Space Details

Key:	CROWD
Name:	Crowd 1.4
Description:	Documentation for the latest version of Crowd single sign-on and identity management
Creator (Creation Date):	justen.stepka@atlassian.com (Sep 28, 2006)
Last Modifier (Mod. Date):	smaddox (May 07, 2008)

Available Pages

- Crowd Documentation
  - Crowd Administration Guide
    - 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
          - Microsoft Active Directory
            - Configuring an SSL Certificate for Microsoft Active Directory
          - SunONE
          - OpenLDAP
          - Apache Directory Server (ApacheDS)
          - Novell eDirectory
          - Posix Schema for LDAP
          - Generic LDAP Directories
        - Configuring a Custom Directory Connector
        - Configuring a Delegated Authentication Directory
      - Specifying Directory Permissions
      - Importing Users and Groups into a Directory
        - Importing Users from Atlassian Confluence
        - Importing Users from Atlassian JIRA
        - 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 Atlassian Bamboo
        - Importing Users from One Crowd Directory into Another
    - Managing Applications

        - Changing or Resetting your Password
            - Changing your Password
            - Resetting your Password
        - Updating your User Profile
        - Viewing your Group Membership
        - Viewing your Role Membership
        - Viewing your Applications
        - Crowd User's Glossary
            - Authorisation to Use Crowd (Glossary Entry)
            - Crowd Administrator (Glossary Entry)
            - Crowd-Connected Application (Glossary Entry)
            - Directory (Glossary Entry)
            - Self-Service Console (Glossary Entry)
            - Single Sign-On (Glossary Entry)
- Navigation
    - Blogs
- __newreleaseCrowd
- TreeNavigation

Crowd Documentation

This page last changed on May 08, 2008 by smaddox.

Crowd 1.4 Documentation	Resources
Installation Guide Upgrade Guide Release Notes Crowd Administration Guide Crowd User Guide CrowdID Administration Guide CrowdID User Guide Integration Guide Development Hub	If you have a question about using Crowd, please contact our support team. You may also want to check out the mailing lists and forums: • Crowd Announcements • Crowd General Forum • Crowd Developers Forum Other handy links: • Crowd Knowledge Base • Javadoc • JIRA Issue Tracker for Crowd

About
Crowd is a web-based single sign-on (SSO) tool that simplifies application provisioning and identity management. Crowd is the perfect solution to: • Give your users the convenience of single sign-on • Manage any number of users, logins and passwords • Centralise user management for applications such as JIRA, Confluence and Bamboo • Connect to multiple LDAP servers, such as Microsoft Active Directory • Integrate or import legacy user repositories • Control access to selected applications by user and group • Easily connect Crowd's application framework to new web applications

Download
You can download the Crowd documentation in PDF, HTML or XML formats.

Previous Versions
Crowd 1.3 Documentation Crowd 1.2 Documentation Crowd 1.1 Documentation Crowd 1.0 Documentation

Table of Contents

CrowdID Administration Guide

- 1. About CrowdID
- 2. Allowing users to access CrowdID
- 3. Specifying the sites to which users can login
- 4. Configuring CrowdID system settings

CrowdID User Guide

- 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

Crowd Installation & Upgrade Guide

- Crowd Release Notes
- Installing Crowd
- Upgrading Crowd

Crowd Knowledge Base

- Deployment FAQ
- Integration FAQ
- More General FAQ
- Troubleshooting

Crowd User Guide

- Introduction to Crowd
- Logging in to Crowd
- Logging out of Crowd
- Changing or Resetting your Password
- Updating your User Profile
- Viewing your Group Membership
- Viewing your Role Membership
- Viewing your Applications
- Crowd User's Glossary

Crowd Administration Guide

This page last changed on May 05, 2008 by smaddox.

Crowd is a web-based single sign-on (SSO) tool that simplifies application provisioning and identity management.

The Crowd Administration Guide is for people who have Crowd administration rights.

Table of Contents

Getting Started

This page last changed on May 04, 2008 by smaddox.

- Concepts
- Supported Applications and Directories
- About the Crowd Administration Console

Concepts

This page last changed on May 08, 2008 by smaddox.

Crowd is an application security framework that handles authentication and authorisation 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 centralised 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 authorised 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:

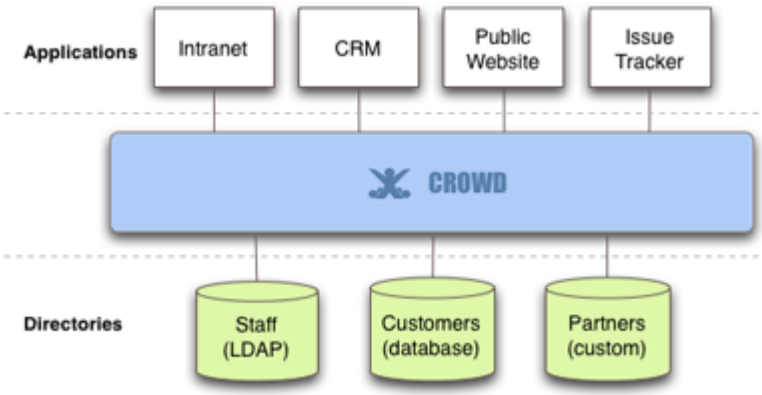
- Java, .NET and PHP applications.
- Popular directory servers such as Microsoft Active Directory, Sun ONE and OpenLDAP. Additionally, custom directory connectors may be developed using the Crowd integration API.

See the list of supported applications and directories.

Architectural Overview

Crowd is a middleware application that integrates web applications into a single security architecture that supports single sign-on and centralised identity management. Crowd works by dispatching authentication and authorisation 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/role management in LDAP.
- A Crowd internal directory for user/group/role 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 authorisation requests to the directory, for all applications that are mapped to that directory. Modification of directory entities (users, groups and roles) 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.

RELATED TOPICS

- Concepts
- Supported Applications and Directories
- About the Crowd Administration Console

Crowd Documentation

Supported Applications and Directories

This page last changed on May 08, 2008 by smaddox.

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 Bamboo
- Atlassian Fisheye
- Atlassian Crucible
- Apache
- Subversion
- Jive Forums
- Atlassian CrowdID
- Acegi
- NTLM for JIRA
- NTLM for Confluence

You can also add your own custom applications.

Directory Connectors

Connecting to LDAP directories:

- Apache Directory Server (ApacheDS)
- Generic LDAP Directory
- Microsoft Active Directory
- Novell eDirectory
- OpenLDAP
- Posix Schema for LDAP
- Sun Java System (SunONE) Directory Server

Internal Crowd 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

This page last changed on May 08, 2008 by smaddox.

The Crowd Administration Console presents the full range of Crowd administration functionality to authorised Crowd administrators.

Authorised 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.

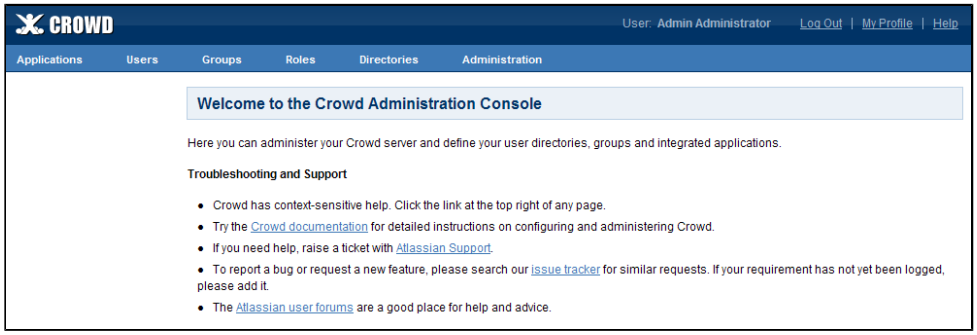
If you are a Crowd administrator, the Crowd Administration Console allows you to perform the following functions:

- Configure applications to access the Crowd framework.
- Create and manage users and adjust their group and role 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, roles and applications associated with your username. Refer to the Crowd 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 look 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.

RELATED TOPICS

- Concepts
- Supported Applications and Directories
- About the Crowd Administration Console

Crowd User Guide
Crowd Documentation

Managing Directories

This page last changed on May 05, 2008 by smaddox.

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/role management in LDAP.
- A Crowd internal directory for user/group/role 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 authorisation requests to the directory, for all applications that are mapped to that directory. Modification of directory entities (users, groups and roles) 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

This page last changed on May 07, 2008 by smaddox.

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/role management in LDAP.
- A Crowd internal directory for user/group/role 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 authorisation requests to the directory, for all applications that are mapped to that directory. Modification of directory entities (users, groups and roles) 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.

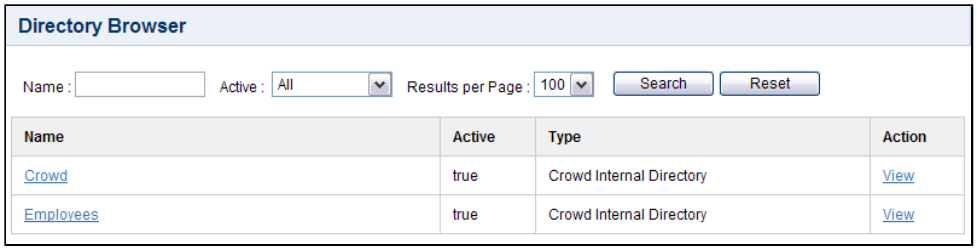
To use the Directory Browser,

1. Log in to the Crowd Administration Console.
2. Click the 'Directories' tab in the top navigation bar.
3. This will display the Directory Browser, showing 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.

   An 'Inactive' directory cannot be used by any applications, regardless of whether or not they are mapped to it.
4. To view or edit a directory's details, click the 'View' link.

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

Screenshot: 'Directory Browser'



RELATED TOPICS

- Using the Directory Browser
- Adding a Directory
  - Configuring an Internal Directory
  - Configuring an LDAP Directory Connector
    - Microsoft Active Directory
      - Configuring an SSL Certificate for Microsoft Active Directory
    - SunONE
    - OpenLDAP
    - Apache Directory Server (ApacheDS)
    - Novell eDirectory

Adding a Directory

This page last changed on May 05, 2008 by smaddox.

Directories contain authentication and authorisation 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, groups and roles in Crowd. You can use Crowd's flexible and simple group management when the LDAP groups do not suit your requirements.

  For example, 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.
- LDAP Directory Connector
  Crowd provides built-in connectors for the most popular LDAP directory servers (Microsoft Active Directory, SunONE/DSEE, OpenLDAP, Apache Directory). These LDAP connectors enable you to quickly integrate existing desktop logins with web applications.
- Custom Directory Connector
  Custom directory connectors allow developers to connect Crowd to custom user-stores, such as existing databases or legacy systems.

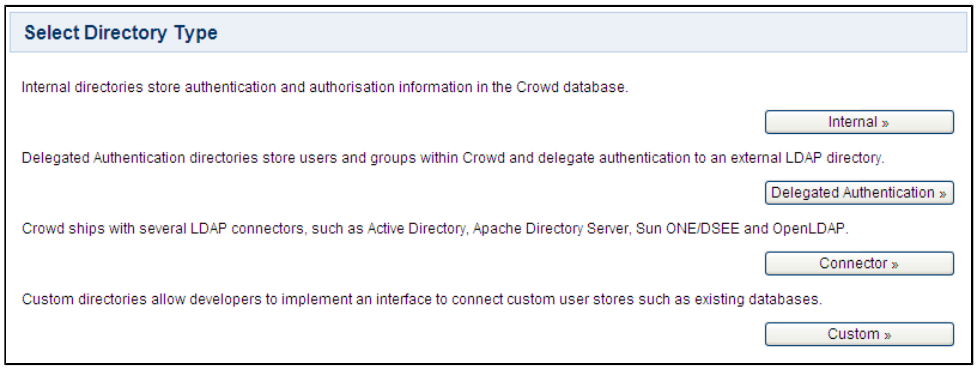
You can add as many directories of each type as you need.

To add a directory,

1. Log in to the Crowd Administration Console.
2. 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 (see below). Click the button corresponding to 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)
   - 'Custom' — see Configuring a Custom Directory Connector

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'



Related Topics

- Using the Directory Browser
- Adding a Directory
  - Configuring an Internal Directory
  - Configuring an LDAP Directory Connector
    - Microsoft Active Directory
      - Configuring an SSL Certificate for Microsoft Active Directory
    - SunONE
    - OpenLDAP
    - Apache Directory Server (ApacheDS)
    - Novell eDirectory
    - Posix Schema for LDAP
    - Generic LDAP Directories
  - Configuring a Custom Directory Connector
  - Configuring a Delegated Authentication Directory
- Specifying Directory Permissions
- Importing Users and Groups into a Directory
  - Importing Users from Atlassian Confluence
  - Importing Users from Atlassian JIRA
  - 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 Atlassian Bamboo
  - Importing Users from One Crowd Directory into Another

Crowd Documentation

Configuring an Internal Directory

This page last changed on May 05, 2008 by smaddox.

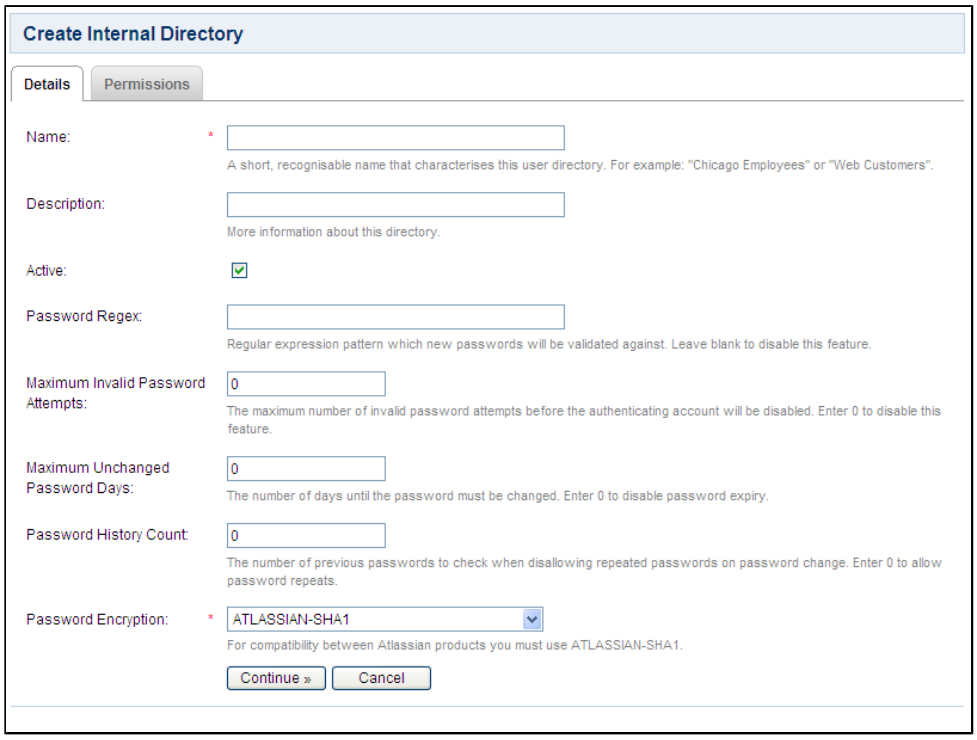
Internal directories use the Crowd database to store user, group and role information. Internal directories are stored in Crowd's database server.

To configure an Internal Directory,

1. Log in to the Crowd Administration Console.
2. Click the 'Directories' tab in the top navigation bar.
3. This will display the Directory Browser. Click 'Add Directory' in the left-hand menu.
4. Click the 'Internal' button.
5. Complete the fields as described in the table below.
6. Click the 'Continue' button to configure the directory's permissions.

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'



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	Only deselect this if you wish to prevent all users within the directory from accessing all mapped applications.
Password Regex	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: *\[A-Za-z0-9\]{8,}* 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.
Maimum 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 'ATLASSIAN-SHA1' in order to ensure password compatibility.

Next Step:

See Specifying Directory Permissions

RELATED TOPICS

- Using the Directory Browser
- Adding a Directory
  - Configuring an Internal Directory
  - Configuring an LDAP Directory Connector
    - Microsoft Active Directory
      - Configuring an SSL Certificate for Microsoft Active Directory
    - SunONE
    - OpenLDAP
    - Apache Directory Server (ApacheDS)
    - Novell eDirectory
    - Posix Schema for LDAP
    - Generic LDAP Directories
  - Configuring a Custom Directory Connector
  - Configuring a Delegated Authentication Directory
- Specifying Directory Permissions
- Importing Users and Groups into a Directory
  - Importing Users from Atlassian Confluence
  - Importing Users from Atlassian JIRA
  - 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 Atlassian Bamboo
  - Importing Users from One Crowd Directory into Another

Crowd Documentation

Configuring an LDAP Directory Connector

This page last changed on May 08, 2008 by smaddox.

Crowd provides built-in connectors for the most popular LDAP directory servers (Microsoft Active Directory, SunONE/DSEE, OpenLDAP, Apache Directory). These LDAP connectors enable you to quickly integrate existing desktop logins with web applications.

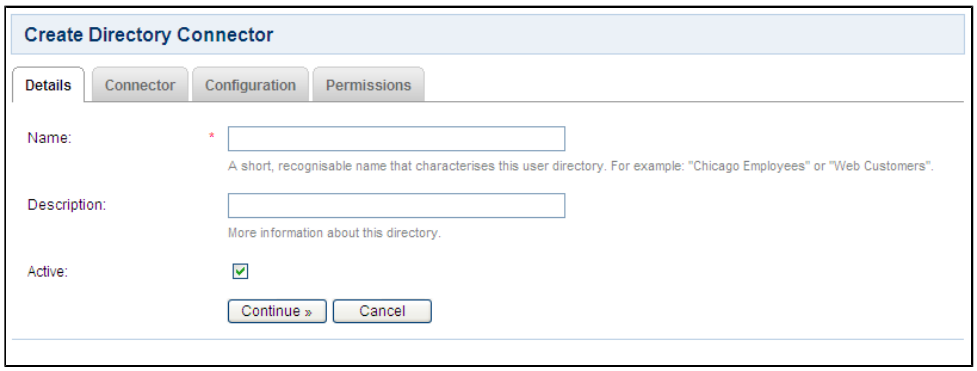
Summary of Configuration Steps

To configure an LDAP directory connector,

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. Click the 'Connector' button.
5. This will display the 'Details' tab (see Screenshot 1 below). Enter the 'Name' and 'Description' fields (see table below), then click the 'Continue' button.
6. This will display the 'Connector' tab (see Screenshot 2 below). Select the relevant connector type, and fill in the basic connection information for your directory server. For details, please see:
   - Microsoft Active Directory
   - SunONE
   - OpenLDAP
   - Apache Directory Server (ApacheDS)
   - Novell eDirectory
   - Posix Schema for LDAP
   - Generic LDAP Directories
7. Click the 'Test Connection' button to verify that Crowd can successfully connect to the directory.
8. Click the 'Continue' button.
9. This will display the 'Configuration' tab (see Screenshot 3 below). Fill in the configuration details for your groups, roles and users, as described in the tables below Screenshot 3. Also please see LDAP Object Structures (below).
10. Click the 'Test Search' button to verify that Crowd can successfully locate groups/roles/users within the directory.
11. Click the 'Continue' button to configure the directory's permissions.

Configuring Directory Details

Screenshot 1: Directory details

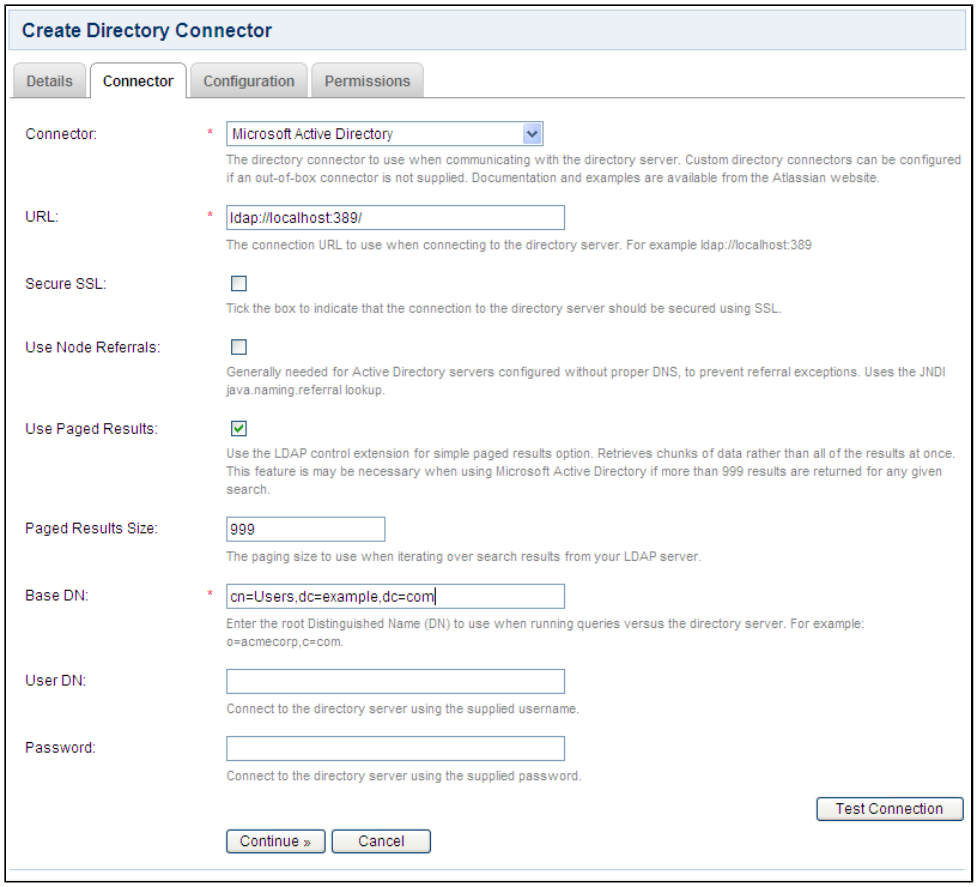


Attribute	Description
Name	The name used to identify the directory within Crowd. This is useful when there are multiple directories configured, e.g. 'Chicago Employees' or 'Web Customers'.
Description	Details about this specific directory.

Active	Only deselect this if you wish to prevent all users within the directory from accessing all mapped applications.

Configuring Connector Details

Screenshot 2: Connector



Attribute	Description
Connector	The directory connector to use when communicating with the directory server.
URL	The connection URL to use when connecting to the directory server, e.g.: `ldap://localhost:389`, or port 636 for SSL.
Secure SSL	Specifies whether the connection to the directory server is an SSL connection.
Use Node Referrals	Use the JNDI lookup java.naming.referral option. Generally needed for Active Directory servers configured without proper DNS, to prevent a 'javax.naming.PartialResultException: Unprocessed Continuation Reference(s)' error.
Use Paged Results	Use the LDAP control extension for simple paging of search results. Retrieves chunks of data rather than all of the search results at once. This feature may be necessary when using Microsoft Active Directory if more than 999 results are returned for any given search.
Paged Results Size	Enter the desired page size i.e. the maximum number of search results to be returned per page, when paged results are enabled. Defaults to 999 results. This option is available from Crowd 1.1.1.

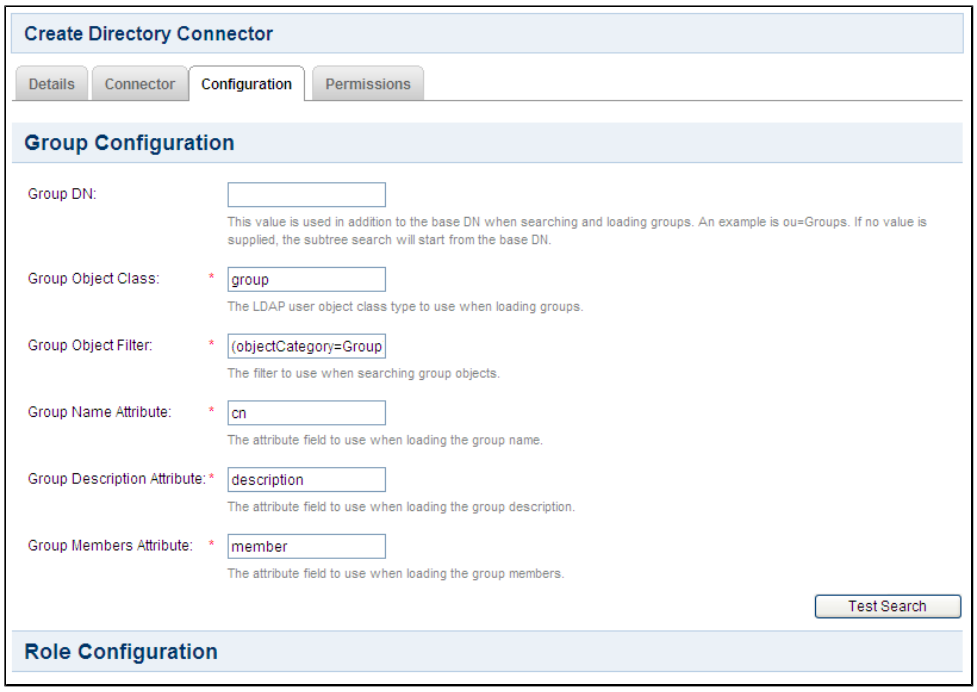
Base DN	Enter the root distinguished name to use when running queries versus the directory server, e.g.: `o=acmecorp,c=com`.
User DN	Distinguished name of the user that Crowd will use when connecting to the directory server.
Password	The password that Crowd will use when connecting to the directory server.

We have shown the settings for Active Directory. For details about the settings for your specific directory server, please see:

- Microsoft Active Directory
- SunONE
- OpenLDAP
- Apache Directory Server (ApacheDS)
- Novell eDirectory
- Posix Schema for LDAP
- Generic LDAP Directories

Configuring LDAP Object and Attribute Settings

Screenshot 3: Configuration



Once you have selected a connector you can modify various LDAP object and attribute settings of the specific LDAP server, as shown on the screenshot above. On first setup, Crowd will provide generic default settings, based on the connector selected.

When configuring your LDAP connector, if you are using non-standard object types, you will need to adjust the default filter and object type configurations. Default values are configured for the predefined LDAP servers. If your connector is added successfully, but you are unable to see any data when browsing your LDAP directory, it is likely that your object and filters are configured incorrectly.

Group Configuration

Attribute	Description
Group DN	This 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	This is the name of the class used for the LDAP group object. For example, groupOfUniqueNames.
Group Object Filter	The filter to use when searching group objects.
Group Name Attribute	The attribute field to use when loading the group's name.
Group Description Attribute	The attribute field to use when loading the group's description.
Group Members Attribute	The attribute field to use when loading the group's members.

Role Configuration

Attribute	Description
Role DN	This value is used in addition to the base DN when searching and loading roles, an example is ou=Roles. If no value is supplied, the subtree search will start from the base DN.
Role Object Class	This is the name of the class used for the LDAP role object.
Role Object Filter	The filter to use when searching role objects.
Role Name Attribute	The attribute field to use when loading the role's name.
Role Description Attribute	The attribute field to use when loading the role's description.
Role Members Attribute	The attribute field to use when loading the role's members.

User Configuration

Attribute	Description
User DN	This value is used in addition to the base DN when searching and loading users, an example is ou=Users. If no value is supplied, the subtree search will start from the base DN.
User Object Class	The LDAP user object class type to use when loading users.
User Object Filter	The filter to use when searching user objects.
User Name	The attribute field to use when loading the username.
User First Name	The attribute field to use when loading the user's first name.
User Last Name	The attribute field to use when loading the user's last name.
User Email	The attribute field to use when loading the user's email.
User Group	The attribute field to use when loading the user's groups.
User Password	The attribute field to use when manipulating a user's password.

LDAP Object Structures

The Crowd LDAP connectors assume that all container objects (groups and roles) have the full DN to the associated member. Currently, the membership attributes on a User object are not used by Crowd; however, in the future these associations may be used to assist with performance when looking up memberships.

To help you identify your LDAP structure, JXplorer is a free tool that allows you to browse your LDAP tree.

Supported Object Types

- groupOfUniqueNames
- inetorgperson
- posixGroup
- posixUser

Zimbra Mail Server
User objects have been tested and are known to work with the `zimbraAccount` LDAP object types.

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.

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 appropriate applications.

RELATED TOPICS

- Using the Directory Browser
- Adding a Directory
  - Configuring an Internal Directory
  - Configuring an LDAP Directory Connector
    - Microsoft Active Directory
      - Configuring an SSL Certificate for Microsoft Active Directory
    - SunONE
    - OpenLDAP
    - Apache Directory Server (ApacheDS)
    - Novell eDirectory
    - Posix Schema for LDAP
    - Generic LDAP Directories
  - Configuring a Custom Directory Connector
  - Configuring a Delegated Authentication Directory
- Specifying Directory Permissions
- Importing Users and Groups into a Directory
  - Importing Users from Atlassian Confluence
  - Importing Users from Atlassian JIRA
  - 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 Atlassian Bamboo
  - Importing Users from One Crowd Directory into Another

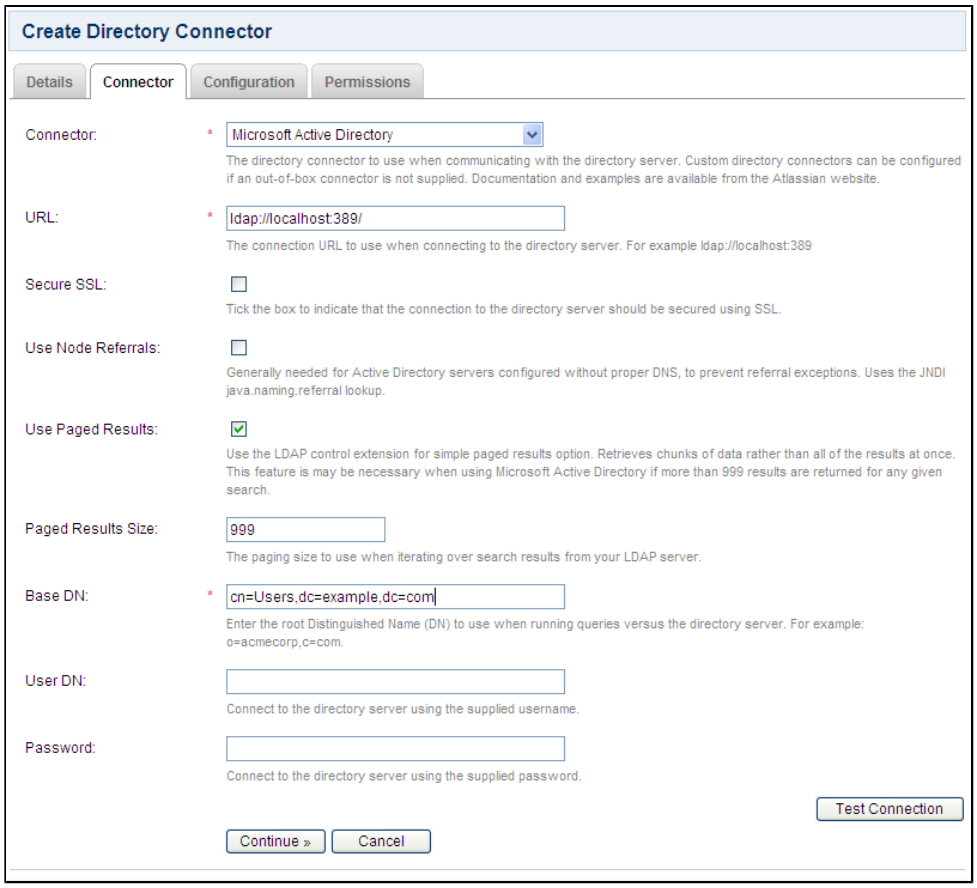
Crowd Documentation

Microsoft Active Directory

This page last changed on May 05, 2008 by smaddox.

This page provides configuration notes for Microsoft Active Directory, in relation to Configuring an LDAP Directory Connector.

Screenshot: 'Connector — Microsoft Active Directory'



Attribute	Description
Connector	The directory connector to use when communicating with the directory server.
URL	The connection URL to use when connecting to the directory server, e.g.: `ldap://localhost:389`, or port 636 for SSL.
Secure SSL	Specifies whether the connection to the directory server is an SSL connection.
Use Node Referrals	Use the JNDI lookup java.naming.referral option. Generally needed for Active Directory servers configured without proper DNS, to prevent a 'javax.naming.PartialResultException: Unprocessed Continuation Reference(s)' error.
Use Paged Results	Use the LDAP control extension for simple paging of search results. Retrieves chunks of data rather than all of the search results at once. This feature may be necessary when using Microsoft Active Directory if more than 999 results are returned for any given search.
Paged Results Size	Enter the desired page size i.e. the maximum number of search results to be returned per page, when paged results are enabled. Defaults to 999 results. This option is available from Crowd 1.1.1.

Base DN	Enter the root distinguished name to use when running queries versus the directory server, e.g.: `o=acmecorp,c=com`.
User DN	Distinguished name of the user that Crowd will use when connecting to the directory server.
Password	The password that Crowd will use when connecting to the directory server.

Configuration notes for Microsoft Active Directory

Active Directory Attribute Example	Value
Base DN	`cn=users,dc=ad,dc=acmecorp,dc=com`
User DN	`administrator@ad.acmecorp.com`

For Microsoft Active Directory, 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.

The URL for Microsoft Active Directory should be in the following format: `ldap://domainname`.

Configuring an SSL Certificate for Microsoft Active Directory

If you wish 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.

Next Step

Go back to Configuring an LDAP Directory Connector

RELATED TOPICS

- Using the Directory Browser
- Adding a Directory
  - Configuring an Internal Directory
  - Configuring an LDAP Directory Connector
    - Microsoft Active Directory
      - Configuring an SSL Certificate for Microsoft Active Directory
    - SunONE
    - OpenLDAP
    - Apache Directory Server (ApacheDS)
    - Novell eDirectory
    - Posix Schema for LDAP
    - Generic LDAP Directories
  - Configuring a Custom Directory Connector
  - Configuring a Delegated Authentication Directory
- Specifying Directory Permissions
- Importing Users and Groups into a Directory
  - Importing Users from Atlassian Confluence
  - Importing Users from Atlassian JIRA
  - 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 Atlassian Bamboo
  - Importing Users from One Crowd Directory into Another

Crowd Documentation

Configuring an SSL Certificate for Microsoft Active Directory

This page last changed on May 05, 2008 by smaddox.

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 certificated generated by your Active Directory server and then install the certificate into your JVM keystore.

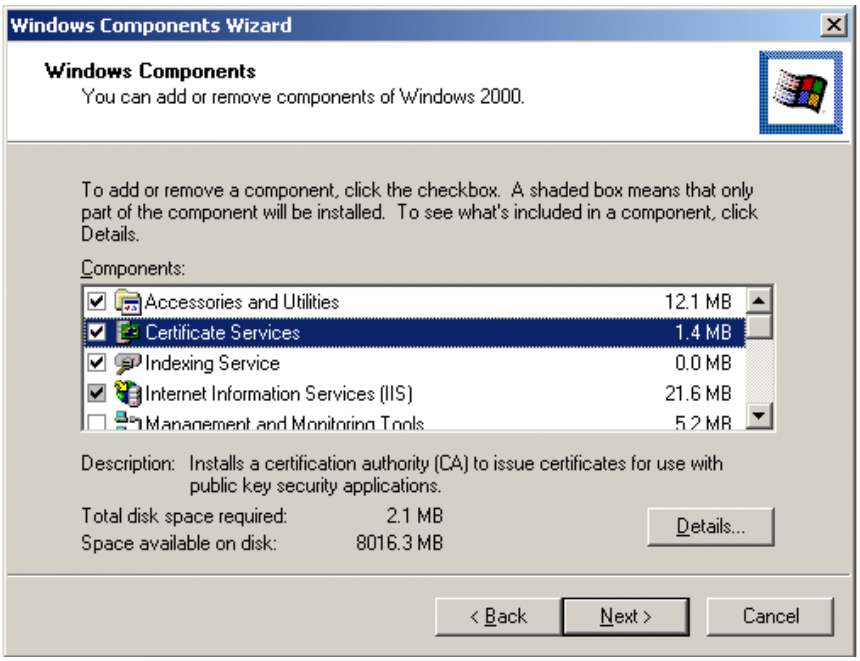
Prerequisites

Make sure that you have the following installed on your Windows server (domain controller):

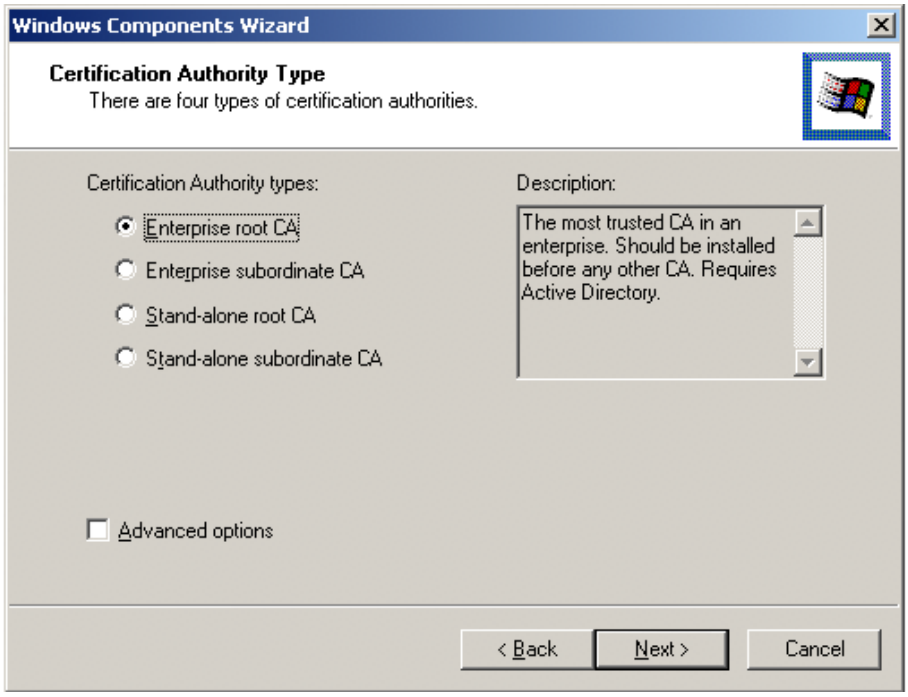
Required Component	Description
Windows 2000 Service Pack 2	Required if you are using Windows 2000
Internet Information Services (IIS)	This is required before you can install Windows Certificate Services.
Windows Certificate Services	This installs a certification authority (CA) which is used to issue certificates.
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 Microsoft Certificate Services

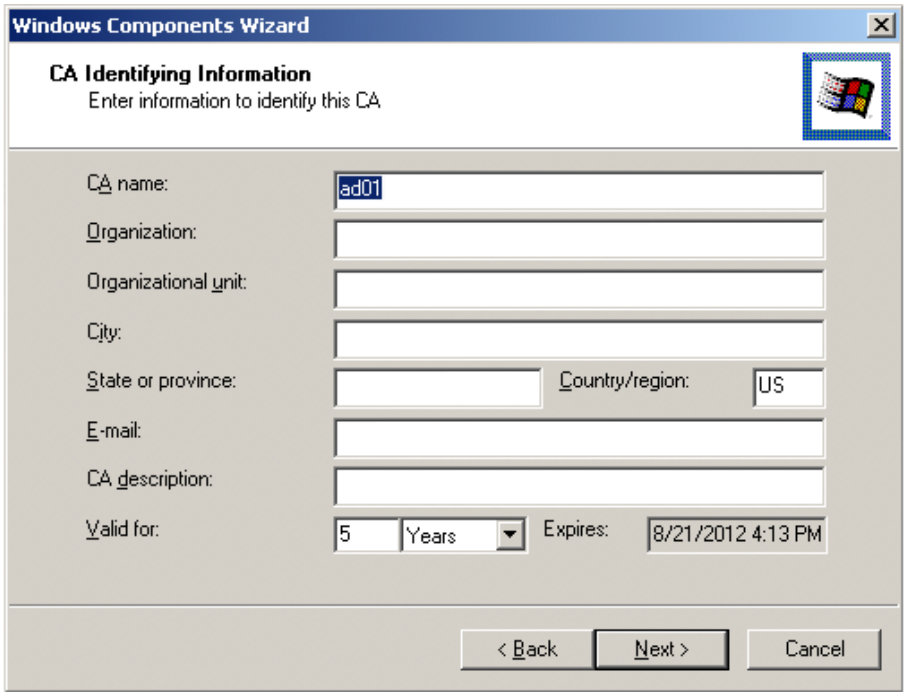
1. Using the Active Directory Control Panel – Add/Remove Programs administration tool:
   - Select 'Add/Remove Windows Components' to start the Windows Components Wizard.
   - Place check marks next to 'Certificate Services' and 'Internet Information Services (IIS)'.
   - Click 'Next>'.



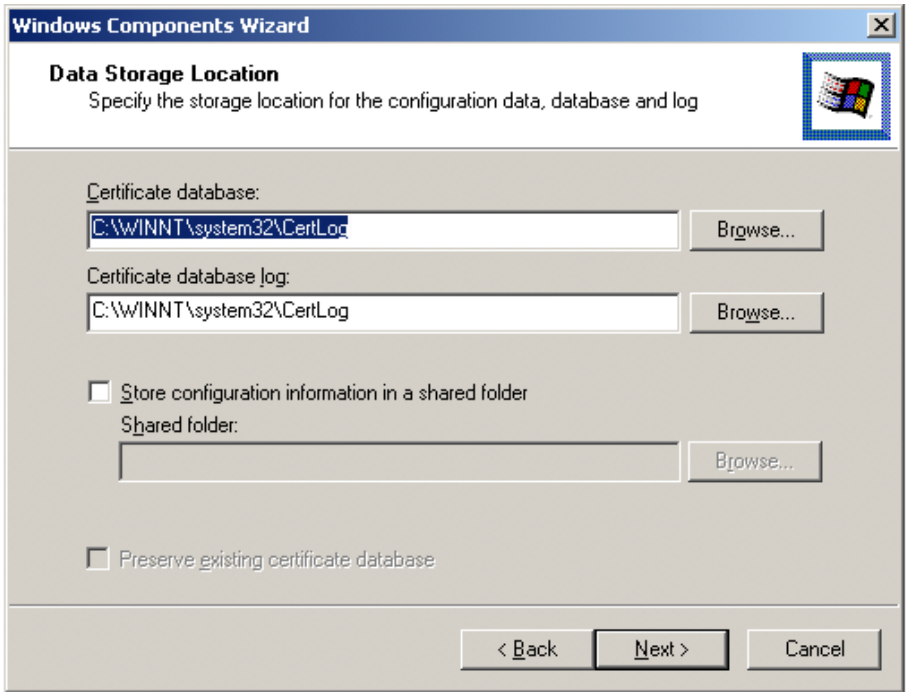
2. Select 'Enterprise root CA' Certificate Authority Type and click 'Next>'.



3. Enter a 'CA name' (server name) and click 'Next>'. On Windows Server 2003, this is the 'Common name for this CA'.

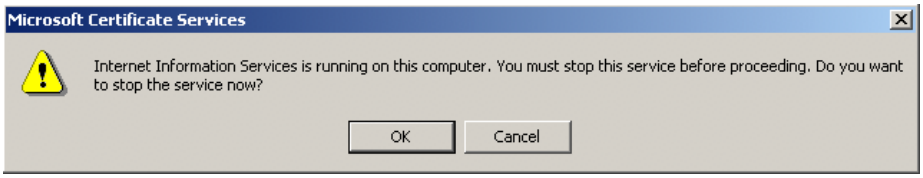


4. Leave the 'Data Storage Locations' as default and click 'Next>'.



5. The software installation process is complete. Click 'Finish'.

6. Click 'OK' to restart IIS.



7. You will now need to restart your Microsoft Active Directory Server.

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 Crowd 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, e.g. `c:\crowd-ad2000.ad01.crowd.atlassian.com_ad01.crt`.

You can also export the certificate by executing this command on the Active Directory server:

```
certutil -ca.cert crowd-client.crt
```

Step 3. Import the Server Certificate

Now you need to import the Active Directory certificate to the list of accepted certificates in your JDK runtime environment.

