You are here: <https://help.xtontech.com/content/administrators-and-power-users/settings-and-configurations/command-line-utility-reference-guide.htm>

Command Line Utility Reference Guide

# Table of Contents

[Table of Contents 1](#_Toc109815534)

[Terminology 2](#_Toc109815535)

[Introduction 2](#_Toc109815536)

[Common Parameters 3](#_Toc109815537)

[Configuration Commands 4](#_Toc109815538)

[SetAdminPassword 4](#_Toc109815539)

[SetMasterPassword 4](#_Toc109815540)

[SetDBPassword 4](#_Toc109815541)

[Init 5](#_Toc109815542)

[CreateUser 5](#_Toc109815543)

[CreateGroup 5](#_Toc109815544)

[RenameGroup 5](#_Toc109815545)

[SetUserPassword 6](#_Toc109815546)

[ADConnect 6](#_Toc109815547)

[ADQuery 7](#_Toc109815548)

[LdapConnect 7](#_Toc109815549)

[ADSConnect 8](#_Toc109815550)

[ADSReplicate 8](#_Toc109815551)

[ADSReplication 9](#_Toc109815552)

[ADSExport 9](#_Toc109815553)

[ADSImport 10](#_Toc109815554)

[DBConnect 10](#_Toc109815555)

[XTConnect 10](#_Toc109815556)

[ConfigureRealms 10](#_Toc109815557)

[EnableSso 11](#_Toc109815558)

[GenerateSSL 11](#_Toc109815559)

[CheckWebStatus 11](#_Toc109815560)

[CheckAdsStatus 11](#_Toc109815561)

[GenerateCASKeys 12](#_Toc109815562)

[GenerateCASCipher 12](#_Toc109815563)

[GenerateCertificate 12](#_Toc109815564)

[ExportCertificate 12](#_Toc109815565)

[File 13](#_Toc109815566)

[EnableNonOpenMode 13](#_Toc109815567)

[Tool Commands 13](#_Toc109815568)

[Sign 13](#_Toc109815569)

[Encrypt 14](#_Toc109815570)

[TestCertificate 14](#_Toc109815571)

[SSLPoke 14](#_Toc109815572)

[SSLImport 14](#_Toc109815573)

[ADTest 15](#_Toc109815574)

[DecryptFile 15](#_Toc109815575)

[PublishedVersion 15](#_Toc109815576)

[Break Glass Commands 16](#_Toc109815577)

[ListExport 16](#_Toc109815578)

[Extract 16](#_Toc109815579)

[Database Configuration Commands 16](#_Toc109815580)

[DBReleaseLockedAdmins 16](#_Toc109815581)

[DBListAdmins 17](#_Toc109815582)

[DBMakeAdmin 17](#_Toc109815583)

[DBListUsers 17](#_Toc109815584)

[DBUnblock 18](#_Toc109815585)

[Appendix A: Summary of commands 18](#_Toc109815586)

# Terminology

* PAM – Privileged Access Management
* CLU – PAM Command Line Utility
* CLI – command line interface
* $PAM – PAM Server deployment folder
* ADS – internal user directory service
* CDN – content distribution network for PAM binaries

# Introduction

PAM Command Line Utility (CLU) is a helper utility to simplify and to facilitate PAM Server configuration and deployment activities.

This guide expects CLU will be launched from the $PAM folder by providing full or relative path to the utility launcher. A t*ypical issue with launching CLU is running it from a location other than the $PAM folder (typically $PAM/bin which is incorrect)*

Example of the launch command for Linux hosts is given below

./bin/PamDirectory.sh COMMAND PARAMETER1 PARAMETER2 …

Example of the launch command for Windows hosts is given below

.\bin\PamDirectory.cmd COMMAND PARAMETER1 PARAMETER2 …

Below are several examples of the command run on a Linux host to integrate with Active Directory prompting for the user’s password

