'General Configuration'

CROWD ID

Home Administration

Administration

- General Configuration
- Trust Relationships
- Crowd Server

General Configuration

Where is the server installed?

Base URL:

This is the base URL of this installation of the Crowd OpenID server. All links created (for display, email etc) will be prefixed by this URL, ie http://openid.mycompany.com/

What configuration options would you like?

Allow localhost Authentications: Enables authentications to be redirected back to localhost.

Allow Immediate Authentication Requests: Allows sites to request immediate authentication responses, preventing user interaction (such as logging in). Immediate mode will only successfully authenticate the site if the user is logged in and has always allowed authentication to the site.

Allow Stateless Clients: Allows sites to request authentication without establishing a pre-shared secret (association) with the server. This will enable less secure communication to take place between external sites and this server.

Update » Cancel

RELATED TOPICS

- [4.1 Specifying the CrowdID URL](#)
- [4.2 Enabling localhost authentication](#)
- [4.3 Enabling immediate authentication requests](#)
- [4.4 Enabling communication with stateless clients](#)

[Crowd documentation](#)

RELATED TOPICS

- [4.1 Specifying the CrowdID URL](#)
- [4.2 Enabling localhost authentication](#)
- [4.3 Enabling immediate authentication requests](#)
- [4.4 Enabling communication with stateless clients](#)

[Crowd documentation](#)

4.3 Enabling immediate authentication requests

Enabling 'Allow immediate authentication requests' allows an OpenID-enabled site to check whether the user is logged in, without actually prompting the user to login. Known as *pass-through authentication*, this provides greater convenience for end-users, particularly when an end-user visits a site for which they have previously selected 'Allow Always' (see [2.4 Allowing or denying a login](#) in the *CrowdID User Guide*).

To enable 'Allow immediate authentication requests',

1. Login to CrowdID.
2. Click the 'Administration' link in the top navigation bar.
3. Click the 'General Configuration' link in the left navigation column.
4. Select the 'Allow immediate authentication requests' checkbox.
5. Click the 'Update' button.

Screenshot: 'General Configuration'

The screenshot shows the CrowdID Administration interface. At the top, there is a navigation bar with 'Home' and 'Administration' links. Below this, a sidebar on the left contains 'Administration', 'General Configuration', 'Trust Relationships', and 'Crowd Server'. The main content area is titled 'General Configuration' and is divided into two sections. The first section, 'Where is the server installed?', contains a 'Base URL' field with the value 'https://openid.atlassian.com/'. Below the field is a note: 'This is the base URL of this installation of the Crowd OpenID server. All links created (for display, email etc) will be prefixed by this URL, ie http://openid.mycompany.com/'. The second section, 'What configuration options would you like?', contains three checkboxes, all of which are checked: 'Allow localhost Authentications' (with a note: 'Enables authentications to be redirected back to localhost.'), 'Allow Immediate Authentication Requests' (with a note: 'Allows sites to request immediate authentication responses, preventing user interaction (such as logging in). Immediate mode will only successfully authenticate the site if the user is logged in and has always allowed authentication to the site.'), and 'Allow Stateless Clients' (with a note: 'Allows sites to request authentication without establishing a pre-shared secret (association) with the server. This will enable less secure communication to take place between external sites and this server.'). At the bottom right of the form are 'Update »' and 'Cancel' buttons.

RELATED TOPICS

- [4.1 Specifying the CrowdID URL](#)
- [4.2 Enabling localhost authentication](#)
- [4.3 Enabling immediate authentication requests](#)
- [4.4 Enabling communication with stateless clients](#)

[Crowd documentation](#)

RELATED TOPICS

- [4.1 Specifying the CrowdID URL](#)
- [4.2 Enabling localhost authentication](#)
- [4.3 Enabling immediate authentication requests](#)
- [4.4 Enabling communication with stateless clients](#)

[Crowd documentation](#)

4.4 Enabling communication with stateless clients

Some OpenID-enabled sites do not support pre-shared secrets (associations). Selecting **allow stateless clients** enables your CrowdID server to communicate with such sites.

To allow stateless clients,

1. Login to CrowdID.
2. Click the '**Administration**' link in the top navigation bar.
3. Click the '**General Configuration**' link in the left navigation column.
4. Select the '**Allow stateless clients**' checkbox.
5. Click the '**Update**' button.

Screenshot: 'General Configuration'

CROWD ID

Home Administration

Administration

- General Configuration
- Trust Relationships
- Crowd Server

General Configuration

Where is the server installed?

Base URL:

This is the base URL of this installation of the Crowd OpenID server. All links created (for display, email etc) will be prefixed by this URL, ie http://openid.mycompany.com/

What configuration options would you like?

Allow localhost Authentications: Enables authentications to be redirected back to localhost.

Allow Immediate Authentication Requests: Allows sites to request immediate authentication responses, preventing user interaction (such as logging in). Immediate mode will only successfully authenticate the site if the user is logged in and has always allowed authentication to the site.

Allow Stateless Clients: Allows sites to request authentication without establishing a pre-shared secret (association) with the server. This will enable less secure communication to take place between external sites and this server.

RELATED TOPICS

- [4.1 Specifying the CrowdID URL](#)
- [4.2 Enabling localhost authentication](#)
- [4.3 Enabling immediate authentication requests](#)
- [4.4 Enabling communication with stateless clients](#)

[Crowd documentation](#)

RELATED TOPICS

- [4.1 Specifying the CrowdID URL](#)
- [4.2 Enabling localhost authentication](#)
- [4.3 Enabling immediate authentication requests](#)
- [4.4 Enabling communication with stateless clients](#)

[Crowd documentation](#)

CrowdID User Guide

i With Crowd comes **CrowdID**, your OpenID provider. **CrowdID** is an [Atlassian](#) product which allows you to use a single login for all OpenID-enabled websites.

This means that you don't have to remember a separate username and password for each different site that you visit. You can just use your OpenID for all of them.

You can use CrowdID if your administrator has installed it for your organization. For instructions on setting up CrowdID, please see the [CrowdID Administration Guide](#).

The *CrowdID User Guide* tells you how to

- Log in to websites using CrowdID.
- Instruct CrowdID to always allow login to a specific site.
- Set up your own profile(s) within CrowdID.
- Use CrowdID to change your password.

Contents of the *CrowdID User Guide*

- [1. Getting started with CrowdID](#)
 - [1.1 What is OpenID?](#)
 - [1.2 What is CrowdID?](#)
 - [1.3 What is an OpenID URL or identifier?](#)
 - [1.4 Viewing the CrowdID page](#)
- [2. Logging in to a website using OpenID](#)
 - [2.1 Does the website support OpenID?](#)
 - [2.2 Entering your OpenID URL](#)
 - [2.3 Logging in to CrowdID](#)
 - [2.4 Allowing or denying a login](#)
 - [2.5 Providing additional profile information to a website](#)
- [3. Viewing your always-approved websites](#)
- [4. Viewing your login history](#)
- [5. Updating your profile](#)
- [6. Using more than one profile](#)
 - [6.1 Adding a profile](#)
 - [6.2 Choosing a profile for a website](#)
 - [6.3 Setting a default profile](#)
 - [6.4 Deleting a profile](#)
- [7. Changing or resetting your password](#)
 - [7.1 Changing your password](#)
 - [7.2 Resetting your password](#)
- [8. Requesting Forgotten Usernames](#)

1. Getting started with CrowdID

CrowdID is an [Atlassian](#) product which allows you to use a single login for all OpenID-enabled websites.

This means that you don't have to remember a separate username and password for each different site that you visit. You can just use your OpenID for all of them.

You can use CrowdID if your administrator has installed it for your organization.

- [1.1 What is OpenID?](#)
- [1.2 What is CrowdID?](#)
- [1.3 What is an OpenID URL or identifier?](#)
- [1.4 Viewing the CrowdID page](#)

1.1 What is OpenID?

The term '**OpenID**' has two meanings:

- The OpenID protocol, described below.
- Your own [identifier or URL](#).

[OpenID](#) is an open, free protocol which allows you to use a single [identifier](#) to log in to any OpenID-enabled website. OpenID allows the website to communicate with your OpenID provider (e.g. your organization's [CrowdID](#) server) when attempting to verify your login.



Do you have a zillion usernames and passwords, which you use for logging in to blogs and websites all over the place? **OpenID** allows you to throw them all away, for all websites that support it. More and more sites are coming on board.

RELATED TOPICS

- [1.1 What is OpenID?](#)
- [1.2 What is CrowdID?](#)
- [1.3 What is an OpenID URL or identifier?](#)
- [1.4 Viewing the CrowdID page](#)

[CrowdID User Guide](#)

1.2 What is CrowdID?

CrowdID is an [Atlassian](#) product which makes use of the [OpenID](#) protocol to allow you to use a single login for a number of websites. To put it another way: CrowdID is an '**OpenID provider**'. You can use CrowdID if your administrator has installed it for your organization.

This means that you can:

- Securely store your username and password on your organization's server.
- Use your [OpenID](#) as a single identifier to log in to all websites which support OpenID.
- Control how you allow or deny login requests from websites.



Your organization can use **CrowdID** to set up an internal OpenID provider. There are also other OpenID providers, where you can get a free OpenID.

RELATED TOPICS

- [1.1 What is OpenID?](#)
- [1.2 What is CrowdID?](#)
- [1.3 What is an OpenID URL or identifier?](#)
- [1.4 Viewing the CrowdID page](#)

[CrowdID User Guide](#)

1.3 What is an OpenID URL or identifier?

To log in to an OpenID-enabled website you need an OpenID identifier, also called an OpenID URL or simply an OpenID. Your OpenID is a URL (web address) which points to your organization's CrowdID server. Here are some examples of what your OpenID may look like:

```
http://my.server.name/myname  
http://myname.mysite.com
```

To find your OpenID URL, you can:

- Ask your system administrator, or
- Click the 'My OpenID' link on the 'Home' tab of the [CrowdID page](#).

Endpoint URLs

You can also use CrowdID's **endpoint URL** to log in to OpenID-enabled websites. The OpenID URLs for specific users may look like:

```
http://my.server.name/openidserver/users/myname  
http://my.server.name/openidserver/users/anothername
```

Crowd also provides an endpoint URL that you can pass to login sites to have your identifier automatically selected:

```
http://my.server.name/openidserver/op
```

RELATED TOPICS

- [1.1 What is OpenID?](#)
- [1.2 What is CrowdID?](#)
- [1.3 What is an OpenID URL or identifier?](#)
- [1.4 Viewing the CrowdID page](#)

[CrowdID User Guide](#)

1.4 Viewing the CrowdID page

The **CrowdID** page allows you to:

- View your [OpenID URL](#).
- Set up your [profile\(s\)](#).
- View your list of [always-approved sites](#).
- View your [login history](#).
- [Resume approval](#) of a login. (This option appears only during a login process, if you move away from the 'OpenID Verification' page.)
- Change your [password](#).

There are two ways to access the CrowdID page:

- While you are logging in to another site.
- Directly via the CrowdID URL.

To access the CrowdID page while you are logging in to another site,

1. Use your OpenID to [log in](#) to the website you want to visit.
2. [Log in to CrowdID](#) if prompted.
3. The CrowdID **'OpenID Verification'** page will appear, provided that you have not previously added the website to your list of always-approved sites. You can choose any of the CrowdID options on the left-hand navigation panel, even during the login process.
4. When you have finished your tasks in CrowdID, you can [resume the login](#).

To access CrowdID directly via the CrowdID URL,

1. Ask your administrator for the CrowdID address (URL) as configured for your organization.
2. Type or paste the address into the address or navigation bar of your internet browser.
3. The [CrowdID Login page](#) will appear. Type in your username and password.
4. Click the **'Login'** button.
5. The CrowdID **'My OpenID'** page will appear. The CrowdID options are displayed in the left-hand navigation panel and top menu bar.

Screenshot: CrowdID My OpenID page



RELATED TOPICS

- [1.1 What is OpenID?](#)
- [1.2 What is CrowdID?](#)
- [1.3 What is an OpenID URL or identifier?](#)
- [1.4 Viewing the CrowdID page](#)

[CrowdID User Guide](#)


2. Logging in to a website using OpenID

CrowdID enables you to log in to a website using your [OpenID](#). The login process depends upon the following:

- Have you logged in to CrowdID already during this browser session?
- Have you previously added the website to your list of always-approved sites?
- Does the website you are visiting require additional profile information?

Steps in the login process:


1. [Find the OpenID login](#) page or section on the website you want to visit.
2. [Enter your OpenID](#) and click the login button.
3. If prompted, [log in to CrowdID](#). (Required if you have not already logged in during this browser session.)
4. If prompted, instruct CrowdID to [allow the website login](#). (Required if you have not previously added the website to your list of always-approved sites.)
5. If prompted, [supply additional profile information](#). (Required if the website you are visiting wants more information.)

 The login process can be very simple: just the first two steps above, provided that you have already logged in to CrowdID this session and have already added the website to your list of always-approved sites.

2.1 Does the website support OpenID?

You can only use your OpenID (also called an [OpenID URL or identifier](#)) to log in to a website if the site supports the [OpenID](#) protocol. The number of websites that support OpenID is growing rapidly.

To see if a particular website supports OpenID, check the site's login page for one or more of the following:

- The word 'OpenID'.
- The OpenID logo 

RELATED TOPICS

- [2.1 Does the website support OpenID?](#)
- [2.2 Entering your OpenID URL](#)
- [2.3 Logging in to CrowdID](#)
- [2.4 Allowing or denying a login](#)
- [2.5 Providing additional profile information to a website](#)

[CrowdID User Guide](#)

2.2 Entering your OpenID URL

With CrowdID, you can use your 'OpenID' (also called an [OpenID URL or identifier](#)) to log in to a website that [supports the OpenID protocol](#).

To log in to a website which supports OpenID,

1. Go to the login page of the website you want to visit.
2. [Look for](#) the OpenID login section.
 - 📘 Sometimes the OpenID login will be on the same page as the standard login. Other sites will have a separate OpenID login page.
3. Type or paste [your OpenID](#) into the login text box.
 - 📘 Usually, you must enter the full OpenID. In some sites, you can enter the OpenID without 'http://'
4. Click the login button. The button will probably be labeled 'Log in', 'Sign in' or 'Go'.

One of the following things will happen now:

- If you have not already logged in to CrowdID during this browser session, you will see the CrowdID [login page](#).
- If you have already logged in to CrowdID and you have previously instructed CrowdID to allow this website always, then you will be logged straight into the website.
- If you have already logged in to CrowdID but have not previously set this site to "Allow Always", then CrowdID will ask you to [approve the login](#).
- If your administrator has blocked access to this website, CrowdID will display an 'OpenID Verification Error' message.

RELATED TOPICS

- [2.1 Does the website support OpenID?](#)
- [2.2 Entering your OpenID URL](#)
- [2.3 Logging in to CrowdID](#)
- [2.4 Allowing or denying a login](#)
- [2.5 Providing additional profile information to a website](#)

[CrowdID User Guide](#)

2.3 Logging in to CrowdID

CrowdID will ask you to log in, if you have not already done so during this browser session or if your session has timed out. The CrowdID login may appear during the process of logging in to another website, or when you are accessing CrowdID directly.

To log in to CrowdID,

1. Type in your username and password.
2. Click the **'Login'** button.

You can [reset your password](#), if you have forgotten it.

Screenshot: CrowdID login page

CROWD ID [Help](#)

[Home](#)

Login

Username: *

Password: *

[Forgotten your password?](#)

Powered by [Atlassian Crowd](#) Version: 1.1.0 (Build:#153 - Jun 13, 2007) [Report a bug](#) | [Request a feature](#) | [Contact Atlassian](#)

If you are in the process of logging in to another web site, CrowdID will now ask you to [approve the login](#).

RELATED TOPICS

- [2.1 Does the website support OpenID?](#)
- [2.2 Entering your OpenID URL](#)
- [2.3 Logging in to CrowdID](#)
- [2.4 Allowing or denying a login](#)
- [2.5 Providing additional profile information to a website](#)

[CrowdID User Guide](#)

2.4 Allowing or denying a login

When you use your OpenID to log in to a website, CrowdID will present the **'OpenID Verification'** page where you can allow or deny the login.

✔ If you have previously instructed CrowdID to allow this site always, you will not see this page. You can remove a site from the ['Allow Always' list](#) in CrowdID.

You can instruct CrowdID to:

- [Allow the login](#) for this session only (**'Allow Once'**).
- [Allow login](#) to this site every time you use your OpenID (**'Allow Always'**).
- [Refuse login](#) to this site (**'Deny'**).
- [Use a specific profile](#).

If you move away from the 'OpenID Verification' page within CrowdID, you can go back to the page and [resume approval](#).

Screenshot: OpenID Verification page

CROWD ID User: Sarah Maddox | [Logoff](#) | [Change Password](#) | [Help](#)

Home Administration

My Identity

- My OpenID
- Profiles
- Approved Sites
- Login History

OpenID Verification

The following site:

<http://teamtastic.com/>

has requested that you confirm the following address as your personal identity:

<https://someopenidserver.com/somename>

and is requesting the following information:

email nickname

Allow Once

Allow Always

Deny

Select Profile

Use this profile:

Nickname	smaddox
Full Name	Sarah Maddox
Email	smaddox@atlassian.com
Country	United States
Language	English

Powered by Atlassian CrowdID Version: 1.1.0 (Build:#161 - Jun 19, 2007) [Report a bug](#) | [Request a feature](#) | [Contact Atlassian](#)

To allow the login for this session only,

1. Click **'Allow Once'** on the right of the CrowdID 'OpenID Verification' page.
2. CrowdID will send you back to the original site, passing your profile information as well as the confirmed login. The website you are visiting may ask you to [complete your profile information](#).

To allow login to this site every time you use your OpenID,

1. Click '**Allow Always**' on the right of the CrowdID 'OpenID Verification' page.
2. CrowdID will add the website to your list of [approved sites](#) and send you back to the original site, passing your profile information as well as the confirmed login. The website you are visiting may ask you to [complete your profile information](#).


To refuse login to this site,

1. Click '**Deny**' on the right of the CrowdID 'OpenID Verification' page.
2. CrowdID will send you back to the original site and refuse the login. The original site will probably show a message something like 'Verification canceled'.

To use a specific profile,

1. If you have defined [more than one profile](#), you can choose a specific profile for the website you are visiting. Select a profile from the dropdown list labeled '**Use this profile**' on the CrowdID 'OpenID Verification' page.
2. The profile details will change in the 'Select Profile' section of the page. CrowdID will pass these profile details to the website when you [allow the login](#).

To go back to the 'OpenID Verification' page and resume approval,

1. Click '**Resume Approval**' in the left-hand navigation panel.
 This option will appear if you move away from the 'OpenID Verification' page during the login process.
2. CrowdID will return to the '**OpenID Verification**' page, where you can [allow the login](#).


RELATED TOPICS

- [2.1 Does the website support OpenID?](#)
- [2.2 Entering your OpenID URL](#)
- [2.3 Logging in to CrowdID](#)
- [2.4 Allowing or denying a login](#)
- [2.5 Providing additional profile information to a website](#)

[CrowdID User Guide](#)


2.5 Providing additional profile information to a website

When you [log in](#) to a website using your OpenID, CrowdID passes your [profile information](#) to the website. Some websites will then log you in immediately, while other websites may ask you to confirm or complete the profile information.

 You are now outside CrowdID. Any dialog here is between you and the website you are visiting.

To provide additional profile information to a website,

1. Check the profile information displayed, and add extra information as you wish.
2. Click the button or other option supplied by the website to complete the login process.

 You can [change your profile information](#) and [define more than one profile](#) in CrowdID.

RELATED TOPICS

- [2.1 Does the website support OpenID?](#)
- [2.2 Entering your OpenID URL](#)
- [2.3 Logging in to CrowdID](#)
- [2.4 Allowing or denying a login](#)
- [2.5 Providing additional profile information to a website](#)

[CrowdID User Guide](#)

3. Viewing your always-approved websites

When logging in to a website, you can instruct CrowdID to [allow login](#) to the site every time you use your OpenID ('**Allow Always**').

The CrowdID '**Approved Sites**' page allows you to:

- [View your list](#) of always-approved sites.
- [Remove a site](#) from the list.
- [Choose a profile](#) for use when logging in to a site.





- If you have never instructed CrowdID to '[Allow Always](#)' for any sites, The 'Approved Sites' page will display a message like '**You currently have no approved sites.**'
- You can [add profiles](#) on the CrowdID 'Profiles' page.


To view your list of always-approved sites,

1. [Access CrowdID](#).
2. Click '**Approved Sites**' in the left-hand navigation panel.

To remove a site from the list,

1. [Access CrowdID](#).
2. Click '**Approved Sites**' in the left-hand navigation panel.
3. Your list of always-approved sites will appear. Click the remove button  next to the site which you want to remove.
4. Click the '**Apply**' button.
5. '**Update Successful**' message is displayed.
 -  If you do not click the 'Apply' button, your changes will be canceled.

To choose a profile for use when logging in to a site,

1. [Access CrowdID](#).
2. Click '**Approved Sites**' in the left-hand navigation panel.
3. Your list of always-approved sites will appear. Select the profile you want from the dropdown list next to the applicable site.
4. Click the '**Apply**' button.
5. '**Update Successful**' message is displayed.
 -  If you do not click the 'Apply' button, your changes will be canceled.

Screenshot: CrowdID Approved Sites page

CROWD ID User: Sarah Maddox [Logoff](#) | [Change Password](#) | [Help](#)

Home Administration

My Identity

- My OpenID
- Profiles
- Approved Sites
- Login History

Approved Sites

The sites below are automatically allowed to authenticate with the specified profile. Sites are added to this approved list if the "Allow Always" button is clicked when an external site requests OpenID authentication.