- Assuming your JDK is installed here `C:\Program Files\Java\jdk1.5.0_12`, you will need to run the following command:
  C:\Program Files\Java\jdk1.5.0_12\keytool \-import \-alias crowd_crt \-file crowd-client.crt \-keystore "C:\Program Files\Java\jdk1.5.0_12\jre\lib\security\cacerts"
- The `keytool` import will prompt you for a password during import. The default keystore password is `changeit`.
- When prompted `Trust this certificate? [no]:` enter 'yes' to confirm the Active Directory Server 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
```

You may now use the `Secure SSL` option when connecting to an Active Directory server with Crowd's built in connectors.

Related Topics

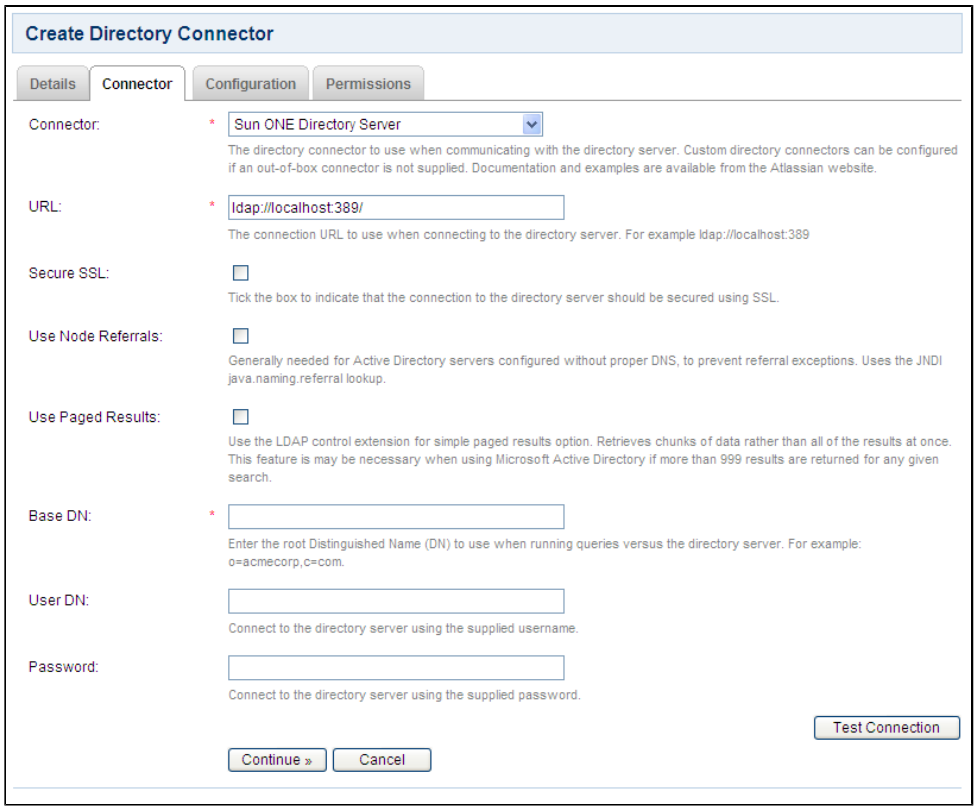
Microsoft Active Directory
Configuring Crowd to Work with SSL

SunONE

This page last changed on May 05, 2008 by smaddox.

This page provides configuration notes for SunONE Directory Server, in relation to Configuring an LDAP Directory Connector.

Screenshot: 'Connector — SunONE Directory Server'



Attribute	Description
Connector	The directory connector to use when communicating with the directory server.
URL	The connection URL to use when connecting to the directory server, e.g.: `ldap://localhost:389`, or `port 639` for SSL.
Secure SSL	Specifies if the connection to the directory server is a SSL connection.
Use Node Referrals	Use the JNDI lookup java.naming.referral option. Generally needed for Active Directory servers configured without proper DNS, to prevent a 'javax.naming.PartialResultException: Unprocessed Continuation Reference(s)' error.
Use Paged Results	Use the LDAP control extension for simple paged results option. Retrieves chunks of data rather than all of the results at once. This feature may be necessary when using Microsoft Active Directory if more than 999 results are returned for any given search.
Base DN	Enter the root distinguished name to use when running queries versus the directory server, e.g.: `o=acmecorp,c=com`.
User DN	The username that Crowd will use when connecting to the directory server.
Password	The password that Crowd will use when connecting to the directory server.

Configuration details for SunONE

SunONE Example	Value
Base DN	dc=acmecorp,dc=com
User DN	cn=Directory Manager

Next Step

Go back to Configuring an LDAP Directory Connector

RELATED TOPICS

- Using the Directory Browser
- Adding a Directory
  - Configuring an Internal Directory
  - Configuring an LDAP Directory Connector
    - Microsoft Active Directory
      - Configuring an SSL Certificate for Microsoft Active Directory
    - SunONE
    - OpenLDAP
    - Apache Directory Server (ApacheDS)
    - Novell eDirectory
    - Posix Schema for LDAP
    - Generic LDAP Directories
  - Configuring a Custom Directory Connector
  - Configuring a Delegated Authentication Directory
- Specifying Directory Permissions
- Importing Users and Groups into a Directory
  - Importing Users from Atlassian Confluence
  - Importing Users from Atlassian JIRA
  - 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 Atlassian Bamboo
  - Importing Users from One Crowd Directory into Another

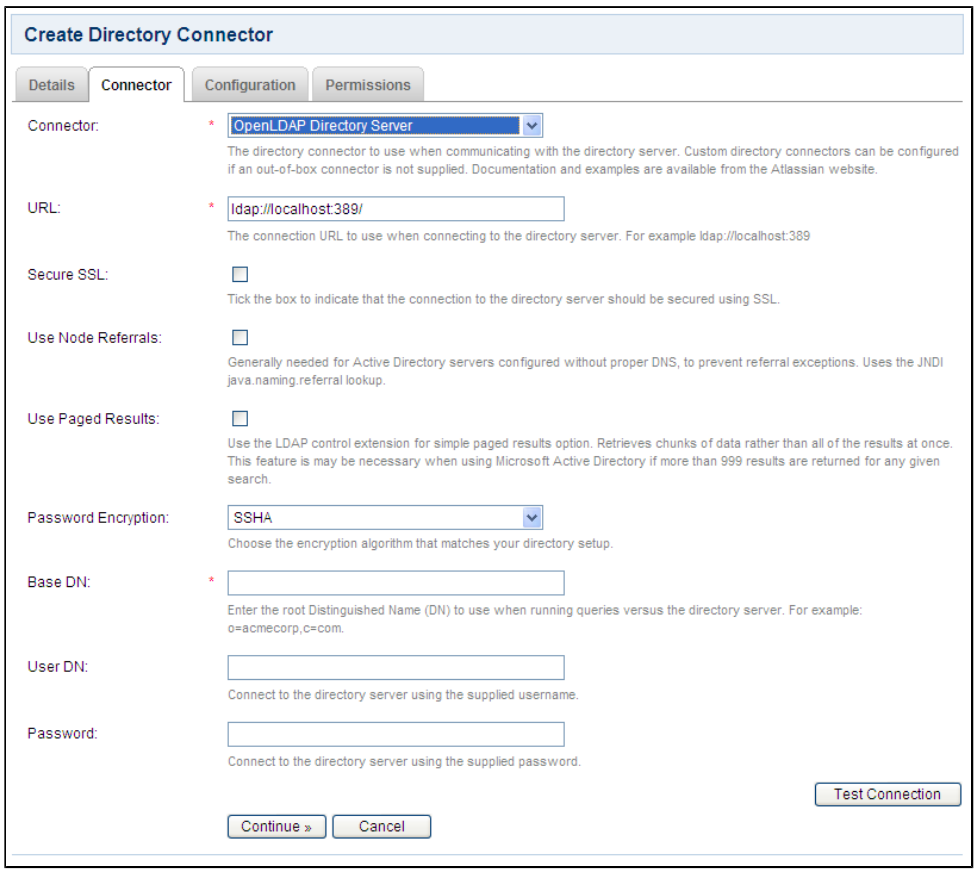
Crowd Documentation

OpenLDAP

This page last changed on May 05, 2008 by smaddox.

This page provides configuration notes for OpenLDAP, in relation to Configuring an LDAP Directory Connector.

Screenshot: 'Connector — OpenLDAP'



Attribute	Description
Connector	The directory connector to use when communicating with the directory server.
URL	The connection URL to use when connecting to the directory server, e.g.: `ldap://localhost:389`, or `port 639` for SSL.
Secure SSL	Specifies if the connection to the directory server is a SSL connection.
Use Node Referrals	Use the JNDI lookup java.naming.referral option. Generally needed for Active Directory servers configured without proper DNS, to prevent a 'javax.naming.PartialResultException: Unprocessed Continuation Reference(s)' error.
Use Paged Results	Use the LDAP control extension for simple paged results option. Retrieves chunks of data rather than all of the results at once. This feature may be necessary when using Microsoft Active Directory if more than 999 results are returned for any given search.
Password Encryption	Select the type of encryption that the directory uses.
Base DN	Enter the root distinguished name to use when running queries versus the directory server, e.g.: `o=acmecorp,c=com`.
User DN	

	Distinguished name of the user that Crowd will use when connecting to the directory server.
Password	The password that Crowd will use when connecting to the directory server.

Configuration Details for OpenLDAP

OpenLDAP Directory Example	Value
Base DN	dc=example,dc=com
User DN	cn=Manager,dc=example,dc=com

Next Step

Go back to Configuring an LDAP Directory Connector.

RELATED TOPICS

- Using the Directory Browser
- Adding a Directory
  - Configuring an Internal Directory
  - Configuring an LDAP Directory Connector
    - Microsoft Active Directory
      - Configuring an SSL Certificate for Microsoft Active Directory
    - SunONE
    - OpenLDAP
    - Apache Directory Server (ApacheDS)
    - Novell eDirectory
    - Posix Schema for LDAP
    - Generic LDAP Directories
  - Configuring a Custom Directory Connector
  - Configuring a Delegated Authentication Directory
- Specifying Directory Permissions
- Importing Users and Groups into a Directory
  - Importing Users from Atlassian Confluence
  - Importing Users from Atlassian JIRA
  - 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 Atlassian Bamboo
  - Importing Users from One Crowd Directory into Another

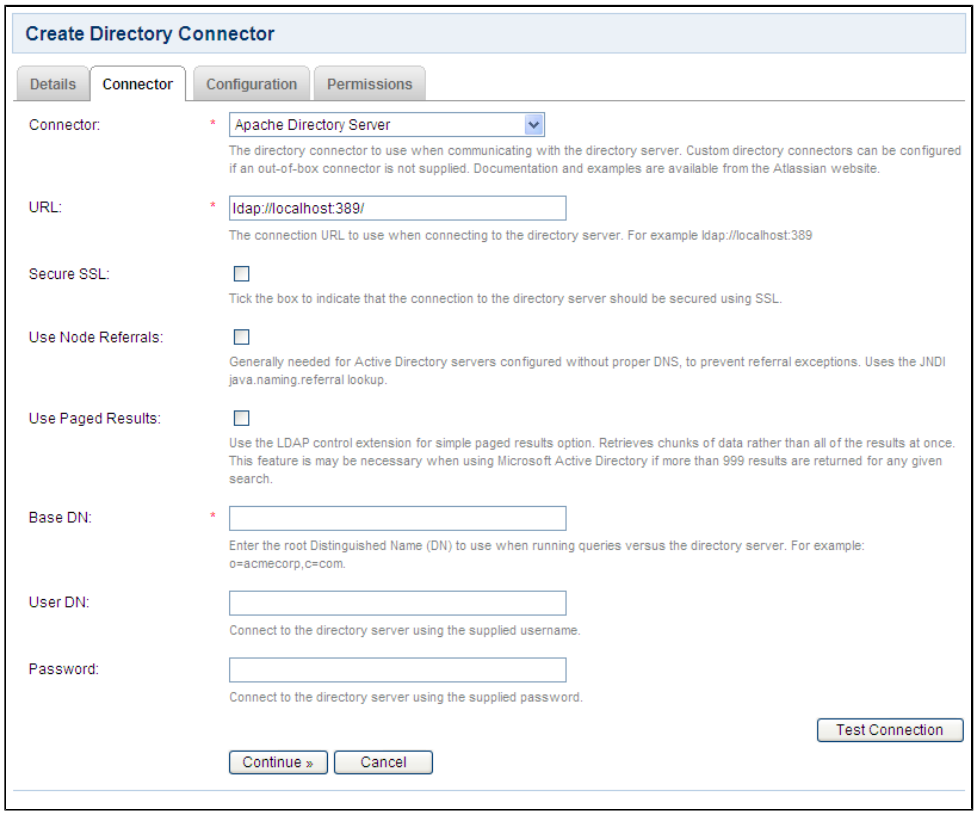
Crowd Documentation

Apache Directory Server (ApacheDS)

This page last changed on May 05, 2008 by smaddox.

This page provides configuration notes for Apache Directory Server, in relation to Configuring an LDAP Directory Connector.

Screenshot: 'Connector — Apache '



Attribute	Description
Connector	The directory connector to use when communicating with the directory server.
URL	The connection URL to use when connecting to the directory server, e.g.: `ldap://localhost:389`, or `port 639` for SSL.
Secure SSL	Specifies if the connection to the directory server is a SSL connection.
Use Node Referrals	Use the JNDI lookup java.naming.referral option. Generally needed for Active Directory servers configured without proper DNS, to prevent a 'javax.naming.PartialResultException: Unprocessed Continuation Reference(s)' error.
Use Paged Results	Use the LDAP control extension for simple paged results option. Retrieves chunks of data rather than all of the results at once. This feature may be necessary when using Microsoft Active Directory if more than 999 results are returned for any given search.
Base DN	Enter the root distinguished name to use when running queries versus the directory server, e.g.: `o=acmecorp,c=com`.
User DN	The username that Crowd will use when connecting to the directory server.
Password	The password that Crowd will use when connecting to the directory server.

Configuration details for ApacheDS

OpenLDAP Directory Example	Value
Base DN	dc=example,dc=com

Next Step

Go back to Configuring an LDAP Directory Connector

RELATED TOPICS

- Using the Directory Browser
- Adding a Directory
  - Configuring an Internal Directory
  - Configuring an LDAP Directory Connector
    - Microsoft Active Directory
      - Configuring an SSL Certificate for Microsoft Active Directory
    - SunONE
    - OpenLDAP
    - Apache Directory Server (ApacheDS)
    - Novell eDirectory
    - Posix Schema for LDAP
    - Generic LDAP Directories
  - Configuring a Custom Directory Connector
  - Configuring a Delegated Authentication Directory
- Specifying Directory Permissions
- Importing Users and Groups into a Directory
  - Importing Users from Atlassian Confluence
  - Importing Users from Atlassian JIRA
  - 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 Atlassian Bamboo
  - Importing Users from One Crowd Directory into Another

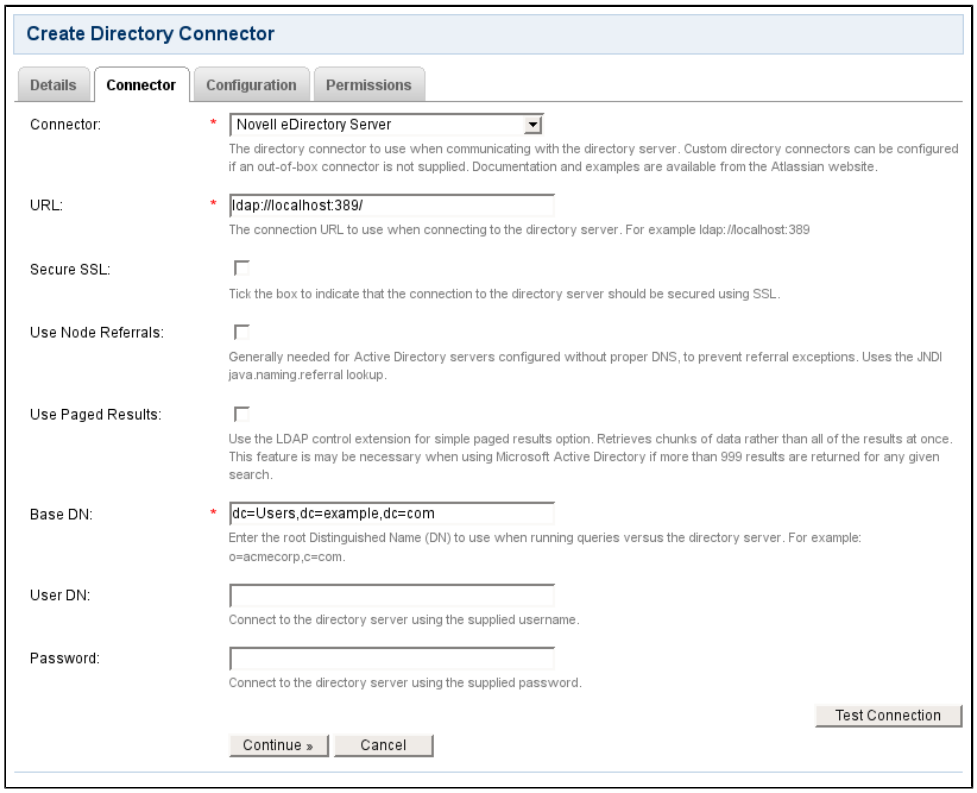
Crowd Documentation

Novell eDirectory

This page last changed on May 08, 2008 by smaddox.

This page provides configuration notes for Novell eDirectory, in relation to Configuring an LDAP Directory Connector.

Screenshot: 'Connector — Novell eDirectory Server'



Attribute	Description
Connector	The directory connector to use when communicating with the directory server.
URL	The connection URL to use when connecting to the directory server, e.g.: `ldap://localhost:389`, or port 636 for SSL.
Secure SSL	Specifies whether the connection to the directory server is an SSL connection.
Use Node Referrals	Use the JNDI lookup java.naming.referral option. Generally needed for Active Directory servers configured without proper DNS, to prevent a 'javax.naming.PartialResultException: Unprocessed Continuation Reference(s)' error.
Use Paged Results	Use the LDAP control extension for simple paging of search results. Retrieves chunks of data rather than all of the search results at once. This feature may be necessary when using Microsoft Active Directory if more than 999 results are returned for any given search.
Base DN	Enter the root distinguished name to use when running queries versus the directory server, e.g.: `o=acmecorp,c=com`.
User DN	Distinguished name of the user that Crowd will use when connecting to the directory server.
Password	The password that Crowd will use when connecting to the directory server.

Next Step

Go back to Configuring an LDAP Directory Connector

RELATED TOPICS

- Using the Directory Browser
- Adding a Directory
    - Configuring an Internal Directory
    - Configuring an LDAP Directory Connector
        - Microsoft Active Directory
            - Configuring an SSL Certificate for Microsoft Active Directory
        - SunONE
        - OpenLDAP
        - Apache Directory Server (ApacheDS)
        - Novell eDirectory
        - Posix Schema for LDAP
        - Generic LDAP Directories
    - Configuring a Custom Directory Connector
    - Configuring a Delegated Authentication Directory
- Specifying Directory Permissions
- Importing Users and Groups into a Directory
    - Importing Users from Atlassian Confluence
    - Importing Users from Atlassian JIRA
    - 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 Atlassian Bamboo
    - Importing Users from One Crowd Directory into Another

Crowd Documentation

Posix Schema for LDAP

This page last changed on May 08, 2008 by smaddox.

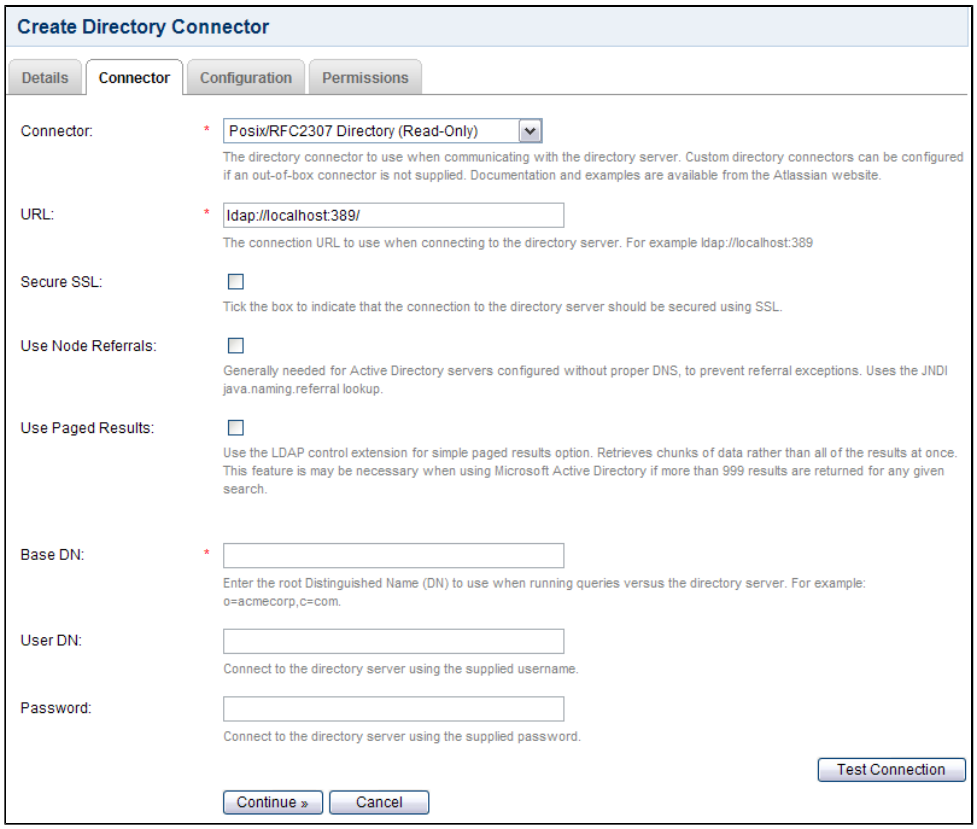
This page provides configuration notes for an LDAP directory using the Posix/NIS schema RFC 2307, in relation to Configuring an LDAP Directory Connector.

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's Posix support is read-only and OpenLDAP only

Currently, Crowd supports read-only access to the directory based on the Posix schema. You cannot add or update user details. We support only OpenLDAP with Posix, though in future we may support other directories based on this schema too.

Screenshot: 'Connector — LDAP using Posix Schema'



Attribute	Description
Connector	The directory connector to use when communicating with the directory server.
URL	The connection URL to use when connecting to the directory server, e.g.: `ldap://localhost:389`, or `port 639` for SSL.
Secure SSL	Specifies if the connection to the directory server is a SSL connection.
Use Node Referrals	Use the JNDI lookup java.naming.referral option. Generally needed for Active Directory servers configured without proper DNS, to prevent a 'javax.naming.PartialResultException: Unprocessed Continuation Reference(s)' error.
Use Paged Results	Use the LDAP control extension for simple paged results option. Retrieves chunks of data rather than all of the results at once. This feature may be

	necessary when using Microsoft Active Directory if more than 999 results are returned for any given search.
Base DN	Enter the root distinguished name to use when running queries versus the directory server, e.g.: `o=acmecorp,c=com`.
User DN	Distinguished name of the user that Crowd will use when connecting to the directory server.
Password	The password that Crowd will use when connecting to the directory server.

Group Relationships

Crowd will check both the `gidNumber` and the `memberUid` attributes to determine if a user is a member of a group. In Crowd 1.4, 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 we do not have support for nested groups in the Posix schema.

Next Step

Go back to Configuring an LDAP Directory Connector.

RELATED TOPICS

- Using the Directory Browser
- Adding a Directory
  - Configuring an Internal Directory
  - Configuring an LDAP Directory Connector
    - Microsoft Active Directory
      - Configuring an SSL Certificate for Microsoft Active Directory
    - SunONE
    - OpenLDAP
    - Apache Directory Server (ApacheDS)
    - Novell eDirectory
    - Posix Schema for LDAP
    - Generic LDAP Directories
  - Configuring a Custom Directory Connector
  - Configuring a Delegated Authentication Directory
- Specifying Directory Permissions
- Importing Users and Groups into a Directory
  - Importing Users from Atlassian Confluence
  - Importing Users from Atlassian JIRA
  - 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 Atlassian Bamboo
  - Importing Users from One Crowd Directory into Another

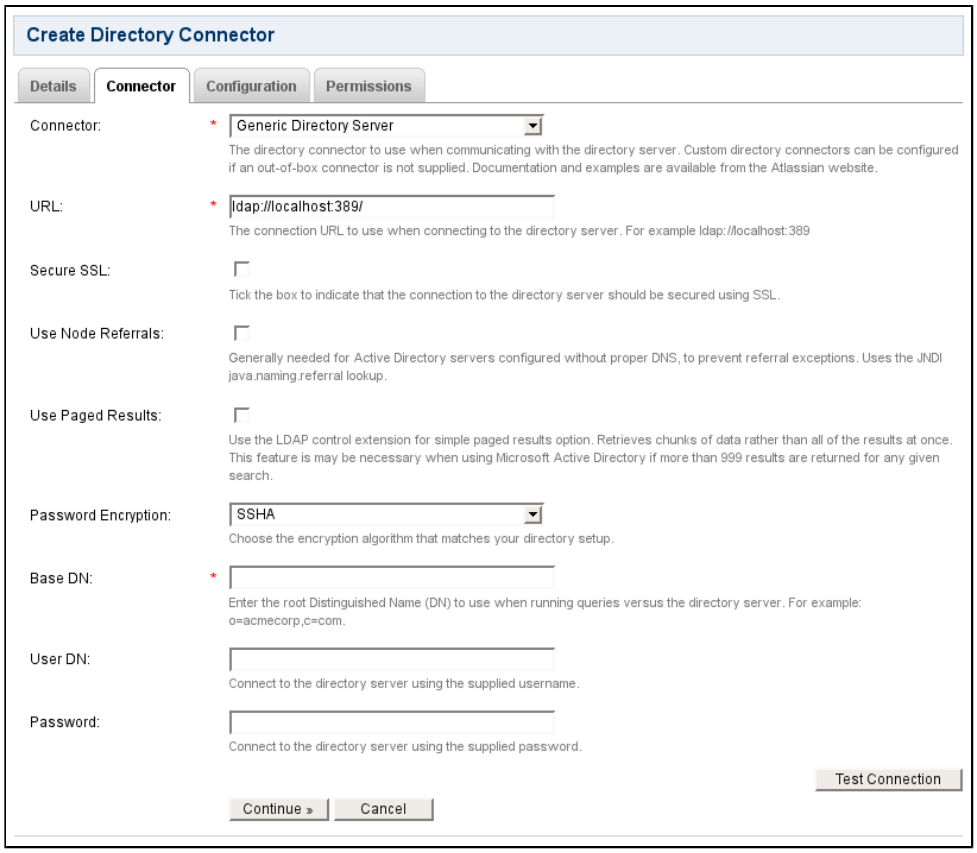
Crowd Documentation

Generic LDAP Directories

This page last changed on May 08, 2008 by smaddox.

This page provides configuration notes for generic LDAP directories, in relation to Configuring an LDAP Directory Connector.

Screenshot: 'Connector — Generic Directory Server'



Attribute	Description
Connector	The directory connector to use when communicating with the directory server.
URL	The connection URL to use when connecting to the directory server, e.g.: `ldap://localhost:389`, or `port 639` for SSL.
Secure SSL	Specifies if the connection to the directory server is a SSL connection.
Use Node Referrals	Use the JNDI lookup java.naming.referral option. Generally needed for Active Directory servers configured without proper DNS, to prevent a 'javax.naming.PartialResultException: Unprocessed Continuation Reference(s)' error.
Use Paged Results	Use the LDAP control extension for simple paged results option. Retrieves chunks of data rather than all of the results at once. This feature may be necessary when using Microsoft Active Directory if more than 999 results are returned for any given search.
Password Encryption	Select the type of encryption that the directory uses.
Base DN	Enter the root distinguished name to use when running queries versus the directory server, e.g.: `o=acmecorp,c=com`.
User DN	The username that Crowd will use when connecting to the directory server.

Password	The password that Crowd will use when connecting to the directory server.

Next Step

Go back to Configuring an LDAP Directory Connector

RELATED TOPICS

- Using the Directory Browser
- Adding a Directory
  - Configuring an Internal Directory
  - Configuring an LDAP Directory Connector
    - Microsoft Active Directory
      - Configuring an SSL Certificate for Microsoft Active Directory
    - SunONE
    - OpenLDAP
    - Apache Directory Server (ApacheDS)
    - Novell eDirectory
    - Posix Schema for LDAP
    - Generic LDAP Directories
  - Configuring a Custom Directory Connector
  - Configuring a Delegated Authentication Directory
- Specifying Directory Permissions
- Importing Users and Groups into a Directory
  - Importing Users from Atlassian Confluence
  - Importing Users from Atlassian JIRA
  - 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 Atlassian Bamboo
  - Importing Users from One Crowd Directory into Another

Crowd Documentation

Configuring a Custom Directory Connector

This page last changed on May 05, 2008 by smaddox.

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.

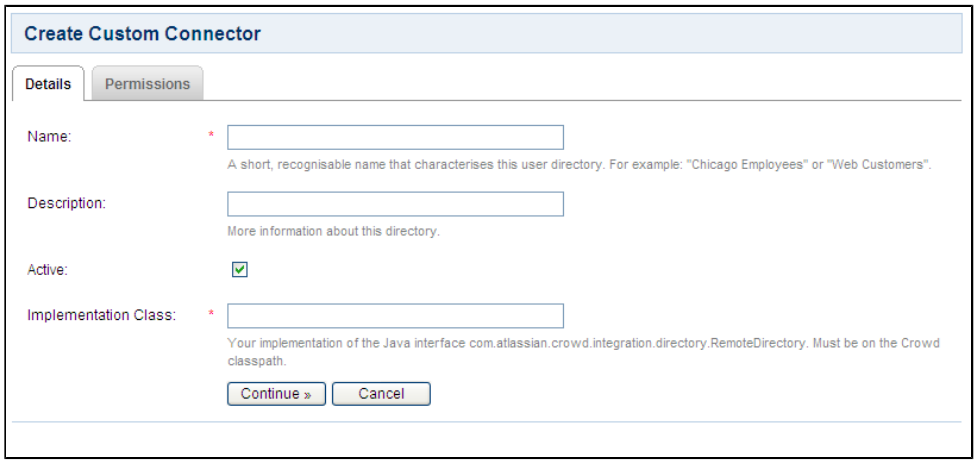
Once you have added your JAR file to the Crowd `WEB-INF/lib` folder, you are ready to configure a Custom Directory Connector, as described below.

To configure a Custom Directory Connector,

1. Log in to the Crowd Administration Console.
2. Click the 'Directories' link in the top navigation bar.
3. This will display the Directory Browser. Click the 'Add Directory' link.
4. Click the 'Custom' button.
5. Complete the fields as described in the table below.
6. Click the 'Continue' button to configure the directory's permissions.

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'



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.
Implementation Class	Implementation of `com.atlassian.crowd.integration.directory.RemoteDirecto` Java interface. Must be in the Crowd CLASSPATH.

Next Step:

See Specifying Directory Permissions

Related Topics

- Using the Directory Browser
- Adding a Directory
  - Configuring an Internal Directory
  - Configuring an LDAP Directory Connector
    - Microsoft Active Directory
      - Configuring an SSL Certificate for Microsoft Active Directory
    - SunONE
    - OpenLDAP
    - Apache Directory Server (ApacheDS)
    - Novell eDirectory
    - Posix Schema for LDAP
    - Generic LDAP Directories
  - Configuring a Custom Directory Connector
  - Configuring a Delegated Authentication Directory
- Specifying Directory Permissions
- Importing Users and Groups into a Directory
  - Importing Users from Atlassian Confluence
  - Importing Users from Atlassian JIRA
  - 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 Atlassian Bamboo
  - Importing Users from One Crowd Directory into Another

Crowd Documentation

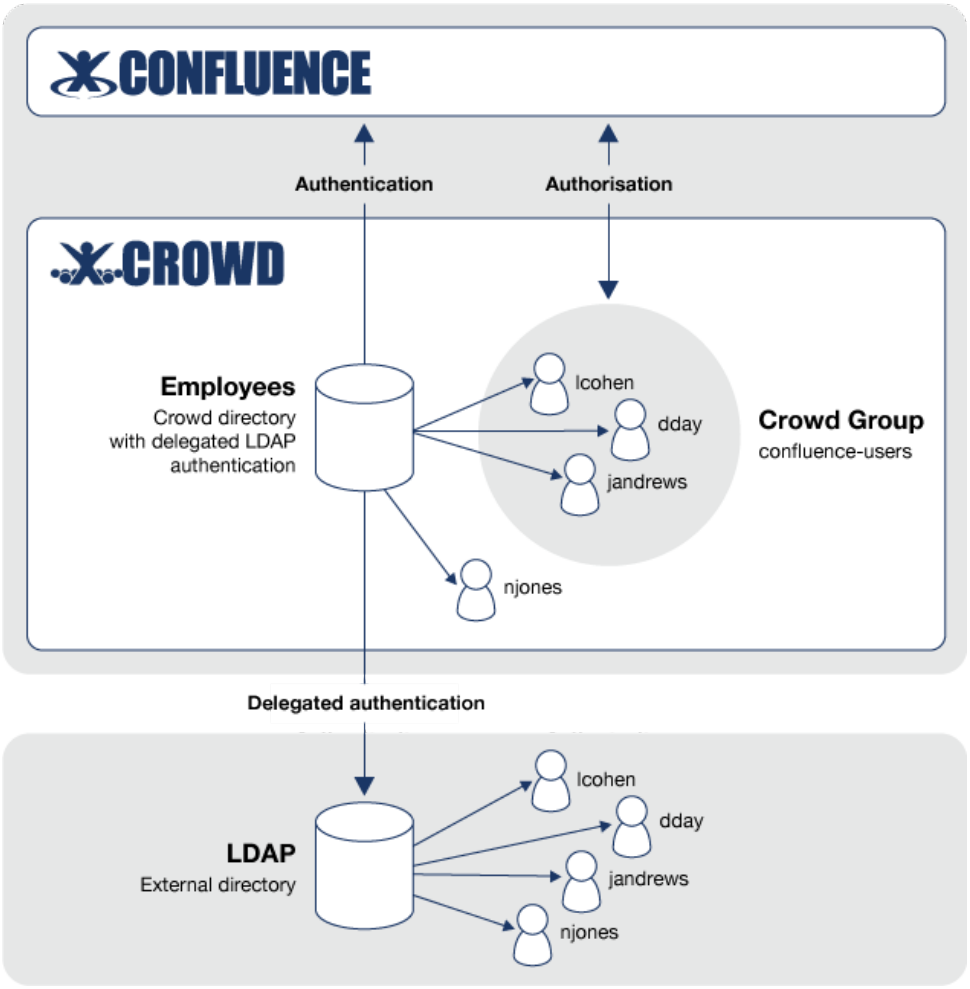
Configuring a Delegated Authentication Directory

This page last changed on May 05, 2008 by smaddox.

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, groups and roles in Crowd. You can use Crowd's flexible and simple group management when the LDAP groups do not suit your requirements.

For example, 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).



Summary of Configuration Steps

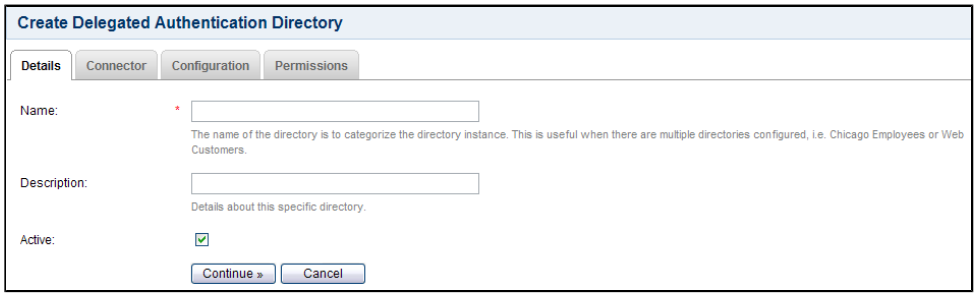
To configure a Delegated Authentication directory,

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. Click the 'Delegated Authentication' button.
5. This will display the 'Details' tab (see Screenshot 1 below). Enter the 'Name' and 'Description' fields, then click the 'Continue' button.
6. This will display the 'Connector' tab (see Screenshot 2 below). Select the relevant connector type, and fill in the basic connection information for your directory server. For details, please see: Unable to render {children} Page not found: 2.2.2 Configuring an LDAP Directory Connector
7. Click the 'Test Connection' button to verify that Crowd can successfully connect to the directory.
8. Click the 'Continue' button.
9. This will display the 'Configuration' tab (see Screenshot 3 below). Fill in the configuration details for your users.
10. Click the 'Continue' button to configure the directory's permissions.

Configuring Directory Details

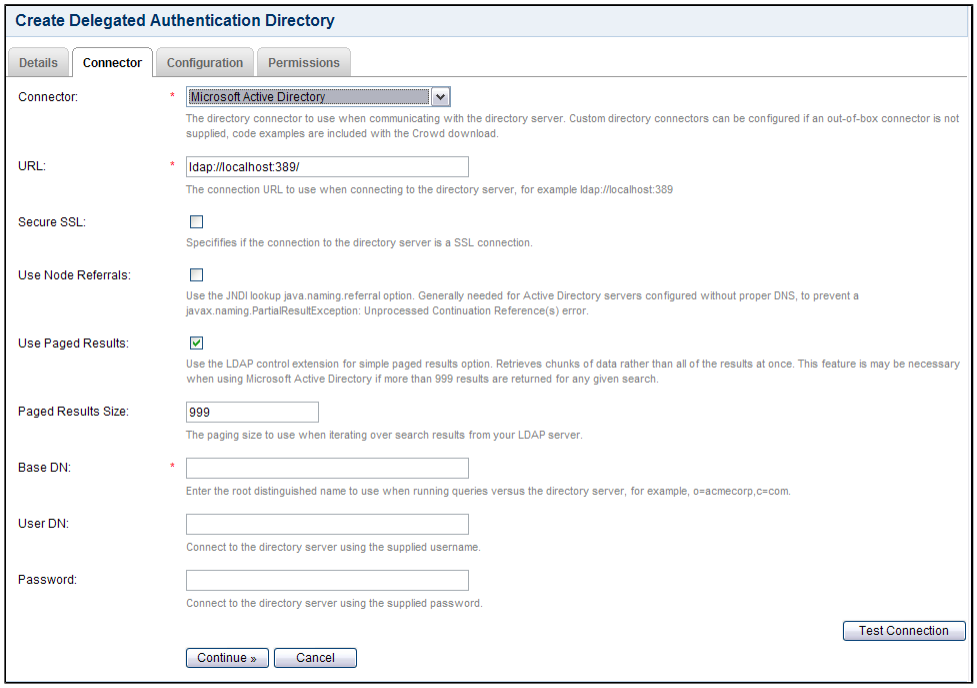
Screenshot 1: Directory details



Attribute	Description
Name	The name used to identify the directory within Crowd. For example: 'Chicago Employees' or 'Web Customers'.
Description	More information about this directory.
Active	Only deselect this if you wish to prevent all users within the directory from accessing all mapped applications.

Configuring Connector Details

Screenshot 2: Connector



Attribute	Description
Connector	The directory connector to use when communicating with the directory server.
URL	The connection URL to use when connecting to the directory server, e.g.: `ldap://localhost:389`, or port 636 for SSL.
Secure SSL	Specifies whether the connection to the directory server is an SSL connection.
Use Node Referrals	Use the JNDI lookup java.naming.referral option. Generally needed for Active Directory servers configured without proper DNS, to prevent a 'javax.naming.PartialResultException: Unprocessed Continuation Reference(s)' error.
Use Paged Results	Use the LDAP control extension for simple paging of search results. Retrieves chunks of data rather than all of the search results at once. This feature may be necessary when using Microsoft Active Directory if more than 999 results are returned for any given search.
Paged Results Size	Enter the desired page size i.e. the maximum number of search results to be returned per page, when paged results are enabled. Defaults to 999 results. This option is available from Crowd 1.1.1.
Base DN	Enter the root distinguished name to use when running queries versus the directory server, e.g.: `o=acmecorp,c=com`.
User DN	Distinguished name of the user that Crowd will use when connecting to the directory server.
Password	The password that Crowd will use when connecting to the directory server.

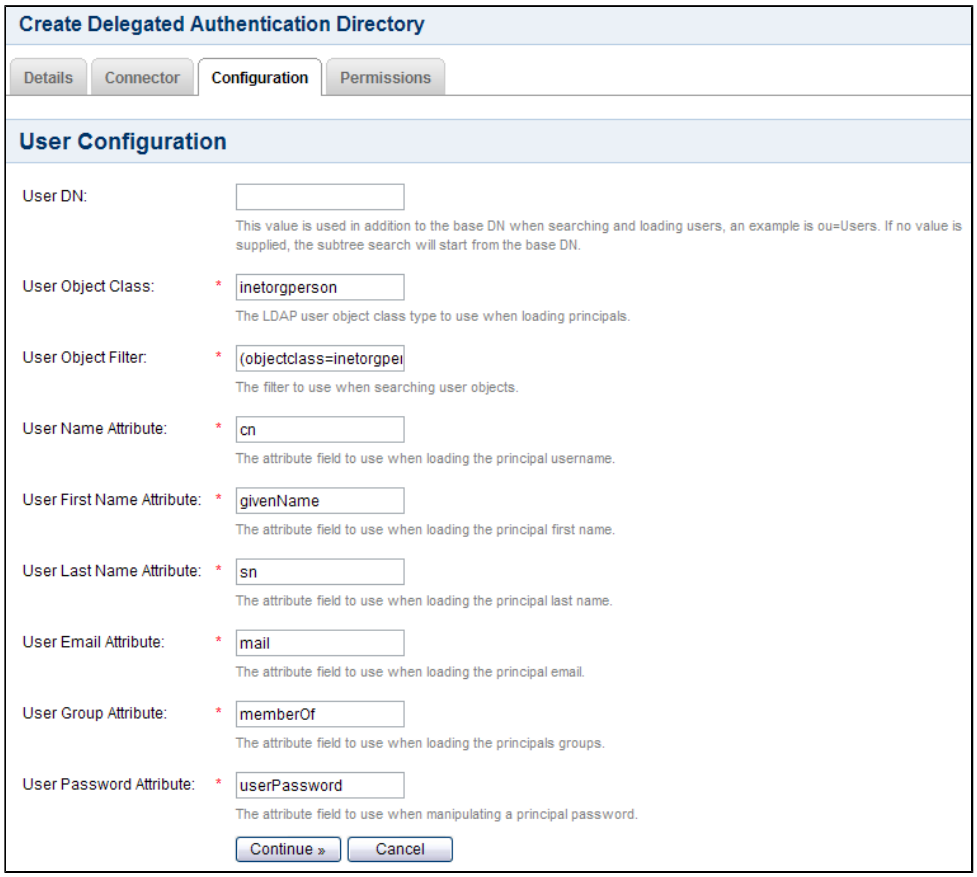
We have shown the settings for Active Directory. For details about the settings for your specific directory server, please see:

- Microsoft Active Directory
- SunONE
- OpenLDAP

- Apache Directory Server (ApacheDS)
- Novell eDirectory
- Posix Schema for LDAP
- Generic LDAP Directories

Configuring LDAP Object and Attribute Settings

Screenshot 3: Configuration



Attribute	Description
User DN	This value is used in addition to the base DN when searching and loading users. An example is `ou=Users`. If no value is supplied, the subtree search will start from the base DN.
User Object Class	This is the name of the class used for the LDAP user object.
User Object Filter	The filter to use when searching user objects.
User Name Attribute	The attribute field to use when loading the username.
User First Name Attribute	The attribute field to use when loading the user's first name.
User Last Name Attribute	The attribute field to use when loading the user's last name.
User Email Attribute	The attribute field to use when loading the user's email address.
User Group Attribute	The attribute field to use when loading the user's groups.
User Password Attribute	The attribute field to use when loading a user's password.

Please refer to the notes on LDAP object structures in the page about LDAP connectors.

Next Steps

Once you have configured the directory's permissions, you have finished configuring your new directory.

Next steps will be:

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. 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.

The username must be the same in the Crowd Delegated Authentication directory and in the LDAP directory. Changing the username in LDAP will break the link to the Crowd Delegated Authentication directory.

RELATED TOPICS

- Using the Directory Browser
- Adding a Directory
    - Configuring an Internal Directory
    - Configuring an LDAP Directory Connector
    - Configuring a Custom Directory Connector
    - Configuring a Delegated Authentication Directory
- Specifying Directory Permissions
- Importing Users and Groups into a Directory
    - Importing Users from Atlassian Confluence
    - Importing Users from Atlassian JIRA
    - Importing Users from Jive Forums
    - Importing Users from CSV Files
    - Importing Users from Atlassian Bamboo
    - Importing Users from One Crowd Directory into Another

Crowd Documentation

Specifying Directory Permissions

This page last changed on May 05, 2008 by smaddox.

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, groups and roles 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.

Below, we tell you about directory-level permissions. You can also read more about application-level directory permissions.

Directory-Level Permissions

Permission	Description
Add Group	Allows applications to add groups to the directory.
Add User	Allows applications to add users to the directory.
Add Role	Allows applications to add roles to the directory.
Modify Group	Allows applications to modify groups in the directory.
Modify User	Allows applications to modify users in the directory.
Modify Role	Allows applications to modify roles in the directory.
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.
Remove Role	Allows applications to delete roles from the directory.

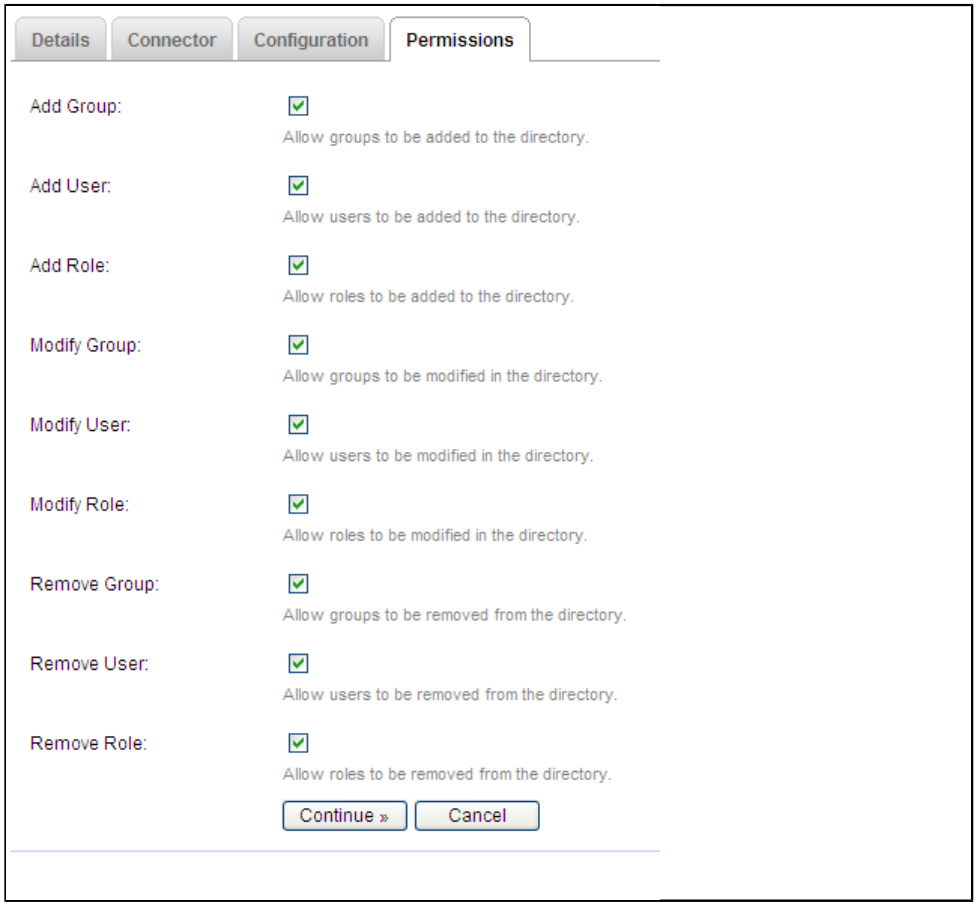
When you add a new directory, all of its permissions are enabled by default.

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. Click the 'Permissions' tab. This will display a list of permissions as shown in the screenshot below.

- To enable a directory permission, select the corresponding check-box.
- To disable a directory permission, deselect the corresponding check-box.

Screenshot: 'Directory Permissions'



See Also

To control which users within a directory may access a mapped application, see Specifying which Groups can access an Application.

RELATED TOPICS

Specifying an Application's Directory Permissions

- Using the Directory Browser
- Adding a Directory
  - Configuring an Internal Directory
  - Configuring an LDAP Directory Connector
    - Microsoft Active Directory
      - Configuring an SSL Certificate for Microsoft Active Directory
    - SunONE
    - OpenLDAP
    - Apache Directory Server (ApacheDS)
    - Novell eDirectory
    - Posix Schema for LDAP
    - Generic LDAP Directories
  - Configuring a Custom Directory Connector
  - Configuring a Delegated Authentication Directory
- Specifying Directory Permissions
- Importing Users and Groups into a Directory
  - Importing Users from Atlassian Confluence

Importing Users and Groups into a Directory

This page last changed on May 05, 2008 by smaddox.

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 organisation, and give you a consolidated, centralised 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).

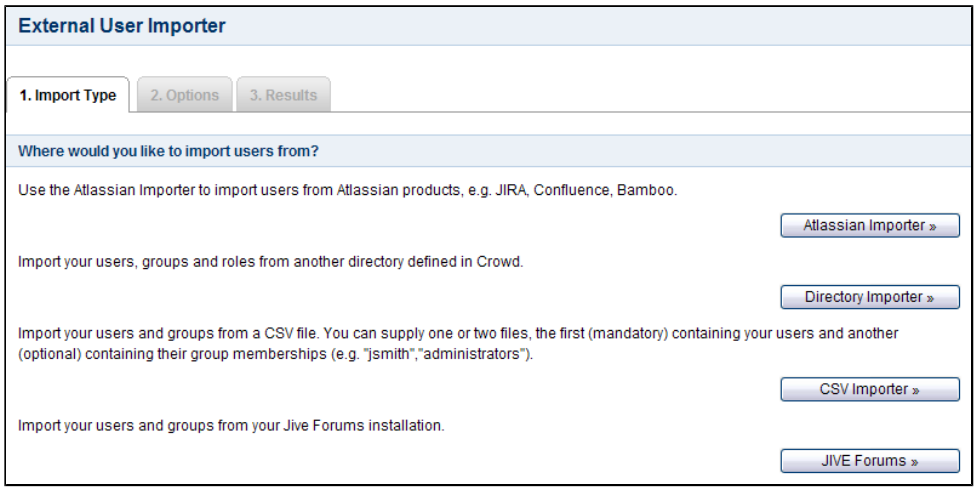
For example, your organisation might currently have user IDs for Atlassian JIRA users stored within JIRA's database, and user IDs for Jive Forums users stored within Jive's database. You could use Crowd to import all the user IDs from both places into Microsoft Active Directory.

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. 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 (see below). Click the button corresponding to 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

Screenshot: 'Select Import Type'



Related Topics

- Using the Directory Browser
- Adding a Directory
  - Configuring an Internal Directory
  - Configuring an LDAP Directory Connector
    - Microsoft Active Directory
      - Configuring an SSL Certificate for Microsoft Active Directory
    - SunONE
    - OpenLDAP
    - Apache Directory Server (ApacheDS)
    - Novell eDirectory
    - Posix Schema for LDAP
    - Generic LDAP Directories

Importing Users from Atlassian Confluence

This page last changed on May 05, 2008 by smaddox.

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.

Before you begin

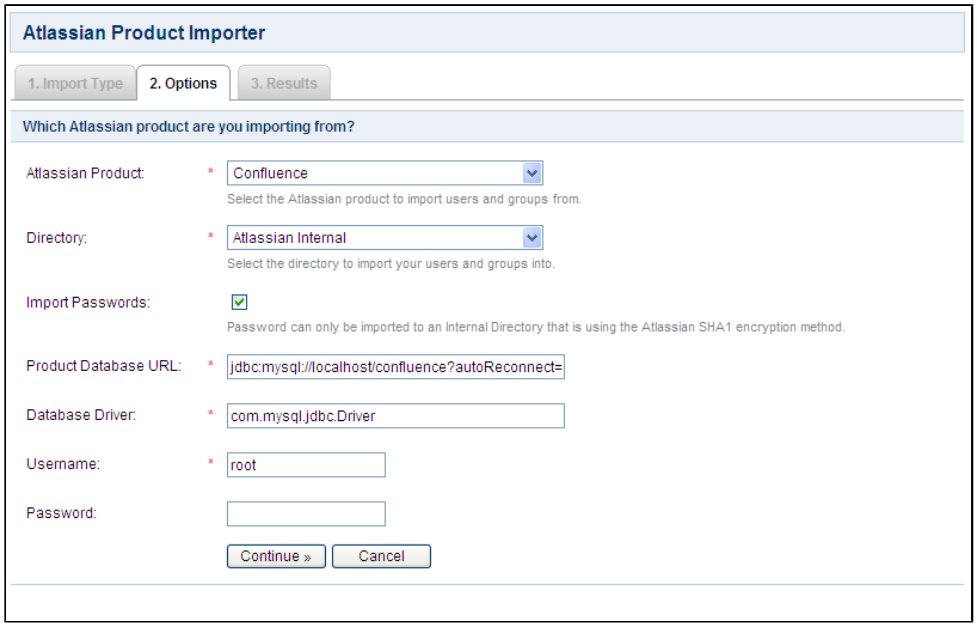
You will need to have installed the Confluence instance's database JDBC driver in the Crowd CLASS-PATH.

To import users and groups from Atlassian Confluence into a Crowd 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 'Import Users' link.
4. This will display the 'Import Type' screen. Click the 'Atlassian Importer' button.
5. This will display the 'Options' screen. Complete the fields as follows:
   - 'Atlassian Product' — Select 'Confluence'.
   - 'Directory' — Select the directory that is mapped to the Confluence application.
   - '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.

    The import process will log in to the database, not into Confluence.
6. Click the 'Continue' button to import the users from your Confluence instance into your Crowd directory.
7. The 'Results' 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 Confluence Users'



Next Step

To give the imported groups access to the Confluence 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
    - Microsoft Active Directory
      - Configuring an SSL Certificate for Microsoft Active Directory
    - SunONE
    - OpenLDAP
    - Apache Directory Server (ApacheDS)
    - Novell eDirectory
    - Posix Schema for LDAP
    - Generic LDAP Directories
  - Configuring a Custom Directory Connector
  - Configuring a Delegated Authentication Directory
- Specifying Directory Permissions
- Importing Users and Groups into a Directory
  - Importing Users from Atlassian Confluence
  - Importing Users from Atlassian JIRA
  - 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 Atlassian Bamboo
  - Importing Users from One Crowd Directory into Another

Crowd Documentation

Importing Users from Atlassian JIRA

This page last changed on May 05, 2008 by smaddox.

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-SHA1'. Otherwise, users' passwords will not be copied across to Crowd.

Before you begin

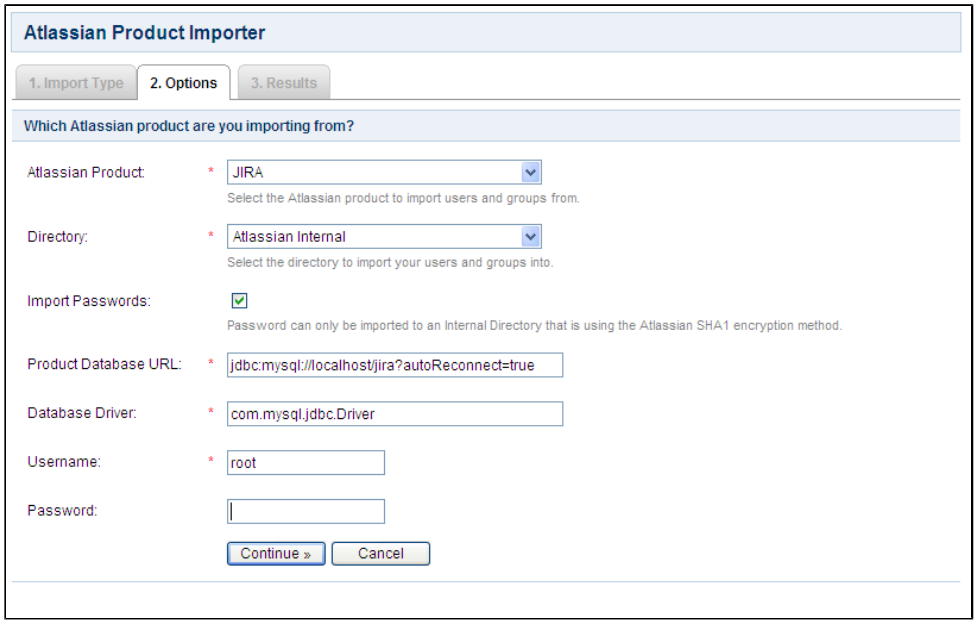
You will need to have installed the JIRA instance's database JDBC driver in the Crowd CLASS-PATH.

To import users and groups from Atlassian JIRA into a Crowd 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 'Import Users' link.
4. This will display the 'Import Type' screen. Click the 'Atlassian Importer' button.
5. This will display the 'Options' screen. Complete the fields as follows:
   - 'Atlassian Product' — Select 'JIRA'.
   - 'Directory' — Select the directory that is mapped to the JIRA application.
   - '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`).
   - '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.
6. Click the 'Continue' button to import the users from your JIRA instance into your Crowd directory.
7. The 'Results' 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 JIRA Users'



Next Step

To give the imported groups access to the JIRA 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
    - Microsoft Active Directory
      - Configuring an SSL Certificate for Microsoft Active Directory
    - SunONE
    - OpenLDAP
    - Apache Directory Server (ApacheDS)
    - Novell eDirectory
    - Posix Schema for LDAP
    - Generic LDAP Directories
  - Configuring a Custom Directory Connector
  - Configuring a Delegated Authentication Directory
- Specifying Directory Permissions
- Importing Users and Groups into a Directory
  - Importing Users from Atlassian Confluence
  - Importing Users from Atlassian JIRA
  - 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 Atlassian Bamboo
  - Importing Users from One Crowd Directory into Another

Crowd Documentation

Importing Users from Jive Forums

This page last changed on May 05, 2008 by smaddox.

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-5.5.20/common/lib` and restart Crowd.

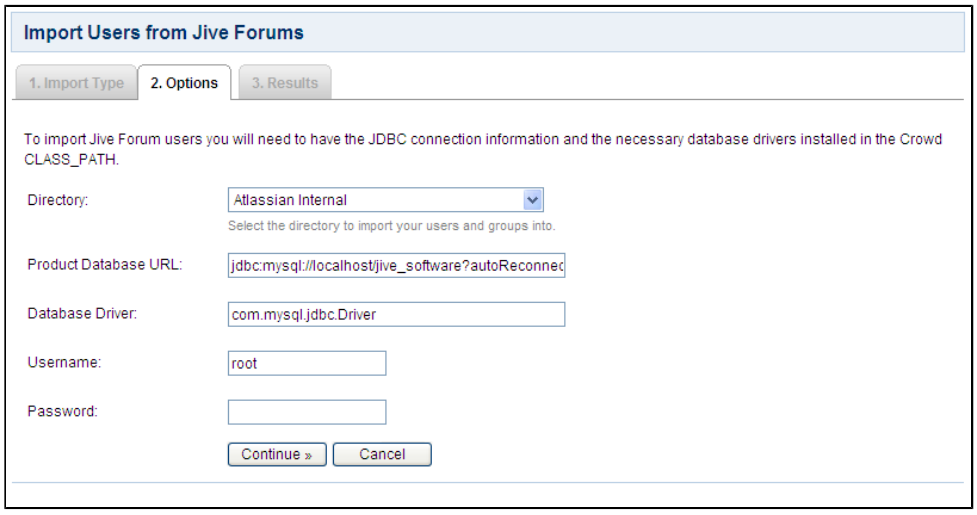
Note: the passwords for users in Jive will not be copied across to Crowd as they are stored as hashes in Jive's internal database.

To import users and groups from Jive Forums into a Crowd directory,

1. Login to the Crowd Administration Console.
2. Click the 'Directories' link in the top navigation bar.
3. This will display the Directory 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.

     The import process will log in to the database, not to Jive Forums.
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'



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
    - Microsoft Active Directory
      - Configuring an SSL Certificate for Microsoft Active Directory
    - SunONE
    - OpenLDAP
    - Apache Directory Server (ApacheDS)
    - Novell eDirectory
    - Posix Schema for LDAP
    - Generic LDAP Directories
  - Configuring a Custom Directory Connector
  - Configuring a Delegated Authentication Directory
- Specifying Directory Permissions
- Importing Users and Groups into a Directory
  - Importing Users from Atlassian Confluence
  - Importing Users from Atlassian JIRA
  - 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 Atlassian Bamboo
  - Importing Users from One Crowd Directory into Another

Crowd Documentation

Importing Users from CSV Files

This page last changed on May 07, 2008 by smaddox.

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.

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.
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 labelled '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. Login to the Crowd Administration Console.
2. Click the 'Directories' link in the top navigation bar.
3. This will display the Directory Browser. Click the 'Import Users' link.
4. This will display the 'Import Type' screen. Click the 'CSV Importer' button.
5. This will display the 'Configuration' tab of the 'CSV Importer'.
6. Enter the details of the CSV files as described in 'Configuring the CSV Importer'.

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

This page last changed on May 07, 2008 by smaddox.

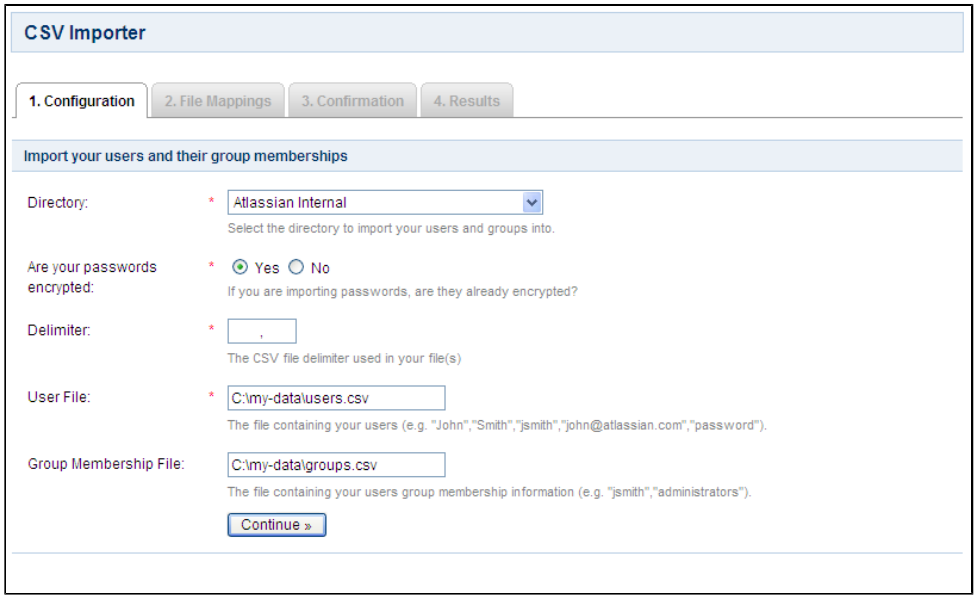
Once you have started the CSV Importer, the 'Configuration' screen allows you to specify information about the Crowd directory and CSV file(s) involved in the import.

Refer to information on preparing your CSV files.

To configure the CSV importer,

1. Start the CSV Importer.
2. This will display the 'Configuration' screen. Complete the fields as follows:
    - '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.
    - '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.
    - '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.
3. Click the 'Continue' button to map the CSV fields to the Crowd directory fields.

Screenshot: 'CSV Importer - Configuration'



RELATED TOPICS

- Using the Directory Browser
- Adding a Directory
    - Configuring an Internal Directory
    - Configuring an LDAP Directory Connector
        - Microsoft Active Directory
            - Configuring an SSL Certificate for Microsoft Active Directory
        - SunONE
        - OpenLDAP
        - Apache Directory Server (ApacheDS)
        - Novell eDirectory
        - Posix Schema for LDAP
        - Generic LDAP Directories
    - Configuring a Custom Directory Connector

Mapping CSV Fields to Crowd Fields

This page last changed on May 07, 2008 by smaddox.

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
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.
Username	Required. One of the rows on the screen must map this value to the CSV field containing the usernames.
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:

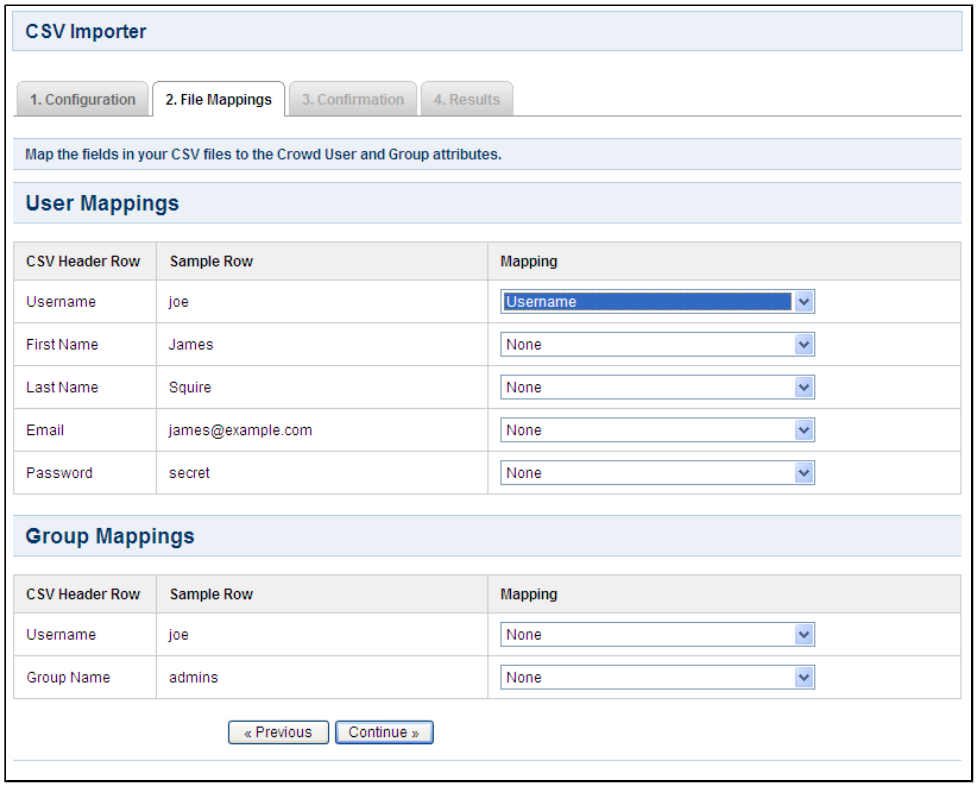
Crowd field	Description
Group Name	Required. One of the rows on the screen must map this value to the CSV field containing the names of the groups.
Username	Required. One of the rows on the screen must map this value to the CSV field containing the usernames.
None	

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'



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

Confirming the CSV Importer Configuration

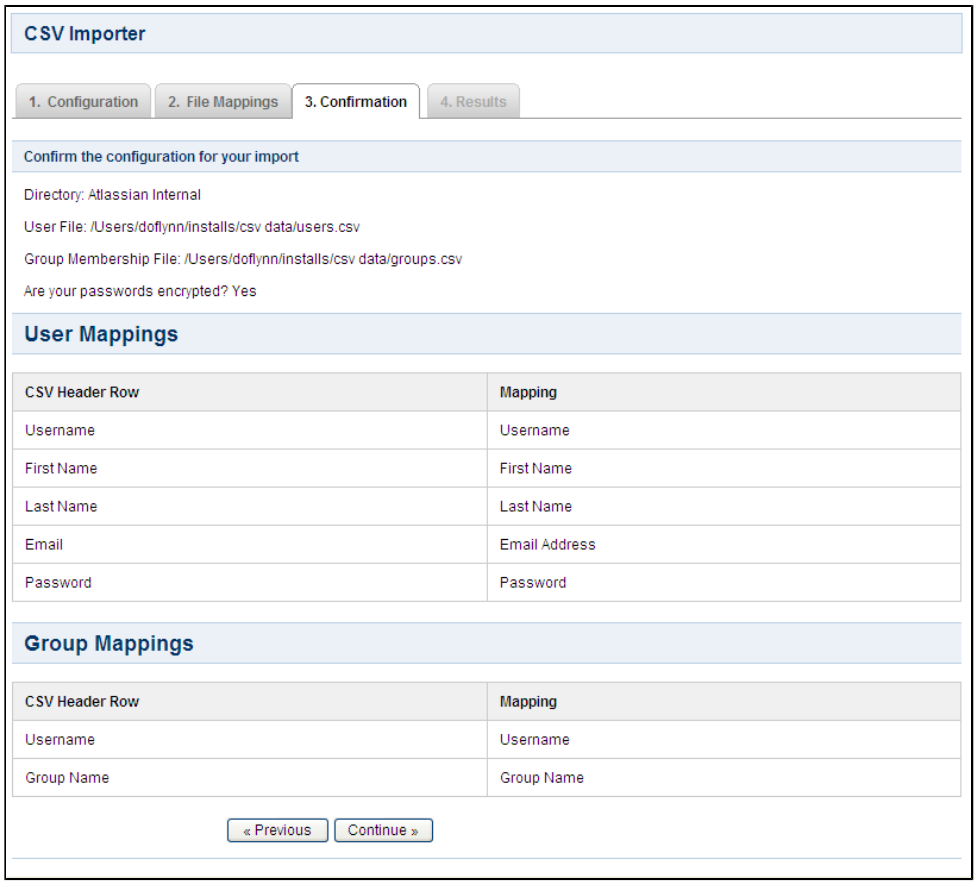
This page last changed on May 07, 2008 by smaddox.

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'



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

Viewing the Results of the Import

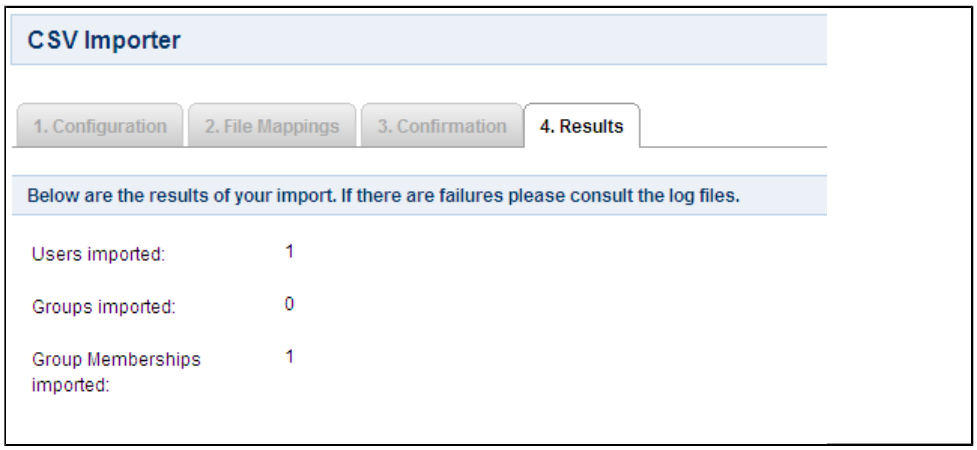
This page last changed on May 07, 2008 by smaddox.

The 'Results' screen shows the outcome of the CSV import.

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'



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

Importing Users from Atlassian Bamboo

This page last changed on May 05, 2008 by smaddox.

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.

Before you begin

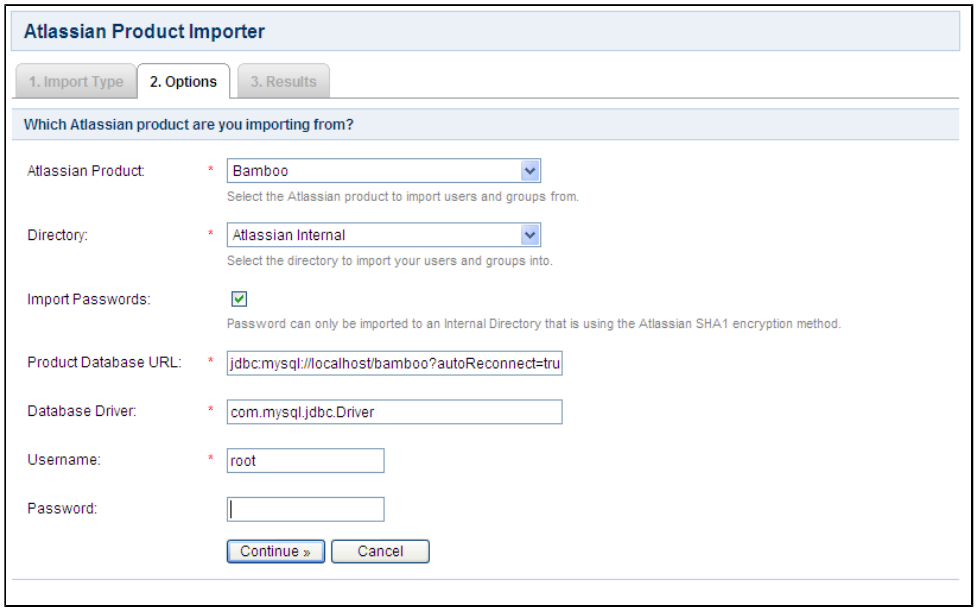
You will need to have installed the Bamboo instance's database JDBC driver in the Crowd CLASS-PATH.

To import users and groups from Atlassian Bamboo into a Crowd 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 'Import Users' link.
4. This will display the 'Import Type' screen. Click the 'Atlassian Importer' button.
5. This will display the 'Options' screen. Complete the fields as follows:
   - 'Atlassian Product' — Select 'Bamboo'.
   - 'Directory' — Select the directory that you have mapped to the Bamboo application in Crowd.
   - '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.

      The import process will log in to the database, not into Bamboo.
6. Click the 'Continue' button to import the users from your Bamboo instance into your Crowd directory.
7. The 'Results' 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 Bamboo Users'



Next Step

To give the imported groups access to the Bamboo 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
    - Microsoft Active Directory
      - Configuring an SSL Certificate for Microsoft Active Directory
    - SunONE
    - OpenLDAP
    - Apache Directory Server (ApacheDS)
    - Novell eDirectory
    - Posix Schema for LDAP
    - Generic LDAP Directories
  - Configuring a Custom Directory Connector
  - Configuring a Delegated Authentication Directory
- Specifying Directory Permissions
- Importing Users and Groups into a Directory
  - Importing Users from Atlassian Confluence
  - Importing Users from Atlassian JIRA
  - 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 Atlassian Bamboo
  - Importing Users from One Crowd Directory into Another

Crowd Documentation

Importing Users from One Crowd Directory into Another

This page last changed on May 06, 2008 by smaddox.

Once you have added a directory, you can import users, groups and roles 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, roles and memberships:

- From an LDAP directory to a Delegated Authentication directory.
- From one internal Crowd directory to another internal Crowd directory.

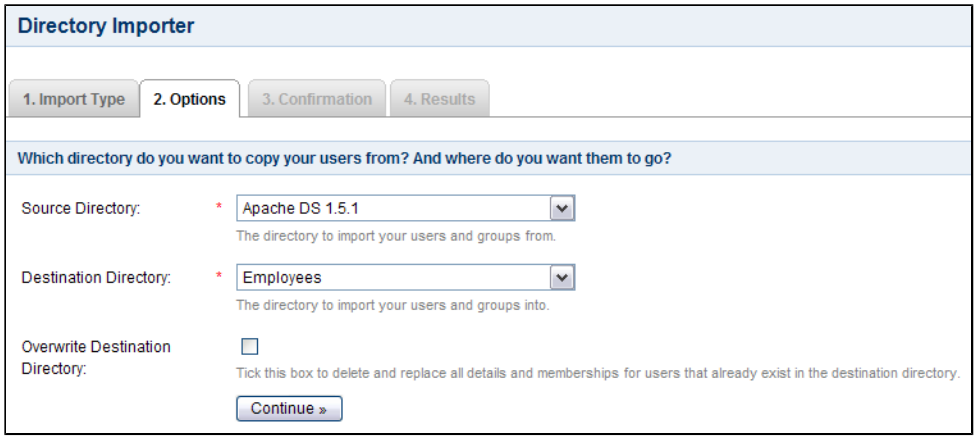
The 'Password Encryption' method must be the same in both directories, otherwise you will not be able to copy the users across.

The 'source directory' is the directory you want to copy users, groups and roles 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, groups and roles 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 'Directories' link in the top navigation bar.
5. This will display the Directory 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, groups and roles you want to copy.
    - 'Destination Directory' — Select the directory that you want to copy the users, groups and roles 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 or role 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, groups and roles have been imported into your Crowd directory.
    If the import of any users, groups or roles failed, please check the log files to find out why.

Screenshot: 'Import users from one directory to another'



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
        - Microsoft Active Directory
            - Configuring an SSL Certificate for Microsoft Active Directory
        - SunONE
        - OpenLDAP
        - Apache Directory Server (ApacheDS)
        - Novell eDirectory
        - Posix Schema for LDAP
        - Generic LDAP Directories
    - Configuring a Custom Directory Connector
    - Configuring a Delegated Authentication Directory
- Specifying Directory Permissions
- Importing Users and Groups into a Directory
    - Importing Users from Atlassian Confluence
    - Importing Users from Atlassian JIRA
    - 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 Atlassian Bamboo
    - Importing Users from One Crowd Directory into Another

Crowd Documentation

Managing Applications

This page last changed on May 05, 2008 by smaddox.

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 last changed on May 07, 2008 by smaddox.

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 three default applications:

- 'crowd' — this is the Crowd Administration Console, i.e. 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. The page How CrowdID works with Crowd does not exist.
- '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. To access the 'demo' application, go to http://localhost:8095/demo.

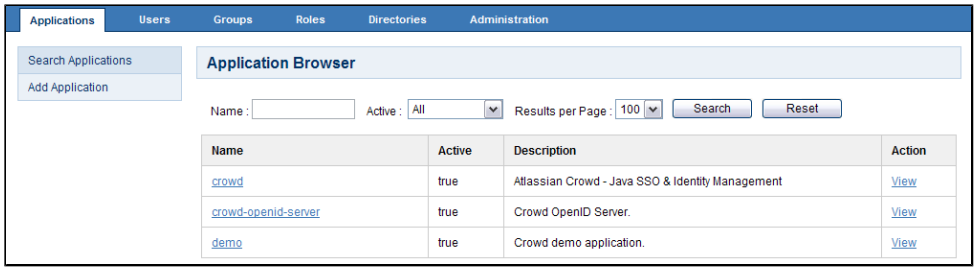
About the Application Browser

The Application Browser allows you to view and search for integrated applications.

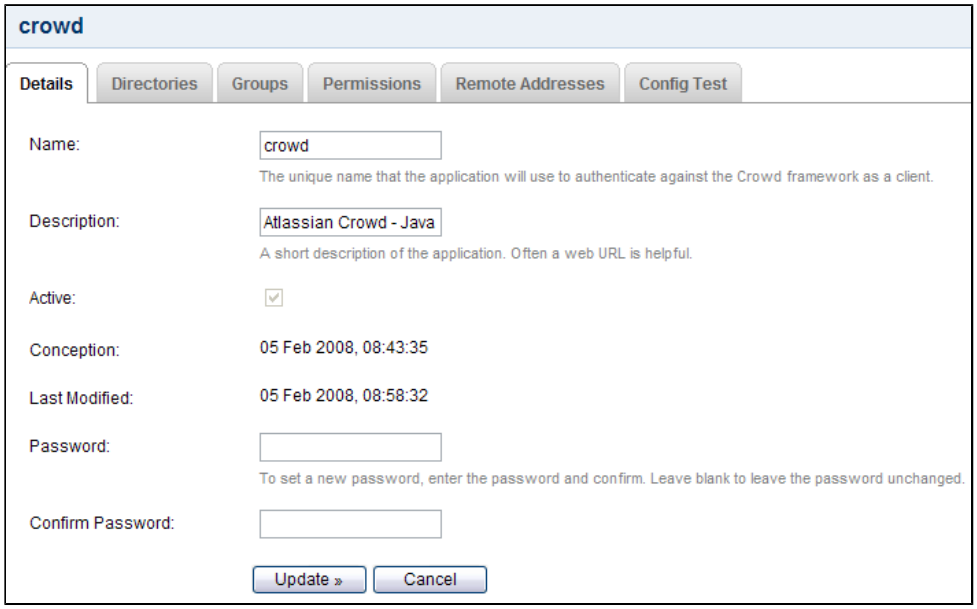
To use the Application Browser,

1. Log in to the Crowd Administration Console.
2. Click the 'Applications' tab in the top navigation bar.
3. 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.
4. To view/edit an application's details, click the 'View' link next to the specific application.

Screenshot 1: 'Application Browser'



Screenshot 2: 'View Application'



RELATED TOPICS

- Using the Application Browser
- Adding an Application
  - Integrating Crowd with Atlassian Bamboo
  - Integrating Crowd with Atlassian Confluence
    - Configuring Confluence for NTLM SSO
  - Integrating Crowd with Atlassian CrowdID
  - Integrating Crowd with Atlassian Crucible
  - Integrating Crowd with Atlassian FishEye
    - Configuring FishEye 1.3.x to talk to Crowd
  - Integrating Crowd with Atlassian JIRA
    - Configuring JIRA for NTLM SSO
  - Integrating Crowd with Acegi Security
    - Integrating AppFuse - a Crowd-Acegi Integration Tutorial
  - Integrating Crowd with Apache
  - Integrating Crowd with Jive Forums
    - Jive SSO
  - Integrating Crowd with Subversion
  - Integrating Crowd with a Custom Application
- Mapping a Directory to an Application
  - Specifying the Directory Order for an Application
  - Specifying an Application's Directory Permissions
    - Example of Directory Permissions
- Specifying which Groups can access an Application
- Specifying an Application's Address or Hostname
- Testing a User's Login to an Application
- Managing an Application's Session
- Deleting or Deactivating an Application

Crowd Documentation

Adding an Application

This page last changed on May 05, 2008 by smaddox.

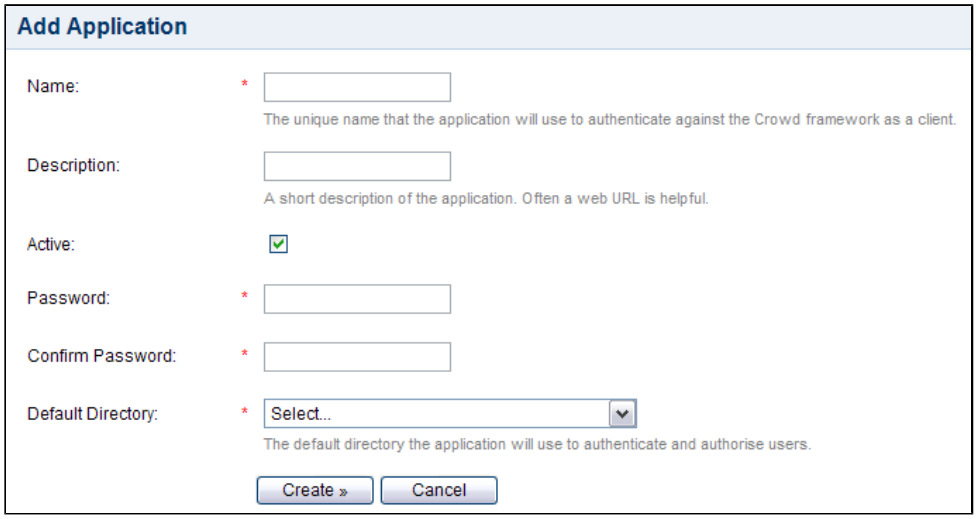
There are two overall steps to integrating an application with Crowd:

- Step 1. Configure Crowd to talk to the application — that is, add the application to Crowd via the Crowd Administration Console (see below). The application will then 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. Please see 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 Acegi Security
  - Integrating Crowd with Apache
  - Integrating Crowd with Jive Forums
  - Integrating Crowd with Subversion
  - Integrating Crowd with a Custom Application

To add an application to Crowd,

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 'Add Application' in the left-hand menu.
4. This will display the 'Add Application' screen (see screenshot). Complete the fields as described in the table below. Note that you will need to select a suitable directory to contain the application's users.
5. Click the 'Create' button to create the application. A number of tabs will now be displayed.
6. To choose which users within the directory may authenticate against the application, either:
   - Click the 'Groups' tab and select one or more groups of users, then click the 'Add' button; OR
   - Click the 'Directories' tab and change 'Allow all to authenticate' to 'True'. (The default is 'False'.)
7. Click the 'Permissions' tab and set the directory permissions for the application.
8. Click the 'Remote Addresses' tab and specify the IP address or hostname of the application. (The default is 'localhost'.)
9. Click the 'Update' button to save your changes.
10. If you wish, you can click the 'Config Test' tab and verify that a user can log in to the application.

Screenshot: 'Add Application'



Attribute	Description
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 web URL is often helpful.
Active	Only deselect this if you wish to prevent all users (from all directories) from accessing this application.
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.
Default Directory	A directory that contains relevant users. Note: Additional directories can be added later.

Next Steps

After adding an application, you may want to:

- map additional directories to the application, and
- set each directory's permissions for the application.

RELATED TOPICS

- Using the Application Browser
- Adding an Application
  - Integrating Crowd with Atlassian Bamboo
  - Integrating Crowd with Atlassian Confluence
    - Configuring Confluence for NTLM SSO
  - Integrating Crowd with Atlassian CrowdID
  - Integrating Crowd with Atlassian Crucible
  - Integrating Crowd with Atlassian FishEye
    - Configuring FishEye 1.3.x to talk to Crowd
  - Integrating Crowd with Atlassian JIRA
    - Configuring JIRA for NTLM SSO
  - Integrating Crowd with Acegi Security
    - Integrating AppFuse - a Crowd-Acegi Integration Tutorial
  - Integrating Crowd with Apache
  - Integrating Crowd with Jive Forums
    - Jive SSO
  - Integrating Crowd with Subversion
  - Integrating Crowd with a Custom Application
- Mapping a Directory to an Application

Crowd Documentation

Integrating Crowd with Atlassian Bamboo

This page last changed on May 07, 2008 by alui.

Atlassian's Bamboo integration server can quickly be configured to use the atlassian-user libraries to link in single or multiple directory servers through Crowd.

Currently Crowd supports centralised authentication and single sign-on for Bamboo versions 1.2.2 and later.

Due to incompatible atlassian-user libraries, Bamboo releases prior to 1.2.2 are not compatible with latest version of Crowd. We recommend that you upgrade to the latest version of Bamboo before attempting to integrate 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 Bamboo (version 1.2.2 or later). Refer to the Bamboo Installation Guide for detailed information on how to do this. We will refer to the Bamboo root folder as `BAMBOO`. For the purposes of this document, we will assume that you have used the Standalone (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, and repackage the EAR/WAR.
3. After Bamboo is set up, make sure Bamboo is not running when 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

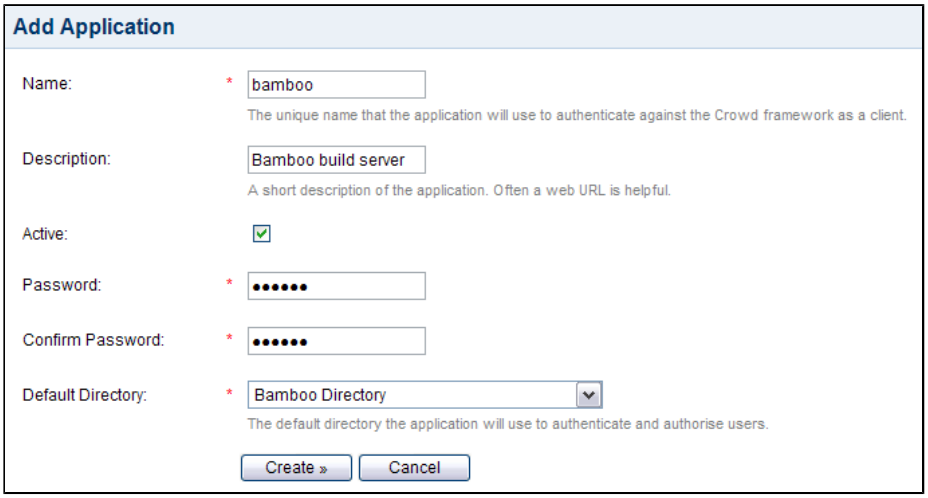
1. Create a Crowd directory: The Bamboo application will need to authenticate users against a directory configured in Crowd. You will need to set up a directory in Crowd for Bamboo. For more information on how to do this, see Adding a Directory. We will assume that the directory is called Bamboo Directory 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 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 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 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 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 Bamboo Directory:

1. Log in to the Crowd Administration Console and navigate to Applications > Add Application.
2. Complete the form to add the Bamboo application:

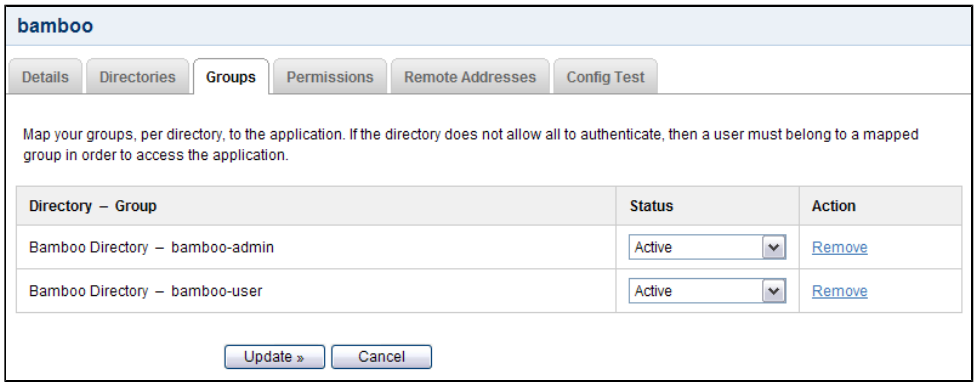


Attribute	Description
Name	The username which the application will use when it authenticates against the Crowd framework as a client. This value must be unique, i.e. it cannot be used by more than one application client.
Description	A short description of the application. Note: A web URL is often helpful.
Active	Only deselect this if you wish to prevent all users (from all directories) from accessing this application.
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.
Default Directory	A directory that contains relevant users. Note: Additional directories can be added later.

The Name and Password values must match the application.name and application.password that you set in the `Bamboo/webapp/WEB-INF/classes/crowd.properties` (see Step 2 below).

1.3 Specify which Users can Log In to Bamboo

Now that Crowd is aware of the Bamboo application, Crowd needs to know which users can authenticate (log in) to Bamboo via Crowd. 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 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

Please see Specifying an Application's Address or Hostname. Please note:

- If Bamboo is on a different host to Crowd:
  If you are running Bamboo on a different host to Crowd, you will need to modify the permissible hosts via the Remote Addresses tab. This lists the hosts/IP addresses that are allowed to authenticate to Crowd. If Bamboo is remote to Crowd, add the IP address of your Bamboo server and ensure the "Status" field is set to "true". Remove the entry for `localhost`.
- If Bamboo is on the same host as Crowd:
  By default, when you add an application, `localhost` is a permissible foreign host. However, you will also need to manually add the IP address `127.0.0.1`, as incoming requests to Crowd from Bamboo (both on the same, local, host) may be from the host `127.0.0.1` and not `localhost`. Crowd does not do a DNS lookup of the hostname; rather, it compares the values as is. Ensure the "Status" field is set to "true".

Step 2. Configuring Bamboo to Talk to Crowd

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

If you are using Bamboo 2.0 or later, you can skip this step. The Crowd client libraries and `crowd.properties` file will be included in the Bamboo 2.0 installation download.

Bamboo 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 modify the Bamboo application by editing the standalone application, which is an exploded WAR stored in `BAMBOO/webapp`.

1. 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
CROWD/client/conf/crowd.properties	BAMBOO/webapp/WEB-INF/classes

   Bamboo 1.2.4 release is not compatible with Crowd by default.
   You will need to remove the following file from Bamboo's `WEB-INF/lib/seraph-0.7.23.jar` directory and replace it with the following:
   http://repository.atlassian.com/maven2/com/atlassian/seraph/atlassian-seraph/0.10/atlassian-seraph-0.10.jar
   Note that Bamboo versions 2.0 or later are compatible with Crowd, and you don't need to replace the Seraph jar.

2.2 Edit Bamboo's crowd.properties file

Configure the Bamboo application's properties to determine how Crowd will interact with Bamboo.

1. Edit `BAMBOO/webapp/WEB-INF/classes/crowd.properties`. Change the following properties:

Key	Value
application.name	bamboo
application.password	set a password
crowd.server.url	http://localhost:8095/crowd/services/
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.

   If your Crowd server's port is configured differently from the default (8095), set it accordingly.

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). Bamboo does not use any of the other attributes of 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.

This applies to the Crowd Administration Console's `crowd.properties` file too. You may find this particularly useful when integrating with a WAR deployment of an integrated application.

Example:

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

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 `Bamboo/webapp/WEB-INF/classes/atlassian-user.xml` file 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 centralised authentication. If you wish to enable single sign-on (SSO) to Bamboo, edit `BAMBOO/webapp/WEB-INF/classes/seraph-config.xml`. Comment out the `authenticator` node :

```
<!--<authenticator class="com.atlassian.bamboo.user.authentication.BambooAuthenticator"/>-->
```

and add a new one:

```
<authenticator class="com.atlassian.crowd.integration.seraph.BambooAuthenticator"/>
```

Bamboo's authentication and access request calls will now be performed using Seraph.

2.4 Configure External User Management in Bamboo

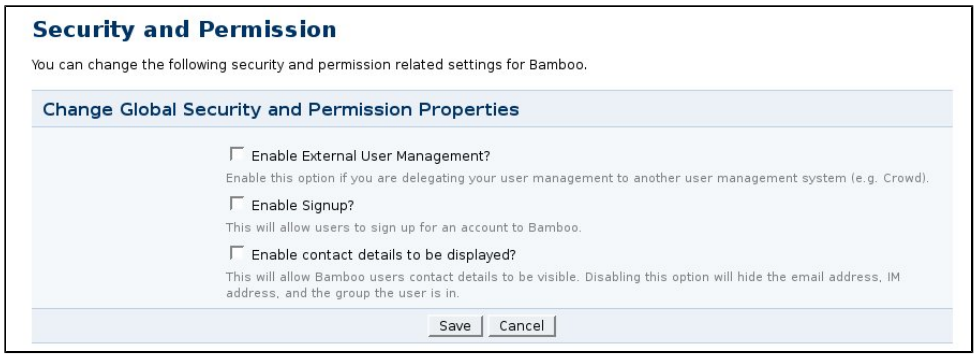
For Bamboo to integrate successfully with Crowd, Bamboo's 'External User Management' option needs to be:

- 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

SSO is optional

Single sign-on (SSO) is optional when integrating Bamboo and other Atlassian products. To use centralised authentication without SSO, skip the steps below.

To configure Seraph-based authentication:

1. Edit the `\bamboo\webapp\WEB-INF\classes\seraph-config.xml` and change the authenticator node to read:

```
<authenticator class="com.atlassian.crowd.integration.seraph.BambooAuthenticator"/>
```

2. Bamboo will also require the latest version of Atlassian Seraph. Copy this JAR file into Bamboo's `\bamboo\webapp\WEB-INF\lib` directory and remove the old file.

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).

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 centralised authentication in action!
- If you have enabled SSO, you can try adding the 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!

RELATED TOPICS

- Using the Application Browser

- Adding an Application
  - Integrating Crowd with Atlassian Bamboo
  - Integrating Crowd with Atlassian Confluence
    - Configuring Confluence for NTLM SSO
  - Integrating Crowd with Atlassian CrowdID
  - Integrating Crowd with Atlassian Crucible
  - Integrating Crowd with Atlassian FishEye
    - Configuring FishEye 1.3.x to talk to Crowd
  - Integrating Crowd with Atlassian JIRA
    - Configuring JIRA for NTLM SSO
  - Integrating Crowd with Acegi Security
    - Integrating AppFuse - a Crowd-Acegi Integration Tutorial
  - Integrating Crowd with Apache
  - Integrating Crowd with Jive Forums
    - Jive SSO
  - Integrating Crowd with Subversion
  - Integrating Crowd with a Custom Application
- Mapping a Directory to an Application
  - Specifying the Directory Order for an Application
  - Specifying an Application's Directory Permissions
    - Example of Directory Permissions
- Specifying which Groups can access an Application
- Specifying an Application's Address or Hostname
- Testing a User's Login to an Application
- Managing an Application's Session
- Deleting or Deactivating an Application

Crowd Documentation

Integrating Crowd with Atlassian Confluence

This page last changed on May 05, 2008 by smaddox.

Atlassian's popular Confluence wiki can quickly be configured to use the `atlassian-user` libraries to link in single or multiple directory servers through Crowd.

If you are using NTLM for Windows authentication, you may want to read about configuring Crowd's Confluence NTLM plugin for single sign on.

Compatibility of Confluence and Crowd Versions

For best performance and support, please ensure that your Crowd and Confluence versions are compatible:

- Crowd versions 1.2 and later support Confluence 2.6.2 and later.
- Confluence 2.6.1 and earlier are not supported with this version of Crowd.
- If you are using Confluence 2.8 or later, please upgrade to Crowd 1.3.2 or later.
  Explanation: With Confluence 2.8 the atlassian-user interface has changed, and Crowd 1.3.2 provides the required update to Crowd's atlassian-user integration module.

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 Confluence (version 2.6.2 or later). Refer to the Confluence installation guide for detailed information on how to do this. We will refer to the Confluence root folder as `CONFLUENCE`. For the purposes of this document, we will assume that the Standalone (ie. the easier) installation method of Confluence has been used. If you need to install Confluence as an EAR/WAR, simply explode the EAR/WAR and make the necessary changes as described below, and repackage the EAR/WAR.
3. After Confluence is set up, make sure Confluence is not running when 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:

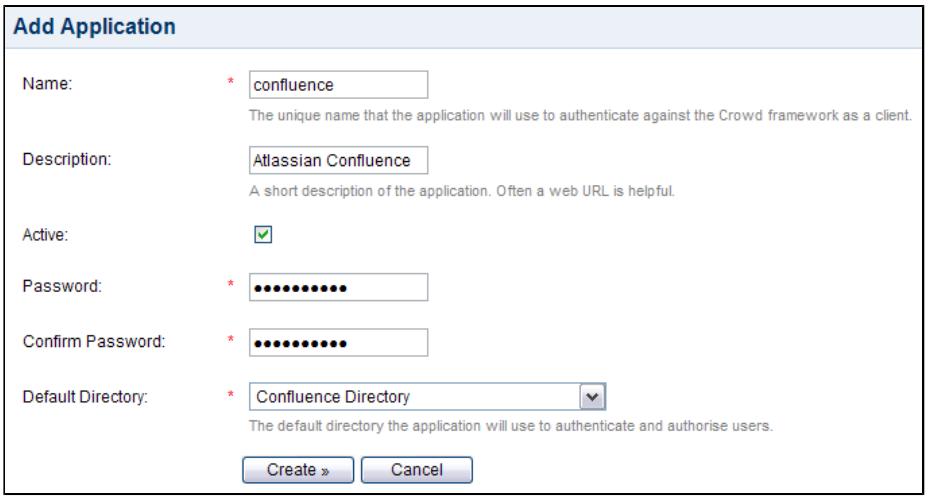
- If you have an existing Confluence deployment and would like to import existing users and groups into Crowd, use the Confluence Importer tool by navigating to Users > Import Users > Atlassian Importer. Select 'Confluence' as the Atlassian product, and the Confluence Directory as the directory into 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. Fill out the form to add the Confluence application:

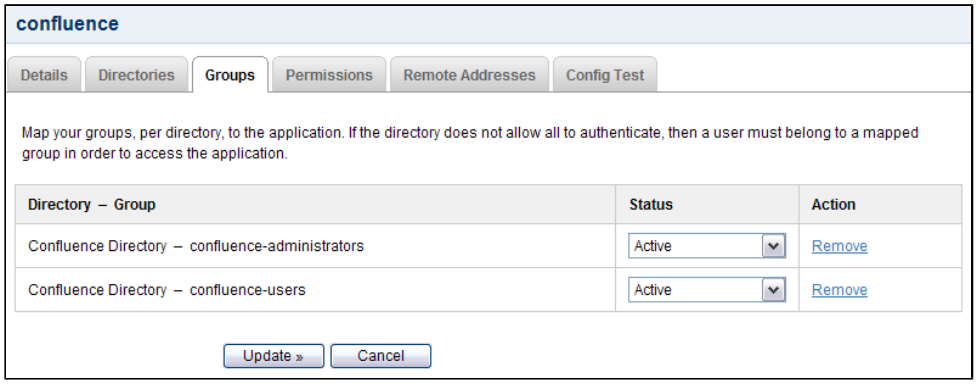


Attribute	Description
Name	The username which the application will use when it authenticates against the Crowd framework as a client. This value must be unique, i.e. it cannot be used by more than one application client.
Description	A short description of the application. Note: A web URL is often helpful.
Active	Only deselect this if you wish to prevent all users (from all directories) from accessing this application.
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.
Default Directory	A directory that contains relevant users. Note: Additional directories can be added later.

The Name and Password values must match the application.name and application.password that you set in the `CONFLUENCE/confluence/WEB-INF/classes/crowd.properties` (see Step 2 below).

1.3 Specify which Users can Log In to Confluence

Now that Crowd is aware of the Confluence application, Crowd needs to know which users can authenticate (log in) to Confluence via Crowd. 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

Please see Specifying an Application's Address or Hostname. Please note:

- If Confluence is on a different host to Crowd:
  If you are running Confluence on a different host to Crowd, you will need to modify the permissible hosts via the Remote Addresses tab. This lists the hosts/IP addresses that are allowed to authenticate to Crowd. If Confluence is remote to Crowd, add the IP address of your Confluence server and ensure the "Status" field is set to "true". Remove the entry for `localhost`.
- If Confluence is on the same host as Crowd:
  By default, when you add an application, `localhost` is a permissible foreign host. However, you will also need to manually add the IP address `127.0.0.1`, because incoming requests to Crowd from Confluence (both on the same, local, host) may be from the host `127.0.0.1` and not `localhost`. Crowd does not do a DNS lookup of the hostname. Instead, it compares the values themselves. Ensure the "Status" field is set to "true".

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 standalone application, which is an exploded WAR stored in `CONFLUENCE/confluence`.

1. 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.

   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-0.X.X.jar` file from Confluence's `CONFLUENCE-HOME/WEB-INF/lib/seraph-0.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

2. Edit `CONFLUENCE/confluence/WEB-INF/classes/crowd.properties`. Change the following properties:

Key	Value
application.name	confluence

application.password	Set a password.
crowd.server.url	http://localhost:8095/crowd/services/
session.validationinterval	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.

If your Crowd server's port is configured differently from the default (i.e. 8095), set it accordingly. 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). Confluence does not use any of the other attributes of 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.

This applies to the Crowd Administration Console's `crowd.properties` file too. You may find this particularly useful when integrating with a WAR deployment of an integrated application.

Example:

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

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 contents of the file is:

```
<atlassian-user>
    <repositories>

        <crowd key="crowd" name="Crowd Repository"/>

    </repositories>
</atlassian-user>
```

2. At this stage, Confluence is set up for centralised authentication. If you wish, you can now enable single sign-on (SSO) to Confluence.

   Skip this step if you are using the Confluence NTLM plugin to enable SSO. Instead, follow the instructions on configuring Confluence for NTLM SSO.
   Edit `CONFLUENCE/confluence/WEB-INF/classes/seraph-config.xml`. Comment out the `authenticator` node :

```
<!--<authenticator class="com.atlassian.confluence.user.ConfluenceAuthenticator"/>-->
```

   and add a new one:

```
<authenticator class="com.atlassian.crowd.integration.seraph.ConfluenceAuthenticator"/>
```

Confluence's authentication and access request calls will now be performed using Seraph.

2.3 Enable Confluence's External User Management

Once the setup is complete, you may optionally wish to enable a Confluence feature known as 'External User Management' in Confluence, to prevent Confluence administrators from creating/modifying users. For more information please see the Confluence documentation regarding External User Management.

- 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 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

When using the atlassian-user and Crowd framework together with Confluence, 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.)

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 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

- 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 centralised 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!

RELATED TOPICS

- Using the Application Browser
- Adding an Application
  - Integrating Crowd with Atlassian Bamboo
  - Integrating Crowd with Atlassian Confluence
    - Configuring Confluence for NTLM SSO
  - Integrating Crowd with Atlassian CrowdID
  - Integrating Crowd with Atlassian Crucible
  - Integrating Crowd with Atlassian FishEye
    - Configuring FishEye 1.3.x to talk to Crowd
  - Integrating Crowd with Atlassian JIRA
    - Configuring JIRA for NTLM SSO
  - Integrating Crowd with Acegi Security
    - Integrating AppFuse - a Crowd-Acegi Integration Tutorial
  - Integrating Crowd with Apache
  - Integrating Crowd with Jive Forums
    - Jive SSO
  - Integrating Crowd with Subversion
  - Integrating Crowd with a Custom Application
- Mapping a Directory to an Application
  - Specifying the Directory Order for an Application

Configuring Confluence for NTLM SSO

This page last changed on May 05, 2008 by smaddox.

Confluence NTLM plugin not officially supported by Atlassian

The Confluence NTLM plugin was written by a third party. Atlassian does not officially support the plugin. The Atlassian Crowd team will do our best to advise on any Crowd integration problems. Please refer to the plugin documentation for installation instructions and further support.

Out of the box, Confluence does not support Single Sign On (SSO) functionality. This page describes how to set up Confluence with NTLM SSO functionality using the Confluence NTLM plugin, Crowd, and Active Directory (AD) as your LDAP user repository.

Summary

The Confluence NTLM plugin enables the following authentication scenario:

- A user in a Windows domain logs into the Windows network, using their Active Directory username/password.
- Then, when they open Confluence in an Internet Explorer browser, they are seamlessly logged into Confluence.

The Crowd component then allows you to manage all users and groups in Active Directory. Crowd automatically ensures that users and groups are synchronised between AD and Confluence. For example, if a user/group is added/deleted from AD it will be automatically added/deleted from Confluence.

Components

Confluence NTLM plugin	NTLM is the protocol used by Windows for authentication. The Confluence NTLM plugin takes care of the Windows domain / Active Directory login to Confluence. You must be running a Windows Domain Controller with accounts set up in AD in order to use this plugin. If NTLM authentication is not available, the plugin allows standard form-based login to Confluence. Note: This plugin is not officially supported by Atlassian.
Crowd	Crowd takes care of the synchronisation of users/groups between Active Directory and Confluence. You will need to create an SSL connection between Crowd and the AD server if you would like to create users through Crowd. AD will not allow Crowd to add users or change their passwords unless the communication occurs over a secure connection.
Active Directory (AD) on Windows 2003 Server	Active Directory (AD) on Windows 2003 Server — you must already have an AD instance set up and running with a domain controller.
Confluence	The machine running Confluence must be part of the Windows domain or installed on the same box as the domain controller.

Steps

1. Back up your Confluence installation files and data:
   - Confluence Home directory. (See Confluence's Important Directories and Files for how to locate this).
   - Confluence installation directory (if you are using Confluence Standalone) or your Confluence webapp (if you are using Confluence EAR-WAR).
   - Your database (if you are not using the embedded database).

2. Download the Confluence NTLM plugin.
3. Install the plugin, following the instructions on the plugin documentation page.
4. In the `ldaputil.properties` file, insert the appropriate LDAP and Domain Controller information along with other parameters.
5. Install and configure Crowd.
6. Create a directory in Crowd for the AD LDAP server.
7. Create the Confluence application in Crowd and configure Crowd and Confluence to talk to each other, as described in Integrating Crowd with Atlassian Confluence.

   
   When following the above instructions, do not change the `seraph-config.xml` file to enable Crowd's SSO functionality. (I.e. don't change the `authenticator` node to read `<authenticator class="com.atlassian.crowd.integration.seraph.ConfluenceAuthenticator"/>`. Instead of Crowd's SSO authentication, we'll be using the Confluence NTLM plugin.
8. In AD, create the groups confluence-users and confluence-administrators. They should then appear in Crowd.
9. In AD, create an admin user and make them a member of the above groups in AD.
10. Create any additional groups that you would like in AD.
11. Log in to the Windows domain using your desktop login and then open Confluence in an Internet Explorer browser. You should be logged in automatically.

Additional Crowd Performance Tips

- Change the default cache setting timeout in the file `<CONFLUENCE>\WEB-INF\classes\crowd-ehcache.xml`. For performance reasons, increase the object caching to 7,200 seconds (2 hours): `timeToIdleSeconds="7200" timeToLiveSeconds="7200"`.
  This reduces the frequency of the requests from Crowd to the LDAP server when changes to LDAP objects (such as a group name or user attribute) are made, thus reducing the performance overhead.

- Turn on the 'Use Paged Results' option in the directory connector tab for the directory you've set up in Crowd.

Integrating Crowd with Atlassian CrowdID

This page last changed on May 07, 2008 by smaddox.

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.

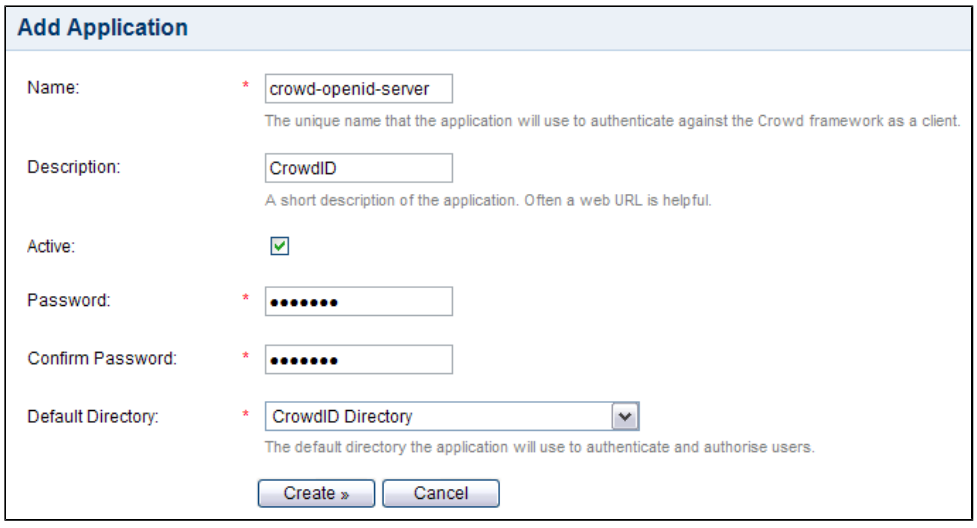
CrowdID also requires an administrator group to exist in the directory. You need to ensure that a `crowd-administrators` groups exist in the CrowdID Directory. Any user in this group will have CrowdID administrator access.

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. Fill out the form to add the CrowdID application:



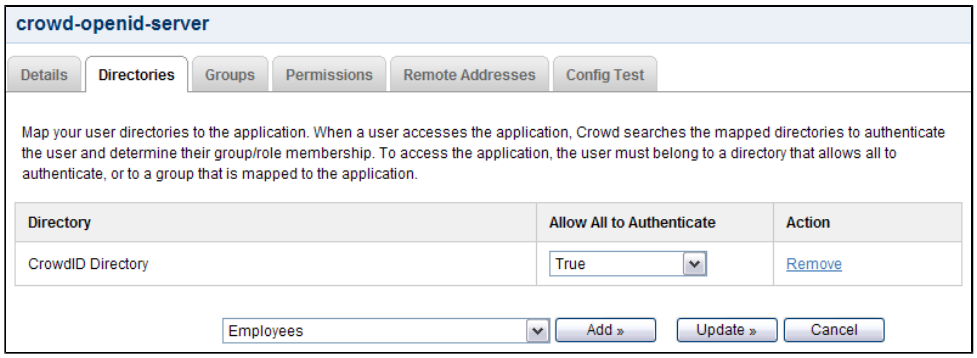
Attribute	Description

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 web URL is often helpful.
Active	Only deselect this if you wish to prevent all users (from all directories) from accessing this application.
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.
Default Directory	A directory that contains relevant users. Note: Additional directories can be added later.

The Name and Password values must match the application.name and application.password that you set in the `CROWD/crowd-openidserver-webapp/WEB-INF/classes/crowd.properties` (see Step 2 below).

1.3 Specify which Users can Log In to CrowdID

Now that Crowd is aware of the CrowdID application, Crowd needs to know which directories or users can authenticate (log in) via Crowd. 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:



For details please see Specifying which Groups can access an Application.

1.4 Specify the Address from which CrowdID can Log In to Crowd

Please see Specifying an Application's Address or Hostname. Please note:

- If CrowdID is on a different host to Crowd:
  If you are running the CrowdID on a different host to Crowd, you will need to modify the permissible hosts via the Remote Addresses tab. This lists the hosts/IP addresses that are allowed to authenticate to Crowd. If CrowdID is remote to Crowd, add the IP address of your CrowdID server and ensure the "Status" field is set to "true". Remove the entry for `localhost`.
- If CrowdID is on the same host as Crowd:
  By default, when you add an application, `localhost` is a permissible foreign host. However, you will also need to manually add the IP address `127.0.0.1`, as incoming requests to Crowd from CrowdID (both on the same, local, host) may be from the host `127.0.0.1` and not `localhost`. Crowd does not do a DNS lookup of the hostname. Instead, it compares the values as is. Ensure the 'Status' field is set to 'true'.

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
application.password	Set a password.
application.login.url	http://localhost:8095/openidserver
crowd.server.url	http://localhost:8095/crowd/services/
session.validationinterval	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.

If your Crowd server's port is configured differently from the default (i.e. 8095), set it accordingly. 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). The application.login.url should point to the correct host and port of the CrowdID application.

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
    - Configuring Confluence for NTLM SSO
  - Integrating Crowd with Atlassian CrowdID
  - Integrating Crowd with Atlassian Crucible
  - Integrating Crowd with Atlassian FishEye
    - Configuring FishEye 1.3.x to talk to Crowd
  - Integrating Crowd with Atlassian JIRA
    - Configuring JIRA for NTLM SSO
  - Integrating Crowd with Acegi Security
    - Integrating AppFuse - a Crowd-Acegi Integration Tutorial
  - Integrating Crowd with Apache
  - Integrating Crowd with Jive Forums
    - Jive SSO
  - Integrating Crowd with Subversion
  - Integrating Crowd with a Custom Application
- Mapping a Directory to an Application
  - Specifying the Directory Order for an Application
  - Specifying an Application's Directory Permissions
    - Example of Directory Permissions
- Specifying which Groups can access an Application
- Specifying an Application's Address or Hostname
- Testing a User's Login to an Application
- Managing an Application's Session
- Deleting or Deactivating an Application

Crowd Documentation

Integrating Crowd with Atlassian Crucible

This page last changed on May 05, 2008 by smaddox.

You can use Crowd to provide external authentication and authorisation for Atlassian's Crucible code review tool.

Crucible and FishEye

When you purchase and install Crucible, you also receive Atlassian's FishEye source-repository viewer. FishEye and Crucible share a common authentication mechanism and integration with 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 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.

Do not create a separate Crowd application for Crucible

Crucible will authenticate to Crowd using the application name and password which you have created for the FishEye application. (See Integrating Crowd with Atlassian FishEye.)

Configure Authorisation 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
- Mapping a Directory to an Application
- Specifying which Groups can access an Application
- Specifying an Application's Address or Hostname
- Testing a User's Login to an Application
- Managing an Application's Session
- Deleting or Deactivating an Application

Crowd Documentation

Integrating Crowd with Atlassian FishEye

This page last changed on May 05, 2008 by smaddox.

You can use Crowd to provide external authentication and authorisation for Atlassian's FishEye source-repository viewer.

Crowd supports centralised 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 to integrate Crowd with FishEye.
- Then follow the further instructions to integrate Crowd with Crucible.

On this page:

Error formatting macro: toc: java.lang.NullPointerException

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`.
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.

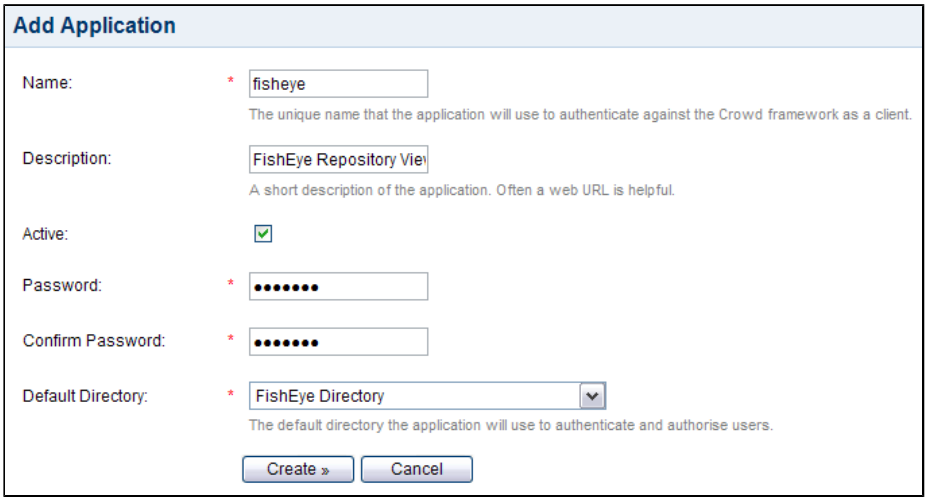
If you wish to use Crowd groups to control access to your FishEye repositories, you should set up your groups in Crowd. See the documentation on Creating Groups for more information on how to define these groups.

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. Fill out the form to add the FishEye application:

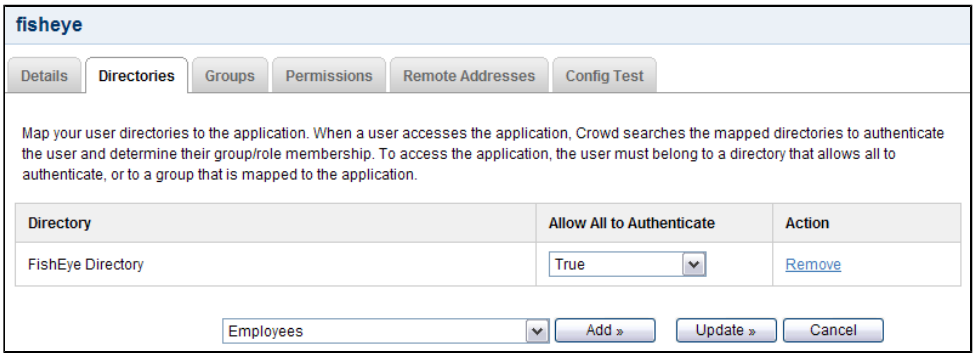


Attribute	Description
Name	The username which the application will use when it authenticates against the Crowd framework as a client. This value must be unique, i.e. it cannot be used by more than one application client.
Description	A short description of the application. Note: A web URL is often helpful.
Active	Only deselect this if you wish to prevent all users (from all directories) from accessing this application.
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.
Default Directory	A directory that contains relevant users. Note: Additional directories can be added later.

The Name and Password values 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

Now that Crowd is aware of the FishEye application, Crowd needs to know which users can authenticate (log in) to FishEye via Crowd. 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:



If you wish to authorise 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

Please see Specifying an Application's Address or Hostname. Please note:

- If FishEye is on a different host to Crowd:

If you are running FishEye on a different host to Crowd, you will need to modify the permissible hosts via the Remote Addresses tab. This lists the hosts/IP addresses that are allowed to authenticate to Crowd. If FishEye is remote to Crowd, add the IP address of your FishEye server and ensure the "Status" field is set to "true". Remove the entry for `localhost`.

- If FishEye is on the same host as Crowd:
  By default, when you add an application, `localhost` is a permissible foreign host. However, you will also need to manually add the IP address `127.0.0.1`, as incoming requests to Crowd from FishEye (both on the same, local, host) may be from the host `127.0.0.1` and not `localhost`. Crowd does not do a DNS lookup of the hostname. Rather, it compares the values as is. Ensure the "Status" field is set to "true".

Step 2. Configuring FishEye to Talk to Crowd

The instructions below are for FishEye 1.4.x and later. If you are using FishEye 1.3.x, please follow the guide for earlier versions of FishEye.

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-in' to 'crowd'. This is required for the new authorisation 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 screens and navigate to 'Security'.
2. Select 'Setup 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 the 'Remove' link 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/`
     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.

For more information, please see the Fisheye documentation on configuring external authentication sources.

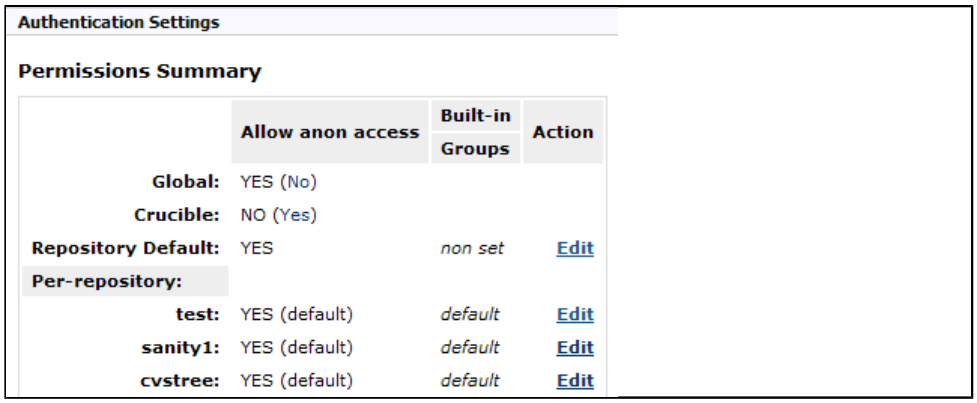
2.3 Configure Group Authorisation in FishEye (If Required)

If you have created groups in the Crowd directory which is mapped to your FishEye application (see Step 1 above), the Crowd groups can be seen in FishEye. Now you can set up group authorisation for your FishEye repositories.

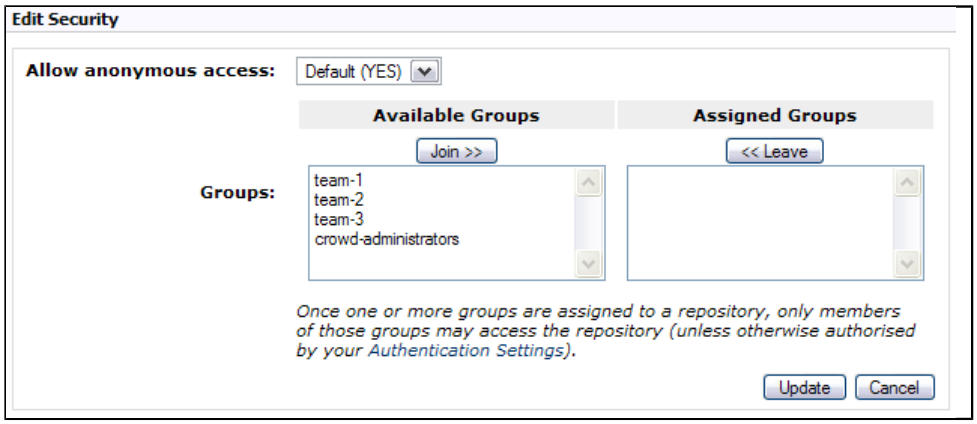
Allow the groups to access your FishEye repositories as follows:

1. In the FishEye Administration menu, select 'Security' under 'Global Settings'.
2. This will display the 'Authentication Settings' screen. In the 'Permissions Summary' section, click 'Edit' next to the required repository name under 'Per-repository'.
3. The 'Edit Security' screen will appear. Select the group name(s) and click the 'Join' button. Click 'Update'. The group(s) will appear in the 'Built-in Groups' section of the 'Authentication Settings' screen.

Screenshot 1: 'Authentication Settings'



Screenshot 2: 'Edit Security'



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
- Mapping a Directory to an Application
- Specifying which Groups can access an Application
- Specifying an Application's Address or Hostname
- Testing a User's Login to an Application
- Managing an Application's Session
- Deleting or Deactivating an Application

Crowd Documentation

Configuring FishEye 1.3.x to talk to Crowd

This page last changed on May 05, 2008 by smaddox.

This page forms part of the guide on Integrating Crowd with Atlassian FishEye and Crucible.

Use the instructions below if you are integrating Crowd with FishEye version 1.3.x. If you are using FishEye 1.4.x or later, refer to the instructions for later versions of FishEye.

Step 1. Configuring Crowd to talk to FishEye

Please complete Step 1 in the full Crowd/FishEye integration instructions.

Step 2. Configuring FishEye to talk to Crowd

Before you begin
For any usernames that are already configured through the Fisheye Administration console, you will need to change the account type from 'built-in' to 'custom'. This is required for the new authorisation through Crowd to work properly.

For details please see the Fisheye documentation.

2.1 Install the Crowd Client Libraries into FishEye

Copy the Crowd integration libraries and configuration files as described in Integrating Crowd with a Custom Application. This involves copying all client library JARs to the library folder of FishEye:

The version numbers have been omitted. Select the JAR which matches the name. This listing has been verified with FishEye 1.3.1.

Files to Copy	Destination
CROWD/client/crowd-integration-client-X.X.X.jar	$FISHEYE_INST/lib
CROWD/client/lib/commons-codec-1.3.jar	$FISHEYE_INST/lib
CROWD/client/lib/commons-httpclient-3.0.jar	$FISHEYE_INST/lib
CROWD/client/lib/commons-lang-2.3.jar	$FISHEYE_INST/lib
CROWD/client/lib/jdom-1.0.jar	$FISHEYE_INST/lib
CROWD/client/lib/stax-api-1.0.1.jar	$FISHEYE_INST/lib
CROWD/client/lib/wsdl4j-1.6.1.jar	$FISHEYE_INST/lib
CROWD/client/lib/wstx-asl-3.2.0.jar	$FISHEYE_INST/lib
CROWD/client/lib/xfire-core-1.2.6.jar	$FISHEYE_INST/lib

2.2 Configure FishEye to use Crowd's Authenticator

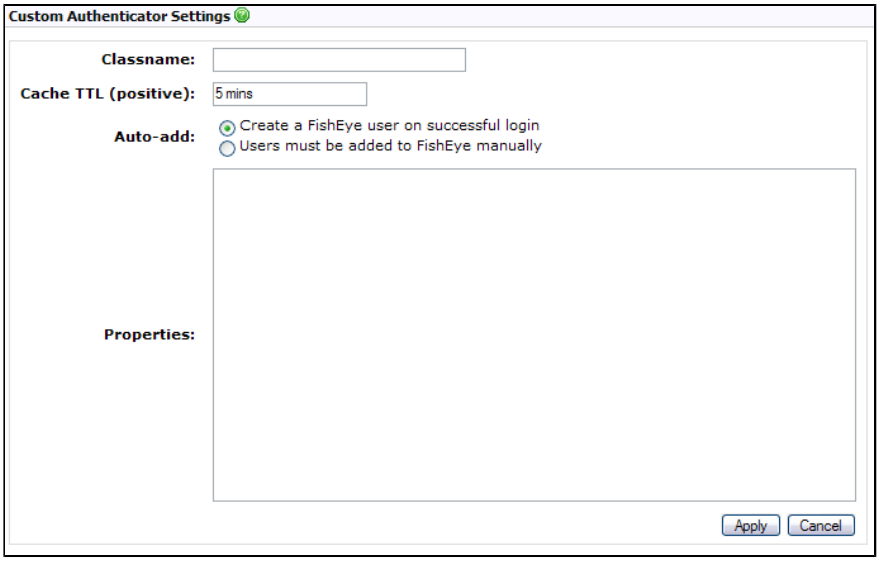
1. Log in as an administrator to FishEye and navigate to 'Users/Security'. Select 'Setup Custom authentication'.
   - Enter the following 'Classname' for the authenticator:

     ```
     com.atlassian.crowd.integration.fisheye.FisheyeAuthenticator
     ```

     Leave the cache and auto-add settings at their default values. This will mean authentication calls to Crowd will be cached (improves performance) and that users will be automatically enrolled into FishEye after their initial login to FishEye via Crowd.
   - Fisheye requires you to pass in the configuration attributes for Crowd. Add the following information in the 'Properties' text box, replacing the information with your own configuration data – match the values set in Step 1.

     ```
     application.name            fisheye
     application.password        password
     application.login.url       http://localhost:8080/
     crowd.server.url            http://localhost:8095/crowd/services/
     session.isauthenticated     session.isauthenticated
     session.tokenkey            session.tokenkey
     ```

```
session.validationinterval  0
session.lastvalidation      session.lastvalidation
```



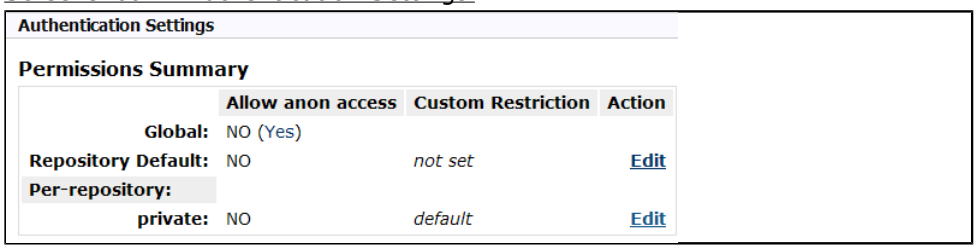
Refer to the FishEye documentation for further details on using the FishEye setup screens.

2.3 Configure Groups for FishEye Source Repositories (If Required)

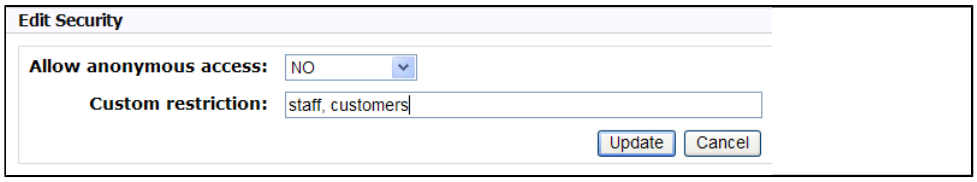
If you are using any FishEye groups to control access to particular source repositories, you will need to create the groups in Crowd and then configure FishEye as follows:

1. In the FishEye Administration menu, select 'Global Settings', then 'Users/Security'.
2. This will display the 'Authentication Settings' screen. In the 'Permissions Summary' section, edit the 'Per-repository' field and enter the group names (separated by commas) in the 'Custom restriction' field.

Screenshot 1: 'Authentication Settings'



Screenshot 2: 'Custom Restriction'



Related Topics

Unable to render {children} Page not found: 3. Managing Applications

Crowd Documentation

Integrating Crowd with Atlassian JIRA

This page last changed on May 05, 2008 by smaddox.

Atlassian's popular JIRA issue management system takes advantage of the OSUser framework and can quickly be configured to use OSUser to link in single or multiple directory servers through Crowd. Crowd provides integration libraries for the OpenSymphony OSUser module, which has a simple-to-use API for user-management that allows pluggable implementations. More about the OSUser API can be reviewed at http://www.opensymphony.com/osuser/.

Currently Crowd supports centralised authentication and single sign-on for JIRA versions 3.7.4 and later.

If you are using NTLM for Windows authentication, you may want to read about configuring Crowd's JIRA NTLM plugin for single sign-on.

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 JIRA (version 3.7.4 or later). Refer to the JIRA installation guide for detailed information on how to do this. We will refer to the JIRA root folder as `JIRA`. For the purposes of this document, we will assume that the 'standalone' (i.e. the easier and recommended) installation method of JIRA has been used. If you need to install JIRA as an EAR/WAR, simply explode the EAR/WAR and make the necessary changes as described below, and repackage the EAR/WAR.
3. Make sure JIRA is not running when you begin the integration process described below.

Step 1. Configuring Crowd to talk to JIRA

1.1 Prepare Crowd's Directories/Groups/Users for JIRA

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. For information on how to do this, see Adding a Directory. We will assume that the directory is called JIRA 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 JIRA Directory to house JIRA users.

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:

1. `jira-users`
2. `jira-developers`
3. `jira-administrators`

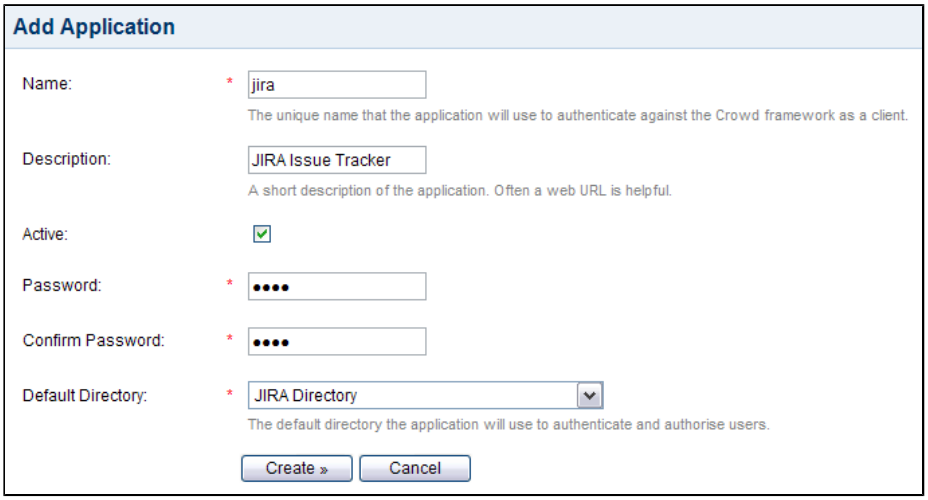
You also need to ensure that the JIRA Directory contains at least one user who is a member of all three groups. You can either:

- 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 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 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.

1.2 Define the JIRA Application in Crowd

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.

1. Log in to the Crowd Administration Console and navigate to Applications > Add Application.
2. Fill out the form to add the JIRA application:

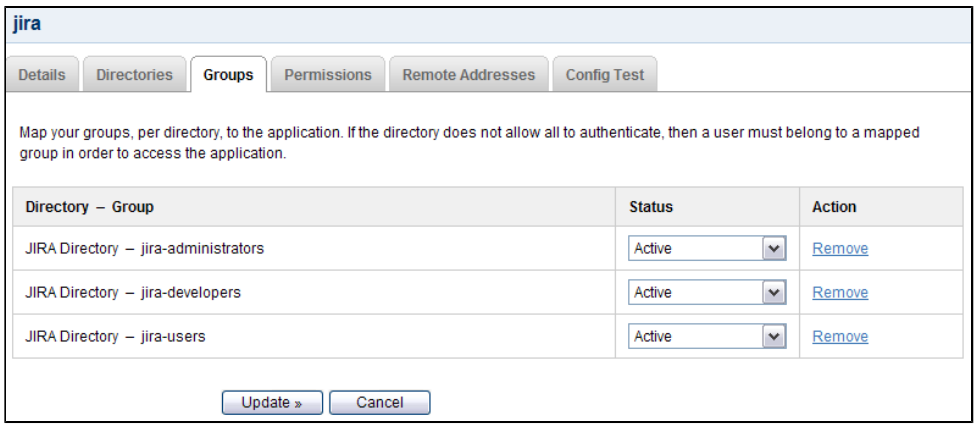


Attribute	Description
Name	The username which the application will use when it authenticates against the Crowd framework as a client. This value must be unique, i.e. it cannot be used by more than one application client.
Description	A short description of the application. Note: A web URL is often helpful.
Active	Only deselect this if you wish to prevent all users (from all directories) from accessing this application.
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.
Default Directory	A directory that contains relevant users. Note: Additional directories can be added later.

The Name and Password values must match the application.name and application.password that you set in the `JIRA/atlassian-jira/WEB-INF/classes/crowd.properties` (see Step 2 below).

1.3 Specify which users can log in to JIRA

Now that Crowd is aware of the JIRA application, Crowd needs to know which directories or users can authenticate (log in) via Crowd. 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 to authenticate:



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

Please see Specifying an Application's Address or Hostname. Please note:

- If JIRA is on a different host to Crowd
  If you are running the JIRA on a different host to Crowd, you will need to modify the permissible hosts via the Remote Addresses tab. This lists the hosts/IP addresses that are allowed to authenticate to Crowd. If JIRA is remote to Crowd, add the IP address of your JIRA server and ensure the "Status" field is set to "true". Remove the entry for `localhost`.
- If JIRA is on the same host as Crowd
  By default, when you add an application, `localhost` is a permissible foreign host. However, you will also need to manually add the IP address `127.0.0.1`, as incoming requests to Crowd from JIRA (both on the same, local, host) may be from the host `127.0.0.1` and not `localhost`. Crowd does not do a DNS lookup of the hostname, rather, it compares the values as is. Ensure the "Status" field is set to "true".

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 standalone application, which is an exploded WAR stored in `JIRA/atlassian-jira`.

1. 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

   JIRA 3.11 and earlier are incompatible with Crowd 1.2.
   You will need to remove the following file from JIRA's `WEB-INF/lib/seraph-0.7.12.jar` directory and replace it with the following:
   http://repository.atlassian.com/maven2/com/atlassian/seraph/atlassian-seraph/0.10/atlassian-seraph-0.10.jar

2. Edit `JIRA/atlassian-jira/WEB-INF/classes/crowd.properties`. Change the following properties:

Key	Value
application.name	jira
application.password	set a password
crowd.server.url	http://localhost:8095/crowd/services/
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.

If your Crowd server's port is configured differently from the default (i.e. 8095), set it accordingly. 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).

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.

This applies to the Crowd Administration Console's `crowd.properties` file too. You may find this particularly useful when integrating with a WAR deployment of an integrated application.

Example:

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

2.2 Configure JIRA to use Crowd's Authenticator

Now that the Crowd client libraries exist, we need to configure JIRA to use 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>
```

2. View `JIRA/atlassian-jira/WEB-INF/classes/propertyset.xml`. If an entry doesn't exists for the CrowdPropertySet, to add the following <propertyset> at the end of the file as the last <propertyset>:

```
<propertyset name="crowd" class="com.atlassian.crowd.integration.osuser.CrowdPropertySet"/>
```

3. At this stage, JIRA is set up for centralised authentication. If you wish, you can now enable single sign-on (SSO) to JIRA.

   Skip this step if you are using the JIRA NTLM plugin to enable SSO. Instead, follow the instructions on configuring JIRA for NTLM SSO.
   Edit `JIRA/atlassian-jira/WEB-INF/classes/seraph-config.xml`. Change the `authenticator` node to read:

```
<authenticator class="com.atlassian.crowd.integration.seraph.JIRAAuthenticator"/>
```

JIRA's authentication and access request calls will now be performed using Seraph. Now when authentication or access request calls are performed versus the OSUser framework, the JIRA stack will call the Crowd providers and propertyset implementations.

2.3 Enable JIRA's 'External User Management'

Once the setup is complete, you can configure JIRA to allow external user management. Go to the JIRA Administration Console. In the General Configuration section, turn 'External user management' and 'External password management' on or off. (See the JIRA Administrator's Guide for details).

JIRA with external user management ON:

This is recommended, because it allows you to use Crowd's powerful cross-directory user administration features.

If you turn external user management on, the following functions can no longer be performed from within the JIRA administration interface: adding users, adding groups, editing users, editing groups.

 If you are using Crowd 1.1.1 or earlier, you must turn external user management on in JIRA.

JIRA with external user management OFF:

This means that you can allow signup via JIRA, and you can manage your users within JIRA. Changes will flow through to Crowd.

JIRA has an automatic group membership feature. This means that any new user added through JIRA will automatically be a member of all groups which have the JIRA Users permission. In this way, you can ensure that a new user is automatically added to several groups when they sign up with JIRA.

 Any group or user changes will cascade to all directories assigned to the JIRA application in Crowd. For example, if user 'jbloggs' registers in JIRA, 'jbloggs' will be added to every Crowd directory linked with the JIRA application.

2.4 (Optional) Tune the Cache

When utilising the atlassian-user and Crowd framework together with JIRA, 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.)

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 in Crowd to 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).

 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).

2.5 (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 centralised authentication in action!
- If you have enabled SSO, you can try adding the JIRA Directory 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

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`.

The current workaround for this is to deactivate the user's account (by removing them from the `jira-users` group). This issue can be tracked here: http://jira.atlassian.com/browse/CWD-202

RELATED TOPICS

- Using the Application Browser
- Adding an Application
  - Integrating Crowd with Atlassian Bamboo
  - Integrating Crowd with Atlassian Confluence
    - Configuring Confluence for NTLM SSO
  - Integrating Crowd with Atlassian CrowdID
  - Integrating Crowd with Atlassian Crucible
  - Integrating Crowd with Atlassian FishEye
    - Configuring FishEye 1.3.x to talk to Crowd
  - Integrating Crowd with Atlassian JIRA
    - Configuring JIRA for NTLM SSO
  - Integrating Crowd with Acegi Security
    - Integrating AppFuse - a Crowd-Acegi Integration Tutorial
  - Integrating Crowd with Apache
  - Integrating Crowd with Jive Forums
    - Jive SSO
  - Integrating Crowd with Subversion
  - Integrating Crowd with a Custom Application
- Mapping a Directory to an Application
  - Specifying the Directory Order for an Application
  - Specifying an Application's Directory Permissions
    - Example of Directory Permissions
- Specifying which Groups can access an Application
- Specifying an Application's Address or Hostname
- Testing a User's Login to an Application
- Managing an Application's Session
- Deleting or Deactivating an Application

Crowd Documentation

Configuring JIRA for NTLM SSO

This page last changed on May 05, 2008 by smaddox.

JIRA NTLM plugin not officially supported by Atlassian

The JIRA NTLM plugin was written by a third party. Atlassian does not officially support the plugin. The Atlassian Crowd team will do our best to advise on any Crowd integration problems. Please refer to the plugin documentation for installation instructions and further support.

Out of the box, JIRA does not support Single Sign On (SSO) functionality. This page describes how to set up JIRA with NTLM SSO functionality using the JIRA NTLM plugin, Crowd, and Active Directory (AD) as your LDAP user repository.

Summary

The JIRA NTLM plugin enables the following authentication scenario:

- A user in a Windows domain logs into the Windows network, using their Active Directory username/password.
- Then, when they open JIRA in an Internet Explorer browser, they are seamlessly logged into JIRA.

The Crowd component then allows you to manage all users and groups in Active Directory. Crowd automatically ensures that users and groups are synchronised between AD and JIRA. For example, if a user/group is added/deleted from AD it will be automatically added/deleted from JIRA.

Components

JIRA NTLM plugin	NTLM is the protocol used by Windows for authentication. The JIRA NTLM plugin takes care of the Windows domain / Active Directory login to JIRA. You must be running a Windows Domain Controller with accounts set up in AD in order to use this plugin. If NTLM authentication is not available, the plugin allows standard form-based login to JIRA. Note: This plugin is not officially supported by Atlassian.
Crowd	Crowd takes care of the synchronisation of users/groups between Active Directory and JIRA. You will need to create an SSL connection between Crowd and the AD server if you would like to create users through Crowd. AD will not allow Crowd to add users or change their passwords unless the communication occurs over a secure connection.
Active Directory (AD) on Windows 2003 Server	Active Directory (AD) on Windows 2003 Server — you must already have an AD instance set up and running with a domain controller.
JIRA	The machine running JIRA must be part of the Windows domain or installed on the same box as the domain controller.

Steps

1. Back up your entire JIRA installation directory and run an XML backup of your data.
2. Download the JIRA NTLM plugin.
3. Read the `README` file included in the plugin zip file, and then follow the instructions in the `INSTALL` file to install the plugin.
4. In the `ntlm_ldap.properties` file, insert the appropriate LDAP and Domain Controller information along with other parameters.
5. Install and configure Crowd.
6. Create a directory in Crowd for the AD LDAP server.

7. Create the JIRA application in Crowd and configure Crowd and JIRA to talk to each other, as described in Integrating Crowd with Atlassian JIRA.

   
   When following the above instructions, do not change the `seraph-config.xml` file to enable Crowd's SSO functionality. (I.e. don't change the `authenticator` node to read `<authenticator class="com.atlassian.crowd.integration.seraph.JIRAAuthenticator"/>`. Instead of Crowd's SSO authentication, we'll be using the JIRA NTLM plugin.
8. In AD, create the groups jira-users, jira-developers, and jira-administrators. They should then appear in Crowd.
9. In AD, create an admin user and make them a member of the above three groups in AD.
10. Create any additional groups that you would like in AD.
11. Log into the Windows domain using your desktop login and then open JIRA in an Internet Explorer browser. You should be logged in automatically.

Additional Crowd Performance Tips

- Change the default cache setting timeout in the file <JIRA>\WEB-INF\classes\crowd-ehcache.xml. For performance reasons, increase the object caching to 7,200 seconds (2 hours): timeToIdleSeconds="7200" timeToLiveSeconds="7200".
  This reduces the frequency of the requests from Crowd to the LDAP server when changes to LDAP objects (such as a group name or user attribute) are made, thus reducing the performance overhead.

- Turn on the 'Use Paged Results' option in the directory connector tab for the directory you've set up in Crowd.

Integrating Crowd with Acegi Security

This page last changed on May 05, 2008 by smaddox.

Crowd provides centralised authentication and single sign-on connectors for the web security framework Acegi. Acegi provides a modular and highly configurable approach to authentication and authorisation for J2EE applications.

The connectors are available with Crowd 1.2 and later and have been developed and tested with Acegi 1.0.5.

Please consult the Acegi quick start guide or reference guide for a thorough insight into the Acegi framework. You might also find useful information in our Crowd-Acegi integration tutorial.

This guide assumes developer-level knowledge

This guide assumes you have Crowd 1.3 or later installed and that you want to integrate your Acegi-based web application with Crowd's security server. This guide is more for developers than administrators.

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 Acegi-based custom application ready for tweaking. We will refer to your custom application as 'AcegiApp'.

Step 1. Configuring Crowd to Talk to your Acegi Application

Crowd needs to be aware that AcegiApp will be making authentication requests to Crowd. In brief, you will need to do the following:

1. Add the AcegiApp application to Crowd.
2. Add and configure the directories visible to AcegiApp.
3. Add and map the groups which are allowed to authenticate with AcegiApp.

Please see Adding an Application for a detailed guide.

Step 2. Installing the Crowd Acegi Connector

2.1 Adding the Crowd Acegi Connector to your Acegi Application

You will need to add the Crowd Acegi connector library and its associated dependencies to your Acegi application. You can do this manually by copying over the JAR files to your Acegi application or, if your Acegi application is a Maven project, you can add the Crowd Acegi connector as a project dependency.

2.1.1 Manually Adding the Crowd Acegi 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 at least copy across the following file to your Acegi application:

Copy From	Copy To
CROWD/client/crowd-integration-client-X.X.X.jar	AcegiApp/WEB-INF/lib
CROWD/client/lib/*.jar	AcegiApp/WEB-INF/lib

2.1.2 Adding the Crowd Acegi Connector as a Maven Dependency

 Follow either 2.1.1 or 2.1.2 (not both).

To integrate Crowd with your Maven 2 project, you will need to include the following dependency in your `pom.xml`:

```
<dependency>
        <groupId>com.atlassian.crowd</groupId>
        <artifactId>crowd-integration-client</artifactId>
        <version>X.X</version>
        <type>pom</type>
</dependency>
```

Because the Crowd libraries are not published to the standard Maven repository, you will need to add Atlassian's public repository:

```
<repositories>
        <repository>
                <id>central</id>
                <url>https://m2proxy.atlassian.com/repository/public</url>
                <snapshots>
                        <enabled>true</enabled>
                        <updatePolicy>always</updatePolicy>
                </snapshots>
                <releases>
                        <enabled>true</enabled>
                </releases>
        </repository>
</repositories>
```

See more information on Maven 2 integration.

2.2 Configuring the Crowd Acegi Connector Properties

The Crowd Acegi connector needs to be configured with the details of the Crowd server.

1. Copy the default `crowd.properties` file to the classpath of your Acegi application:

Copy From	Copy To
CROWD/client/conf/crowd.properties	AcegiApp/WEB-INF/classes

2. Edit the `crowd.properties` and 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.validationinterval	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.

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.

This applies to the Crowd Administration Console's `crowd.properties` file too. You may find this particularly useful when integrating with a WAR deployment of an integrated application.

Example:

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

Step 3. Configuring your Acegi Application to Use the Crowd Acegi Connector

There are two ways you can integrate your application with Crowd:

- Centralised 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 centralised user repository for retrieving user details as well as performing authentication.
- Single sign-on: In addition to centralised 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 behaviour 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 the `applicationContext-CrowdClient.xml` Spring configuration file, found in `crowd-integration-client.jar`. For example, if you are configuring Spring using a context listener, you can add the following parameter in your `WEB-INF/web.xml`:

```
<context-param>
        <param-name>contextConfigLocation</param-name>
        <param-value>
            ...
            classpath:/applicationContext-CrowdClient.xml
            ...
        </param-value>
    </context-param>
```

Next, open the `applicationContext.xml` file relevant to your application, which contains the Acegi configuration. You are likely to have a bean configuration similar to this snippet:

```
<bean id="filterChainProxy" class="org.acegisecurity.util.FilterChainProxy">
  <property name="filterInvocationDefinitionSource">
    <value>
      CONVERT_URL_TO_LOWERCASE_BEFORE_COMPARISON
      PATTERN_TYPE_APACHE_ANT
      /images/**=#NONE#
      /scripts/**=#NONE#
      /styles/**=#NONE#
/**=httpSessionContextIntegrationFilter,logoutFilter,authenticationProcessingFilter,securityContextHolderAwareR
    </value>
  </property>
</bean>
```

3.1 Configuring Centralised User Management

Perform the following updates to your Acegi Spring configuration:

1. Add the definition of the CrowdUserDetailsService:

```
<bean id="crowdUserDetailsService"
 class="com.atlassian.crowd.integration.acegi.CrowdUserDetailsService">
        <property name="securityServerClient" ref="securityServerClient"/>
    </bean>
```

2. Add the definition of the CrowdAuthenticationProvider:

```
<bean id="crowdAuthenticationProvider"
 class="com.atlassian.crowd.integration.acegi.CrowdAuthenticationProvider">
        <property name="userDetailsService" ref="crowdUserDetailsService"/>
        <property name="httpAuthenticator" ref="httpAuthenticator"/>
        <property name="securityServerClient" ref="securityServerClient"/>
    </bean>
```

3. Update the definition of your AuthenticationManager / ProviderManager to use the CrowdAuthenticationProvider. If you need multiple authentication providers, you can append the CrowdAuthenticationProvider to your list.

```
<bean id="authenticationManager" class="org.acegisecurity.providers.ProviderManager">
  <property name="providers">
    <list>
      <ref local="crowdAuthenticationProvider"/>
      ...
    </list>
  </property>
</bean>
```

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 centralised user management

Single sign-on is optional. If you wish to configure SSO you must first configure centralised user management as described in step 3.1 above.

Perform the following additional updates to your Acegi Spring configuration:

1. Update the definition of the AuthenticationProcessingFilter to use the CrowdAuthenticationProcessingFilter:

```
<bean id="authenticationProcessingFilter"
 class="com.atlassian.crowd.integration.acegi.CrowdAuthenticationProcessingFilter">
        <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"/>
        ...
    </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:

```
<bean id="logoutFilter" class="org.acegisecurity.ui.logout.LogoutFilter">
        <constructor-arg value="/index.jsp"/>
        <constructor-arg>
            <list>
                ...
                <ref bean="crowdLogoutHandler"/>
                <bean class="org.acegisecurity.ui.logout.SecurityContextLogoutHandler"/>
            </list>
        </constructor-arg>
        <property name="filterProcessesUrl" value="/logout.jsp"/>
    </bean>
```

Step 4. Restarting your Acegi Application

Bounce your application. You should now have centralised authentication and single sign-on with Crowd.

Authorisation

For the purposes of Crowd integration with Acegi, you should map Acegi's roles to Crowd's groups. To put it another way: in order to use Acegi's authorisation features, users in Crowd will have their Acegi roles specified by their group names.

For example if user 'admin' is in the 'crowd-admin' group, then the user 'admin' will be authorised to view pages restricted to the 'crowd-admin' role in Acegi.

RELATED TOPICS

- Using the Application Browser
- Adding an Application
- Mapping a Directory to an Application
- Specifying which Groups can access an Application
- Specifying an Application's Address or Hostname
- Testing a User's Login to an Application
- Managing an Application's Session
- Deleting or Deactivating an Application

Crowd Documentation

Integrating AppFuse - a Crowd-Acegi Integration Tutorial

This page last changed on May 05, 2008 by smaddox.

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 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.

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=true
```

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>1.3</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.

Step 4. Hook Up Centralised 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 centralised 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.CrowdUserDetailsService">
       <property name="securityServerClient" ref="securityServerClient"/>
   </bean>
   ```

2. Add the definition of the CrowdAuthenticationProvider which will delegate Acegi's authentication requests to Crowd:

   ```
   <bean id="crowdAuthenticationProvider"
    class="com.atlassian.crowd.integration.acegi.CrowdAuthenticationProvider">
       <property name="userDetailsService" ref="crowdUserDetailsService"/>
       <property name="httpAuthenticator" ref="httpAuthenticator"/>
       <property name="securityServerClient" ref="securityServerClient"/>
   </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=true
```

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 `ROLE_USER` and `ROLE_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 authorisation 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 centralised authentication 

Application-level centralised 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 centralised user management, AppFuse will need to delegate its calls to create, retrieve, update and delete users to Crowd via Crowd's remote API. This will prevent data redundancy and eliminate the hassle of data synchronisation. 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.CrowdAuthenticationProcessingFilter">
    <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,securityContextHolderAw
        </value>
    ...
```

5. Now repeat:

```
mvn jetty:run-war -Dmaven.test.skip=true
```

SSO will only work for users that are able to authenticate with both appplications and are authorised 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 Apache

This page last changed on May 05, 2008 by smaddox.

Crowd provides a number of modules that allow you to configure Crowd to authenticate HTTP Authentication requests made to an Apache web server.

The following features are provided:

- Use Crowd to password-protect resources on your website.
- Configure website locations to restrict access to specific Crowd groups or users.

Note: These instructions assume some Unix system and Apache configuration knowledge.

Using Subversion under Apache?

Crowd's Subversion connector allows you to password-protect a Subversion repository and provide fine grained access by group or user. Read more.

Prerequisites

- Apache web server version 2.0 or above with the `mod_perl` module installed and configured.
- The following third-party Perl modules:
  - `SOAP::Lite` (v0.69 or greater required)
  - `Digest::SHA1`
  - `Error`
  - `Cache::Cache`

Installation and Configuration

The following instructions are for Unix systems. If you're running Apache on Windows, see the notes below.

Installing the Third-Party Perl Modules

Download the required Perl modules from CPAN using the links above and install them as follows:

```
tar xvzf  Cache-Cache-1.05.tar.gz
cd Cache-Cache-1.05
perl Makefile.PL
make
make install
```

See http://search.cpan.org/~jhi/perl-5.8.0/pod/perlmodinstall.pod for a detailed description of the various ways of installing Perl modules on your system.

Installing the Crowd Perl Modules

1. Download the three Crowd module files attached to this page:

Attached file	Description
Crowd-Apache-Connector-1.2.2.zip	Crowd authentication, authorisation and perl module for Apache 2

2. Extract the Crowd-Apache-Connection archive file and install the three modules using the same procedure as for the third party modules.

```
unzip Crowd-Apache-Connector-1.2.2.zip
cd Atlassian-Crowd-1.2.2/
perl Makefile.PL
make
make install
cd ../Apache-CrowdAuth-1.2.2/
perl Makefile.PL
```

```
make
make install
cd ../Apache-CrowdAuthz-1.2.2/
perl Makefile.PL
make
make install
```

Configuring Apache

Ensure that `mod_perl` is enabled. Your Apache config file should contain a line like the following:

```
LoadModule perl_module modules/mod_perl.so
```

Many common distributions of Apache come with `mod_perl` preconfigured.

Configuring Authentication

To tell Apache to use Crowd to authenticate requests for a particular location, edit the Apache config file to add the following commands to a `<Location>` or `<Directory>` section.

```
Alias /crowd/ "/var/crowd/"
<Directory "/var/crowd/">
.
.
.
  AuthName crowd
  AuthType Basic

  PerlAuthenHandler Apache::CrowdAuth
  PerlSetVar CrowdAppName appname
  PerlSetVar CrowdAppPassword apppassword
  PerlSetVar CrowdSOAPURL http://localhost:8095/crowd/services/SecurityServer
  PerlSetVar CrowdCacheEnabled on
  PerlSetVar CrowdCacheLocation /tmp/CrowdAuth
  PerlSetVar CrowdCacheExpiry 300

  require valid-user
.
.
.
</Directory>
```

Command	Explanation
`AuthName crowd`	Defines the realm of the authentication. This information is typically provided to the user in the dialog box popped up by their browser
`AuthType Basic`	Tells apache to use basic authentication
`PerlAuthenHandler Apache::CrowdAuth`	Tells Apache to delegate authentication to the CrowdAuth module
`PerlSetVar CrowdAppName`	Set the Application Apache should authenticate as
`PerlSetVar CrowdAppPassword`	Set the password for the Application
`PerlSetVar CrowdSOAPURL`	The URL of the Crowd SOAP service
`PerlSetVar CrowdCacheEnabled`	[optional] Controls whether CrowdAuth caches authentications locally to improve performance. Set to "on" or "off". Caching is "on" by default
`PerlSetVar CrowdCacheLocation`	[optional] The directory in which CrowdAuth's local cache is stored. Defaults to `/tmp/FileCache` if not set.
`PerlSetVar CrowdCacheExpiry`	[optional] The time (in seconds) before cached authentications in CrowdAuth's local cache expire. Defaults to 300 seconds (5 minutes)

`require valid-user`	Tells Apache that clients must provide a valid username/password to access the location

Configuring Authorisation

If you want to restrict access to a certain Directory or Location in your Apache configuration to a subset of Crowd users and/or groups, add the following lines to your configuration:

```
<Location>
  .
  .
  .
  PerlAuthzHandler Apache::CrowdAuthz

  PerlSetVar CrowdAllowedUsers johnh,kevinr
  PerlSetVar CrowdAllowedGroups developers,crowd-administrators,customers:r
  .
  .
  .
</Location>
```

Command	Explanation
`PerlAuthzHandler Apache::CrowdAuthz`	Tells Apache to use the `Apache::CrowdAuthz}` for authorisation
`PerlSetVar CrowdAllowedUsers johnh,kevinr`	Allow only the users `johnh` or `kevinr` to access the location
`PerlSetVar CrowdAllowedGroups developers,crowd-administrators,customers:r`	Allow only members of the `developers` or `crowd-administrators` groups to access the location. You can indicate that a group has read-only access to the location by appending ":r" to the group name. This is mainly useful for giving a group read-only access to a Subversion repository

Note:

- Typically, only one of the `CrowdAllowedUsers` or `CrowdAllowedGroups` would be needed for a particular location. You can define both. If you do, then access is granted if either is satisfied.
- If the `CrowdCacheEnabled` setting is on, then authorisation checks are cached in order to increase performance. This means that changes to group membership in Crowd may not be reflected immediately in user access.

Troubleshooting

The `CrowdAuth` module logs detailed output if the Apache LogLevel parameter is set to `info` or `debug`. This can be useful in diagnosing problems.

Apache Log Error Message	Possible Cause and Next Steps
`CrowdAppName or CrowdAppPassword is not defined`	One or both of the `CrowdAppName` or `CrowdAppPassword` parameters is missing from the Apache config file.
`Failed to authenticate application`	The attempt to authenticate the application with Crowd failed. Check the values of the `CrowdAppName` or `CrowdAppPassword` parameters.
`Failed to authenticate principal`	Failed to authenticate a username/password pair provided by the client. This may just mean that the username or password supplied is incorrect. Note that `CrowdAuth` won't log successful authentications unless the `LogLevel` is `info` or above.
`User token not found in SOAP response for user <user>`	Internal SOAP protocol error.
`error 500...at CrowdAuth.pm..`	Indicates that Apache can't connect to the Crowd SOAP service.

`error 404...at CrowdAuth.pm...`	Indicates that the URL used to connect to the Crowd SOAP service is incorrect. Check the value of the `CrowdSOAPURL` parameter.
`failed to resolve handler `Apache::CrowdAuth': Can't locate Apache/CrowdAuth.pm ...`	The `CrowdAuth.pm` file isn't located on the Perl include path (or it has incorrect permissions).
`failed to resolve handler `Apache::CrowdAuth': Can't locate SOAP/Lite.pm...`	The `SOAP:Lite` module hasn't been installed.
`Can't locate object method "call" via package "SOAP::SOM" at ...`	This message indicates a missing or old installation of SOAP::Lite. Try installing (or reinstalling) version 0.69 SOAP:Lite. On Windows, you will get this error if you haven't manually upgraded the SOAP::Lite ppm (see below)

Installing Perl, **`mod_perl`** and Perl Modules on Windows

Setting up `CrowdAuth` on an Apache instance running on Windows requires that some things be done differently. The following instructions assume you are using ActivePerl as your Perl environment.

- If you don't already have a Perl interpreter installed, you'll need one. The following instructions assume an install of ActiveState's ActivePerl. We strongly recommend running version 5.8.8 of ActivePerl rather than any newer version.
- Windows installations of Apache are less likely to come with `mod_perl` pre-installed. A Win32 version of `mod_perl` in PPM format is available here.
- The `.tar.gz` format used to distribute CrowdAuth (and other modules) is supported by most modern Windows archiving utilities (WinZip, for example).
- The `make` utility used to build the Perl modules is not part of a Windows. `nmake`, Microsoft's equivalent, is available as a self-extracting archive here.

Installing Perl Modules on Windows

The required modules (`Digest::SHA1`, `Error`, `Cache::FileCache`, `SOAP:Lite`) are available through the Perl Package Manager utility.

`CrowdAuth` needs a newer version of SOAP::Lite than is supplied with ActivePerl (0.69 vs 0.55). A prebuilt ppm of the correct version can be installed from the University of Winnipeg's repository using the following command:

```
C:\>ppm install http://theoryx5.uwinnipeg.ca/ppms/SOAP-Lite.ppd
```

Installing **`Apache::CrowdAuth`** on Windows

```
Extract Apache-CrowdAuth-0.06.zip using Winzip or equivalent...
cd Apache-CrowdAuth-0.06
perl Makefile.PL
nmake
nmake install
```

When editing the `httpd.conf` file and adding the `mod_perl.so` module to Apache, you may need to add the following line above the LoadModule line:

```
LoadFile "C:/Perl/bin/perl58.dll"
LoadModule perl_module modules/mod_perl.so
```

This LoadFile line points to the `perl58.dll` in your Perl install directory.

Related Topics

- Using the Application Browser
- Adding an Application
  - Integrating Crowd with Atlassian Bamboo
  - Integrating Crowd with Atlassian Confluence
    - Configuring Confluence for NTLM SSO
  - Integrating Crowd with Atlassian CrowdID

Integrating Crowd with Jive Forums

This page last changed on May 07, 2008 by smaddox.

Jive Forums offers you the ability to specify an implementation to provide authentication and authorisation external to the application. This document outlines how to integrate Crowd's authenticator with Jive Forums.

Crowd provides centralised authentication and single sign-on (SSO) for Jive Forums version 5.5.13.

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

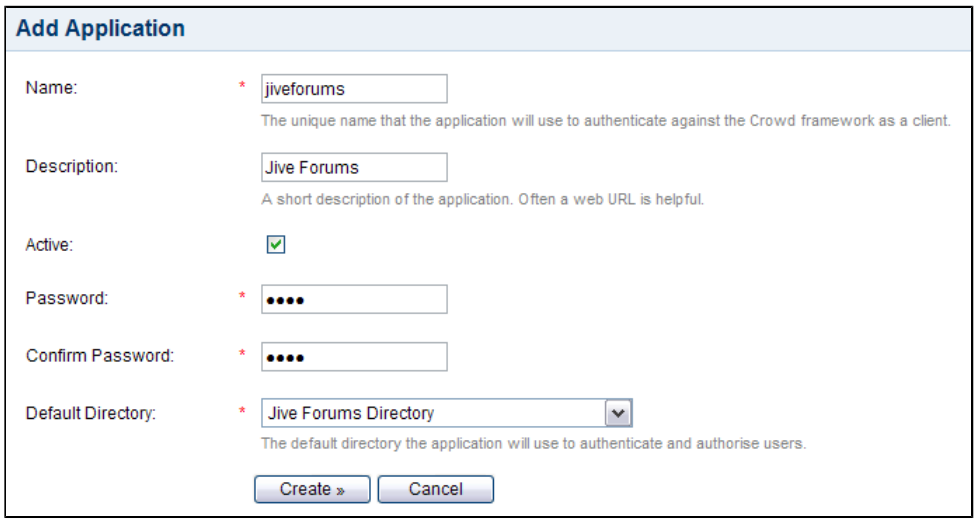
The Jive Forums application will need to locate users from a directory configured in Crowd. You will need to set up a directory in Crowd for Jive. For more information on how to do this, see Adding a Directory. We will assume that the directory is called Jive Forum Directory for the rest of this document. 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. Fill out the form to add the Jive Forums application:



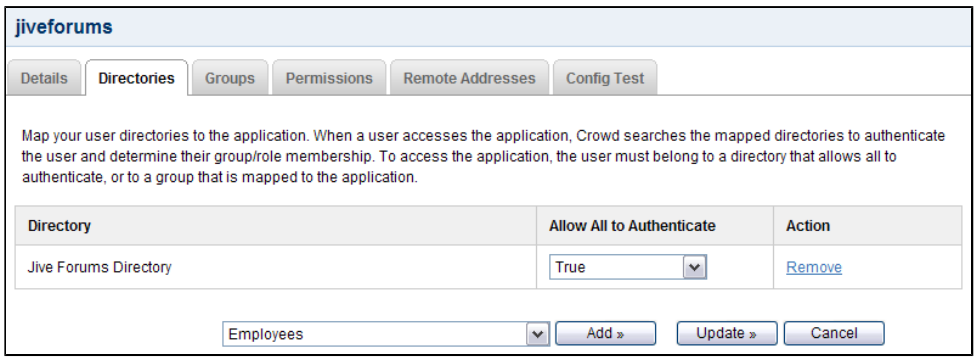
Attribute	Description
Name	

	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 web URL is often helpful.
Active	Only deselect this if you wish to prevent all users (from all directories) from accessing this application.
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.
Default Directory	A directory that contains relevant users. Note: Additional directories can be added later.

The Name and Password values must match those set in the `JIVEFORUMS/WEB-INF/classes/crowd.properties` (see Step 2 below).

1.3 Specify which Users can Log In to Jive Forums

Now that Crowd is aware of the Jive Forums application, Crowd needs to know which directories or users can authenticate (log in) via Crowd. You can either configure entire directories to authenticate or allow particular groups. In our example, we can simply allow the entire directory to authenticate:



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

Please see Specifying an Application's Address or Hostname. Please note:

- Jive Forums is on a different host to Crowd
  If you are running Jive Forums on a different host to Crowd, you will need to modify the permissible hosts via the Remote Addresses tab. This lists the hosts/IP addresses that are allowed to authenticate to Crowd. If Jive Forums is remote to Crowd, add the IP address of your Jive Forums server and ensure the "Status" field is set to "true". Remove the entry for `localhost`.
- Jive Forums is on the same host as Crowd
  By default, when you add an application, `localhost` is a permissible foreign host. However, you will also need to manually add the IP address `127.0.0.1`, as incoming requests to Crowd from Jive (both on the same, local, host) may be from the host `127.0.0.1` and not `localhost`. Crowd does not do a DNS lookup of the hostname, rather, it compares the values as is. Ensure the "Status" field is set to "true".

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 summarised 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. 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.
3. Edit `JIVEFORUMS/WEB-INF/classes/crowd.properties`. Change the following properties:

Key	Value
application.name	jiveforums
application.password	set a password

   The name and password values must match those set when defining the application in Crowd (see Step 1 above).

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.

This applies to the Crowd Administration Console's `crowd.properties` file too. You may find this particularly useful when integrating with a WAR deployment of an integrated application.

Example:

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

2.2 Configure Jive Forums to use Crowd's Authenticator

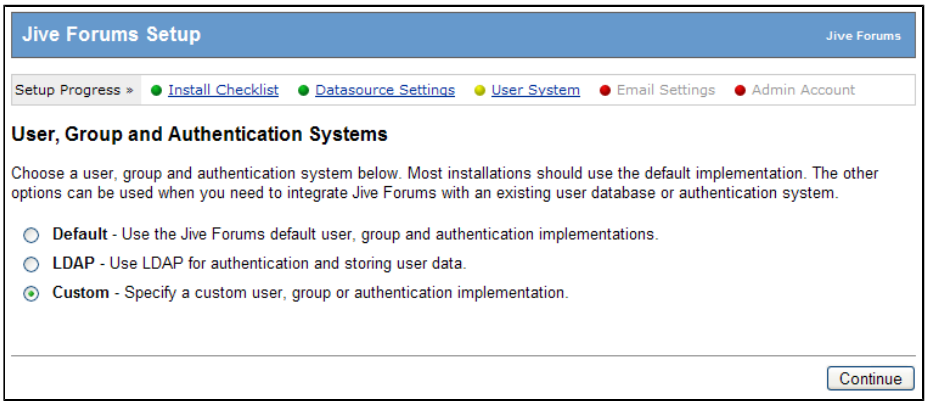
Crowd is now set up to provide authentication services to Jive. Now Jive needs to be set up to use Crowd's authenticator. There are a few ways of doing this. The most user-friendly method is outlined below:

1. In your `jiveHome` directory, edit a file named `jive_startup.xml`. Modify the `<setup>` node to be `false`:

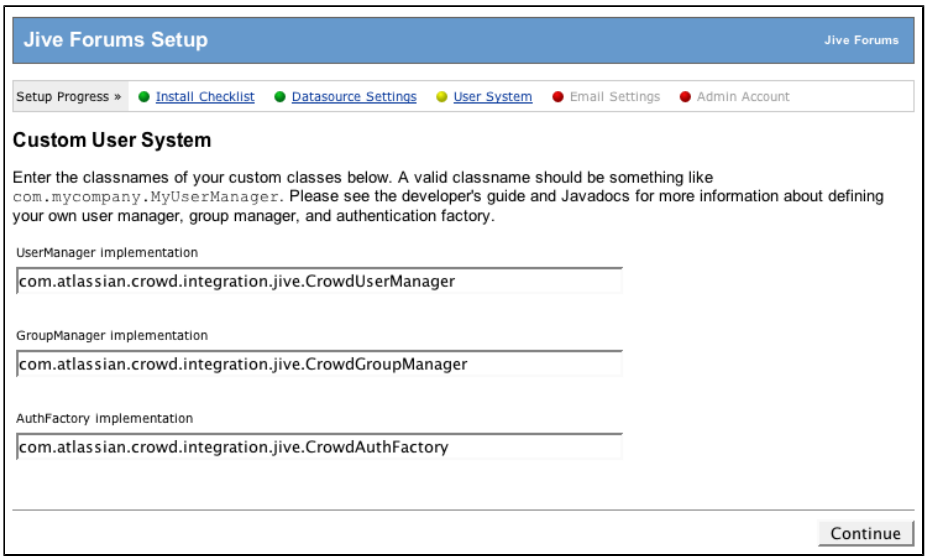
```
<jive>
  <!-- When setup is false, you can access the setup tool. -->
  <setup>false</setup>
  ...
  <!-- Allow SSO login for admins -->
  <admin>
    <tryAlternativeLogin>true</tryAlternativeLogin>
  </admin>
</jive>
```

   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':



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
```

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'.

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.

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
    - Configuring Confluence for NTLM SSO
  - Integrating Crowd with Atlassian CrowdID
  - Integrating Crowd with Atlassian Crucible
  - Integrating Crowd with Atlassian FishEye
    - Configuring FishEye 1.3.x to talk to Crowd
  - Integrating Crowd with Atlassian JIRA
    - Configuring JIRA for NTLM SSO
  - Integrating Crowd with Acegi Security
    - Integrating AppFuse - a Crowd-Acegi Integration Tutorial
  - Integrating Crowd with Apache
  - Integrating Crowd with Jive Forums
    - Jive SSO
  - Integrating Crowd with Subversion
  - Integrating Crowd with a Custom Application
- Mapping a Directory to an Application
  - Specifying the Directory Order for an Application
  - Specifying an Application's Directory Permissions
    - Example of Directory Permissions
- Specifying which Groups can access an Application
- Specifying an Application's Address or Hostname
- Testing a User's Login to an Application
- Managing an Application's Session
- Deleting or Deactivating an Application

Crowd Documentation

Jive SSO

This page last changed on May 05, 2008 by smaddox.

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 authorised to authenticate. This user also needs to NOT be on any user/IP ban lists within the Jive application. The Crowd integration will honour 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 utilise Crowd's "remember me", so that "remember me" is centralised.

- 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 honoured 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 recognises 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
        - Configuring Confluence for NTLM SSO
    - Integrating Crowd with Atlassian CrowdID
    - Integrating Crowd with Atlassian Crucible
    - Integrating Crowd with Atlassian FishEye
        - Configuring FishEye 1.3.x to talk to Crowd
    - Integrating Crowd with Atlassian JIRA
        - Configuring JIRA for NTLM SSO
    - Integrating Crowd with Acegi Security
        - Integrating AppFuse - a Crowd-Acegi Integration Tutorial
    - Integrating Crowd with Apache
    - Integrating Crowd with Jive Forums
        - Jive SSO
    - Integrating Crowd with Subversion
    - Integrating Crowd with a Custom Application
- Mapping a Directory to an Application
    - Specifying the Directory Order for an Application
    - Specifying an Application's Directory Permissions
        - Example of Directory Permissions
- Specifying which Groups can access an Application
- Specifying an Application's Address or Hostname
- Testing a User's Login to an Application
- Managing an Application's Session
- Deleting or Deactivating an Application

Crowd Documentation

Integrating Crowd with Subversion

This page last changed on May 05, 2008 by smaddox.

Install the Crowd Apache connector first

To use the Subversion connector, you will need to have the Crowd Apache Connector already installed.

Crowd's Subversion connector allows you to password-protect a Subversion repository and provide fine grained access by group or user.

Prerequisites

- Crowd Apache Connector.

Configuring Crowd Authentication for Subversion

If you are using Apache to manage access to a Subversion repository (instructions), and are using Crowd to manage the Subversion authentication, then you can use the same configuration method to delegate user authentication to Crowd.

Example:

```
<Location /svn>

 # Uncomment this to enable the repository,
 DAV svn

 # Set this to the path to your repository
 SVNPath /var/lib/svn

 AuthName crowd
 AuthType Basic

 PerlAuthenHandler Apache::CrowdAuth
 PerlSetVar CrowdAppName subversion
 PerlSetVar CrowdAppPassword svn
 PerlSetVar CrowdSOAPURL http://localhost:8095/crowd/services/SecurityServer

 require valid-user

 # The following three lines allow anonymous read, but make
 # committers authenticate themselves.
 <LimitExcept GET PROPFIND OPTIONS REPORT>
 Require valid-user
 </LimitExcept>

</Location>
```

Note that Apache will have to be restarted before any changes to its config files will take effect.

Configuring Crowd Authorisation for Subversion

To restrict Subversion repository access to certain groups and/or users, you can add the `Apache::CrowdAuthz` module and the `CrowdAllowedGroups` and `CrowdAllowedUsers` directives (described here).

For more fine-grained access, the `CrowdAuthzSVNAccessFile` directive is provided. For example:

```
PerlAuthzHandler Apache::CrowdAuthz
PerlSetVar CrowdAuthzSVNAccessFile /etc/apache2/dav_svn.authz
```

The `CrowdAuthzSVNAccessFile` setting lets you define a file where you can configure group and user access on a per-directory level.

The format of the file is the same as that used by Subversion's own authorisation module, `mod_authz_svn`. Here's a small example:

```
# Everyone has read access to the repository
# (unless modified below).
[/]
* = r

# Members of the bazdevelopers group can
# read and write to the BazWord project
[/BazWord]
@bazdevelopers = rw

# Members of the foodevelopers group can read and write
# to the FooCalc project
[/FooCalc]
@foodevelopers = rw

# Members of foodevelopers can read the branches
# directory but only user juliag (the release manager)
# can write to this path
[/FooCalc/branches]
juliag = rw
@foodevelopers = r