./bin/PamDirectory.sh ADConnect web dc-server ad-service-user -

./bin/PamDirectory.sh ADConnect web ldaps://dc-server:636 ad-service-user -

# Common Parameters

* catalina.home is the folder where PAM Server WEB Container is located. Typically, it is $PAM/web folder. When the CLU is launched from the $PAM folder it could be specified using a relative path: web
* password – in majority of the cases it is possible to specify dash ( - ) instead of the password in the command line to make the CLU to allow a user to type the password in a secure input prompt. This option masks the password during CLU execution and to simplify the entry of special characters.
* File PATH\_TO\_THE\_INSTRUCTION\_FILE – parameter *File* allows one to provide CLU the text file with the list of parameters instead of providing command parameters using command line. The parameters file is a regular properties file with all parameters defined under the same section called: **instruction**. Command parameter is specified using property **command**. All other parameters are specified as properties with the property name as it is defined in this guide. Note that unlike the ordered unnamed command line parameters, parameters given by file specification are unordered and named with the required names defined in this document in the Parameters section. Example of the File properties file for the command SetAdminPassword is given below

[instruction]

catalina.home=web

admin.password=pam-generate

# Configuration Commands

This section describes the commands mostly used to manage the deployment configuration. Parameters in the commands in this section could be specified as positional command line parameters or using the instruction file using single File parameter.

## SetAdminPassword

The command replaces password of the service account (DN: uid=admin,ou=system) in the local user directory services. The current password is obtained from the existing configuration. The command also updates this password encrypted in the PAM Server configuration file. A service restart is required after executing this command.

Parameters

* catalina.home
* admin.password - New administrator password parameter might be a password itself, a dash to prompt a user to enter a password, or one of the keywords <generate> or pam-generate to generate new password. Generated password will be printed on the console after the command execution

## SetMasterPassword

The command sets a new master password used to encrypt and decrypt sensitive data in the PAM Vault. Note that the command does not re-encrypt existing records so updating master password might cause existing records to become unreadable by the system in case they were encrypted by different master password.

The command is useful in cases like deploying new node to the existing farm or to migrate back-end database to another system.

Parameters

* catalina.home
* master.password - New master password parameter might be a password itself, a dash to prompt a user to enter a password, or one of the keywords <generate> or pam-generate to generate new password. Generated password will be printed on the console after the command execution

## SetDBPassword

This command replaces a password of a service account for the embedded database in cases when the system is deployed with the internal Apache Derby database. The command also updates this password encrypted in the PAM Server configuration file. A service restart is required after executing this command.

Parameters

* catalina.home
* db.home is a folder where the internal Apache Derby database is deployed. Typically, it is $PAM/db folder. When the CLU is launched from the $PAM folder it could be specified using the relative path: db
* db.password – New database administrator password parameter might be a password itself, a dash to prompt a user to enter a password, or one of the keywords <generate> or pam-generate to generate new password. Generated password will be printed on the console after the command execution

## 

## Init

The command initializes new local user directory service with the LDAP structure required for PAM operations. Note that the software initializes the directory services during installation. The command is useful to build a new directory service structure. It is not recommended to use this command on an existing, functional deployment.

Parameters

* catalina.home

## CreateUser

This command creates a local user in the local user directory.

Parameters

* catalina.home
* user.login is the login name of the newly create account
* user.firstName is the first name
* user.lastName is the last name
* user.password is a new user password that might be a password itself, a dash to prompt a user to enter a password, a keyword GENERATE to generate new password. Generated password will be printed on the console after the command execution. The password might be started with base64- prefix. In this case the command will treat the string after the prefix as a Base64 decided password.

## CreateGroup

This command creates a local group in the local user directory optionally with one specified group member.

Parameters

* catalina.home
* group.name is a name of a new group
* group.description is a description of a new group
* group.member is the optional member of a new group given as the login name of the account

## RenameGroup

The command renames a local group in the local user directory

Parameters

* catalina.home
* group.name is the existing group name
* group.newName is the new group name

## SetUserPassword

This command sets a new password for an existing local user.

Parameters

* catalina.home
* ads.password is a local user directory service account password
* user.login is the user login
* user.password is the new user password. The password might be started with base64- prefix. In this case the command will treat the string after the prefix as a Base64 decided password.

## ADConnect

This command sets up integration with Microsoft Active Directory. The result of the successful execution of this command is the set of AD connection properties defined in the $PAM/web/conf/catalina.properties file. The PAM service needs to be restarted after executing this command. This command establishes connection to one Active Directory server. Establishing another connection will replace the previous connection. To manage integration with multiple LDAP / AD servers use the LdapConnect command.