Site URL	Profile
http://www.wooblelab.com/	june
http://claimid.com/	june
http://wikitravel.org/en/	My Profile
https://www.hampr.com	sm2

Apply Cancel

Powered by Atlassian CrowdID Version: 1.1.0 (Build:#161 - Jun 19, 2007) [Report a bug](#) | [Request a feature](#) | [Contact Atlassian](#)

RELATED TOPICS

- [1. Getting started with CrowdID](#)
- [2. Logging in to a website using OpenID](#)
- [3. Viewing your always-approved websites](#)
- [4. Viewing your login history](#)
- [5. Updating your profile](#)
- [6. Using more than one profile](#)
- [7. Changing or resetting your password](#)
- [8. Requesting Forgotten Usernames](#)

4. Viewing your login history

The CrowdID **Login History** page displays a list of the sites you have visited and the type of approval you gave on each visit:

- **'Allow Always'** - At the time of this login, you instructed CrowdID to allow login to the site every time you use your OpenID.
- **'(Auto) Allow Always'** - This login was allowed automatically, because you have previously instructed CrowdID to allow login to the site every time you use your OpenID.
- **'Allow Once'** - You instructed CrowdID to allow login to the site at that time only.
- **'Deny'** - You instructed CrowdID to refuse the login to the site at that time.

To view your login history,

1. [Access CrowdID](#).
2. Click **Login History** in the left-hand navigation panel.

i If you have used your OpenID many times, the login history items will be shown on more than one page. To move from one page to another, click the page numbers or the **Next** and **Prev** links at the bottom of the page.

Screenshot: CrowdID Login History page

CROWD ID
User: Sarah Maddox | [Logoff](#) | [Change Password](#) | [Help](#)

Home
Administration

My Identity

[My OpenID](#)

[Profiles](#)

[Approved Sites](#)

[Login History](#)

Login History

The following is a record of all authentication activity with external sites with your account.

Time	URL	Action
19-06-2007 09:24:18	http://wikitravel.org/en/	● Allow Always
19-06-2007 09:23:32	http://www.wooblelab.com/	● Allow Always
19-06-2007 09:10:11	http://wikitravel.org/en/	● Allow Always
19-06-2007 12:49:03	http://www.wooblelab.com/	● Allow Once
19-06-2007 12:46:42	http://www.wooblelab.com/	● Allow Once
18-06-2007 11:30:06	http://wikitravel.org/en/	● (Auto) Allow Always
18-06-2007 11:28:50	http://wikitravel.org/en/	● Allow Always
18-06-2007 11:26:06	http://wikitravel.org/en/	● Allow Always
18-06-2007 11:12:21	http://teamtastic.com/	● Allow Once
17-06-2007 11:35:46	http://*.openid.net/	● Deny
17-06-2007 11:25:46	https://www.hampr.com	● Allow Once
17-06-2007 11:24:23	https://www.hampr.com	● Allow Once
17-06-2007 09:25:07	http://wikitravel.org/en/	● (Auto) Allow Always
17-06-2007 09:22:15	http://wikitravel.org/en/	● (Auto) Allow Always
17-06-2007 09:20:34	http://wikitravel.org/en/	● Allow Always
17-06-2007 09:17:55	http://wikitravel.org/en/	● Allow Once
17-06-2007 09:12:06	http://wikitravel.org/en/	● (Auto) Allow Always
17-06-2007 08:53:41	http://claimid.com/	● Allow Always
17-06-2007 08:51:03	http://www.wooblelab.com/	● (Auto) Allow Always
17-06-2007 08:49:28	http://www.wooblelab.com/	● Allow Always
15-06-2007 12:54:36	http://www.wooblelab.com/	● Allow Always
15-06-2007 12:40:47	http://wikitravel.org/en/	● Allow Always
14-06-2007 08:57:06	http://www.wooblelab.com/	● Allow Once
14-06-2007 08:45:41	http://www.wooblelab.com/	● Deny
14-06-2007 08:43:36	http://wikitravel.org/en/	● Deny

1 2 [Next >>](#)

Powered by [Atlassian CrowdID](#) Version: 1.2-SNAPSHOT (Build:#180 - Jun 22, 2007)
[Report a bug](#) | [Request a feature](#) | [Contact Atlassian](#)

RELATED TOPICS

- [1. Getting started with CrowdID](#)
- [2. Logging in to a website using OpenID](#)
- [3. Viewing your always-approved websites](#)
- [4. Viewing your login history](#)
- [5. Updating your profile](#)
- [6. Using more than one profile](#)
- [7. Changing or resetting your password](#)
- [8. Requesting Forgotten Usernames](#)

5. Updating your profile

When you log in to a website using your OpenID, CrowdID will pass some information to the website. The information is copied from your profile on CrowdID. When your profile is first created, CrowdID will auto-fill the information where possible, by copying:

- Country and language from the language information in your browser.
- Name and email address from your organization's user directory.

You can update your profile information on CrowdID, as described below.

You can also:

- [Add a new profile](#).
- [Choose a profile](#) for a website.
- [Set a profile as default](#).
- [Delete a profile](#).

To update your profile,

1. [Access CrowdID](#).
2. Click '**Profiles**' in the left-hand navigation panel.
3. Select the required profile from the '**Profile**' dropdown list, if you have [more than one profile](#).
4. Update the profile details then click the '**Save**' button.
5. '**Profile updated**' message is displayed at the top of the page.

Screenshot: CrowdID Profiles page

CROWD ID
User: Sarah Maddox | [Logoff](#) | [Change Password](#) | [Help](#)

Home
Administration

My Identity

My OpenID

Profiles

Approved Sites

Login History

Profiles

Select a profile to edit or create a new profile

Profile:

Update profile details

Profile Name: My Profile

Nickname:
A short name to describe yourself. Often used to identify you in places like an online forum.

Full Name:

Email:

Birth Date:
You can partially enter a date of birth, eg. 1980, if you dont want to send the exact details of your birth date.

Gender:

Postcode:

Country:

Timezone:

Language:

Powered by Atlassian CrowdID Version: 1.1.0 (Build:#161 - Jun 19, 2007)
[Report a bug](#) | [Request a feature](#) | [Contact Atlassian](#)

RELATED TOPICS

- [1. Getting started with CrowdID](#)
- [2. Logging in to a website using OpenID](#)
- [3. Viewing your always-approved websites](#)
- [4. Viewing your login history](#)
- [5. Updating your profile](#)
- [6. Using more than one profile](#)
- [7. Changing or resetting your password](#)
- [8. Requesting Forgotten Usernames](#)

6. Using more than one profile

You can create multiple profiles in CrowdID and then allocate specific profiles to specific websites.

- [6.1 Adding a profile](#)
- [6.2 Choosing a profile for a website](#)
- [6.3 Setting a default profile](#)
- [6.4 Deleting a profile](#)

6.1 Adding a profile

When you log in to a website using your OpenID, CrowdID will pass some information to the website. The information is copied from your profile on CrowdID. When your profile is first created, CrowdID will auto-fill the information where possible, by copying:

- Country and language from the language information in your browser.
- Name and email address from your organization's user directory.

To add a profile,

1. [Access CrowdID](#).
2. Click '**Profiles**' in the left-hand navigation panel.
3. Select '-- **Create New Profile** --' from the '**Profile**' dropdown list.
4. CrowdID will auto-fill the information where possible. Update the profile details then click the '**Save**' button.
5. '**Profile updated**' message is displayed at the top of the page.

Screenshot: CrowdID adding a profile

CROWD ID User: Sarah Maddox | Logoff | Change Password | Help

Home Administration

My Identity

- My OpenID
- Profiles**
- Approved Sites
- Login History

Profiles

Select a profile to edit or create a new profile

Profile: -- Create New Profile --

Update profile details

Profile Name: *
The Profile Name is the unique name of the new profile to be created in your CrowdID account.

Nickname:
A short name to describe yourself. Often used to identify you in places like an online forum.

Full Name:

Email:

Birth Date:
You can partially enter a date of birth, eg. 1980, if you dont want to send the exact details of your birth date.

Gender: --

Postcode:

Country:

Timezone: --

Language:

Powered by [Atlassian CrowdID](#) Version: 1.1.0 (Build:#161 - Jun 19, 2007) [Report a bug](#) | [Request a feature](#) | [Contact Atlassian](#)

- [6.1 Adding a profile](#)
- [6.2 Choosing a profile for a website](#)
- [6.3 Setting a default profile](#)
- [6.4 Deleting a profile](#)

[CrowdID User Guide](#)

6.2 Choosing a profile for a website

You can choose a specific profile for use when logging in to a website. There are different ways to choose a profile:

- Choose a profile for a specific login, during the [login process](#). You can do this for sites which you have not set to 'Allow Always'.
- Choose a profile for a specific website, on the CrowdID ['Approved Sites' page](#). You can do this for sites which you have set to 'Allow Always'.
- [Set your default profile](#) on the CrowdID **'Profiles'** page.

RELATED TOPICS

- [6.1 Adding a profile](#)
- [6.2 Choosing a profile for a website](#)
- [6.3 Setting a default profile](#)
- [6.4 Deleting a profile](#)

[CrowdID User Guide](#)

6.3 Setting a default profile

If you have more than one profile, you can choose one of them as default.

Effect of the 'default' profile when you are logging in to a website:

- If you have never logged in to the website before or have previously allowed or denied authentication to that site, the default profile will be pre-selected. You can still choose a different profile during the login.
- If you have set the website to 'Always Allow', CrowdID will use the profile selected for the site on the [Approved Sites](#) page.

To set a default profile,

1. [Access CrowdID](#).
2. Click '**Profiles**' in the left-hand navigation panel.
3. Select the required profile in the '**Profile**' dropdown list
4. Click the '**Make Default**' link next to the 'Profile' dropdown list.
 - The '**Make Default**' link does not appear if the selected profile is already the default.
5. The word '**(default)**' appears next to the profile name in the dropdown list.

Screenshot: CrowdID setting a default profile

The screenshot shows the CrowdID administration interface. At the top, the user is identified as Sarah Maddox with links for Logoff, Change Password, and Help. The navigation menu includes Home and Administration. The left sidebar shows 'My Identity' with sub-items: My OpenID, Profiles, Approved Sites, and Login History. The main content area is titled 'Profiles' and contains a dropdown menu for selecting a profile to edit or create a new profile. The selected profile is 'sm2', and a 'Make Default' link is visible next to it. Below this is the 'Update profile details' section, which includes the following fields:

- Profile Name: sm2
- Nickname: sm2 (with a note: "A short name to describe yourself. Often used to identify you in places like an online forum.")
- Full Name: Sarah Maddox
- Email: sarah@atlassian.com
- Birth Date: 2 February 1980 (with a note: "You can partially enter a date of birth, eg. 1980, if you dont want to send the exact details of your birth date.")
- Gender: Male
- Postcode: 2100
- Country: Australia
- Timezone: Australia/Brisbane
- Language: English

At the bottom of the form are three buttons: Save, Delete, and Cancel.

RELATED TOPICS

- [6.1 Adding a profile](#)
- [6.2 Choosing a profile for a website](#)
- [6.3 Setting a default profile](#)
- [6.4 Deleting a profile](#)

[CrowdID User Guide](#)

6.4 Deleting a profile

You can delete one of your profiles on CrowdID, provided that it is not your [default profile](#).

To delete a profile,

1. [Access CrowdID](#).
2. Click **'Profiles'** in the left-hand navigation panel.
3. Select the required profile in the **'Profile'** dropdown list
4. Click the **'Delete'** button.
5. **'Profile deleted'** message is displayed at the top of the page.

i If you delete a profile which is linked to one or more of your [always-approved websites](#), CrowdID will remove the affected website(s) from the list.

Screenshot: CrowdID profiles page

CROWD ID User: Sarah Maddox [Logoff](#) | [Change Password](#) | [Help](#)

Home Administration

My Identity

- My OpenID
- Profiles**
- Approved Sites
- Login History

Profiles

Select a profile to edit or create a new profile

Profile: [Make Default](#)

Update profile details

Profile Name: sm2

Nickname:
A short name to describe yourself. Often used to identify you in places like an online forum.

Full Name:

Email:

Birth Date:
You can partially enter a date of birth, eg. 1980, if you dont want to send the exact details of your birth date.

Gender:

Postcode:

Country:

Timezone:

Language:

Powered by [Atlassian CrowdID](#) Version: 1.1.0 (Build:#161 - Jun 19, 2007) [Report a bug](#) | [Request a feature](#) | [Contact Atlassian](#)

RELATED TOPICS

- [6.1 Adding a profile](#)

- [6.2 Choosing a profile for a website](#)
- [6.3 Setting a default profile](#)
- [6.4 Deleting a profile](#)

[CrowdID User Guide](#)

7. Changing or resetting your password

If your administrator has allowed it, you can use CrowdID to [change your password](#) across all Crowd applications. Note that you will need to be logged in to Crowd before you can do this.

When attempting to log in to Crowd, you can also [reset your password](#). This is useful when you have forgotten the password. Crowd will send you an email message containing a unique, randomly-generated URL. When you click the link on that URL, you will go to a screen where you can choose your own new password.

RELATED TOPICS

- [1. Getting started with CrowdID](#)
- [2. Logging in to a website using OpenID](#)
- [3. Viewing your always-approved websites](#)
- [4. Viewing your login history](#)
- [5. Updating your profile](#)
- [6. Using more than one profile](#)
- [7. Changing or resetting your password](#)
- [8. Requesting Forgotten Usernames](#)

7.1 Changing your password


The CrowdID 'Change Your Password' page allows you to change your password across all applications in your organization, provided that the application is linked to **Crowd**.

Note:

- Crowd will attempt to change your password in all the user directories linked to Crowd. This will be successful where the directory allows it.
- Your administrator may disable password-change via CrowdID. In that case, you will receive an error message when you apply the change.

To change your password,

1. [Access CrowdID](#).
2. Click '**Change Password**' in the top menu bar.
3. The '**Change Your Password**' page will appear. Type in your old password once, and the new password twice.
4. Click the '**Update**' button.
5. The '**Password updated**' message is displayed.

 If the change is successful, your password may also have changed in other Crowd-connected applications.

Screenshot: CrowdID Change Your Password page




RELATED TOPICS

- [1. Getting started with CrowdID](#)
- [2. Logging in to a website using OpenID](#)
- [3. Viewing your always-approved websites](#)
- [4. Viewing your login history](#)
- [5. Updating your profile](#)
- [6. Using more than one profile](#)
- [7. Changing or resetting your password](#)
- [8. Requesting Forgotten Usernames](#)

7.2 Resetting your password

The CrowdID **Login** page allows you to reset your password. This is useful when you have forgotten the password. Crowd will send you an email message containing a unique, randomly-generated URL. When you click the link on that URL, you will go to a screen where you can choose your own new password.

 This will reset your password across all applications that are connected to Crowd.

To reset your password,

1. [Access CrowdID](#).
2. The CrowdID login page will appear. Click the link labeled '**Can't access your account?**'.
3. The '**Help! I forgot my login details**' screen appears. Select the option labeled '**I have forgotten my password**'.
4. A panel opens where you can enter your username. Enter your Crowd username and click the '**Continue**' button.
5. You will receive an email message containing a link to a unique, randomly-generated URL. This link remains available for 24 hours. Click the link in the email message or copy the URL to your browser address bar.
6. The '**Reset Password**' screen appears. Change your password to one you can remember easily.

The password reset email will contain a link based on `thecrowd.server.url` from the [crowd.properties file](#). If the public address is different you may set `crowd.base.url`.

RELATED TOPICS

- [1. Getting started with CrowdID](#)
- [2. Logging in to a website using OpenID](#)
- [3. Viewing your always-approved websites](#)
- [4. Viewing your login history](#)
- [5. Updating your profile](#)
- [6. Using more than one profile](#)
- [7. Changing or resetting your password](#)
- [8. Requesting Forgotten Usernames](#)

8. Requesting Forgotten Usernames

You can go to the CrowdID **Login** screen and ask CrowdID to email you your username(s). This is useful when you have forgotten your username. CrowdID will send a message to the email address you specify, containing all the usernames that are registered for that email address.

To request your username(s),

1. [Access CrowdID](#).
2. The CrowdID login page appears. Click the link labeled '**Can't access your account?**'.
3. The '**Help! I forgot my login details**' screen appears. Select the option labeled '**I have forgotten my username**'.
4. A panel opens where you can enter your email address. Enter the email address that you used when you registered with CrowdID and click the '**Continue**' button.
5. You will receive an email message containing the usernames registered in CrowdID for that email address.
6. If you have forgotten your password too, you can now ask to [reset your password](#).

RELATED TOPICS

- [1. Getting started with CrowdID](#)
- [2. Logging in to a website using OpenID](#)
- [3. Viewing your always-approved websites](#)
- [4. Viewing your login history](#)
- [5. Updating your profile](#)
- [6. Using more than one profile](#)
- [7. Changing or resetting your password](#)
- [8. Requesting Forgotten Usernames](#)

Crowd FAQ

Known issues, hints and tips and answers to commonly asked questions about Crowd:

Concepts:

- [What is single sign-on \(SSO\)?](#)
- [What is authorization?](#)
- [What is authentication?](#)
- [What is centralized authentication?](#)
- [What is identity management?](#)
- [What is a directory?](#)

Technical:

- [How does Crowd work? How is Crowd an "application security framework"?](#)
- [What is an application connector?](#)
- [What is a directory connector?](#)
- [How many users can Crowd manage?](#)
- [How many applications can be used with Crowd?](#)
- [We already have an LDAP server for Confluence and/or Jira. Do we really need Crowd?](#)

Compatibility:

- [What are Crowd's system requirements?](#)
- [What directories and applications does Crowd support out of the box?](#)
- [How can Crowd be connected to new or currently unsupported applications?](#)
- [How does Crowd integrate with other Atlassian products?](#)
- [Does Crowd include Kerberos integration?](#)

Common Evaluator Questions:

Unable to render {children}. [Page not found: CONFEVAL:Frequently Asked Questions For Crowd.](#)

[Crowd Resources](#)

[Deployment FAQ](#)

- [Deploying Multiple Atlassian Applications in a Single Tomcat Container](#)
- [Finding the atlassian-crowd.log File](#)
- [Finding your Crowd home and shared directories](#)
- [Removing the 'crowd' Context from the Application URL](#)
- [Resetting the Domain Cookie Value](#)
- [Restarting the Setup Wizard from Scratch](#)
- [Self Signed Certificate](#)

[Guides, Hints and Tips](#)

- [How to Print Only Tomcat Logs into Crowd's catalina.out](#)
- [Principals and Users](#)
- [Using Apache Directory Studio for LDAP Configuration](#)
 - [Creating a Connection to your LDAP Directory](#)
 - [Getting an LDIF Export of a User or Group](#)
 - [Restricting LDAP Scope for User and Group Search](#)

[Integration FAQ](#)

- [All Integrations](#)
 - [If I delete a user from Crowd, how will this affect integrated applications?](#)
 - [Passing the crowd.properties File as an Environment Variable](#)
- [Atlassian Product Integration](#)
 - [Application Caching](#)
 - [Jira integration](#)
 - [Public Signup Setup](#)
- [IBM Lotus Domino Integration](#)
- [IBM Websphere Integration](#)

[Support Policies](#)

- [Bug Fixing Policy](#)
- [How to Report a Security Issue](#)
- [New Features Policy](#)
- [Security Advisory Publishing Policy](#)
- [Security Bugfix Policy](#)
- [Security Patch Policy](#)
- [Severity Levels for Security Issues](#)

Troubleshooting

- [Finding Known Issues](#)
- [Characters in User or Group DN's that will cause problems when using Crowd](#)
- [Problems when Importing Users into MySQL](#)
- [Troubleshooting LDAP Error Codes](#)
 - [Active Directory LDAP Errors](#)
- [Troubleshooting LDAP User Management](#)
- [Troubleshooting SSL certificates and Crowd](#)
- [How to Optimize Crowd Client Caching](#)
- [Troubleshooting Crowd Performance](#)
- [Troubleshooting SSO with Crowd](#)
 - [Debugging SSO in environments with Proxy Servers](#)
- [Troubleshooting CrowdID](#)
- [Provide Crowd Information to Atlassian Support](#)

Contributing to the Crowd Documentation

- [Tips of the Trade](#)
- [Crowd Documentation in Other Languages](#)

What is single sign-on?

[Single sign-on](#) enables users to authenticate (login) once and gain access to multiple web applications within a single domain. See also [centralized authentication](#).

What is authorization?

Authorization is the act of deciding whether a person is allowed to access a specific resource or web application. This often comes in the form of groups, roles and permissions.

What is authentication?

Authentication is the act of verifying that a user is who they say they are. This is often done through a credential such as *user name* and *password*.

What is centralized authentication?

Centralized authentication is when an end-user has the same username and password used across all web applications, even if the application cannot participate in [single sign-on](#). This is often a major milestone before single sign-on is achieved within an organization.

Crowd provides centralization authentication and/or single sign-on depending on your application's capabilities.

What is identity management?

Identity management is the process of defining a user (a 'principal') and managing their attributes. In addition to username and credentials (e.g. password), attributes might include phone number, address, etc.

Identity management also includes assigning users to relevant groups and roles, so that users can access appropriate applications and resources.

Another important part of identity management is managing the entire user lifecycle, for example, disabling the user account when someone leaves the organization.

What is a directory?

A directory is a repository of information containing user identities, their attributes and their group and role memberships.

How does Crowd work? How is Crowd an "application security framework"?

Crowd is made up of two parts:

- **Administration console:** a brilliantly simple and powerful web interface that manages directories, users, and their security rights.
- **Integration API:** a single security architecture where multiple web applications are integrated. With the integration API, applications can quickly access user information or perform security checks.

What is an application connector?

An [application connector](#) is the link between Crowd and one of your applications. An application connector makes it possible to connect, say, Crowd and Jira. When you download and install Crowd, you'll automatically get its application connectors, along with an integration API so that you can code your own application connectors too.

What is a directory connector?