# peterc is a contractor, so he's denied all access to the statistics
# module (which is full of trade secrets).
[/FooCalc/trunk/statistics]
peterc =
```

Some notes:

- The format is a series of one or more repository paths (minus the leading URL) followed by one or more group or user directives for each path.
- You don't have to include every single path. If an exact path match isn't found, the settings for the nearest parent directory are used.
- A user or group can be set to one of:
  - `rw`: read and write access.
  - `r`: read-only access.
  - `<blank>`: no access.
- Group names are indicated by a leading '@' character.
- Lines starting with a '#' are comments.
- Note that the `[groups]` section of the file described in the Subversion docs is ignored by `Apache::CrowdAuthz` as group memberships come from Crowd.
- Don't prefix the paths in the file with the repository name (e.g. '[calc:/foo]') (see note on `SVNParentPath` below).
- If you specify a `CrowdAuthzSVNAccessFile` as well as one or both of `CrowdAllowedGroups` and `CrowdAllowedUsers`, only the `CrowdAuthzSVNAccessFile` is used for authorisation.

  SVNParentPath Not Supported
  Subversion provides the SVNParentPath directive, which allows multiple repositories in the same directory to use the same URL. The Crowd Apache integration modules do not support the use of SVNParentPath.

  For a detailed description of this file format, see the Subversion documentation.

RELATED TOPICS

- Using the Application Browser
- Adding an Application
  - Integrating Crowd with Atlassian Bamboo
  - Integrating Crowd with Atlassian Confluence
    - Configuring Confluence for NTLM SSO
  - Integrating Crowd with Atlassian CrowdID

Crowd Documentation

Integrating Crowd with a Custom Application

This page last changed on May 05, 2008 by smaddox.

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 SOAP API.
For assistance, please see 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 classpath, typically `WEB-INF/lib`.
    - These files will be in the `client` folder 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.

This applies to the Crowd Administration Console's `crowd.properties` file too. You may find this particularly useful when integrating with a WAR deployment of an integrated application.

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
        - Configuring Confluence for NTLM SSO
    - Integrating Crowd with Atlassian CrowdID
    - Integrating Crowd with Atlassian Crucible
    - Integrating Crowd with Atlassian FishEye
        - Configuring FishEye 1.3.x to talk to Crowd
    - Integrating Crowd with Atlassian JIRA
        - Configuring JIRA for NTLM SSO
    - Integrating Crowd with Acegi Security
        - Integrating AppFuse - a Crowd-Acegi Integration Tutorial

[Crowd Documentation

Mapping a Directory to an Application

This page last changed on May 05, 2008 by smaddox.

Mapping a directory to an application defines the user-base for an application. Sometimes known as 'application provisioning', directory mappings determine which user stores will be used when authenticating and authorising a user's access request. Read more about users, groups and roles.

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.

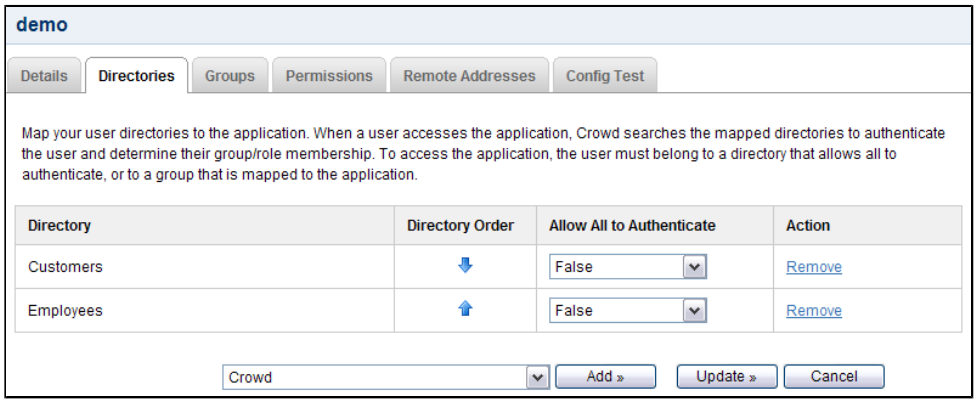
To map a directory to an application,

1. Log in to the Crowd Administration Console.
2. 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 map.
4. This will display the 'View Application' screen. Click the 'Directories' tab.
5. This will display a list of directories that are currently mapped to the application. Select the new directory from the drop-down list and click the 'Add' button.
6. The new directory will be added to the bottom of the list of mapped directories. You can use the blue up-arrow or down-arrow to move a directory higher or lower in the order:

   Why directory order is important
7. You now need to choose which users within the directory may authenticate against the application. You have two choices:
   - To allow all users within the directory to authenticate against the application, change 'Allow all to Authenticate' to 'True', then click the 'Update' button.
     OR:
   - To allow only specific groups of users within the directory to authenticate against the application, see Specifying which Groups can access an Application.
8. Next, you should define the application's ability to add/update users in the directory. Click the 'Permissions' tab and set the directory permissions for the application.

Screenshot: 'Application — Map Directories'



RELATED TOPICS

- Using the Application Browser
- Adding an Application
  - Integrating Crowd with Atlassian Bamboo
  - Integrating Crowd with Atlassian Confluence
    - Configuring Confluence for NTLM SSO
  - Integrating Crowd with Atlassian CrowdID
  - Integrating Crowd with Atlassian Crucible
  - Integrating Crowd with Atlassian FishEye

Specifying the Directory Order for an Application

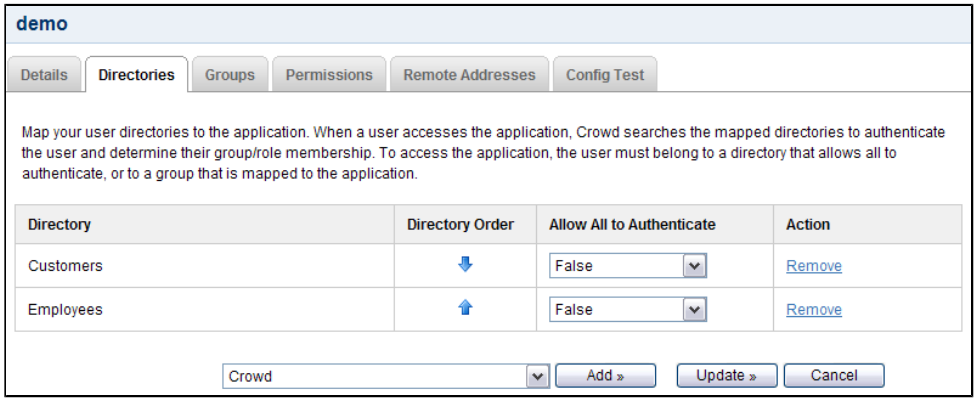
This page last changed on May 05, 2008 by smaddox.

When you map multiple directories to an application, you also need to define the directory order. This is important in case the same user exists in multiple directories. When a user attempts to access an application, Crowd will search the directories in the order you specified, and will use the credentials (password, etc) of the first occurrence of the user to validate the login attempt (see diagram below).

To specify the directory order,

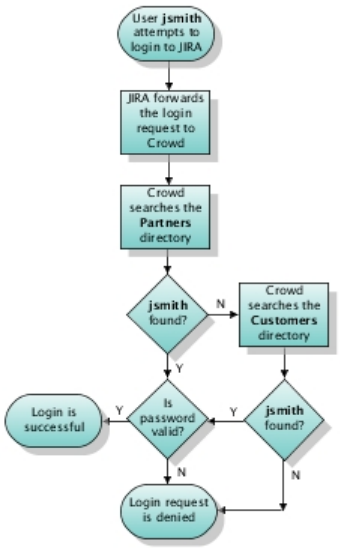
1. Log in to the Crowd Administration Console.
2. Click the 'Applications' tab in the top navigation bar.
3. This will display the Application Browser. Click the 'View' link that corresponds to the application you wish to map.
4. This will display the 'View Application' screen. Click the 'Directories' tab.
5. This will display a list of directories that are currently mapped to the application. Use the blue up-arrow or down-arrow to move a directory higher or lower in the order:

Screenshot: 'Application---Mapped Directories'



How it works

Let's assume that JIRA has been set up as a Crowd application, and has been mapped to two directories, 'Partners' and 'Customers', in that order (as shown in the above screenshot).
Here is what happens when a user attempts to log in to JIRA:



When granting the user access to an application, Crowd amalgamates the group memberships in the directories, as described in Editing a User's Group and Role Membership.

RELATED TOPICS

- Using the Application Browser
- Adding an Application
  - Integrating Crowd with Atlassian Bamboo
  - Integrating Crowd with Atlassian Confluence
    - Configuring Confluence for NTLM SSO
  - Integrating Crowd with Atlassian CrowdID
  - Integrating Crowd with Atlassian Crucible
  - Integrating Crowd with Atlassian FishEye
    - Configuring FishEye 1.3.x to talk to Crowd
  - Integrating Crowd with Atlassian JIRA
    - Configuring JIRA for NTLM SSO
  - Integrating Crowd with Acegi Security
    - Integrating AppFuse - a Crowd-Acegi Integration Tutorial
  - Integrating Crowd with Apache
  - Integrating Crowd with Jive Forums
    - Jive SSO
  - Integrating Crowd with Subversion
  - Integrating Crowd with a Custom Application
- Mapping a Directory to an Application
  - Specifying the Directory Order for an Application
  - Specifying an Application's Directory Permissions
    - Example of Directory Permissions
- Specifying which Groups can access an Application
- Specifying an Application's Address or Hostname
- Testing a User's Login to an Application
- Managing an Application's Session
- Deleting or Deactivating an Application

Crowd Documentation

Specifying an Application's Directory Permissions

This page last changed on May 05, 2008 by smaddox.

When you map a directory to an application, you can also define the application's ability to add/update/delete users, groups and roles 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.

For details on directory-level permissions, refer to the instructions on specifying directory permissions. Below are instructions on setting the application-level directory permissions.

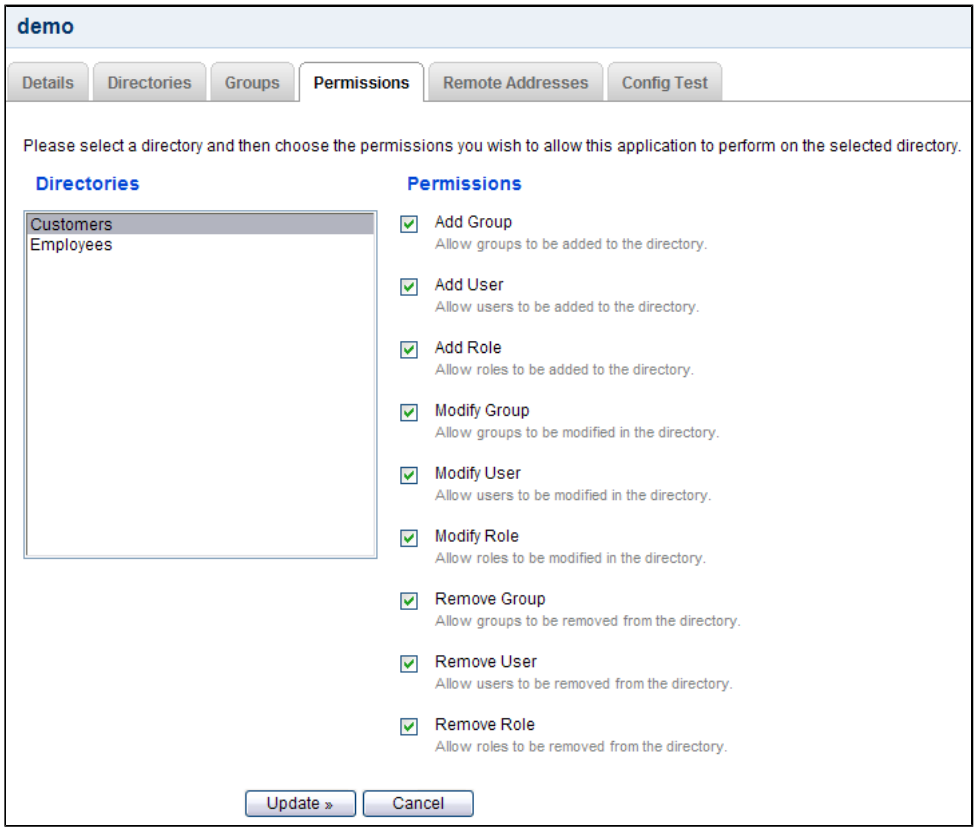
Permission	Description
Add Group	Allows the application to add groups to the selected directory.
Add User	Allows the application to add users to the selected directory.
Add Role	Allows the application to add roles to the selected directory.
Modify Group	Allows the application to modify groups in the selected directory.
Modify User	Allows the application to modify users in the selected directory.
Modify Role	Allows the application to modify roles 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.
Remove Role	Allows the application to delete roles from the selected directory.

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. Click the 'Applications' tab in the top navigation bar.
3. This will display the Application Browser. Click the 'View' link next to the application you wish to update.
4. This will display the 'View Application' screen. Click the 'Permissions' tab.
5. This will display a list of directories that are currently mapped to the application, and a set of permission check-boxes. Select a directory from the list on the left.
6. The 'Permissions' check-boxes will change to show the application's existing permissions for that directory.
    - To enable a directory permission, select the corresponding check-box.
    - To disable a directory permission, deselect the corresponding check-box.

Screenshot: Setting directory permissions for an application



On the application permissions screen, the words '(disabled globally)' will appear next to any permission that is disabled at directory level.

RELATED TOPICS

- Specifying Directory Permissions

- Using the Application Browser
- Adding an Application
    - Integrating Crowd with Atlassian Bamboo
    - Integrating Crowd with Atlassian Confluence
        - Configuring Confluence for NTLM SSO
    - Integrating Crowd with Atlassian CrowdID
    - Integrating Crowd with Atlassian Crucible
    - Integrating Crowd with Atlassian FishEye
        - Configuring FishEye 1.3.x to talk to Crowd
    - Integrating Crowd with Atlassian JIRA
        - Configuring JIRA for NTLM SSO
    - Integrating Crowd with Acegi Security

Example of Directory Permissions

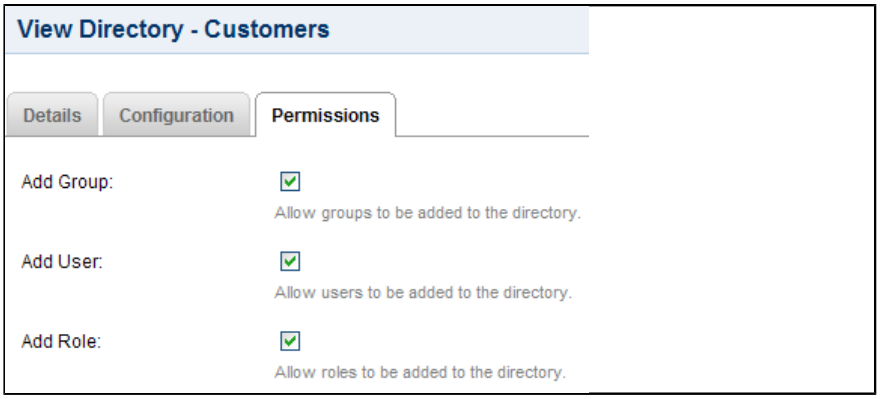
This page last changed on May 05, 2008 by smaddox.

Let's assume that you want to:

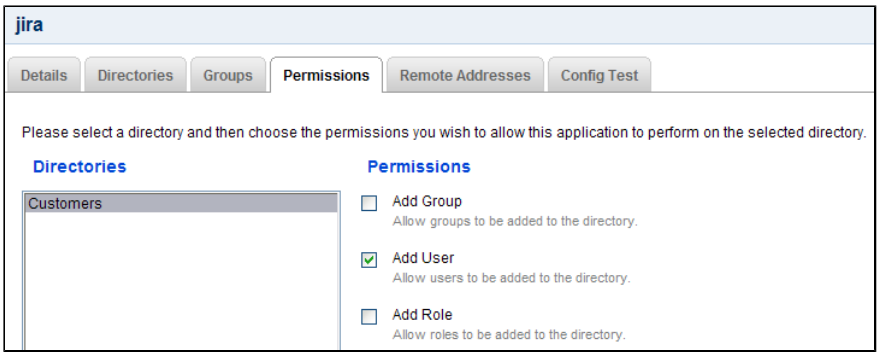
- Allow self-registration (automatic signup) of new users in your 'Customers' directory via JIRA, and
- Disable self-registration via Confluence.

Here's how you would set the directory-level and application-level permissions in Crowd.

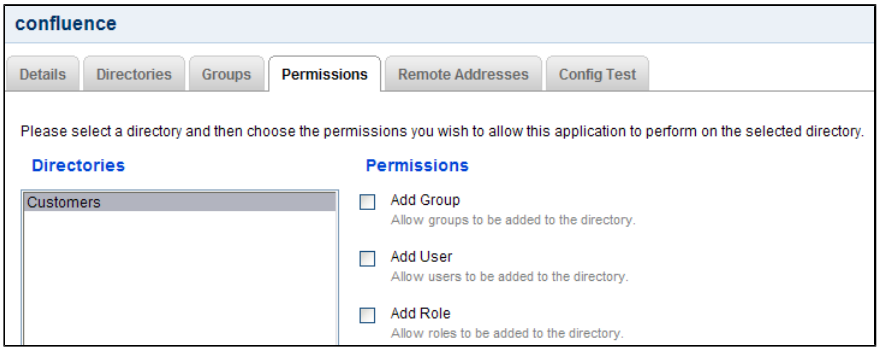
1. At directory level, enable the 'Add User' permission (and any other permissions you want):
   a. In the Crowd Administration Console, click the 'Directories' tab in the top navigation bar.
   b. Select the 'Customers' directory.
   c. Click the 'Permissions' tab.
   d. Select the 'Add User' check-box.



2. At application level, make sure the 'Add User' permission is enabled for the JIRA application:
   a. Click the 'Applications' tab in the top navigation bar.
   b. Click the 'View' link next to the JIRA application.
   c. In the 'View Application' screen, click the 'Permissions' tab.
   d. Select the 'Customers' directory.
   e. Select the 'Add User' check-box.



3. At application level, disable the 'Add User' permission the Confluence application:
   a. Click the 'Applications' tab in the top navigation bar.
   b. Click the 'View' link next to the Confluence application.
   c. Click the 'Permissions' tab.
   d. Select the 'Customers' directory.
   e. Deselect the 'Add User' check-box.



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.

RELATED TOPICS

- Specifying Directory Permissions
- Specifying an Application's Directory Permissions

Crowd Documentation

Specifying which Groups can access an Application

This page last changed on May 07, 2008 by smaddox.

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.

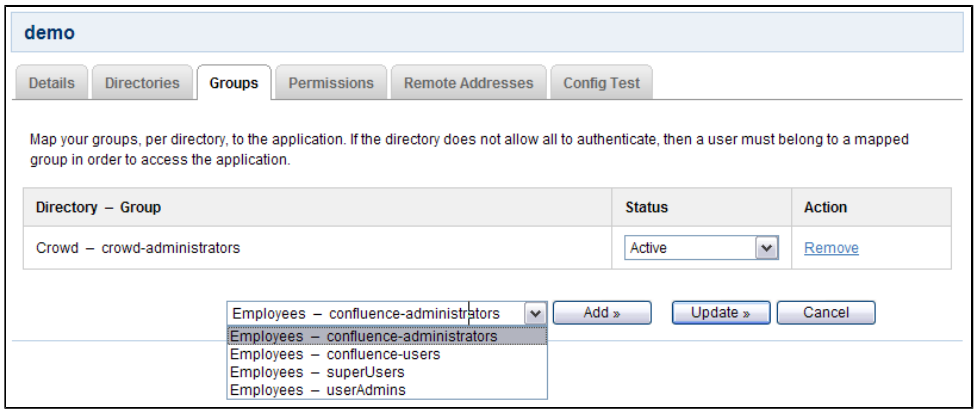
For example, the default group `crowd-administrators`, which is automatically created in the default directory that you specified during setup, is allowed to access the Crowd Administration Console. This means that users who belong to the group `crowd-administrators` are allowed to log in to the Crowd Administration Console (assuming they supply a valid password).

To allow a group to access 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 map.
4. This will display the 'View Application' screen. Click the 'Groups' tab.
5. This will display a list of groups that currently have access to the application. Click the drop-down arrow next to the 'Add' button.
6. This will display a list of all the groups that exist within each directory. Select the new group from the drop-down list and click the 'Add' button.

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'



RELATED TOPICS

- Managing Users, Groups and Roles
- Using the Application Browser
- Adding an Application
  - Integrating Crowd with Atlassian Bamboo
  - Integrating Crowd with Atlassian Confluence
    - Configuring Confluence for NTLM SSO
  - Integrating Crowd with Atlassian CrowdID
  - Integrating Crowd with Atlassian Crucible
  - Integrating Crowd with Atlassian FishEye
    - Configuring FishEye 1.3.x to talk to Crowd
  - Integrating Crowd with Atlassian JIRA
    - Configuring JIRA for NTLM SSO
  - Integrating Crowd with Acegi Security
    - Integrating AppFuse - a Crowd-Acegi Integration Tutorial
  - Integrating Crowd with Apache
  - Integrating Crowd with Jive Forums

Crowd Documentation

Specifying an Application's Address or Hostname

This page last changed on May 05, 2008 by smaddox.

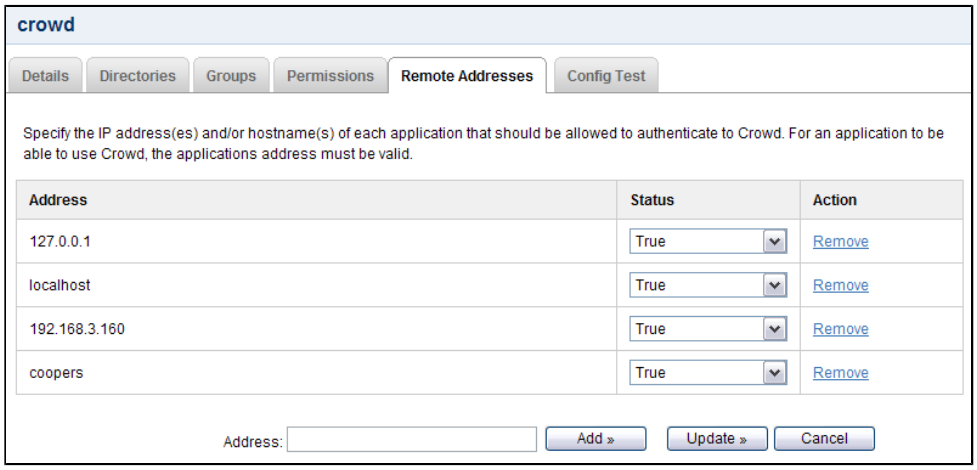
To ensure that your Crowd server can only be used by legitimate applications, Crowd will only allow applications to log in from known addresses. This means that you need to specify the IP address(es) and/or hostname(s) of each application.

When you add a new application, it is restricted by default to `localhost` (`127.0.0.1`). If your application is on a different host, you will need to add the applicable host name or IP address, as described below.

To specify an application's IP address or hostname,

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 map.
4. This will display the 'View Application' screen. Click the 'Remote Addresses' tab.
5. This will display a list of IP addresses and hostnames that are currently mapped to the application. Type the new IP address or hostname into the 'Address' field and click the 'Add' button.
6. The new address will be added to the bottom of the list.

Screenshot: 'Application — Addresses'



Common Misconfiguration
For an application to be able to use Crowd, the application's address must be valid and active.

RELATED TOPICS

- Using the Application Browser
- Adding an Application
  - Integrating Crowd with Atlassian Bamboo
  - Integrating Crowd with Atlassian Confluence
    - Configuring Confluence for NTLM SSO
  - Integrating Crowd with Atlassian CrowdID
  - Integrating Crowd with Atlassian Crucible
  - Integrating Crowd with Atlassian FishEye
    - Configuring FishEye 1.3.x to talk to Crowd
  - Integrating Crowd with Atlassian JIRA
    - Configuring JIRA for NTLM SSO
  - Integrating Crowd with Acegi Security
    - Integrating AppFuse - a Crowd-Acegi Integration Tutorial
  - Integrating Crowd with Apache
  - Integrating Crowd with Jive Forums
    - Jive SSO
  - Integrating Crowd with Subversion

Crowd Documentation

Testing a User's Login to an Application

This page last changed on May 05, 2008 by smaddox.

You can use an application's 'Config 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.

Performing the Test

The test works like this:

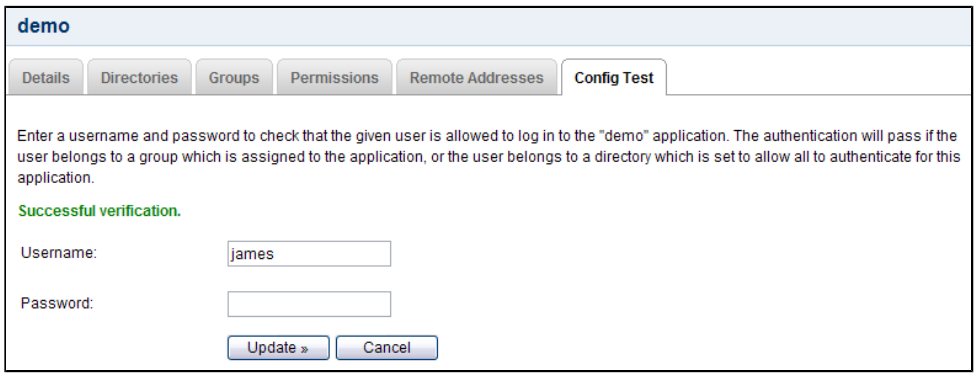
1. You enter the username and password of the user you wish to verify has access to a given application.
2. Crowd searches for the user with that username in the application's mapped directories, and verifies the password.
3. If the user is not found or the password is invalid, the authentication fails 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.

To test a user's login to an application,

1. Log in to the Crowd Administration Console.
2. Click the 'Applications link in the top navigation bar.
3. This will display the Application Browser. Click the 'View' link that corresponds to the application you wish to verify.
4. This will display the 'View Application' screen. Click the 'Config 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: 'Application---Config Test showing successful verification'



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.

RELATED TOPICS

- Using the Application Browser
- Adding an Application
  - Integrating Crowd with Atlassian Bamboo
  - Integrating Crowd with Atlassian Confluence
    - Configuring Confluence for NTLM SSO
  - Integrating Crowd with Atlassian CrowdID
  - Integrating Crowd with Atlassian Crucible
  - Integrating Crowd with Atlassian FishEye
    - Configuring FishEye 1.3.x to talk to Crowd
  - Integrating Crowd with Atlassian JIRA
    - Configuring JIRA for NTLM SSO
  - Integrating Crowd with Acegi Security
    - Integrating AppFuse - a Crowd-Acegi Integration Tutorial
  - Integrating Crowd with Apache
  - Integrating Crowd with Jive Forums
    - Jive SSO
  - Integrating Crowd with Subversion
  - Integrating Crowd with a Custom Application
- Mapping a Directory to an Application
  - Specifying the Directory Order for an Application
  - Specifying an Application's Directory Permissions
    - Example of Directory Permissions
- Specifying which Groups can access an Application
- Specifying an Application's Address or Hostname
- Testing a User's Login to an Application
- Managing an Application's Session
- Deleting or Deactivating an Application

Crowd Documentation

Managing an Application's Session

This page last changed on May 05, 2008 by smaddox.

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. Click the 'Administration' tab in the top navigation bar.
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.

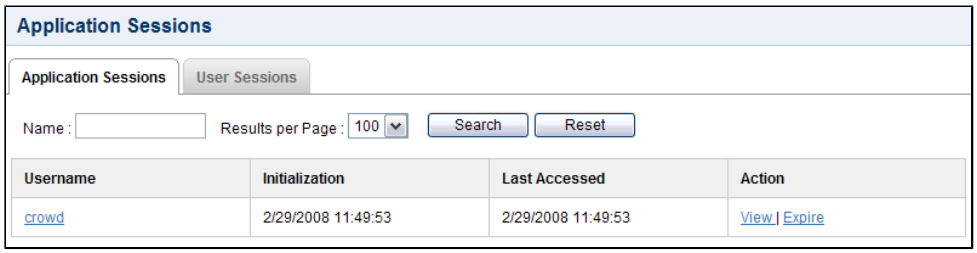
   You can refine your search by specifying an application's 'Name'. (Note that this is case sensitive.)

To force an application to log out of Crowd,

1. Log in to the Crowd Administration Console.
2. Click the 'Administration' tab in the top navigation bar.
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. Click the application's 'Expire' link.

If you want to permanently prevent an application from logging in to Crowd, please see Deleting or Deactivating an Application.

Screenshot: 'Sessions — Applications'



RELATED TOPICS

- Managing a User's Session
- Session Configuration

- Using the Application Browser
- Adding an Application
  - Integrating Crowd with Atlassian Bamboo
  - Integrating Crowd with Atlassian Confluence
    - Configuring Confluence for NTLM SSO
  - Integrating Crowd with Atlassian CrowdID
  - Integrating Crowd with Atlassian Crucible
  - Integrating Crowd with Atlassian FishEye
    - Configuring FishEye 1.3.x to talk to Crowd
  - Integrating Crowd with Atlassian JIRA
    - Configuring JIRA for NTLM SSO
  - Integrating Crowd with Acegi Security

Deleting or Deactivating an Application

This page last changed on May 05, 2008 by smaddox.

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'

RELATED TOPICS

- Using the Application Browser
- Adding an Application
  - Integrating Crowd with Atlassian Bamboo
  - Integrating Crowd with Atlassian Confluence
    - Configuring Confluence for NTLM SSO

Crowd Documentation

Managing Users, Groups and Roles

This page last changed on May 05, 2008 by smaddox.

In Crowd, users are referred to as user entity objects or just users.

Groups and roles 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.

Roles are used less frequently, depending on the requirements of individual applications.

Crowd's role-based access control could be enhanced

At present, the implementation of roles in Crowd is identical to the implementation of groups. Additional development work would be needed to differentiate the functionality of roles from groups. If you would like to use role-based access control via Crowd, please would you add a comment to the improvement request CWD-931, letting us know what enhancements you would like to see.

This section describes how to add/edit users, groups and roles via the Crowd Administration Console. Note that the ability to do this depends on the permissions of the directory which contains the users, groups and roles.

Using the User Browser

This page last changed on May 05, 2008 by smaddox.

In Crowd, users are referred to as user entity objects or just users.

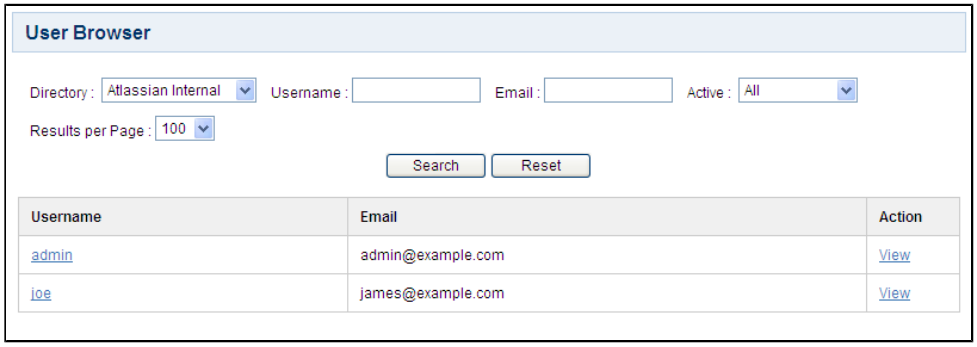
The User Browser allows you to search, view, add and edit users within a specified directory.

To use the User Browser,

1. Log in to the Crowd Administration Console.
2. Click the 'Users' tab in the top navigation bar.
3. This will display the User Browser. Select the directory in which you are interested, then click the 'Search' button to list all the users that exist in that directory.

   You can refine your search by specifying a 'Username' and/or 'Email' (note that these are case-sensitive), or 'Active'/'Inactive' users. (An 'Inactive' user is typically someone who has left your organisation.)
4. To view/edit a user's details, click the 'View' link.

Screenshot: 'User Browser'



RELATED TOPICS

- Using the User Browser
- Adding a User
- Deleting or Deactivating a User
- Managing a User's Session
- Editing a User's Details and Password
- Specifying a User's Attributes
- Editing a User's Group and Role Membership
- Granting Crowd Administration Rights to a User
- Granting Crowd User Rights to a User
- Using the Group Browser and Role Browser
- Adding a Group or Role
- Deleting or Deactivating a Group
- Viewing Members of a Group
  - Nested Groups in Crowd
  - Adding a Sub-Group
  - Removing a Sub-Group

Crowd Documentation

Adding a User

This page last changed on May 05, 2008 by smaddox.

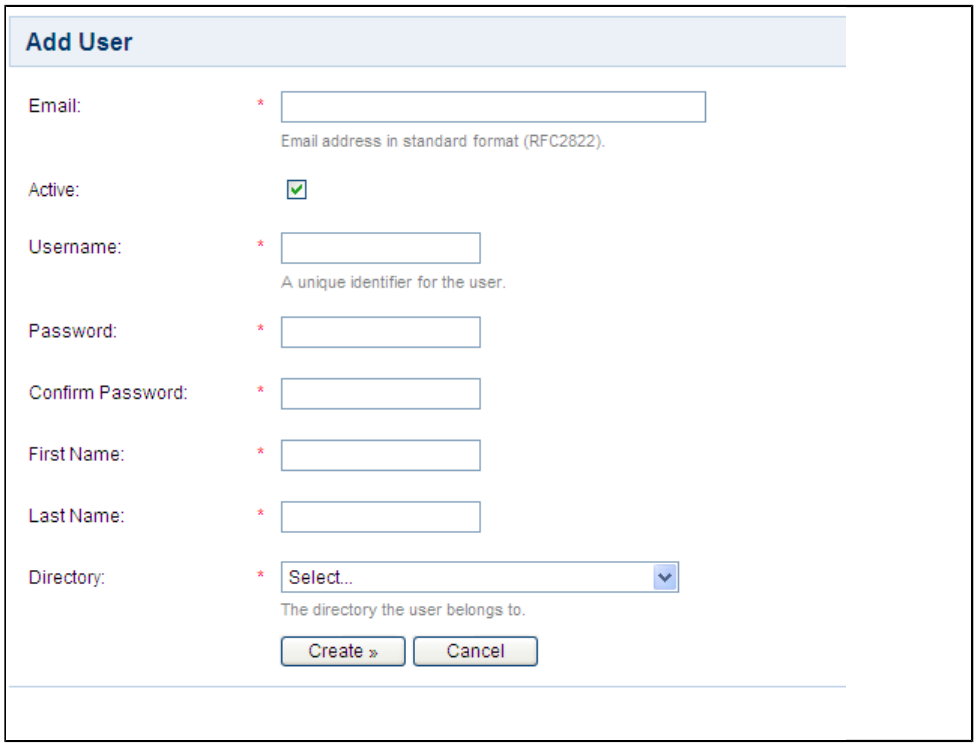
In Crowd, users are referred to as user entity objects or just users.
You can either import users into Crowd in bulk (see Importing Users and Groups into a Directory), or add them individually as described below.

To add a user,

1. Log in to the Crowd Administration Console.
2. Click the 'Users' tab in the top navigation bar.
3. This will display the User Browser. Click 'Add User' in the left-hand menu.
4. Complete the following fields:
    - 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.

      If you have configured an email server and a notification template, Crowd will send the user an email notification about their new 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 the 'Create' button to add the user.
6. After creating the user, you will be able to specify the user's attributes and group/role membership. If you wish, you can also verify that the user can log in to appropriate applications.

Screenshot: 'Add User'



RELATED TOPICS

- Using the User Browser

- Adding a User
- Deleting or Deactivating a User
- Managing a User's Session
- Editing a User's Details and Password
- Specifying a User's Attributes
- Editing a User's Group and Role Membership
- Granting Crowd Administration Rights to a User
- Granting Crowd User Rights to a User
- Using the Group Browser and Role Browser
- Adding a Group or Role
- Deleting or Deactivating a Group
- Viewing Members of a Group
    - Nested Groups in Crowd
    - Adding a Sub-Group
    - Removing a Sub-Group

Crowd Documentation

Deleting or Deactivating a User

This page last changed on May 05, 2008 by smaddox.

Deactivating a user prevents them from logging in to any applications that use the Crowd framework. You would typically do this when a user leaves your organisation.

Deleting a user removes them completely from the relevant directory.

Consider deactivating instead of deleting

We recommend that you deactivate a user rather than delete them, in case some applications contain historical data, e.g. documents that the user has created. Read more.

To deactivate a user,

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, locate the user you wish to deactivate, and click the 'View' link that corresponds to the user.
4. This will display the 'User Details' screen. Deselect the 'Active' check-box, then click the 'Update' button. The user will now be unable to log in to any applications which use the Crowd framework.

To delete a user,

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. Click the 'View' link that corresponds to the user you wish to delete.
4. This will display the 'User Details' screen. Click 'Remove User' in the left-hand menu.

The user will be removed from the relevant directory and will no longer appear in the User Browser..

RELATED TOPICS

- Using the User Browser
- Adding a User
- Deleting or Deactivating a User
- Managing a User's Session
- Editing a User's Details and Password
- Specifying a User's Attributes
- Editing a User's Group and Role Membership
- Granting Crowd Administration Rights to a User
- Granting Crowd User Rights to a User
- Using the Group Browser and Role Browser
- Adding a Group or Role
- Deleting or Deactivating a Group
- Viewing Members of a Group
  - Nested Groups in Crowd
  - Adding a Sub-Group
  - Removing a Sub-Group

Crowd Documentation

Managing a User's Session

This page last changed on May 05, 2008 by smaddox.

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.

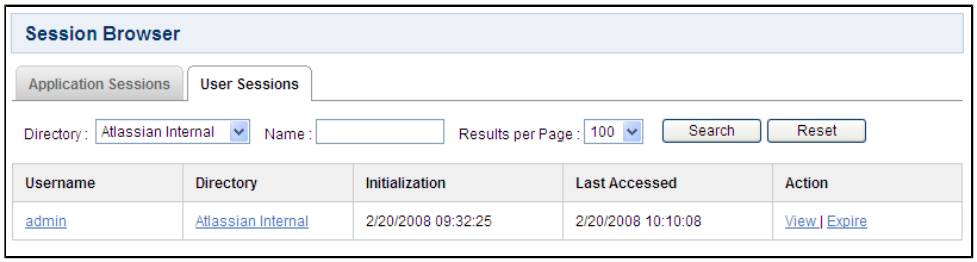
You can also force any session to expire, that is, you can log the user out of Crowd.

To see which users are currently logged in to Crowd,

1. Log in to the Crowd Administration Console.
2. Click the 'Administration' tab in the top navigation bar.
3. Click 'Current Sessions' in the left-hand menu.
4. This will display the 'Session Browser'. Click the 'User Sessions' tab.
5. Select the directory containing the users in which you are interested, and click the 'Search' button.
6. 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).

Screenshot: 'Session Browser — Users'



To log a user out of Crowd,

1. Login to the Crowd Administration Console.
2. Click the 'Administration' link in the top navigation bar.
3. Click 'Current Sessions' in the left-hand menu.
4. Click the 'User Sessions' tab.
5. This will display a list of all users which are currently logged in to the Crowd framework. Click the user's 'Expire' link.

If you want to permanently prevent a user from logging in to Crowd, please see Deleting or Deactivating a User.

See Also

Managing an Application's Session
Session Configuration

RELATED TOPICS

- Using the User Browser
- Adding a User
- Deleting or Deactivating a User
- Managing a User's Session
- Editing a User's Details and Password
- Specifying a User's Attributes
- Editing a User's Group and Role Membership
- Granting Crowd Administration Rights to a User
- Granting Crowd User Rights to a User
- Using the Group Browser and Role Browser

- Adding a Group or Role
- Deleting or Deactivating a Group
- Viewing Members of a Group
    - Nested Groups in Crowd
    - Adding a Sub-Group
    - Removing a Sub-Group

Crowd Documentation

Editing a User's Details and Password

This page last changed on May 08, 2008 by smaddox.

In Crowd, users are referred to as user entity objects or just users.

To edit a user's details,

1. Log in to the Crowd Administration Console.
2. Click the 'Users' tab in the top navigation bar.
3. This will display the User Browser. Select the relevant directory, locate the user in which you are interested, then click the 'View' link corresponding to the user.
4. This will display the 'User Details' screen.
5. Edit the details as required, then click the 'Update' button.

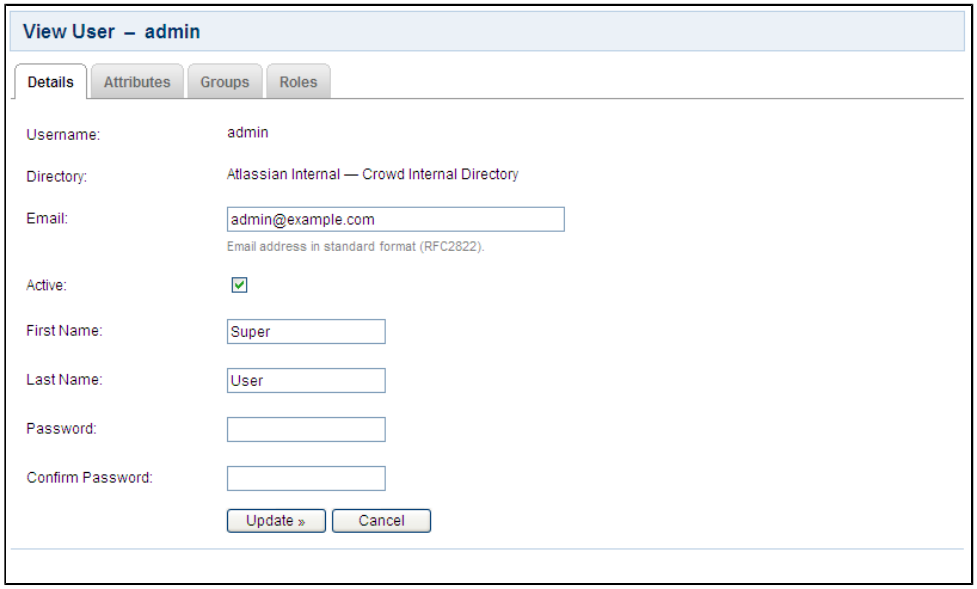
To change a user's password,

1. Log in to the Crowd Administration Console.
2. Click the 'Users' tab in the top navigation bar.
3. This will display the User Browser. Select the relevant directory, locate the user in which you are interested, then click the 'View' link corresponding to the user.
4. This will display the 'User Details' screen. You can either:

- Enter the new password, then click the 'Update' button;
  OR
- Click 'Reset Password' in the left-hand menu. This will generate a new password (i.e. one which you do not know) and email it to the user.

If you have configured an Email Server and a Notification Template, Crowd will send the user an email notification about their new password.

Screenshot: 'User Details'



Users can update their own profiles

Authorised Crowd users can log in to the Self Service Console and update their own user profiles, as described in the Crowd User Guide.

RELATED TOPICS

- Using the User Browser
- Adding a User
- Deleting or Deactivating a User
- Managing a User's Session
- Editing a User's Details and Password
- Specifying a User's Attributes
- Editing a User's Group and Role Membership
- Granting Crowd Administration Rights to a User
- Granting Crowd User Rights to a User
- Using the Group Browser and Role Browser
- Adding a Group or Role
- Deleting or Deactivating a Group
- Viewing Members of a Group
    - Nested Groups in Crowd
    - Adding a Sub-Group
    - Removing a Sub-Group

Crowd Documentation

Specifying a User's Attributes

This page last changed on May 05, 2008 by smaddox.

In Crowd, users are referred to as user entity objects or just users.

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 updated 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.

To edit a user's attributes,

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, locate the user in which you are interested, then click the 'View' link corresponding to the user.
4. This will display the 'User Details' screen. Click the 'Attributes' tab.

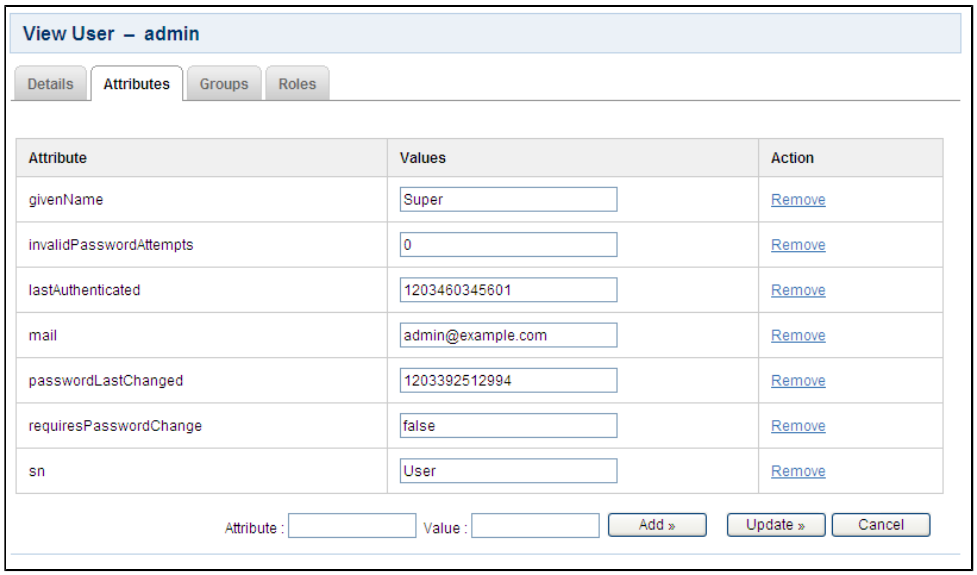
- To add a new attribute,

   You cannot add an attribute to an LDAP directory — see note above.

  1. Type the name of the new attribute (e.g. `phone`) in the 'Attribute' field at the bottom of the screen.
  2. Type the value of the new attribute (e.g. `0123456789`) in the 'Value' field at the bottom of the screen.
  3. Click the 'Add' button.
- To edit an existing attribute, edit the corresponding field in the 'Values' column, then click the 'Update' button.
- 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.

Screenshot: 'User Attributes'



RELATED TOPICS

- Using the User Browser

- Adding a User
- Deleting or Deactivating a User
- Managing a User's Session
- Editing a User's Details and Password
- Specifying a User's Attributes
- Editing a User's Group and Role Membership
- Granting Crowd Administration Rights to a User
- Granting Crowd User Rights to a User
- Using the Group Browser and Role Browser
- Adding a Group or Role
- Deleting or Deactivating a Group
- Viewing Members of a Group
    - Nested Groups in Crowd
    - Adding a Sub-Group
    - Removing a Sub-Group

Crowd Documentation

Editing a User's Group and Role Membership

This page last changed on May 05, 2008 by smaddox.

Within any given directory, you can choose the groups and roles to which each user belongs.

Note that a user's group membership is particularly important, as groups are often used to control access to applications.

What happens if a user exists in more than one directory?

If the same username exists in more than one directory assigned to an application, Crowd treats these usernames as the same user. Crowd searches all the assigned directories for the user, and amalgamates the group and role memberships.

For example, let's assume you have a user 'P' who exists in both directories 'D1' and 'D2', and is a member of group 'G1' in 'D1' and 'G2' in 'D2'. Crowd will grant access to the user based on membership of both 'G1' and 'G2'.

When authenticating the user, Crowd uses the first directory in which the username occurs, as described in Specifying the Directory Order for an Application.

To add a user to a group,

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, locate the user you wish to add, and click the 'View' link that corresponds to the user.
4. This will display the 'User Details' screen. Click the 'Groups' tab.
5. A list of the user's current groups (if any) will be displayed. Select the relevant group from the drop-down box below the list, then click the 'Add' button.

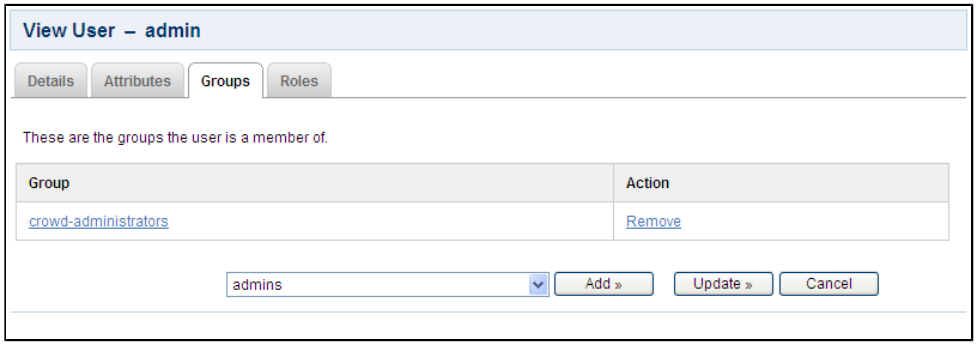
The user will now be authorised to use any applications that use this group to control access.

To remove a user from a group,

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, locate the user you wish to remove, and click the 'View' link that corresponds to the user.
4. This will display the 'User Details' screen. Click the 'Groups' tab.
5. A list of the user's current groups (if any) will be displayed. Click the 'Remove' link corresponding to the relevant group.

The user will now be unable to log in to any applications that use this group to control access.

Screenshot: 'User — Groups'



The adding or removing of a user to or from a role is performed via the Role Browser, but is otherwise identical to the process for groups.

RELATED TOPICS

- Using the User Browser
- Adding a User
- Deleting or Deactivating a User
- Managing a User's Session
- Editing a User's Details and Password
- Specifying a User's Attributes
- Editing a User's Group and Role Membership
- Granting Crowd Administration Rights to a User
- Granting Crowd User Rights to a User
- Using the Group Browser and Role Browser
- Adding a Group or Role
- Deleting or Deactivating a Group
- Viewing Members of a Group
    - Nested Groups in Crowd
    - Adding a Sub-Group
    - Removing a Sub-Group

Crowd Documentation

Granting Crowd Administration Rights to a User

This page last changed on May 07, 2008 by smaddox.

Members of the 'crowd-administrators' group have administration privileges — that is, the ability to:

- access the Crowd Administration Console and perform the functions described in the Crowd 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. Click the 'Users' tab in the top navigation bar.
3. This will display the User Browser. Select the directory which contains the user to whom you wish to grant administration rights.
4. Locate the user and 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.

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.

RELATED TOPICS

- Using the User Browser
- Adding a User
- Deleting or Deactivating a User
- Managing a User's Session
- Editing a User's Details and Password
- Specifying a User's Attributes
- Editing a User's Group and Role Membership
- Granting Crowd Administration Rights to a User
- Granting Crowd User Rights to a User
- Using the Group Browser and Role Browser
- Adding a Group or Role
- Deleting or Deactivating a Group
- Viewing Members of a Group
  - Nested Groups in Crowd
  - Adding a Sub-Group
  - Removing a Sub-Group

Crowd Documentation

Granting Crowd User Rights to a User

This page last changed on May 08, 2008 by smaddox.

This page tells you how to authorise users to access Crowd, without giving them Crowd administration rights. Only Crowd administrators can authorise other users to access Crowd.

Administrators versus Non-Administrators

The Crowd Administration Console presents the full range of Crowd administration functionality to authorised Crowd administrators.

Authorised 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 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 authorisation details. But they will not be able to view or update other user profiles, nor perform any Crowd administration functions.

Authorising Non-Administrators to Use the Crowd Self-Service Console

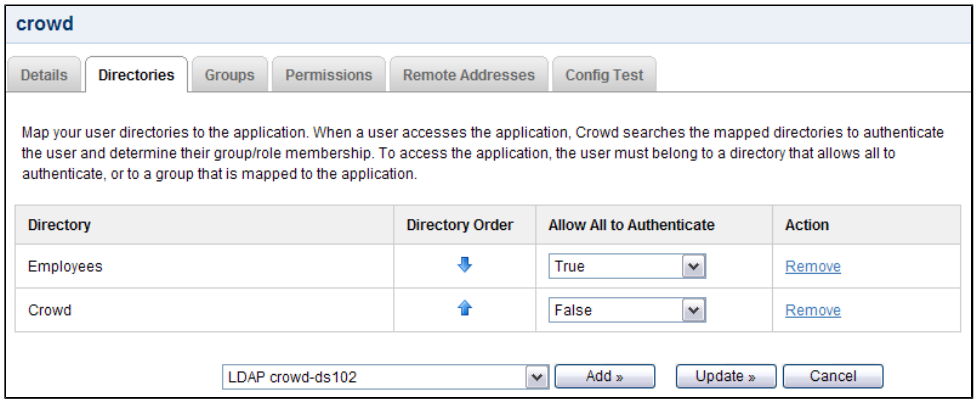
To authorise a non-administrator to use Crowd, you should ensure that:

- the person's username is in a user directory where all users are authorised 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 'Directories' tab, set the 'Allow All to Authenticate' option to 'True'.
4. Add the user(s) to the directory, if not already added.

Screenshot: Granting an entire directory access to the 'crowd' application



RELATED TOPICS

Granting Crowd Administration Rights to a User
Crowd User Guide
Crowd Documentation

Using the Group Browser and Role Browser

This page last changed on May 07, 2008 by smaddox.

About Groups and Roles

Groups and roles 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.

Roles are used less frequently, depending on the requirements of individual applications.

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. Currently, nested groups are supported for LDAP directory connectors only.

For more details about nested groups, refer to Nested Groups in Crowd.

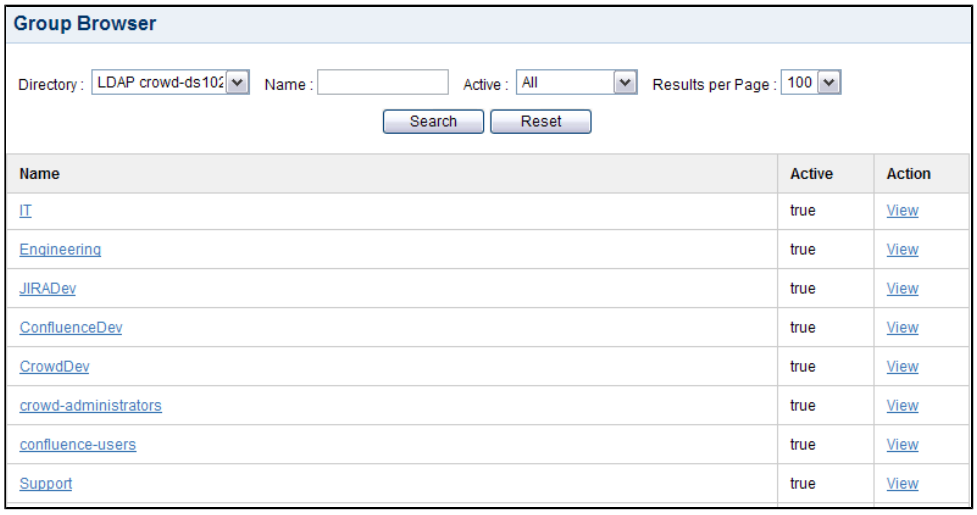
About the Group Browser and the Role Browser

The Group Browser and the Role Browser are very similar. They allow you to search, view, add and edit the various groups and roles stored within a specified directory.

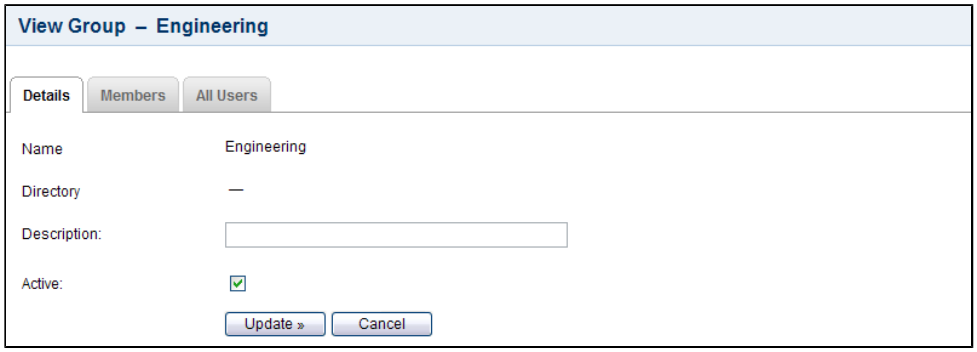
To use the Group Browser,

1. Log in to the Crowd Administration Console.
2. Click the 'Groups' tab in the top navigation bar.
3. The Group Browser will appear. Select the directory in which you are interested, then click the 'Search' button to list all the groups that exist in that directory.
   You can refine your search by specifying a 'Name' (note that this is case-sensitive), or 'Active'/'Inactive' groups.
4. To view or edit a group's details, click the 'View' link.
5. Click the 'Members' tab to view the immediate members of the group, including users and other groups.
6. Click the 'All Users' tab (if present) to view all users who are included in the group and in its sub-groups
   - The 'All Users' tab will appear only if the group you are viewing contains sub-groups.
   - You can read more about group members in Viewing Members of a Group.

Screenshot 1: 'Group Browser'



Screenshot 2: 'View and Update Group Details'



RELATED TOPICS

- Using the User Browser
- Adding a User
- Deleting or Deactivating a User
- Managing a User's Session
- Editing a User's Details and Password
- Specifying a User's Attributes
- Editing a User's Group and Role Membership
- Granting Crowd Administration Rights to a User
- Granting Crowd User Rights to a User
- Using the Group Browser and Role Browser
- Adding a Group or Role
- Deleting or Deactivating a Group
- Viewing Members of a Group
    - Nested Groups in Crowd
    - Adding a Sub-Group
    - Removing a Sub-Group

Crowd Documentation

Adding a Group or Role

This page last changed on May 08, 2008 by smaddox.

Groups and roles 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.

Roles are used less frequently, depending on the requirements of individual applications.

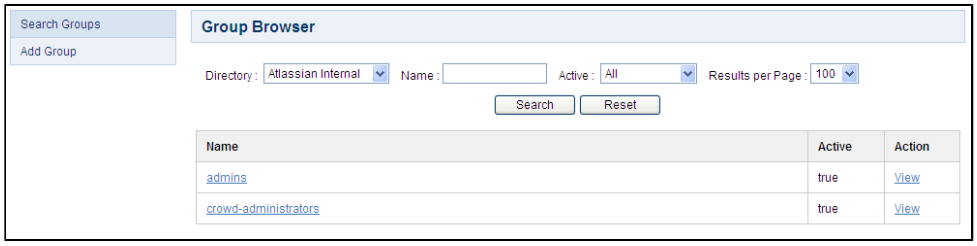
To add a group or role,

1. Log in to the Crowd Administration Console.
2. Click the 'Groups' or 'Roles' link in the top navigation bar.
3. This will display the Group Browser (or Role Browser). Click 'Add Group' or 'Add Role' in the left-hand menu.
4. Complete the fields as described in the table below, then click the 'Create' button.

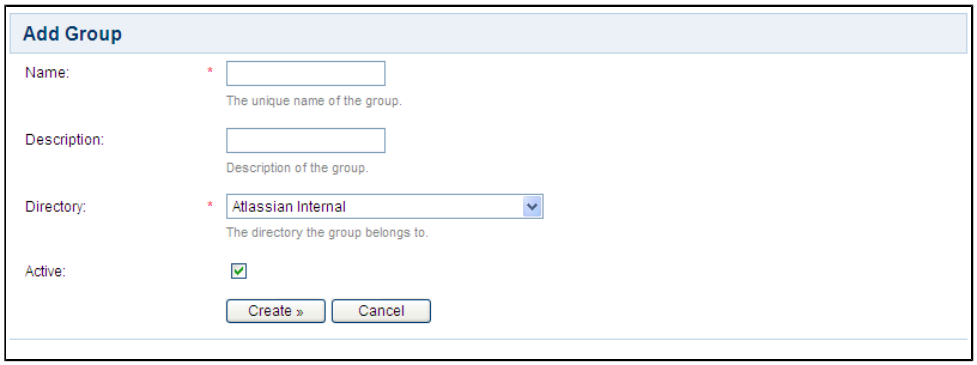
   You can now add users to the new group or role. If your directory supports nested groups, you can now add sub-groups.

Field	Description
Name	The unique name of the group or role. Within a given directory, the Name must be unique. Note that the Name cannot be changed once the group or role is created.
Description	A short description of the group or role.
Directory	The directory to which the group or role will be added. Note that the group or role 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 or role.

Screenshot 1: 'Group Browser'



Screenshot 2: 'Add Group'



Groups (not roles) can also be added via Crowd's migration tools — see Importing Users and Groups into a Directory.

See Also

Specifying which Groups can access an Application

RELATED TOPICS

- Using the User Browser
- Adding a User
- Deleting or Deactivating a User
- Managing a User's Session
- Editing a User's Details and Password
- Specifying a User's Attributes
- Editing a User's Group and Role Membership
- Granting Crowd Administration Rights to a User
- Granting Crowd User Rights to a User
- Using the Group Browser and Role Browser
- Adding a Group or Role
- Deleting or Deactivating a Group
- Viewing Members of a Group
    - Nested Groups in Crowd
    - Adding a Sub-Group
    - Removing a Sub-Group

Crowd Documentation

Deleting or Deactivating a Group

This page last changed on May 08, 2008 by smaddox.

Deactivating a group prevents its members from logging in to any applications that use the Crowd framework. Deleting a group removes it completely from the relevant directory.

To deactivate a group,

1. Log in to the Crowd Administration Console.
2. Click the 'Groups' tab in the top navigation bar.
3. This will display the Group Browser. Select the relevant directory, locate the group you wish to deactivate, and click the 'View' link that corresponds to the group.
4. This will display the 'Group Details' screen. Deselect the 'Active' check-box, then click the 'Update' button.

To delete a group,

1. Log in to the Crowd Administration Console.
2. Click the 'Groups' tab in the top navigation bar.
3. This will display the Group Browser. Select the relevant directory, locate the group you wish to deactivate, and click the 'View' link that corresponds to the group.
4. This will display the 'Group Details' screen. Click 'Remove Group' in the left-hand menu.

RELATED TOPICS

- Using the User Browser
- Adding a User
- Deleting or Deactivating a User
- Managing a User's Session
- Editing a User's Details and Password
- Specifying a User's Attributes
- Editing a User's Group and Role Membership
- Granting Crowd Administration Rights to a User
- Granting Crowd User Rights to a User
- Using the Group Browser and Role Browser
- Adding a Group or Role
- Deleting or Deactivating a Group
- Viewing Members of a Group
  - Nested Groups in Crowd
  - Adding a Sub-Group
  - Removing a Sub-Group

Crowd Documentation

Viewing Members of a Group

This page last changed on May 07, 2008 by smaddox.

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.

About nested groups

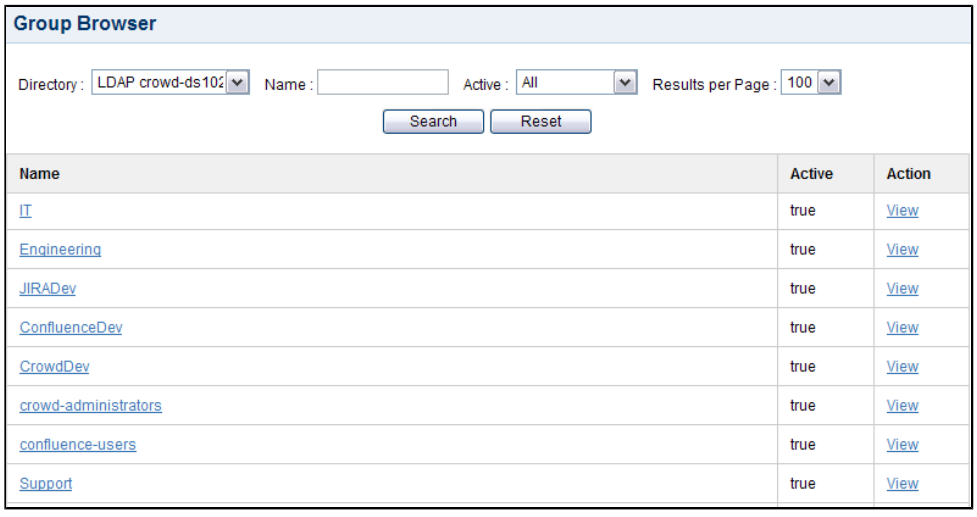
Some user directories allow you to define a group as a member of another group. Groups in such a structure are called 'nested groups'. In Crowd, you can map any group to an application, including a group which contains other groups. Currently, nested groups are supported for LDAP directory connectors only.

For more details about nested groups, refer to Nested Groups in Crowd.

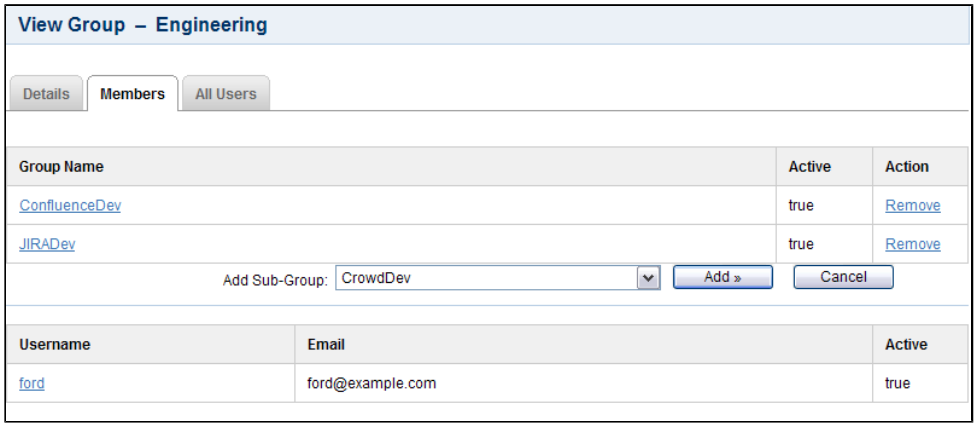
To view the members of a group,

1. Log in to the Crowd Administration Console.
2. Click the 'Groups' tab in the top navigation bar.
3. The Group Browser will appear, as shown in screenshot 1 below. Select the directory in which you are interested, then click the 'Search' button to list all the groups that exist in that directory. You can refine your search by specifying a 'Name' (note that this is case-sensitive), or 'Active'/'Inactive' groups.
4. Click the 'View' link to view a specific group's details.
5. The 'View Group — Details' screen will appear. Click the '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 '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 'Members' tab will show only users.
6. Click the 'All Users' tab (if present) to view all users who are included in the group and in its sub-groups, as shown in screenshot 3 below.
   - The 'All Users' tab will appear only if the group you are viewing contains sub-groups.
   - This tab shows users belonging to the group you are viewing, plus the users belonging to all its sub-groups.

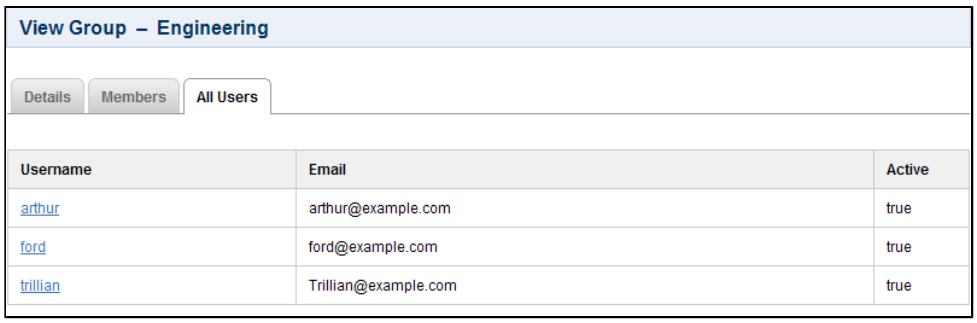
Screenshot 1: Group Browser



Screenshot 2: Viewing Members of a Group



Screenshot 3: Viewing All Users in a Group



RELATED TOPICS

Using the Group Browser and Role Browser
Adding a Group or Role
Crowd Documentation

Nested Groups in Crowd

This page last changed on May 07, 2008 by smaddox.

This page describes the way Crowd handles nested groups, i.e. groups which contain other groups as members and groups which are members of other groups.

On this page:

Error formatting macro: toc: java.lang.NullPointerException

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. Currently, nested groups are supported for LDAP directory connectors only.

Here's the effect on authorisation 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.

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

In Crowd 1.4, nested groups are supported for LDAP directory connectors only.

Nested groups are not supported for internal directories, delegated authentication directories or custom directory connectors.

Please vote for CWD-980 if you would like a future version of Crowd to support nested groups in internal and delegated authentication directories.

Use Case: Importing from LDAP into a Delegated Authentication Directory

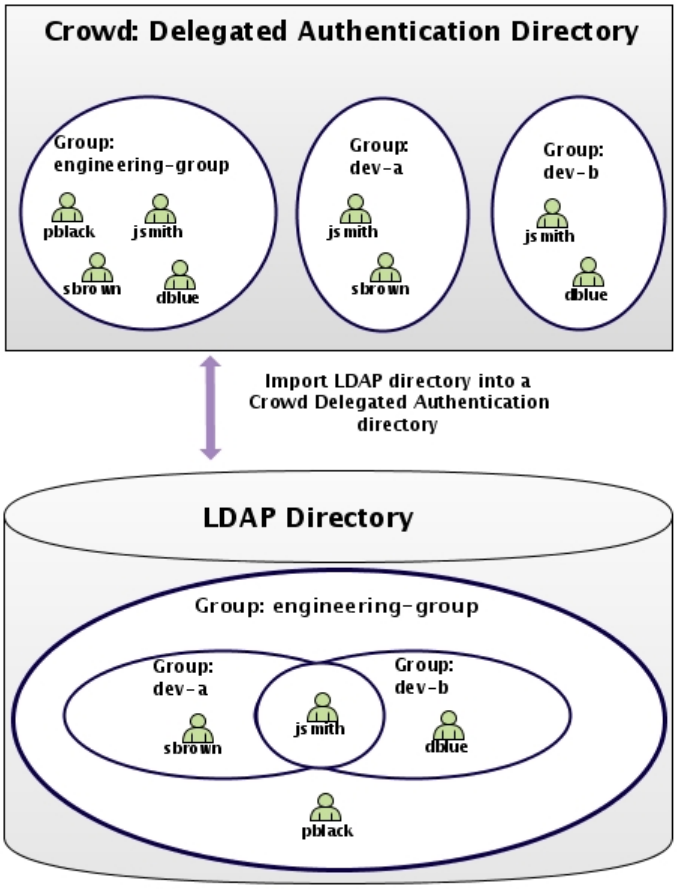
If you import an LDAP directory into a delegated authentication directory, Crowd will load all users and all their group memberships into the delegated authentication directory. Each group will contain all users belonging to the group and all users from the group's sub-groups.

For example:

- In LDAP we have group 'engineering-group' and sub-groups 'dev-a' and 'dev-b'. Memberships are:
  - engineering-group — sub-groups: dev-a, dev-b; users: pblack

    - dev-a — users: jsmith, sbrown
    - dev-b — users: jsmith, dblue
- After importing to a Crowd delegated authentication directory, we have groups 'engineering-group', 'dev-a' and 'dev-b'. Memberships are:
    - engineering-group — users: pblack, jsmith, sbrown, dblue
    - dev-a — users: jsmith, sbrown
    - dev-b — users: jsmith, dblue

Diagram 1: Importing from LDAP into a Delegated Authentication Directory



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.

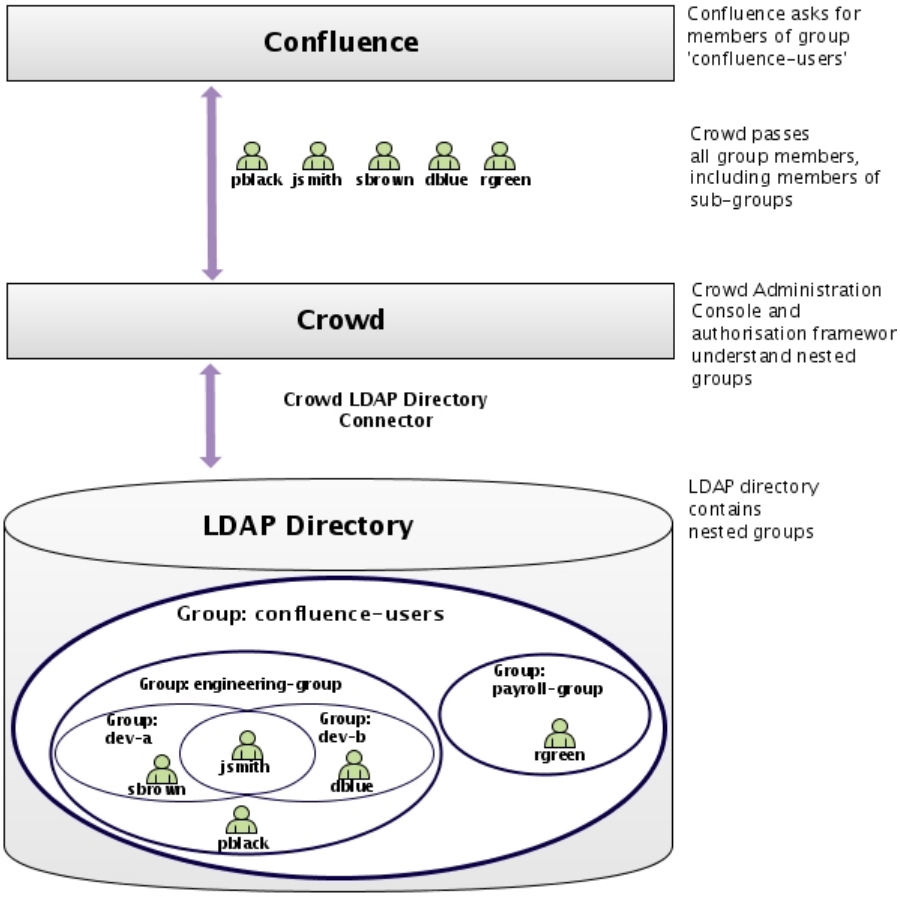
Use Case: Confluence Requests a List of Users in 'confluence-users' group

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 2: Presenting Flattened Lists of Users to Integrated Applications



User Management via Integrated Applications

Recommendation: Enable External User Management

If you have JIRA, Confluence, 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.

RELATED TOPICS

Using the Group Browser and Role Browser
Adding a Group or Role
Viewing Members of a Group
Adding a Sub-Group
Removing a Sub-Group
Crowd Documentation

Adding a Sub-Group

This page last changed on May 08, 2008 by smaddox.

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. Currently, nested groups are supported for LDAP directory connectors only.

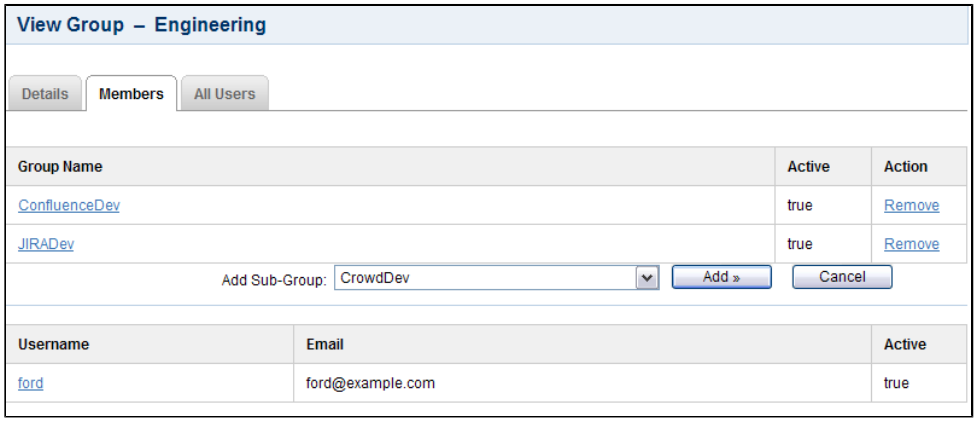
For more details about nested groups, refer to Nested Groups in Crowd.

To add a sub-group,

1. Log in to the Crowd Administration Console.
2. Click the 'Groups' tab in the top navigation bar.
3. The Group Browser will appear. Select the directory in which you are interested, then click the 'Search' button to list all the groups that exist in that directory.
   You can refine your search by specifying a 'Name' (note that this is case-sensitive), or 'Active'/'Inactive' groups.
4. If the sub-group does not yet exist in the directory, add it now:
   - Click 'Add Group' in the left-hand menu.
   - Complete the fields as described in Adding a Group or Role, then click the 'Create' button.
5. 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 'View' link next to that group.
6. The 'View Group — Details' screen will appear. Click the 'Members' tab.
7. Select the sub-group from the 'Add Sub-Group' dropdown list and click the 'Add' button.

   The 'Add Sub-Group' dropdown list will appear only if your user directory supports nested groups.

Screenshot: Adding a Sub-Group



RELATED TOPICS

Nested Groups in Crowd
Using the Group Browser and Role Browser
Adding a Group or Role
Crowd Documentation

Removing a Sub-Group

This page last changed on May 08, 2008 by smaddox.

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

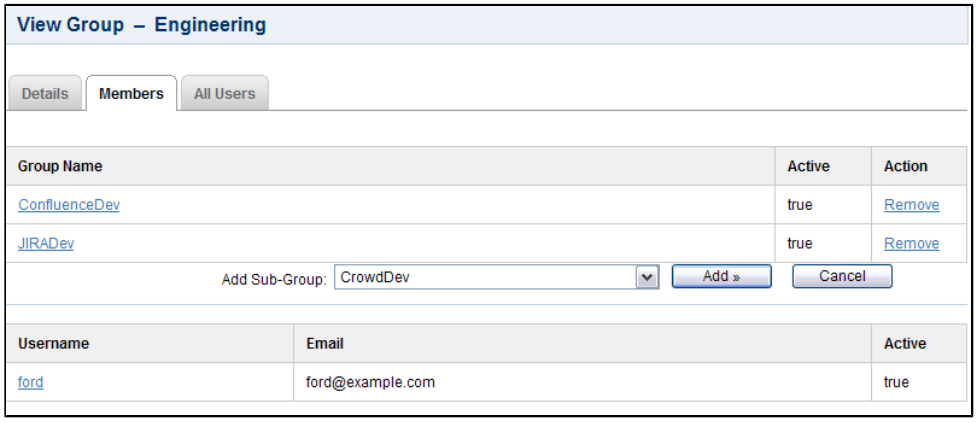
Some user directories allow you to define a group as a member of another group. Groups in such a structure are called 'nested groups'. In Crowd, you can map any group to an application, including a group which contains other groups. Currently, nested groups are supported for LDAP directory connectors only.

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. Click the 'Groups' tab in the top navigation bar.
3. The Group Browser will appear. Select the directory in which you are interested, then click the 'Search' button to list all the groups that exist in that directory.
   You can refine your search by specifying a 'Name' (note that this is case-sensitive), or 'Active'/'Inactive' groups.
4. Find the parent group in the list of groups and click the 'View' link next to that group.
5. The 'View Group — Details' screen will appear. Click the 'Members' tab.
6. Click the 'Remove' link next to the sub-group whose group membership you want to remove.

Screenshot: Removing a Sub-Group



RELATED TOPICS

Nested Groups in Crowd
Adding a Sub-Group
Using the Group Browser and Role Browser
Crowd Documentation

System Administration

This page last changed on May 05, 2008 by smaddox.

Configuring Server Settings

This page last changed on May 05, 2008 by smaddox.

You can alter the settings which were specified when your Crowd server was installed:

- Deployment Title
- Domain
- Token Seed
- Session Configuration
- Caching
- Compression of Server Output
- Licensing

RELATED TOPICS

- Configuring Server Settings
  - Deployment Title
  - Domain
  - Token Seed
  - Session Configuration
  - Caching
    - Configuring Caching for an Application
  - Compression of Server Output
  - Licensing
- Configuring SMTP Email
  - Creating an Email Notification Template
- Viewing Crowd's System Information
- Backing Up and Restoring Data
- Logging and Profiling
  - Performance Profiling

Crowd Documentation

Deployment Title

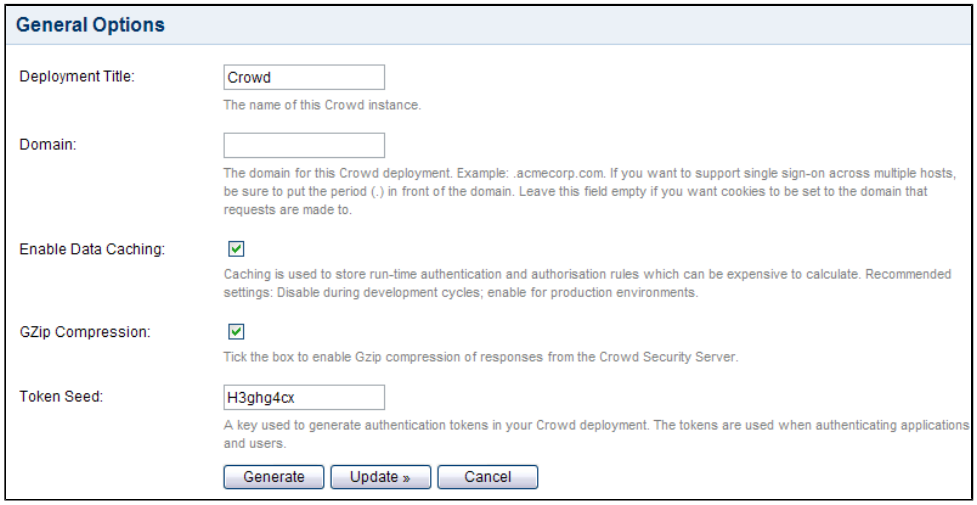
This page last changed on May 05, 2008 by smaddox.

The deployment title specifies a unique name for your Crowd instance. The deployment title can be used when sending email notifications.

To specify the deployment title,

1. Log in to the Crowd Administration Console.
2. Click the 'Administration' tab in the top navigation bar.
3. The 'General Options' screen will appear. Type the new name into the 'Deployment Title' field.
4. Click the 'Update' button.

Screenshot: 'General Options'



RELATED TOPICS

- Configuring Server Settings
  - Deployment Title
  - Domain
  - Token Seed
  - Session Configuration
  - Caching
    - Configuring Caching for an Application
  - Compression of Server Output
  - Licensing
- Configuring SMTP Email
  - Creating an Email Notification Template
- Viewing Crowd's System Information
- Backing Up and Restoring Data
- Logging and Profiling
  - Performance Profiling

Crowd Documentation

Domain

This page last changed on May 05, 2008 by smaddox.

The domain is used when setting HTTP authentication cookies in a user's browser. If this attribute is not correct, single sign-on will not work when the user switches between applications.

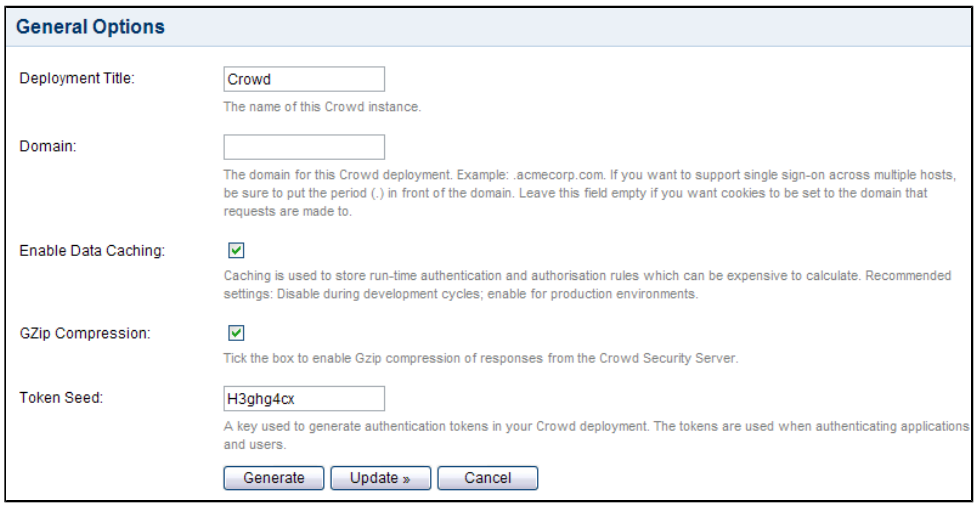
Note:

- When developing on your local machine, the domain should be set to `localhost`.
- If you wish to have single sign-on (SSO) support for *.mydomain.com, you will need to set the domain to `.mydomain.com`. Please note the full stop ('.') before the top-level-domain.

To specify the domain,

1. Log in to the Crowd Administration Console.
2. Click the 'Administration' tab in the top navigation bar.
3. The 'General Options' screen will appear. Type the new domain into the 'Domain' field.
4. Click the 'Update' button.

Screenshot: 'General Options'



RELATED TOPICS

- Configuring Server Settings
  - Deployment Title
  - Domain
  - Token Seed
  - Session Configuration
  - Caching
    - Configuring Caching for an Application
  - Compression of Server Output
  - Licensing
- Configuring SMTP Email
  - Creating an Email Notification Template
- Viewing Crowd's System Information
- Backing Up and Restoring Data
- Logging and Profiling
  - Performance Profiling

Crowd Documentation

Token Seed

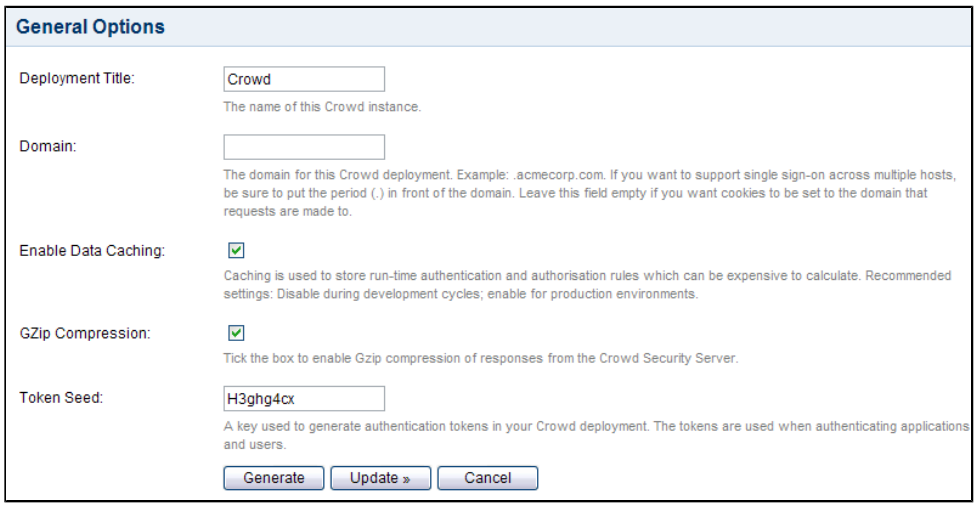
This page last changed on May 05, 2008 by smaddox.

The token seed is a unique key for each site deployment of Crowd. This key is used when generating tokens for an authenticated application.

To specify the token seed,

1. Log in to the Crowd Administration Console.
2. Click the 'Administration' tab in the top navigation bar.
3. The 'General Options' screen will appear. Now you can either:
   - Type the new key into the 'Token Seed' field, then click the 'Update' button.
     OR
   - Click the 'Generate' button to create a random key automatically.

Screenshot: 'General Options'



RELATED TOPICS

- Configuring Server Settings
  - Deployment Title
  - Domain
  - Token Seed
  - Session Configuration
  - Caching
    - Configuring Caching for an Application
  - Compression of Server Output
  - Licensing
- Configuring SMTP Email
  - Creating an Email Notification Template
- Viewing Crowd's System Information
- Backing Up and Restoring Data
- Logging and Profiling
  - Performance Profiling

Crowd Documentation

Session Configuration

This page last changed on May 05, 2008 by smaddox.

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. Click the 'Administration' tab in the top navigation bar.
3. Click 'Session Config' in the left-hand menu.
4. The 'Session Config' screen will appear, as shown below. Type the new value into the 'Session Timeout' field, then click the 'Update' button.

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 some significant drawbacks:

- Sessions will not be saved across Crowd restarts. If you restart Crowd, all your users will have to log in again.
- Clustering will not be possible. Atlassian does not officially support clustering Crowd, but a number of our customers are successfully using it in this manner.

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:

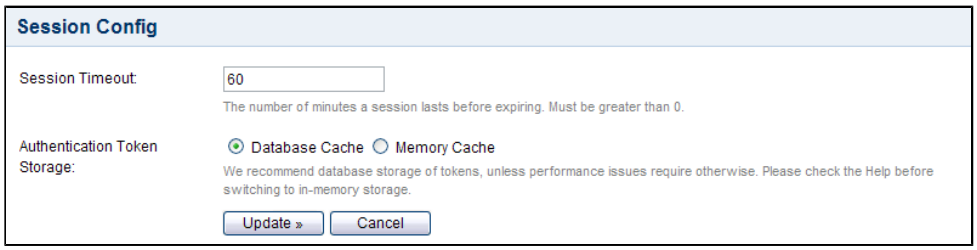
Number of Tokens:	100	500	1000	5000	10000
Database -> Memory	0.1s	0.7s	1.2s	4.2s	8.2s
Memory -> Database	1.2s	4.8s	9.2s	45s	90s

To switch the token storage location,

1. Log in to the Crowd Administration Console.
2. Click the 'Administration' tab in the top navigation bar.
3. Click 'Session Config' in the left-hand menu.
4. The 'Session Config' screen will appear, as shown below. Select one of the radio buttons next to Authentication Token Storage:
   - '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.
   - 'Memory Cache' — Select this option to store your tokens in memory.

5. Click the 'Update' button.

Screenshot: 'Session Config'



RELATED TOPICS

- Managing an Application's Session
- Managing a User's Session

- Configuring Server Settings
  - Deployment Title
  - Domain
  - Token Seed
  - Session Configuration
  - Caching
    - Configuring Caching for an Application
  - Compression of Server Output
  - Licensing
- Configuring SMTP Email
  - Creating an Email Notification Template
- Viewing Crowd's System Information
- Backing Up and Restoring Data
- Logging and Profiling
  - Performance Profiling

Crowd Documentation

Caching

This page last changed on May 05, 2008 by smaddox.

Caching is used to store run-time authentication and authorisation rules, which can be expensive to calculate. We recommend that you turn caching off during development cycles, and enable caching for production use.

In Crowd, data caching occurs in two main areas:

- The Crowd server itself — certain parts of the Crowd Administration Console application are stored in a local cache to improve performance.
- The applications that are connected to Crowd — e.g. JIRA, Confluence and Bamboo. These 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.

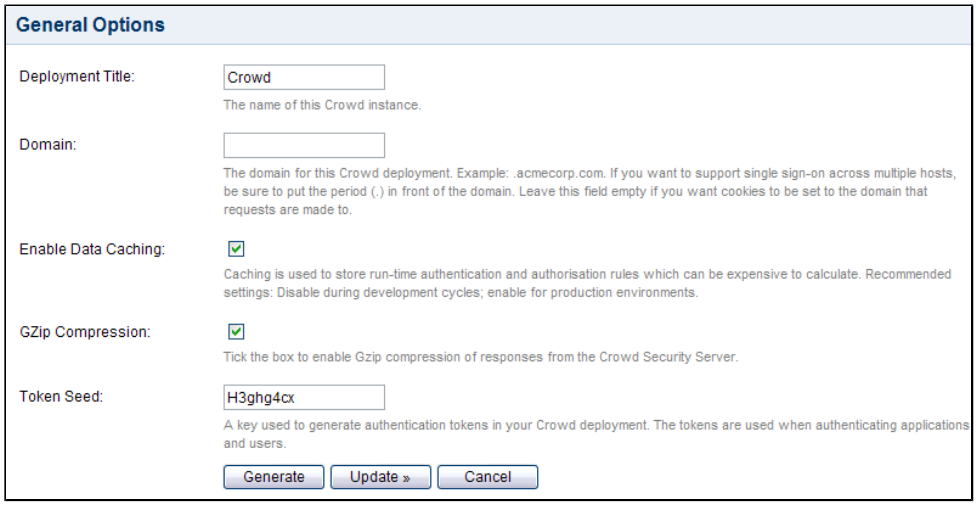
The configuration option described below manages both types of caching.

To fine-tune how caching works for your Crowd-integrated applications, please see Configuring Caching for an Application.

To enable data caching,

1. Log in to the Crowd Administration Console.
2. Click the 'Administration' tab in the top navigation bar.
3. The 'General Options' screen will appear. Put a tick in the 'Enable Data Caching' checkbox.
4. Click the 'Update' button.

Screenshot: 'Caching'



RELATED TOPICS

- Configuring Server Settings
  - Deployment Title
  - Domain
  - Token Seed
  - Session Configuration
  - Caching
    - Configuring Caching for an Application
  - Compression of Server Output
  - Licensing
- Configuring SMTP Email
  - Creating an Email Notification Template
- Viewing Crowd's System Information

- Backing Up and Restoring Data
- Logging and Profiling
    - Performance Profiling

Crowd Documentation

Configuring Caching for an Application

This page last changed on May 05, 2008 by smaddox.

The page 5.2.6 Caching does not exist.

To enable server caching, please see 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.

Below is a small snippet of the `crowd-ehcache.xml` file.