A [directory connector](#) is the link between Crowd and one of your directories. It makes it possible to connect, say, Crowd and Active Directory. When you download and install Crowd, you'll automatically get its directory connectors, along with an integration API so that you can code your own directory connectors too.

How many users can Crowd manage?

Crowd can support over 500 users depending upon which license you purchase. View the [licensing and pricing](#) breakdown for more information. We have customers using Crowd successfully with tens of thousands of users.

How many applications can be used with Crowd?

So long as they're compatible with Crowd, you can add in as many [applications](#) as your organization needs.

We already have an LDAP server for Confluence and/or Jira. Do we really need Crowd?

If one or more of the following apply, Crowd will be of benefit to you:

- your organization uses multiple [applications](#) and they have not yet been integrated into the LDAP server
- you are looking for an easy way to manage all your Jira and Confluence users in one database with one or more directory servers
- your organization has not yet implemented [single sign-on](#)
- you are looking for a way to help save you and your organization time, frustration, and much more!

What are Crowd's system requirements?

For information on compatible databases, application servers, and operating systems, read the [Supported Platforms](#) page.

What directories and applications does Crowd support out-of-the-box?

A complete list of currently supported applications and directories can be found in Crowd's [documentation](#). Check back often, as new connectors will be added regularly.

How can Crowd be connected to new or currently unsupported applications?

Crowd provides a simple and intuitive [integration](#) API (backed by [REST or SOAP](#)) that allows you to connect in your new or existing applications. This makes it easy to choose how much or how little to integrate based on your needs.

How does Crowd integrate with other Atlassian products?

Crowd ships with [connectors](#) for Atlassian products.

Using the out-of-the-box connectors you can [consolidate](#) all of your users into a single repository giving you the ability to [manage](#) all of your users in a single location. Users can then take advantage of [single sign-on](#), giving them one username and password to access all of your applications.

Does Crowd include kerberos integration?

Crowd does not currently support kerberos-based authentication.

For licensing and pricing please see the [Purchasing FAQ](#).

Crowd Resources

Resources for Evaluators

- [Free Trial](#)
- [Feature Tour](#)

Resources for Administrators

- [Crowd Knowledge Base](#)
- [Tips of the Trade](#)
- [Guide to Installing an Atlassian Integrated Suite](#)

Downloadable Documentation

- [Crowd documentation in PDF, HTML or XML formats](#)

Plugins and Extensions

- [Atlassian Plugin Exchange](#)

Support

- [Atlassian Support](#)
- [Support Policies](#)

Forums and Announcements

- [Crowd Announcements](#)
- [Answers from the community](#)

Mailing Lists

- Visit <http://my.atlassian.com> to sign up for mailing lists relating to Atlassian products, such as technical alerts, product announcements and developer updates.

Feature Requests

- [Issue Tracker and Feature Requests for Crowd](#)

Deployment FAQ

- [Deploying Multiple Atlassian Applications in a Single Tomcat Container](#)
- [Finding the atlassian-crowd.log File](#)
- [Finding your Crowd home and shared directories](#)
- [Removing the 'crowd' Context from the Application URL](#)
- [Resetting the Domain Cookie Value](#)
- [Restarting the Setup Wizard from Scratch](#)
- [Self Signed Certificate](#)

Deploying Multiple Atlassian Applications in a Single Tomcat Container

Deploying multiple Atlassian applications in a single Tomcat container is **not supported**. We do not test this configuration and upgrading any of the applications (even for point releases) is likely to break it. There are also a number of known issues with this configuration:

- You may not be able to start up all of the applications in the container, due to class conflicts (in 3rd party libraries bundled with our application) that result from the Atlassian applications sharing a single JVM in the Tomcat container.
- You will not be able to determine the startup order of the applications. Hence, you may experience problems such as JIRA starting before Crowd, rather than vice versa.
- Memory problems are also common as one application may allocate all of the memory in the Tomcat JVM to itself, starving the other applications.

We also do not support deploying multiple Atlassian applications to a single Tomcat container for a number of practical reasons. Firstly, you must shut down Tomcat to upgrade any application and secondly, if one application crashes, the other applications running in that Tomcat container will be inaccessible.

Finally, we recommend not deploying *any other applications* to the same Tomcat container that runs the Atlassian application, especially if these other applications have large memory requirements or require additional libraries in Tomcat's `lib` subdirectory.

Finding the atlassian-crowd.log File

When you report a problem to Atlassian Support, we may ask you to send us your `atlassian-crowd.log` file. The location of the log file may vary, depending on your Crowd installation type. Provided that you have not changed the log file location from the default, the Crowd log file is at the location described below.

Installation Type	Location of Log File
Crowd Standalone edition	Crowd 2.0.3 and older versions: In the root directory of your Crowd application, e.g. <code>atlassian-crowd-2.0.0/atlassian-crowd.log</code> Crowd 2.0.4 and newer versions: In the Crowd application Home Directory, e.g. <code>Crowd-Home-Directory/logs/atlassian-crowd.log</code>
Crowd Standalone running as a Windows service	<code>C:\Windows\system32\atlassian-crowd.log</code>

How do I Change the Location?

You can change the location of the log file by modifying the following line in the `WEB-INF/classes/log4j.properties` file of your Crowd installation to use an absolute file path:

```
log4j.appender.filelog.File=atlassian-crowd.log
```

For more information, please refer to the page on [logging and profiling](#).

RELATED TOPICS

[Logging and Profiling](#)
[Important directories and files](#)

Finding your Crowd home and shared directories

You can check the location of your Crowd home directory on the [System Information](#) screen.

Home directory this is where Crowd stores its configuration information. If you're using the embedded HSQLDB database supplied for evaluation purposes, it will also be there (note however that the CrowdID database will be in the installation directory, not the home directory).

Shared directory this is a sub-directory of the Crowd home directory. It contains common data for your Crowd installation, such as database configuration and plugins. If you're using Crowd Data Center, content of this directory will be shared by each node in the cluster.

Read more:

- [Setting your home directory during installation.](#)
- [The location and function of the Crowd home directory and other important files and directories.](#)

Removing the 'crowd' Context from the Application URL

For many different reasons, when using the Crowd distribution, you may want to access the Crowd console using `http://localhost:8095` instead of `http://localhost:8095/crowd`.

To remove the `/crowd` part from the URL:

1. In `<Crowd-Install>/build.properties` set the `crowd.url` variable to the following:

```
# Crowd context root
crowd.url=http://localhost:8095/
```

2. Run `<Crowd-Install>/build.sh` (**UNIX**) or `<Crowd-Install>\build.bat` (**Windows**).
3. Change your `<Crowd-Install>/apache-tomcat/conf/server.xml` file to include the following element in the **Host** section configuration:

```
<Context path="" docBase="../../crowd-webapp" debug="0">
  <Manager pathname="" />
</Context>
```

Sample **Host** configuration:

```
<Engine defaultHost="localhost" name="Catalina">
  <Host appBase="webapps" autoDeploy="true" name="localhost" unpackWARs="true">
    <Context path="" docBase="../../crowd-webapp" debug="0">
      <Manager pathname="" />
    </Context>
    <Valve className="org.apache.catalina.valves.ErrorReportValve"
      showReport="true"
      showServerInfo="false" />
  </Host>
</Engine>
```

4. Perform a backup of the `crowd.xml` file in `<Crowd-Install>/apache-tomcat/conf/Catalina/localhost` to another directory.
5. From `<Crowd-Install>/apache-tomcat/conf/Catalina/localhost`, remove the `crowd.xml` file to prevent Tomcat from loading the `/crowd` context.
6. After the restart, in the [Server Settings](#) screen, change the base URL.

Resetting the Domain Cookie Value

If you have set the [SSO Domain](#) to an invalid value, you may be prevented from authenticating to the Crowd Console.

To reset the SSO (single sign-on) cookie domain, run the following SQL command on the Crowd database:

```
update cwd_property set property_value = '' where property_name = 'domain';
```

Once you have done this you will need to restart Crowd, clear your browser's cache and then log in. This will reset any domain SSO token misconfiguration.

Restarting the Setup Wizard from Scratch

If you get part-way through the [Crowd Setup Wizard](#) and then decide you want to start again from scratch, you can delete the **Crowd Home** directory. (See [Important directories and files](#).)

Crowd uses the `crowd.cfg.xml` file, stored in the Crowd Home directory, to 'remember' the step you have reached in the setup procedure. Clearing the file will cause the Setup Wizard to start at the beginning again.

This strategy is useful if you want to re-do your setup without having to download Crowd again.

To restart the Crowd Setup Wizard:

1. Shut down Crowd.
2. Delete your **Crowd Home** directory.
3. Start Crowd again.
4. Go go `http://localhost:8095/crowd`.
5. The Crowd Setup Wizard will start. Follow the steps from the beginning, as described in [Running the Setup Wizard](#).



Embedded database will disappear too

If you are using the [embedded database](#), the database files are stored in the Crowd Home directory too. Deleting the Crowd Home directory will remove all your Crowd Administration Console data as well (users, groups, roles, directories, applications and other configuration data).

Self Signed Certificate

I have a self Signed Certificate

You will need to add the self-signed certificate to your JDK truststore using the JDK keytool: <https://docs.oracle.com/javase/8/docs/technotes/tools/windows/keytool.html>

Guides, Hints and Tips

- [How to Print Only Tomcat Logs into Crowd's catalina.out](#)
- [Principals and Users](#)
- [Using Apache Directory Studio for LDAP Configuration](#)

How to Print Only Tomcat Logs into Crowd's catalina.out

Crowd writes its logs into [atlassian-crowd.log](#) and Tomcat logs as well. However, this might only be noticed for Crowd installed in *nix based server, as in Windows these are printed in the console. Below is the start-up process written in `catalina.out`, where it describe the start-up process of all applications bundled inside Crowd standalone, such as Crowd OpenID and Crowd itself:

Crowd OpenID Start-up

```
May 31, 2012 6:20:03 PM org.apache.coyote.http11.Http11Protocol init
INFO: Initializing Coyote HTTP/1.1 on http-9424
May 31, 2012 6:20:03 PM org.apache.catalina.startup.Catalina load
INFO: Initialization processed in 265 ms
May 31, 2012 6:20:03 PM org.apache.catalina.realm.JAASRealm setContainer
INFO: Set JAAS app name Catalina
May 31, 2012 6:20:03 PM org.apache.catalina.core.StandardService start
INFO: Starting service Catalina
May 31, 2012 6:20:03 PM org.apache.catalina.core.StandardEngine start
INFO: Starting Servlet Engine: Apache Tomcat/6.0.32
May 31, 2012 6:20:03 PM org.apache.catalina.startup.HostConfig deployDescriptor
INFO: Deploying configuration descriptor crowd.xml
2012-05-31 18:20:12,341 main INFO [com.atlassian.crowd.startup] System Information:
2012-05-31 18:20:12,342 main INFO [com.atlassian.crowd.startup] Timezone: Malaysia Time
2012-05-31 18:20:12,342 main INFO [com.atlassian.crowd.startup] Java Version: 1.6.0_31
2012-05-31 18:20:12,342 main INFO [com.atlassian.crowd.startup] Java Vendor: Sun Microsystems Inc.
2012-05-31 18:20:12,342 main INFO [com.atlassian.crowd.startup] JVM Version: 20.6-b01
2012-05-31 18:20:12,342 main INFO [com.atlassian.crowd.startup] JVM Vendor: Sun Microsystems Inc.
2012-05-31 18:20:12,342 main INFO [com.atlassian.crowd.startup] JVM Runtime: Java HotSpot(TM) 64-
Bit Server VM
2012-05-31 18:20:12,342 main INFO [com.atlassian.crowd.startup] Username: sultan
2012-05-31 18:20:12,342 main INFO [com.atlassian.crowd.startup] Operating System: Linux3.2.0-23-
generic
2012-05-31 18:20:12,342 main INFO [com.atlassian.crowd.startup] Architecture: amd64
2012-05-31 18:20:12,342 main INFO [com.atlassian.crowd.startup] File Encoding: UTF-8
2012-05-31 18:20:12,342 main INFO [com.atlassian.crowd.startup] JVM Statistics:
2012-05-31 18:20:12,342 main INFO [com.atlassian.crowd.startup] Total Memory: 124MB
2012-05-31 18:20:12,342 main INFO [com.atlassian.crowd.startup] Used Memory: 34MB
2012-05-31 18:20:12,342 main INFO [com.atlassian.crowd.startup] Free Memory: 89MB
2012-05-31 18:20:12,342 main INFO [com.atlassian.crowd.startup] Runtime Information:
2012-05-31 18:20:12,342 main INFO [com.atlassian.crowd.startup] Version: 2.4.2
2012-05-31 18:20:12,342 main INFO [com.atlassian.crowd.startup] Build Number: 563
2012-05-31 18:20:12,342 main INFO [com.atlassian.crowd.startup] Build Date: 07-05-2012
2012-05-31 18:20:12,342 main INFO [crowd.console.listener.StartupListener] Upgrades not performed since the
application has not been setup yet.
2012-05-31 18:20:12,368 main INFO [ContainerBase.[Catalina].[localhost].[./crowd]] org.tuckey.web.filters.
urlrewrite.UrlRewriteFilter INFO: loaded (conf ok)
May 31, 2012 6:20:12 PM org.apache.catalina.startup.HostConfig deployDescriptor
INFO: Deploying configuration descriptor openidserver.xml
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/sultan/atlassian/CROWD/atlassian-crowd-2.4.2/crowd-openidserver-
webapp/WEB-INF/lib/slf4j-jcl-1.0.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/sultan/atlassian/CROWD/atlassian-crowd-2.4.2/crowd-openidserver-
webapp/WEB-INF/lib/slf4j-log4j12-1.5.8.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
2012-05-31 18:20:14,313 main INFO [com.atlassian.crowd.startup] Starting Crowd OpenID Server, Version:
2.4.2 (Build:#563 - 07-05-2012)
```

Crowd Start-up

```

2012-05-31 21:17:17,314 main INFO [com.atlassian.crowd.startup] System Information:
2012-05-31 21:17:17,314 main INFO [com.atlassian.crowd.startup]     Timezone: Malaysia Time
2012-05-31 21:17:17,314 main INFO [com.atlassian.crowd.startup]     Java Version: 1.6.0_31
2012-05-31 21:17:17,314 main INFO [com.atlassian.crowd.startup]     Java Vendor: Sun Microsystems Inc.
2012-05-31 21:17:17,314 main INFO [com.atlassian.crowd.startup]     JVM Version: 20.6-b01
2012-05-31 21:17:17,314 main INFO [com.atlassian.crowd.startup]     JVM Vendor: Sun Microsystems Inc.
2012-05-31 21:17:17,315 main INFO [com.atlassian.crowd.startup]     JVM Runtime: Java HotSpot(TM) 64-
Bit Server VM
2012-05-31 21:17:17,315 main INFO [com.atlassian.crowd.startup]     Username: sultan
2012-05-31 21:17:17,315 main INFO [com.atlassian.crowd.startup]     Operating System: Linux3.2.0-23-
generic
2012-05-31 21:17:17,315 main INFO [com.atlassian.crowd.startup]     Architecture: amd64
2012-05-31 21:17:17,315 main INFO [com.atlassian.crowd.startup]     File Encoding: UTF-8
2012-05-31 21:17:17,315 main INFO [com.atlassian.crowd.startup] JVM Statistics:
2012-05-31 21:17:17,315 main INFO [com.atlassian.crowd.startup]     Total Memory: 124MB
2012-05-31 21:17:17,315 main INFO [com.atlassian.crowd.startup]     Used Memory: 24MB
2012-05-31 21:17:17,315 main INFO [com.atlassian.crowd.startup]     Free Memory: 99MB
2012-05-31 21:17:17,315 main INFO [com.atlassian.crowd.startup] Runtime Information:
2012-05-31 21:17:17,315 main INFO [com.atlassian.crowd.startup]     Version: 2.4.2
2012-05-31 21:17:17,315 main INFO [com.atlassian.crowd.startup]     Build Number: 563
2012-05-31 21:17:17,315 main INFO [com.atlassian.crowd.startup]     Build Date: 07-05-2012
2012-05-31 21:17:17,315 main INFO [com.atlassian.crowd.startup]     Application Server: Apache Tomcat/6.
0.32
2012-05-31 21:17:17,315 main INFO [com.atlassian.crowd.startup] Database Information:
2012-05-31 21:17:17,315 main INFO [com.atlassian.crowd.startup]     JDBC URL: jdbc:mysql://localhost
/crowddb242?autoReconnect=true&characterEncoding=utf8&useUnicode=true
2012-05-31 21:17:17,315 main INFO [com.atlassian.crowd.startup]     JDBC Driver: com.mysql.jdbc.Driver
2012-05-31 21:17:17,315 main INFO [com.atlassian.crowd.startup]     JDBC Username: root
2012-05-31 21:17:17,315 main INFO [com.atlassian.crowd.startup]     Hibernate Dialect: org.hibernate.
dialect.MySQL5InnoDBDialect
2012-05-31 21:17:17,316 main INFO [com.atlassian.crowd.startup] License Information:
2012-05-31 21:17:17,316 main INFO [com.atlassian.crowd.startup]     License Server ID: BTFR-LW50-LYA5-
W1IT
2012-05-31 21:17:17,316 main INFO [com.atlassian.crowd.startup] Directories:
2012-05-31 21:17:17,359 main INFO [com.atlassian.crowd.startup]     CROWD242 (InternalDirectory)
    JIRA504 (InternalDirectory)

```

This might be a redundant information as they've been written into `atlassian-crowd.log`.

In order to get Crowd print only Tomcat logs into `catalina.out`, please modify these files:

- `<Crowd_Install_Directory>/crowd-webapp/WEB-INF/classes/log4j.properties`
- `<Crowd_Install_Directory>/crowd-openidserver-webapp/WEB-INF/classes/log4j.properties`
- `<Crowd_Install_Directory>/crowd-openidclient-webapp/WEB-INF/classes/log4j.properties`
- `<Crowd_Install_Directory>/demo-webapp/WEB-INF/classes/log4j.properties`

modify this line:

DEFAULT

```
log4j.rootLogger=INFO, console, crowdlog
```

into:

Modify it to

```
log4j.rootLogger=INFO, crowdlog
```

Save all of the modified `log4j.properties` and restart Crowd.

Principals and Users

As far as Crowd is concerned, the terms '**principals**' and '**users**' are equivalent: they mean the same thing. Earlier versions of Crowd used the term 'principals'. From Crowd 1.3 onwards, we call them 'users'.

Using Apache Directory Studio for LDAP Configuration

This is a basic tutorial on using a wonderful Eclipse-based LDAP browser, known as [Apache Directory Studio](#), to gather the information you need for your [LDAP configuration](#).

Before you Start

Step 1. Get Apache Directory Studio

- Download and install [Apache Directory Studio](#).

Step 2. (*Optional*) Do Some Background Reading

If you are an LDAP newbie, there are two great articles that may help you gain a better understanding of LDAP and LDAP search filters before you begin using Apache Directory Studio:

- [An Introduction to LDAP](#)
- [How to write an LDAP search filter](#)

Table of Contents

[Creating a Connection to your LDAP Directory](#)

[Getting an LDIF Export of a User or Group](#)

[Restricting LDAP Scope for User and Group Search](#)

RELATED TOPICS

[Configuring an LDAP Directory Connector](#)

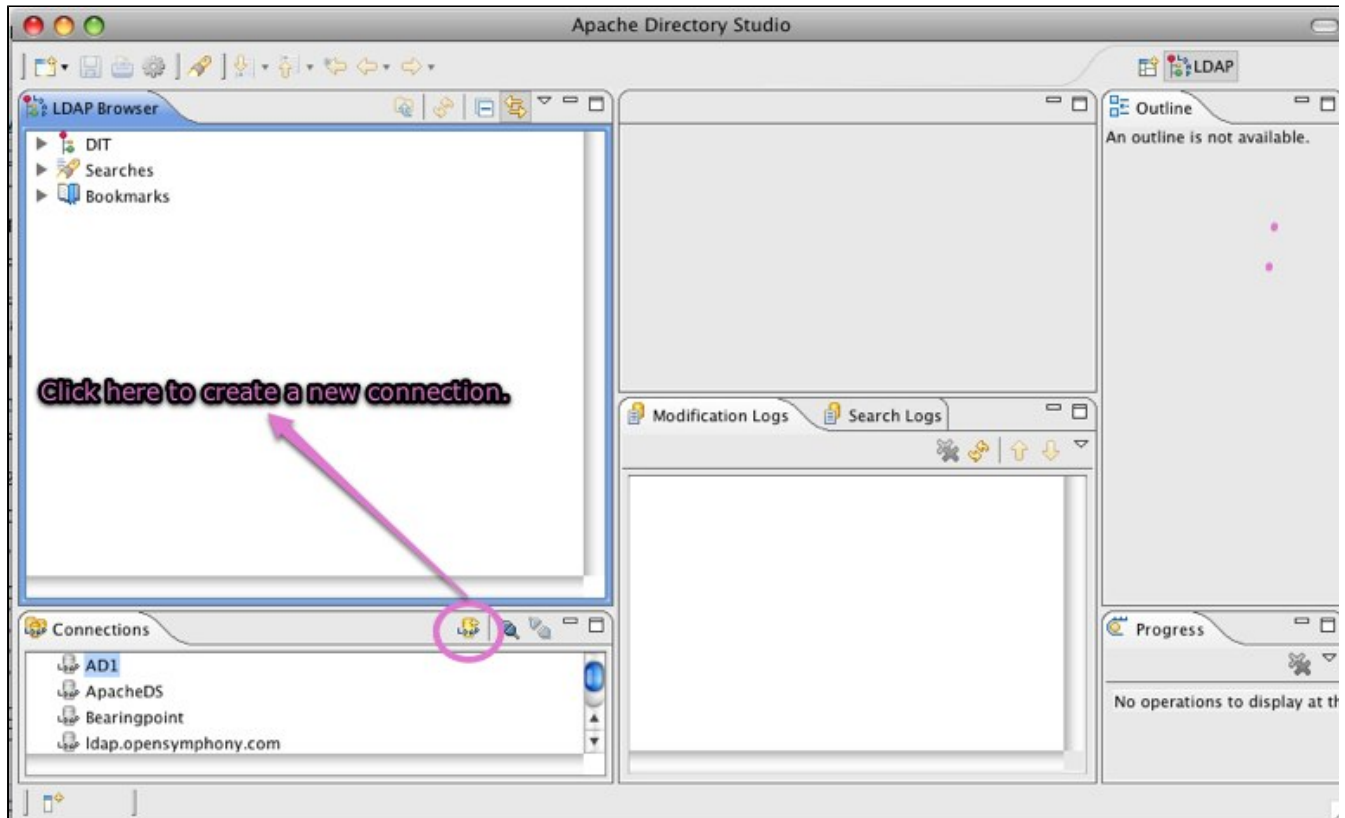
Creating a Connection to your LDAP Directory

You may find an LDAP browser useful to gather the information you need for your Crowd configuration. This page shows you how to create a connection to your LDAP directory when using [Apache Directory Studio](#). You can then use the connection information gathered, to [set up your LDAP directory in Crowd](#).

Step 1: Create a New Connection in Apache Directory Studio

1. Start up [Apache Directory Studio](#).
2. Click the LDAP icon to create a new connection.

Screenshot: Creating a new connection in Apache Directory Studio



Step 2: Enter your Connection Information

1. Enter a name for your connection.
2. Enter the '**Network Parameter**' information as follows:

Host name	The domain name for your LDAP server. If the LDAP server is not on the same network as Crowd, you may need to use the FQDN or IP address of the LDAP server.
Port	For normal LDAP connectivity, use 389. For SSL connectivity, use 636.

3. Click the '**Check Network Parameter**' button to ensure your connection is successful.
4. Click '**Next**'.

Screenshot: Entering the connection information in Apache Directory Studio

New LDAP Connection

Network Parameter
Please enter connection name and network parameters.

Connection name: Crowd Active Directory

Network Parameter

Hostname: crowd-ad1

Port: 389

Encryption method: No encryption

Warning: The current version doesn't support certificate validation, be aware of invalid certificates or man-in-the-middle attacks!

Check Network Parameter

< Back Next > Cancel Finish

Step 3: Enter your Authentication Information

1. Choose the '**Authentication Method**' from the dropdown list.
 - Some LDAP servers allow anonymous access. If your LDAP server allows this, you can change the 'Authentication Method' dropdown from 'Simple Authentication' to 'Anonymous Authentication' and click '**Finish**' to go straight to [Step 4](#).

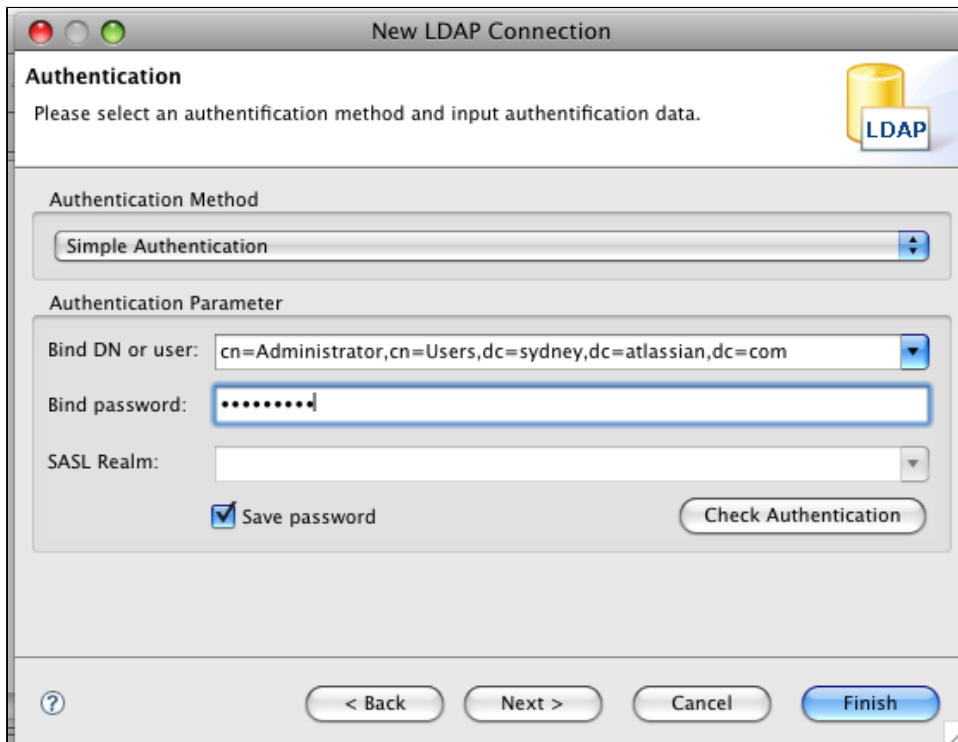


2. Enter the '**Authentication Parameter**' information as follows:

Bind DN or user	Enter the full DN of the account that will be used to connect to the LDAP directory. This account should have the ability to browse the entire LDAP directory tree.
Bind password	Enter the password for the Bind DN account.

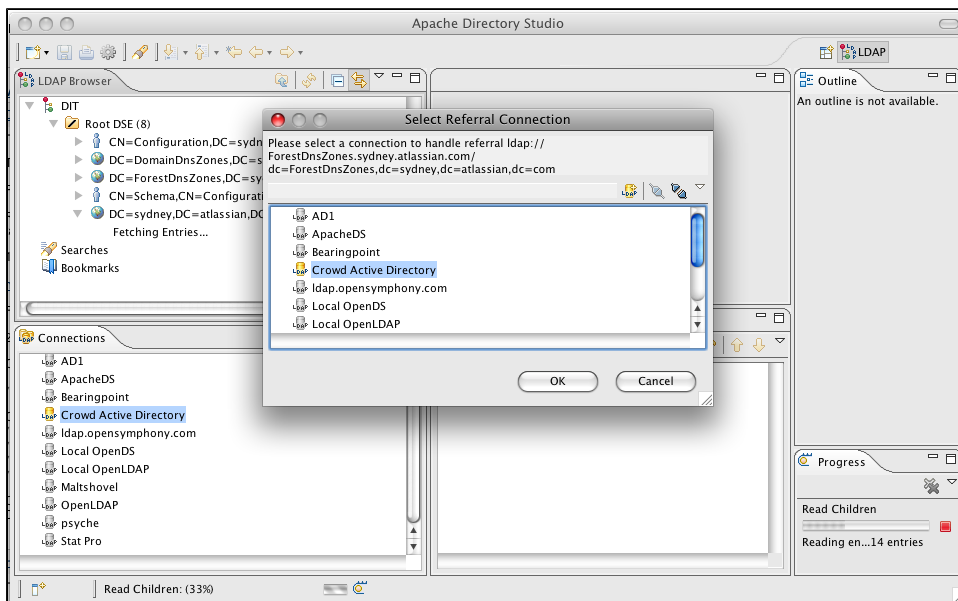
3. Click the '**Check Authentication**' button to ensure this account can authenticate.
4. If this authentication is successful, click '**Finish**'.

Screenshot: Entering the authentication information in Apache Directory Studio



5. If you are prompted for a 'Referral Connection', select the same directory.

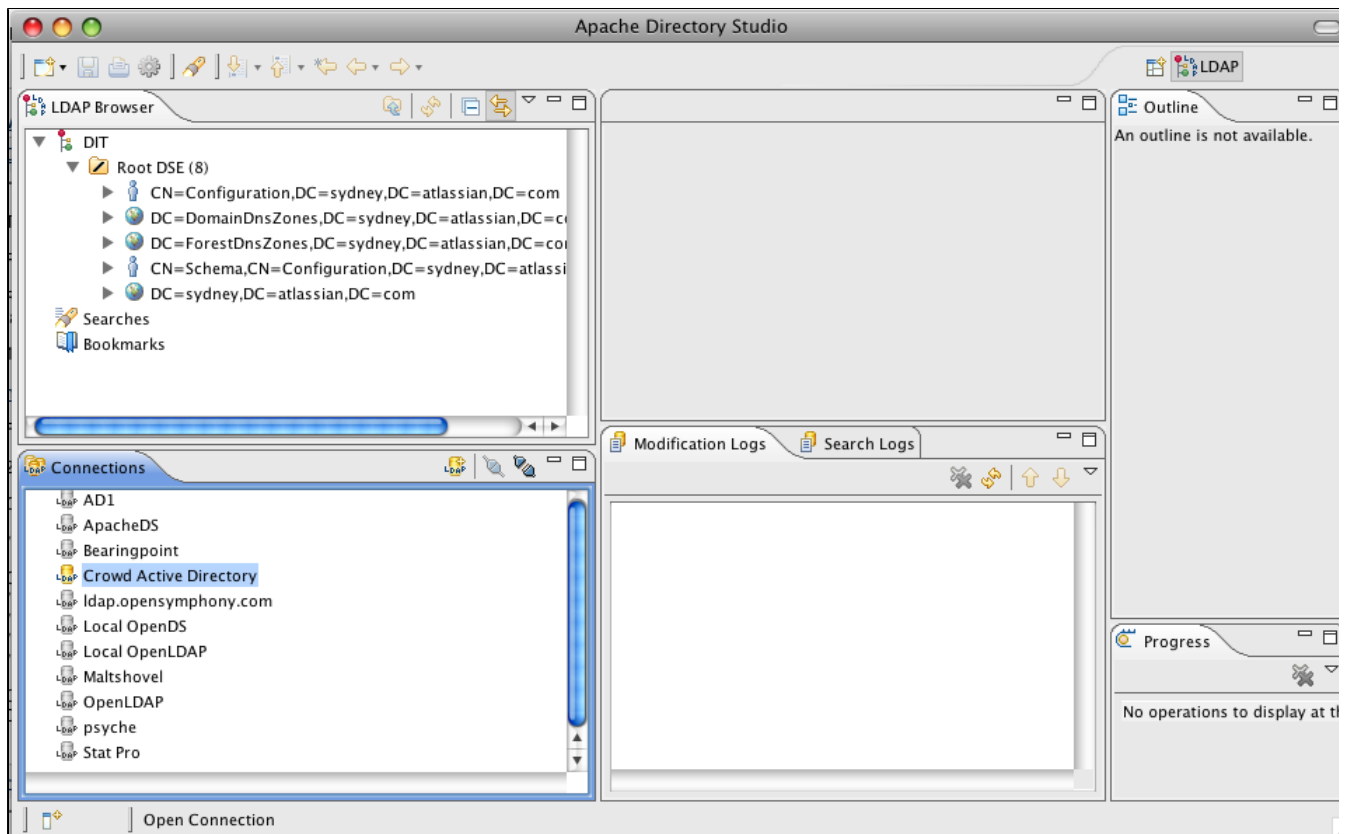
Screenshot: Selecting a referral connection in Apache Directory Studio



Step 4: See the Base DNs

If the configuration is successful, you should now have a list of the base DNs available under this LDAP directory's root DSE.

Screenshot: Viewing the base DNs in Apache Directory Studio



Step 5: Use the Same Connection Information in Crowd

Use the same connection information to [set up your LDAP directory in Crowd](#).

Screenshot: LDAP directory configuration in Crowd

The screenshot shows the 'View Directory - AD1' configuration page in the Crowd interface. The 'Connector' tab is selected, showing the following configuration details:

- Type:** Microsoft Active Directory
- URL:** ldap://crowd-ad1:389/ (Annotated with 'Hostname and Port from Step 1')
- Secure SSL:**
- Use Node Referrals:**
- Use Nested Groups:**
- Use Paged Results:**
- Paged Results Size:** 999
- Base DN:** dc=sydney,dc=atlassian,dc=com (Annotated with 'Base DN displayed from Step 4')
- User DN:** cn=Administrator,cn=Users,dc=sydney,dc=atlassian,dc=com (Annotated with 'Bind DN from Step 3')
- Password:** (Annotated with 'Bind Password from Step 3')

Buttons for 'Update' and 'Cancel' are visible at the bottom of the form.

RELATED TOPICS

- [Using Apache Directory Studio for LDAP Configuration](#)
- [Configuring an LDAP Directory Connector](#)

Restricting LDAP Scope for User and Group Search

While you should already know the user DN (Distinguished Name) you are using for your LDAP connection, it can be helpful to review the users and groups in [Apache Directory Studio](#) to determine the best scope for your Crowd LDAP directory configuration.

Crowd comes with default configurations that will work for most customers. In the examples below, we illustrate some common options for changing your user and group configurations.

There are a number of other attributes, not shown here, that can also be used to narrow the scope of users and groups.

Important Search Filter Notes

- If you are unfamiliar with LDAP search filter syntax, please review [this guide](#).
- See [Creating a Connection to your LDAP Directory](#) for details of how to connect Apache Directory Studio to your LDAP directory.
- In order to use Object Filters larger than 255 characters, you will need to upgrade to [Crowd to 1.5.1 or later](#), by installing a new Crowd instance (with a new database) and restoring an XML backup from your previous Crowd installation. For more information on upgrading Crowd please review the [Upgrade Guide](#)
- If you are using [Nested Groups in Crowd](#), your group filter must include all sub-groups to pick up the sub-group members

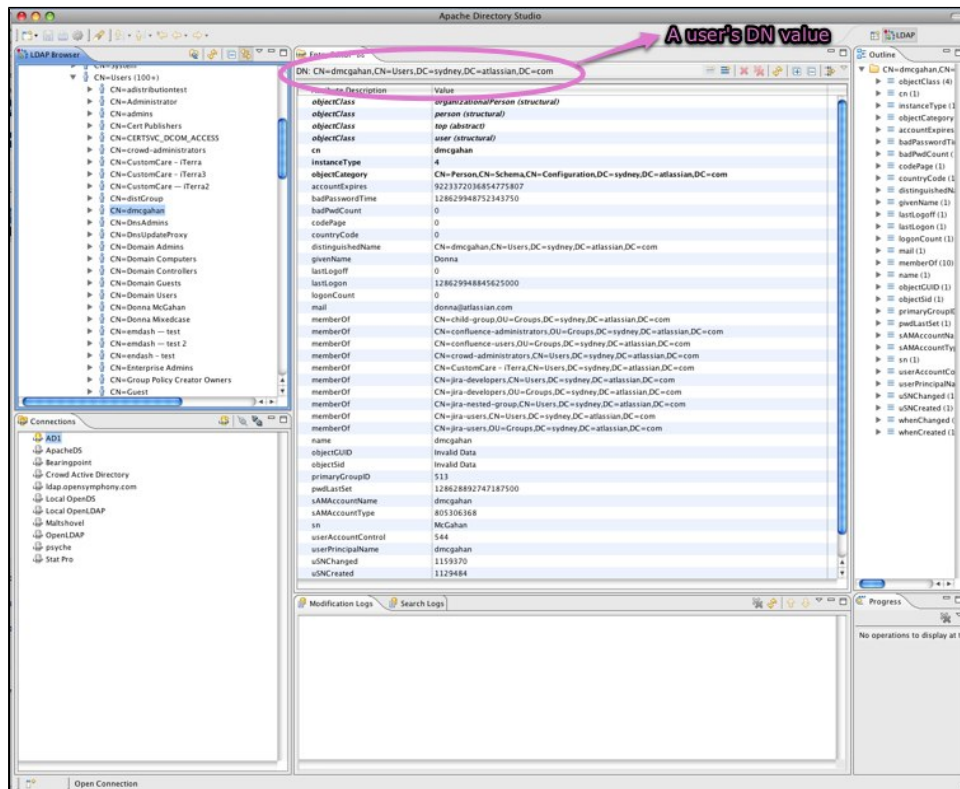
On this page:

- [Example 1. Using a User's DN for Crowd Configuration](#)
- [Example 2: Using a Group's DN for Crowd Configuration](#)

Example 1. Using a User's DN for Crowd Configuration

1. Find a user in the scope you wish to use for Crowd. Highlight that user in [Apache Directory Studio](#).

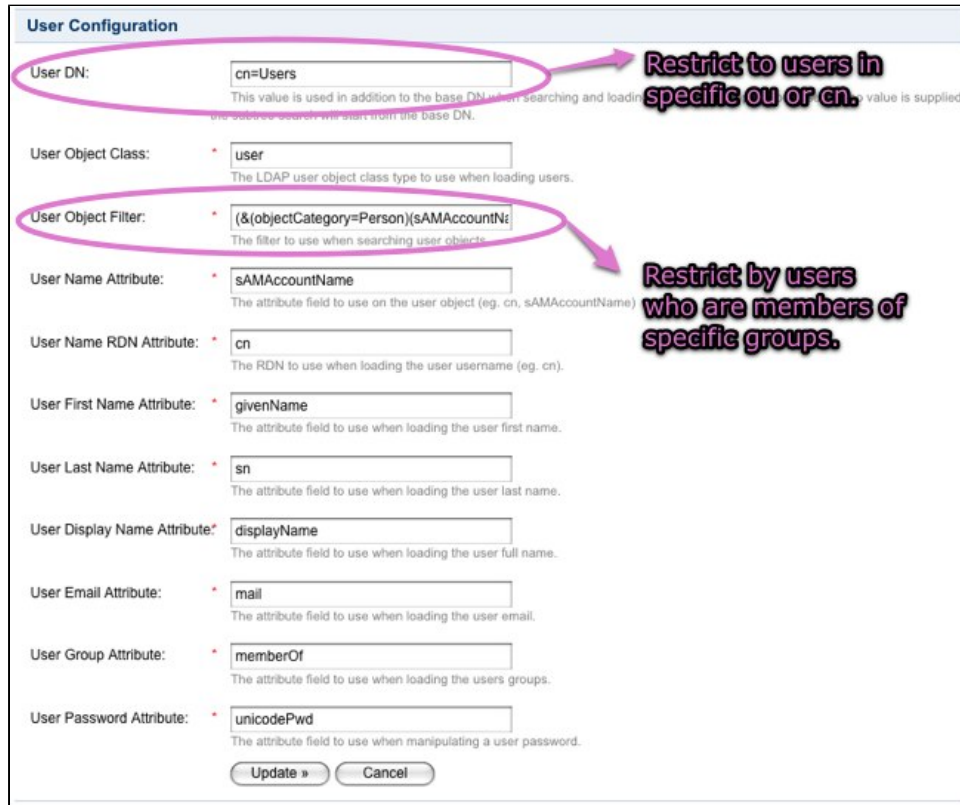
Screenshot: User information in Apache Directory Studio



- Using the information about the user `dmcgahan`, you can narrow down the users returned in the Crowd directory to those in `cn=Users` who are members of either the `confluence-users` or the `confluence-administrators` group.

User DN:	<code>cn=Users</code>
User Object Filter:	<pre>(& (objectCategory=Person) (sAMAccountName=*) ((memberOf=cn=confluence-users, ou=Groups, dc=sydney, dc=atlassian, dc=com) (memberOf=cn=confluence-administrators, ou=Groups, dc=sydney, dc=atlassian, dc=com)))</pre>

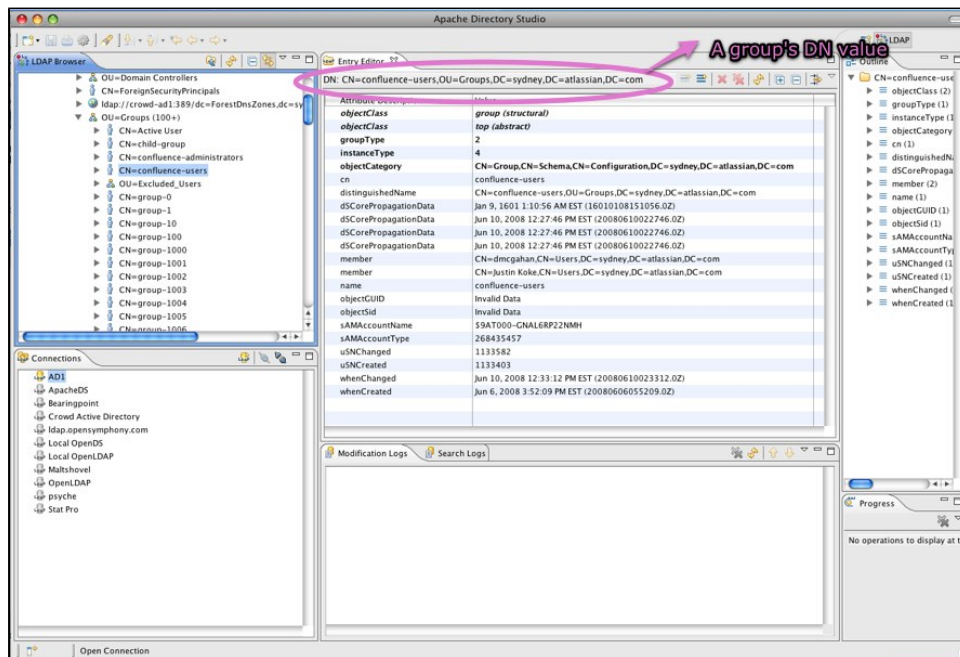
Screenshot: The resulting user configuration in Crowd



Example 2: Using a Group's DN for Crowd Configuration

1. Find a group in the scope you wish to use for Crowd. Highlight that group in [Apache Directory Studio](#).

Screenshot: Group information in Apache Directory Studio



2. Using the information about the group *confluence-users*, you can narrow down the groups returned in the Crowd directory to those in *ou=Groups* and return only the *confluence-users* or the *confluence-administrators* group. Under most circumstances, it is best to apply any changes to both group and role configuration for consistency.

Group DN:	ou=Groups
Group Object Filter:	<pre>(&(objectCategory=Group)((cn=confluence-users)(cn=confluence-administrators)))</pre>

Screenshot: The resulting group/role configuration in Crowd

Group Configuration

Group DN: **Restrict by ou or cn**
The value is used in addition to the base DN when searching and loading groups. An example is ou=Groups. If no value is supplied, the subtree search will start from the base DN.