```
<ehcache>

    <diskStore path="java.io.tmpdir"/>

    <defaultCache
        maxElementsInMemory="50000"
        eternal="false"
        overflowToDisk="false"
        timeToIdleSeconds="300"
        timeToLiveSeconds="300"
        diskPersistent="false"
        diskExpiryThreadIntervalSeconds="120"/>

</ehcache>
```

Some basic features of defaultCache:

- 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.
- timeToIdleSeconds: 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 set 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)`

RELATED TOPICS

Unable to render {children} Page not found: 5. System Administration
Crowd Documentation

Compression of Server Output

This page last changed on May 05, 2008 by smaddox.

By default, Crowd compresses the output from the security server, using the Gzip compression format, before sending the data to the client over the network. Compression of server output is optional. You can turn it on or off via the Crowd Administration Console.

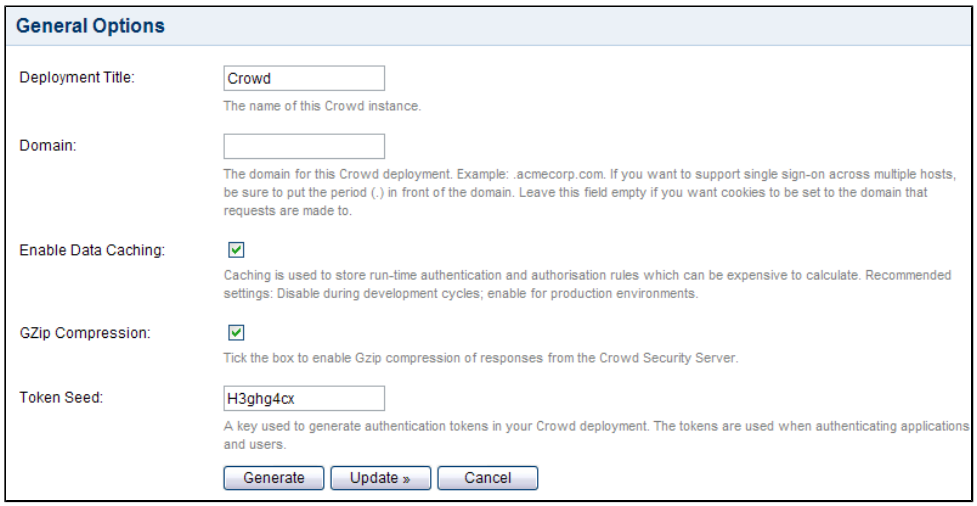
Here are some reasons why you may want to turn compression off:

- It may be easier to debug problems using uncompressed data.
- Some agents, such as older versions of Internet Explorer, have problems with the Gzip format.

To enable/disable compression of server output,

1. Log in to the Crowd Administration Console.
2. Click the 'Administration' tab in the top navigation bar.
3. The 'General Options' screen will appear. Set the 'Gzip Compression' option as follows:
    - Put a tick in the checkbox to instruct the Crowd Security Server to use Gzip compression when sending responses.
    - Leave the checkbox empty to instruct Crowd to send uncompressed data.

Screenshot: 'Setting the Compression of Server Output'



RELATED TOPICS

- Configuring Server Settings
    - Deployment Title
    - Domain
    - Token Seed
    - Session Configuration
    - Caching
        - Configuring Caching for an Application
    - Compression of Server Output
    - Licensing
- Configuring SMTP Email
    - Creating an Email Notification Template
- Viewing Crowd's System Information
- Backing Up and Restoring Data
- Logging and Profiling
    - Performance Profiling

Crowd Documentation

Licensing

This page last changed on May 05, 2008 by smaddox.

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.

Note:

If the number of users who are allowed to log in to the Crowd framework exceeds the user license limit, no-one will be able to log in to any applications (other than the Crowd Administration Console). If this happens, you can obtain a temporary license from Atlassian.

To minimise 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.

To enter your license key,

1. Log in to the Crowd Administration Console.
2. Click the 'Administration' tab in the top navigation bar.
3. Click 'Licensing' in the left-hand menu.
4. Type (or paste) your license key into the 'License' field.
5. Click the 'Update' button.

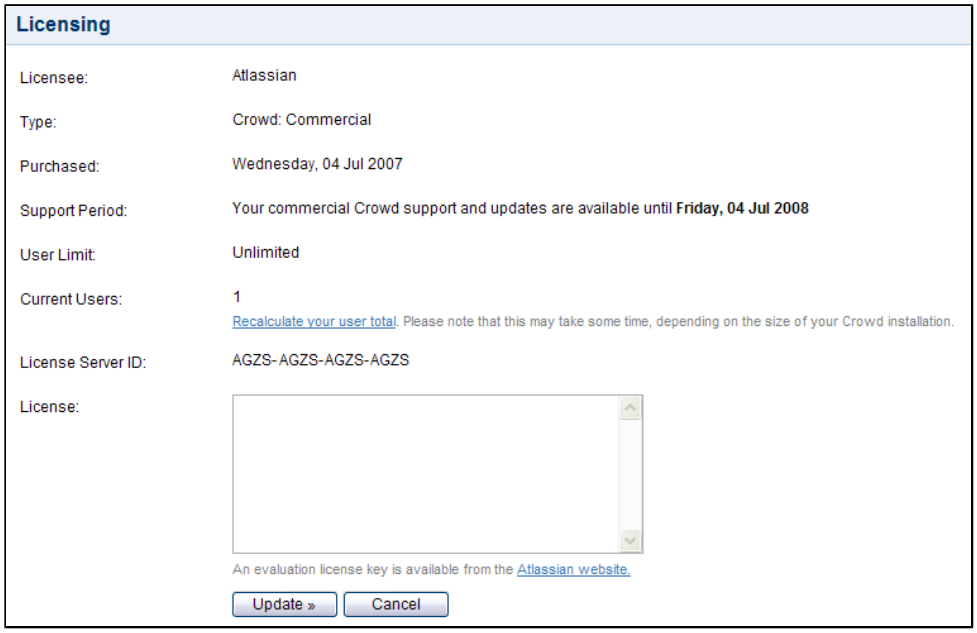
Your Server ID is generated automatically, based on your license key.

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. Click the 'Administration' tab in the top navigation bar.
3. Click 'Licensing' in the left-hand menu.
4. Click the link labelled 'Recalculate your user total'.
   The recalculation may take a while, depending on the size of your user base.

Screenshot: 'Licensing'



RELATED TOPICS

- Configuring Server Settings
  - Deployment Title
  - Domain
  - Token Seed
  - Session Configuration
  - Caching
    - Configuring Caching for an Application
  - Compression of Server Output
  - Licensing
- Configuring SMTP Email
  - Creating an Email Notification Template
- Viewing Crowd's System Information
- Backing Up and Restoring Data
- Logging and Profiling
  - Performance Profiling

Crowd Documentation

Configuring SMTP Email

This page last changed on May 05, 2008 by smaddox.

Once you have configured SMTP email as described below, Crowd can send email notifications to users at specific events, such as when a user's password is reset or a server event occurs.

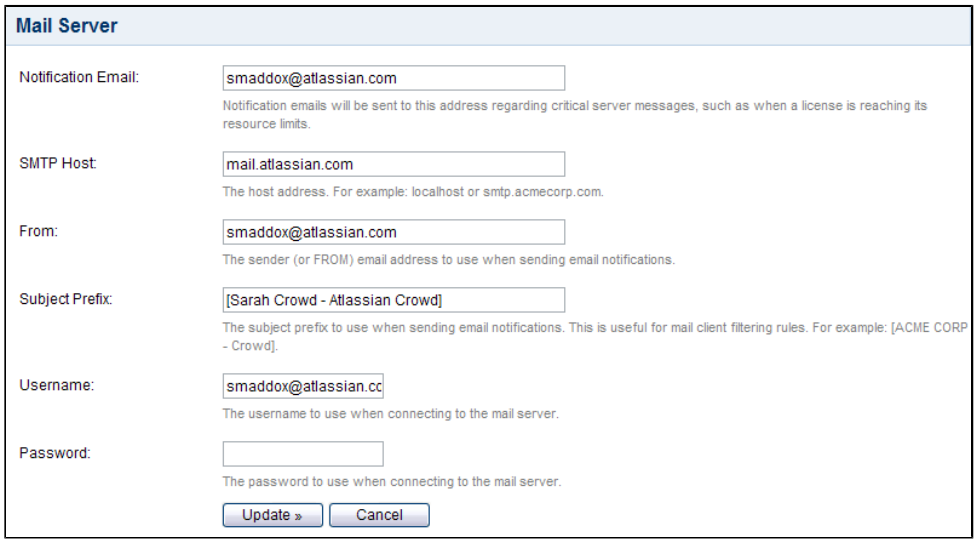
To configure SMTP email,

1. Log in to the Crowd Administration Console.
2. Click the 'Administration' tab in the top navigation bar.
3. Click 'Mail Server' in the left-hand menu.
4. Enter the details of your mail server, and the username and password (if required) that Crowd will use to log in to your mail server, then click the 'Update' button:

- Notification Email — The email address which will receive notifications about server events.
- SMTP Host — The hostname of the SMTP mail server, e.g. 'localhost' or 'smtp.acme.com'.
- From — The email address from which password notifications will be 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.
- 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.

To customise the password notification message, please see Creating an Email Notification Template.

Screenshot: 'Mail Server'



RELATED TOPICS

- Configuring Server Settings
  - Deployment Title
  - Domain
  - Token Seed
  - Session Configuration
  - Caching
    - Configuring Caching for an Application
  - Compression of Server Output
  - Licensing
- Configuring SMTP Email
  - Creating an Email Notification Template
- Viewing Crowd's System Information
- Backing Up and Restoring Data
- Logging and Profiling
  - Performance Profiling

Crowd Documentation

Creating an Email Notification Template

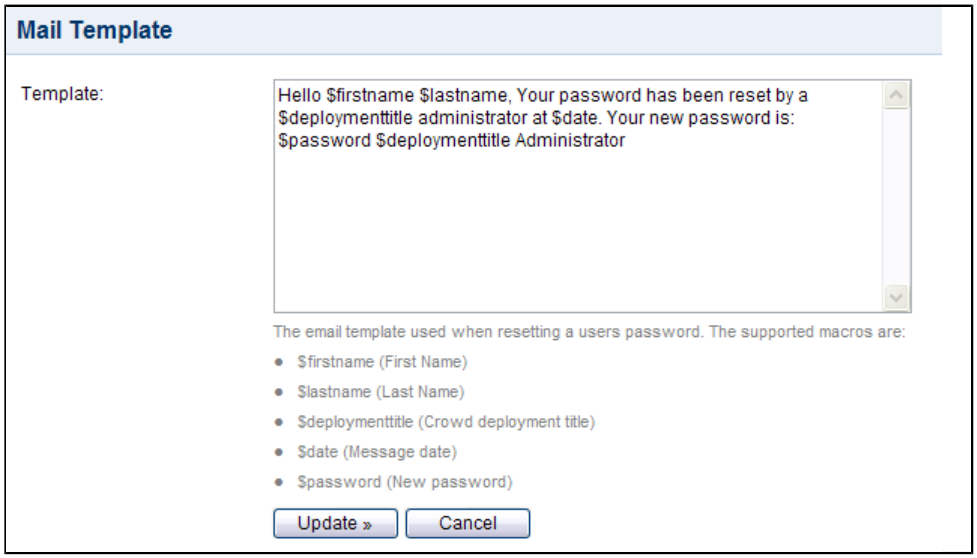
This page last changed on May 05, 2008 by smaddox.

The email template is used when sending a notification to a user, e.g. when resetting a user's password.

To set up your email template,

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. Enter the text and macros that will be used to compose the body of the email message. The following macros are available:
   - $firstname — will be replaced by the user's first name.
   - $lastname — will be replaced by the user's last name.
   - $deploymenttitle — will be replaced by the title of your Crowd deployment, as defined in Deployment Title.
   - $date — will be replaced by the date/time of the message event.
   - $password — will be replaced by the user's new password.
5. Click the 'Update' button.

Screenshot: 'Mail Template'



RELATED TOPICS

- Configuring Server Settings
  - Deployment Title
  - Domain
  - Token Seed
  - Session Configuration
  - Caching
    - Configuring Caching for an Application
  - Compression of Server Output
  - Licensing
- Configuring SMTP Email
  - Creating an Email Notification Template
- Viewing Crowd's System Information
- Backing Up and Restoring Data
- Logging and Profiling
  - Performance Profiling

Crowd Documentation

Viewing Crowd's System Information

This page last changed on May 07, 2008 by smaddox.

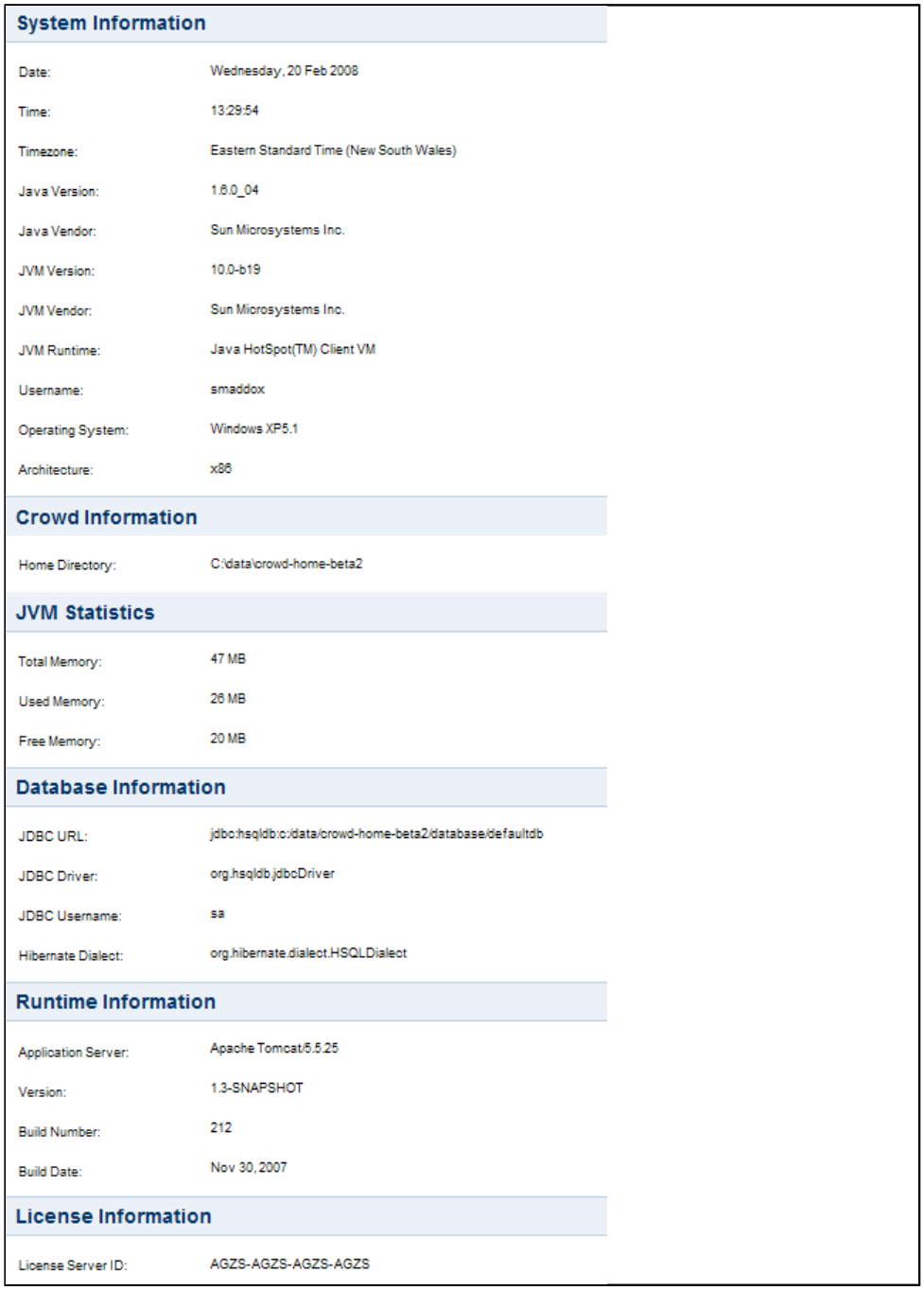
Crowd provides a useful summary of your server's system information, including:

- 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. Click the 'Administration' tab in the top navigation bar.
3. Click 'System Info' in the left-hand menu.

Screenshot: 'System Information'



System Information

Date:	Wednesday, 20 Feb 2008
Time:	13:29:54
Timezone:	Eastern Standard Time (New South Wales)
Java Version:	1.6.0_04
Java Vendor:	Sun Microsystems Inc.
JVM Version:	10.0-b19
JVM Vendor:	Sun Microsystems Inc.
JVM Runtime:	Java HotSpot(TM) Client VM
Username:	smaddox
Operating System:	Windows XP5.1
Architecture:	x86

Crowd Information

Home Directory:	C:\data\crowd-home-beta2

JVM Statistics

Total Memory:	47 MB
Used Memory:	26 MB
Free Memory:	20 MB

Database Information

JDBC URL:	jdbc:hsqldb:c:/data/crowd-home-beta2/database/defaultdb
JDBC Driver:	org.hsqldb.jdbcDriver
JDBC Username:	sa
Hibernate Dialect:	org.hibernate.dialect.HSQLDialect

Runtime Information

Application Server:	Apache Tomcat/5.5.25
Version:	1.3-SNAPSHOT
Build Number:	212
Build Date:	Nov 30, 2007

License Information

License Server ID:	AGZS-AGZS-AGZS-AGZS

RELATED TOPICS

- Configuring Server Settings
  - Deployment Title
  - Domain
  - Token Seed
  - Session Configuration
  - Caching
    - Configuring Caching for an Application
  - Compression of Server Output
  - Licensing
- Configuring SMTP Email
  - Creating an Email Notification Template
- Viewing Crowd's System Information
- Backing Up and Restoring Data
- Logging and Profiling

- Performance Profiling

Crowd Documentation

Backing Up and Restoring Data

This page last changed on May 07, 2008 by smaddox.

You can back up your Crowd data by exporting it to an XML file. 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

At present, Crowd does not allow you to schedule periodic backups. We do have an open feature request for this. Until this feature is added to Crowd, we recommend using alternative backup methods such as:

- A periodic backup or dump of your database using tools provided by your database.
- A backup of your Crowd Home directory using external backup tools.

We recommend that you back up your data regularly, especially after any significant configuration changes. You should also perform regular backups of your database.

To back up your Crowd data,

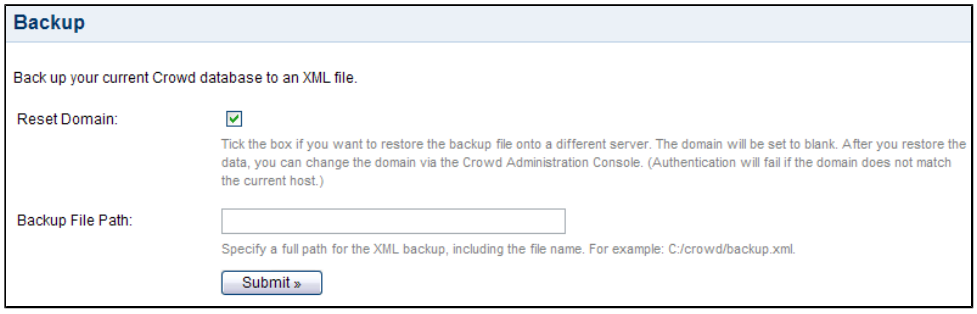
1. Log in to the Crowd Administration Console.
2. Click the 'Administration' tab in the top navigation bar.
3. Click 'Backup' in the left-hand menu.
4. Select the 'Reset Domain' checkbox if the backup file will be restored onto a different server. Selecting 'Reset Domain' will reset the domain to blank. (After you restore the data, you can change the domain as described in Domain.)
5. Type an appropriate 'Backup File Path', including the name of the XML file.
6. Click the 'Submit' button.

To restore your Crowd data,

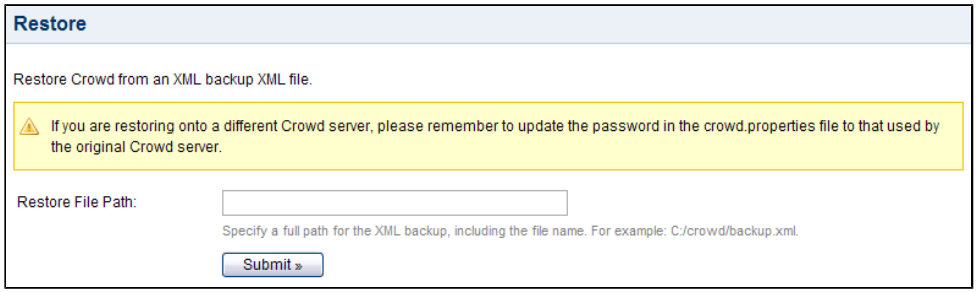
Before you begin: If you created the XML backup file on a different server, edit the `crowd.properties` file and change the password to match the password of the server on which you created the XML backup file.

1. Log in to the Crowd Administration Console.
2. Click the 'Administration' tab in the top navigation bar.
3. Click 'Restore' in the left-hand menu.
4. In the 'Restore File Path' field, type the path to the backup file, including the name of the XML file.
5. Click the 'Submit' button.

Screenshot 1: 'Backup'



Screenshot 2: 'Restore'



RELATED TOPICS

- Configuring Server Settings
  - Deployment Title
  - Domain
  - Token Seed
  - Session Configuration
  - Caching
    - Configuring Caching for an Application
  - Compression of Server Output
  - Licensing
- Configuring SMTP Email
  - Creating an Email Notification Template
- Viewing Crowd's System Information
- Backing Up and Restoring Data
- Logging and Profiling
  - Performance Profiling

Crowd Documentation

Logging and Profiling

This page last changed on May 07, 2008 by smaddox.

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.

On this page:

Error formatting macro: toc: java.lang.NullPointerException

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	Fine-grained information that is useful for debugging only. These are low-level details that most people never need to know about.
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

Provided that you have not changed the log file location from the default, the Crowd log file is at the following location:

- For Standalone installations of Crowd: `{CROWD-STANDALONE-INSTALL}/atlassian-crowd.log`
- For WAR installations: The directory on your application server, from which the Crowd application was started.

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.
- Or 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.

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 behaviour 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.

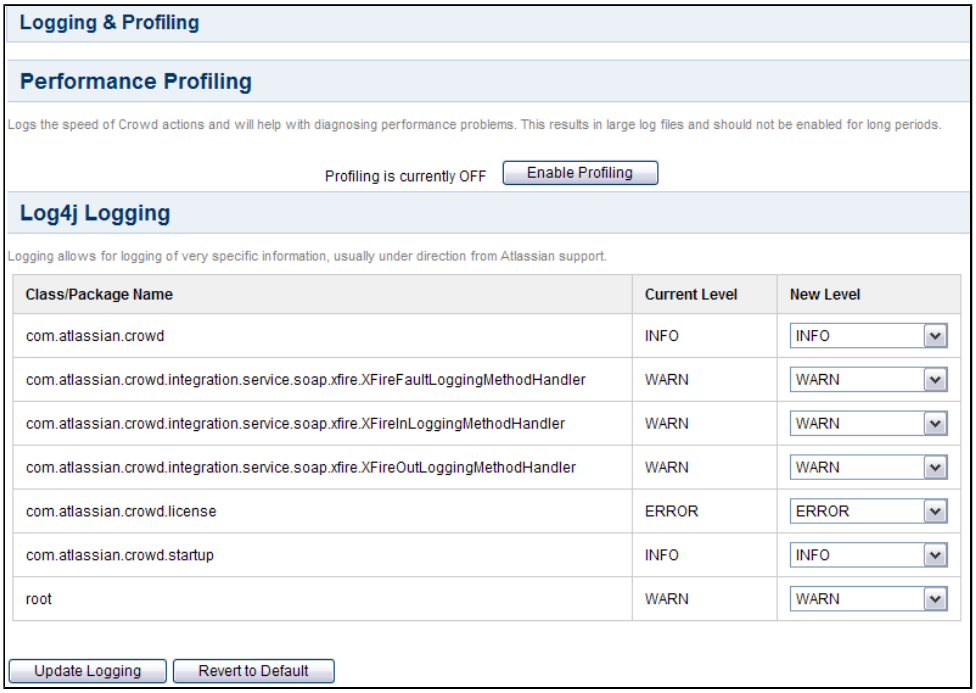
  With performance profiling turned on, your system output console will show a record of the time it takes (in milliseconds) to complete each Crowd action. This will help with diagnosing performance problems. The resulting output will be large, so you should not enable it for long periods.

  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. Click the 'Administration' tab in the top navigation bar.
3. Click 'Logging & Profiling' in the left-hand menu.
4. 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.
5. Click the 'Update Logging' button to save any changes you have made in the 'Log4j Logging' section.

Screenshot: Changing Log Levels and Profiling



Description of the loggers:

Logger	Description
`com.atlassian.crowd`	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.
`com.atlassian.crowd....XFireFaultLoggingMethodHandler`	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.
`com.atlassian.crowd....XFireOutLoggingMethodHandler`	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.
`com.atlassian.crowd....XFireInLoggingMethodHandler`	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.
`com.atlassian.crowd.license`	Useful for troubleshooting certain licensing issues in Crowd.
`com.atlassian.crowd.startup`	Can be helpful for troubleshooting startup errors in Crowd.
`root`	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 behaviour is defined in the following properties file:

- For Standalone installations of Crowd: `{CROWD-STANDALONE-INSTALL}/crowd-webapp/WEB-INF/classes/log4j.properties`
- For WAR installations: `{CROWD-WAR-INSTALL}/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.

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 Standalone installations of CrowdID: `{CROWDID-STANDALONE-INSTALL}/crowd-openidserver-webapp/WEB-INF/classes/log4j.properties`
- For WAR installations: `{CROWDID-WAR-INSTALL}/WEB-INF/classes/log4j.properties`

RELATED TOPICS

- Configuring Server Settings
  - Deployment Title
  - Domain
  - Token Seed
  - Session Configuration
  - Caching
    - Configuring Caching for an Application
  - Compression of Server Output
  - Licensing
- Configuring SMTP Email
  - Creating an Email Notification Template
- Viewing Crowd's System Information
- Backing Up and Restoring Data
- Logging and Profiling
  - Performance Profiling

Crowd Documentation

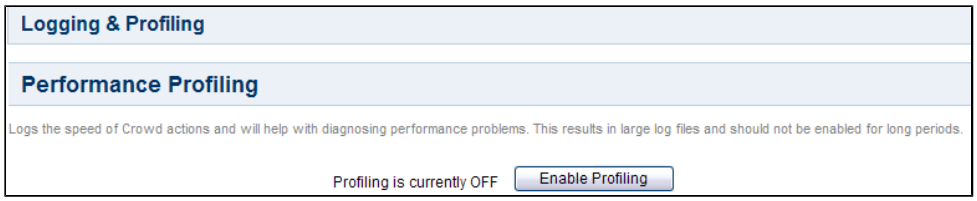
Performance Profiling

This page last changed on May 06, 2008 by smaddox.

When troubleshooting problems with your Crowd installation, it is often useful to turn on performance profiling.

To enable profiling, go to the 'Logging & Profiling' tab under 'Administration' in the Crowd Administration Console. Full instructions are in the section on logging and profiling.

Screenshot: 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.

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()
```

RELATED TOPICS

Logging and Profiling

This page last changed on May 05, 2008 by smaddox.

Connecting your own Web Applications or User Directories to Crowd

Crowd comes with a number of application and directory connectors. If you have a web application or a user directory which does not use a pre-supplied connector, your development team can create a custom connector for you.

This is a reasonably quick and easy job.

Skills Required to Write a Custom Connector for Crowd

- Familiarity with HTTP/webserver technology.
- Familiarity with the concepts of identity management, authentication and authorisation.
- Programming skills to write an application connector:
  - Medium-level experience with the Java programming language.
  - Or you can use our SOAP API in combination with a language like PHP, Ruby, etc. You need a medium-level understanding of SOAP APIs.
- Programming skills to write a directory connector:
  - Medium-level experience with the Java programming language.

Getting Help

- Atlassian partners can help you to write a custom connector and get your applications up and running. We recommend CustomWare and Appfire Technologies.
- Consult the online Crowd documentation.
- Ask questions and share tips in the Atlassian forum.
- If you have a problem with Crowd, log a ticket at Atlassian Support. Atlassian does not support customised Crowd source.

Writing the Connectors

- Creating a custom application connector
- Creating a custom directory connector

More Information on Extending Crowd

You can access Crowd through our APIs to manipulate data or integrate with your custom web applications and user directories, as described above. You can even customise the Crowd source, to change the way things work or add an entirely new feature.

To help you get started:

- When you download Crowd, you also get the source code of a demo application.
- Read the guidelines on setting up your IDE.
- For ideas, take a look at what other people are doing.
- We supply full API documentation.

A 'principal' is a 'user'

In Crowd, the term 'principal' is equivalent to the term 'user'. In Crowd 1.3.0 and later, the Crowd Administration Console uses the term 'user'. Earlier versions of Crowd, and also certain API libraries, use the term 'principal'.

Creating a Crowd Client for your Custom Application

This page last changed on May 05, 2008 by smaddox.

Crowd allows your applications to authenticate users against Crowd's user directories.

Crowd ships with ready-made connectors ('Crowd Clients') for several popular applications (see Supported Applications and Directories for the complete list). If you need to connect Crowd to one of these applications, please see Managing Applications. If you need to connect Crowd to an application that is not listed, you can achieve this by creating a Crowd Client for your application, using the SOAP API.

Creating a Crowd Client

Crowd ships with Java Client Libraries which simplify the process of communicating with the SOAP API. If you have a Java application, you can use these libraries. If you are using a language other than Java (e.g. PHP, Ruby, etc), please use the SOAP API directly.

For assistance please see:

- Application Integration Overview
  - Sample Application ('demo')
- Java Integration Libraries
  - Compiling the Crowd Source
  - Maven 2 Integration
- SOAP API
  - Axis 1.x Client Stub Generation
  - Microsoft .NET Client

A 'principal' is a 'user'

In Crowd, the term 'principal' is equivalent to the term 'user'. In Crowd 1.3.0 and later, the Crowd Administration Console uses the term 'user'. Earlier versions of Crowd, and also certain API libraries, use the term 'principal'.

Next Steps:

After creating your Crowd Client, please see Integrating Crowd with a Custom Application.

Related Topics

- Creating a Crowd Client for your Custom Application
  - Application Integration Overview
    - Sample Application ('demo')
  - Java Integration Libraries
    - Compiling the Crowd Source
    - Maven 2 Integration
  - SOAP API
    - Axis 1.x Client Stub Generation
    - Microsoft .NET Client
- Creating a Custom Directory Connector
- Crowd Developer FAQ
- IntelliJ IDEA Setup Guide
  - Setting up Tomcat in IDEA for Crowd

Crowd Documentation

Application Integration Overview

This page last changed on May 05, 2008 by smaddox.

The Crowd framework allows an application to perform authentication and authorisation calls against a mapped directory, including:

- Authenticate a principal (i.e. a user).
- Validate and invalidate an existing principal authentication.
- Find a principal by their authentication token.
- Search principals, groups and roles by name or attributes
- Add principals, groups and roles.
- Validate a principal's group and role membership.
- Add and remove principals from groups and roles.
- Update a principal's attribute data.
- Update or reset a principal's authentication credentials.

Crowd's application provisioning allows an application to be mapped to multiple directories. When an application needs to authenticate or authorise a principal, Crowd will call the directory listed first. If the security call can be processed by the directory, the operation will then return the result. If the call cannot be processed, the next directory in the list will then be used when processing the security call until all directories have been exhausted. If the security call cannot be processed, an `Exception` (based on the method) will be thrown.

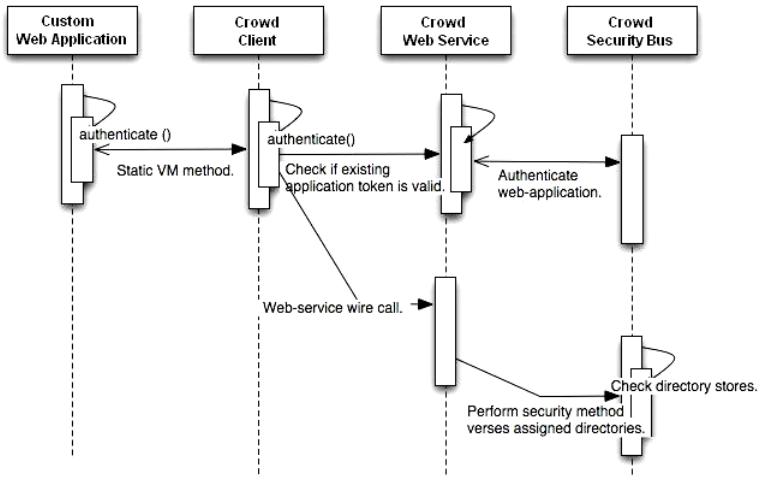
Integration Overview

When an application needs to perform a security request (that is, needs to authenticate or authorise a user) via Crowd's API, the following two steps need to occur:

1. The application authenticates itself with Crowd; the authentication token may be reused by the application during subsequent calls. During this step, Crowd validates the application's credentials and address against known application credentials/addresses.
2. Using the authenticated token from the previous step, the application then performs the security request for a particular user.

Should the application's requesting token become invalid, the client library will attempt to re-authenticate and perform the security request. If the second authentication request fails, an `Exception` will be thrown, specifying that the application's credentials are invalid.

Diagram — Application Authorisation Sequence:



Next Step

- If you are using the Java Integration Libraries, the application authorisation sequence above is fully handled by the supplied Java implementation.
- If you are using the SOAP interface, you will need to explicitly implement each step of the application authorisation sequence. As an example, please see the Microsoft .NET Client.

Related Topics

- Application Integration Overview
  - Sample Application ('demo')
- Java Integration Libraries
  - Compiling the Crowd Source
  - Maven 2 Integration
- SOAP API
  - Axis 1.x Client Stub Generation
  - Microsoft .NET Client

Crowd Documentation

Sample Application ('demo')

This page last changed on May 07, 2008 by smaddox.

To assist you when integrating your web applications, the entire sourcecode to the sample 'demo' application is included in the `src` folder of the Crowd download archive, and is (optionally) configured when you run the Setup Wizard.

The 'demo' application highlights best practices when using the Crowd framework, and can be used as an example when integrating your own web applications.

To access the 'demo' application, go to http://localhost:8095/demo.

Related Topics

- Application Integration Overview
  - Sample Application ('demo')
- Java Integration Libraries
  - Compiling the Crowd Source
  - Maven 2 Integration
- SOAP API
  - Axis 1.x Client Stub Generation
  - Microsoft .NET Client

Crowd Documentation

Java Integration Libraries

This page last changed on Feb 17, 2008 by smaddox.

This page provides sample code for creating a Crowd Client using the supplied Java Integration Libraries.

SecurityServerClient

The `SecurityServerClient` is useful for common create, update and delete operations for principals, groups and roles. To accomplish this, the `SecurityServerClient` maps 1-to-1 with the SOAP API of the Crowd server. The class reads in the `crowd.properties` configuration file from your application's class path, setting client specific details such as the Crowd server URL and SSO integration details. When the client is loaded into memory, it will then authenticate the the client application with the Crowd security server for future SOAP requests.

A full list of the available methods for the `SecurityServerClient` is available here:

- http://docs.atlassian.com/crowd/current/com/atlassian/crowd/integration/service/soap/client/SecurityServerClient.html

HttpAuthenticator

The `HttpAuthenticator` simplifies the authentication of HTTP based clients. When an authentication or invalidation is performed, the `HttpAuthenticator` manages the setting and resetting of integration variables for the principal's HTTP session. If the application has little need beyond authentication and validation, the `HttpAuthenticator` is a simple and very straightforward integration piece. Shown below is a code example of authenticating and logging off a principal:

Example 1:

```
HttpAuthenticatorFactory.getHttpAuthenticator().authenticate(request, response, username,
 password);
```

Example 2:

```
HttpAuthenticatorFactory.getHttpAuthenticator().logoff(request, response);
```

If there were any issues with the authentication or logoff calls, an `Exception` will be thrown to the application.

The `HttpAuthenticator` manages the following:

- Authenticating an HTTP request, and setting the session with the correct attributes for other integration points of the framework.
- Invalidating an HTTP request includes removing session related attributes.
- Obtaining a principal's authenticated token from a session or browser cookie.
- Validating an existing HTTP authentication for single sign-on. If another application in the same domain has already authenticated the principal, the HttpAuthenticator will attempt to validate the existing authentication.
- Building a standard AuthenticationContext for a principal. This can be used to assure the authentication is consistent across all applications when setting validation factors of the application client.

Note both the `HttpAuthenticatorFactory` and `SecurityServerClientFactory` manage singleton instances of the `HttpAuthenticator` and `SecurityServerClient` implementations respectively. You should never need to instantiate the `HttpAuthenticator` or `SecurityServerClient` manually.

If you are using an Inversion-of-Control / Dependency-Injection container in your web application (such as Spring) to manage your singletons, read the information about dependency injection lower down the page.

VerifyTokenFilter

The `VerifyTokenFilter` is an HTTP servlet filter that protects secured resources by verifying the session or cookie token is active and the principal has access to the requesting application. The token filter works in conjunction with the `HttpAuthenticator`, validating and setting various session and cookie attributes. Should the principal's token become expired or invalid due to security restrictions, the principal will be redirected to the URL provided by the `crowd.properties`.

Using the token filter is very straight forward, simply edit your `web.xml` deployment descriptor to reflect the filter and desired resource mapping:

```
<filter>
    <filter-name>VerifyTokenFilter</filter-name>
    <filter-class>com.atlassian.crowd.integration.http.VerifyTokenFilter</filter-class>
</filter>

<filter-mapping>
    <filter-name>VerifyTokenFilter</filter-name>
    <url-pattern>/secure/*</url-pattern>
</filter-mapping>
```

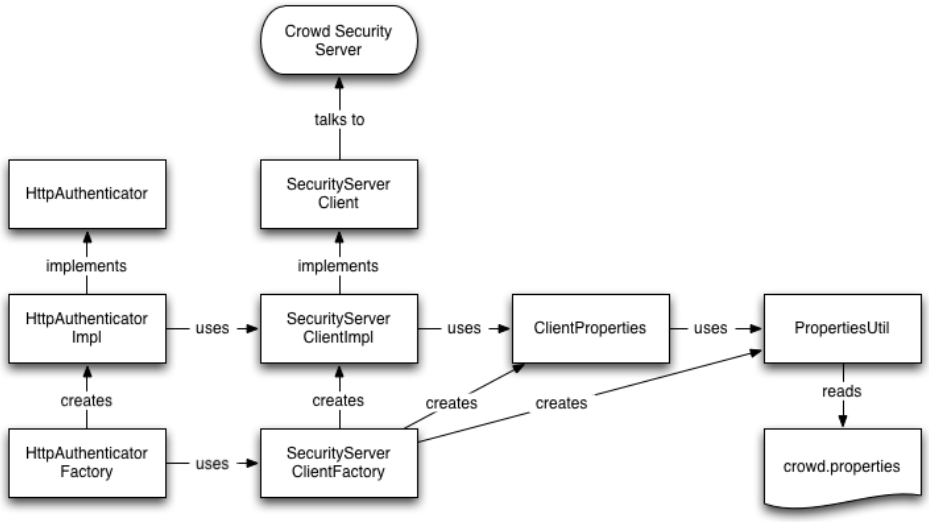
In this example, the verify token filter will prevent any pages on the /secure/ path from being accessed unless a valid token is found.

Should the token expire or be found invalid, the original URL will be stored in the principal's session at a String with the key of `VerifyTokenFilter.ORIGINAL_URL`. This is useful because, when the principal later authenticates, the original URL and parameters can then be used as a redirect bringing the principal back to their original `POST`. An example of how this can be accomplished at login is shown below:

```
HttpAuthenticatorFactory.getHttpAuthenticatory().authenticate(request, response, username,
 password);

// Check if principal was requesting a page that was prevented, if so, redirect.
String requestingPage = (String) getSession().getAttribute(VerifyTokenFilter.ORIGINAL_URL);

if (requestingPage != null) {
    // redirect the principal to the requesting page
    response().sendRedirect(requestingPage);

} else {
    // return the to the login page
    return SUCCESS;
}
```

Using dependency injection?

If you are using a dependency injection container which manages singleton instances, rather than using the `SecurityServerClientFactory` and `HttpAuthenticatorFactory` to manage singletons, you can wire up the objects themselves as shown in the following diagram:



Please use EITHER dependency injection OR the factories. Using both will result in multiple instances being maintained throughout your application.

If you are using Spring for dependency injection, a convenient `applicationContext-CrowdClient.xml` has been provided in the `crowd-integration-client.jar`. This Spring configuration file wires up the `HttpAuthenticator` and `SecurityServerClient` factory as beans named `httpAuthenticator` and `securityServerClient` respectively.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN 2.0//EN" "http://www.springframework.org/dtd/spring-
beans-2.0.dtd">

<beans>

    <bean id="propertyUtils" class="com.atlassian.crowd.util.PropertyUtils"/>

    <bean id="clientProperties"
 class="com.atlassian.crowd.integration.service.soap.client.ClientProperties">
        <constructor-arg ref="propertyUtils"/>
    </bean>

    <bean id="securityServerClient"
 class="com.atlassian.crowd.integration.service.soap.client.SecurityServerClientImpl">
        <constructor-arg ref="clientProperties"/>
    </bean>

    <bean id="httpAuthenticator"
 class="com.atlassian.crowd.integration.http.HttpAuthenticatorImpl">
        <constructor-arg ref="securityServerClient"/>
    </bean>

    <bean id="verifyTokenFilter" class="com.atlassian.crowd.integration.http.VerifyTokenFilter"
 lazy-init="true">
        <constructor-arg ref="httpAuthenticator"/>
    </bean>

    <bean id="crowdAuthenticationInterceptor"
 class="com.atlassian.crowd.integration.xwork.CrowdAuthenticationInterceptor" lazy-init="true">
        <constructor-arg ref="httpAuthenticator"/>
    </bean>

</beans>
```

To use a Spring-injected `VerifyTokenFilter` change the filter definition in your `web.xml` to:

```
<filter>
    <filter-name>verifyTokenFilter</filter-name>
    <filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>
</filter>
```

Related Topics

- Application Integration Overview
  - Sample Application ('demo')
- Java Integration Libraries
  - Compiling the Crowd Source
  - Maven 2 Integration
- SOAP API
  - Axis 1.x Client Stub Generation
  - Microsoft .NET Client

Crowd Documentation

Compiling the Crowd Source

This page last changed on Dec 11, 2007 by justen.stepka@atlassian.com.

To compile the Crowd source code you will need to have Maven 2 installed.

Once you have installed Maven 2, you will then need to do the following:

1. Copy or merge the `/atlassian-crowd-1.2.0-source/maven/conf/settings.xml` with your `~/.m2/settings.xml` maven 2 configuration file.
2. Install the JTA library available from Sun Microsystem's website into your local maven 2 repository

```
Missing:
----------
1) javax.transaction:jta:jar:1.0.1B

  Try downloading the file manually from:
      http://java.sun.com/products/jta

  Then, install it using the command:
      mvn install:install-file -DgroupId=javax.transaction -DartifactId=jta \
          -Dversion=1.0.1B -Dpackaging=jar -Dfile=/path/to/file
Alternatively, if you host your own repository you can deploy the file there:       mvn
 deploy:deploy-file -DgroupId=javax.transaction -DartifactId=jta \
          -Dversion=1.0.1B -Dpackaging=jar -Dfile=/path/to/file \
           -Durl=[url] -DrepositoryId=[id]

  Path to dependency:
        1) com.atlassian.crowd:crowd-core:jar:1.2.0
        2) javax.transaction:jta:jar:1.0.1B

----------
1 required artifact is missing.

for artifact:
  com.atlassian.crowd:crowd-core:jar:1.2.0

from the specified remote repositories:
  central (http://repo1.maven.org/maven2),
  atlassian-m1-repository (http://repository.atlassian.com),
  atlassian-proxy (https://m2proxy.atlassian.com/repository/public)
```

Once you have completed this you will be able to then run the command `mvn compile`.

Maven 2 Integration

This page last changed on Feb 17, 2008 by shamid.

To integrate Crowd with your Maven 2 project, you will need to include the following dependency in your `pom.xml`:

```
<dependency>
        <groupId>com.atlassian.crowd</groupId>
        <artifactId>crowd-integration-client</artifactId>
        <version>X.X</version>
        <type>pom</type>
</dependency>
```

Because the Crowd libraries are not published to the standard Maven repository, you will need to add Atlassian's public repository:

```
<repositories>
        <repository>
                <id>central</id>
                <url>https://m2proxy.atlassian.com/repository/public</url>
                <snapshots>
                        <enabled>true</enabled>
                        <updatePolicy>always</updatePolicy>
                </snapshots>
                <releases>
                        <enabled>true</enabled>
                </releases>
        </repository>
</repositories>
```

SOAP API

This page last changed on Apr 07, 2008 by andreask@atlassian.com.

This page provides sample code for creating a Crowd Client using the SOAP API.

The Crowd API has been tested with: Axis 1/2, Microsoft .NET and XFire.

The SOAP WSDL is available on the following URL for the Crowd Standalone version (after you have downloaded and installed Crowd Standalone):

- http://localhost:8095/crowd/services/SecurityServer?wsdl

The Java Remote Interface that is used to generate the SOAP service is available here:

- http://docs.atlassian.com/crowd/current/com/atlassian/crowd/integration/service/soap/server/SecurityServer.html

This JavaDoc file details inputs and outputs for the available Crowd security server SOAP server. You will see that all methods require an `AuthenticatedToken`. A valid token can be obtained by calling the `authenticateApplication` service method.

Like a user token, the application client token is valid only for the same period of time a user token would be. If you receive a SOAP fault for an invalid application client you will need to re-authenticate your application client and re-invoke the SOAP service.

Crowd ships with out of the box Java Integration Libraries that map one-to-one to these web services.

authenticateApplication - Authenticating an Application Client

Here is the server request which passes in the server name and a password credential.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://
www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <soap:Body>
                <authenticateApplication xmlns="urn:SecurityServer">
                        <in0>
                                <credential xmlns="http://
authentication.integration.crowd.atlassian.com">
                                        <credential>password</credential>
                                </credential>

                                <name xmlns="http://
authentication.integration.crowd.atlassian.com">jira</name>

                                <validationFactors xmlns="http://
authentication.integration.crowd.atlassian.com" xsi:nil="true" />
                        </in0>
                </authenticateApplication>
        </soap:Body>
</soap:Envelope>
```

The server will respond with an application token:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://
www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <soap:Body>
                <authenticateApplicationResponse xmlns="urn:SecurityServer">
                        <out>
                                <name xmlns="http://
authentication.integration.crowd.atlassian.com">jira</name>
                                <token xmlns="http://
authentication.integration.crowd.atlassian.com">9vN5haaWY+xGBs3XitgAIg==</token>
                        </out>
```

```
                </authenticateApplicationResponse>
        </soap:Body>
</soap:Envelope>
```

authenticatePrincipal - Authenticating a Principal

In this message the principal is authenticated using the previously obtained application token.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://
www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <soap:Body>
                <authenticatePrincipal xmlns="urn:SecurityServer">
                        <in0>
                                <name xmlns="http://
authentication.integration.crowd.atlassian.com">jive</name>
                                <token xmlns="http://
authentication.integration.crowd.atlassian.com">9vN5haaWY+xGBs3XitgAIg==</token>
                        </in0>
                        <in1>
                                <application xmlns="http://
authentication.integration.crowd.atlassian.com">jive</application>
                                <credential xmlns="http://
authentication.integration.crowd.atlassian.com">
                                        <credential>password</credential>
                                </credential>
                                <name xmlns="http://
authentication.integration.crowd.atlassian.com">jstepka</name>
                                <validationFactors xmlns="http://
authentication.integration.crowd.atlassian.com" />
                        </in1>
                </authenticatePrincipal>
        </soap:Body>
</soap:Envelope>
```

The server then responds with the token for the now authenticated user:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://
www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <soap:Body>
                <authenticatePrincipalResponse xmlns="urn:SecurityServer">
                        <out>o7MSozJJbKQttOLvC4hN2w==</out>
                </authenticatePrincipalResponse>
        </soap:Body>
</soap:Envelope>
```

An invalid authentication attempt will look like the following:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://
www.w3.org/2001/XMLSchema"
               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soap:Body>
        <soap:Fault>
            <faultcode>soap:Server</faultcode>
            <faultstring>Fault:
 com.atlassian.crowd.integration.exception.InvalidAuthenticationException</faultstring>
            <detail>
                <InvalidAuthenticationException xmlns="urn:SecurityServer"/>
            </detail>
        </soap:Fault>
    </soap:Body>
</soap:Envelope>
```

findPrincipalByToken - Finding a Principal by their Authenticated Token

Now that the principal is authenticated, we may want to find additional details about the principal. With the authenticated principal token, the application can now look up a user by a token or their name. The example below shows looking up a principal by their authenticated token:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://
www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <soap:Body>
                <findPrincipalByName xmlns="urn:SecurityServer">
                        <in0>
                                <name xmlns="http://
authentication.integration.crowd.atlassian.com">jive</name>
                                <token xmlns="http://
authentication.integration.crowd.atlassian.com">9vN5haaWY+xGBs3XitgAIg==</token>
                        </in0>
                        <in1>jstepka</in1>
                </findPrincipalByName>
        </soap:Body>
</soap:Envelope>
```

The server lookup response for the principal token:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://
www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soap:Body>
        <findPrincipalByNameResponse xmlns="urn:SecurityServer">
            <out>
                <ID xmlns="http://soap.integration.crowd.atlassian.com">-1</ID>
                <active xmlns="http://soap.integration.crowd.atlassian.com">true</active>
                <attributes xmlns="http://soap.integration.crowd.atlassian.com">
                    <SOAPAttribute>
                        <name>sn</name>
                        <values>
                            <ns1:string xmlns:ns1="urn:SecurityServer">Stepka</ns1:string>
                        </values>
                    </SOAPAttribute>
                    <SOAPAttribute>
                        <name>invalidPasswordAttempts</name>
                        <values>
                            <ns1:string xmlns:ns1="urn:SecurityServer">0</ns1:string>
                        </values>
                    </SOAPAttribute>
                    <SOAPAttribute>
                        <name>requiresPasswordChange</name>
                        <values>
                            <ns1:string xmlns:ns1="urn:SecurityServer">false</ns1:string>
                        </values>
                    </SOAPAttribute>
                    <SOAPAttribute>
                        <name>mail</name>
                        <values>

  <ns1:string xmlns:ns1="urn:SecurityServer">justen.stepka@atlassian.com</ns1:string>
                        </values>
                    </SOAPAttribute>
                    <SOAPAttribute>
                        <name>lastAuthenticated</name>
                        <values>
                            <ns1:string xmlns:ns1="urn:SecurityServer">1169440408520</ns1:string>
                        </values>
                    </SOAPAttribute>
                    <SOAPAttribute>
                        <name>givenName</name>
```

```
                        <values>
                            <ns1:string xmlns:ns1="urn:SecurityServer">Justen</ns1:string>
                        </values>
                    </SOAPAttribute>
                    <SOAPAttribute>
                        <name>passwordLastChanged</name>
                        <values>
                            <ns1:string xmlns:ns1="urn:SecurityServer">1168995491407</ns1:string>
                        </values>
                    </SOAPAttribute>
                </attributes>
                <conception xmlns="http://
soap.integration.crowd.atlassian.com">2007-01-17T11:58:11+11:00</conception>
                <description xmlns="http://soap.integration.crowd.atlassian.com" xsi:nil="true"/>
                <directoryID xmlns="http://soap.integration.crowd.atlassian.com">1</directoryID>
                <lastModified xmlns="http://
soap.integration.crowd.atlassian.com">2007-01-17T18:38:51+11:00
                </lastModified>
                <name xmlns="http://soap.integration.crowd.atlassian.com">jstepka</name>
            </out>
        </findPrincipalByNameResponse>
    </soap:Body>
</soap:Envelope>
```

Related Topics

- Application Integration Overview
  - Sample Application ('demo')
- Java Integration Libraries
  - Compiling the Crowd Source
  - Maven 2 Integration
- SOAP API
  - Axis 1.x Client Stub Generation
  - Microsoft .NET Client

Crowd Documentation

Axis 1.x Client Stub Generation

This page last changed on Apr 10, 2008 by smaddox.

Please note that these instructions assume that you are using Axis 1.x +. For more general information on installing and using this version of Axis please visit the Axis 1 website.

Refer to the attached sample client stub generated with Axis 1.4, including test client.

Add the required Axis libraries to your AXISCLASSPATH

This can be done as follows:

Windows:

```
set AXIS_HOME=c:\axis
set AXIS_LIB=%AXIS_HOME%\lib
set AXISCLASSPATH=%AXIS_LIB%\axis.jar;%AXIS_LIB%\commons-discovery.jar;
  %AXIS_LIB%\commons-logging.jar;%AXIS_LIB%\jaxrpc.jar;%AXIS_LIB%\saaj.jar;
  %AXIS_LIB%\log4j-1.2.8.jar;:%AXIS_LIB%\wsdl4j-1.5.1.jar
```

Unix:

```
export AXIS_HOME=/opt/java/axis-1_4/
export AXIS_LIB=$AXIS_HOME/lib
export AXISCLASSPATH=$AXIS_LIB/axis.jar:$AXIS_LIB/commons-discovery-0.2.jar:$AXIS_LIB/commons-
logging-1.0.4.jar:
   $AXIS_LIB/jaxrpc.jar:$AXIS_LIB/saaj.jar:$AXIS_LIB/log4j-1.2.8.jar:$AXIS_LIB/wsdl4j-1.5.1.jar
```

If you are using a version of Java earlier than 1.5 you may need to add xml-apis.jar and xercesImpl.jar to the Axis Classpath.

Generating the actual Axis stub classes

Assuming you have set up your AXISCLASSPATH as above, run the following command:

```
java -cp "$AXISCLASSPATH" org.apache.axis.wsdl.WSDL2Java http://localhost:8095/crowd/services/
SecurityServer?wsdl
```

When the necessary objects are created off the Crowd server WSDL, you will end up with a directory structure similar to this:

```
drwxr-xr-x   6 jstepka  jstepka  204 Apr 19 16:56 SecurityServer_pkg
drwxr-xr-x   3 jstepka  jstepka  102 Apr 19 16:55 com
drwxr-xr-x   4 jstepka  jstepka  136 Apr 19 17:05 java
```

Compiling the generated sources

When you attempt to compile the generated class files, you will end up with a compilation error similar to the following:

```
java/rmi/RemoteException.java:[10,7] cyclic inheritance involving java.rmi.RemoteException
java/rmi/RemoteException.java:[11,32] modifier private not allowed here
java/rmi/RemoteException.java:[12,29] modifier private not allowed here
java/rmi/RemoteException.java:[64,29] modifier private not allowed here
java/rmi/RemoteException.java:[86,20] modifier private not allowed here
java/rmi/RemoteException.java:[104,56] modifier private static not allowed here
com/atlassian/crowd/integration/exception/InvalidCredentialException.java:[26,30] incompatible
 types
```

To resolve these compilation errors you will need to delete the generated `java` package and also remove all references to these custom RemoteException and Throwable exceptions in the stubs that Axis created. We highly recommend using an IDE for this as you will need to modify a number of classes.

A small example of using the Axis-generated stubs

The security server can then be used as below:

```
// connect to the crowd server, using the supplied service URL, similar to http://localhost:8095/
crowd/services/SecurityServer?wsdl
        SecurityServerLocator secServer = new SecurityServerLocator();

 secServer.setSecurityServerHttpPortEndpointAddress(secServer.getSecurityServerHttpPortAddress());

        // obtain a reference to the SOAP service, which axis manages.
        SecurityServerHttpBindingStub stub = null;
        try
        {
            stub = (SecurityServerHttpBindingStub) secServer.getSecurityServerHttpPort();

            // authenticate the integrated crowd application
            AuthenticatedToken token = stub.authenticateApplication(new
 ApplicationAuthenticationContext(
                    new PasswordCredential("password", Boolean.FALSE), "demo",
                    new ValidationFactor[0]));

            // do your custom calls here, eg:
            SOAPPrincipal principal = new SOAPPrincipal();
            principal.setName("test-user");
            principal.setActive(Boolean.TRUE);
            principal.setAttributes(new SOAPAttribute[]{
                    new SOAPAttribute("mail", new String[]{"test@example.com"}),
                    new SOAPAttribute("givenName", new String[]{"Paul"}),
                    new SOAPAttribute("sn", new String[]{"Smith"})
            });
            stub.addPrincipal(token, principal, new PasswordCredential("secret", Boolean.FALSE));

        }
        catch (Exception e)
        {
            System.err.print(e);
        }
```

Microsoft .NET Client

This page last changed on Nov 11, 2007 by smaddox.

An updated version of this library has been made available through the Atlassian Codegeist competition.

You will need to create a .NET proxy to the SOAP API, as follows:

1. Open a Microsoft Visual Studio .NET Command Prompt.
2. Run the following command to generate a proxy class (change the location of the WSDL according to your installation):

   ```
   wsdl /l:CS /protocol:SOAP http://localhost:8080/crowd/services/SecurityServer?wsdl
   ```

   (Note: Ignore any schema validation warnings returned here.)
3. Compile the generated class with the following references:

   ```
   csc /t:library /r:System.Web.Services.dll /r:System.Xml.dll SecurityServer.cs
   ```

   This should generate a .NET assembly called `SecurityServer.DLL`.

When creating your .NET client application, remember to add a reference to this proxy. You will also need to add a reference to `System.Web.Services.DLL`.

The sample code calls methods from the proxy to perform authentication in a sample Crowd application. Change the constants at the top of the code relevant to any application you have previously set up in Crowd.

Related Topics

- Application Integration Overview
  - Sample Application ('demo')
- Java Integration Libraries
  - Compiling the Crowd Source
  - Maven 2 Integration
- SOAP API
  - Axis 1.x Client Stub Generation
  - Microsoft .NET Client

Crowd Documentation

Creating a Custom Directory Connector

This page last changed on May 05, 2008 by smaddox.

Crowd comes with a number of supplied directory connectors. If your directory is not listed in Supported Applications and Directories then you will need to create your own custom directory connector.

The page 2.2.3 Configuring a Custom Directory Connector does not exist.

The directory connectors that come with Crowd implement the Java interface `RemoteDirectory`. The `RemoteDirectory` interface defines generic methods for authentication, searching and entity create, remove and update operations.

To connect Crowd to a custom directory server, you will need to write a Java class file that implements the `RemoteDirectory` interface.

In our example below, the `MyCustomDirectoryServer` class extends the Crowd `DirectoryEntity` utility class. The utility class manages setting runtime properties that may be used by the Crowd server in the future for operations such as getting the Crowd `ID` number of the directory server.

```
package com.atlassian.crowd.integration.directory.custom;

import com.atlassian.crowd.integration.SearchContext;
import com.atlassian.crowd.integration.authentication.PasswordCredential;
import com.atlassian.crowd.integration.directory.RemoteDirectory;
import com.atlassian.crowd.integration.exception.*;
import com.atlassian.crowd.integration.model.DirectoryEntity;
import com.atlassian.crowd.integration.model.RemoteGroup;
import com.atlassian.crowd.integration.model.RemotePrincipal;
import com.atlassian.crowd.integration.model.RemoteRole;

import java.rmi.RemoteException;
import java.util.List;

public class MyCustomDirectoryServer extends DirectoryEntity implements RemoteDirectory
{
    public RemotePrincipal authenticate(String name, PasswordCredential[] credentials) throws
 RemoteException,
            InvalidPrincipalException, InactiveAccountException, InvalidAuthenticationException {
        // Perform your custom directory server authentication code here.
        //
        // The source code to the InternalDirectory, which comes with your commercial license is a
 good implementation example.
    }

    // Other RemoteDirectory interface methods will also need to be implemented ...
}
```

Once you have finished implementing all of the methods defined by the `RemoteDirectory` interface, you will then need to:

1. Create a JAR of the `MyCustomDirectoryServer` and any supporting class files.
2. Shut down Crowd.
3. Place the newly-created JAR from step one in the `CROWD/crowd-webapp/WEB-INF/lib` folder.
4. Start Crowd.
5. Follow the instructions on configuring a custom directory connector through the Crowd Administration Console.

Full Javadoc for the `RemoteDirectory` interface can be found here: http://docs.atlassian.com/crowd-core/1.3/crowd-core/apidocs/com/atlassian/crowd/integration/directory/RemoteDirectory.html.

A 'principal' is a 'user'

In Crowd, the term 'principal' is equivalent to the term 'user'. In Crowd 1.3.0 and later, the Crowd Administration Console uses the term 'user'. Earlier versions of Crowd, and also certain API libraries, use the term 'principal'.

Next Steps

After creating your directory connector, please see Configuring a Custom Directory Connector.

Related Topics

- Creating a Crowd Client for your Custom Application
  - Application Integration Overview
    - Sample Application ('demo')
  - Java Integration Libraries
    - Compiling the Crowd Source
    - Maven 2 Integration
  - SOAP API
    - Axis 1.x Client Stub Generation
    - Microsoft .NET Client
- Creating a Custom Directory Connector
- Crowd Developer FAQ
- IntelliJ IDEA Setup Guide
  - Setting up Tomcat in IDEA for Crowd

Crowd Documentation

Crowd Developer FAQ

This page last changed on Dec 17, 2007 by jnolen.

IntelliJ IDEA Setup Guide

This page last changed on May 05, 2008 by smaddox.

Atlassian does not support customised Crowd source

This document is intended to serve as an IntelliJ IDEA setup guide for those who have a Crowd source license and wish to customise Crowd 1.2.2 or later. For support beyond this document, please refer to our online forum.

Prerequisites

1. IntelliJ IDEA 6 or greater is installed.
2. JDK 1.5 or greater is installed.
3. Tomcat 5 or greater is installed. We will refer to the Tomcat root folder as `TOMCAT`.
4. MySQL 5 or greater is installed. Any Crowd-supported database server will work. This guide will assume you are using MySQL.
5. MySQL Connector/J 5.1 JDBC or greater is downloaded. We will refer to the extracted archive path as `JDBC`.
6. Maven 2.0.7 or greater is installed.
7. The latest Crowd source (version 1.2.2 or greater) is downloaded and extracted. We will refer to this extracted archive path as `SOURCE`.

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

Step 2. Configure Tomcat

1. Copy the **`JDBC/mysql-connector-java-5.x.x-bin.jar`** to the **`TOMCAT/common/lib/`** directory.
2. Copy the <MAVEN_REPOSITORY>/repository/javax/transaction/jta/1.0.1B/jta-1.0.1B.jar to the **`TOMCAT/common/lib/`** directory.
3. Copy the <MAVEN_REPOSITORY>/repository/javax/activation/jaf/1.1/jaf-1.1.jar to the **`TOMCAT/common/lib/`** directory.
4. Copy the <MAVEN_REPOSITORY>/repository/javax/mail/mail/1.4/mail-1.4.jar to the **`TOMCAT/common/lib/`** directory.
5. Edit the **`TOMCAT/conf/context.xml`** configuration file to add a global JNDI JDBC connection. Make sure to customise the username and password for your specific environment.

```
<Context>
<!-- Default set of monitored resources -->
  <WatchedResource>WEB-INF/web.xml</WatchedResource>
<!-- Uncomment this to disable session persistence across Tomcat restarts -->
<!--
    <Manager pathname="" />
    -->
  <Resource type="javax.sql.DataSource" name="jdbc/CrowdDS" username="crowduser"
 password="password" driverClassName="com.mysql.jdbc.Driver" url="jdbc:mysql://localhost/
crowddb?autoReconnect=true" auth="Container"/>
</Context>
```

Step 3. Run the Maven Build Commands

1. Copy (or merge) the **`SOURCE/maven/conf/settings.xml`** to your **`~/.m2`** directory.
2. Run the following Maven command in the root source directory (normally called atlassian-crowd-x.x.x-source) to download the project dependencies (otherwise known as download the Internet):

   ```
   mvn clean install -Dmaven.test.skip
   ```

   The download may take a few hours when you do it for the first time.

If this build is successful, run the following Maven command:

```
mvn idea:idea -Dmaven.test.skip
```

Step 4. Configure IntelliJ IDEA

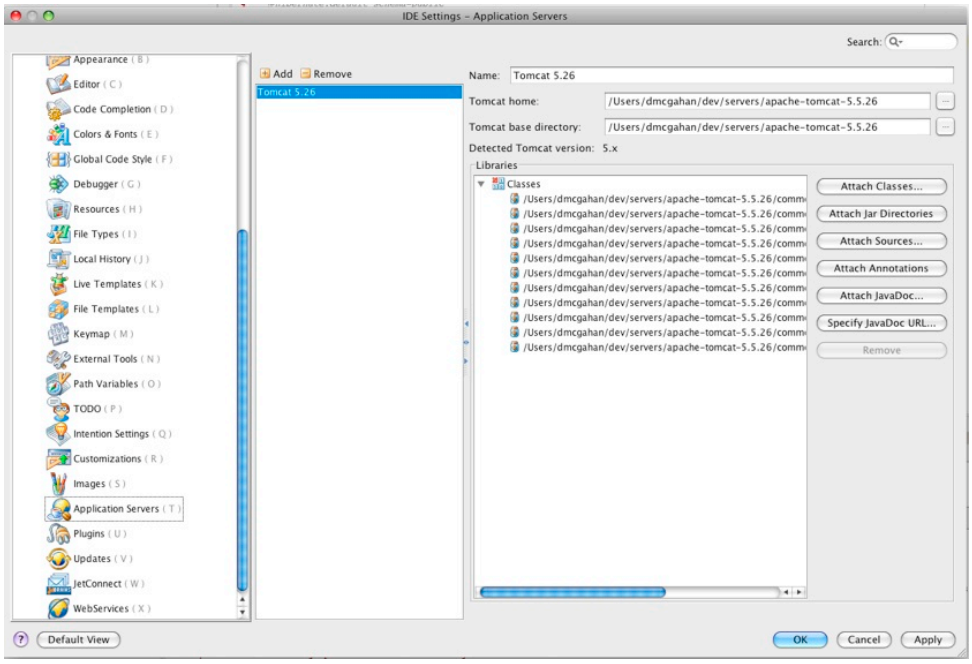
1. Open the `SOURCE/atlassian-crowd.ipr` with IntelliJ IDEA.
2. Edit the `SOURCE/atlassian-crowd/crowd-web-app/src/main/resources/jdbc.properties` configuration file and change `hibernate.dialect` to `org.hibernate.dialect.MySQLDialect`.
3. Edit the `SOURCE/atlassian-crowd/crowd-web-app/src/main/resources/crowd.properties` configuration file and change `application.login.url` to contain the port and IP address you have configured Tomcat to run on. The default is `localhost:8080`.
4. Configure IntelliJ IDEA to have a new `TOMCAT` runtime configuration for the `crowd-web-app` module.

Setting up Tomcat in IDEA for Crowd

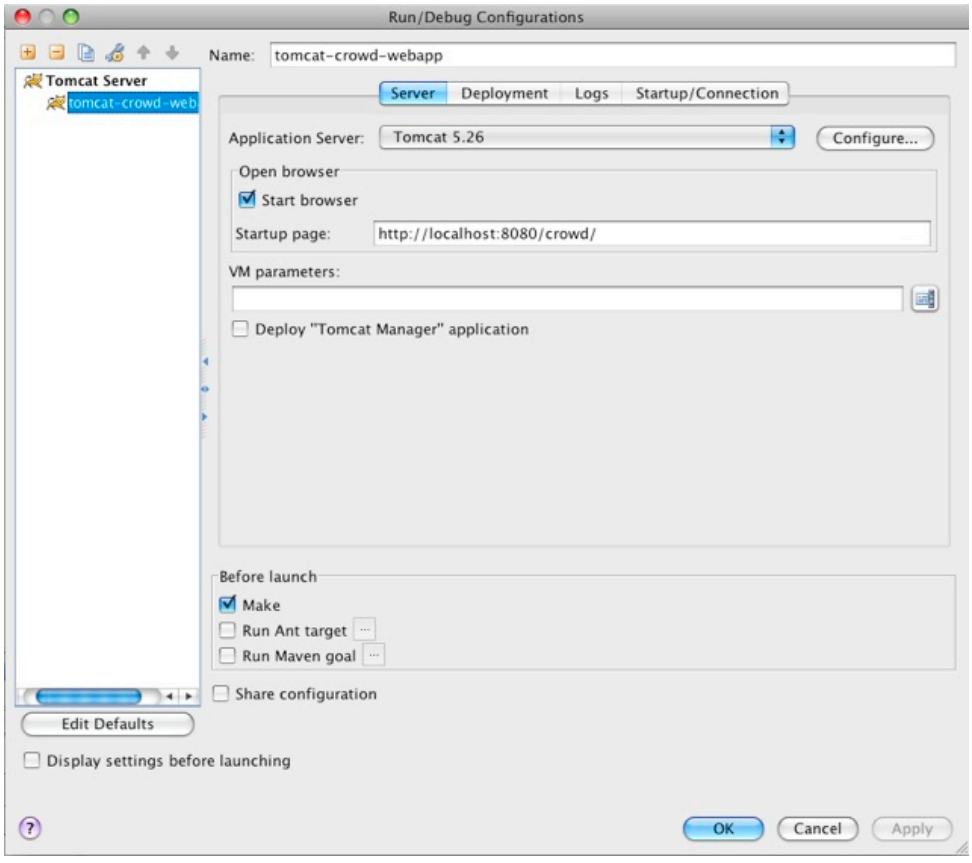
This page last changed on Feb 11, 2008 by donna@atlassian.com.

This guide assumes that you are running IDEA 7.x and that you have already installed and tested Tomcat 5.25 or greater.

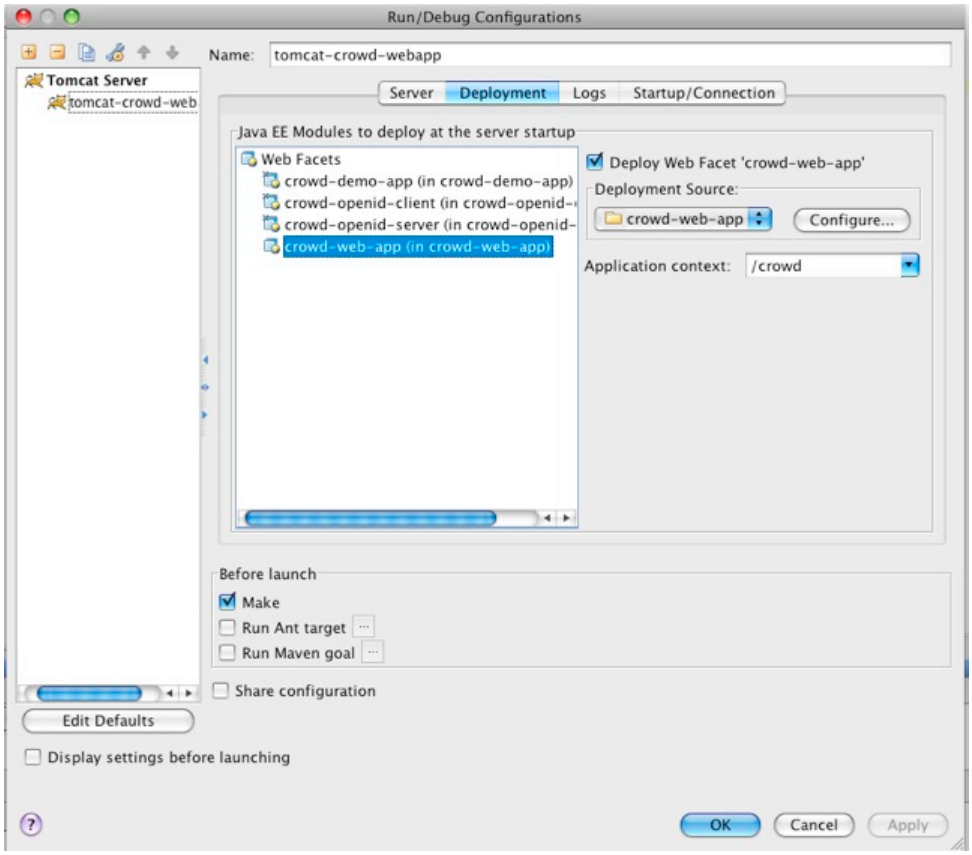
1. Set up Tomcat as one of your application servers on this IDEA screen (Preferences -> IDE Settings -> Application Servers).



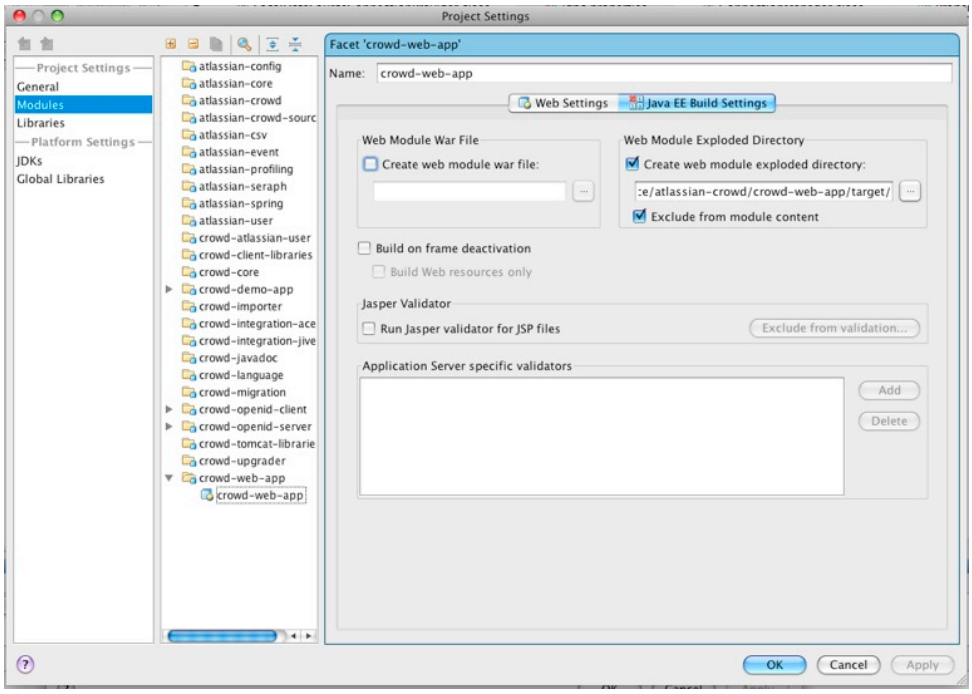
2. Add a new IDE configuration (Run -> Edit Configurations) called tomcat-crowd-webapp that uses the Tomcat application server added in the last step.



3. Click the Deploy tab. Select to deploy the crowd-webapp module. Once this is done, click the Configure button.



4. Select the Java EE Build Settings tab. Check Create web module exploded directory and Exclude from module content.



CrowdID Administration Guide

This page last changed on May 07, 2008 by smaddox.

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.

Table of Contents

1. About CrowdID

This page last changed on May 04, 2008 by smaddox.

CrowdID is a free add-on that ships with Crowd versions 1.1 and later. It gives administrators a secure way to provide [OpenID|http://openid.net/] 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 Crowd Administration Guide.

To access CrowdID, go to http://localhost:8095/openidserver.

1.1 How CrowdID works with Crowd

This page last changed on May 07, 2008 by smaddox.

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 Crowd 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 organisations 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 Crowd Administration Guide.

To access CrowdID, go to http://localhost:8095/openidserver.

RELATED TOPICS

1.1.1 Determining the name of the CrowdID application

This page last changed on May 05, 2008 by smaddox.

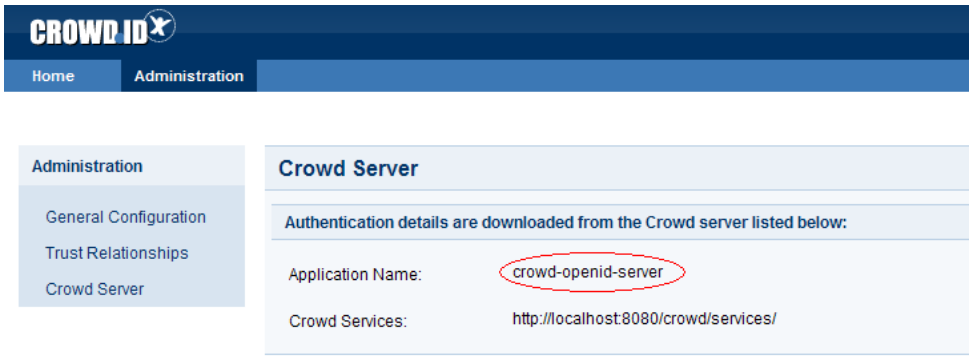
CrowdID is a Crowd-connected application (for more information please see Managing Applications in the Crowd Administration Guide).