Group Object Class: *
The LDAP user object class type to use when loading groups.

Group Object Filter: **Restrict by groups.**
The filter to use when searching group objects.

Group Name Attribute: *
The attribute field to use when loading the group name.

Group Description Attribute: *
The attribute field to use when loading the group description.

Group Members Attribute: *
The attribute field to use when loading the group members.

RELATED TOPICS

[Using Apache Directory Studio for LDAP Configuration](#)

Integration FAQ

- All Integrations
 - If I delete a user from Crowd, how will this affect integrated applications?
 - Passing the crowd.properties File as an Environment Variable
- Atlassian Product Integration
 - Application Caching
 - Jira integration
 - Public Signup Setup
- IBM Lotus Domino Integration
- IBM Websphere Integration

All Integrations

- [If I delete a user from Crowd, how will this affect integrated applications?](#)
- [Passing the crowd.properties File as an Environment Variable](#)

If I delete a user from Crowd, how will this affect integrated applications?

We recommend that you **deactivate** a user rather than deleting them, in case some applications contain historical data, e.g. documents that the user has created.

For example, a user may be a participant in a [JIRA](#) issue. If you remove the user from the directory managed by Crowd, JIRA will not be able to find the user details when referencing the issue. If you do need to remove the user from Crowd, you must first remove the user's involvement in any JIRA issues, as described in the [JIRA documentation](#).

Read more about [deleting or deactivating users](#) in Crowd.

Passing the crowd.properties File as an Environment Variable

When [integrating a client application](#) with Crowd, you need a `crowd.properties` file containing configuration details for that application. (See [Important directories and files](#).)

You can pass the location of a client application's `crowd.properties` file to the client application as an environment variable when starting the client application. This means that you can choose a suitable location for the `crowd.properties` file, instead of putting it in the client application's `WEB-INF/classes` directory.

Example:

```
-Dcrowd.properties={FILE-PATH}/crowd.properties
```

Atlassian Product Integration

This section covers general questions around Crowd's integration with other Atlassian products.

General Integration Questions

[Why don't my Groups and Users show up in Bamboo, Confluence, Fisheye or JIRA?](#)

[I want to allow public signups, but don't want 'public' users in my company LDAP repository. How should I configure Crowd?](#)

Confluence Integration

JIRA Integration

[What is the difference between JIRA's direct LDAP integration & Crowd's JIRA integration?](#)
[If I delete a user from Crowd, how will this affect JIRA?](#)

Bamboo Integration

Fisheye Integration

Application Caching

When Crowd is deployed into Bamboo, Confluence, Fisheye or JIRA, the Crowd client may be using caching. If you notice that changes made in Crowd do not appear in one of Crowd's configured applications, this will most likely mean that the changes have not yet propagated into the client caches.

 The Crowd development team has opened [an improvement request \(CWD-1283\)](#) for this issue. Please vote on this issue and add it to your [JIRA watch list](#) for future updates.

For more information, refer to:

- [An overview of the different caching options in Crowd.](#)
- [Configuring caching for an application.](#)
- [Caching of user permissions on the Crowd server.](#)
- [Caching for LDAP directories.](#)

Jira integration

What is the difference between Jira's LDAP integration and Crowd's JIRA integration?

[Jira's LDAP integration](#) only delegates authentication to LDAP. This means that you still need to create groups and users in JIRA, and those users must have usernames that match your users in LDAP.

When you use [Crowd's Jira integration](#), all user and group management is delegated to Crowd. This means that you no longer have to create users and groups in Jira. Crowd gives you access to all these users and groups in your underlying LDAP directories.

Public Signup Setup

This tip applies if you:

- Have public-facing JIRA, Confluence and Bamboo servers and private LDAP repositories.
- Allow public signup via Jira, Confluence and/or Bamboo.
- Want to partition where users are created via the public signup functionality.

Crowd allows for multiple directories to be assigned to an application. Follow these steps to direct all public signups into your chosen Crowd directory:

1. Define two directories in Crowd:
 - a. An internal directory for 'public' users.
 - b. An LDAP directory for staff and contractors.
2. Assign both these directories to the 'Jira' application in Crowd. (See [Mapping a Directory to an Application](#).)
3. Use the 'ordering' arrows to move the internal 'public' directory into the first position. (See [Specifying the Directory Order for an Application](#).)
4. Grant the 'Add User' permission to the 'Jira' application in the internal 'public' directory. (See [Specifying an Application's Directory Permissions](#).)
5. Ensure that the 'Add User' permission is disabled for the 'Jira' application in the private LDAP directory.

Using this configuration, when Crowd receives a request from Jira to create a user, Crowd will create the user in the 'public' internal directory only.

 Unless otherwise instructed, Crowd will add the user to **all** directories assigned to the 'Jira' application. The above steps allow you to ensure that the signed-up users are added to your 'public' directory only.

IBM Lotus Domino Integration

Customers have reported successful Crowd integration with [IBM Lotus Domino](#). For more information, take a look at [CWD-125](#).

i The Atlassian Crowd team does not officially support this integration, because we do not have test environments set up for Lotus Domino.

IBM Websphere Integration

If your client application is running in Websphere, there is a known problem with Websphere's XML libraries.

Crowd uses [XFire](#) to handle the requests between the client application (Jira, Confluence, Bamboo etc.) and Crowd, XFire requires a newer version of an XML library than what is shipped with Websphere 5.1.

More information and a link to a newer version of the relevant JAR file is available on the [XFire website](#)

You will need to add the **qname.jar** file to the `WebSphere\AppServer\lib` directory and remove the old file.

Some users have also reported errors like the following:

```
java.lang.VerifyError:  
(class: org/codehaus/xfire/aegis/type/basic/ObjectType, method: writeSchema signature:  
(Lorg/jdom/Element;)V) Incompatible argument to method
```

This is related to the following [XFire issue](#) the suggested fix for this is to upgrade the version of JDOM that is shipped with Websphere to something greater than 1.0 (Websphere ships with JDOM Beta 6).

If you add a later version of [JDOM](#) to the `WebSphere\AppServer\lib` directory and remove the old version, this should fix the above problem.

Support Policies

Welcome to the support policies index page. Here, you'll find information about how Atlassian Support can help you and how to get in touch with our helpful support engineers. Please choose the relevant page below to find out more.

- [Bug Fixing Policy](#)
- [How to Report a Security Issue](#)
- [New Features Policy](#)
- [Security Advisory Publishing Policy](#)
- [Security Bugfix Policy](#)
- [Security Patch Policy](#)
- [Severity Levels for Security Issues](#)

To request support from Atlassian, please raise a support issue in our online support system. To do this, visit support.atlassian.com, log in (creating an account if need be) and create an issue under Crowd. Our friendly support engineers will get right back to you with an answer.

Bug Fixing Policy

Summary

- Our Support team will help with workarounds and bug reporting
- We'll generally fix critical bugs in the next maintenance release
- We schedule non-critical bugs according to a variety of considerations

[Report a bug](#)

Developing an app (add-on) for an Atlassian product or using one of our APIs? Report any related bugs in our [Ecosystem Jira](#).

On this page:

- [Summary](#)
- [Bug fixes for server products](#)
- [Bug reports](#)
- [Search existing bug reports](#)
- [How we approach bug fixing](#)
 - [Severity 1 - Critical](#)
 - [Severity 2 - Major](#)
 - [Severity 3 - Minor](#)
- [About our bug fix workflow](#)
- [How to get access to bug fixes](#)
- [Release terminology for Data Center and server products](#)
 - [Long Term Support releases](#)

Important changes to our server and Data Center products

We've ended sales for new server licenses, and will end support for server on February 2, 2024. We're continuing our investment in Data Center with several key improvements. [Learn what this means for you](#)

Bug fixes for server products

We'll continue to provide bug fixes for server products until February 2, 2022 PT. After this, we'll only provide security bug fixes for critical vulnerabilities until the end of support date on February 2, 2024 PT. [Learn more about these changes](#)

Bug reports

Atlassian Support is eager and happy to help verify bugs we take pride in it! Create an issue in our [support system](#), providing as much information as you can about how to replicate the problem you're experiencing. We'll replicate the bug to verify, then lodge the report for you. We'll also try to construct workarounds if possible.

Search existing bug reports

Use our [public issue tracker](#) to search for existing bugs, add your report, and watch the ones that are important to you. When you watch an issue, we'll send you an e-mail notification when the issue's updated.

How we approach bug fixing

Bug fix releases are more frequent than feature releases, and target the most critical bugs affecting customers. The notation for a bug fix release is the final number in the version (the 1 in 6.0.1, for example).

We assess each bug based on the symptom severity (that is, when this bug causes symptoms, how severe are those symptoms). There are three levels of symptom severity.

Severity 1 - Critical

Your application is unavailable. Users aren't able to perform their job function, and no workarounds are available.

- login failure affecting all users
- all or most pages don't display
- out of memory errors cause application failure
- significant data loss
- node communication failures
- administration tools fail.

Severity 2 - Major

A feature is unavailable, application performance is significantly degraded, or users job functions are impaired.

- the application performs slowly and fails intermittently
- application is functional, but frequently used gadgets or macros don't work
- application links fail
- specific editing features fail
- or a Severity 1 (critical) issue where there is a viable workaround.

Severity 3 - Minor

The application or specific feature isn't working as expected, but there is a workaround available. Users experience is impacted, but their job function is not impaired.

- some searches fail
- sections of pages load slowly
- administrative features fail intermittently, but a workaround is available
- visual defects, that don't affect function
- minor translation or localization problems
- keyboard shortcuts not functioning as expected.

Assessing bugs using symptom severity makes sure that we prioritise the most impactful fixes. We give high priority to [security issues](#).

About our bug fix workflow

If you watch or mark a bug as affecting your team, its useful to understand how we review, prioritize, and resolve them in our public issue tracker jira.atlassian.com.

We prioritize issues using a metric called User Impact Score (UIS), which is individually calculated for every issue. It takes into account the number of affected users, the severity of the issue, recent interest, and the percentage of users affected per instance. The higher the UIS score, the more pervasive and severe the issue is.

We have also standardised our workflow statuses across Data Center and server products to make it easy for you to see where an issue is at. Heres the current workflow, and a description of each status.



Workflow status	Definition	Phase
Needs triage	This issue is waiting to be reviewed by a member of the Atlassian product team. Typically, only recently created issues are in this status. Our product teams review these issues regularly.	Review
Gathering impact	This issue has been reviewed, but needs more supporting information to gauge how pervasive the problem is.	Prioritization
Long term backlog	A fix for this issue is required, but planned for farther in the future. This is because its not as severe or pervasive as other issues.	
Short term backlog	A fix for this issue is required, and will be prioritised in the near future. This is because its more severe or pervasive than other issues.	
In progress	The development team is currently working on this issue.	Implementation
In review	A fix for this issue has been proposed and is being reviewed and quality-tested by the development team.	
Waiting for release	A fix for this issue has been implemented and is waiting to be shipped in a release.	
Closed	Work on this issue is complete. If its fixed, the resolution will be Fixed and the Fix Version field will indicate the product version that contains the fix. If no code changes were required, the resolution will be Duplicate', 'Won't fix', 'Handled by support', 'Timed out', or similar.	Closure

How to get access to bug fixes

To get access to bug fixes you will need to upgrade to a release that contains the fix.

Release terminology for Data Center and server products

- **Platform release**(example: 4.0) contains significant or breaking changes. For example changes or removal of existing APIs, significant changes to the user experience, or removal or a major feature.
- **Feature release**(example: 4.6) can contain new features, changes to existing features, changes to supported platforms (such as databases, operating systems, Git versions), or removal of features. These were previously referred to as 'major' releases by most products.

- **Bugfix release**(example: 4.6.2) can contain bug fixes and stability and performance improvements. Depending on the nature of the fixes they may introduce minor changes to existing features, but do not include new features or high-risk changes, so can be adopted quickly. We recommend regularly upgrading to the latest bugfix release for your current version. These were previously referred to as 'maintenance' releases by most products.

In addition to the three main release types, a feature release can also be designated a **Long Term Support release**(formerly known as an Enterprise release), which means it will receive bug fixes for a longer period of time than a standard feature release.

Long Term Support releases

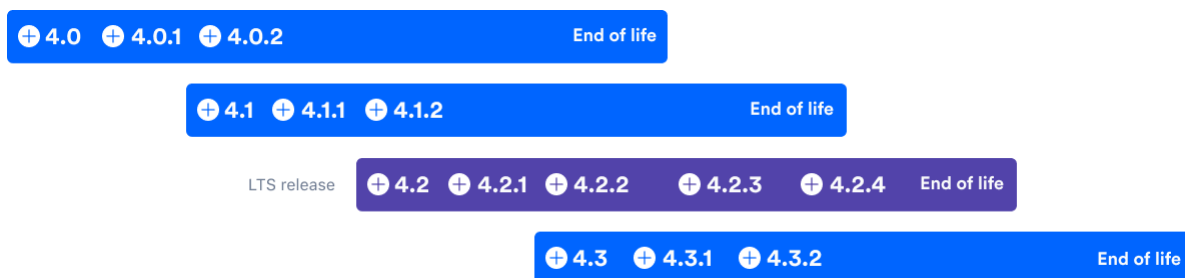
Long Term Support releases (formerly known as Enterprise releases) are for Server and Data Center customers who prefer to allow more time to prepare for upgrades to new feature versions, but still need to receive critical bug fixes. If you only upgrade to a new feature version about once a year, a Long Term Support release may be a good fit for your organisation. For Jira Software and Confluence we will:

- Designate a feature release as a Long Term Support release, at least every 12 months.
- Backport critical security fixes, as outlined in our current security bug fix policy, and fixes relating to stability, data integrity or critical performance issues.
- Make bug fix releases available for the designated version until it reaches end of life.
- Provide a change log of all changes between one Long Term Support release and the next to make upgrading easier.

Not all bug fixes will be backported. We'll target the bugs and regressions that we deem most critical, focusing on stability, data integrity, or performance issues. There may also be some fixes that we choose not to backport due to risk, complexity or because the fix requires changes to an API, code used by third party apps (also known as add-ons), or infrastructure that we would usually reserve for a platform release.

For Jira Software Data Center customers, we'll endeavour to allow zero downtime upgrades between one Long Term Support release and the next Long Term Support release, but can't guarantee that down time will not be required, depending on the nature of the changes. The change log will indicate if zero downtime upgrade will be available.

In the example below, version 4.2 has been designated a Long Term Support release. The number of bug fix releases and timing illustrated below is just an example, your product's release cadence may differ.



i Long Term Support changes for server customers

If you have a server license, you'll only be eligible to upgrade to versions released prior to February 2, 2024 PT, when we officially [end support for our server product line](#).

Further reading

See [Atlassian Support Offerings](#) for more support-related information.

How to Report a Security Issue

We've moved!

Go to [Atlassian.com/security](https://atlassian.com/security) for the latest information.

Further reading

See [Atlassian Support Offerings](#) for more support-related information.

New Features Policy

 This policy does not apply to bugs. See our [Server Bug Fix Policy](#) or [Cloud Bug Fix Policy](#) to learn about our approach to bug fixing.

How we choose what to implement

There are many factors that influence our product roadmaps and determine the features we implement. When making decisions about what to prioritize and work on, we combine your feedback and suggestions with insights from our support teams, product analytics, research findings, and more. This information, combined with our medium- and long-term product and platform vision, determines what we implement and its priority order.

How to track when features are implemented

Cloud products

We're continuously improving and updating our Cloud products. To see the latest changes, take a look at the [Atlassian Cloud release notes blog](#).

Data Center products

When a new feature or improvement is scheduled, we'll update the fix version on the relevant Jira issue to indicate the earliest product version that will include the change. This update often happens close to the product release date.

For a summary of changes, see the release notes for your product:

- [Jira Software](#)|[Jira Service Desk](#)|[Jira platform](#)|[Advanced Roadmaps for Jira](#)
- [Confluence](#)|[Questions for Confluence](#)|[Team Calendars for Confluence](#)
- [Bitbucket](#)|[Bamboo](#)|[Fisheye](#)|[Crucible](#)

Server products

We're simplifying our self-managed offerings and sharpening our focus to our cloud and Data Center products. This means we've discontinued new feature development in our server product line. [Learn more about these changes](#)

We'll still be offering bug fixes for server customers with active maintenance. For details, see our [Atlassian Data Center and Server bug fix policy](#).

Product roadmaps

We publish a [public roadmap](#) for Jira Cloud products, Confluence Cloud, Bitbucket Cloud, and our Cloud Platform. This lets you know what's coming soon and what we're thinking about for future updates.

The [Atlassian Cloud release notes blog](#) and [Bitbucket Cloud blog](#) may also contain information on upcoming changes.

We don't provide specific release dates for upcoming changes.

Feature and improvement suggestions

We encourage you to suggest improvements and new features for our products. You can create feature suggestions, or vote, watch, and comment on existing suggestions, at <https://jira.atlassian.com/>.

We get a large number of suggestions and feature requests. Your comments and votes on suggestions help us understand what you're passionate about and how you want our products to support you and your team. The most helpful information you can provide us when commenting on issues is how a particular suggestion would help you. If you describe your use-case to us, and how the suggested change would benefit you and your team, it lets us gain a much deeper understanding of the need behind the suggestion.

Suggestions often have an impact on what we work on, even if we ultimately choose not to implement a suggestion exactly as its described. Our ultimate goal is to understand what you and all of our customers need and to create products that meet those needs. Occasionally, that'll mean implementing a suggestion as described, but it usually means working to understand the need behind the suggestion and how we can meet that need for as many users as possible.

While we endeavor to update and respond to popular suggestions, the volume we receive means there will often be occasions when we can't provide an update or response. We don't provide any compensation or credit for feature suggestions that we implement.

Join the conversation on Atlassian Community

Our Product Managers regularly post articles about new features and changes to the [Atlassian Community](#). You can comment on these posts, ask questions, and discuss with our PMs and other Atlassian users.

Release terminology for Data Center and server products

- **Platform release**(example: 4.0) contains significant or breaking changes. For example changes or removal of existing APIs, significant changes to the user experience, or removal or a major feature.
- **Feature release**(example: 4.6) can contain new features, changes to existing features, changes to supported platforms (such as databases, operating systems, Git versions), or removal of features. These were previously referred to as 'major' releases by most products.
- **Bugfix release**(example: 4.6.2) can contain bug fixes and stability and performance improvements. Depending on the nature of the fixes they may introduce minor changes to existing features, but do not include new features or high-risk changes, so can be adopted quickly. We recommend regularly upgrading to the latest bugfix release for your current version. These were previously referred to as 'maintenance' releases by most products.

In addition to the three main release types, a feature release can also be designated a **Long Term Support release**(formerly known as an Enterprise release), which means it will receive bug fixes for a longer period of time than a standard feature release.

Long Term Support releases

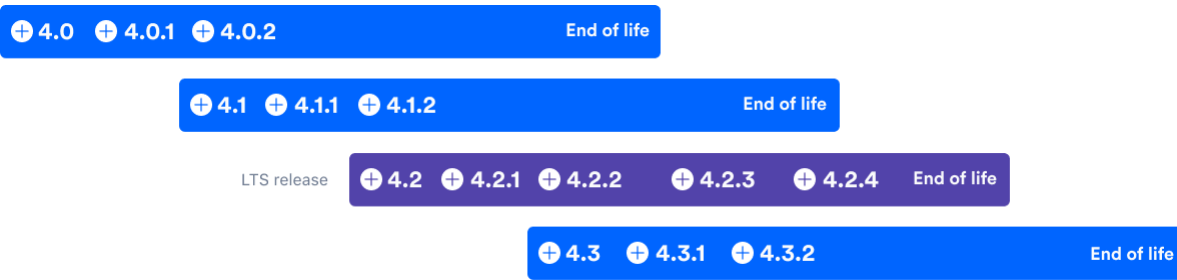
Long Term Support releases (formerly known as Enterprise releases) are for Server and Data Center customers who prefer to allow more time to prepare for upgrades to new feature versions, but still need to receive critical bug fixes. If you only upgrade to a new feature version about once a year, a Long Term Support release may be a good fit for your organisation. For Jira Software and Confluence we will:

- Designate a feature release as a Long Term Support release, at least every 12 months.
- Backport critical security fixes, as outlined in our current security bug fix policy, and fixes relating to stability, data integrity or critical performance issues.
- Make bug fix releases available for the designated version until it reaches end of life.
- Provide a change log of all changes between one Long Term Support release and the next to make upgrading easier.

Not all bug fixes will be backported. We'll target the bugs and regressions that we deem most critical, focusing on stability, data integrity, or performance issues. There may also be some fixes that we choose not to backport due to risk, complexity or because the fix requires changes to an API, code used by third party apps (also known as add-ons), or infrastructure that we would usually reserve for a platform release.

For Jira Software Data Center customers, we'll endeavour to allow zero downtime upgrades between one Long Term Support release and the next Long Term Support release, but can't guarantee that down time will not be required, depending on the nature of the changes. The change log will indicate if zero downtime upgrade will be available.

In the example below, version 4.2 has been designated a Long Term Support release. The number of bug fix releases and timing illustrated below is just an example, your product's release cadence may differ.



i Long Term Support changes for server customers

If you have a server license, you'll only be eligible to upgrade to versions released prior to February 2, 2024 PT, when we officially [end support for our server product line](#).

Further reading

See [Atlassian Support Offerings](#) for more support-related information.

Security Advisory Publishing Policy

See <https://www.atlassian.com/trust/security> for more information.

Security Bugfix Policy

See [Security @ Atlassian](#) for more information on our security bugfix policy.

Security Patch Policy

We've moved!

Go to [Atlassian.com/security](https://atlassian.com/security) for the latest information.

Further reading

See [Atlassian Support Offerings](#) for more support-related information.