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'



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
- 1.1 How OpenID sites interact with CrowdID

Crowd Documentation

1.1.2 Locating the Crowd Server that CrowdID is using

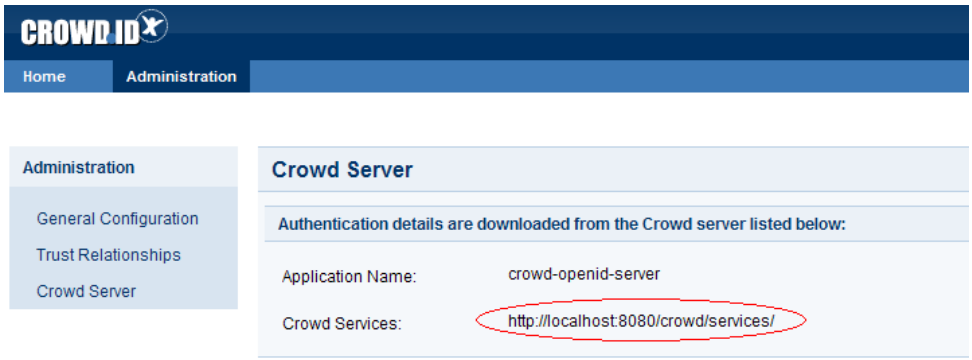
This page last changed on Jun 15, 2007 by rosie@atlassian.com.

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'



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
- 1.1 How OpenID sites interact with CrowdID

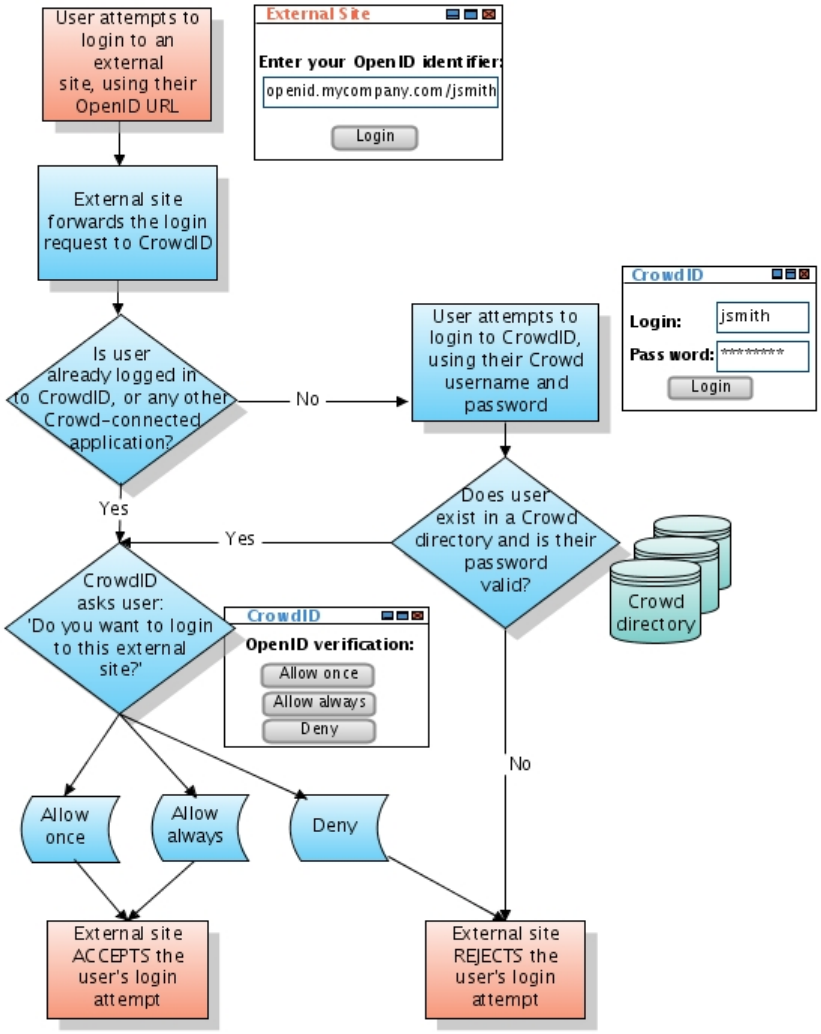
Crowd Documentation

1.1 How OpenID sites interact with CrowdID

This page last changed on Jun 19, 2007 by rosie@atlassian.com.

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
- 1.1 How OpenID sites interact with CrowdID

Crowd Documentation

2. Allowing users to access CrowdID

This page last changed on Feb 19, 2008 by smaddox.

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

This page last changed on May 05, 2008 by smaddox.

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[1].
2. View your CrowdID application[2] as described in Using the Application Browser in the Crowd 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 Crowd 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[1].
2. Map the directory to your CrowdID application[2] as described in Mapping a Directory to an Application in the Crowd Administration Guide.

To grant CrowdID access rights to a particular group of users within a directory,

1. Login to your Crowd server[1].
2. Map the group to your CrowdID application[2] as described in Specifying which Groups can access an Application in the Crowd Administration Guide.

[1] To find your Crowd server's URL, see 1.1.2 Locating the Crowd Server that CrowdID is using.
[2] 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

This page last changed on May 05, 2008 by smaddox.

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:

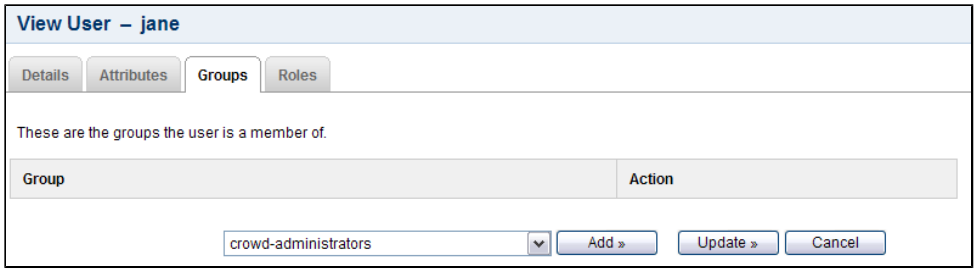
- Adding a user to the 'crowd-administrators' group will also give them Crowd administration rights (unless you choose to use a different group to contain Crowd administrators). See Granting Crowd Administration Rights to a User in the Crowd Administration Guide.
- The 'crowd-administrators' group always 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[1].
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.

[1] To find your Crowd server's URL, see 1.1.2 Locating the Crowd Server that CrowdID is using.

Screenshot: Granting Crowd administration rights



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 login

This page last changed on Jun 14, 2007 by rosie@atlassian.com.

There are three ways to specify which OpenID hosts (i.e. websites or IP addresses) your users can login to using their CrowdID:

- No restriction — your CrowdID users can login to any OpenID host
- Blacklist — your CrowdID users can login to any OpenID host except the one(s) that you specify
- Whitelist — your CrowdID users can login to only those OpenID host(s) that you specify

3.1 Allowing all hosts

This page last changed on Jun 14, 2007 by rosie@atlassian.com.

There are three ways to specify which OpenID hosts (i.e. websites or IP addresses) your users can login to using their CrowdID:

- No restriction — your CrowdID users can login to any OpenID host
- Blacklist — your CrowdID users can login to any OpenID host except the one(s) that you specify
- Whitelist — your CrowdID users can login 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')

Crowd Documentation

3.2 Allowing all except specified hosts ('Blacklist')

This page last changed on Jun 14, 2007 by rosie@atlassian.com.

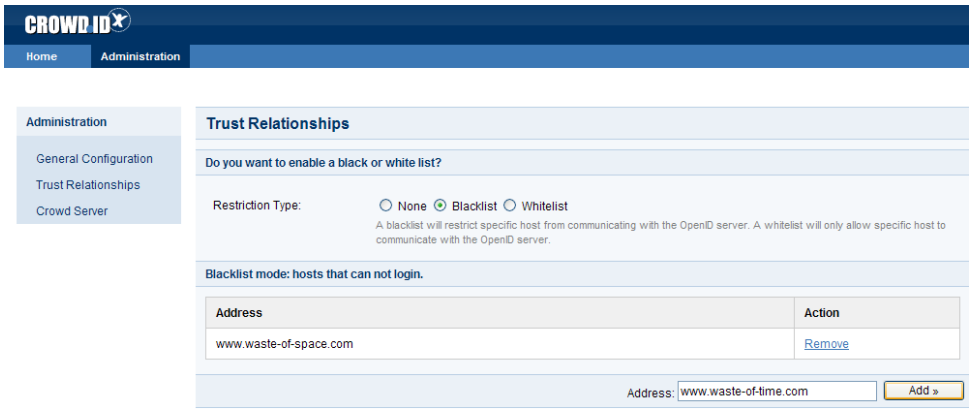
There are three ways to specify which OpenID hosts (i.e. websites or IP addresses) your users can login to using their CrowdID:

- No restriction — your CrowdID users can login to any OpenID host
- Blacklist — your CrowdID users can login to any OpenID host except the one(s) that you specify
- Whitelist — your CrowdID users can login 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'



RELATED TOPICS

- 3.1 Allowing all hosts
- 3.2 Allowing all except specified hosts ('Blacklist')
- 3.3 Allowing specified hosts only ('Whitelist')

Crowd Documentation

3.3 Allowing specified hosts only ('Whitelist')

This page last changed on Jun 14, 2007 by rosie@atlassian.com.

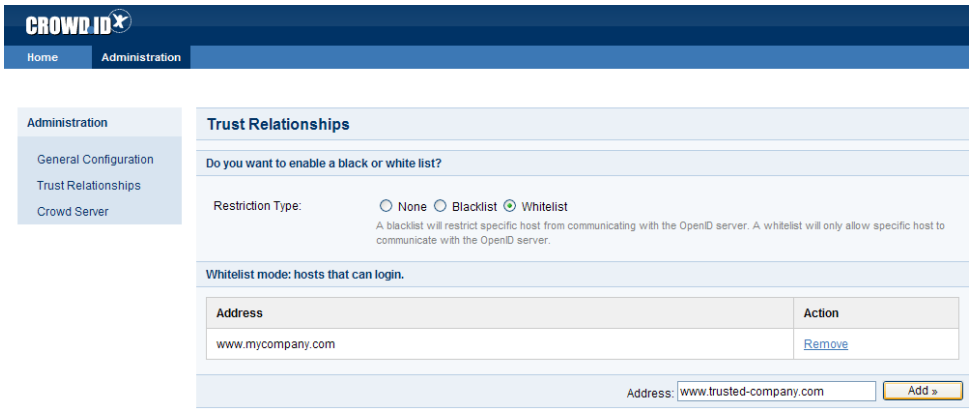
There are three ways to specify which OpenID hosts (i.e. websites or IP addresses) your users can login to using their CrowdID:

- No restriction — your CrowdID users can login to any OpenID host
- Blacklist — your CrowdID users can login to any OpenID host except the one(s) that you specify
- Whitelist — your CrowdID users can login 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'



RELATED TOPICS

- 3.1 Allowing all hosts
- 3.2 Allowing all except specified hosts ('Blacklist')
- 3.3 Allowing specified hosts only ('Whitelist')

Crowd Documentation

4. Configuring CrowdID system settings

This page last changed on Jun 13, 2007 by rosie@atlassian.com.

4.1 Specifying the CrowdID URL

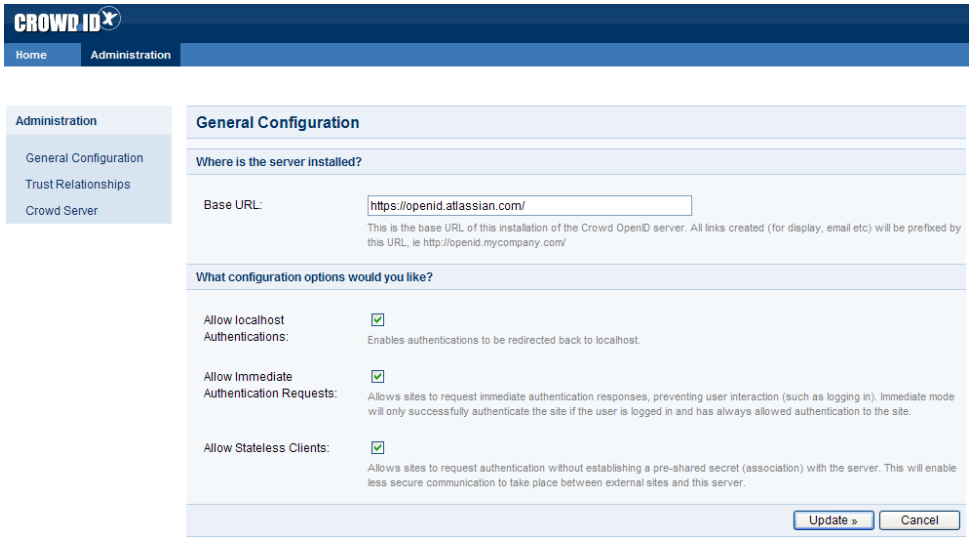
This page last changed on Jun 18, 2007 by smaddox.

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

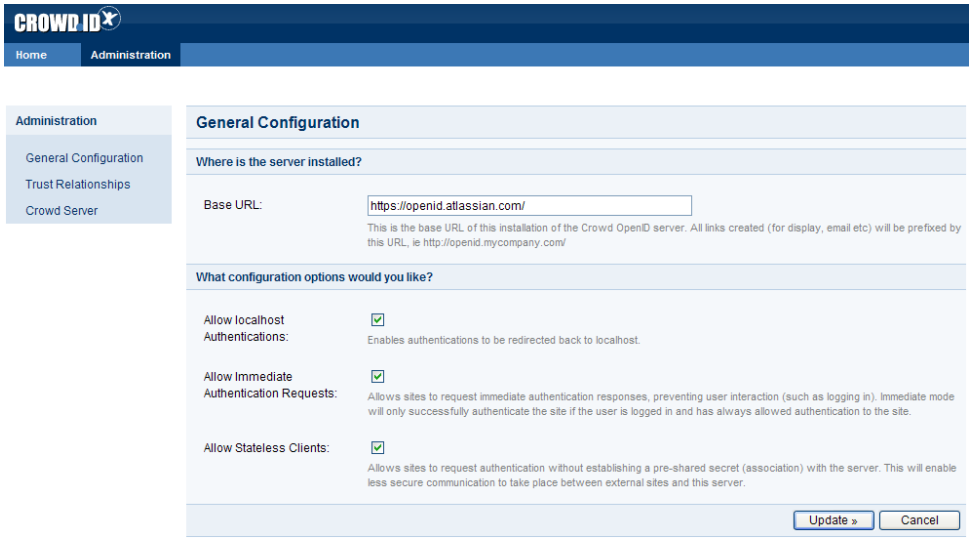
This page last changed on Jun 18, 2007 by rosie@atlassian.com.

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'



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

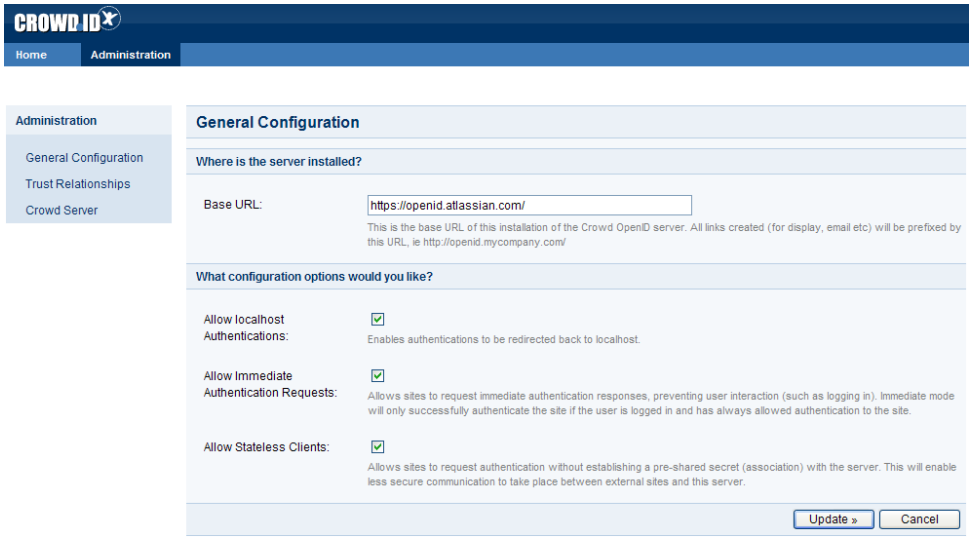
This page last changed on Jun 18, 2007 by rosie@atlassian.com.

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'



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

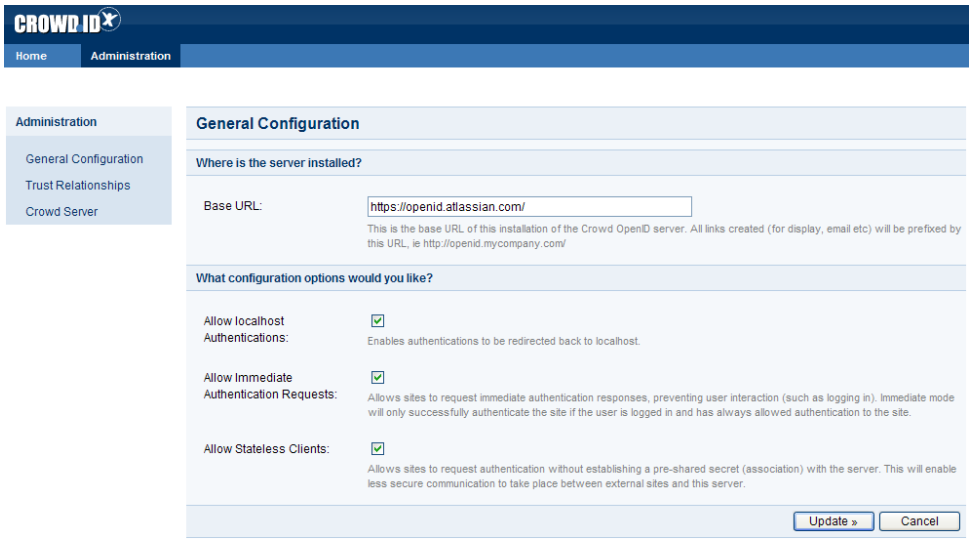
This page last changed on Jun 20, 2007 by shamid.

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'



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

This page last changed on Nov 26, 2007 by smaddox.

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 organisation. 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

This page last changed on Jun 15, 2007 by smaddox.

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 organisation.

1.1 What is OpenID?

This page last changed on Jun 19, 2007 by smaddox.

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 login to any OpenID-enabled website. OpenID allows the website to communicate with your OpenID provider (e.g. your organisation'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?

This page last changed on Jun 19, 2007 by smaddox.

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 organisation.

This means that you can:

- Securely store your username and password on your organisation'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 organisation 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?

This page last changed on Jun 19, 2007 by smaddox.

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 organisation'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.

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

This page last changed on Jun 19, 2007 by smaddox.

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.

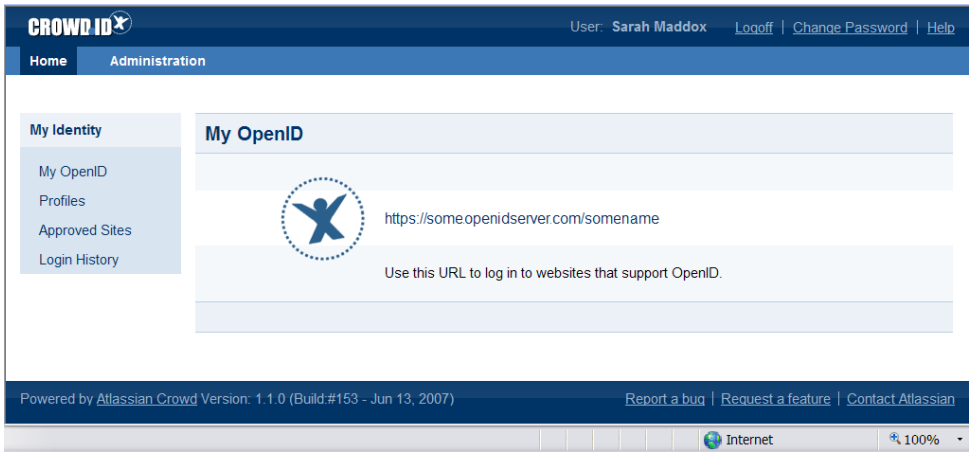
To access the CrowdID page while you are logging in to another site,

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 organisation.
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

2. Logging in to a website using OpenID

This page last changed on Jun 19, 2007 by smaddox.

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?

This page last changed on Jun 19, 2007 by smaddox.

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

This page last changed on Jun 19, 2007 by smaddox.

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 labelled '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

This page last changed on Jun 19, 2007 by smaddox.

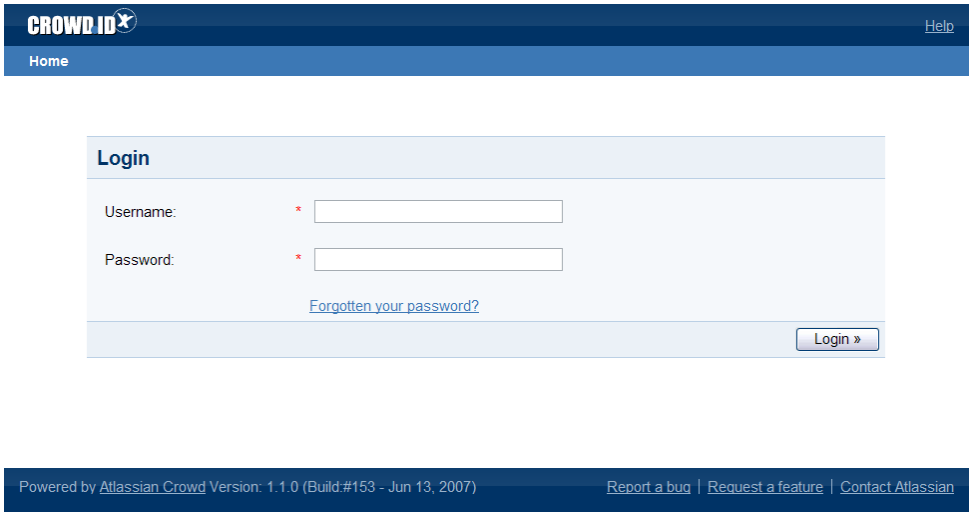
CrowdID will ask you to log in, if you have not already done so during this browser session or if your session has timed out. The CrowdID login may appear during the process of logging in to another website, or when you are accessing CrowdID directly.

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



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

This page last changed on Jun 19, 2007 by smaddox.

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.

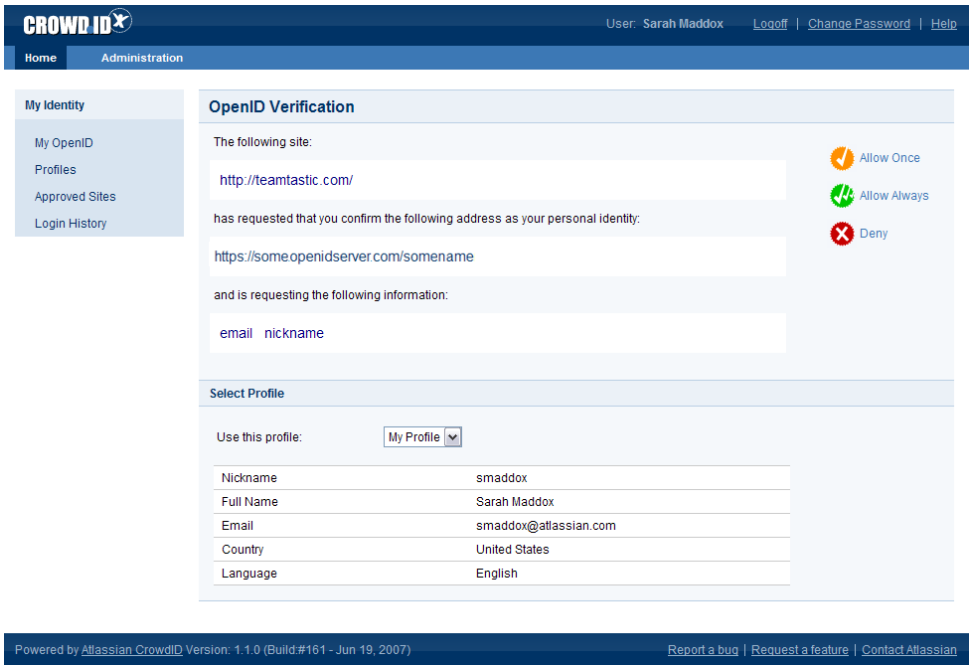
If you have previously instructed CrowdID to allow this site always, you will not see this page. You can remove a site from the 'Allow Always' list in CrowdID.

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



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 cancelled'.

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 labelled '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

This page last changed on Jun 19, 2007 by smaddox.

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 dialogue 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

This page last changed on Jun 19, 2007 by smaddox.

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.

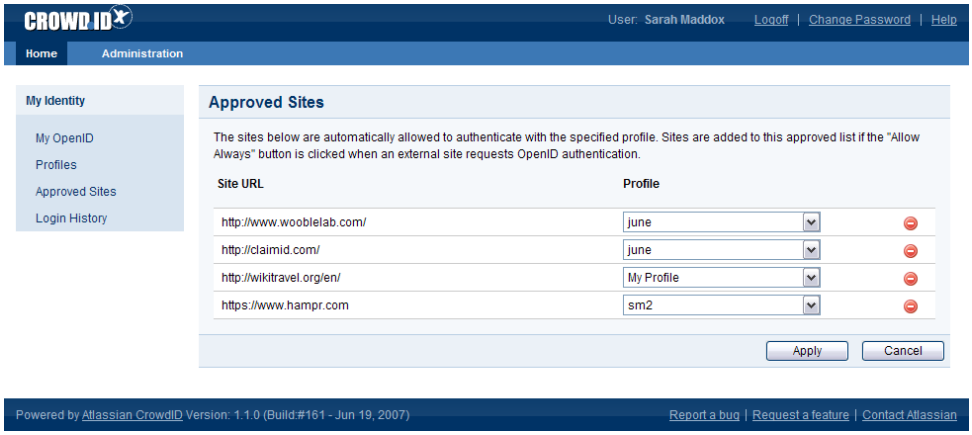
   If you do not click the 'Apply' button, your changes will be cancelled.

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 cancelled.

Screenshot: CrowdID Approved Sites 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

4. Viewing your login history

This page last changed on Nov 26, 2007 by smaddox.

The CrowdID 'Login History' page displays a list of the sites you have visited and the type of approval you gave on each visit:

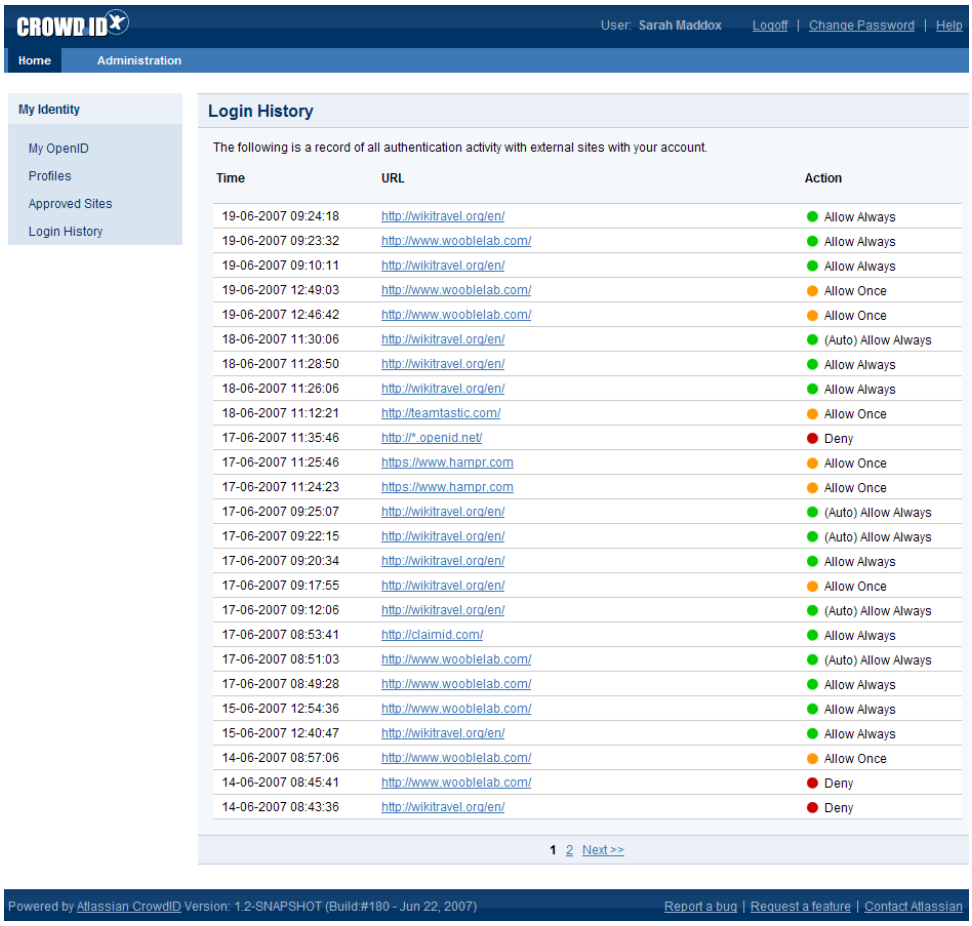
- 'Allow Always' - At the time of this login, you instructed CrowdID to allow login to the site every time you use your OpenID.
- '(Auto) Allow Always' - This login was allowed automatically, because you have previously instructed CrowdID to allow login to the site every time you use your OpenID.
- 'Allow 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.

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



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

This page last changed on Jun 18, 2007 by smaddox.

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 organisation's user directory.

You can update your profile information on CrowdID, as described below.

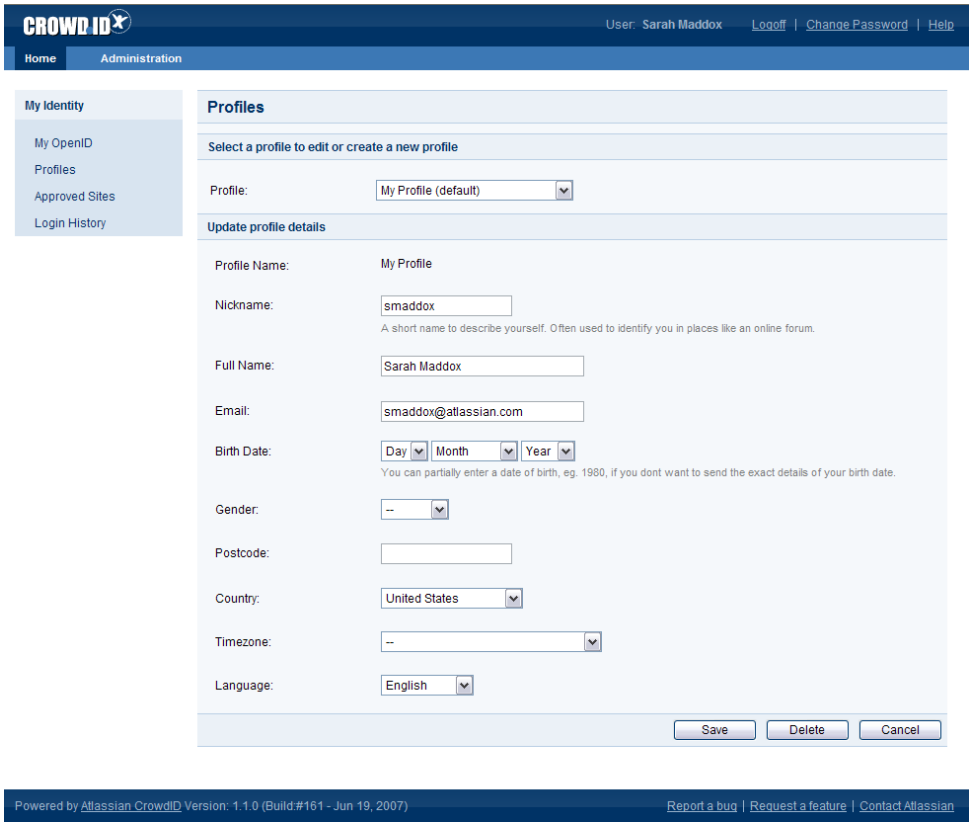
You can also:

- 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



RELATED TOPICS

- 1. Getting started with CrowdID
- 2. Logging in to a website using OpenID
- 3. Viewing your always-approved websites

6. Using more than one profile

This page last changed on Jun 18, 2007 by smaddox.

You can create multiple profiles in CrowdID and then allocate specific profiles to specific websites.

6.1 Adding a profile

This page last changed on Jun 17, 2007 by smaddox.

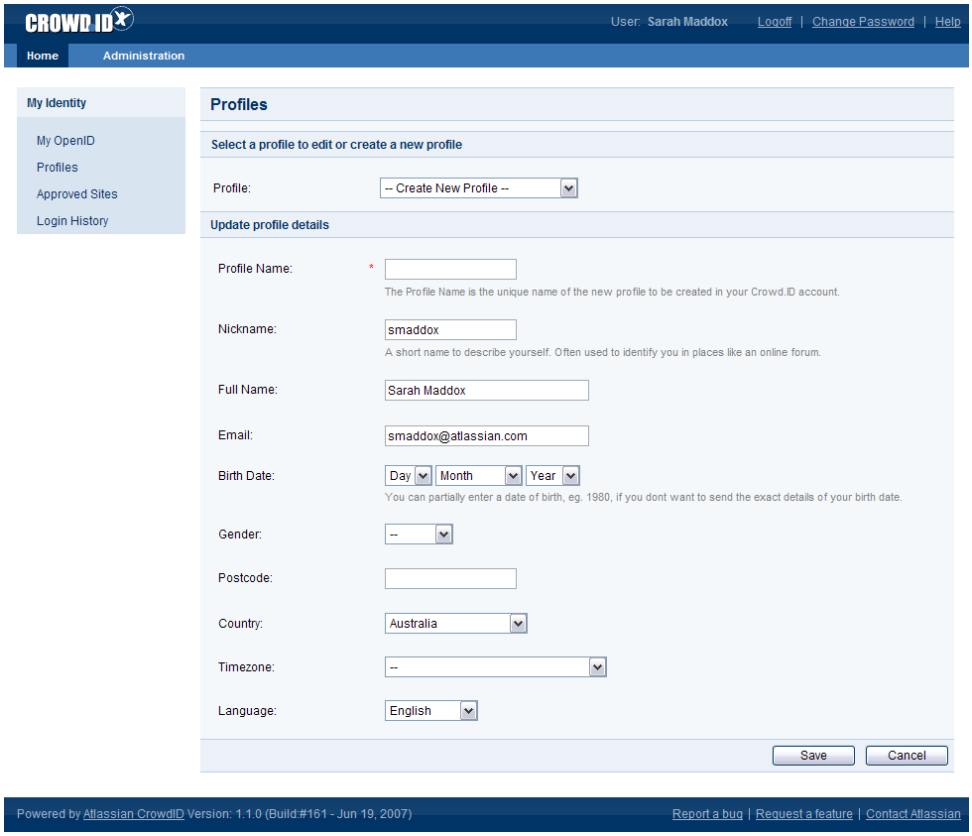
When you log in to a website using your OpenID, CrowdID will pass some information to the website. The information is copied from your profile on CrowdID. When your profile is first created, CrowdID will auto-fill the information where possible, by copying:

- Country and language from the language information in your browser.
- Name and email address from your organisation'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



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

6.2 Choosing a profile for a website

This page last changed on Jun 20, 2007 by smaddox.

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

This page last changed on Jun 20, 2007 by shamid.

If you have more than one profile, you can choose one of them as default.

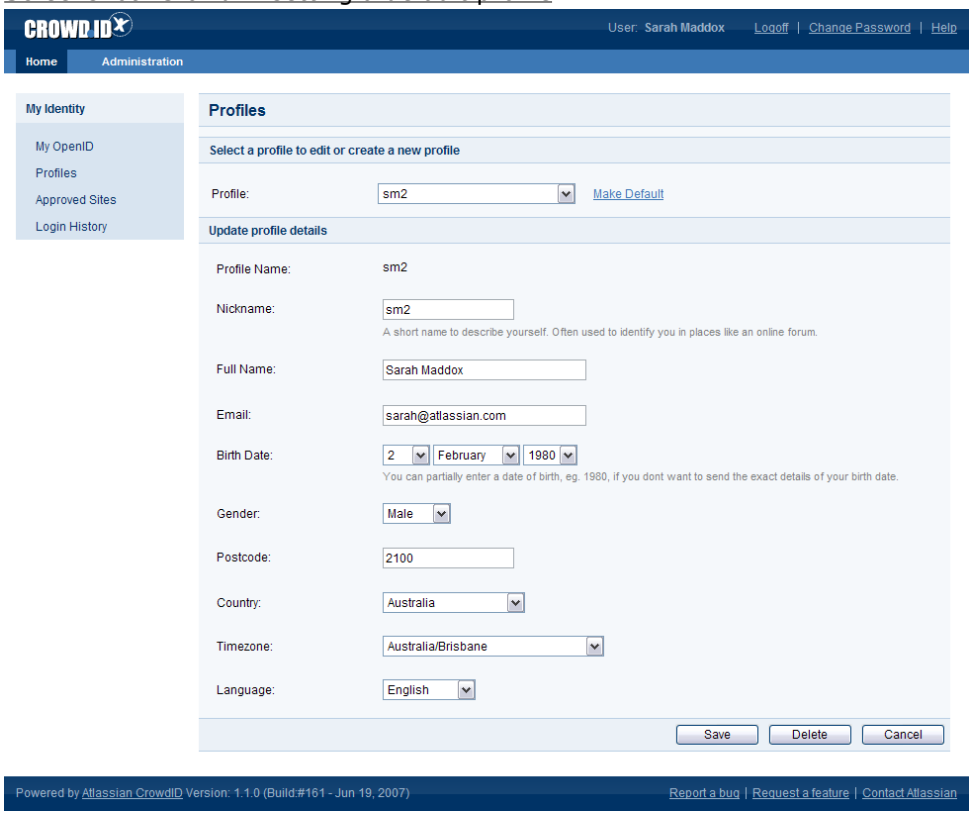
Effect of the 'default' profile when you are logging in to a website:

- 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



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

This page last changed on Jun 19, 2007 by smaddox.

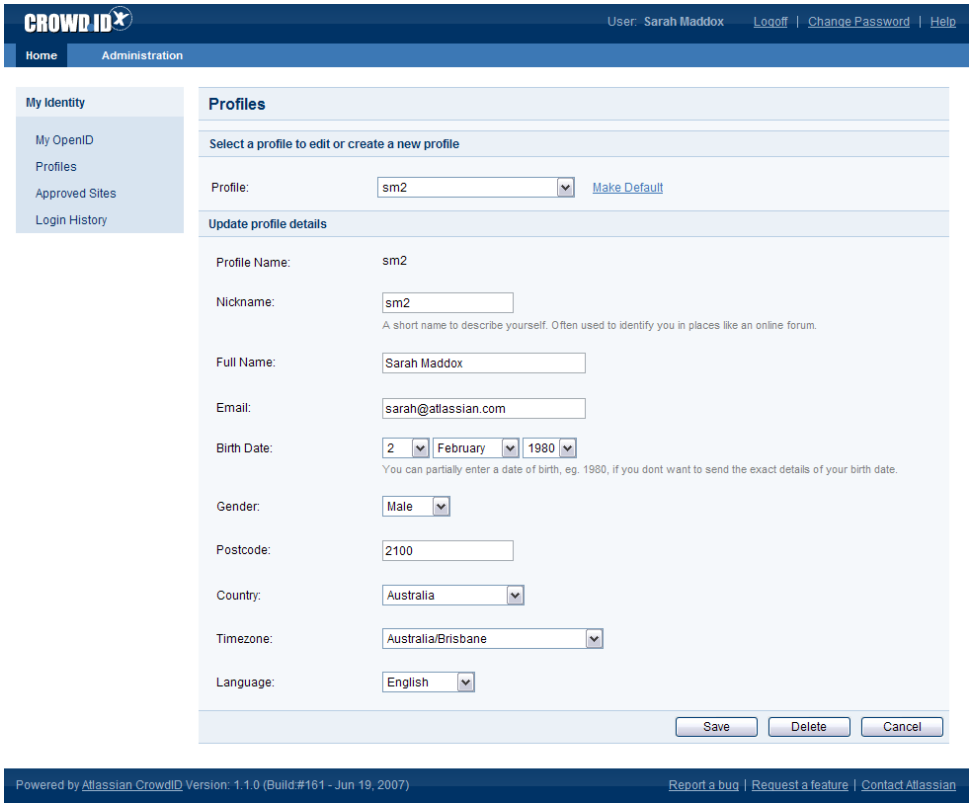
You can delete one of your profiles on CrowdID, provided that it is not your default profile.

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.

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



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

This page last changed on Jun 19, 2007 by smaddox.

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, if you have forgotten the old one. A new password will be emailed to you.

7.1 Changing your password

This page last changed on Jun 17, 2007 by smaddox.

The CrowdID 'Change Your Password' page allows you to change your password across all applications in your organisation, 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. '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

7.2 Resetting your password

This page last changed on Jun 19, 2007 by smaddox.

The CrowdID 'Login' page allows you to reset your password, which is useful when you have forgotten the password.

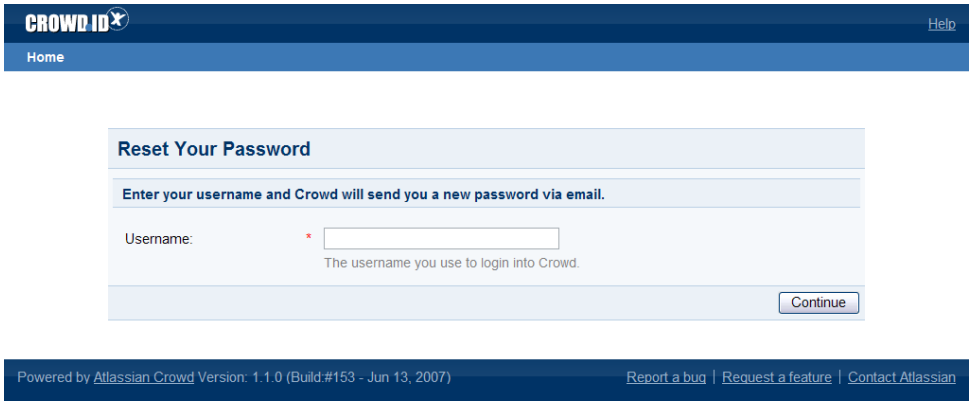
This will reset your password across all applications in your organisation, provided that the application is linked to Crowd.

To reset your password,

1. Access CrowdID.
2. Click the 'Forgotten your password?' link on the CrowdID Login page.
3. The 'Reset Your Password' page will appear. Type in your Crowd username and click the 'Continue' button.
4. A message will appear: 'Your new password is on the way!'. Click the 'Home' link at the top of the page.
5. You will receive an email message with your new password. Copy the password.
6. Log in to CrowdID using the new password.
7. Change your password to one you can remember easily.

If the change is successful, your password may also have changed in other Crowd-connected applications.

Screenshot: CrowdID Reset 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

Crowd Installation & Upgrade Guide

This page last changed on Mar 12, 2007 by rosie@atlassian.com.

Crowd Release Notes

This page last changed on May 05, 2008 by smaddox.

Crowd 1.4 has now been released — see the Crowd 1.4 Release Notes

Installation

Information for installing Crowd can be found here. If upgrading from a previous version, please follow the Upgrade Guide.

Crowd Release Notes

- Crowd 1.4 Release Notes
- Crowd 1.3 Release Notes
- Crowd 1.3 Beta Release Notes
- Crowd 1.3.2 Release Notes
- Crowd 1.3.1 Release Notes
- Crowd 1.2 Release Notes
- Crowd 1.2.2 Release Notes
- Crowd 1.2.1 Release Notes
- Crowd 1.1.2 Release Notes
- Crowd 1.1.1 Release Notes
- Crowd 1.1.0 Release Notes
- Crowd 1.0.7 Release Notes
- Crowd 1.0.6 Release Notes
- Crowd 1.0.5 Release Notes
- Crowd 1.0.4 Release Notes
- Crowd 1.0.3 Release Notes
- Crowd 1.0.2 Release Notes
- Crowd 1.0.1 Release Notes
- Crowd 1.0.0 Release Notes
- Crowd 0.4 Beta Release Notes
- Crowd 0.4.5 Beta Release Notes
- Crowd 0.4.4 Beta Release Notes
- Crowd 0.4.3 Beta Release Notes
- Crowd 0.4.2 Beta Release Notes
- Crowd 0.4.1 Beta Release Notes
- Crowd 0.3 Beta Release Notes
- Crowd 0.3.3 Beta Release Notes
- Crowd 0.3.2 Beta Release Notes
- Crowd 0.2 Beta Release Notes

Crowd 0.2 Beta Release Notes

This page last changed on Mar 11, 2007 by rosie@atlassian.com.

Crowd 1.4 has now been released — see the Crowd 1.4 Release Notes

Crowd 0.2

- Standalone version - Tomcat 5.5 with HSQL - .zip (59.5Mbs)
- Standalone version - Tomcat 5.5 with HSQL - .tar.gz (59.7Mbs)

Points of Interest

- There is an error when unzipping on the Windows platform, the archive integrity is fine and this will be fixed for the 0.3 release.
- The focus of this distribution is for JIRA and Confluence integration. Performance enhancements will be added for the 0.3 release which will allow large user-databases to be integrated.

Crowd 0.3.2 Beta Release Notes

This page last changed on Mar 11, 2007 by rosie@atlassian.com.

Crowd 1.4 has now been released — see the Crowd 1.4 Release Notes

The Crowd development team has released a new version of Crowd - 0.3.2.

This release addresses a Seraph SSO issue when integrating JIRA, Confluence and Bamboo.

http://jira.atlassian.com/secure/IssueNavigator.jspa?reset=true&pid=11291&fixfor=12540

You can now download Crowd from http://www.atlassian.com/Crowd

Cheers,

The Atlassian Crowd Development Team

Crowd 0.3.3 Beta Release Notes

This page last changed on Mar 11, 2007 by rosie@atlassian.com.

Crowd 1.4 has now been released — see the Crowd 1.4 Release Notes

The Crowd development team has released a new version of Crowd - 0.3.3.

This release addresses the following:

- Upgrade from Webwork 1 to Webwork 2
- Workaround for Active Directory to support CN forwards.

CRITICAL POSTGRES UPGRADE NOTES: http://jira.atlassian.com/browse/CWD-71

We started testing on IE7 and have noticed the CSS bugs and will work to get this addressed for the next build.

http://jira.atlassian.com/secure/IssueNavigator.jspa?reset=true&pid=11291&fixfor=12544

You can now download Crowd from http://www.atlassian.com/Crowd

Cheers,

The Atlassian Crowd Development Team

Crowd 0.3 Beta Release Notes

This page last changed on Mar 11, 2007 by rosie@atlassian.com.

Crowd 1.4 has now been released — see the Crowd 1.4 Release Notes

Crowd 0.3

- Standalone version - Tomcat 5.5 with HSQL - .zip (65.3 Mbs)
- Standalone version - Tomcat 5.5 with HSQL - .tar.gz (64.7 Mbs)

Points of Interest

- The focus of this distribution is on performance for a large number of users and groups when integrating JIRA, Confluence and Bamboo integration.

Crowd 0.4.1 Beta Release Notes

This page last changed on Mar 11, 2007 by rosie@atlassian.com.

Crowd 1.4 has now been released — see the Crowd 1.4 Release Notes

The Crowd development team has released a new version of Crowd - 0.4.1.

This addresses bugs which can be viewed through our JIRA issue tracker:

http://jira.atlassian.com/secure/IssueNavigator.jspa?reset=true&pid=11291&fixfor=12600

You can now download Crowd from http://www.atlassian.com/Crowd

Cheers,

The Atlassian Crowd Development Team

Crowd 0.4.2 Beta Release Notes

This page last changed on Mar 11, 2007 by rosie@atlassian.com.

Crowd 1.4 has now been released — see the Crowd 1.4 Release Notes

The Crowd development team has released a new version of Crowd - 0.4.2.

This addresses bugs which can be viewed through our JIRA issue tracker:

http://jira.atlassian.com/secure/IssueNavigator.jspa?reset=true&pid=11291&fixfor=12623

You can now download Crowd from http://www.atlassian.com/Crowd

Cheers,

The Atlassian Crowd Development Team

Crowd 0.4.3 Beta Release Notes

This page last changed on Mar 11, 2007 by rosie@atlassian.com.

Crowd 1.4 has now been released — see the Crowd 1.4 Release Notes

The Crowd development team has released a new version of Crowd - 0.4.3.

This addresses bugs which can be viewed through our JIRA issue tracker:

http://jira.atlassian.com/secure/IssueNavigator.jspa?reset=true&pid=11291&fixfor=12267

- Support for AD when there are more than 999 records in a search result.
- Reduced the number of necessary libs for a client application.
- Improved the 'build.properties' file configuration.

You can now download Crowd from http://www.atlassian.com/Crowd

Cheers,

The Atlassian Crowd Development Team

Crowd 0.4.4 Beta Release Notes

This page last changed on Mar 11, 2007 by rosie@atlassian.com.

Crowd 1.4 has now been released — see the Crowd 1.4 Release Notes

The Crowd development team has released a new version of Crowd - 0.4.4.

This addresses bugs which can be viewed through our JIRA issue tracker:

http://jira.atlassian.com/secure/IssueNavigator.jspa?reset=true&pid=11291&fixfor=12642

- Caching improvement for Confluence.
- Removed an additional attribute that was causing integration problems with SOAP services when using Active Directory.

You can now download Crowd from http://www.atlassian.com/Crowd

Cheers,

The Atlassian Crowd Development Team

Crowd 0.4.5 Beta Release Notes

This page last changed on Mar 11, 2007 by rosie@atlassian.com.

Crowd 1.4 has now been released — see the Crowd 1.4 Release Notes

The Crowd development team has released a new version of Crowd - 0.4.5.

This addresses bugs which can be viewed through our JIRA issue tracker:

http://jira.atlassian.com/secure/IssueNavigator.jspa?reset=true&pid=11291&fixfor=12652

- Improved Active Directory LDAP attribute filtering.
- UI improvements with new screen layouts.
- Spring TX management.

You can now download Crowd from http://www.atlassian.com/Crowd

Cheers,

The Atlassian Crowd Development Team

Crowd 0.4 Beta Release Notes

This page last changed on Mar 11, 2007 by rosie@atlassian.com.

Crowd 1.4 has now been released — see the Crowd 1.4 Release Notes

The Crowd development team has released a new version of Crowd - 0.4.

This release addresses several critical issues:

- Seraph Logout code fails to logout the user in Confluence, Bamboo and JIRA.
- Unable to search for a Principal by email address.
- Accept header authentication factor unreliable with Mozilla based browsers.
- Default 'localhost' configuration not added valid IP address of 127.0.0.1.

New features include:

- Allow all to authenticate.
- New LDAP connectors build off Spring LDAP Template with better performance enhancements.
- Support for LDAP filters

All Postgres DB will need to have the following command ran:

```
alter table "APPLICATIONDIRECTORIES" add column "ALLOWALLTOAUTHENTICATE" boolean;
```

http://jira.atlassian.com/secure/IssueNavigator.jspa?reset=true&pid=11291&fixfor=12266

You can now download Crowd from http://www.atlassian.com/Crowd

Cheers,

The Atlassian Crowd Development Team

Crowd 1.0.0 Release Notes

This page last changed on Mar 12, 2007 by rosie@atlassian.com.

Crowd 1.4 has now been released — see the Crowd 1.4 Release Notes

The Crowd development team has released Crowd 1.0.

This addresses bugs which can be viewed through our JIRA issue tracker:

- UI improvements with new screen layouts.
- Import and Export process for XML.
- LDAP Fixes for OpenLDAP and Microsoft Active Directory.
- Improved error reporting.
- Apache / Subversion support.

You can now download Crowd from http://www.atlassian.com/Crowd. If upgrading from a previous version, please follow the Upgrade Guide.

Atlassian JIRA (10 issues)				
Key	Summary	Pr	Status	
CWD-173	Implement an import and export function in Crowd			Closed
CWD-188	License update (when invalid) page should detail current license details.			Closed
CWD-184	Make Crowd's internal exception extend NestableException from commons-lang			Closed
CWD-180	Schema violation with LDAP and Groups/Roles			Closed
CWD-178	LDAP flags are incorrect for Active Directory/LDAP (Win2k3 domain)			Closed
CWD-150	Build fails			Closed
CWD-101	Unable to upgrade from 0.2 to 0.3.3			Closed
CWD-97	Apache mod Crowd integration			Closed
CWD-90	sso support for fisheye			Closed
CWD-62	500 page.			Closed

Cheers,

The Atlassian Crowd Development Team

Crowd 1.0.1 Release Notes

This page last changed on Mar 12, 2007 by rosie@atlassian.com.

Crowd 1.4 has now been released — see the Crowd 1.4 Release Notes

The Crowd development team has released Crowd 1.0.1.

This addresses 3 critical bugs which can be viewed through our JIRA issue tracker:

- Create new group/role broken using OpenLDAP.
- XFireFault exception: "No write method for property".
- Single sign on Seraph authentication fails when the host on a domain is not the same.

You can now download Crowd from http://www.atlassian.com/Crowd

Atlassian JIRA (3 issues)				
Key	Summary	Pr		Status
CWD-190	XFireFault exception: "No write method for property".			Closed
CWD-189	Create new group/role broken using OpenLDAP			Closed
CWD-82	Single sign on Seraph authentication fails when the host on a domain is not the same.			Closed

Cheers,

The Atlassian Crowd Development Team

Crowd 1.0.2 Release Notes

This page last changed on Mar 22, 2007 by rosie@atlassian.com.

Crowd 1.4 has now been released — see the Crowd 1.4 Release Notes

The Crowd development team has released Crowd 1.0.2.

This addresses bugs and feature improvements which can be viewed through our JIRA issue tracker:

- Included missing libraries for build archive.
- Added logging for input and output operations on SOAP services.
- Improved Jira caching for Crowd data.
- Added support for SSO beyond centralised authentication for Jive Forums.

You can now download Crowd from http://www.atlassian.com/Crowd

Atlassian JIRA (6 issues)				
Key	Summary	Pr	Status	
CWD-199	Missing libraries from the Crowd distribution			Closed
CWD-198	I renamed the docs from "Documentation" to "Crowd Documentation" (sorry). Can you please fix the "Help link?			Closed
CWD-197	XFire service input and output logging.			Closed
CWD-196	Improve the ability to configure the internal cache's used by the Crowd client and the Crowd console			Closed
CWD-195	Implement SSO for Jive Forums			Closed
CWD-193	Download archive is missing wsdl4j-1.5.2.jar			Closed

Cheers,

The Atlassian Crowd Development Team

Crowd 1.0.3 Release Notes

This page last changed on Apr 02, 2007 by rosie@atlassian.com.

Crowd 1.4 has now been released — see the Crowd 1.4 Release Notes

The Crowd development team has released Crowd 1.0.3.

This build is a mix of new features, bugs fixes and feature improvements:

- Improved SSO integration with Seraph for JIRA, Confluence and Bamboo.
- First builds of Apache Directory Server connector.
- Now supports directory server version that do not have the paged ldap control.
- Documentation updates.

You can now download Crowd from http://www.atlassian.com/Crowd

Atlassian JIRA (9 issues)				
Key	Summary	Pr	Status	
CWD-163	Administration Console allows login of unauthorized users			Closed
CWD-218	When an application is searching for its members from an LDAP repo AND an Internal Directory a HibernateException is thrown around trying to persist elements in a RemoteGroup.members			Closed
CWD-216	Crowd session token should be unique for each user, directory, machine			Closed
CWD-214	Login should logout any previous logged in users before a new login			Closed
CWD-179	Paged results control option for LDAP connectors.			Closed
CWD-177	Fisheye connector logs unnecessary exception.			Closed
CWD-175	Computers show up in the Principal list within Crowd from MSAD			Closed
CWD-169	NullPointerException on add OpenLDAP directory			Closed
CWD-121	Setting a "Remember Me" flag in Confluence, JIRA or Bamboo does not work, since the Token Reaper 'reaps' all session when the timeout is reached			Closed

Cheers,

The Atlassian Crowd Development Team

Crowd 1.0.4 Release Notes

This page last changed on Apr 11, 2007 by rosie@atlassian.com.

Crowd 1.4 has now been released — see the Crowd 1.4 Release Notes

The Crowd development team has released Crowd 1.0.4.

This build focused on bug fixes:

- Import export process was failing with Oracle DB.
- Implemented updating known attribute types on an LDAP object..
- Importing JIRA users is fixed for MySQL on a Unix like filesystem.

You can now download Crowd from http://www.atlassian.com/Crowd

Atlassian JIRA (6 issues)				
Key	Summary	Pr	Status	
CWD-221	Add documentation (marketing) section for Apache Directory Server			Closed
CWD-220	Implement RemoteDirectory updatePrincipal(RemotePrincipal) method for LDAP servers using InetOrgPerson as the Principal object.			Resolved
CWD-225	Import and export of Crowd fails when the database is Oracle			Closed
CWD-213	The Sitemesh and Webwork cleanup filters are being wrapped around the XFire requests.			Resolved
CWD-206	JIRA User Import Doesn't Set Groups on Principals			Resolved
CWD-172	Remove this error: SEVERE: No Store configured, persistence disabled			Resolved

Cheers,

The Atlassian Crowd Development Team

Crowd 1.0.5 Release Notes

This page last changed on Apr 19, 2007 by rosie@atlassian.com.

Crowd 1.4 has now been released — see the Crowd 1.4 Release Notes

The Crowd development team has released Crowd 1.0.5.

If you are running Confluence version 2.4.4 or before, you will need to upgrade the confluence/WEB-INF/lib/atlassian-user-XXXX-XX-XX.jar Atlassian User library to version 2007-04-05. The original library file will need to be backed up, removed, and then replaced with the new version listed above.

This build is mix of bug fixes, documentation improvements, and feature enhancements:

You can now download Crowd from http://www.atlassian.com/Crowd

Atlassian JIRA (15 issues)				
Key	Summary	Pr		Status
CWD-252	Active Directory filter does not exclude accounts which are no sAMAccountName type.			Closed
CWD-244	Set compile flags with maven build scripts to be vs. 1.4			Resolved
CWD-259	Username is not displayed in Confluence (2.4.X) when first logging in.			Closed
CWD-258	Domain for multihost single sign-on is not setting the cookie correctly.			Closed
CWD-257	VerifyTokenFilter missing from the Demo application.			Closed
CWD-256	Importer success screens display success even on an exception.			Resolved
CWD-254	review Installation documentation			Resolved
CWD-248	CLONE -The Sitemesh and Webwork cleanup filters are being wrapped around the XFire requests.			Closed
CWD-243	Document how you can not delete the Crowd console.			Resolved
CWD-242	You can delete the integrated Crowd application			Resolved
CWD-235	System error when no directory is selected when adding a group			Resolved
CWD-234	Add Websphere installation notes for Crowd.			Resolved
CWD-229	Transactions wrapping transactions. The transaction manager is not aware about the wrapping transaction.			Resolved
CWD-222	Crowd is not handling latin1 characters correctly			Resolved

CWD-226	browser window title should say 'View Application'			Resolved

Cheers,

The Atlassian Crowd Development Team

Crowd 1.0.6 Release Notes

This page last changed on Apr 16, 2007 by justen.stepka@atlassian.com.

The Crowd development team has released Crowd 1.0.6.

This build is a quick fix for problems reported with the SSO integration for multi host environments:

You can now download Crowd from http://www.atlassian.com/Crowd

Atlassian JIRA (3 issues)				
Key	Summary	Pr	Status	
CWD-265	Confluence displays the users fullname instead of email when integrated with Crowd			Resolved
CWD-263	Fails with exception on Search			Resolved
CWD-262	Improve the management of the Crowd domain during setup and in the Console.			Resolved

Cheers,

The Atlassian Crowd Development Team

Crowd 1.0.7 Release Notes

This page last changed on May 09, 2007 by justin.

The Crowd development team has released Crowd 1.0.7.

This release is a highly recommended upgrade from Crowd 1.0.6 and fixes 2 major issues found in 1.0.6:

Atlassian JIRA (5 issues)				
Key	Summary	Pr	Status	
CWD-296	LDAP update password implementation.			Resolved
CWD-316	Active Directory principals can signin with a blank password			Resolved
CWD-181	Continually asked to re-auth with Apache			Resolved
CWD-287	Reset password option for the Console			Resolved
CWD-233	javadoc SecurityServer			Resolved

Cheers,

The Atlassian Crowd Development Team

Crowd 1.1.0 Release Notes

This page last changed on Jun 22, 2007 by justen.stepka@atlassian.com.

The Atlassian Crowd team is proud to announce the release of Crowd 1.1.

This release contains a whole host of new features targeted at implementing OpenID, along with core updates to the Crowd Administration Console.

OpenID-enable your organisation with CrowdID

OpenID enables you to use a centralised identity to login to any website that supports OpenID. It opens up the possibilities of massive scale cross-domain SSO.

Think about all the accounts you have online: blogs, wikis, to-do lists, photo galleries. The list is endless. Even simple tasks such as leaving comments on someone else's blog may require you to register an account with that particular blogging system. This leaves you, as an end user, to set up and manage numerous accounts on each of these sites. With OpenID, rather than managing all these disparate accounts individually, users can manage their identity in one place via an authentication server.

With the ever-increasing adoption of this open authentication framework, including names such as Microsoft, AOL, Sun, Verisign and Firefox, expect to see many applications enabled for OpenID authentication.

CrowdID offers OpenID to an organisation's user base, allowing users to manage their online identity. Everything from configuring different profiles, managing trusted sites to reviewing authentication activity, is accessible from CrowdID. Administrators can set up whitelists/blacklists so that only trusted hosts can request authentication and can set up secure communication via SSL. All of the users can be managed via Crowd's security server, utilizing LDAP services from products such as Microsoft Active Directory.

Included with CrowdID is a sample OpenID client application, providing a working example of an OpenID enabled application. This will help developers kick start OpenID-enabling their applications.

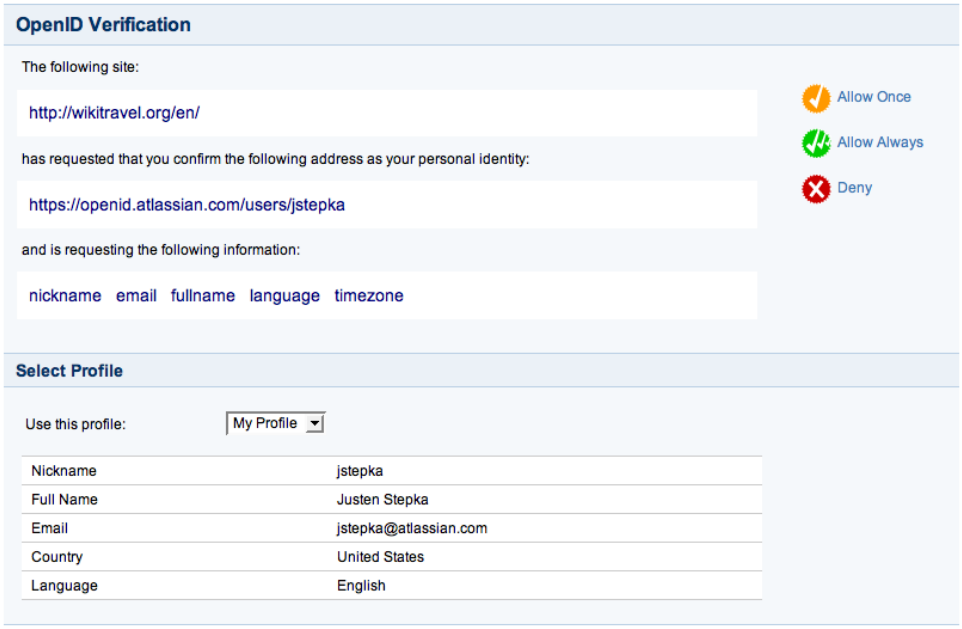
Using OpenID

Rather than registering and typing in your username and password on each site that you visit, OpenID allows you to type a URL similar to 'openid.mycompany.com/users/jstepka':

The OpenID website that you are logging in to will redirect you to CrowdID, which will ask you if you would like to allow authentication with the requesting site.

You can even choose to 'Always' allow authentication with particular OpenID sites, which allows pass-through authentication if you are already logged into your CrowdID server. If you do this, then when you visit the site later, simply provide your URL (e.g. 'openid.mycompany.com/users/jstepka') and you are in.

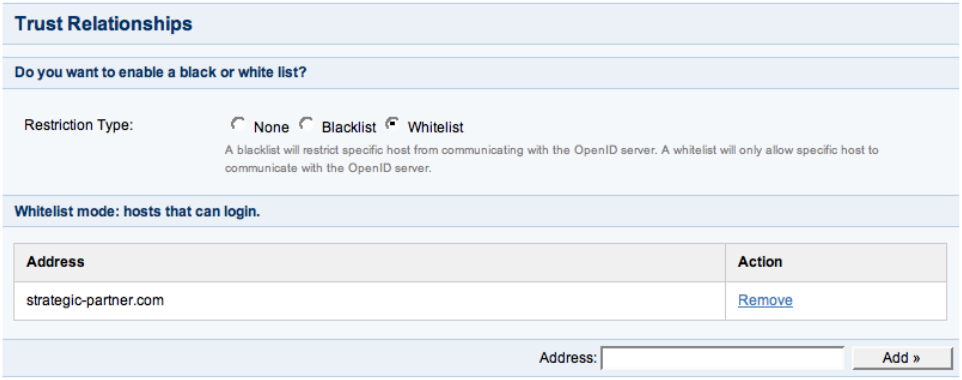
Think of it as 'Remember Me' for the whole internet!



'Blacklist' and 'Whitelist'

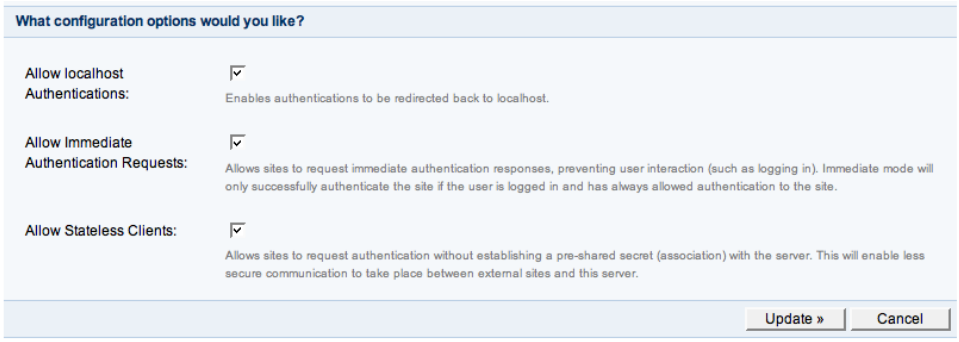
'Blacklists' and 'whitelists' allow administrators to lock down CrowdID their server so that, if necessary, it can only communicate with trusted hosts with which you have established relationships.

A blacklist will prevent specific hosts from communicating with the OpenID server. A whitelist will allow only specific hosts to communicate with the OpenID server.



OpenID Advanced Options

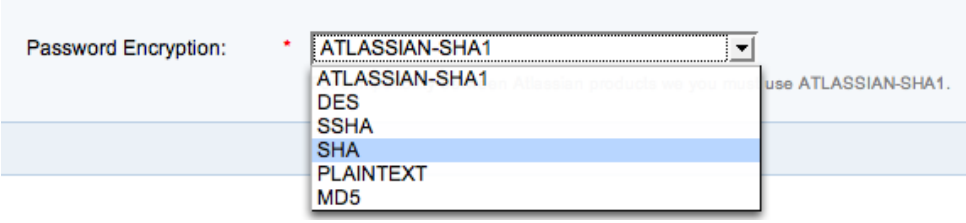
Some external sites implement security better than others. With CrowdID, you can pick how tough you want to be on OpenID sites that communicate with your Crowd OpenID server.



Crowd Console and Server Updates

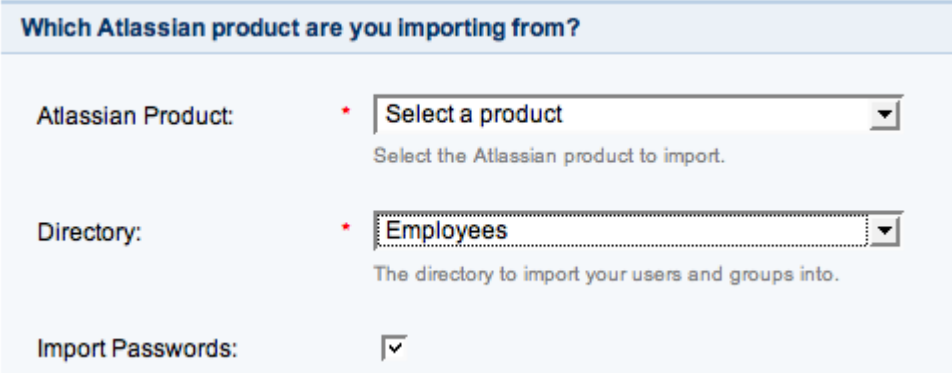
Choose Your Encryption Type

Every administrator has their own password policies. When using a Crowd Internal Directory you can now select the level of encryption you need.



Import Your JIRA and Confluence Passwords

Migration can be a pain. To ease your switch from existing Atlassian products, Crowd can now import your existing passwords!



Faster Web-Services

Crowd web-services now support GZip compression, improving the performance when downloading large amounts of data such as the all the members of a large group or when performing large search.

Improved Apache and Subversion Integration

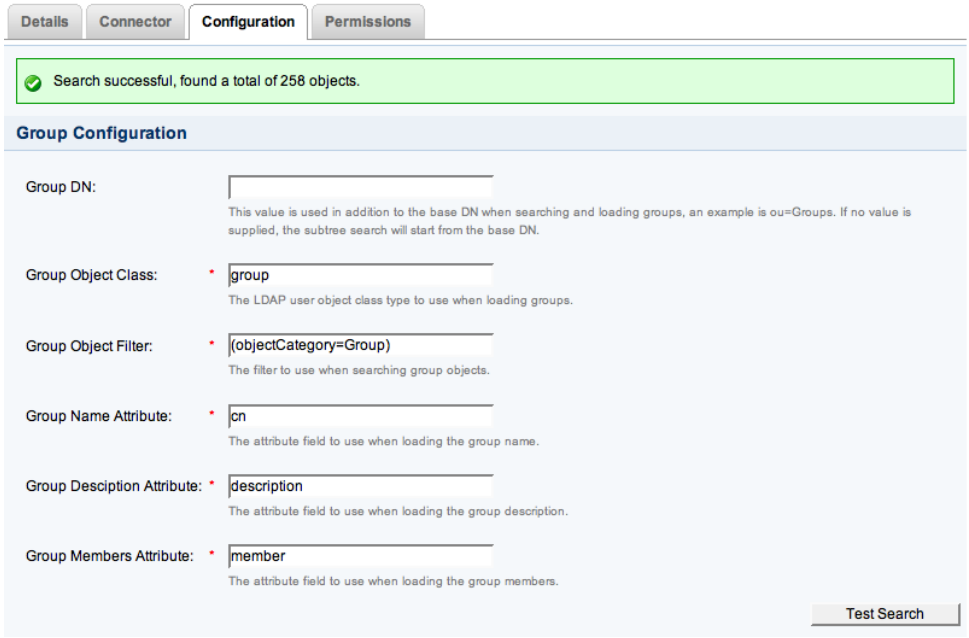
The Apache and Subversion library performance has been improved with the implementation of client-side caching of approved authentication requests.

Jive Forums 5.5 Support

The Jive Forums centralised authentication connector has been updated to support the new 5.5 major release of Jive Forums.

LDAP Configuration Tester

When setting up a Crowd LDAP connection you can now verify that your configuration connects as expected.



JIRA Issue Tracker

Atlassian JIRA (50 issues)				
Key	Summary	Pr	Status	
CWD-376	Export fails when an application does not have a description.			Resolved
CWD-359	'Blacklist' and 'Whitelist' options display intermittently in IE			Resolved
CWD-271	Login and Logoff for OpenID Server application.			Closed
CWD-245	Jive Forums 5.5 Support			Resolved
CWD-379	Change Password link on openid.atlassian.com throws 'No Action' error page			Resolved
CWD-377	Updating an Application will update the password for an application, even when you do not type in a new password			Closed
CWD-360	ORA-01000: maximum open cursors exceeded			Resolved
CWD-354	suggestions for the OpenID login page			Resolved
CWD-351	When logging out of Bamboo and anonymous mode is turned off, users still have the ability to create plans etc.			Resolved
CWD-343	Atlassian-user integration - get display name attribute from attributes if there rather than building display name adhoc.			Resolved

CWD-332	Test configuration buttons when creating an LDAP directory connector.		Resolved
CWD-323	Test connection utility for LDAP servers.		Resolved
CWD-320	Improve the importing of users from Confluence and JIRA so these users do not need to reset their passwords		Resolved
CWD-319	The export function of Crowd needs to have a flag to say don't export domain.		Resolved
CWD-318	ApacheDS crowd integration does not currently support the adding of groups		Resolved
CWD-313	The Apache module needs some kind of cache implemented similar to our other 'clients', to help improve performance around apache integration		Resolved
CWD-305	Add optional GZIP compression support for XFire SOAP services and client.		Resolved
CWD-304	Auto configure openid server as part of the setup process.		Resolved
CWD-302	Skin the OpenID Server		Closed
CWD-301	OpenID Client - Dummy Mode		Resolved
CWD-300	OpenID Server - dummy mode		Resolved
CWD-299	OpenID Client - Check Immediate		Resolved
CWD-298	OpenID Server - Check Immediate		Resolved
CWD-294	Test OpenIDClient Form Redirection		Resolved
CWD-292	OpenID Server Implementation		Resolved
CWD-291	Auto configure openid server as part of the setup process.		Closed
CWD-290	Upgrade webwork from 2.2.4 to 2.2.5		Resolved
CWD-288	Change application titles - not footers		Resolved
CWD-286	Skin Demo RP application		Resolved
CWD-285	Display attributes in the demo application upon login (store in session for display)		Resolved
CWD-284	Login and Logoff for OpenID demo relying party application.		Resolved
CWD-283	Configure request attributes for demo app		Resolved
CWD-280	Document OpenID server configuration		Closed

CWD-279	Attribute/Profile Management			Resolved
CWD-278	Authentication redirect from relying party.			Resolved
CWD-277	Skin Server			Resolved
CWD-276	Profile authentication history			Resolved
CWD-275	Enable/disable localhost relying parties.			Resolved
CWD-274	Whitelist and Blacklist Editor			Resolved
CWD-273	Force Association			Resolved
CWD-272	Reset password option.			Resolved
CWD-269	document the management of the Crowd domain during setup and in the Console			Resolved
CWD-246	Update documentation with new information about installing connector for 5.5.X version of JIVE.			Resolved
CWD-232	add 'SecurityServerClient'			Resolved
CWD-154	Apache DS connector			Resolved
CWD-144	Add 'green' success message to 'update' actions on Console.			Resolved
CWD-65	Explore OpenID support			Closed
CWD-368	Stray backslash on Groups administration screen			Resolved
CWD-365	Typo in hint for Password Encryption during initial directory setup			Resolved
CWD-325	Directory details tab shows empty pink error box			Resolved

Cheers,

The Atlassian Crowd Development Team

Crowd 1.1.1 Release Notes

This page last changed on May 07, 2008 by smaddox.

The Crowd development team has released Crowd 1.1.1.

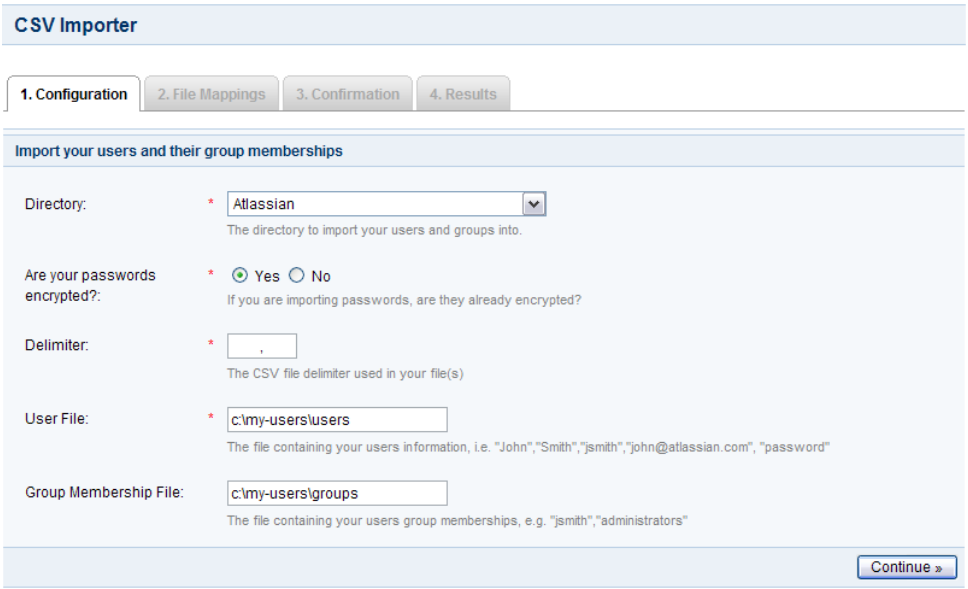
This release is a highly recommended upgrade from Crowd 1.1.0 since it provides a security fix to XWork, the technology underlying the web framework WebWork which is used by Crowd.