Severity Levels for Security Issues

We've moved!

Go to <https://www.atlassian.com/trust/security/security-severity-levels> for the latest information

Further reading

See [Atlassian Support Offerings](#) for more support-related information.

Troubleshooting



- [Finding Known Issues](#)
 - [Characters in User or Group DN's that will cause problems when using Crowd](#)
 - [Problems when Importing Users into MySQL](#)
 - [Troubleshooting LDAP Error Codes](#)
 - [Active Directory LDAP Errors](#)
 - [Troubleshooting LDAP User Management](#)
 - [Troubleshooting SSL certificates and Crowd](#)
 - [How to Optimize Crowd Client Caching](#)
 - [Troubleshooting Crowd Performance](#)
 - [Troubleshooting SSO with Crowd](#)
 - [Debugging SSO in environments with Proxy Servers](#)
 - [Troubleshooting CrowdID](#)
 - [Provide Crowd Information to Atlassian Support](#)
-
- [Troubleshooting your Configuration on Setup](#)

Finding Known Issues

We track the feature requests and bug reports in the [Crowd project on our JIRA site](#). To find a known issue:

1. Browse the list of [unresolved bugs and requests](#).
2. Click the '**Edit**' button on the left.
3. Under '**Text Search**', type keywords for your problem into the '**Query**' field.
4. Click '**View**' and browse the summaries of the unresolved issues.
5. Click an issue key to view the details of the issue and any fixes or workarounds.

Characters in User or Group DN's that will cause problems when using Crowd

 As of Crowd 2.5.1,  **GWD-2042** - Forward slashes not escaped correctly in DN's etc. **CLOSED** is resolved. This document is now historical.


At present, the `AbstractEncodingFilter` used by Crowd, JIRA and Confluence silently translates certain 'dangerous' characters. The `AbstractEncodingFilter` exists because Microsoft Word uses some special Unicode characters for text (e.g. curly quotes). Not all fonts on non-Windows systems contain these characters. This causes issues in JIRA and Confluence when users copy and paste text from Word into a page or issue. Users on non-Windows systems will see question marks or other odd characters if their fonts don't have these characters.

<http://jira.atlassian.com/browse/CORE-100>

Unfortunately, these translations obviously cause problems when querying for users or groups in Crowd which contain these characters.

<http://jira.atlassian.com/browse/CWD-1152>

There is another ticket tracking problems with certain characters, eg '/':

 **GWD-2042** - Forward slashes not escaped correctly in DN's etc. **CLOSED**

Until we are able to resolve this issue, customers should be aware that user or group DN's that contain the following characters will not work in Crowd:

UTF-8

Decimal ASCII value	AbstractEncodingFilter Replacement Value	Description
183	"_ "	Middle dot, Georgian comma, Greek middle dot
8211	"-"	En dash
8216	"'"	Left single quotation mark
8217	"'"	Right single quotation mark
8220	"\""	Left double quotation mark
8221	"\""	Right double quotation mark
8230	"..."	Horizontal ellipsis, three dot leader

ISO-8859-1

Decimal ASCII value	AbstractEncodingFilter Replacement Value	Description
133	"..."	Horizontal ellipsis, three dot leader
145	"'"	Left single quotation mark
146	"'"	Right single quotation mark
147	"\""	Left double quotation mark

148	"\""	Right double quotation mark
150	"-"	En dash

Problems when Importing Users into MySQL

If your Crowd installation is using a [MySQL database](#), you may find that the [user and group import process](#) does not perform a complete import.

To solve this problem, please check the transaction level in your MySQL startup options, as defined in the `my.cnf` configuration file. See the [Crowd MySQL configuration guide](#) for instructions.

Troubleshooting LDAP Error Codes

Useful Links for translating LDAP Error codes:

- [LDAP Error Codes](#)
- [How LDAP Error Codes Map to JNDI Exceptions](#)
- [Active Directory LDAP Errors](#)
- [Novell eDirectory or NDS Error Code List](#)

Active Directory LDAP Errors

AD-specific errors appear after the word "data" and before "vece" or "v893" in the actual error string returned to the binding process*

525	user not found
52e	invalid credentials
530	not permitted to logon at this time
531	not permitted to logon at this workstation
532	password expired
533	account disabled
701	account expired
773	user must reset password
775	user account locked

*This information provided by the following [IBM support document](#).

To enable LDAP logging on your AD server, please review this Microsoft [guide](#).

Troubleshooting LDAP User Management

Scope

This page describes troubleshooting LDAP user management configurations and setup in JIRA Software. Note other specific documents:

Page	Description
Troubleshooting User Management Upgrade Issues	Describes user management issues encountered during an upgrade to JIRA 4.3.x or later.
Connecting to an LDAP Directory	Atlassian's primary documentation on LDAP Directory configuration.

About Apache Directory Studio

[Apache Directory Studio](#) is an open source project of the Apache Software Foundation. It includes an LDAP browser/editor, a schema browser, an LDIF editor, a DSML editor and more. It is a highly useful tool for troubleshooting integration problems with JIRA Software and Confluence. It is an Eclipse RCP application that is cross-platform or run within Eclipse itself as a plugin. For more information on Eclipse, please refer to the Wikipedia article on [Eclipse Software](#).

We recommend using it for testing as a method of isolating where the problem with an integration exists. It allows you to test if an application other than JIRA Software or Confluence can connect to the LDAP/AD server. If a successful connection cannot be established it can indicate problems on the side of the directory, if it can connect it can indicate problems with the configuration of JIRA Software or Confluence.

On this page:

- [Installation and Set Up](#)
- [Browsing the Directory](#)
 - [Directory Attributes](#)
 - [Go to DN](#)
- [Troubleshooting](#)
 - [Identify Active Directory DNs](#)
 - [Testing the LDAP/ADServer Connectivity](#)
 - [Windows](#)
 - [Linux](#)
 - [Common LDAP Errors](#)
 - [Common Configuration Problems](#)
- [Providing Additional Information](#)
 - [Generating an LDIF Export of a User or Group](#)

Installation and Set Up

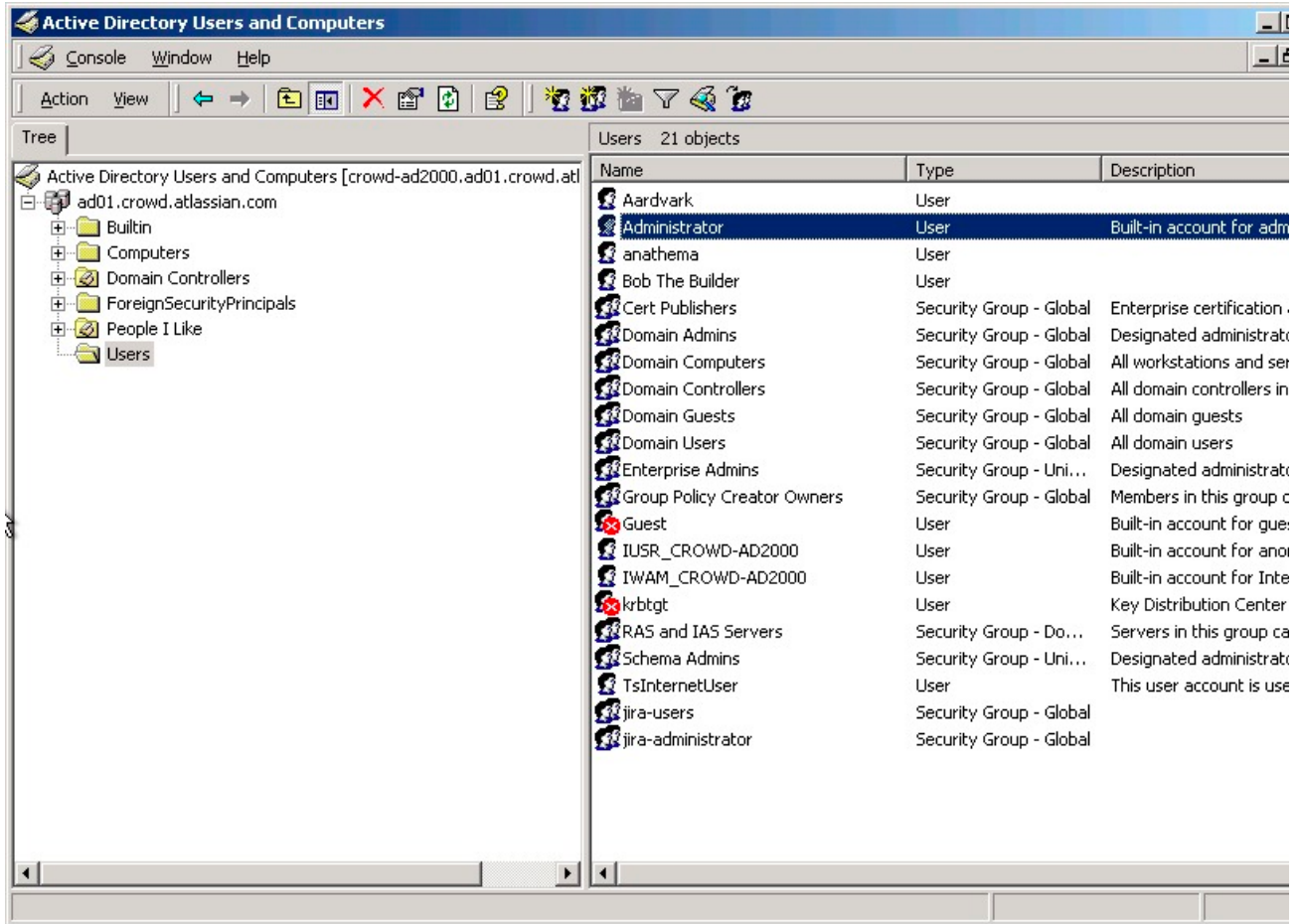
It is recommended to install Directory Studio on a computer other than the server that hosts the LDAP/ADserver (a local workstation is usually best). Directory Studio is able to connect to the directory using the LDAP protocol and display the contents of the directory using its built-in browser. To establish the connection, please refer to the [Create connection](#) section in the Directory Studio user guide.

⊖ If Directory Studio is unable to initiate a connection it can indicate there is a problem with the LDAP/AD server connection. Refer to the troubleshooting section below.

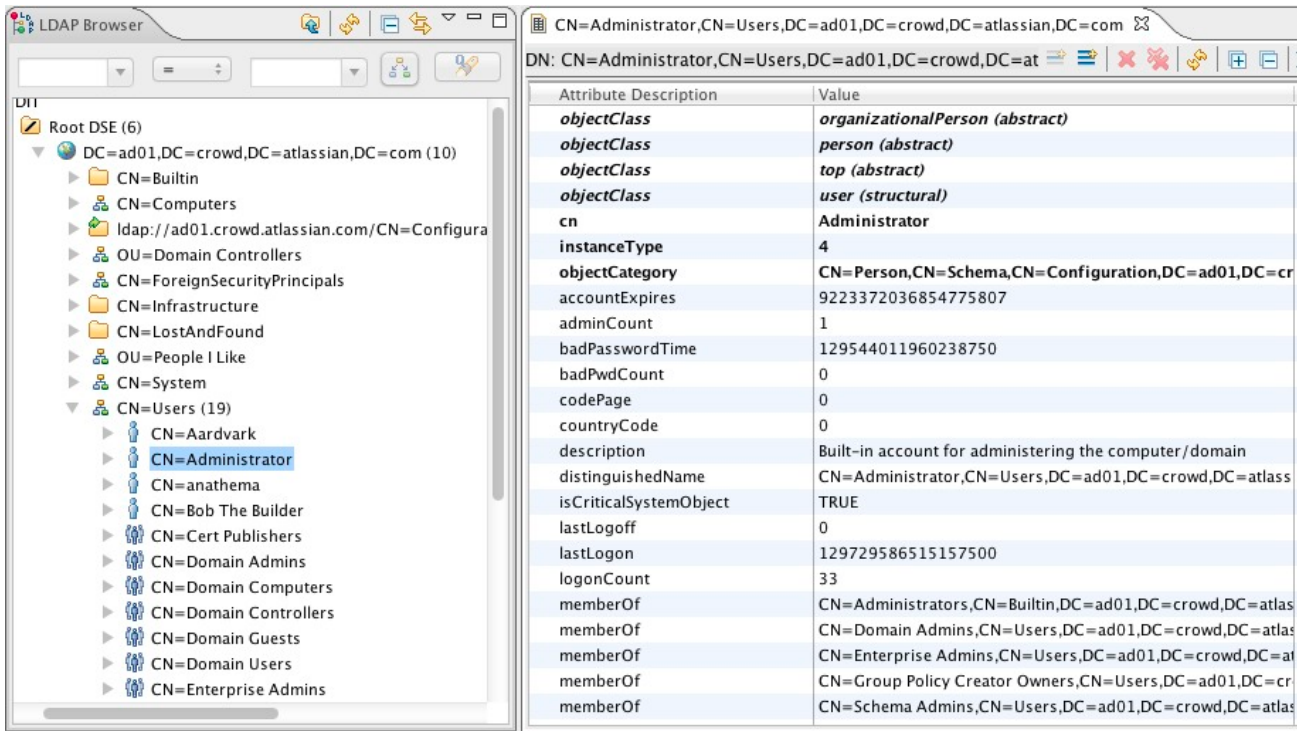
Browsing the Directory

Using the Browser functionality in Directory Studio you can navigate through the Directory tree to see where users and groups are stored. It will also allow you to identify how data is stored within the directory, including the names of attributes and specific values.

This is a comparison between the Active Directory and Apache Directory Studio. Below is Active Directory:



And this is how it would look in Directory Studio:



As can be seen from the above example, Directory Studio is very useful for identifying which attributes to use when setting the Schema Settings and User Schema Settings as described in [Connecting to an LDAP Directory](#) as it shows the names and values of the attributes in a clear, readable format.

Directory Attributes

LDAP uses a schema (DIT) to define how it stores data. These are much like a database - an attribute holds a value just like a field within a table.

Common attributes:

Attribute Description	Attribute Value
DIT	Directory Information Tree
DC	Domain Controller
DN	Distinguished Name
CN	Common Name
SN	Surname
OU	Organizational Unit
UID	Unique ID

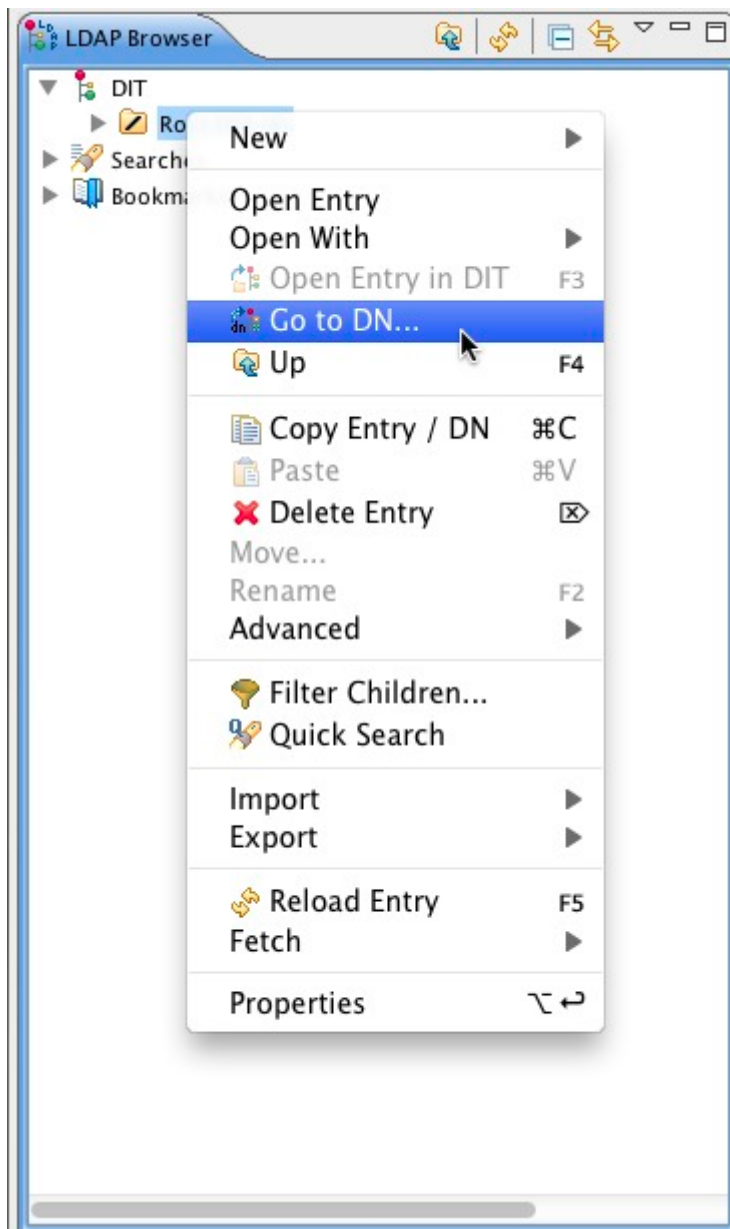
Example of an LDAP/AD tree for the Administrator user:

- **Domain Controller:** ad01,crowd,atlassian,com
 - **Common Name:** Users
 - **Common Name:** Administrator

Distinguished Name: CN=Administrator,CN=Users,DC=ad01,DC=crowd,DC=atlassian,DC=com

Go to DN

The Go to DN is essentially a search option that allows you to find an LDAP element easily. It can be accessed through the context menu in the LDAP Browser (right click). It is useful for quickly finding Distinguished Names that would relate to configuration settings, such as Base DN, Additional User DN or Additional Group DN.



Troubleshooting

Prior to troubleshooting, please ensure that you have verified the following credentials of the LDAP/AD server with your System Administrator. If they are not correct, you will not be able to successfully connect or bind to the LDAP/AD server.

- Server Hostname (or IP).
- Server Port.
- Bind Username (DN).
- Bind Password.

A successful connection requires the following two steps:

1. **Server Connectivity:** the application is able to communicate with the LDAP/AD server.
 - Uses Hostname and Port.
2. **LDAP/AD Bind:** the application is able to log into the directory.
 - Uses Bind Username & Password.

Identify Active Directory DNs

If using Active Directory, the `dsquery` command can be executed on the command-line to identify DNs to particular objects, as per this [Dsquery Windows Server](#) article. Please refer to the examples below for further information.

Access the command line and execute the following to identify the DN for a user:

```
C:\>dsquery user -name Administrator
"CN=Administrator,CN=Users,DC=sydney,DC=atlassian,DC=com"
```

In the above example, this breaks down to:

- **Domain Controller:**sydney,atlassian,com
 - **Common Name:**Users
 - **Common Name:**Administrator

The following can then be used to identify the DN for any groups:

```
C:\>dsquery group -name jira-administrators
"CN=jira-administrators,OU=Groups,DC=sydney,DC=atlassian,DC=com"
```

In the above example, this breaks down to:

- **Domain Controller:**sydney,atlassian,com
 - **Organizational Unit:**Groups
 - **Common Name:**jira-administrators

Using the the above results, when entering the configuration for an Active Directory Connector in JIRA Software, the following DNs would be used:

Configuration	Parameter
Base DN	DC=sydney,DC=atlassian,DC=com
Additional User DN	CN=Users
Additional Group DN	OU=Groups

The above settings would synchronize all users in `CN=Users,DC=sydney,DC=atlassian,DC=com` and all groups in `OU=Groups,DC=sydney,DC=atlassian,DC=com`.

Testing the LDAP/ADServer Connectivity

If any of the below tests fail it indicates there is most likely a problem with the LDAP/AD Server or the port/IP is not correct. Troubleshooting those problems is outside of the scope of this document.

Testing the LDAP/AD Bind is done through Directory Studio.

Windows

Test that the server is reachable by pinging it.

```
C:\>ping -n 1 ldap-host
Pinging ldap-host.example.com [192.168.1.100] with 32 bytes of data:
Reply from 192.168.1.100: bytes=32 time=4ms TTL=127
Ping statistics for 172.20.4.167:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 4ms, Maximum = 4ms, Average = 4ms
```

Check that the port is open (typically 389). A successful connection to the port indicates it is open. If a successful connection is made, the screen will go blank. If not, the below message will be generated. To exit telnet, use CTRL+C.

```
C:\>telnet ldap-host 389
Connecting To ldap-host...Could not open connection to the host, on port 389: Connect failed
```

Linux

Test that the server is reachable by pinging it.

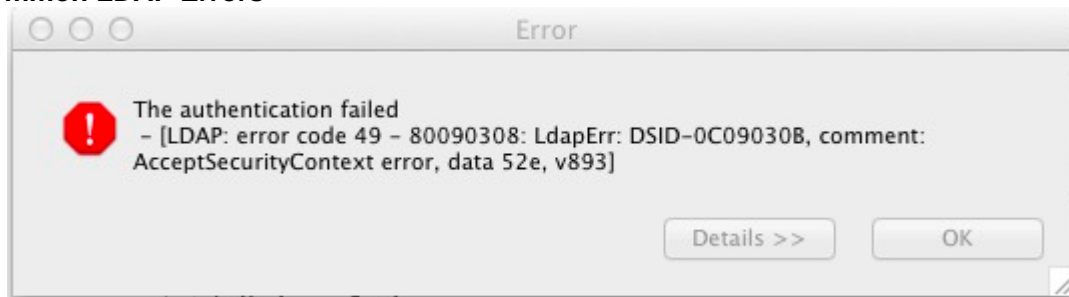
```
$ ping -c1 ldap-host
PING ldap-host (192.168.1.100): 56 data bytes
64 bytes from 192.168.1.100: icmp_seq=0 ttl=127 time=0.521 ms

--- crowd-adl ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.521/0.521/0.521/0.000 ms
```

Check that the port is open (typically 389). A successful connection to the port indicates it is open. To exit telnet, use CTRL+C.

```
$ telnet ldap-host 389
Trying 192.168.1.100...
Connected to crowd-adl.
Escape character is '^'.
```

Common LDAP Errors



These are a list of errors returned from Directory Studio and Embedded Crowd that can help identify configuration issues with the LDAP/AD server.

Error	Possible Cause(s)
simple bind failed: ldap-host:389	This is a very general error, and it means something went wrong when trying to bind to LDAP/AD. Check to see if the LDAP/AD server name and/or port number you have specified is incorrect or an incorrect DN was specified as the administrator username. It could also indicate the Encryption method was specified incorrectly in the Network Parameter tab of the connector properties in Directory Studio.
Connection refused (ldap-host:389) No connection	The port number specified for the LDAP/AD server is incorrect or a firewall/network configuration is blocking the connection.

Unknown Host: ldap-host No connection	The LDAP/AD server name/IP specified is incorrect.
<ul style="list-style-type: none"> LDAP: error code 2 InvalidSearchFilterException 	<p>Invalid search filter passed to the LDAP/AD server or the filter is malformed (e.g.: it's missing an open and closing parenthesis).</p> <p>Please refer to this LDAP Query Basics Microsoft TechNet article for further information on LDAP/AD queries.</p>
LDAP: error code 3	The application is timing out when waiting on a response from LDAP. Check the LDAP filters are correct, the LDAP engine is working correctly. The default timeout can be increased in the configuration.
LDAP: error code 4	The Active Directory/LDAP Server that JIRA Software is synchronizing with has a limitation on the number of objects it can return with one query. Enable 'Use Paged Results'.
LDAP: error code 34	The DN is most likely incorrect - confirm this by checking it against Directory Studio or using <code>ds query</code> if using Active Directory.
LDAP: error code 48	The Authentication Method in the Authentication tab of the connector properties is incorrect.
LDAP: error code 32	LDAP_NO_SUCH_OBJECT - Indicates the target object cannot be found. This code is not returned on following operations: Search operations that find the search base but cannot find any entries that match the search filter. Bind operations.
LDAP: error code 49	<p>The Bind Username (DN) or Password is incorrect. This could also be caused by the following:</p> <ul style="list-style-type: none"> Pointed to non-user DN. Pointed to a non-existent DN. Pointed to an incorrect admin DN (might be using UID instead of CN). Pointed to a non administrator user. Pointed to a valid administrator user, however the password is incorrect. <p>A DN describes a path to the object in the database, so the CN may be correct however the OUs and DCs in the DN specified may be incorrect or in the wrong order.</p>
NoSuchAttributeException	This can be caused by providing a name for an attribute that is not correct or does not exist.
InvalidNameException	<p>This can be caused by a bad prefix specified in the Settings tab on most LDAP/AD systems.</p> <p>This could mean you did not specify a prefix at all, which means the LDAP/AD server did not receive a full DN from CPS or that you did not specify a correct prefix, such as CN instead of UID, which results in the LDAP/AD server not receiving a correct DN from CPS.</p> <p>Can also be caused by a missing comma at the beginning of the suffix or an extra comma at the end of the suffix. This error could also mean the authentication type is incorrect.</p>
AuthenticationException	Could not authenticate the user trying to login. This can be the result of an incorrect username or password or an incorrect prefix and/or suffix specified in the Settings tab, depending on the type of LDAP/AD system. Could also mean the authentication type is incorrect.

Common Configuration Problems


Problem	Possible Cause(s)	Possible Fix
<ul style="list-style-type: none"> JIRA Software runs out of memory, experiences slow performance or high CPU load during User Directory sync. Users are imported, however groups are not. Groups are imported, however users are not. 	<p>The Schema Settings may be incorrect causing JIRA Software to attempt to import unnecessary or incorrect data from the LDAP /AD server.</p>	<p>Verify by browsing with Directory Studio (using 'Go to DN' is recommended) and ensure the following settings are correct within the Schema Settings in JIRA Software:</p> <ul style="list-style-type: none"> Base DN Additional User DN Additional Group DN
<p>User details (e.g.: Username, Email, Full Name, Password) are not correct after a successful sync.</p>	<p>The User Schema Settings may be incorrect or the value within the attributes is incorrect.</p> <ul style="list-style-type: none"> If this applies to only one user, it is likely the user details are incorrect. If it applies to multiple, it is likely the JIRA Software configuration is incorrect. 	<p>Verify by browsing with Directory Studio (using 'Go to DN' is recommended) and ensure the following attributes are correct within the User Schema Settings in JIRA Software:</p> <ul style="list-style-type: none"> User Name User First Name User Last Name User Display Name User Email
<p>Users and Groups are imported, however Users aren't automatically added to the Groups they belong to.</p>	<p>The Membership Schema Settings may be incorrect.</p> <ul style="list-style-type: none"> If this applies to only one user, it is likely the user details are incorrect. If it applies to multiple, it is likely the JIRA Software configuration is incorrect. 	<p>Verify by browsing with Directory Studio (check with a sample user) and ensure the following attributes are correct within the Membership Schema Settings in JIRA Software:</p> <ul style="list-style-type: none"> Group Members Attribute User Membership Attribute
<p>Unable to add/delete Users or make changes to their Groups, Project Roles or Edit them.</p>	<p>The LDAP/AD User Directory is set to Read Only.</p>	<p>Verify the Permissions Settings are correct for the external User Directory.</p> <p>If you want to make changes, Read/Write will need to be activated. If using Active Directory, please refer to this page on Configuring an SSL Certificate.</p>

Providing Additional Information

Often when the above troubleshooting steps don't resolve the problem and in particularly complicated scenarios, further information will need to be provided so Atlassian can investigate further.

Generating an LDIF Export of a User or Group

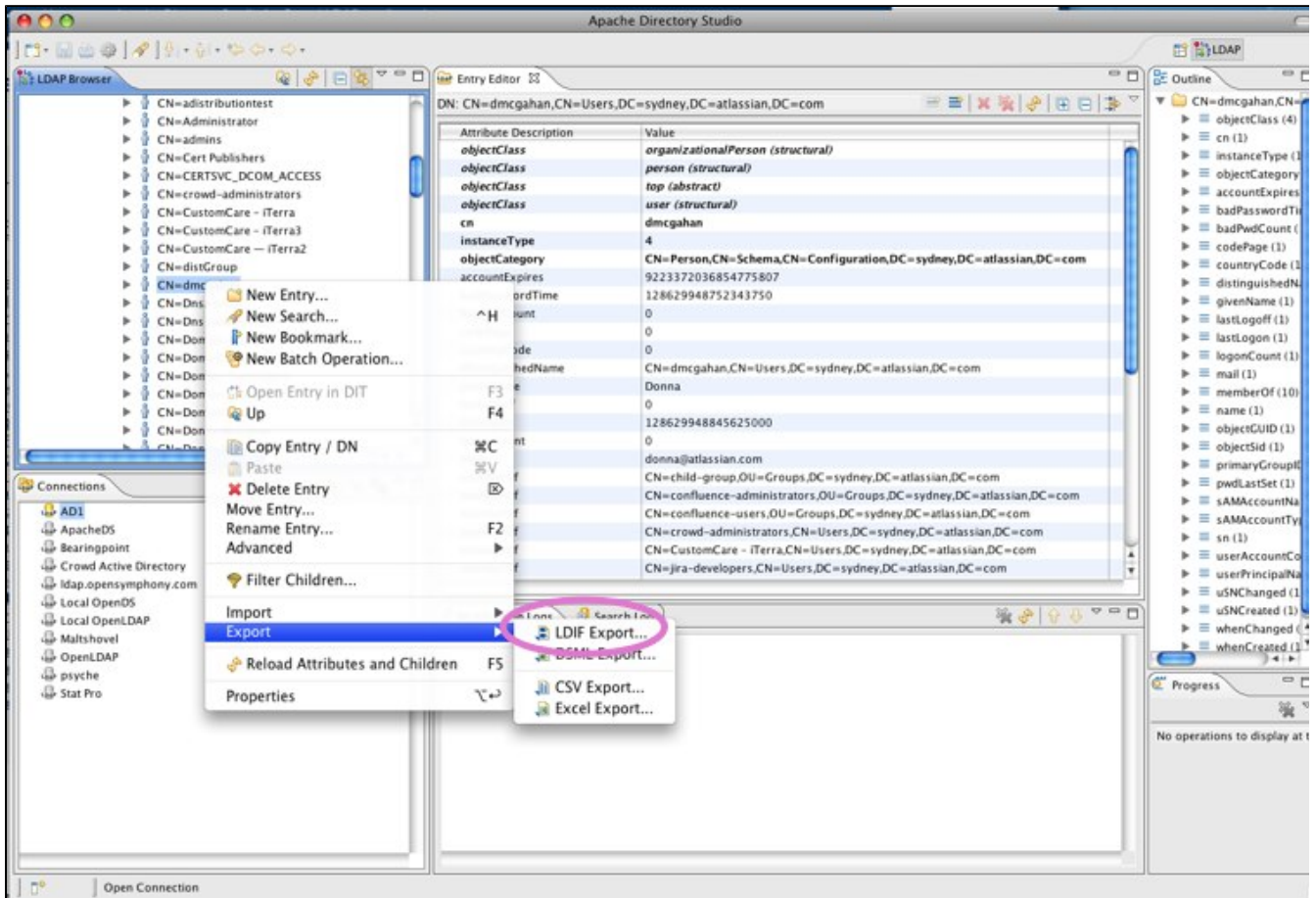
Occasionally, Atlassian Crowd Support may request an LDIF export of a user or group. LDIF is the LDAP Data Interchange Format. You can export all or part of your LDAP directory to an LDIF file. This page shows you how to do that when using [Apache Directory Studio](#).

 LDAP browser [Apache Directory Studio](#) may convert the underlying value data automatically which could mask the real value. An LDIF will show the actual value that's being used.

To generate an LDIF export of a user or group,

1. Highlight the user or group in [Apache Directory Studio](#).
2. Right-click on the user or group.
3. Choose **Export -> LDIF Export**.

Screenshot: Generating an LDIF export of a user in Apache Directory Studio



RELATED TOPICS

[Creating a Connection to your LDAP Directory Using Apache Directory Studio for LDAP Configuration](#)

Troubleshooting SSL certificates and Crowd

1. Ensure that you are not using any parameters in the `JAVA_OPTS` variable that refer to your keystore. For example,

```
-Djavax.net.ssl.trustStore="/my/key/store"
```

The `JAVA_OPTS` variable is normally located in the Crowd distribution's `apache-tomcat/bin/setenv.sh` or `setenv.bat` file (depending on the OS you are using). Remove these references and restart Crowd.

2. Run this command on the Crowd server, replacing `<ip address of LDAP server>` with your LDAP server's IP address:

```
openssl s_client -connect <ip address of LDAP server>:636
```

3. Save the certificate (including the `BEGIN CERTIFICATE` and `END CERTIFICATE` lines) of the response into a local file called `tmp.pem`.

4. Run this command on the local `tmp.pem` file. This should return an MD5 Fingerprint value.

```
openssl x509 -fingerprint -md5 -noout -in tmp.pem
```

5. Run this command on the Crowd server. This assumes you are using the default keystore and the `$JAVA_HOME` (or for Windows `%JAVA_HOME%`) variable has been set. If not, please specify the correct keystore path.

```
keytool -list -keystore $JAVA_HOME/jre/lib/security/cacerts
```

6. Ensure that the MD5 Fingerprint from step 3 is listed in your keystore. If it is not, you will need to import the `tmp.pem` certificate into your keystore.

If you continue to experience issues with your SSL configuration and Crowd, please [open a new support issue](#). Attach the `CROWD_APPLICATION_DIRECTORY/atlassian-crowd.log` file and the output of the tests above to the support issue.

How to Optimize Crowd Client Caching

Crowd-integrated applications can store user, group and role data in a local cache. This helps improve the performance of Crowd since these applications do not have to repeatedly request information from Crowd. Generally, it is not necessary to configure application caching, although this depends on the size of your application deployments. But for larger installations, you may need to configure the application caching. Please refer to more information about:

- [An overview of the different caching options in Crowd](#)
- [Configuring caching for an application.](#)
- [Troubleshooting the caching for Atlassian integrated applications.](#)
- [Caching of user permissions on the Crowd server.](#)
- [Caching for LDAP directories](#)

Troubleshooting Crowd Performance

Please note:

This guide assumes you have already opened a Crowd support issue at <http://support.atlassian.com> and wish to provide additional information about your Crowd configuration in this issue.

1. The Crowd application is slow!

1. Ensure you are running [the latest version](#) of Crowd.
2. Under **Admin > Logging & Profiling** in Crowd:
 - Change the `com.atlassian.crowd` package to `DEBUG`.
 - Enable profiling.
3. Replicate the performance issues you are seeing in Crowd (e.g. log out and log in, browse users, etc.)
4. Attach the resulting `CROWD_DIRECTORY/atlassian-crowd.log` file to your support ticket.
5. List the directories and applications active in your Crowd instance.
6. Provide rough estimates of the number of users and groups that are available in each LDAP directory configuration.
7. Provide information about the network location of any LDAP servers in respect to the Crowd server (e.g. same subnet, different networks, different states).
8. If using Active Directory, is SSL enabled?

2. Jira/Confluence is slow!

1. Confirm that [data caching is enabled](#) in Crowd.
2. Confirm that the only `crowd-integration-client` JAR in the `Jira/Confluence WEB-INF/lib` directory matches the version of Crowd you are running (e.g. `crowd-integration-client-1.5.jar`).
3. Confirm that the `crowd-ehcache.xml` file located in the `Jira/Confluence WEB-INF/classes` directory matches the one in the `CROWD/client/conf` directory.
4. If your Crowd installation contains more than 50,000 users, review the guide at [Configuring Caching for an Application](#).

a. Jira/Confluence still slow?

1. Stop Jira/Confluence.
2. Temporarily replace the `WEB-INF/lib/crowd-integration-client-1.x.JAR` file with the appropriate version from this [issue](#).
3. Restart Jira/Confluence.
4. Under **Admin > Logging & Profiling** in Jira/Confluence:
 - Change the `com.atlassian` package to `DEBUG`.
 - Enable profiling.
5. Perform actions in Jira/Confluence that are slow to respond (e.g. log out and log in, browse users, etc.)
6. Attach the resulting `Jira/Confluence logs/catalina.out` or `stdout.log`. If Confluence, also attach the `atlassian-confluence.log` file in the Confluence home directory (specified in the `confluence-init.properties` file at setup).
7. List the directories and applications active in your Crowd instance for the Jira/Confluence application.
8. Provide rough estimates of the number of users and groups that are available in each LDAP directory configuration for the Jira/Confluence application.
9. Provide information about the network location of any LDAP servers in respect to the Crowd server (e.g. same subnet, different networks, different states).

b. Using Active Directory?

1. Is SSL enabled?
2. Are you using nested groups (is the Use Nested Groups box checked in Crowd)?
3. If login is slow, please connect to your AD server using [Apache Directory Studio](#) and highlight the username used for this login. Provide a screenshot of this user especially the list of `memberOf` attributes for this account (should contain full DNs).
4. Please also confirm that all domain controllers referenced in these groups are resolvable/reachable from the Crowd server using ping:

```
ping ad1.mycompany.com  
ping ad2.mycompany.au
```

RELATED TOPICS

- [Overview of Caching](#)
- [Configuring Caching for an Application](#)
- [Authorization Caching](#)
- [Configuring Caching for an LDAP Directory](#)

Troubleshooting SSO with Crowd

Please follow the steps below to troubleshoot problems with SSO (single sign-on) in Crowd:

1. Confirm that you can log in to each application with the same username and password.
 - In Crowd, click '**Applications**' to view the Application Browser.
 - Click '**View**' next to the application.
 - Click the '**Authentication Test**' tab and follow [these instructions](#).
2. Set each application to use centralized SSO authentication, as follows. Ensure that each Atlassian application's `WEB-INF/classes/seraph-config.xml` file is using the Crowd's `com.atlassian.crowd` authenticator class. For example in Jira, instead of this:

```
<authenticator class="com.atlassian.jira.security.login.JiraSeraphAuthenticator"/>
```

you should have this:

```
<authenticator class="com.atlassian.jira.security.login.SSOseraphAuthenticator"/>
```

Please, see our [Adding an Application Tutorial](#) page to check the SSO authenticator classes for other applications.

3. If you are using a reverse proxy in front of any of the applications, you'll need to make sure that the host header is preserved in the forward. For example, in an Apache reverse proxy, you need to enable the "ProxyPreserveHost" option, and in IIS you need to use "Application Request Routing" to achieve the same.
4. Once each application is using centralized authentication, confirm you can log in to each application with the same username and password.
5. Inside of Crowd, ensure that each application is configured to use the same user directory. SSO will not work if you log in to Confluence through one user directory, but Jira through a different user directory, *even if the usernames are identical*.
6. Ensure that each application is using the same sub-domain. For example:
 - **Jira** -> jira.example.com
 - **Confluence** -> confluence.example.com
 - **Crowd** -> crowd.example.com

i SSO will only work with applications on the same sub-domain. Why? Crowd uses a cookie to manage SSO and your browser only has access to cookies in the same sub domain, e.g. *.[example.com](#).

This is the value that you set in the Domain property (e.g. [.example.com](#)) for Crowd to enable SSO. This is covered in the documentation on [configuring the domain](#).

7. Check if the "Require Consistent Client IP Address" is enabled in the [Session configuration](#). If it is, try disabling it and test SSO.

Still having trouble?

If the above steps have not solved your problem, please gather some debugging information as described below before contacting Atlassian support:

1. In Crowd, go to '**Administration**' -> '**Logging & Profiling**'. Change the `com.atlassian.crowd` package to DEBUG.
2. Replicate the SSO problem you are having.
3. Please raise a support issue on our [Support System](#), attaching your `{CROWD_HOME}/logs/atlassian-crowd.log` file with the debug information gathered.

RELATED TOPICS

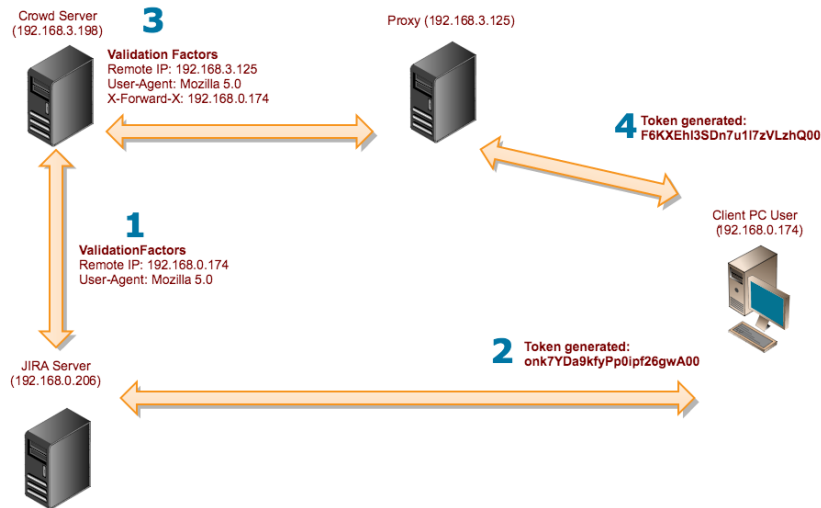
[Overview of SSO](#)

Debugging SSO in environments with Proxy Servers

This is an example log file from Crowd 1.6 with Debugging turned On for *com.atlassian.crowd* under Admin > Logging & Profiling. In this example, I've logged into Crowd Console, then attempt to access Jira.

Example of non-working SSO Configuration

In this example, **admin** signs into Crowd Console, and then visits Jira. Jira is being served behind a Apache proxy (mod_proxy for example).