This release also contains a new CSV importer as well as fixes for some issues found in 1.1.0.

Importing Users and Groups from a CSV File

You can now copy users from an external directory or user base into Crowd via a CSV (comma-separated values) file.

The new CSV Importer allows you to specify a file containing user information, and optionally another file containing the groups to which the users belong. You can then map the CSV fields to the Crowd directory fields. After performing the import, Crowd sums up the results.
Screenshot: 'CSV Importer - Configuration'



Other Fixes in Crowd 1.1.1

Atlassian JIRA (20 issues)				
Key	Summary	Pr	Status	
CWD-445	Internal Directory search for Group by name is failing to aggregate the correct members			Closed
CWD-418	Chained directories are returning multiple groups/roles rather than aggregating group names.			Resolved
CWD-388	Paging principal sessions links are incorrect and do not function.			Resolved
CWD-309	The SunOne LDAP connector is not correctly authenticating users			Resolved

CWD-438	Users shown twice in JIRA		Closed
CWD-437	JIRA's logout via SSO does not clear it's session		Resolved
CWD-435	Exception using Seraph single-sign-on in Bamboo		Resolved
CWD-434	Searching for a group spanning multiple directories by its name will not amalgamate the principals		Resolved
CWD-425	Trim the application address when adding a valid application remote address.		Resolved
CWD-419	displayName attribute is not used with the JIRA connector		Resolved
CWD-414	The CSV Importer needs to display user results for duplicate entries i.e. users that have been ignored since they already exist in Crowd.		Resolved
CWD-400	JIRA attach screenshot does not write file to the filesystem when Crowdified.		Resolved
CWD-397	Document the CSV importer		Resolved
CWD-385	Generated tokens have non-HTML escaped characters.		Resolved
CWD-382	Create custom add successful page does not display directort connector page.		Resolved
CWD-352	Configure the number of paged results for an LDAP connector		Resolved
CWD-290	Upgrade webwork from 2.2.4 to 2.2.5		Resolved
CWD-53	CSV importer		Resolved
CWD-428	Change wording on the Atlassian importer		Resolved
CWD-407	Textual changes to new CSV-importer screens		Resolved

Cheers,

The Atlassian Crowd Development Team

Crowd 1.1.2 Release Notes

This page last changed on Sep 03, 2007 by justen.stepka@atlassian.com.

The Crowd development team has released Crowd 1.1.2.

Crowd 1.1.2 is a recommended upgrade from Crowd 1.1.1 since it provides improved integration with JIRA and Confluence, and tidier functionality for multiple directories.

For cross product compatibility, you must upgrade to the following versions of each product:

- Crowd 1.1.2 or later
- Bamboo 1.2.2 or later
- Confluence 2.5.6 or later
- JIRA 3.7.4 or later

Integration with JIRA user management

With Crowd 1.1.2, you can now turn external user management off in JIRA. This means that you can allow signup via JIRA, and you can manage your users within JIRA. Changes will flow through to Crowd.

JIRA has an automatic group membership feature. This means that any new user added through JIRA will automatically be a member of all groups which have the JIRA Users permission. In this way, you can ensure that a new user is automatically added to several groups when they sign up with JIRA.

RSS feeds

Crowd 1.1.2 fixes the problem experienced accessing RSS feeds from JIRA including retrieving JIRA issues via Confluence macros (e.g. the JIRA portlet macro).

Improved LDAP Performance

Crowd 1.1.2 now queries LDAP repositories in a more efficient manner that will give a dramatic performance increase for those with large numbers of LDAP groups.

Other Fixes in Crowd 1.1.1

Atlassian JIRA (23 issues)				
Key	Summary	Pr	Status	
CWD-314	Not able to Retrieve Issues (RSS) if JIRA is Integrated with Crowd			Resolved
CWD-297	JIRA performance improvements			Resolved
CWD-472	OpenID not working with LiveJournal			Resolved
CWD-540	CrowdID Install Documentation Mistake			Resolved
CWD-503	Cannot modify user profile when using Crowd authentication, fails with NullPointer on RemotePrincipal.getEmail()			Resolved
CWD-497	Crowd integration of Extranet JIRA has authentication problems			Resolved
CWD-496	requiresPasswordChange gets reset to false during login for an InternalDirectory			Resolved
CWD-495	Principals are being added with whitepace in their usernames			Resolved

CWD-492	Concurrent modification exception in JIRAAuthenticator logout code			Resolved
CWD-487	The upgrade manager should run after setup is complete			Resolved
CWD-484	When Confluence 2.6 releases we need to move the code from the bamboo-intergration module back into the atlassian-user module.			Resolved
CWD-478	Update Confluence Integration Doc			Resolved
CWD-462	Implement add user method of OSUser for JIRA			Resolved
CWD-452	JIRA user management should allow admins to update Crowd users			Resolved
CWD-448	Remote application's calls to removePrincipal(name) only removes the first principal it finds			Resolved
CWD-447	Remote application's calls to removeRole(name) only removes the first role it finds			Resolved
CWD-446	Remote application's calls to removeGroup(name) only removes the first group it finds			Resolved
CWD-421	Client JARs in client/lib are incomplete			Resolved
CWD-394	Full Name Search always returns all users			Closed
CWD-132	Windows service registration feature.			Resolved
CWD-420	Configuring multiple repositories may result in duplicate users			Resolved
CWD-390	Browser cookies cause NullPointerException when integrated with Confluence			Resolved
CWD-383	misspelling in wsdl - encryptedCrednetial			Resolved

Cheers,

The Atlassian Crowd Development Team

Crowd 1.2.1 Release Notes

This page last changed on Dec 09, 2007 by smaddox.

10 December 2007: The Crowd development team has released Crowd 1.2.1.

Crowd 1.2.1 fixes some installation problems. Other improvements include the sorting of groups by directory name then group name in the Application Browser.

Fixes in Crowd 1.2.1

Atlassian JIRA (15 issues)				
Key	Summary	Pr	Status	
CWD-648	Xalan is missing from the demo applications WEB-INF/lib folder.			Resolved
CWD-644	Seraph library compatibility issues result in java.lang.NoSuchMethodError: com.atlassian.crowd.integration.seraph.CrowdAuthenticator.getAuthType()Ljava/lang/String;			Resolved
CWD-642	build.xml fails to correctly copy the openid crowd.properties file			Resolved
CWD-638	build.bat no longer properly runs , preventing the environmental changes such as database dialects from be changed automaticly			Resolved
CWD-653	ports in the crowd.properties files are incorrect for the demo and openidserver applications with the distribution			Resolved
CWD-651	Confluence importer error with MySQL			Resolved
CWD-657	Acegi jar is missing from the client directory of the distribution			Resolved
CWD-650	Update the crowd distribution module parent POM version to version 10			Resolved
CWD-649	Update the atlassian-crowd module parent POM version to version 7			Closed
CWD-629	Error found in Internal Directory when a user requires a password change			Resolved
CWD-584	Adding a Principal to Sun DSEE 6.2 throws a NullPointerException			Resolved
CWD-506	LDAP fitlering only supports one fitler.			Resolved
CWD-499	Creating Groups and Principals fails on 2000			Resolved
CWD-342	Sort groups alphabetically or provide a pop-up window to search and			Resolved

CWD-289	choose groups (like Confluence has) Sort groups by name when selecting groups that can access an application			Resolved

Cheers,

The Atlassian Crowd Development Team

Crowd 1.2.2 Release Notes

This page last changed on Jan 15, 2008 by smaddox.

16 January 2008: The Crowd development team has released Crowd 1.2.2.

Crowd 1.2.2 upgrades its packaged version of Apache Tomcat to version 5.5.25, to fix some reported Apache Tomcat vulnerabilities. Tomcat is supplied as the application server in the Crowd Standalone distribution.

This release also solves some problems with the Crowd build and resolves the incompatibility between Crowd single sign-on and the new JIRA/Confluence trusted application feature.

Complete List of Fixes in Crowd 1.2.2

Atlassian JIRA (13 issues)				
Key	Summary	Pr	Status	
CWD-738	Allow configuring of request logs in the Crowd client libraries.			Resolved
CWD-710	Update Tomcat to 5.5.25 to fix reported vulns			Closed
CWD-654	Xalan is missing from the demo applications WEB-INF/lib folder.			Closed
CWD-703	Crowd OpenID WAR file is missing commons-logging jar.			Resolved
CWD-793	Receiving error when trying to build Crowd 1.2.2: taskdef class com.oopsconsultancy.xmltask.ant.XmlTask cannot be found			Resolved
CWD-739	Concurrency Issue in client libraries may result in multiple caches			Resolved
CWD-728	The Internal Directory is throwing a java.lang.IndexOutOfBoundsException: Index: 0, Size: 0 on requiresPasswordChange()			Resolved
CWD-727	Poor logging of a Token miss in the In-memory token cache.			Resolved
CWD-711	The HTTPAuthenticator isAuthenticated method should initially check for a token			Resolved
CWD-699	Crowd SSO is incompatible with JIRA 3.12/Confluence 2.7 trusted application feature.			Resolved
CWD-667	Crowd user caching in JIRA delayed			Resolved
CWD-665	Create an XFire fault logging handler			Resolved
CWD-706	Fix logging on startup for the OpenID Server. Stop the logging of Hibernate INFO.			Resolved

Cheers,

The Atlassian Crowd Development Team

Crowd 1.2 Release Notes

This page last changed on May 05, 2008 by smaddox.

The Atlassian Crowd team is delighted to present Crowd 1.2.

Crowd 1.2 is a major release that focuses on enhanced integration, security and usability. Crowd's directory permissions now allow finer-grained control, so that you can define the permissions per application. The Group and Role Browsers now display group/role membership. We have enhanced group management in the existing Jive Forums and Apache/Subversion connectors. Our NTLM plugin offers SSO (single sign-on) for JIRA and Confluence via NTLM desktop authentication. A new connector lets you integrate your Acegi security solution with Crowd. And you can import your Bamboo users directly into a Crowd directory.

We'd like to say a special thank you to CustomWare for their assistance with deployment and testing of the NTLM plugin.

Stop Press — 27 February 2008: We got a little bit ahead of ourselves with our announcement of full NTLM support in Crowd 1.2. The NTLM plugins for JIRA and for Confluence are provided and supported by a third party, not by Atlassian.

Highlights of this release:

Error formatting macro: toc: java.lang.NullPointerException

Responding to your feedback:

8 new feature requests implemented

68 votes satisfied

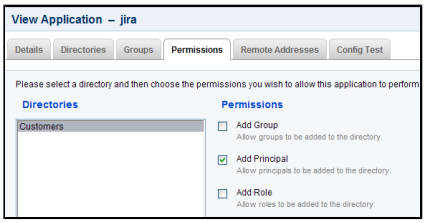
Your votes and issues help us keep improving our products, and are much appreciated.

Upgrading to Crowd 1.2
You can download Crowd from the Atlassian website. If upgrading from a previous version, please read the Upgrade Notes.

Highlights of Crowd 1.2

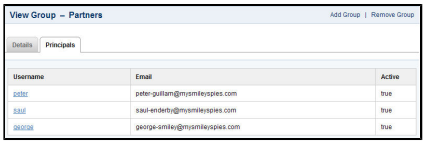
Directory Permissions per Application

- Directory permissions determine whether groups, principals and roles can be added, modified or deleted.
- Before this release, permissions were set at directory level only. Permissions therefore applied across all applications associated with the directory.
- With Crowd 1.2, directory permissions can be set for each application. For example, you could enable the 'Add Principal' permission on the 'Employees' directory for JIRA but disable the permission for Confluence.
- See the screenshot below, and take a look at an example.



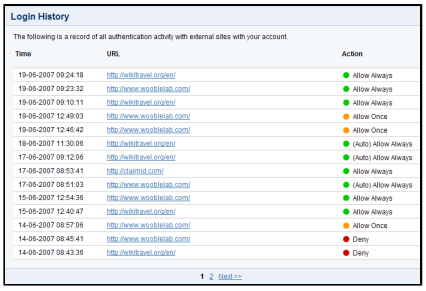
Group and Role Membership Browser

- A new 'Principals' tab in the Group Browser shows all principals belonging to a group.
- You can view membership in the Role Browser too.
- Read the documentation.



Improved Browser for OpenID Login History

- Instead of showing all login history on a single page, the Login History screen now divides the history into pages, for easier viewing.
- To move between pages, click 'Next', 'Prev' or a specific page number.
- In the 'Action' column, a new item '(Auto) Allow Always' tells you which logins were allowed automatically because of a previous 'Allow Always' instruction.



NTLM Support

- NTLM is a Microsoft authentication protocol that allows you to access a website using your desktop login. The protocol utilises an integration between Microsoft Internet Explorer and Active Directory. When using this feature, users will only need to log in to their desktop to access NTLM-integrated applications.

- JIRA and Confluence NTLM connectors are now supported with Crowd 1.2.
- Read the instructions on setting up Confluence and JIRA NTLM support in Crowd.

Improved Integration with Jive Forums

- Crowd 1.2 provides support for group management in Jive Forums.
- Groups and group memberships are now pulled from Crowd.
- You can use the Jive Forums admin console to define application permissions associated with groups.
- This allows Crowd to manage Jive Forums groups and memberships and Jive Forums to handle the permissions associated with the groups.
- Read the documentation.

Acegi Application Connector

- Crowd 1.2 provides a built-in application connector for Acegi, a security solution with a particular emphasis on Spring Java/JEE applications.
- Read the documentation.

Group-Based Authorisation Added for Subversion

- Crowd allows you to password-protect your SVN repository running under Apache.
- You can now also configure fine-grained access by group as well as by user.
- Read more about the Crowd Subversion connector.

New Importer for Bamboo Users

- Our new Bamboo importer allows you to copy your Bamboo users into a Crowd directory.
- Read the documentation.

Plus Over 70 Improvements and Bug-Fixes

Atlassian JIRA (77 issues)			
Key	Summary	Pr	Status
CWD-580	Events, EventType and Event Listeners are not		Resolved

	being exported as part of the XML backup			
CWD-526	Editing groups in Crowd has no effect in Bamboo			Resolved
CWD-398	;jsessionid added to all Crowd links			Resolved
CWD-314	Not able to Retrieve Issues (RSS) if JIRA is Integrated with Crowd			Resolved
CWD-297	JIRA performance improvements			Resolved
CWD-249	Adjust build process to publish maven2 client poms.			Resolved
CWD-534	Upgrade Crowd to Spring Framework 2.0.6 from 1.2.x			Resolved
CWD-388	Paging principal sessions links are incorrect and do not function.			Resolved
CWD-336	No date sent in email headers for messages sent by Crowd			Closed
CWD-637	NullPointerException in DefaultCookieHandler.setCookie			Resolved
CWD-625	If an allowed Principal Attribute is null it is not possible to update this in LDAP			Resolved
CWD-599	When creating and viewing an LDAP connector, we have been			Resolved

	displaying the password as clear text, this should at least be a password field			
CWD-597	License user-limit check event should not execute for unlimited licenses			Resolved
CWD-593	Upgrade to Atlassian-Extras 1.9			Resolved
CWD-588	Jive Forums remote authentication is not working			Resolved
CWD-582	If the two core event listeners do not exist add them via an upgrade task.			Resolved
CWD-579	Role Tab shows the correct number of roles however they all show up as the principal name			Resolved
CWD-578	Allow a crowd administrator to recalculate the user total for a Crowd install			Resolved
CWD-577	Remove Group link on View Principal does not contain a valid directory ID			Resolved
CWD-576	Document Crowd installation on JBoss			Resolved
CWD-575				Resolved

CWD-573	Document the 'config test' tab		Resolved
CWD-567	Multiple cookies are wrote back to the browser during an authentication.		Resolved
CWD-556	HSQL context path storage issues when not using start_crowd.bat/sh		Resolved
CWD-552	Atlassian applications hang and can not start when integrated with Crowd under the same VM.		Resolved
CWD-540	Data imports fail when no application-group associations are in place.		Resolved
CWD-539	CrowdID Install Documentation Mistake		Resolved
CWD-537	Need and EAR/WAR download to use other application servers		Resolved
CWD-524	Method to create a token for a principal without performing an authentication.		Closed
CWD-517	Full Name attribute (displayName/firstName +surname) used differently by atlassian-user and JiveForums		Resolved
	Documentation update for		

	'Upgrading Crowd' as per customer's comment			
CWD-516	JIRA breaks with retrieveUserMetaProperties NPE after adding user in Crowd			Resolved
CWD-514	Move Crowd to use Webwork 2.2.6			Resolved
CWD-513	Move Crowd to use Seraph 0.9			Resolved
CWD-508	Release Crowd EAR/ WAR edition			Resolved
CWD-504	Crowd should be offered as a EAR/WAR package in addition to standalone			Resolved
CWD-503	Cannot modify user profile when using Crowd authentication, fails with NullPointer on RemotePrincipal.getEmail()			Resolved
CWD-502	Unauthenticated user causes session nuking in Crowdified JIRA			Resolved
CWD-501	OpenID history browser			Closed
CWD-500	Directory CRUD permissions on an Application-by-Appliction basis.			Resolved
CWD-497	Crowd integration of Extranet JIRA has authentication problems			Resolved
CWD-496	requiresPasswordChange gets reset to false during			Resolved

CWD-495	login for an InternalDirectory Principals are being added with whitepace in their usernames		Resolved
CWD-492	Concurrent modification exception in JIRAAuthenticator logout code		Resolved
CWD-489	change the Crowd Upgrade Guide to only copy the password from the crowd.properties files, not copy the entire files		Closed
CWD-488	The build.properties file and Ant associated ant task should not overwrite the password attribute in the crowd.properties file		Resolved
CWD-487	The upgrade manager should run after setup is complete		Resolved
CWD-484	When Confluence 2.6 releases we need to move the code from the bamboo-intergration module back into the atlassian-user module.		Resolved
CWD-465	Improve the current Jive integration to provide support for Group management		Resolved

CWD-464	Email address validation is not RFC-2822 compliant		Closed
CWD-462	Implement add user method of OSUser for JIRA		Resolved
CWD-459	Update the SecurityServer SOAP API to enable editing/ updating groups		Resolved
CWD-452	JIRA user management should allow admins to update Crowd users		Resolved
CWD-435	Exception using Seraph single-sign-on in Bamboo		Resolved
CWD-430	CrowdID Not Signing User Attributes Like Nickname or Email		Resolved
CWD-425	Trim the application address when adding a valid application remote address.		Resolved
CWD-421	Client JARs in client/ lib are incomplete		Resolved
CWD-419	displayName attribute is not used with the JIRA connector		Resolved
CWD-417	Libraries in client directory are not enough		Resolved
CWD-415	Tomcat doesn't start if it runs both		Resolved

CWD-414	Crowd and Confluence The CSV Importer needs to display user results for duplicate entries i.e. users that have been ignored since they already exist in Crowd.			Resolved
CWD-392	No group integraton into Subversion			Resolved
CWD-380	Sources gets added to download archive			Resolved
CWD-373	Improve the build process for source releases			Resolved
CWD-349	Create a Bamboo to Crowd Principal and Group importer.			Resolved
CWD-348	When switching from internat authentication to Crowd authentication (using seraph?), exception is throw during login.			Resolved
CWD-281	Build script improvements			Resolved
CWD-209	Maven 2 repository for Crowd client components.			Resolved
CWD-185	The import/ export is confined to a given instance, we need to make it so the XML file can be used on			Resolved

	any Crowd deployment.			
CWD-135	Support NTLM			Closed
CWD-19	Acegi Connector			Resolved
CWD-618	View Principal is throwing a RemoteException when trying to view the Roles of a Principal			Resolved
CWD-617	Browse Principal is not showing an email address for the principal returned.			Resolved
CWD-525	Login to jira with an existing cookie (non-crowd) shows a nullpointer			Resolved
CWD-442	View members of the group or role			Resolved
CWD-428	Change wording on the Atlassian importer			Resolved
CWD-407	Textual changes to new CSV-importer screens			Resolved
CWD-390	Browser cookies cause NullPointerException when integrated with Confluence			Resolved

Crowd 1.3.1 Release Notes

This page last changed on Mar 19, 2008 by smaddox.

20 March 2008
The Crowd development team has released Crowd 1.3.1. This is a bug-fix release, which solves some problems in Crowd 1.3.

Don't have Crowd 1.3 yet?
Take a look at the new features and other highlights in the Crowd 1.3 Release Notes.

Download Latest Version

Complete List of Fixes in Crowd 1.3.1

Atlassian JIRA (13 issues)				
Key	Summary	Pr		Status
CWD-899	When creating an LDAP based directory a password algorithm attribute is being set for all directory types regardless if they use one or not.			Resolved
CWD-924	SSO failure when authenicating two users in two tabs (in one browser)			Resolved
CWD-920	OpenLDAP MD5 encrypted password stored as plain text			Resolved
CWD-914	Viewing OpenLDAP Directoy Connector Info throws an exception			Resolved
CWD-900	Paged result size should not persist on directories that have not have "Use Paged Results" enabled.			Resolved
CWD-898	Crowd 1.3 UI is not compatible with IE 6			Resolved
CWD-875	User groups list in directory should sort alpha-numeric rather than natural.			Resolved
CWD-782	Textual changes on new directory importer screens			Resolved
CWD-527	IllegalDataException from active-directory authentication failure			Resolved
CWD-916	View Principal/User sessions in the Crowd console directory links broken			Resolved
CWD-909	User Name RDN Attribute field is not populated for Delegated Authentication directory screen			Resolved
CWD-561	Support the 'uid' and 'cn' attribute with the inetorgperson object at the same time			Resolved
CWD-439				Resolved

Errors in the Confluence logs about Crowd (XFire prolog EOF)

Cheers,

The Atlassian Crowd Development Team

Crowd 1.3.2 Release Notes

This page last changed on Apr 02, 2008 by smaddox.

3 April 2008

The Crowd development team presents Crowd 1.3.2. The main purpose of this release is to provide compatibility with the upcoming release of Confluence 2.8. We have updated Crowd's atlassian-user integration module to support an interface change in Confluence.

This release also fixes a problem occurring when an application attempts to add a user, where multiple directories are mapped to the application.

Don't have Crowd 1.3 yet?

Take a look at the new features and other highlights in the Crowd 1.3 Release Notes.

Download Latest Version

Complete List of Fixes in Crowd 1.3.2

Atlassian JIRA (3 issues)				
Key	Summary	Pr	Status	
CWD-959	Creation of Principals from a client application (JIRA/Confluence) will fail silently when there is multiple directories, one of those being an Internal Directory.			Resolved
CWD-952	Upgrade atlassian-user to be compatible with interface change for Confluence 2.8			Resolved
CWD-347	Crowd and direct LDAP conenction demanding different DNs (at least against ApacheDS)			Resolved

Cheers,

The Atlassian Crowd Development Team

Crowd 1.3 Beta Release Notes

This page last changed on May 07, 2008 by smaddox.

20 February 2008

Crowd 1.3 will be launched early in March 2008. A beta release is currently undergoing internal testing. These release notes apply to Crowd 1.3 beta. We'll publish the final release notes with the release of Crowd 1.3.0.

If you would like to participate in testing the beta release, please contact Crowd Support.

Upgrading to Crowd 1.3 Beta
If upgrading from a previous version, please read the Upgrade Notes.

What's Coming in Crowd 1.3

1 LDAP Authentication with Crowd Groups and Roles

- Crowd 1.3 provides a new directory type, Delegated Authentication, combining the features of a Crowd internal directory with delegated LDAP authentication.
- This allows you to have your users authenticated via an external LDAP directory while managing the groups and roles in Crowd.
- Use Crowd's flexible and simple group management when the LDAP groups do not suit your requirements. For example, you can set up a group configuration in Crowd for use with Confluence and other Atlassian products.
- Avoid the performance issues which might result from downloading large numbers of groups from LDAP.
- Use the new Directory Importer, described below, to synchronise your LDAP users with your Crowd directory.
- When a user logs in for the first time, Crowd automatically adds them to the Crowd directory if not already present.

2 Cross-Directory User Importer

- Our new Directory Importer allows you to copy your users from one directory into another.
- Provided that the directory is defined in Crowd, you can copy from and to any directory type.
- For example, you might import users, groups, roles and memberships from an LDAP directory to a new Delegated Authentication directory (described above) so that you can manage the users, groups and roles in Crowd while allowing users to log in with their LDAP passwords.
- Read about the Directory Importer.

3 Streamlined User Interface

- The Crowd Administration Console has a new menu structure and an enhanced look-and-feel. sed functions, so that an administrator has fewer steps to perform and interaction is more intuitive.
- The 'Help' links on the Administration Console point directly to the relevant documentation pages.

4 Simplified Installation and Setup

- Database configuration is now part of the Setup Wizard, which will update the configuration files based on the options you select.
- You can choose between a JNDI datasource (i.e. server-managed) or a simpler JDBC configuration.
- When upgrading, you can import an XML backup of your Crowd database or connect to an existing database via the Setup Wizard. This means that you don't have to go through the whole Setup Wizard, nor do a manual backup and restore of your Crowd database files.

Logging and Profiling Configuration via Console

- Enable profiling and configure your logging levels via the Crowd Administration Console.

Improved Performance and Efficiency

- You'll notice faster search results on the Administration Console screens, such as the Application Browser and User Browser, etc.
- That annoying 'POSTDATA has expired' message no longer appears when you click the 'Back' button.
- Search results returned to a Crowd application are now sorted alphabetically — such as the list of groups shown in a Confluence group picker.
- We've fixed the Hibernate StaleStateException error that was causing occasional performance degradation and authentication failures.
- You can choose to store the login session tokens in the Crowd database (as done prior to Crowd 1.3) or in memory (new option as from Crowd 1.3). Depending upon your installation, in-memory storage could greatly improve response times during authentication. Read about configuring token storage.
- Gzip compression of Crowd Security Server output is now optional. You can turn it on or off via the Crowd Administration Console. Some reasons why you may want to turn Gzip compression off:
  - It may be easier to debug problems using uncompressed data.
  - Some agents, such as older versions of Internet Explorer, have problems with the Gzip format.

Highlights for the Developers

- The Java client library API has been upgraded. Read more about the API changes and the upgrade notes.
- You can pass the `crowd.properties` file to a client application as an environment variable.

Updates and Fixes in this Release

Atlassian JIRA (68 issues)				
Key	Summary	Pr		Status
CWD-738	Allow configuring of request logs in the Crowd client libraries.			Resolved
CWD-654	Xalan is missing from the demo applications WEB-INF/lib folder.			Closed
CWD-768	Hibernate DAOs for Principals and Groups close the Hibernate Session when adding			Resolved
CWD-703	Crowd OpenID WAR file is missing commons-logging jar.			Resolved
CWD-639	Crowd hanging client applications, error with token manager			Resolved
CWD-466	Storing login tokens in an external DB is inefficient			Resolved
CWD-350	Tuckey rewrite filter dials home by doing a DNS lookup.			Resolved
CWD-897	Generic LDAP Directory type is displayed as OpenLDAP not Generic			Closed
CWD-882	Unalble to update the 'active' flag of an Application			Resolved

CWD-838	Updating any directory type in Crowd has multiple validation problems.			Resolved
CWD-830	Change Crowd WAR deployment to zip archive.			Resolved
CWD-829	When updating a Delegated or Connector based directory, required fields are not marked as required.			Resolved
CWD-828	When updating an Internal Directory, there is no validation performed on the Configuration tab			Resolved
CWD-824	Session timeout during the installation should be larger than 5 minutes			Resolved
CWD-823	JDBC connection should default to MySQL			Resolved
CWD-822	crowd-init.properties value not set error message during startup is not useful			Resolved
CWD-817	Default results per page to 100			Resolved
CWD-806	Fix log4j.properties so dates are displayed in log files.			Resolved
CWD-805	Crowd's Add Directory Screen indicates we support Open Directory.			Resolved
CWD-800	When associating a Group/Role to a Principal in the Demo application, an error is displayed			Resolved
CWD-790	Have you seen the client/lib directory lately? The current count is about 46 JAR files!			Resolved
CWD-775	Add Logging & Profiling functionality into Crowd Admin screen.			Resolved
CWD-767	Crowd's Client libraries should be slimmed down to a single JAR file containing all required classes for a Crowd Client			Resolved
CWD-765	File missing in 1.2.2 release			Resolved
CWD-758	Hibernate StaleStateExceptions in Crowd			Resolved
CWD-757	Crowd with delegated LDAP auth - update documentation for Bamboo-Crowd integration			Resolved
CWD-739	Concurrency Issue in client libraries may result in multiple caches			Resolved
CWD-731	OGNL Exception being thrown when updating a principal			Resolved

CWD-728	The Internal Directory is throwing a java.lang.IndexOutOfBoundsException: Index: 0, Size: 0 on requiresPasswordChange()		Resolved
CWD-727	Poor logging of a Token miss in the In-memory token cache.		Resolved
CWD-726	java.lang.IllegalStateException: Can't overwrite cause exception seen in Crowd		Resolved
CWD-724	Configuration classs for the LDAP importer		Resolved
CWD-723	LDAP Importer, to migrate data from one directory into another.		Resolved
CWD-720	Enable import from XML in the setup process		Closed
CWD-716	Error when attempting to remove a group		Resolved
CWD-711	The HTTPAuthenticator isAuthenticated method should initially check for a token		Resolved
CWD-700	The isMember call for groups can be slow for very large groups in an Internal Directory		Closed
CWD-699	Crowd SSO is incompatible with JIRA 3.12/Confluence 2.7 trusted application feature.		Resolved
CWD-694	ehcache-1.2.3.jar is missing from client/lib folder.		Resolved
CWD-688	Help links directly in the administration console		Closed
CWD-686	Sort groups, users and roles before returning results to the security server client		Resolved
CWD-685	Write System Info page to atlassian-crowd.log on Crowd startup		Resolved
CWD-675	remove "cache-control: no-store" on search results pages		Resolved
CWD-669	Adding group/role with prefixed space causes Hibernate error		Resolved
CWD-666	Persistence system should use c3p0 so hibernate's default pooling system is not used.		Resolved
CWD-650	Update the crowd distribution module parent POM version to version 10		Resolved
CWD-649	Update the atlassian-crowd module parent POM version to version 7		Closed
CWD-646	Move FishEye connector outside crowd-core		Resolved

CWD-645	Use Spring dependency injection for SecurityServerClient and HttpAuthenticator in Crowd applications			Resolved
CWD-633	Allow the crowd.properties file to be passed to a Client application as an environment variable			Resolved
CWD-622	Make SecurityServerClient not static			Closed
CWD-586	start_crowd.sh and build.sh fail on Solaris			Resolved
CWD-584	Adding a Principal to Sun DSEE 6.2 throws a NullPointerException			Resolved
CWD-499	Creating Groups and Principals fails on 2000			Resolved
CWD-481	Support CRYPT encryption in OpenLDAP connector			Resolved
CWD-427	OpenLDAP Connector should default to SSHA encryption.			Resolved
CWD-389	GZip compression is optional through the administration console.			Resolved
CWD-208	Mixed authentication and authorization support for external directory connectors.			Resolved
CWD-149	Config Test doesn't appear to obey Directory and Group rules			Resolved
CWD-855	OGNL exceptions are thrown when remoing Groups and Roles in the Demo app			Resolved
CWD-849	Rationalise the path to crowd-init.properties that's displayed on startup			Resolved
CWD-818	Admin Console: Selected tab CSS needs tweaking for Windows compatability			Resolved
CWD-799	When creating a Group/ Role to a Principal in the Demo application, an exception is thrown.			Resolved
CWD-798	When adding a Group or Role via the Demo app, the description field is not being persisted.			Resolved
CWD-706	Fix logging on startup for the OpenID Server. Stop the logging of Hibernate INFO.			Resolved
CWD-570	First Name not being displayed from Apache DS			Resolved
CWD-453	Crowd core jar breaks in Grails, need a new slimmed-down client jar			Resolved
CWD-847	Error message is confusing when no			Resolved

directories are mapped to an application

Crowd 1.3 Release Notes

This page last changed on May 07, 2008 by smaddox.

4 March 2008

The Atlassian Crowd team is delighted to present Crowd 1.3. This release includes innovative solutions for LDAP group administration, cross-directory user imports and a streamlined management interface.

A new directory type allows you to combine the features of a Crowd directory with authentication delegated to an LDAP directory. This means that you can use Crowd's flexible group management when the LDAP groups do not suit your requirements. For example, set up a simple group configuration for use with Confluence, JIRA and other Atlassian products.

Our new Directory Importer allows you to copy your users from one directory into another — from and to any type of directory. For example, you can copy users, groups and roles from an LDAP directory to a Crowd directory, or vice versa.

The Crowd Administration Console has a new menu structure with an enhanced look and feel. It's easier to find the functions that you perform most often and interaction is more intuitive.

Installing and setting up Crowd is simpler and faster. Database configuration is now part of the Setup Wizard. When upgrading, you have the option to import your data from an XML backup or point Crowd at your existing database, and so bypass most of the Setup Wizard.

To speed up troubleshooting, you can configure your logging levels and enable performance profiling via the Administration Console. There's a bucketful of improvements in performance and efficiency, and many other fixes and enhancements.

Highlights of this release:

Error formatting macro: toc: java.lang.NullPointerException

Responding to your feedback:

6 new feature requests implemented

36 votes satisfied

Your votes and issues help us keep improving our products. Keep 'em coming!

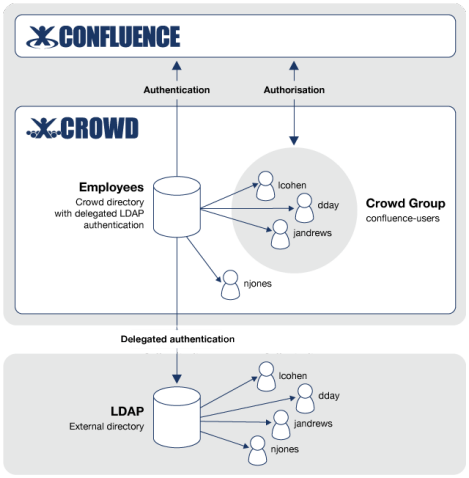
Upgrading to Crowd 1.3
You can download Crowd from the Atlassian website. If upgrading from a previous version, please read the Crowd 1.3 Upgrade Notes.

Highlights of Crowd 1.3

LDAP Authentication with Crowd Groups and Roles

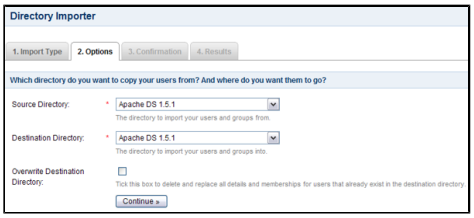
- Crowd 1.3 provides a new directory type, Delegated Authentication, combining the features of a Crowd internal directory with delegated LDAP authentication.
- This allows you to have your users authenticated via an external LDAP directory while managing the groups and roles in Crowd.

- Use Crowd's flexible and simple group management when the LDAP groups do not suit your requirements. For example, you can set up a group configuration in Crowd for use with Confluence, JIRA and other Atlassian products.
- Avoid the performance issues which might result from downloading large numbers of groups from LDAP.
- Use the new Directory Importer, described below, to synchronise your LDAP users with your Crowd directory.
- When a user logs in for the first time, Crowd automatically adds them to the Crowd directory if not already present.



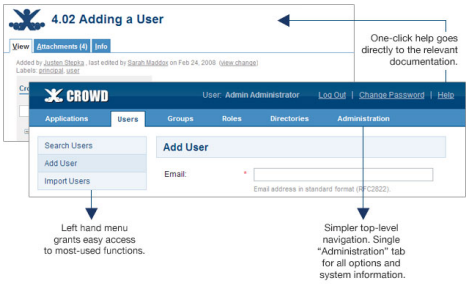
Cross-Directory User Importer

- Our new Directory Importer allows you to copy your users from one directory into another.
- Provided that the directory is defined in Crowd, you can copy from and to any directory type.
- For example, you might import users, groups, roles and memberships from an LDAP directory to a new Delegated Authentication directory (described above) so that you can manage the users, groups and roles in Crowd while allowing users to log in with their LDAP passwords.
- Read about the Directory Importer.



Streamlined User Interface

- The Crowd Administration Console has a new menu structure and an enhanced look and feel.
- A left-hand menu grants easy access to the functions you use most often, such as searching for a user or group.
- A single 'Administration' tab holds the configuration options, system information and backup/restore functions.
- In the interests of simplicity, we've changed the term 'principal' to 'user' throughout.
- When you click a 'Help' link, the relevant documentation page opens immediately.

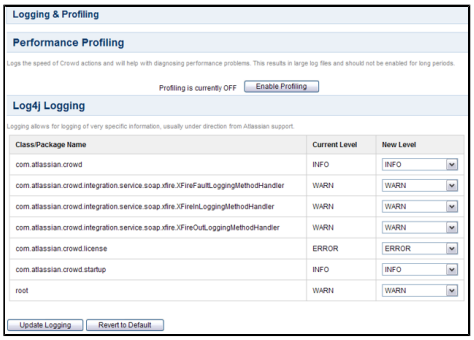


Simplified Installation, Setup and Integration

- Database configuration is now part of the Setup Wizard, which will update the configuration files based on the options you select.
- You can choose between a JNDI datasource (i.e. server-managed) or a simpler JDBC configuration.
- When upgrading, you can import an XML backup of your Crowd database or connect to an existing database via the Setup Wizard. This means that you don't have to go through the whole Setup Wizard, nor do a manual backup and restore of your Crowd database files.
- When integrating an application with Crowd, you'll notice that there's just one single JAR file to copy.

Configuration of Logging and Profiling via Console

- Enable and disable performance profiling.
- Configure your logging levels via the Crowd Administration Console, for quick and simple runtime troubleshooting.
- Edit the log configuration file for more advanced settings.
- Read the documentation.



Improved Performance and Efficiency

- You'll notice faster search results on the Administration Console screens, such as the Application Browser and User Browser.
- That annoying 'POSTDATA has expired' message no longer appears when you click the 'Back' button.
- Search results returned to a Crowd application are now sorted alphabetically — such as the list of groups shown in a Confluence group picker.
- We've fixed the Hibernate StaleStateException error that was causing occasional performance degradation and authentication failures.
- You can choose to store the login session tokens in the Crowd database (as done prior to Crowd 1.3) or in memory (new option as from Crowd 1.3). Depending upon your installation, in-memory storage could greatly improve response times during authentication. Read about configuring token storage.
- Gzip compression of Crowd Security Server output is now optional. You can turn it on or off via the Crowd Administration Console. Some reasons why you may want to turn Gzip compression off:
  - It may be easier to debug problems using uncompressed data.
  - Some agents, such as older versions of Internet Explorer, have problems with the Gzip format.

Highlights for the Developers

- The Java client library API has been upgraded. Read more about the API changes and the upgrade notes.
- You can pass the `crowd.properties` file to a client application as an environment variable.

Plus Over 60 Improvements and Bug-Fixes

Atlassian JIRA (68 issues)			
Key	Summary	Pr	Status
CWD-738	Allow configuring of request logs in the Crowd client libraries.		Resolved
CWD-654	Xalan is missing from the demo applications WEB-INF/lib folder.		Closed
CWD-768	Hibernate DAOs for Principals and Groups close the Hibernate Session when adding		Resolved
CWD-703	Crowd OpenID WAR file is missing commons-logging jar.		Resolved
CWD-639	Crowd hanging client applications, error with token manager		Resolved
CWD-466	Storing login tokens in an external DB is inefficient		Resolved
CWD-350	Tuckey rewrite filter dials home by doing a DNS lookup.		Resolved
CWD-897	Generic LDAP Directory type is displayed as OpenLDAP not Generic		Closed
CWD-882	Unalble to update the 'active' flag of an Application		Resolved
CWD-838	Updating any directory type in Crowd has		Resolved

	multiple validation problems.			
CWD-830	Change Crowd WAR deployment to zip archive.			Resolved
CWD-829	When updating a Delegated or Connector based directory, required fields are not marked as required.			Resolved
CWD-828	When updating an Internal Directory, there is no validation performed on the Configuration tab			Resolved
CWD-824	Session timeout during the installation should be larger than 5 minutes			Resolved
CWD-823	JDBC connection should default to MySQL			Resolved
CWD-822	crowd-init.properties value not set error message during startup is not useful			Resolved
CWD-817	Default results per page to 100			Resolved
CWD-806	Fix log4j.properties so dates are displayed in log files.			Resolved
CWD-805	Crowd's Add Directory Screen indicates we support Open Directory.			Resolved

CWD-800	When associating a Group/ Role to a Principal in the Demo application, an error is displayed			Resolved
CWD-790	Have you seen the client/lib directory lately? The current count is about 46 JAR files!			Resolved
CWD-775	Add Logging & Profiling functionality into Crowd Admin screen.			Resolved
CWD-767	Crowd's Client libraries should be slimmed down to a single JAR file containing all required classes for a Crowd Client			Resolved
CWD-765	File missing in 1.2.2 release			Resolved
CWD-758	Hibernate StaleStateExceptions in Crowd			Resolved
CWD-757	Crowd with delegated LDAP auth - update documentation for Bamboo-Crowd integration			Resolved
CWD-739	Concurrency Issue in client libraries may result in multiple caches			Resolved
CWD-731	OGNL Exception being thrown when updating a principal			Resolved

CWD-728	The Internal Directory is throwing a java.lang.IndexOutOfBoundsException: Index: 0, Size: 0 on requiresPasswordChange()		Resolved
CWD-727	Poor logging of a Token miss in the In-memory token cache.		Resolved
CWD-726	java.lang.IllegalStateException: Can't overwrite cause exception seen in Crowd		Resolved
CWD-724	Configuration classs for the LDAP importer		Resolved
CWD-723	LDAP Importer, to migrate data from one directory into another.		Resolved
CWD-720	Enable import from XML in the setup process		Closed
CWD-716	Error when attempting to remove a group		Resolved
CWD-711	The HTTPAuthenticator isAuthenticated method should initially check for a token		Resolved
CWD-700	The isMember call for groups can be slow for very large groups in an Internal Directory		Closed
CWD-699	Crowd SSO is incompatible with JIRA 3.12/ Confluence 2.7 trusted		Resolved

CWD-694	application feature. ehcache-1.2.3.jar is missing from client/ lib folder.			Resolved
CWD-688	Help links directly in the administration console			Closed
CWD-686	Sort groups, users and roles before returning results to the security server client			Resolved
CWD-685	Write System Info page to atlassian-crowd.log on Crowd startup			Resolved
CWD-675	remove "cache-control: no-store" on search results pages			Resolved
CWD-669	Adding group/ role with prefixed space causes Hibernate error			Resolved
CWD-666	Persistence system should use c3p0 so hibernate's default pooling system is not used.			Resolved
CWD-650	Update the crowd distribution module parent POM version to version 10			Resolved
CWD-649	Update the atlassian-crowd module parent POM version to version 7			Closed
CWD-646				Resolved

	Move FishEye connector outside crowd-core		
CWD-645	Use Spring dependency injection for SecurityServerClient and HttpAuthenticator in Crowd applications		Resolved
CWD-633	Allow the crowd.properties file to be passed to a Client application as an environment variable		Resolved
CWD-622	Make SecurityServerClient not static		Closed
CWD-586	start_crowd.sh and build.sh fail on Solaris		Resolved
CWD-584	Adding a Principal to Sun DSEE 6.2 throws a NullPointerException		Resolved
CWD-499	Creating Groups and Principals fails on 2000		Resolved
CWD-481	Support CRYPT encryption in OpenLDAP connector		Resolved
CWD-427	OpenLDAP Connector should default to SSHA encryption.		Resolved
CWD-389	GZip compression is optional through the administration console.		Resolved
CWD-208	Mixed authentication and authorization support for external		Resolved

	directory connectors.			
CWD-149	Config Test doesn't appear to obey Directory and Group rules			Resolved
CWD-855	OGNL exceptions are thrown when remoing Groups and Roles in the Demo app			Resolved
CWD-849	Rationalise the path to crowd-init.properties that's displayed on startup			Resolved
CWD-818	Admin Console: Selected tab CSS needs tweaking for Windows compatability			Resolved
CWD-799	When creating a Group/ Role to a Principal in the Demo application, an exception is thrown.			Resolved
CWD-798	When adding a Group or Role via the Demo app, the description field is not being persisted.			Resolved
CWD-706	Fix logging on startup for the OpenID Server. Stop the logging of Hibernate INFO.			Resolved
CWD-570	First Name not being displayed from Apache DS			Resolved
CWD-453				Resolved

CWD-847	Crowd core jar breaks in Grails, need a new slimmed-down client jar Error message is confusing when no directories are mapped to an application				Resolved

Known Issues in This Release

We have an enthusiastic and dedicated group of testers and customers who jump in there, try out the new Crowd release, and report any problems so that we can fix them quickly. Here's a list of known issues which will be fixed in our next point release.

A big thank you to everyone who helps us ensure that Crowd keeps getting better and better.

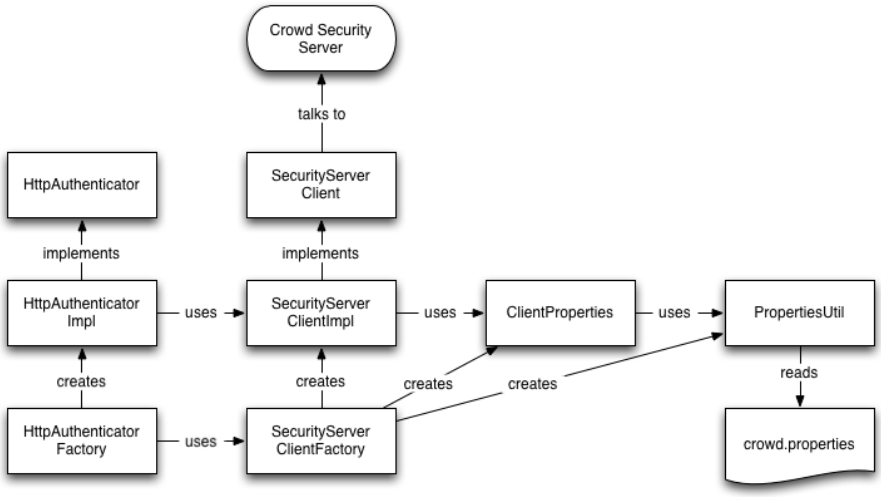
Client API Changes

This page last changed on Mar 03, 2008 by smaddox.

Crowd 1.3 brings a rework of the internals of the Crowd Client library — see CWD-622. This page gives a summary of the API changes.

Description of the changes

- The static implementations of HttpAuthenticator and SecurityServerClient have been removed. They have been replaced with instantiable objects.
- The GenericClient has been removed and its functions have been absorbed into the new SecurityServerClient and the ClientProperties objects.
- The relationships in the new class structure are represented below:



Why go to non-static?

- Makes it easier to unit test your applications. Simply mock out the SecurityServerClient or HttpAuthenticator interfaces to test business logic without being tied to the collaborators.
- Allows you to have multiple 'applications' in one classloader.

But I liked my static calls!

- SecurityServerClientFactory and HttpAuthenticatorFactory are provided to allow for a fast migration to the new API. The logical functionality of the client and authenticator are unchanged.
- So for example, instead of:

```
HttpAuthenticator.isAuthenticated(request);
```

you could use:

```
HttpAuthenticatorFactory.getHttpAuthenticator().isAuthenticated(request);
```

What are my options?

1. Use the supplied factory methods to manage singleton instances, OR
2. Externally manage singleton instances, e.g. via an IoC container like Spring.

Using the factories

The factories, `HttpAuthenticatorFactory` and `SecurityServerClientFactory`, provide quick access to implementations of the `HttpAuthenticator` and `SecurityServerClient`. They manage singleton

instances of the beans. This means that if you do opt to use the factories, then you should never instantiate `HttpAuthenticatorImpl` or `SecurityServerClientImpl` directly.

The factories naturally assume that there is one application client per classloader, i.e. one `SecurityServerClient` and one `HttpAuthenticator`.

Using an IoC container

Managing the singleton implementations externally may be a convenient approach for applications that use an IoC container. For example, Spring could be used to manage the instances of `SecurityServerClientImpl` and `HttpAuthenticatorImpl`. In Crowd, internally, we use this approach.

If you would like to use the standard Spring configuration, which loads the client properties from `crowd.properties`, simply add the `applicationContext-CrowdClient.xml` from the classpath to your Spring configuration:

```
<context-param>
  <param-name>contextConfigLocation</param-name>
  <param-value>
    classpath:/applicationContext-CrowdClient.xml
  </param-value>
</context-param>
```

This file is located in the `crowd-integration-client.jar`.

If you would like to customise your own configuration, modify the bean configuration to suit your needs:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN" "http://www.springframework.org/dtd/spring-
beans.dtd">

<beans>

    <bean id="propertyUtils" class="com.atlassian.crowd.util.PropertyUtils"/>

    <bean id="clientProperties"
 class="com.atlassian.crowd.integration.service.soap.client.ClientProperties">
        <constructor-arg ref="propertyUtils"/>
    </bean>

    <bean id="securityServerClient"
 class="com.atlassian.crowd.integration.service.soap.client.SecurityServerClientImpl">
        <constructor-arg ref="clientProperties"/>
    </bean>

    <bean id="httpAuthenticator"
 class="com.atlassian.crowd.integration.http.HttpAuthenticatorImpl">
        <constructor-arg ref="securityServerClient"/>
    </bean>

    <bean id="verifyTokenFilter" class="com.atlassian.crowd.integration.http.VerifyTokenFilter">
        <constructor-arg ref="httpAuthenticator"/>
    </bean>

    <bean id="crowdAuthenticationInterceptor"
 class="com.atlassian.crowd.integration.xwork.CrowdAuthenticationInterceptor">
        <constructor-arg ref="httpAuthenticator"/>
    </bean>

</beans>
```

Make sure that you do not use the factories (either directly or implicitly) when externally managing singletons.

If you would like to use the `VerifyTokenFilter`, you can use Spring to autowire the servlet filter by defining it in your `web.xml`:

```
<filter>
  <filter-name>verifyTokenFilter</filter-name>
  <filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>
</filter>

<filter-mapping>
  <filter-name>verifyTokenFilter</filter-name>
  <url-pattern>/secure/*</url-pattern>
</filter-mapping>
```

This will protect all resources matching the `/secure/*` pattern.

Known Issues in Crowd 1.3

This page last changed on Mar 02, 2008 by smaddox.

We have an enthusiastic and dedicated group of testers and customers who jump in there, try out the new Crowd release, and report any problems so that we can fix them quickly. Below is a list of known issues. We're working on them, and will have a point release out as soon as possible.

A big thank you to everyone who helps us ensure that Crowd keeps getting better and better.

While you're waiting, take a look at the great new features in Crowd 1.3.

You can also browse the Crowd project in our issue tracker to see what's fixed and what's not, for each release.

Issues to be Fixed in Crowd 1.3.1

Atlassian JIRA (13 issues)				
Key	Summary	Pr	Status	
CWD-899	When creating an LDAP based directory a password algorithm attribute is being set for all directory types regardless if they use one or not.			Resolved
CWD-924	SSO failure when authenicating two users in two tabs (in one browser)			Resolved
CWD-920	OpenLDAP MD5 encrypted password stored as plain text			Resolved
CWD-914	Viewing OpenLDAP Directoy Connector Info throws an exception			Resolved
CWD-900	Paged result size should not persist on directories that have not have "Use Paged Results" enabled.			Resolved
CWD-898	Crowd 1.3 UI is not compatible with IE 6			Resolved
CWD-875	User groups list in directory should sort alpha-numeric rather than natural.			Resolved
CWD-782	Textual changes on new directory importer screens			Resolved
CWD-527	IllegalDataException from active-directory authentication failure			Resolved
CWD-916	View Principal/User sessions in the Crowd console directory links broken			Resolved
CWD-909	User Name RDN Attribute field is not populated for Delegated Authentication directory screen			Resolved
CWD-561	Support the 'uid' and 'cn' attribute with the inetorgperson object at the same time			Resolved
CWD-439				

	Errors in the Confluence logs about Crowd (XFire prolog EOF)			Resolved

Crowd 1.4 Release Notes

This page last changed on May 08, 2008 by smaddox.

8 May 2008

The Atlassian Crowd team is proud to release Crowd 1.4.

Crowd 1.4 supports nested groups in LDAP directories. This means a group can now be a member of another group, making management of permissions much easier. For example, a Crowd-integrated Confluence or JIRA site will see users in sub-groups as members of the parent group.

The new Self-Service Console gives you the option to allow any authorised Crowd user to update their own user profile and password and to view their authorisation details.

There's a new directory connector for Novell eDirectory. Crowd also supports read-only connections to an LDAP directory using the Posix schema. This is useful if you have a Unix installation and want to integrate it with an LDAP directory.

For the development community, a new plugin framework supports customised event listeners and password encoders.

Highlights of this release:

Error formatting macro: toc: java.lang.NullPointerException

Responding to your feedback:

4 new feature requests implemented

90 votes satisfied

Keep logging your votes and issues. They help us decide what needs doing!

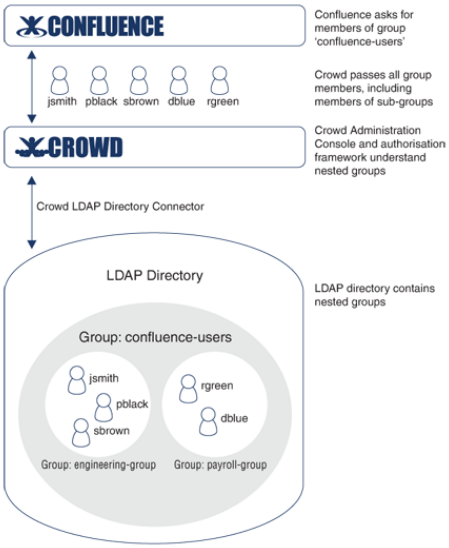
Upgrading to Crowd 1.4
You can download Crowd from the Atlassian website. If upgrading from a previous version, please read the Crowd 1.4 Upgrade Notes.

Highlights of Crowd 1.4

Nested Groups

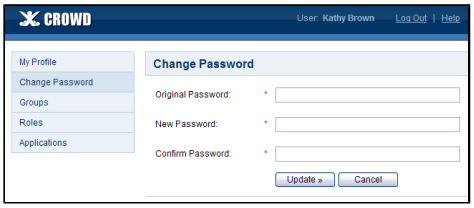
- In your LDAP directory, you can assign a group as a member of another group.
- In Crowd, you can map any group to an application, including a group which contains other groups. Currently, nested groups are supported for LDAP directory connectors only.
- For example, you might have two LDAP groups: 'engineering-group' and 'payroll-group'. Now you want to allow all members of those groups to access your Confluence wiki. You can create a group called 'confluence-users', mapped to the Confluence application, with members 'engineering-group', 'payroll-group' and any other groups and users. Crowd will allow members of those groups and sub-

groups to log in to Confluence. When Confluence requests a list of the users in the 'confluence-users' group, Crowd will present all users in the group plus all users in its sub-groups.
- Good news for our Confluence, JIRA and other Atlassian customers — this feature satisfies your requests for nested groups in those products too.
- Take a look at our documentation.



2 Self-Service Console

- Crowd users, including non-administrators, can log in to Crowd.
- Change or reset your own password.
- Update your user profile.
- View your group and role membership.
- See a list of the applications you can log in to.
- The new Crowd User Guide explains the ins and outs.



3 Novell eDirectory Connector

- Crowd 1.4 provides a built-in directory connector for Novell eDirectory.
- Take a look at our documentation.

Posix Support for LDAP Directories

- Crowd supports read-only connections to an LDAP directory using the Posix/NIS schema.
- Initially, our support is targeted at OpenLDAP directories.
- This is useful if you have a Unix installation and want to integrate with an LDAP directory.
- Here's our documentation on connecting your LDAP directory using the Posix/NIS schema.

Plugin Framework

- For our development community, the new plugin framework supports customised event listeners and password encoders.
- For example, you might decide to write your own event listener to audit failed Crowd authentication requests. Within Crowd itself, the reset password listener uses the new event framework.
- You can create your own plugin to use a specific password encryption algorithm that Crowd does not support out of the box. Crowd's own password encoders provide examples of such plugins.

More than 30 Improvements and Bug-Fixes

Atlassian JIRA (34 issues)				
Key	Summary	Pr		Status
CWD-1011	Atlassian Importer does not import passwords correctly			Closed
CWD-942	Problems when creating users from JIRA/ Confluence in internal Crowd directories			Resolved
CWD-614	Implement caching on Crowd client layer			Resolved
CWD-202	JIRA throws DataAccessException when an external user			Resolved

	(Crowd) is removed.			
CWD-74	Support groups-within-groups			Resolved
CWD-993	XML backup does not include delegated directory users and groups			Resolved
CWD-941	Allow client proxy and connection pool configuration in crowd.properties			Resolved
CWD-25	Plugins System			Resolved
CWD-1035	FATAL log messages produced when calling SecurityServerClientFactory.getSecurityServerClient			Resolved
CWD-1032	Fix Upgrade Task 114 password encryption attribution			Resolved
CWD-1031	Fix XML importer parser			Resolved
CWD-1016	Update common modules			Resolved
CWD-979	Change created AD group type to Distribution			Resolved
CWD-978	Update Spring LDAP to 1.2.1			Resolved
CWD-976	Update directory importer documentation to better explain what's allowed and what isn't			Resolved
CWD-968	Users deleted from JIRA are not removed from the client side cache in CrowdCredentialsProvider.			Closed
CWD-959				Resolved

	Creation of Principals from a client application (JIRA/ Confluence) will fail silently when there is multiple directories, one of those being an Internal Directory.			
CWD-952	Upgrade atlassian-user to be compatible with interface change for Confluence 2.8			Resolved
CWD-936	Provide the ability to choose an encryption type for a Generic Directory			Resolved
CWD-934	Online help links for new 1.4 features			Resolved
CWD-924	SSO failure when authenicating two users in two tabs (in one browser)			Resolved
CWD-920	OpenLDAP MD5 encrypted password stored as plain text			Resolved
CWD-903	Configure redirection of context-sensitive online help links for existing 1.3 release			Resolved
CWD-898	Crowd 1.3 UI is not compatible with IE 6			Resolved
CWD-870	CrowdCredentialsProvider exception handling improvements			Resolved

CWD-782	Textual changes on new directory importer screens			Resolved
CWD-684	Add Crowd Directory Information to the Crowd logs			Closed
CWD-680	Jive Forums 5.5.9 and above			Resolved
CWD-547	Support crowd scans all Person objects in AD when it doesn't need to.			Resolved
CWD-486	Document configuring Novell eDirectory as an LDAP Directory Connector			Resolved
CWD-485	Officially support integration with Novell eDirectory			Resolved
CWD-306	Allow users to manage their accounts and view thier details in a 'self service' console.			Resolved
CWD-676	Event listener exception during startup			Resolved
CWD-569	Unable to store group/role description			Resolved

Installing Crowd

This page last changed on May 05, 2008 by smaddox.

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

RELATED TOPICS

System Requirements

This page last changed on May 07, 2008 by smaddox.

Hardware Requirements

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.

Software Requirements

1. Sun JDK 1.4 (1.5 or higher is preferred). You can download the Java SE Development Kit (JDK) from the Sun website.
2. Note: 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.
3. J2EE 1.4 application server or a Servlet 2.3 web container. NOTE: Crowd ships with Apache Tomcat (5.5.x).
4. JDBC-compliant database that is supported by Hibernate. NOTE: Crowd ships with a built-in HSQL database, which is fine for evaulation purposes. For production environments we recommend configuring Crowd to use an external database.
5. If you are deploying a WAR installation, ensure that the JTA (Java Transaction API) jar is deployed in the shared `lib` folder on the application server. The JTA is available in a couple of places:
   - On the Sun website.
   - In the Crowd Standalone Distribution zip file, available on the Crowd download centre — file `jta-1.0.1B.jar` in `CROWD\apache-tomcat-5.5.20\common\lib`.

     The JTA specifies standard Java interfaces between a transaction manager and the parties involved in a distributed transaction system: the resource manager, the application server, and the transactional applications. Refer to the Sun documentation for more information.

Supported Databases

The following database servers are supported by Hibernate:

- HypersonicSQL
- PostgreSQL
- Microsoft SQL Server
- MySQL
- Oracle 10g (tested on 10.2.0.1)

Of these, the following databases have been tested and are supported by Atlassian:

- HSQLDB
- MS SQL Server
- MySQL
- Oracle
- PostgreSQL

Supported J2EE Servers

The following J2EE servers are supported:

- JBoss (4.2.2 GA)
- Resin (3.0.x) - tested on 3.0.23
- Tomcat (5.5.x) - tested on 5.5.20

Next Step

Installing Crowd and CrowdID

RELATED TOPICS

- System Requirements
  - Setting JAVA_HOME
- 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
  - Installing Crowd and CrowdID WAR Distribution
    - Installing Crowd WAR Distribution
      - Configuring Crowd & CrowdID on Tomcat 5.5.x
      - Installing Crowd WAR on JBoss
    - Installing CrowdID WAR Distribution
  - Specifying your Crowd Home Directory
- Running the Setup Wizard
  - Troubleshooting your Configuration on Setup
- Configuring Crowd
  - Important Directories and Files
    - 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 JAVA_HOME

This page last changed on May 05, 2008 by smaddox.

Once you have installed the JDK (see System Requirements), you need to set the JAVA_HOME environment variable.

To set the JAVA_HOME environment variable on Windows

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.

To set the JAVA_HOME environment variable on 'nix based systems

There are many ways you can do it on 'nix based systems (including Mac OS X). Here are two:

For your current user,

1. Open up a shell / terminal window
2. `vi ~/.profile` (replace vi with your favourite text editor)
3. Add `export JAVA_HOME=/path/to/java/home/dir` on its own line at the end of the file
4. Add `export 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 favourite text editor)
3. Add `export JAVA_HOME=/path/to/java/home/dir` on its own line at the end of the file
4. Add `export 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.

RELATED TOPICS

Installing Crowd and CrowdID

This page last changed on May 07, 2008 by smaddox.

The instructions below tell you how to install the standalone distribution of Crowd, which includes Apache Tomcat. If you wish to deploy a WAR distribution of Crowd or CrowdID on your own existing application server instead, read the instructions on the Crowd WAR distribution.

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.

On this page:

Error formatting macro: toc: java.lang.NullPointerException

1. Install Crowd (Standalone Distribution)

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. To specify the directory:

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

2. Optional Prepare your Database

Hint: 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

3. 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.

RELATED TOPICS

- System Requirements
- Installing Crowd and CrowdID
- Running the Setup Wizard
- Configuring Crowd
- Installing Crowd as a Windows Service

Connecting Crowd to a Database

This page last changed on May 07, 2008 by smaddox.

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 Standalone distribution of Crowd 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

RELATED TOPICS

- System Requirements
- Installing Crowd and CrowdID
- Running the Setup Wizard
- Configuring Crowd
- Installing Crowd as a Windows Service

HSQLDB

This page last changed on May 07, 2008 by smaddox.

The Standalone distribution of Crowd 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.

RELATED TOPICS

MS SQL Server

This page last changed on May 07, 2008 by smaddox.

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`.
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

2. Copy the SQL Server driver to your application server

1. Download the SQL Server JDBC driver from JTDS (recommended), 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 (JRA-5760, JRA-6872), workflow problems (JRA-8443) and Chinese character problems (JRA-5054).
2. Add the SQL Server JDBC driver jar (jtds-[version].jar) to the `common/lib` directory.

Next Steps

Complete the Crowd installation, then start Crowd and run the Setup Wizard as described in the Installation Guide.

RELATED TOPICS

- System Requirements
  - Setting JAVA_HOME
- Installing Crowd and CrowdID
  - Connecting Crowd to a Database
  - Connecting CrowdID to a Database
  - Installing Crowd and CrowdID WAR Distribution
  - Specifying your Crowd Home Directory
- Running the Setup Wizard
  - Troubleshooting your Configuration on Setup
- Configuring Crowd
  - Important Directories and Files
  - 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

MySQL

This page last changed on May 07, 2008 by smaddox.

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 (5.0.37 and later) 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. crowddb).
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 `common/lib/` directory. NOTE: 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).

Next Steps

Complete the Crowd installation, then start Crowd and run the Setup Wizard as described in the Installation Guide.

RELATED TOPICS

- System Requirements
  - Setting JAVA_HOME
- Installing Crowd and CrowdID
  - Connecting Crowd to a Database
  - Connecting CrowdID to a Database
  - Installing Crowd and CrowdID WAR Distribution
  - Specifying your Crowd Home Directory
- Running the Setup Wizard
  - Troubleshooting your Configuration on Setup
- Configuring Crowd
  - Important Directories and Files
  - 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

Oracle

This page last changed on May 07, 2008 by smaddox.

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 Oracle for Crowd.

1. Configure Oracle

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. crowddb).
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 `apache-tomcat-X.X.XX/common/lib` directory.

Next Steps

Complete the Crowd installation, then start Crowd and run the Setup Wizard as described in the Installation Guide.

RELATED TOPICS

- System Requirements
  - Setting JAVA_HOME
- Installing Crowd and CrowdID
  - Connecting Crowd to a Database
  - Connecting CrowdID to a Database
  - Installing Crowd and CrowdID WAR Distribution
  - Specifying your Crowd Home Directory
- Running the Setup Wizard
  - Troubleshooting your Configuration on Setup
- Configuring Crowd
  - Important Directories and Files
  - 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

PostgreSQL

This page last changed on May 07, 2008 by smaddox.

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 (e.g. crowduser).
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

2. Copy the PostgreSQL driver to your application server

1. Download the PostgreSQL JDBC driver from http://jdbc.postgresql.org/download.html. Get the JDBC 3 driver specific to your Postgres version, eg. `postgresql-8.x-xxx.jdbc3.jar`.
2. Add the PostgreSQL JDBC driver jar to the common/lib directory.

Next Steps

Complete the Crowd installation, then start Crowd and run the Setup Wizard as described in the Installation Guide.

RELATED TOPICS

- System Requirements
  - Setting JAVA_HOME
- Installing Crowd and CrowdID
  - Connecting Crowd to a Database
  - Connecting CrowdID to a Database
  - Installing Crowd and CrowdID WAR Distribution
  - Specifying your Crowd Home Directory
- Running the Setup Wizard
  - Troubleshooting your Configuration on Setup
- Configuring Crowd
  - Important Directories and Files
  - 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

Connecting CrowdID to a Database

This page last changed on May 07, 2008 by smaddox.

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

By default, CrowdID in the Crowd 'Standalone' distribution 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 'Standalone' distribution 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.

RELATED TOPICS

- System Requirements
  - Setting JAVA_HOME
- 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
  - Installing Crowd and CrowdID WAR Distribution
    - Installing Crowd WAR Distribution
      - Configuring Crowd & CrowdID on Tomcat 5.5.x
      - Installing Crowd WAR on JBoss
    - Installing CrowdID WAR Distribution
  - Specifying your Crowd Home Directory
- Running the Setup Wizard
  - Troubleshooting your Configuration on Setup
- Configuring Crowd
  - Important Directories and Files
    - 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

HSQLDB for CrowdID

This page last changed on May 07, 2008 by smaddox.

The default version of CrowdID uses an embedded HSQLDB database.

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

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 filesystem (beneath 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.

RELATED TOPICS

MS SQL Server for CrowdID

This page last changed on May 07, 2008 by smaddox.

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 (JRA-5760, [JRA-6872|http://jira.atlassian.com/browse/JRA-6872), workflow problems (JRA-8443) and Chinese character problems (JRA-5054).
2. Add the SQL Server JDBC driver jar (jtds-[version].jar) to the `common/lib` directory.

3. Configure your application server to connect to SQL Server

1. Edit the conf/Catalina/localhost/crowd.xml and customise the username, password, driverClassName and url parameters 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 Standalone distribution) and modify the hibernate.dialect to the following:

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

RELATED TOPICS

MySQL for CrowdID

This page last changed on May 07, 2008 by smaddox.

To connect CrowdID to MySQL,

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. crowddiddb). ⚠ 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 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 `common/lib/` directory. NOTE: 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 customise the username, password, driverClassName and url parameters 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/crowddiddb?
autoReconnect=true&amp;useUnicode=true&amp;characterEncoding=latin1"
              [ delete the minEvictableIdleTimeMillis, timeBetweenEvictionRunsMillis and
 maxActive params here ]
            />

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

</Context>
```

   The URL above assumes a LATIN-1 database - i.e. created with `create database crowddb character set latin1;`.

   ⚠ 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 Standalone distribution) and modify the hibernate.dialect to the following. Please choose only one of the 3 available options depending on how you have configured your database server.

```
*For MySQL set:*
hibernate.dialect=org.hibernate.dialect.MySQLDialect
*For MySQL with InnoDB set:*
hibernate.dialect=org.hibernate.dialect.MySQLInnoDBDialect
*For MySQL with MyISAM set:*
hibernate.dialect=org.hibernate.dialect.MySQLMyISAMDDialect
```

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

Oracle for CrowdID

This page last changed on May 07, 2008 by smaddox.

To connect CrowdID to Oracle,

1. Configure Oracle

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. crowddiddb). ⚠ 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 `common/lib` directory.

3. Configure your application server to connect to Oracle

1. Edit the file `apache-tomcat-X.X.XX/conf/Catalina/localhost/openidserver.xml` and customise the username, password, driverClassName and url parameters 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:crowddiddb"
              [ 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 standalone release) and modify the hibernate.dialect to the following

   ```
   hibernate.dialect=org.hibernate.dialect.OracleDialect
   ```

2. Then run `./build.sh` or `build.bat`. This will configure crowd 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.Oracle
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

- System Requirements
  - Setting JAVA_HOME
- 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
  - Installing Crowd and CrowdID WAR Distribution
    - Installing Crowd WAR Distribution
      - Configuring Crowd & CrowdID on Tomcat 5.5.x
      - Installing Crowd WAR on JBoss
    - Installing CrowdID WAR Distribution
  - Specifying your Crowd Home Directory
- Running the Setup Wizard
  - Troubleshooting your Configuration on Setup
- Configuring Crowd
  - Important Directories and Files
    - 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

PostgreSQL for CrowdID

This page last changed on May 07, 2008 by smaddox.

To connect CrowdID to PostgreSQL,

1. Configure PostgreSQL

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. crowddb). 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 PostgreSQL driver to your application server

1. Download the PostgreSQL JDBC driver from http://jdbc.postgresql.org/download.html. Get the JDBC 3 driver specific to your Postgres version, eg. + postgresql-8.x-xxx.jdbc3.jar.
2. Add the PostgreSQL JDBC driver jar to the common/lib directory.

3. Configure your application server to connect to PostgreSQL

1. Edit the file `apache-tomcat-X.X.XX/conf/Catalina/localhost/openidserver.xml` and customise the username, password, driverClassName and url parameters 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/crowddiddb" [ 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 (which are only needed for HSQL, and degrade performance otherwise).

4. Configure CrowdID to use PostgreSQL

1. Edit the `build.properties` file located in the root of the standalone release and modify the hibernate.dialect to the following

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

2. Then run `./build.sh` or `build.bat`. This will configure crowd to use the PostgreSQL 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.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. Now start up CrowdID and watch the logs for any errors.

RELATED TOPICS

- System Requirements
  - Setting JAVA_HOME
- 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
  - Installing Crowd and CrowdID WAR Distribution
    - Installing Crowd WAR Distribution
      - Configuring Crowd & CrowdID on Tomcat 5.5.x
      - Installing Crowd WAR on JBoss
    - Installing CrowdID WAR Distribution
  - Specifying your Crowd Home Directory
- Running the Setup Wizard
  - Troubleshooting your Configuration on Setup
- Configuring Crowd
  - Important Directories and Files
    - 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

Installing Crowd and CrowdID WAR Distribution

This page last changed on May 07, 2008 by smaddox.

The Crowd and CrowdID WAR distributions are intended for deployment onto an existing J2EE application server. It is assumed that you already know how to deploy a web application onto your chosen application server. If not, please contact your system administrator to assist you, or consider installing the Crowd Standalone distribution instead.

The standard Crowd installation guide tells you how to install the Standalone distribution of Crowd, which includes Apache Tomcat. Instead, you may wish to deploy Crowd or CrowdID onto your own existing application server. For this purpose, we provide WAR (Webapp ARchive) distributions of the Crowd and CrowdID server applications.

Crowd supports all the application servers listed in System Requirements.

The procedures for connecting Crowd and CrowdID are slightly different. The Crowd setup process provides the option of JDBC or JNDI datasource connections via the Crowd Setup Wizard. CrowdID requires a JNDI datasource configuration. Detailed instructions are on the following pages:

- Installing Crowd WAR Distribution
- Installing CrowdID WAR Distribution

RELATED TOPICS

- System Requirements
- Installing Crowd and CrowdID
- Running the Setup Wizard
- Configuring Crowd
- Installing Crowd as a Windows Service

Installing Crowd WAR Distribution

This page last changed on May 07, 2008 by smaddox.

The Crowd and CrowdID WAR distributions are intended for deployment onto an existing J2EE application server. It is assumed that you already know how to deploy a web application onto your chosen application server. If not, please contact your system administrator to assist you, or consider installing the Crowd Standalone distribution instead.

The standard Crowd installation guide tells you how to install the Standalone distribution of Crowd, which includes Apache Tomcat. Instead, you may wish to deploy Crowd or CrowdID onto your own existing application server. For this purpose, we provide WAR (Webapp ARchive) distributions of the Crowd and CrowdID server applications.

Crowd supports all the application servers listed in System Requirements.

Below is a generic overview of the steps required to install the Crowd WAR distribution. You will need to perform specific configuration steps depending upon your application server. As well as the generic instructions below, we also provide specific instructions on the following pages:

- Configuring Crowd & CrowdID on Tomcat 5.5.x
- Installing Crowd WAR on JBoss

Dependencies

Refer to the system requirements.

Please make sure that all dependencies are installed, otherwise Crowd will not run properly.

Overview of the Crowd WAR Installation Steps

1. Download the Crowd WAR distribution from the Crowd download centre.

   You will find the WAR archives for the Crowd and the CrowdID applications by clicking the Show advanced downloads link. You will need to deploy each application separately. For the rest of these instructions, we assume you are deploying Crowd WAR.
2. Please check your unzip program before extracting the downloaded archive, as some unzip programs can cause errors — see the note on the Crowd installation front page.
3. Unzip the download archive into a directory of your choice. We'll call it `CROWD` in the rest of these instructions.
4. Specify your Crowd Home directory by editing the configuration file at `server/default/deploy/crowd.war/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. To specify the directory:

   - Open the `crowd-init.properties` file.
   - 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
       ```
   - Save the `crowd-init.properties` file.
5. Create a database in your chosen database server.

6. Copy the JDBC driver into your application server's classpath.
7. Modify file `CROWD/WEB-INF/classes/crowd.properties` to point to the port of your application server. `8080` is the default, and is shown in the example below:

```
crowd.server.url=http://localhost:8080/crowd/services/
application.login.url=http://localhost:8080/crowd/console/
```

8. Depending upon your application server, you may need to zip up the WAR file again before deploying it. Place the `CROWD` directory or the WAR file into your application server's deployment directory. Please consult the server-specific documentation on how to do this. A few Atlassian best practice guides are listed here:
    - Configuring Crowd & CrowdID on Tomcat 5.5.x
    - Installing Crowd WAR on JBoss
9. Restart your application server.
10. Point a web browser at the IP address and port that your application server is running on (typically `http://localhost:8080`). The Crowd Setup Wizard will start.

RELATED TOPICS

- System Requirements
- Installing Crowd and CrowdID
- Running the Setup Wizard
- Configuring Crowd
- Installing Crowd as a Windows Service

Configuring Crowd & CrowdID on Tomcat 5.5.x

This page last changed on Feb 14, 2008 by donna@atlassian.com.

Configuring a Crowd 'context' in Tomcat:

1. Create a file called crowd.xml that contains the following context:

```
<Context path="/crowd" docBase="/path/to/atlassian-crowd-1.3-war-directory"
 reloadable="false"/>
```

2. Place it in your Tomcat's conf/Catalina/localhost/ directory.

Modify the `/path/to/atlassian-crowd-1.3-war-directory` to reflect the actual path to your Crowd WAR distribution. To avoid problems with your deployment, this should NOT be Tomcat's webapps directory.

If you are installing in Windows, make sure that the paths you specify for the location of the WAR file and database are full paths with drive letters.

Configuring a CrowdID 'context' in Tomcat:

If you are deploying CrowdID, you will need to specify a JNDI datasource.

1. Create a file called openidserver.xml that resembles the following example for a MySQL database:

```
<Context path="/openidserver" docBase="/path/to/atlassian-crowd-openid-1.3-war-directory"
 reloadable="false">
  <Resource name="jdbc/CrowdIDDS"" auth="Container" type="javax.sql.DataSource"
            username="crowdiduser"
            password="mypassword"
            driverClassName="com.mysql.jdbc.Driver"
            url="jdbc:mysql://localhost/crowdopenidserverdb?autoReconnect=true"/>
</Context>
```

1. Place it in your Tomcat's conf/Catalina/localhost/ directory.

Modify the `/path/to/atlassian-crowd-openid-1.3-war-directory` to reflect the actual path to your CrowdID WAR distribution. To avoid problems with your deployment, this should NOT be Tomcat's webapps directory. Modify appropriately for your database (e.g. Oracle, Postgres, etc.)

Please remember to update the driveClassName and copy the JDBC driver jar to your Tomcat's common/lib/ directory.

Installing Crowd WAR on JBoss

This page last changed on May 07, 2008 by smaddox.

The standard Crowd installation guide tells you how to install the Standalone distribution of Crowd, which includes Apache Tomcat. You may wish to deploy Crowd on your own existing application server instead. For this purpose, we provide WAR (Webapp ARchive) distributions of the Crowd and CrowdID server applications.