Login to Crowd directly without a proxy

Crowd detects a user logging in for the first time from the IP address 192.168.0.174, with a Mozilla Browser on Linux. A token of **onk7YDa9kfyPp0ipf26gwA00** is generated from the **ValidationFactors** consisting of an IP address, User-Agent, Random Number.

```
[atlassian.crowd.authentication.TokenKeyGeneratorImpl] Generating Token for principal: admin
[atlassian.crowd.authentication.TokenKeyGeneratorImpl] Adding User-Agent of com.atlassian.crowd.
integration.authentication.ValidationFactor@429340[name=User-Agent,value=Mozilla/5.0 (X11; U; Linux
i686; en-US; rv:1.8.0.9) Gecko/20070316 CentOS/1.5.0.9-10.el5.centos Firefox/1.5.0.9 pango-text]
[atlassian.crowd.authentication.TokenKeyGeneratorImpl] Adding remote address of 192.168.0.174
[atlassian.crowd.authentication.TokenKeyGeneratorImpl] Adding Random-Number of com.atlassian.crowd.
integration.authentication.ValidationFactor@d171c2[name=Random-Number,value=8162711822532519761]
[crowd.manager.application.ApplicationServiceGeneric] Current Validation Factors:
com.atlassian.crowd.integration.authentication.ValidationFactor@83b064[name=remote_address,value=192.
168.0.174]com.atlassian.crowd.integration.authentication.ValidationFactor@429340[name=User-Agent,
value=Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.8.0.9) Gecko/20070316 CentOS/1.5.0.9-10.el5.centos
Firefox/1.5.0.9 pango-text]
[crowd.manager.application.ApplicationServiceGeneric] comparing existing token com.atlassian.crowd.model.
token.Token@ca8293[ID=524387,key=onk7YDa9kfyPp0ipf26gwA00,name=admin,
secretNumber=8162711822532519761,Directory ID=98305] with a validation token com.atlassian.crowd.
model.token.Token@b1b30c[ID=0,key=onk7YDa9kfyPp0ipf26gwA00,name=admin,
secretNumber=8162711822532519761,Directory ID=98305]
[crowd.manager.application.ApplicationServiceGeneric] they match
[crowd.manager.application.ApplicationServiceGeneric] user has access to the application crowd
```

Login to Jira via proxy

After my visit to the Crowd Console, I then visit Jira through a proxy. It detects my same User-Agent, but now sees that my IP is **192.168.3.125** which is really the proxy's. This results in a token that doesn't match my existing one: **F6KXEhI3SDn7u1I7zVLzhQ00** as compared to **onk7YDa9kfyPp0ipf26gwA00** and thus, I'm prompted to login again. A clue that I was going through the proxy is the **X-Forwarded-For** header. It also contains my real IP. The way to fix this is to add **192.168.3.125** to my list of [Trusted Proxies](#).

```
[atlassian.crowd.authentication.TokenKeyGeneratorImpl] Generating Token for principal: admin
[atlassian.crowd.authentication.TokenKeyGeneratorImpl] Adding User-Agent of com.atlassian.crowd.
integration.authentication.ValidationFactor@31f633[name=User-Agent,value=Mozilla/5.0 (X11; U; Linux i686;
en-US; rv:1.8.0.9) Gecko/20070316 CentOS/1.5.0.9-10.el5.centos Firefox/1.5.0.9 pango-text]
[atlassian.crowd.authentication.TokenKeyGeneratorImpl] Adding remote address of 192.168.3.125
[atlassian.crowd.authentication.TokenKeyGeneratorImpl] Adding Random-Number of com.atlassian.crowd.
integration.authentication.ValidationFactor@1a99a7[name=Random-Number,value=8162711822532519761]
[crowd.manager.application.ApplicationServiceGeneric] Current Validation Factors:
com.atlassian.crowd.integration.authentication.ValidationFactor@5db889[name=remote_address,value=192.
168.3.125]com.atlassian.crowd.integration.authentication.ValidationFactor@fe4748[name=X-Forwarded-For,
value=192.168.0.174]com.atlassian.crowd.integration.authentication.ValidationFactor@31f633[name=User-
Agent,value=Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.8.0.9) Gecko/20070316 CentOS/1.5.0.9-10.el5.
centos Firefox/1.5.0.9 pango-text]
[crowd.manager.application.ApplicationServiceGeneric] comparing existing token com.atlassian.crowd.model.
token.Token@417bf8[ID=524387,key=onk7YDa9kfyPp0ipf26gwA00,name=admin,
secretNumber=8162711822532519761,Directory ID=98305] with a validation token com.atlassian.crowd.
model.token.Token@f9d0af[ID=0,key=F6KXEhI3SDn7u1I7zVLzhQ00,name=admin,
secretNumber=8162711822532519761,Directory ID=98305]
[crowd.manager.application.ApplicationServiceGeneric] The token keys don't match
```

Troubleshooting CrowdID

If you are experiencing issues with Crowd's OpenID server (CrowdID), please take the following steps to help diagnose the problem:

Step 1: Change the logging for Crowd's OpenID server and client.

- Change the openid package from INFO to DEBUG in CROWD/crowd-openidserver-webapp/WEB-INF/classes/log4j.properties

```
#####
# CROWD - CLASS-SPECIFIC LOGGING LEVELS
#####

# Set the following lines to DEBUG to enable logging on incoming, outgoing and fault SOAP messages
log4j.logger.com.atlassian.crowd.integration.service.soap.xfire.XFireInLoggingMethodHandler=WARN
log4j.logger.com.atlassian.crowd.integration.service.soap.xfire.XFireOutLoggingMethodHandler=WARN
log4j.logger.com.atlassian.crowd.integration.service.soap.xfire.XFireFaultLoggingMethodHandler=WARN

log4j.logger.com.atlassian.crowd.openid=DEBUG

log4j.logger.com.atlassian.crowd=INFO
```

- Change the openid package from INFO to DEBUG in CROWD/crowd-openidclient-webapp/WEB-INF/classes/log4j.properties

```
#####
# CROWD - CLASS-SPECIFIC LOGGING LEVELS
#####

log4j.logger.com.atlassian.crowd.openid=DEBUG

log4j.logger.com.atlassian.crowd=INFO
```

Step 2: Test CrowdID with the bundled OpenID client:

- `http://<your Crowd URL>:<Crowd port>/openidclient/`

If these tests are not successful, attach the `atlassian-crowd-openid-client.log` and `atlassian-crowd.openid-server.log` files (in the same location specified by [this guide](#)) to a support issue at <http://support.atlassian.com>. Note the username of the account tested.

Step 3: Test CrowdID with your OpenID application:

If these tests are not successful, attach the `atlassian-crowd-openid-client.log` and `atlassian-crowd.openid-server.log` files (in the same location specified by [this guide](#)) to a support issue at <http://support.atlassian.com>. Note the username of the account tested and the OpenID application you are attempting to use.

Provide Crowd Information to Atlassian Support

Purpose

Sending us your log files and configuration details will speed up resolution of your issue.

Solution

Please provide these artifacts when you create a ticket:

- A zip archive of <crowd_home>/logs/
- A zip archive of <crowd_install>/apache-tomcat/logs/
- All the text from **Crowd Admin > System Information > Support Information**.

The <crowd_home> directory is the path defined in <crowd_install>/crowd-webapp/WEB-INF/classes/crowd-init.properties.

The text can be attached in a file or pasted into the description or comment of your ticket at <https://support.atlassian.com>.

If the zip files are less than 125MB you can attach them to your ticket. For larger files please request a transfer account and your support agent create one for you.

Contributing to the Crowd Documentation

Would you like to share your Crowd hints, tips and techniques with us and with other Crowd users? We welcome your contributions.

On this page:

- [Blogging your Technical Tips and Guides](#) [Tips of the Trade](#)
- [Contributing Documentation in Other Languages](#)
- [Updating the Documentation Itself](#)
 - [Getting Permission to Update the Documentation](#)
 - [Our Style Guide](#)
 - [How we Manage Community Updates](#)

Blogging your Technical Tips and Guides [Tips of the Trade](#)

Have you written a blog post describing a specific configuration of Crowd or a neat trick that you have discovered? Let us know, and we will link to your blog from our documentation. [More....](#)

Contributing Documentation in Other Languages

Have you written a guide to Crowd in a language other than English, or translated one of our guides? Let us know, and we will link to your guide from our documentation. [More....](#)

Updating the Documentation Itself

Have you found a mistake in the documentation, or do you have a small addition that would be so easy to add yourself rather than asking us to do it? You can update the documentation page directly.

Getting Permission to Update the Documentation

Please submit the [Atlassian Contributor License Agreement](#).

Our Style Guide

Please read our short [guidelines for authors](#).

How we Manage Community Updates

Here is a quick guide to how we manage community contributions to our documentation and the copyright that applies to the documentation:

- **Monitoring by technical writers.** The Atlassian technical writers monitor the updates to the documentation spaces, using RSS feeds and watching the spaces. If someone makes an update that needs some attention from us, we will make the necessary changes.
- **Wiki permissions.** We use wiki permissions to determine who can edit the documentation spaces. We ask people to sign the [Atlassian Contributor License Agreement \(ACLA\)](#) and submit it to us. That allows us to verify that the applicant is a real person. Then we give them permission to update the documentation.
- **Copyright.** The Atlassian documentation is published under a Creative Commons CC BY license. Specifically, we use a [Creative Commons Attribution 2.5 Australia License](#). This means that anyone can copy, distribute and adapt our documentation provided they acknowledge the source of the documentation. The CC BY license is shown in the footer of every page, so that anyone who contributes to our documentation knows that their contribution falls under the same copyright.

RELATED TOPICS

[Tips of the Trade](#)
[Author Guidelines](#)
[Atlassian Contributor License Agreement](#)

Tips of the Trade

Below are some links to external blog posts and articles containing technical tips and instructions on setting up and using Crowd. This page presents an opportunity for customers and community authors to share information and experiences.

The references here are specific to Crowd and are technical 'how to' guides written by bloggers who use Crowd. For more general information on identity management solutions, best practices and business cases, please refer to the [Atlassian website](#).



Please be aware that these are external blogs and articles.

Most of the links point to external sites, and some of the information is relevant to a specific release of Crowd. Atlassian provides these links because the information is useful and relevant at the time it was written. Please check carefully whether the information is still relevant when you read it, and whether it is relevant to your version of Crowd. **Unless explicitly stated**, Atlassian does not offer support for third-party extensions or plugins. The information in the linked blog posts has not been tested or reviewed by Atlassian. We recommend that you test all solutions on a **test** server before trying them on your production site.

On this page:

- [omniauth_crowd](#)
- [Integrating Enterprise Tester with Crowd A .Net Integration](#)
- [Three's a Crowd - securing a Grails application with Acegi and Crowd](#)
- [SSO for RoundCube Webmail with Atlassian Crowd](#)
- [Nexus Crowd Plugin Introduction](#)
- [Integrating Crowd with Subversion](#)
- [Install Crowd Apache2 Module](#)
- [Video Configuring a Delegated Authentication Directory](#)
- [Video Configuring a Custom Directory Connector](#)
- [Bulk User Management with Crowd's Remote API](#)
- [Crowd Caching in 1.6](#)
- [Hammering Crowd](#)

Application Connectors

[omniauth_crowd](#)

- **By:** Rob Di Marco, on GitHub
- **About:** Integrating Crowd with OmniAuth to allow Rack-based applications (such as Rails, Sinatra) to use Crowd as an authentication source
- **Date and Crowd version:** March 2011, Crowd 2.2

[Integrating Enterprise Tester with Crowd A .Net Integration](#)

- **By:** Catch Limited, on the 'Atlassian Blog'
- **About:** Integrating Enterprise Tester, a .Net application, with Crowd. Enterprise Tester is a test management solution from Catch Limited.
- **Date and Crowd version:** 8 July 2010
- **Related documentation:** [Microsoft .NET Client](#)

[Three's a Crowd - securing a Grails application with Acegi and Crowd](#)

- **By:** Kate Ellingburg, on the 'Atlassian Blog'
- **About:** How to get Grails, Acegi and Crowd going together
- **Date and Crowd version:** 4 March 2008; Crowd 1.3
- **Related documentation:** [Integrating Crowd with Spring Security](#)

[SSO for RoundCube Webmail with Atlassian Crowd](#)

- **By:** Stefan Reuter, on the 'Stefan Reuter' blog
- **About:** Integrating a webmail system (RoundCube Webmail 0.2.2) with Crowd
- **Date and Crowd version:** 24 June 2009; Crowd 1.6
- **Related documentation:** [Creating a Crowd Client for your Custom Application](#)

[Nexus Crowd Plugin Introduction](#)

- **By:** Justin Edelson, on the 'Sonatype Blog'
- **About:** Using Crowd with Sonatype Nexus, via a new plugin for Nexus
- **Date and Crowd version:** 28 February 2009; Crowd 1.6
- **Related documentation:** [Creating a Crowd Client for your Custom Application](#)

[Integrating Crowd with Subversion](#)

- **By:** Trisummit Technologies
- **About:** Integrating Crowd with Subversion and Apache
- **Date and Crowd version:** 4 April 2010

[Install Crowd Apache2 Module](#)

- **By:** Scott Herdman, on blog 'swherdman.com'
- **About:** Integrating Crowd with Apache on Debian
- **Date and Crowd version:** 22 July 2009

Directory Connectors

[Video \[Configuring a Delegated Authentication Directory\]\(#\)](#)

- **By:** Brendan Patterson, on Screencast.com
- **About:** Configuring a Delegated Authentication directory and an LDAP directory connector
- **Date and Crowd version:** August 2010; Crowd 2.0
- **Related documentation:** [Configuring a Delegated Authentication Directory](#)

[Video \[Configuring a Custom Directory Connector\]\(#\)](#)

- **By:** Brendan Patterson, on Screencast.com
- **About:** Creating a custom directory connector
- **Date and Crowd version:** August 2010; Crowd 2.0
- **Related documentation:** [Configuring a Custom Directory Connector](#)

Remote API

[Bulk User Management with Crowd's Remote API](#)

- **By:** Andreas Knecht, on the 'Atlassian Blog'
- **About:** Adding multiple users to a group in Crowd, using Crowd's remote API and Ruby
- **Date and Crowd version:** 11 September 2008; Crowd 1.5
- **Related documentation:** [Managing Group Members](#)

Performance and Load Testing

[Crowd Caching in 1.6](#)

- **By:** Shihab Hamid, on the 'Atlassian Blog'
- **About:** Caching in Crowd 1.6
- **Date and Crowd version:** 4 January 2009; Crowd 1.6
- **Related documentation:**
 - [Overview of Caching](#)
 - [Configuring Caching for an LDAP Directory](#)

[Hammering Crowd](#)

- **By:** Shihab Hamid, on the 'Atlassian Blog'
- **About:** Tips for and finding from Crowd performance and load testing
- **Date and Crowd version:** 30 March 2008; Crowd 1.3
- **Related documentation:**
 - [Overview of Caching](#)
 - [Performance Profiling](#)
 - [Troubleshooting Crowd Performance](#)

Have you written a technical tip for Crowd?

Add a comment to this page, linking to your blog post or article. We will include it if the content fits the requirements of this page.

Feedback?

Your first port of call should be the author of the linked blog post. If you want to let us know how useful (or otherwise) a linked post is, please add a comment to this page.

Other Sources of Information

[Crowd documentation](#)
[Atlassian website](#)
[Answers from the community](#)
[Atlassian Blog](#)
[Crowd plugins](#)

Crowd Documentation in Other Languages

Below are some links to Crowd documentation written in other languages. In some cases, the documentation may be a translation of the English documentation. In other cases, the documentation is an alternative guide written from scratch in another language. This page presents an opportunity for customers and community authors to share documentation that they have written in other languages.

 **Please be aware that these are external guides.**

Most of the links point to external sites, and some of the information is relevant to a specific release of Crowd. Atlassian provides these links because the information is useful and relevant at the time it was written. Please check carefully whether the information is still relevant when you read it, and whether it is relevant to your version of Crowd. The information in the linked guides has not been tested or reviewed by Atlassian.

On this page:

- [No guides yet](#)

None

No guides yet

We do not yet have any guides to link here. Be the first to suggest one!

Adding Your Own Guide to this Page

Have you written a guide for Crowd in another language? Add a comment to this page, linking to your guide. We will include it if the content fits the requirements of this page.

Giving Feedback about One of the Guides

If you have feedback on one of the guides listed above, please give the feedback to the author of the linked guide.

If you want to let us know how useful (or otherwise) one of these guides is, please add a comment to this page.

Other Sources of Information

[Crowd documentation](#)

[Atlassian website](#)

[Atlassian blog](#)

[Crowd plugins and connectors](#)

Crowd Data Center

Crowd Data Center is a deployment option specifically designed to support the unique and complex requirements of enterprise organizations, where Crowd is central to the authentication and user management infrastructure. It's currently available for **Crowd 3.0, or later**. This page provides resources related to Data Center, as well as a high-level introduction to how it works.

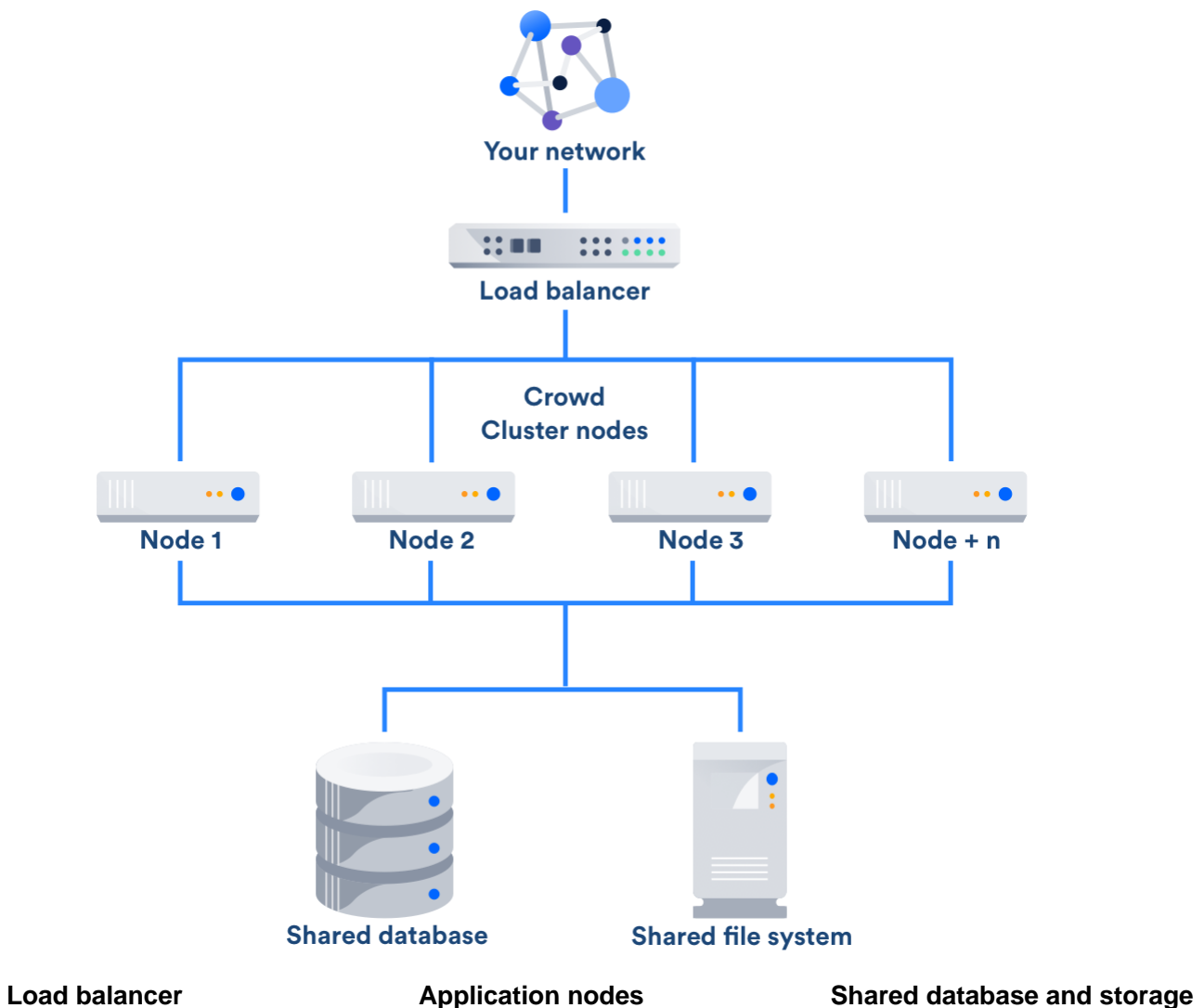
About Crowd Data Center

Crowd Data Center is a cluster of Crowd nodes designed to provide:

- **High availability and failover** If one cluster node goes down, then the remaining cluster nodes can continue servicing requests so that users have uninterrupted access to Crowd, and other systems connected to it.
- **Instant scalability** You can rapidly provision extra capacity with no downtime.

A look at the architecture

Crowd Data Center consists of a cluster of dedicated machines, connected like this:



The load balancer distributes requests from your users to the cluster nodes. If a cluster node goes down, the load balancer immediately detects the failure and automatically directs requests to the other nodes within seconds. You can use any load balancer that supports session affinity.

The cluster of Data Center nodes share the workload of incoming requests. Failure of a cluster node causes virtually no loss of availability for users, because requests are immediately directed to other nodes.

Data Center supports the same databases that are supported for Crowd Server. It also supports any shared file system, which stores: import/export files, plugins, shared caches, and any data directory which includes attachments, avatars and icons.

Get started with Crowd Data Center

Ready to get started?

[Learn more about Crowd Data Center](#) and [purchase a license](#) or [obtain an evaluation license](#).

Once you're there, you might want to look at these pages:

- [Crowd Data Center resources](#)
- [Installing Crowd Data Center](#)
- [FAQ](#)

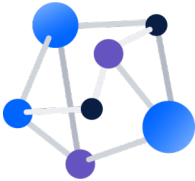
Crowd user management

Does this scenario ring a bell?




I have 2+ Atlassian products

Products are on-premise;
some are Data Center and
some - Server



I have a growing pool of users

I already have or I'm expecting
more than 500 unique users in
every app



I cannot afford downtime

I must keep my services
available and up and running
at all times

Meet Crowd



Crowd allows you to manage your users for all of your Atlassian Server and Data Center applications. Packed with **critical features** such as single sign-on, audit logs, and delegated administration, Crowd allows you to manage users and their authentication permissions from **multiple directories**. Crowd easily integrates with your existing environment be it big or small.

Benefits of Crowd

To stay on top of all user management trends, an increasing number of users opt for Crowd Data Center to have one central place to manage **all** users from **all** Atlassian products but **without the risk** of failure due to high availability.

In-product user management		Crowd Data Center
User management done per app or in Jira	➔	<ul style="list-style-type: none"> ✔ One place to manage all users
When users are managed in Jira, when Jira's down, users cannot log in to other apps		<ul style="list-style-type: none"> ✔ High availability - admins have uninterrupted access to user management and all users to apps
User management fixes and features need to be implemented per app		<ul style="list-style-type: none"> ✔ Fixes and features can easily be implemented
No prime features - high availability or delegated user		<ul style="list-style-type: none"> ✔ SSO, high availability, delegated user - all prime features included
In-product user management can have different versions		<ul style="list-style-type: none"> ✔ One app to upgrade and manage

OK, sounds good. What do I do next?

1.

Read about the hardware and software requirements in our documentation.

[Check installation requirements](#)

2.

Download a free 30-day trial of Crowd Data Center with all latest features.

[Evaluate Crowd Data Center for free](#)

3.

Generate a Data Center license. (Yes, even if you use Crowd Server.)

[Generate Data Center licence](#)

Talk to us

Thinking about implementing Crowd or looking for a place to leave feedback? Check out these resources for more information or to get in touch.

Ask us a question in our community.

[Crowd feedback community](#)

Review the product's FAQ.

[Check out Crowd FAQ](#)

Stay on top of the latest features.

[Check out release notes](#)