This page shows one example - use it as a basis for other installations

- This page tells you how to deploy Crowd onto a JBoss Application Server. For other application servers, refer to the generic WAR setup guide.
- On this page, we have used PostgreSQL as an example of a database connected via a JNDI datasource. Crowd supports all the databases listed in the System Requirements. Refer to Connecting Crowd to a Database for instructions on connecting Crowd to your enterprise database.

Follow the steps below to install Crowd on JBoss 4.2.2 GA using a PostgreSQL database:

1. Download the WAR distribution from the Crowd download centre.

   You will find the WAR archives for the Crowd and the CrowdID applications. You will need to deploy each application separately. For the rest of these instructions, we assume you are deploying Crowd WAR.
2. Please check your unzip program before extracting the downloaded archive, as some unzip programs can cause errors – see the note on the Crowd installation front page.
3. Unzip the download archive into a directory of your choice. We'll call it `server/default/deploy/crowd.war` in the rest of these instructions.
4. Specify your Crowd Home directory by editing the configuration file at `server/default/deploy/crowd.war/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. To specify the directory:

   - Open the `crowd-init.properties` file.
   - 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
       ```
   - Save the `crowd-init.properties` file.
5. Add file `server/default/deploy/crowd.war/WEB-INF/jboss-web.xml`, with the following contents:

```
<jboss-web>
  <resource-ref>
    <res-ref-name>jdbc/CrowdDS</res-ref-name>
    <res-type>javax.sql.DataSource</res-type>
    <jndi-name>java:CrowdDS</jndi-name>
  </resource-ref>
</jboss-web>
```

6. Create database `crowd_db` in PostgreSQL.

7. Add a datasource definition file `server/default/deploy/postgres-ds.xml`:

```
<datasources>
  <local-tx-datasource>
    <jndi-name>CrowdDS</jndi-name>
    <connection-url>jdbc:postgresql://localhost:5432/crowd_db</connection-url>
    <driver-class>org.postgresql.Driver</driver-class>
    <user-name>postgres</user-name>
    <password>postgres</password>
  </local-tx-datasource>
</datasources>
```

8. Modify file `server/default/deploy/crowd.war/WEB-INF/classes/crowd.properties` to point to the port of the JBoss server. `8080` is the default port number, and is shown in the example below:

```
crowd.server.url=http://localhost:8080/crowd/services/
application.login.url=http://localhost:8080/crowd/console/
```

9. Start JBoss with `run.sh` (Unix-based systems) or `run.bat` (Windows).
10. Point a web browser at http://localhost:8080/ where you will see the Crowd Setup Wizard.

Related Topics

- System Requirements
- Installing Crowd and CrowdID
- Running the Setup Wizard
- Configuring Crowd
- Installing Crowd as a Windows Service

Installing CrowdID WAR Distribution

This page last changed on May 07, 2008 by smaddox.


The Crowd and CrowdID WAR distributions are intended for deployment onto an existing J2EE application server. It is assumed that you already know how to deploy a web application onto your chosen application server. If not, please contact your system administrator to assist you, or consider installing the Crowd Standalone distribution instead.

The standard Crowd installation guide tells you how to install the Standalone distribution of Crowd, which includes Apache Tomcat. Instead, you may wish to deploy Crowd or CrowdID onto your own existing application server. For this purpose, we provide WAR (Webapp ARchive) distributions of the Crowd and CrowdID server applications.

Crowd supports all the application servers listed in System Requirements.

Below is a generic overview of the steps required to install the CrowdID WAR distribution. You will need to perform specific configuration steps, depending upon your application server. As well as the generic instructions below, we also provide server-specific instructions on the following pages:

- Configuring Crowd & CrowdID on Tomcat 5.5.x

Dependencies

Refer to the system requirements.


Please make sure that all dependencies are installed, otherwise Crowd will not run properly.

Overview of the CrowdID WAR Installation Steps

1. Download the CrowdID WAR distribution from the Crowd download centre.

   
   You will find the WAR archives for the Crowd and the CrowdID applications. You will need to deploy each application separately. For the rest of these instructions, we assume you are deploying CrowdID WAR.
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. We'll call it `CROWDID` in the rest of these instructions.
4. Create a database in your chosen database server and add the required datasource definition file to your application server.
5. Modify file `CROWDID/WEB-INF/classes/jdbc.properties` to use your chosen Hibernate database dialect, as explained in the previous step.
6. Modify file `CROWDID/WEB-INF/classes/crowd.properties` to point to the port of your application server. `8080` is the default, and is shown in the example below:

   ```
   crowd.server.url=http://localhost:8080/crowd/services/
   application.login.url=http://localhost:8080/crowd/console/
   ```
7. Depending upon your application server, you may need to zip up the WAR file again before deploying it. Place the `CROWDID` directory or the WAR file into your application server's deployment directory. Please consult the server-specific documentation on how to do this.
8. Restart your application server.
9. Point a web browser at the IP address and port that your application server is running on (typically `http://localhost:8080`). The Crowd Setup Wizard will start.

RELATED TOPICS

- System Requirements
- Installing Crowd and CrowdID
- Running the Setup Wizard
- Configuring Crowd
- Installing Crowd as a Windows Service

Specifying your Crowd Home Directory

This page last changed on Feb 20, 2008 by smaddox.

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. To specify the directory:

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

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

Running the Setup Wizard

This page last changed on May 07, 2008 by smaddox.

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:

Error formatting macro: toc: java.lang.NullPointerException

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.

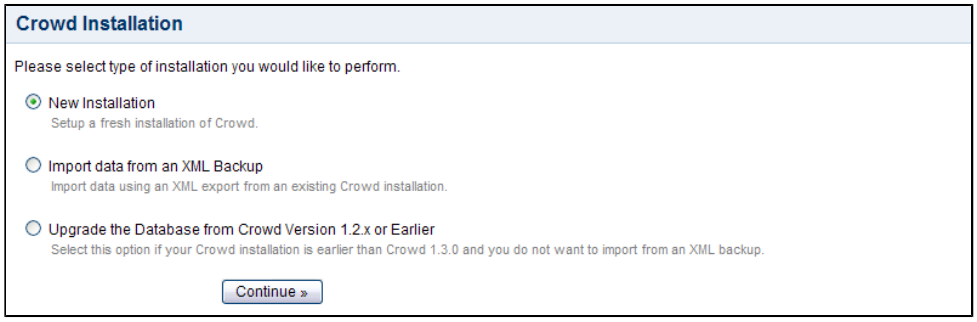
Step 2. Licensing

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



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.
- 'Upgrade the Database from Crowd Version 1.2.x or Earlier' — Select this option if your Crowd installation is earlier than Crowd 1.3.0 and you don't want to import your data from an XML backup. This option is provided for backwards-compatibility with earlier versions of Crowd. It allows you to upgrade a pre-1.3.0 Crowd database.

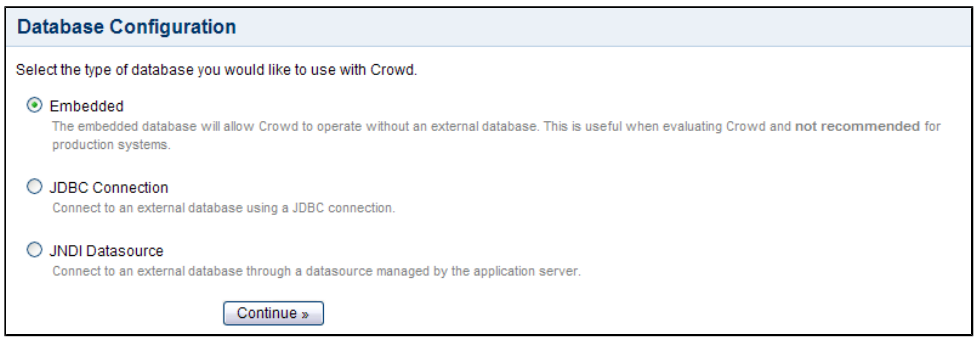
  If your current Crowd installation is for Crowd 1.3.0 or later, you cannot use this option. Instead, please follow the upgrade instructions.

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.

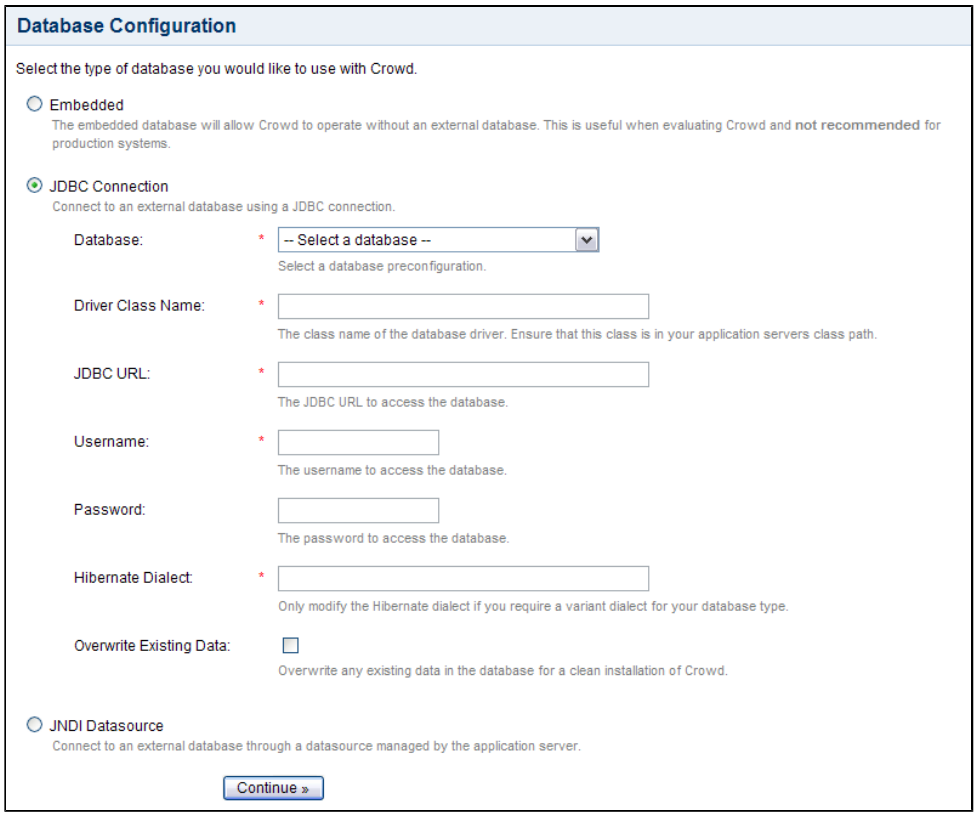
Option 1: Embedded HSQLDB Database (For Evaluation Purposes Only)



Crowd 'Standalone' 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.

Option 2: JDBC Connection

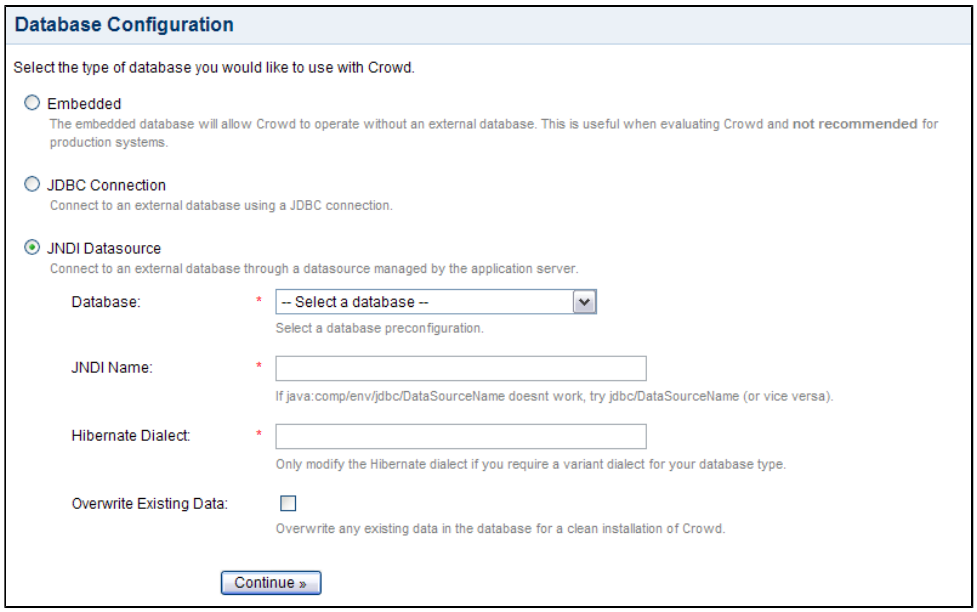


Select the 'JDBC Connection' if you want to connect to an external database via a JDBC connection. (If you have not yet created your database for Crowd, follow the database setup instructions.)

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.
Username	Enter the username which Crowd will use to access the database.
Password	Enter the password corresponding to the above username.
Hibernate Dialect	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.
Overwrite Existing Data	Crowd will ask you to confirm that existing data should be overwritten, if both of the following are true: • 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.

Option 3: JNDI Datasource

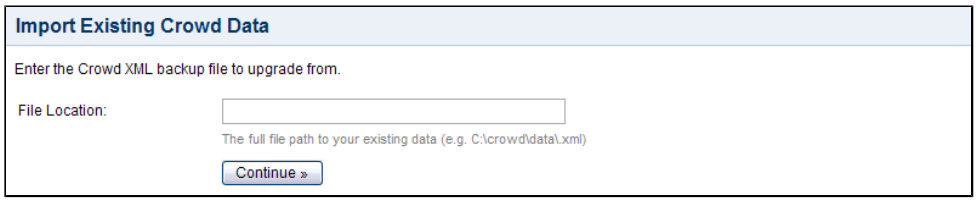


Select the 'JNDI Datasource' if you want to connect to an external database via a datasource managed by your application server.

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	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.
Overwrite Existing Data	Crowd will prompt you to confirm that existing data should be overwritten, if both of the following are true: • 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



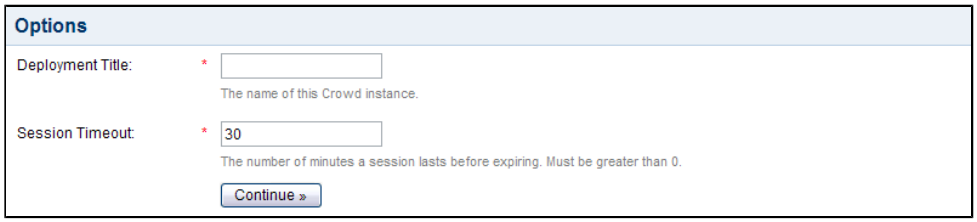
This screen will appear only if you selected 'Import data from an XML Backup' in Step 3 above.

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.

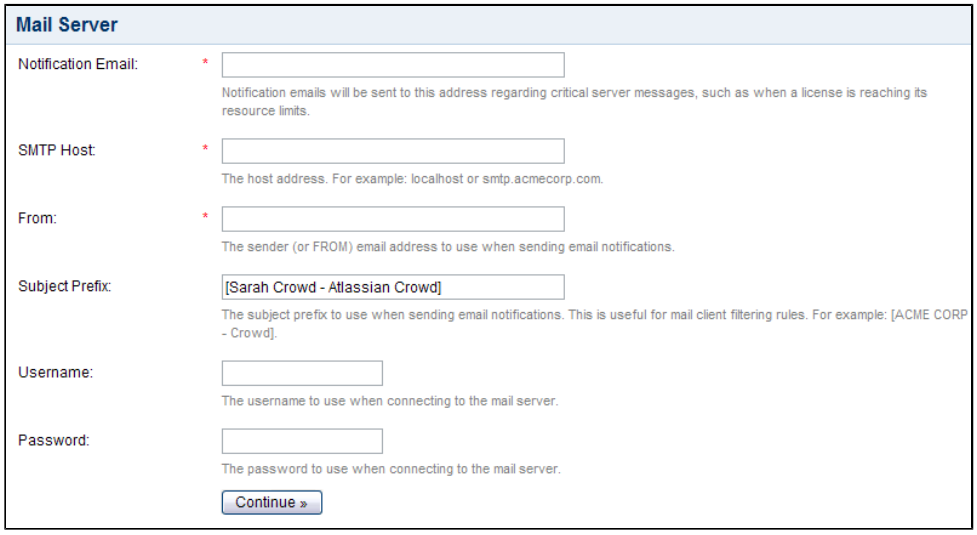
Step 6. Options



This part of the setup process allows you to specify general options for the Crowd server. You can change these values later, via the Crowd Administration Console.

- The deployment title specifies a unique name for your Crowd instance. The deployment title can be used when sending email notifications.
- 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`.

Step 7. Mail Server

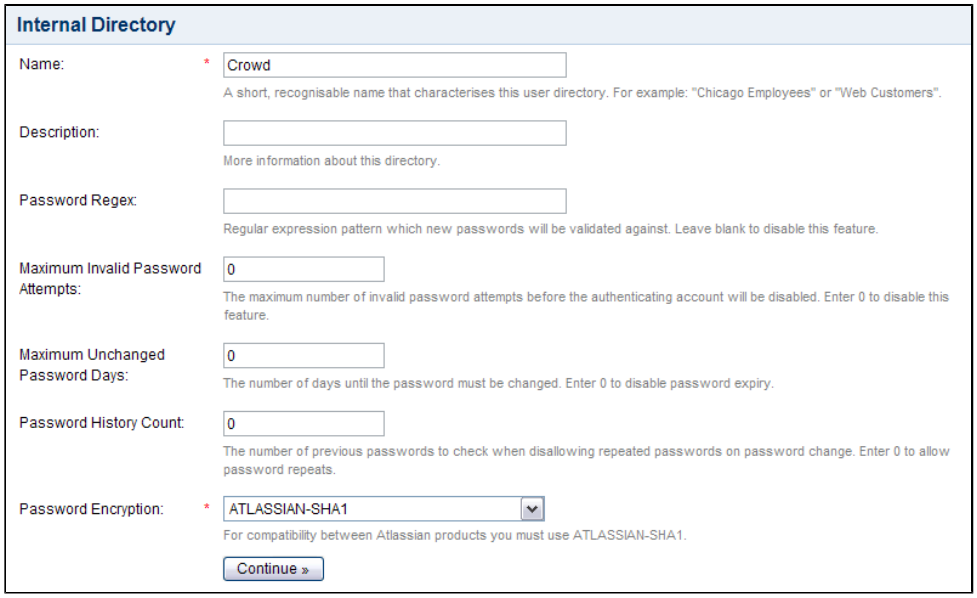


Crowd can send email notifications to users for specific events, such as when a password is reset.

Enter the details of your mail server, and the username and password (if required) that Crowd will use to log in to your mail server, then click the 'Update' button:

- Notification Email — The email address which will receive notifications about server events.
- SMTP Host — The hostname of the SMTP mail server, e.g. 'localhost' or 'smtp.acme.com'.
- From — The email address from which password notifications will be 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.
- 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.

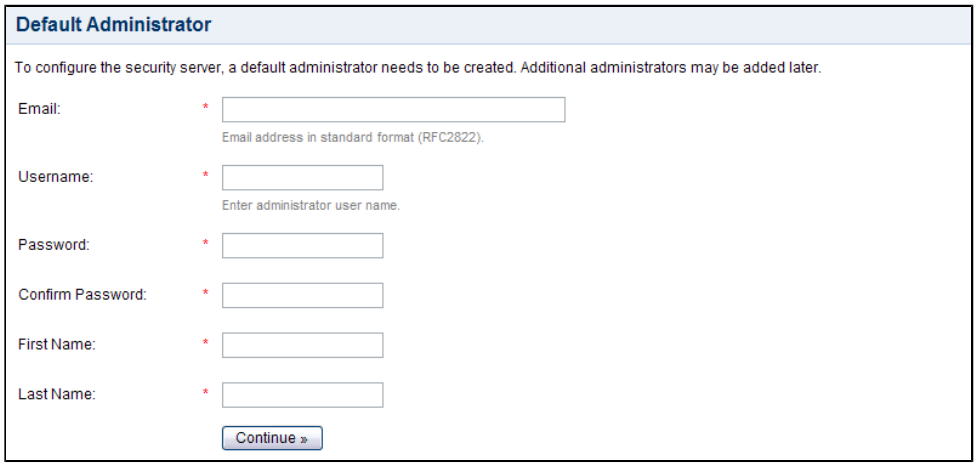
Step 8. Default Directory



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.

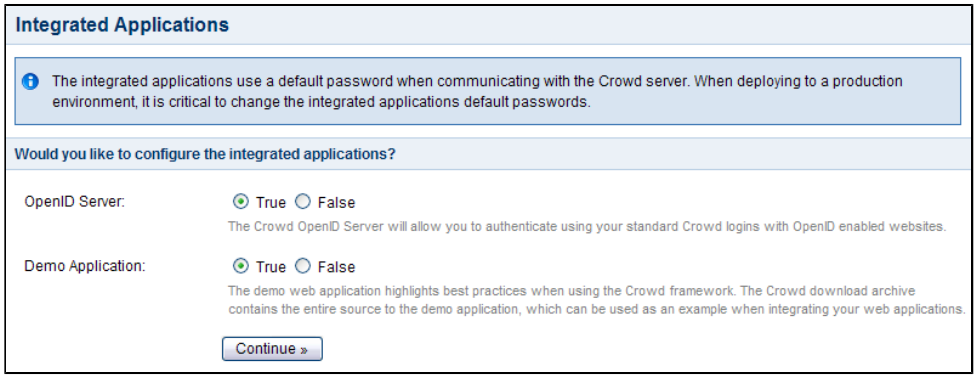
Crowd administrators group is in default directory
The default group `crowd-administrators` will be automatically created in the default directory. Members of this group have rights to administer Crowd.

Step 9. Default Administrator



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.

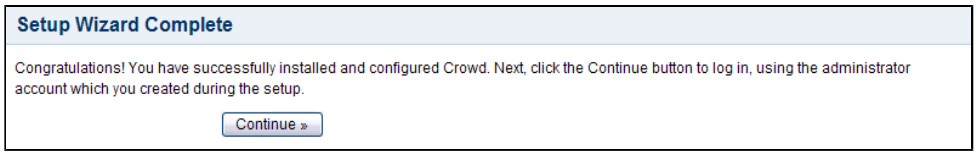
Step 10. Integrated Applications



You have the option to auto-configure two applications. We recommend that you select 'True' for both:

- 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 11. Setup Complete



You are now ready to use the Crowd Administration Console. For details, please see the Crowd Administration Guide.

RELATED TOPICS

- System Requirements
- Installing Crowd and CrowdID
- Running the Setup Wizard
- Configuring Crowd
- Installing Crowd as a Windows Service

Troubleshooting your Configuration on Setup

This page last changed on May 07, 2008 by smaddox.

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.

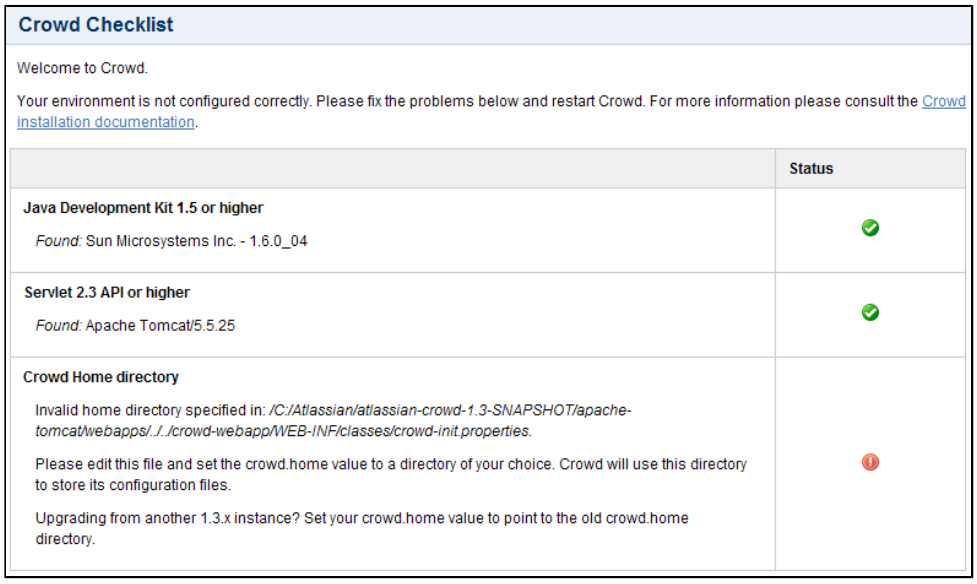
The 'Crowd Checklist' appears only if there is an error in your environment configuration, preventing you from completing the Setup Wizard.

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	(The screen will show the version of JDK detected in your system, with a red exclamation mark in the 'Status' column if insufficient.)	Refer to the System Requirements page for information about the JDK required and where you can get it.
Servlet 2.3 API or higher	(The screen will show the application server and version detected in your system, with a red exclamation mark in the 'Status' column if insufficient.)	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'



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

RELATED TOPICS

- System Requirements

- Installing Crowd and CrowdID
- Running the Setup Wizard
- Configuring Crowd
- Installing Crowd as a Windows Service

Configuring Crowd

This page last changed on May 07, 2008 by smaddox.

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

RELATED TOPICS

- Specifying your Crowd Home Directory
- Configuring an SSL Certificate for Microsoft Active Directory
- Troubleshooting your Configuration on Setup

- System Requirements
- Installing Crowd and CrowdID
- Running the Setup Wizard
- Configuring Crowd
- Installing Crowd as a Windows Service

Important Directories and Files

This page last changed on May 07, 2008 by smaddox.

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

On this page:

Error formatting macro: toc: java.lang.NullPointerException

When configuring an application to work with Crowd, you will be interested in the crowd.properties file.

The 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.

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

Error formatting macro: toc-zone: java.lang.NullPointerException

The **`crowd.cfg.xml`** File

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

- 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.

The file is located in the Crowd Home directory, described above.

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>initial</setupType>
  <buildNumber>212</buildNumber>
  <properties>
    <property name="crowd.server.id">AWWP-AWWP-AWWP-AWWP</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:Crowd/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.hash">MnnmpMnnmpMnnmpMnnmpMnnmpMnnmpMnnmpMnnmpMnnmpMnnmpMnnmpMnnmpMnnmpMnnmpMnnmpMnnmpMnnmpMnnmpMnnmpMnnmpMnnmp
property>
    <property
 name="license.message">nQQuwnpnROOOrQMrppvxRqROoprPnQomnTtxSsuXstWwSxpOrrMsmvUUnnuprorommmmmUUnnuprorommmmmUU1
property>
  </properties>
</application-configuration>
```

The 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:

Error formatting macro: toc-zone: java.lang.NullPointerException

The **`crowd-init.properties`** File

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

The 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 customised:

```
## 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
```

The **`crowd.properties`** File

The `crowd.properties` file contains application configuration settings. Each application integrated with Crowd will have a corresponding `crowd.properties` file. See Adding an Application.

The Crowd Administration Console application also has its own `crowd.properties` file, which is located in the Crowd Installation directory at `{CROWD_INSTALL}\crowd-webapp\WEB-INF\classes`

For more information about the settings, refer to The crowd.properties File. See also Passing the crowd.properties File as an Environment Variable.

The **`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: • `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

The **`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.sh`) 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
```

RELATED TOPICS

- System Requirements
- Installing Crowd and CrowdID
- Running the Setup Wizard
- Configuring Crowd
- Installing Crowd as a Windows Service

The crowd.properties File

This page last changed on May 07, 2008 by smaddox.

The attributes of the `crowd.properties` file are as follows:

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	The URL to which to redirect the user should their authentication token expire or be invalid due to security restrictions.
crowd.server.url	The URL to use when connecting with the integration libraries to communicate with the Crowd server.
session.isauthenticated	The session key to use when storing a `Boolean` value indicating whether the user is authenticated or not.
session.tokenkey	The session key to use when storing a `String` value of the user's authentication token.
session.validationinterval	The session key to use when storing an `Integer` value of the number of minutes between authentication validation. If this value is set to `0`, each HTTP request will be authenticated.
session.lastvalidation	The session key to use when storing a `Date` value of the user's last authentication.

RELATED TOPICS

Passing the crowd.properties File as an Environment Variable
Important Directories and Files
Adding an Application

Changing the Port that Crowd uses

This page last changed on May 07, 2008 by smaddox.

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. Edit the `build.properties` file, as described in Important Directories and Files.
2. Change the `crowd.url` property to the new port on which the Crowd Administration Console will be accessed.
3. Change the `demo.url` property to the new port on which the Crowd 'demo' application will be accessed.
4. Change the `openidserver.url` property to the new port on which the CrowdID Server will be accessed.
5. Run the `build.xml` script, as described in Important Directories and Files.

RELATED TOPICS

Configuring Crowd to Work with SSL

This page last changed on May 07, 2008 by smaddox.

When web applications are accessed across the internet, there is always the possibility of usernames and passwords being intercepted by intermediaries. These intercepts may occur when the data is travelling between a client and the server. It is often a good idea to enable access via HTTPS (HTTP over SSL) and require the use of HTTPS for pages where passwords are sent.

In some cases where transmitted data is sensitive, all pages should be accessed via HTTPS.

Note: Using HTTPS may result in slower performance.

What is SSL?

The Secure Sockets Layer (SSL) is a commonly-used protocol for managing the security of message transmission on the internet. SSL is included as part of most web browsers and web server products. For more information, take a look at Sun's Introduction to SSL.

On this page:

Error formatting macro: toc: java.lang.NullPointerException

Using Crowd over SSL

The process of enabling SSL 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 SSL Access

Edit `CROWD/apache-tomcat/conf/server.xml`, and at the bottom before the `</Service>` tag, add this section (or uncomment it if it's already there):

```
<Connector port="8443" maxHttpHeaderSize="8192" maxThreads="150"
minSpareThreads="25" maxSpareThreads="75" enableLookups="false"
disableUploadTimeout="true" acceptCount="100" scheme="https" secure="true"
clientAuth="false" sslProtocol="TLS" />
```

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).

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)
```

The keytool utility will prompt you for two passwords: the keystore password and the key password for Tomcat. You must use the same value for both passwords, and the value must be either:

1. 'changeit' (this is the default value Tomcat expects), or
2. if you use a value other than 'changeit', you must also specify this value in `conf/server.xml`. You must add the following attribute to the Connector tag described above:

```
keystorePass="<password value>"
```

For information on adding a key pair issued by a Certificate Authority (CA), refer to the section entitled 'Installing a Certificate from a Certificate Authority' in the Apache Tomcat documentation.

IE7 on Vista Issue
If your clients will access Crowd from Internet Explorer 7 on Vista, please ensure that you specify the -keyalg RSA flag. By default the SHA1 algorithm is used, which results in error 'Internet Explorer cannot display the webpage'.
Apparently on JDK 1.6 you also need to specify the -sigalg MD5withRSA flag since -keyalg RSA will still result in SHA1 being used. If you like, you can refer to this Atlassian developer blog post for more information.

Importing a CA-Issued Certificate

When using certificates issued by a Certificate Authority, you also need 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 Sun documentation for more information on the keytool:

- Solaris and Linux
- Windows

Try this blog post for a handy tutorial:

- Talkingtree blog post

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.

Step 3: Modify crowd.properties

Modify your `crowd-webapp/WEB-INF/classes/crowd.properties` file to reflect your new SSL settings. For example:

```
#Wed Apr 09 12:36:21 EST 2008
session.lastvalidation=session.lastvalidation
session.isauthenticated=session.isauthenticated
application.password=password
application.name=crowd
session.validationinterval=0
crowd.server.url=https\://localhost:8443/crowd/services/
session.tokenkey=session.tokenkey
application.login.url=https\://localhost:8443/crowd/console/
```

Step 4: Create or Modify setenv.sh or setenv.bat

In order to ensure that XFire calls work over SSL you will need to pass keystore values to the JVM. To do this either edit or create a `setenv.sh` or `setenv.bat` file located in Tomcat's bin directory: `apache-tomcat/bin/setenv.sh or setenv.bat`

The contents of the file should look similar to this:

```
JAVA_OPTS="-Xms128m -Xmx256m $JAVA_OPTS -Djavax.net.ssl.keyStore=/<pathtokeystore>/.keystore -
Djavax.net.ssl.keyStorePassword=changeit -Djavax.net.ssl.trustStore=/<pathtokeystore>/.keystore -
Djavax.net.ssl.trustStorePassword=changeit"
```

Replace `<pathtokeystore>` with the path to your .keystore file and the password with your keystore's password if modified.

Now 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/idaniel/.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

This page last changed on May 07, 2008 by smaddox.

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.

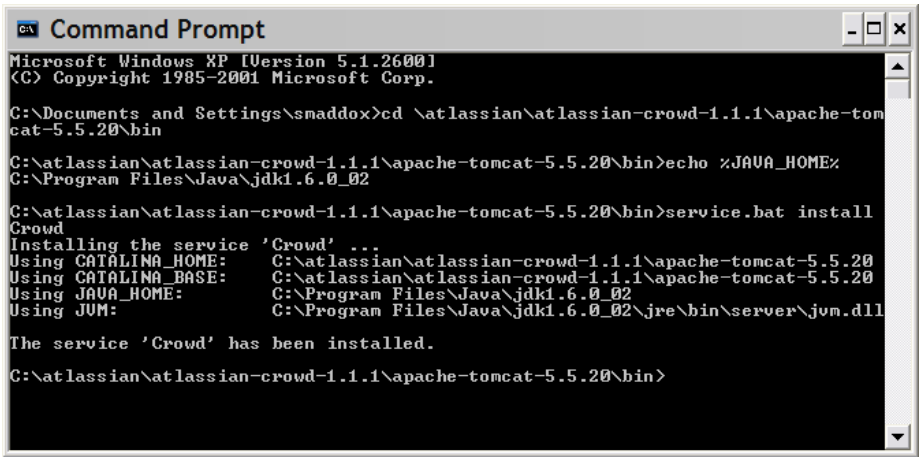
Running Crowd as a Windows service has other advantages. When Crowd is started manually, a console window opens - there is a risk that someone may accidentally shut down Crowd by closing the window. Also, the Crowd logs are properly managed by the Windows service (reliably found in `\atlassian-crowd.log`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

Installing Crowd as a Windows Service

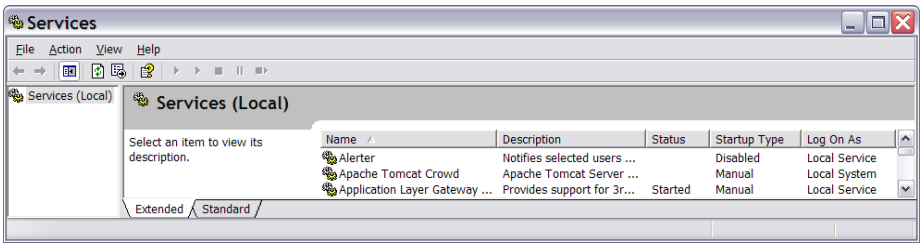
1. Open a DOS prompt.
2. 'cd' to your Crowd directory, and then the Tomcat `bin` subdirectory, e.g. `{CROWD_INSTALL}\apache-tomcat-5.5.20\bin`
3. If a directory in the path has spaces (e.g. `C:\Program Files\..`), please convert it to its eight-character equivalent (e.g. `c:\Progra~1\..`).
4. Ensure 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



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



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

```
tomcat5 //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 `regedt32.exe`. There should be an entry at `HKEY_LOCAL_MACHINE -> SOFTWARE -> Apache Software Foundation -> Procrun 2.0 -> Crowd`.

Additional Crowd Setup Options (Optional)

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

```
tomcat5 //US//Crowd --JvmMx 512
```

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

```
tomcat5 //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

    ```
    tomcat5 //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:

    ```
    tomcat5 //US//Crowd ++JvmOptions="-Xloggc:c:\crowdinstall\logs\atlassian-gc.log"
    ```

- If you are using HSQL as your database server: after installing Crowd as a Windows service, you will need to copy your database files.
  1. Create a folder called c:\windows\system32\database
  2. 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 last changed on May 07, 2008 by smaddox.

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

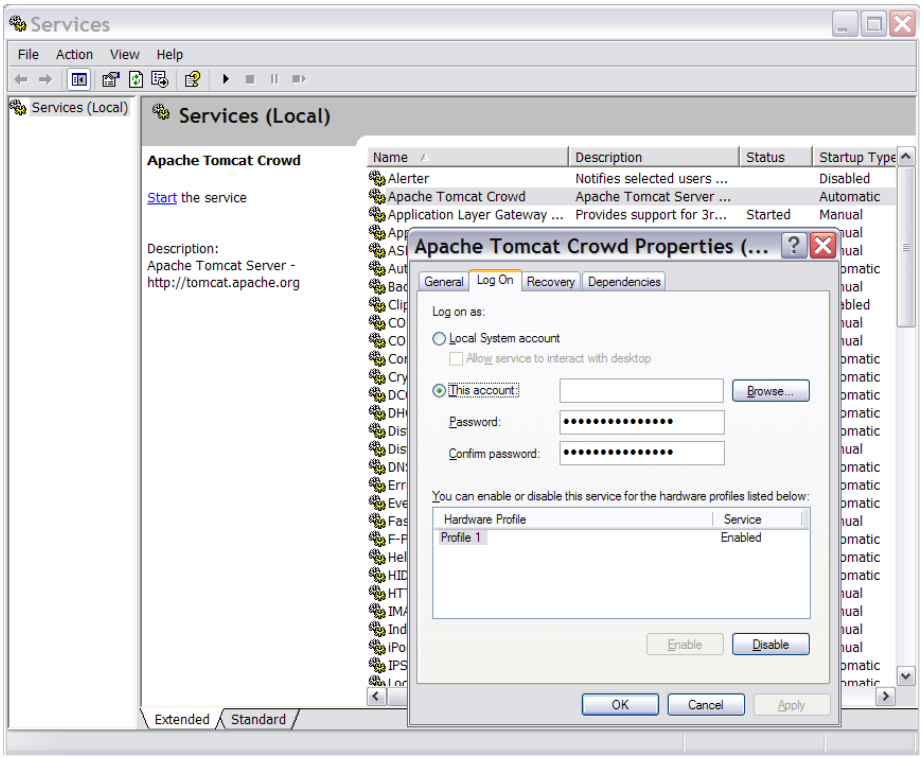
Changing the User for the Crowd Windows Service

This page last changed on May 07, 2008 by smaddox.

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 last changed on May 07, 2008 by smaddox.

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 last changed on May 07, 2008 by smaddox.

This page is relevant if you have installed Crowd as a Windows service.

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`

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

Upgrading Crowd

This page last changed on May 07, 2008 by smaddox.

Below are instructions on upgrading an existing Crowd installation to the latest version of Crowd.

Read the Release Notes and Upgrade Notes

Please read:

- 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:
  - Crowd 1.4 Upgrade Notes
  - Crowd 1.3 Beta Upgrade Notes
  - Crowd 1.3 Upgrade Notes
  - Crowd 1.2 Upgrade Notes
  - Crowd 1.1 Upgrade Notes
  - Crowd 1.0 Upgrade Notes

Select the Instructions for your Crowd Version

From Crowd 1.3.0, the upgrade procedure becomes much simpler. For that reason, we provide two sets of instructions. Select a link below, depending on the version of your current Crowd installation:

- Upgrading from Crowd 1.3.0 or Later
- Upgrading from Crowd 1.2.x or Earlier

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.

RELATED TOPICS

- Crowd Release Notes
- Installing Crowd
- Upgrading Crowd

Upgrading from Crowd 1.3.0 or Later

This page last changed on May 08, 2008 by smaddox.

Follow the instructions below if both the following are true:

- Your current version of Crowd is Crowd 1.3.0 or later, and
- You want to upgrade to the latest version of Crowd.

If your current installation is earlier than Crowd 1.3.0, follow the instructions on upgrading from Crowd 1.2.x or earlier.

On this page:

Error formatting macro: toc: java.lang.NullPointerException

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. Make a backup of 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:
   - Your Crowd Home directory — recommended in case something goes wrong during the upgrade process.
   - 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.
   - The crowd.properties file for the CrowdID application, located at `{CROWD_INSTALL}/crowd-openidserver-webapp/WEBINF/classes/crowd.properties` — you will need to copy this file to your new Crowd installation.
3. If you have installed Crowd on a separate application server, you need to back up your customised configuration files.
4. We recommend that you rename your existing {CROWD_INSTALL} directory, as 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 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'll refer to this installation directory as {CROWD_INSTALL}.
   - Please make sure that your new {CROWD_INSTALL} directory has a different name from your old {CROWD_INSTALL} directory.
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. To specify the directory:

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

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


You must 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:
   - Copy the `crowd.properties` file for the Crowd Administration Console to your new `{CROWD_INSTALL}\crowd-webapp\WEB-INF\classes` directory.
   - Copy the `crowd.properties` file for the CrowdID application to your new `{CROWD_INSTALL}/crowd-openidserver-webapp/WEBINF/classes` directory.
   - If you have installed Crowd on a separate application server, copy your customised configuration files.

Step 4. Update your Integrated Applications

1. Copy the new `{CROWD_INSTALL}\client\crowd-integration-client-X.X.X.jar` file to each Crowd-integrated application's `WEB-INF/lib` folder, replacing the existing `crowd-integration-client-X.X.X.jar` file.
   For details please see the configuration instructions 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 Acegi Security
   - Integrating Crowd with Apache
   - Integrating Crowd with Jive Forums
   - Integrating Crowd with Subversion
   - Integrating Crowd with a Custom Application
2. 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.
3. For better caching, copy the new `{CROWD_INSTALL}\client\conf\crowd-ehcache.xml` file to each Crowd-integrated application's `WEB-INF/classes/` folder, replacing the existing file.


If you are using CrowdID with an external database, you will still need to use the manual JNDI datasource configuration method to configure an external database connection.

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.

RELATED TOPICS

- Crowd Release Notes
- Installing Crowd
- Upgrading Crowd

Upgrading from Crowd 1.2.x or Earlier

This page last changed on May 08, 2008 by smaddox.

Follow the instructions below if both the following are true:

- Your current version of Crowd is Crowd 1.2.x or earlier, and
- You want to upgrade to the latest version of Crowd.

If your current installation is Crowd 1.3.0 or later, follow the instructions on upgrading from Crowd 1.3.0 or later.

There are two options for upgrading your pre-1.3.0 installation:

- Allow the Crowd upgrade process to upgrade your existing Crowd database.
- Or export your database to XML, then re-import it into your new Crowd installation.

Further details are given in the instructions below. Follow the steps and choose your preferred option when prompted.

On this page:

Error formatting macro: toc: java.lang.NullPointerException

Step 1. Optional Export your Crowd Database to XML

You can choose to export your Crowd database to XML and then re-import it when upgrading. You might choose this option if you want to use a different database configuration in your new Crowd installation, such as when you are moving from the HSQLDB evaluation database to an external database.

If you choose this option, follow the instructions on backing up your Crowd database to an XML file.

Step 2. Shut down Crowd and All Integrated Applications

Shut down Crowd and all Crowd-connected applications.

Step 3. Back Up your Crowd Files

1. Make a backup of 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 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.
   - The crowd.properties file for the CrowdID application, located at `{CROWD_INSTALL}/crowd-openidserver-webapp/WEBINF/classes/crowd.properties` — you will need to copy this file to your new Crowd installation.
3. If you have installed Crowd on a separate application server, you need to back up your customised configuration files.
4. We recommend that you rename your existing {CROWD_INSTALL} directory, as legacy files may cause problems if you unzip the new Crowd installation into an existing directory.

Step 4. Re-Install Crowd

1. Download Crowd.
2. Unzip the download archive into a directory of your choice, taking note of the following:
   - 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'll refer to this installation directory as {CROWD_INSTALL}.

- Please make sure that your new {CROWD_INSTALL} directory has a different name from your old {CROWD_INSTALL} directory.

3. Specify a 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. To specify the directory:

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

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

4. Copy the following files, saved in Step 3 above, to your new Crowd installation:
   - Copy the `crowd.properties` file for the Crowd Administration Console to your new `{CROWD_INSTALL}\crowd-webapp\WEB-INF\classes` directory.
   - Copy the `crowd.properties` file for the CrowdID application to your new `{CROWD_INSTALL}/crowd-openidserver-webapp/WEBINF/classes` directory.
   - If you have installed Crowd on a separate application server, copy your customised configuration files.

Step 5. Start Crowd and Run 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.
3. Enter your license key on the 'License' screen, as described in the instructions on the Setup Wizard.
4. When asked for your Installation Type, choose one of the following options:
   - 'Import data from an XML Backup' — Choose this option if you want to import your Crowd data from an XML file which you exported in Step 1 above.
   - 'Upgrade the Database from Crowd Version 1.2.x or Earlier' — Choose this option if you want the Crowd upgrade process to automatically upgrade your existing database.
5. The Setup Wizard will now ask you to configure your database.
   - If you want the upgrade process to update your existing Crowd database, supply the JNDI datasource or JDBC connection details of your existing database.
   - If you are planning to import your data from an XML backup, supply connection details to a new database or to your existing database.

     ⚠ If you decide to import directly into your existing database, please ensure that you have made a backup first.
6. If you have chosen to import from XML, the Import Existing Crowd Data screen will now appear. Enter the location of your XML backup file.
7. The Setup Wizard is now complete. You are now ready to log in to the Crowd Administration Console, using your administrator account from your existing Crowd installation.

Step 6. Update your Integrated Applications

1. From Crowd 1.3.0 onwards, there is just a single Crowd integration library. You will need to remove the legacy integration libraries. Perform the following steps in the **`WEB-INF/lib`** folder of each Crowd-integrated application:
   - Copy the new `{CROWD_INSTALL}\client\crowd-integration-client-X.X.X.jar` file into the folder.
   - Remove the existing `crowd-core-X.X.X.jar` and `crowd-atlassian-user-X.X.X.jar` files from the folder.
     For details please see the configuration instructions 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 Acegi Security
     - Integrating Crowd with Apache
     - Integrating Crowd with Jive Forums
     - Integrating Crowd with Subversion
     - Integrating Crowd with a Custom Application
2. 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.
3. For better caching, copy the new `{CROWD_INSTALL}\client\conf\crowd-ehcache.xml` file to each Crowd-integrated application's `WEB-INF/classes/` folder, replacing the existing file.

 If you are using CrowdID with an external database, you will still need to use the manual JNDI datasource configuration method to configure an external database connection.

RELATED TOPICS

- Crowd Release Notes
- Installing Crowd
- Upgrading Crowd

Upgrade Notes

This page last changed on Feb 18, 2008 by smaddox.

- Crowd 1.0 Upgrade Notes
- Crowd 1.1 Upgrade Notes
- Crowd 1.2 Upgrade Notes
- Crowd 1.3 Beta Upgrade Notes
- Crowd 1.3 Upgrade Notes
- Crowd 1.4 Upgrade Notes

Crowd 1.0 Upgrade Notes

This page last changed on Feb 18, 2008 by smaddox.

- All LDAP configuration now need to have filters set
- If you are using PostgreSQL you need to change the column name attributevalues.attributevalueid to attributevalues.ATTRIBUTEVALUEID (make it uppercase).

Crowd 1.1 Upgrade Notes

This page last changed on May 07, 2008 by smaddox.

To upgrade to Crowd 1.1.x from 1.0.x or earlier,

- Follow the usual steps for upgrading crowd.
- Configure two additional web applications, as described below.

Configuring OpenID Server and OpenID Demo Client applications

In Crowd 1.1, two new web applications have been added to Crowd, along with the Crowd Administration Console and the Demo Application. The new applications are:

Application	Description
OpenID Server	Note: Logically, the OpenID Server is a client application of the Crowd Server, and must be configured as such. The OpenID Server requires a database. By default, a HSQL database is used.
OpenID Demo Client	A simple web application which can be used as a starting point to develop OpenID-enabled Java applications. This application is lightweight. It has no persistence store and does not talk to the Crowd Security Server.

Perform the following steps to finish the upgrade:

1. Create a database to house the data specific to the OpenID Server.
2. Point the application context to the new database. The application context for the OpenID Server is in `atlassian-crowd-1.1.0/apache-tomcat-5.5.20/conf/catalina/localhost/openidserver.xml` More information on how to modify this file for your particular database can be found in Connecting CrowdID to a Database.
3. Update the `atlassian-crowd-1.1.0/crowd-openidserver-webapp/WEB-INF/classes/jdbc.properties` to reflect the dialect of your database.
4. Update `atlassian-crowd-1.1.0/crowd-openidserver-webapp/WEB-INF/classes/crowd.properties` to use a secure password for the OpenID Server application.
5. Add the application via the Crowd Administration Console. The default name of the application is crowd-openid-server and the password is whatever you specified in `crowd.properties` in the previous step. For more information on how to add an application, see Adding an Application.
6. Restart the server. This should set up the OpenID Server in Crowd.

Crowd 1.2 Upgrade Notes

This page last changed on May 05, 2008 by smaddox.

Upgrade Procedure

To upgrade to Crowd 1.2.x from 1.1.x or earlier,

- Follow the instructions on upgrading Crowd.

Upgrade Notes

Application Directory Permissions

With Crowd 1.2, directory permissions can now be set at application level. When you upgrade to Crowd 1.2:

- The upgrade procedure will set all application-level permissions equal to your existing directory-level permissions. This means that, for a particular directory, all applications will have the same permissions immediately after the upgrade i.e. the permissions which were set at directory level before the upgrade.
- You can alter the permissions for each application after the upgrade is complete, if you wish.

Developer Notes

SOAP Service API

There are changes to the Crowd API, including new SOAP methods (see CWD-459 and CWD-537), so you should re-generate your WSDL bindings to the Crowd server.

Crowd 1.3 Beta Upgrade Notes

This page last changed on May 07, 2008 by smaddox.

Crowd 1.3 will be launched in early March 2008. A beta release is currently undergoing internal testing. These upgrade notes apply to Crowd 1.3 beta. We'll publish the final upgrade notes with the release of Crowd 1.3.0.

Upgrade Procedure

To upgrade to Crowd 1.3.x from 1.2.x or earlier, please follow these upgrade instructions.

Upgrade Notes

Database Configuration

Crowd database configuration is now part of the Setup Wizard. You can choose between a JNDI datasource (i.e. server-managed) or a JDBC configuration.

 If you are using CrowdID with an external database, you will still need to use the manual JNDI datasource configuration method to configure an external database connection.

Database Import

You can now import an XML backup of your Crowd database when upgrading. So you don't have to go through the whole Setup Wizard again, nor do a manual backup and restore of your Crowd database files. Full instructions are in the Upgrade Guide.

Integrated Applications

Crowd's client libraries have been slimmed down to a single JAR file containing all required classes for a Crowd client. (See CWD-767.)

 Before upgrading, please remove all previous client libraries (`crowd-XXXX-X.X.X.jar`) from each Crowd-integrated application's `WEB-INF/lib` folder.

Developer Notes

Restructuring of Crowd Client Library

In Crowd 1.3, the Java client library API has been upgraded. This affects applications using the Crowd Client libraries and connectors. Read more about the Client API Changes.

Spring Configuration Upgrade for Crowd Acegi Connector

Applications using the Crowd Acegi connector will need to upgrade their Spring configuration. Refer to the updated documentation for more information.

Crowd 1.3 Upgrade Notes

This page last changed on May 07, 2008 by smaddox.

On this page:

Error formatting macro: toc: java.lang.NullPointerException

Upgrade Notes

Database Configuration

Crowd database configuration is now part of the Setup Wizard. You can choose between a JNDI datasource (i.e. server-managed) or a JDBC configuration.

 If you are using CrowdID with an external database, you will still need to use the manual JNDI datasource configuration method to configure an external database connection.

Database Import

You can now import an XML backup of your Crowd database when upgrading. So you don't have to go through the whole Setup Wizard again, nor do a manual backup and restore of your Crowd database files. Full instructions are in the Upgrade Guide.

Integrated Applications

Crowd's client libraries have been slimmed down to a single JAR file containing all required classes for a Crowd client. (See CWD-767.)

 Before upgrading, please remove all previous client libraries (`crowd-XXXX-X.X.X.jar`) from each Crowd-integrated application's `WEB-INF/lib` folder.

Developer Notes

Restructuring of Crowd Client Library

In Crowd 1.3, the Java client library API has been upgraded. This affects applications using the Crowd Client libraries and connectors. Read more about the Client API Changes.

Spring Configuration Upgrade for Crowd Acegi Connector

Applications using the Crowd Acegi connector will need to upgrade their Spring configuration. Refer to the updated documentation for more information.

Upgrade Procedure

To upgrade to Crowd 1.3.x from 1.2.x or earlier, please follow these upgrade instructions.

Crowd 1.4 Upgrade Notes

This page last changed on May 08, 2008 by smaddox.

This document contains notes on upgrading an existing Crowd installation to Crowd 1.4. You can see the features of this release in the Crowd 1.4 Release Notes.

On this page:

Error formatting macro: toc: java.lang.NullPointerException

Upgrade Notes

Crowd administrators must be in a group mapped to the 'crowd' application

With Crowd 1.4 and later, non-administrators as well as Crowd administrators can log in to Crowd. Non-administrators can update their user profiles and view their authorisation details. To support this, the Crowd permissions now distinguish between Crowd administrators (users in groups mapped to the 'crowd' application) and other Crowd users (all users in directories allowed to authenticate to Crowd).

Impact:

- In previous versions of Crowd, any user authorised to log in to the 'crowd' application had access to the full functionality of the Crowd Administration Console. The default setup used the 'crowd-administrators' group to manage these users. Most of our customers will have used the default group or customised groups for their Crowd administrators. But it was possible to grant entire directories administration access to Crowd, by mapping the directory to the 'crowd' application and allowing all to authenticate.
- In Crowd 1.4 and later, every Crowd administrator must be a member of a group mapped to the 'crowd' application (in any mapped directory). Other users will be able to log in to Crowd and use the Self-Service Console if they are members of mapped directories where all can authenticate. But if they are not members of mapped groups, they will not have full access to the Administration Console.

Before upgrading, check that you have a valid administrator

Before starting the upgrade, ensure that there is at least one user in a group that is mapped to the 'crowd' application.

Additional file to copy for client applications: crowd-ehcache.xml

For better caching, you will need to copy the new `{CROWD_INSTALL}\client\conf\crowd-ehcache.xml` file to each Crowd-integrated application's `WEB-INF/classes/` folder, replacing the existing file.

We have included the above step in the upgrade instructions.

Upgrade Procedure

To upgrade to Crowd 1.4.x from 1.3.x or earlier, please follow these upgrade instructions.

Crowd Knowledge Base

This page last changed on May 07, 2008 by smaddox.

General FAQ on the Atlassian Website

Concepts:

- What is single sign-on (SSO)?
- What is authorisation?
- What is authentication?
- What is centralised 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?
- Does Crowd support SAML or Liberty Alliance?

Deployment FAQ

- Finding your Crowd Home Directory
- Recovering your Console application password
- Resetting the Domain Cookie Value
- Restarting the Setup Wizard from Scratch
- Self Signed Certificate

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 Websphere Integration

More General FAQ

- Principals and Users

Troubleshooting

- Troubleshooting SSO with Crowd

RELATED TOPICS

- Troubleshooting your Configuration on Setup

Deployment FAQ

This page last changed on Mar 11, 2007 by rosie@atlassian.com.

- Finding your Crowd Home Directory
- Recovering your Console application password
- Resetting the Domain Cookie Value
- Restarting the Setup Wizard from Scratch
- Self Signed Certificate

Finding your Crowd Home Directory

This page last changed on May 07, 2008 by smaddox.

The Crowd Home directory is where Crowd stores its configuration information. If you are using the embedded HSQLDB database supplied for evaluation purposes, Crowd will also store its database in this directory.

Crowd's System Information screen shows the location of your Crowd Home directory.

Read more about:

- Setting your Home Directory during installation.
- The location and function of the Crowd Home directory and other important files and directories.

Recovering your Console application password

This page last changed on Oct 10, 2007 by smaddox.

The Crowd console itself must authenticate to the Crowd framework to perform authentication and authorisation calls.

Like an integrated application, if you have an improper password in the `crowd.properties` configuration file, the following exception will be thrown when the application attempts to connect to Crowd SOAP services:

```
Caused by: com.atlassian.crowd.integration.exception.InvalidAuthenticationException: Invalid
 application client.
        at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)
        at
 sun.reflect.NativeConstructorAccessorImpl.newInstance(NativeConstructorAccessorImpl.java:39)
        at
 sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessorImpl.java:27)
        at java.lang.reflect.Constructor.newInstance(Constructor.java:494)
        at org.codehaus.xfire.aegis.type.basic.BeanType.createFromFault(BeanType.java:235)
        at org.codehaus.xfire.aegis.type.basic.BeanType.readObject(BeanType.java:105)
        at
 org.codehaus.xfire.aegis.AegisBindingProvider.readParameter(AegisBindingProvider.java:169)
        at
 org.codehaus.xfire.client.ClientFaultConverter.processFaultDetail(ClientFaultConverter.java:51)
        at org.codehaus.xfire.client.ClientFaultConverter.invoke(ClientFaultConverter.java:32)
        at org.codehaus.xfire.handler.HandlerPipeline.invoke(HandlerPipeline.java:131)
        at org.codehaus.xfire.client.Client.onReceive(Client.java:424)
        at org.codehaus.xfire.transport.http.HttpChannel.sendViaClient(HttpChannel.java:139)
        at org.codehaus.xfire.transport.http.HttpChannel.send(HttpChannel.java:48)
        at org.codehaus.xfire.handler.OutMessageSender.invoke(OutMessageSender.java:26)
        at org.codehaus.xfire.handler.HandlerPipeline.invoke(HandlerPipeline.java:131)
        at org.codehaus.xfire.client.Invocation.invoke(Invocation.java:79)
        at org.codehaus.xfire.client.Invocation.invoke(Invocation.java:114)
        at org.codehaus.xfire.client.Client.invoke(Client.java:336)
        at org.codehaus.xfire.client.XFireProxy.handleRequest(XFireProxy.java:77)
        at org.codehaus.xfire.client.XFireProxy.invoke(XFireProxy.java:57)
        at $Proxy8.authenticateApplication(Unknown Source)
        at
 com.atlassian.crowd.integration.service.soap.client.GenericClient.authenticate(GenericClient.java:263)
        ... 73 more
Caused by: org.codehaus.xfire.fault.XFireFault: Invalid application client.
        at
 org.codehaus.xfire.fault.Soap11FaultSerializer.readMessage(Soap11FaultSerializer.java:31)
        at org.codehaus.xfire.fault.SoapFaultSerializer.readMessage(SoapFaultSerializer.java:28)
        at
 org.codehaus.xfire.soap.handler.ReadHeadersHandler.checkForFault(ReadHeadersHandler.java:111)
        at org.codehaus.xfire.soap.handler.ReadHeadersHandler.invoke(ReadHeadersHandler.java:67)
        at org.codehaus.xfire.handler.HandlerPipeline.invoke(HandlerPipeline.java:131)
        at org.codehaus.xfire.client.Client.onReceive(Client.java:406)
        ... 84 more
```

If the password for the Crowd console is lost, the only method of recovery is to reset the password in the `crowd.properties` configuration file to a known application password. To do this you will need to have access to the Crowd database server and run the following commands:

1. Get a list of the applications integrated with Crowd:

```
mysql> select id, name from application;
+--------+--------------------+
| id     | name               |
+--------+--------------------+
|  98305 | crowd              |
|  98306 | demo               |
|  98307 | crowd-openid-server |
| 655361 | jira               |
| 753665 | jiveforums         |
```

```
+--------+--------------------+
```

2. Choose an application for which you have the password, and where you're happy to use the same password for the Crowd application. Let's call your application 'X'. Use application X's `ID` to query the database and retrieve X's credentials:

```
mysql> select credential from applicationcredentials where applicationid = 655361;
+-----------------------------------------------------------------------------------------+
| credential                                                                              |
+-----------------------------------------------------------------------------------------+
| sQnzu7wkTrgkQZF+0G1hi5AI3Qmzvv0bXgc5THBqi7mAsdd4Xll27ASbRt9fEyavWi6m0QP9B8lThf+rDKy8hg== |
+-----------------------------------------------------------------------------------------+
```

3. Now query the database for the `ID` of the Crowd application and set Crowd's application credentials to the `credential` of your application X:

```
mysql> update applicationcredentials set credential = 'sQnzu7wkTrgkQZF
+0G1hi5AI3Qmzvv0bXgc5THBqi7mAsdd4Xll27ASbRt9fEyavWi6m0QP9B8lThf+rDKy8hg==' where applicationid
 = 98305;
Query OK, 0 rows affected (0.00 sec)
Rows matched: 1  Changed: 0  Warnings: 0
```

4. Update your `atlassian-crowd-1.1.2/crowd-webapp/WEB-INF/classes/crowd.properties` `application.password` value to the value of X's password.
5. You may now start Crowd.

Further information

- If you have installed only Crowd and no other integrated applications, you'll need to clear all the database tables (if you've already hooked up to a database server) and re-install Crowd. This should not cause you to lose much data, since no other applications have yet been defined.
- The issue is that the password for the crowd application is being changed during the setup process for crowd. This problem will be resolved with Crowd 1.2 - see CWD-488.
- You may be tempted to try changing the password back to 'password'. Alas, this won't work, because the passwords are encrypted using SHA1.

Resetting the Domain Cookie Value

This page last changed on Oct 09, 2007 by smaddox.

To reset the SSO (single sign-on) cookie domain, run the following SQL command on the Crowd database:

```
update serverproperty set value = '' where name = 7;
```

Once you have done this you will need to restart Crowd and then log in. This will reset any domain SSO token misconfiguration.

Restarting the Setup Wizard from Scratch

This page last changed on May 07, 2008 by smaddox.

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

This page last changed on Nov 30, 2006 by justen.stepka@atlassian.com.

I have a self Signed Certificate

You will need to add the self-signed certificate to your JDK truststore using the JDK keytool: http://java.sun.com/j2se/1.3/docs/tooldocs/win32/keytool.html

Integration FAQ

This page last changed on Sep 17, 2007 by smaddox.

All Integrations

This page last changed on Oct 02, 2007 by smaddox.

- 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?

This page last changed on May 05, 2008 by smaddox.

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

This page last changed on May 07, 2008 by smaddox.

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.

This applies to the Crowd Administration Console's `crowd.properties` file too. You may find this particularly useful when integrating with a WAR deployment of an integrated application.

Example:

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

Atlassian Product Integration

This page last changed on Feb 17, 2008 by smaddox.

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 what '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

This page last changed on May 05, 2008 by smaddox.

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.

For more information, refer to:

- Caching – information on turning the cache on or off on the Crowd server.
- Configuring Caching for an Application – information on fine tuning the caching properties of the client application.

JIRA integration

This page last changed on May 05, 2008 by smaddox.

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 page last changed on May 05, 2008 by smaddox.

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 Websphere Integration

This page last changed on Sep 11, 2007 by justin.

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.

More General FAQ

This page last changed on Feb 13, 2008 by smaddox.

- Principals and Users

Principals and Users

This page last changed on Feb 13, 2008 by smaddox.

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'.

Troubleshooting

This page last changed on May 07, 2008 by smaddox.

- Troubleshooting SSO with Crowd
- Troubleshooting your Configuration on Setup

Troubleshooting SSO with Crowd

This page last changed on May 07, 2008 by smaddox.

1. Confirm that you can log into each application with the same username and password.

Applications -> Click View next to the application -> Config Test

2. Ensure each Atlassian application's WEB-INF/classes/seraph-config.xml file is using the original authenticator class instead of the com.atlassian.crowd authenticator class. For example in JIRA:

```
<authenticator class="com.atlassian.crowd.integration.seraph.JIRAAuthenticator"/>
```

should revert to

```
<authenticator class="com.atlassian.seraph.auth.DefaultAuthenticator"/>
```

3. Once each application is using centralized authentication instead of SSO, confirm you can log in to each application with the same username and password.

4. Ensure that each application is using the same sub-domain. For example:

- JIRA -> jira.example.com
- Confluence -> confluence.example.com
- Crowd -> crowd.example.com

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 following documentation:

- Configuring the domain

Still having trouble?

1. Under Admin -> Logging & Profiling, please change the com.atlassian.crowd package to DEBUG.
2. Replicate the SSO problem you are having.
3. Attach the resulting {CROWD}/atlassian-crowd.log file to a support issue at http://support.atlassian.com.

Crowd User Guide

This page last changed on May 08, 2008 by smaddox.

About Crowd

Atlassian's Crowd is a software application installed by your system administrator. The administrator has also connected one or more of your organisation's applications to Crowd. When you log in to a Crowd-connected application, Crowd will verify your password and login permissions.

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 | Forums | Knowledge base

Table of Contents

Updating your User Profile

Viewing your Group Membership

Viewing your Role Membership

Viewing your Applications

Crowd User's Glossary

- Authorisation to Use Crowd (Glossary Entry)
- Crowd Administrator (Glossary Entry)
- Crowd-Connected Application (Glossary Entry)
- Directory (Glossary Entry)
- Self-Service Console (Glossary Entry)
- Single Sign-On (Glossary Entry)

Introduction to Crowd

This page last changed on Apr 06, 2008 by smaddox.

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 your system administrator. The administrator has also connected one or more of your organisation's applications to Crowd. When you log in to a Crowd-connected application, Crowd will verify your password and login permissions.

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 Crowd Administration Guide.

Every authorised 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 Crowd 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.

- Authorisation to Use Crowd (Glossary Entry) — If you are authorised 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 (Glossary Entry) — A Crowd administrator is a user who has access to the Crowd Administration Console, which provides the functions described in the Crowd 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 (Glossary Entry) — A 'Crowd-connected application' is a software application which has been defined to and integrated with Crowd. 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 authorisation purposes, and allow single sign-on across the Crowd domain. The Crowd Administration Guide tells you how to connect an application to Crowd.
- Directory (Glossary Entry) — 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.
- Self-Service Console (Glossary Entry) — Authorised 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 Crowd User Guide describes this functionality. The Crowd Administration Console presents the full range of Crowd administration functionality to authorised Crowd administrators.
- Single Sign-On (Glossary Entry) — 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:

RELATED TOPICS

Logging in to Crowd
Crowd User Guide

Logging in to Crowd

This page last changed on Apr 06, 2008 by smaddox.

If you are authorised 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, then you only need to log in once. When you start another Crowd-connected application, you will be logged in automatically.

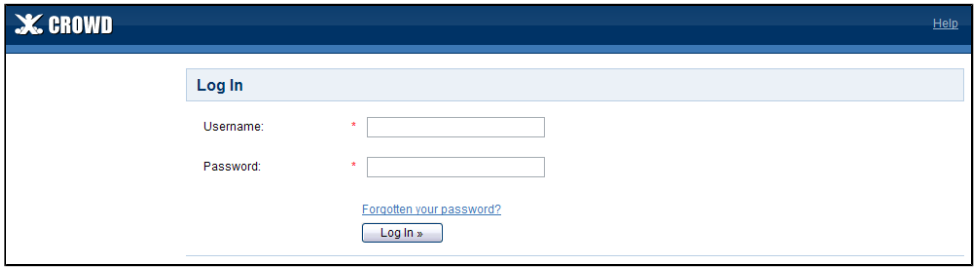
To log in to Crowd,

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. Type in your username and password.
3. Click the 'Log In' button.

Screenshot: Crowd login screen



If you have forgotten your password, you can click the 'Forgotten your password' link. Crowd will email you a new password. Read more about resetting your password.

RELATED TOPICS

Logging out of Crowd
Resetting your Password
Crowd User Guide

Logging out of Crowd

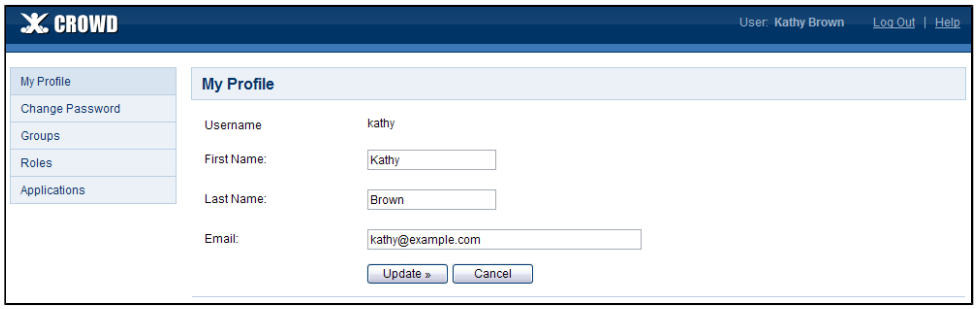
This page last changed on Apr 06, 2008 by smaddox.

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, then you will be automatically logged out of all Crowd-connected applications when you log out of Crowd.

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.

Screenshot: Crowd screen showing 'Log Out' link



RELATED TOPICS

Logging in to Crowd
Crowd User Guide

Changing or Resetting your Password

This page last changed on May 05, 2008 by smaddox.

If you are authorised 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 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 organisations 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 or reset 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.

RELATED TOPICS

Logging in to Crowd
Crowd User Guide

Changing your Password

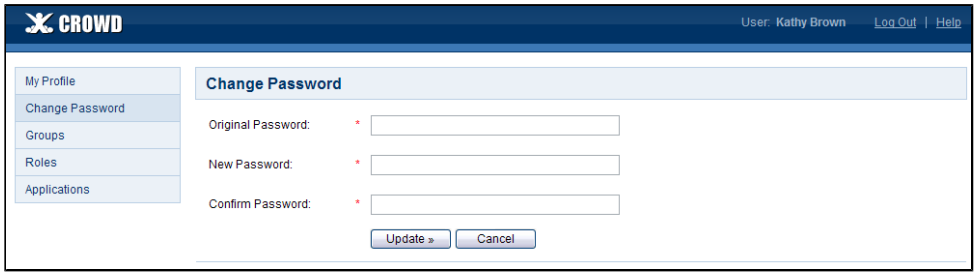
This page last changed on Apr 11, 2008 by smaddox.

If you are authorised to use Crowd, you can log in to Crowd's Self-Service Console and change 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:
    - Original Password — Your current password.
    - New Password — Your new password.
    - 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.

Screenshot: Crowd's Change Password Screen



Password change applies to one user directory only

In most cases, your username will be defined in one user directory only. But some organisations 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 or reset 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.

RELATED TOPICS

Resetting your Password
Logging in to Crowd
Crowd User Guide

Resetting your Password

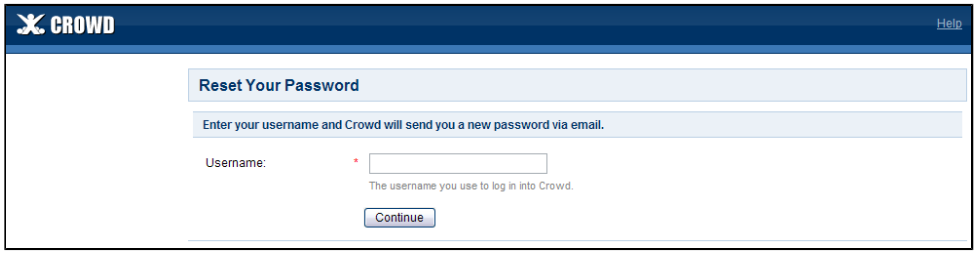
This page last changed on Apr 11, 2008 by smaddox.

The Crowd 'Login' screen allows you to reset your password. This is useful when you have forgotten the password.

To reset your password,

1. Open Crowd in your browser.
2. Click the 'Forgotten your password' link on the Crowd login screen.
3. The 'Reset Your Password' screen will appear, as shown in the screenshot below. Type in your Crowd username and click the 'Continue' button.
4. A message will appear: 'Your new password is on the way!' Click the 'Home' link at the top of the screen.
5. You will receive an email message with your new password. Copy the password.
6. Log in to Crowd using the new password.
7. Change your password to one you can remember easily.

Screenshot: Crowd's Reset Your Password screen



Password change applies to one user directory only

In most cases, your username will be defined in one user directory only. But some organisations 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 or reset 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.

RELATED TOPICS

Changing your Password
Logging in to Crowd
Crowd User Guide

Updating your User Profile

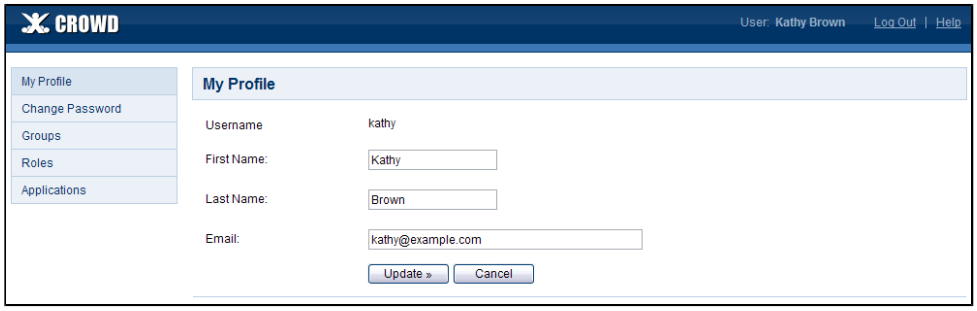
This page last changed on May 05, 2008 by smaddox.

Provided that you are authorised 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.

Screenshot: Crowd user profile



Which user directories are updated?

In most cases, your username will be defined in one user directory only. But some organisations 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.

RELATED TOPICS

Changing or Resetting your Password
Crowd User Guide

Viewing your Group Membership

This page last changed on Apr 11, 2008 by smaddox.

Provided that you are authorised 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.

Screenshot: Groups

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 organisations 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.

RELATED TOPICS

Crowd User Guide

Viewing your Role Membership

This page last changed on Apr 11, 2008 by smaddox.

Provided that you are authorised to use Crowd, you can see a list of the roles to which your username is assigned.

To see which roles you have been assigned,

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 'Roles' in the left-hand menu.
4. The 'Roles' screen will appear, as shown in the screenshot below.

Screenshot: Roles

Each role appears only once

Even if you are a member of the same role in more than one directory, the role 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 organisations 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 role (e.g. 'hr-admin') in both directories. On the Crowd 'Roles' screen, the role 'hr-admin' will appear only once.

RELATED TOPICS

Crowd User Guide

Viewing your Applications

This page last changed on Apr 11, 2008 by smaddox.

Provided that you are authorised to use Crowd, you can see a list of the applications you are authorised to log in to.

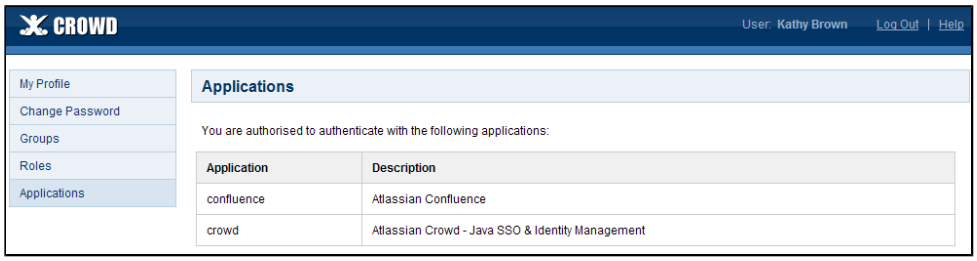
More information about the applications listed:

- Crowd verifies all logins to these applications. Your Crowd administrator has defined them as Crowd-connected applications.
- Your username is authorised 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.

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.

Screenshot: Applications



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.

RELATED TOPICS

Viewing your Group Membership
Crowd User Guide

Crowd User's Glossary

This page last changed on Apr 06, 2008 by smaddox.

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.

- Authorisation to Use Crowd (Glossary Entry) — If you are authorised 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 (Glossary Entry) — A Crowd administrator is a user who has access to the Crowd Administration Console, which provides the functions described in the Crowd 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 (Glossary Entry) — A 'Crowd-connected application' is a software application which has been defined to and integrated with Crowd. 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 authorisation purposes, and allow single sign-on across the Crowd domain. The Crowd Administration Guide tells you how to connect an application to Crowd.
- Directory (Glossary Entry) — 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.
- Self-Service Console (Glossary Entry) — Authorised 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 Crowd User Guide describes this functionality. The Crowd Administration Console presents the full range of Crowd administration functionality to authorised Crowd administrators.
- Single Sign-On (Glossary Entry) — 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:

RELATED TOPICS

Introduction to Crowd
Crowd User Guide

Authorisation to Use Crowd (Glossary Entry)

This page last changed on May 05, 2008 by smaddox.

If you are authorised 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.

RELATED TOPICS

Introduction to Crowd
Crowd User Guide

Crowd Administrator (Glossary Entry)

This page last changed on May 05, 2008 by smaddox.

A Crowd administrator is a user who has access to the Crowd Administration Console, which provides the functions described in the Crowd 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.

RELATED TOPICS

Introduction to Crowd
Crowd User Guide

Crowd-Connected Application (Glossary Entry)

This page last changed on May 05, 2008 by smaddox.

A 'Crowd-connected application' is a software application which has been defined to and integrated with Crowd. 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 authorisation purposes, and allow single sign-on across the Crowd domain. The Crowd Administration Guide tells you how to connect an application to Crowd.

RELATED TOPICS

Introduction to Crowd
Crowd User Guide

Directory (Glossary Entry)

This page last changed on May 04, 2008 by smaddox.

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.

RELATED TOPICS

Introduction to Crowd
Crowd User Guide

Self-Service Console (Glossary Entry)

This page last changed on May 05, 2008 by smaddox.

Authorised 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 Crowd User Guide describes this functionality. The Crowd Administration Console presents the full range of Crowd administration functionality to authorised Crowd administrators.

RELATED TOPICS

Introduction to Crowd
Crowd User Guide

Single Sign-On (Glossary Entry)

This page last changed on Apr 06, 2008 by smaddox.

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.

RELATED TOPICS

Introduction to Crowd
Crowd User Guide

This page last changed on Mar 08, 2007 by rosie@atlassian.com.

Crowd Home

- Crowd Documentation
- Blogs

Development

- Feedback
- Request a Feature

Blogs

This page last changed on Sep 29, 2006 by justen.stepka@atlassian.com.

Title	Author	Date Posted

__newreleaseCrowd

Crowd 1.4 has now been released — see the Crowd 1.4 Release Notes

TreeNavigation

This page last changed on Aug 09, 2007 by rosie@atlassian.com.

Index