Read more about integration with MS Active Directory in the application’s help system.

Parameters

* catalina.home
* ldap.server is the AD Domain Controller server or URL. When just server is specified the CLU attempts to connect to the AD Global Catalogue first and then to the AD Domain Controller itself using LDAPS protocol. Custom protocols or ports could be specified using full URL notation like in the example ldaps://ad-server.company.com:port . Note that for the secure communications using LDAPS protocol the AD server certificate should be imported into the PAM Server keystore (see the command SSLImport) and the name on the certificate should match the name of the AD-server host in the URL.
* ldap.user is the service account in MS Active Directory
* ldap.password is the service account password that might be a password itself or a dash to prompt a user to enter a password. The password might be started with base64- prefix. In this case the command will treat the string after the prefix as a Base64 decided password.

Example

Configure a new AD connection requesting the command to prompt for the service user password

*./bin/PamDirectory.sh ADConnect web ldaps://eu-dc-server pam-service-eu -*

## ADQuery

This command executes a provided LDAP Query in the first integrated MS Active Directory to test the connection and to test the query.

Parameters

* catalina.home
* ldap.query is the LDAP query to execute
* -v is the optional parameter to enable additional logging information printed during the query execution. Note that for the File method to pass parameters this parameter in the properties file should be called *verbose*

## LdapConnect

This command manages integrations with multiple LDAP user directories. Specifically, the command can either add new or delete existing integrations. The result of the successful execution of this command is the set of LDAP connection properties defined in the $PAM/web/conf/catalina.properties file. The PAM service needs to be restarted after executing this command. To manage integration with a single AD server, administrators might use the ADConnect command.

Read more about integration with multiple LDAP user directories in the application’s help system.

Parameters

* catalina.home
* ldap.name is the friendly name of the integrated LDAP server. This name is used in the consequent LDAP management commands as well as in the PAM system as a reference point for the external users. Use alphanumeric characters only.
* ldap.server is LDAP server or URL. Custom protocols or ports could be specified using full URL notation like in the example ldaps://ldap-server.company.com:port . Note that for the secure communications using LDAPS protocol the LDAP server certificate should be imported into the PAM Server keystore (see the command SSLImport) and the name on the certificate should match the name of the LDAP server host in the URL. Alternatively, this parameter might contain the keyword **DISABLE** to remove integration with this LDAP server referenced by name parameter. When DISABLE keyword is specified in this parameter no further positional parameters are required.
* ldap.user is the service account in LDAP server
* ldap.password is the service account password that might be a password itself or a dash to prompt a user to enter a password. The password might be started with base64- prefix. In this case the command will treat the string after the prefix as a Base64 decided password.

Examples

Configure new LDAP connection assigning it new name EU-DC requesting the command to prompt for the service user password

*./bin/PamDirectory.sh LDAPConnect web EU-DC ldaps://eu-dc-server pam-service-eu -*

Delete configured LDAP connection with the name EU-DC

*./bin/PamDirectory.sh LDAPConnect web EU-DC DISABLE*

## ADSConnect

This command establishes a connection with the internal user directory service (ADS). The connection with the local internal user directory is established during installation. The command is useful to connect to a remotely deployed internal user directory.

Parameters

* catalina.home
* ads.server is ADS server or URL. Custom protocols or ports could be specified using full URL notation like in the example ldaps://ldap-server.company.com:port .
* ads.password is the service account (for the account DN: uid=admin,ou=system) password that might be a password itself or a dash to prompt a user to enter a password. The password might be started with base64- prefix. In this case the command will treat the string after the prefix as a Base64 decided password.

## ADSReplicate

This command establishes a local user directory replication with the other node of a two-node deployment. Executing this command with the different server will replace the previously established replication. To manage replication for more than two node deployments use the ADSReplication command.

Read more about the internal user directory replication in high availability deployments in the help system.

Parameters

* catalina.home
* ads.remote.server is the host of the replicating system
* ads.remote.password is the service account (for the account DN: uid=admin,ou=system) password that might be a password itself or a dash to prompt a user to enter a password. The password might be started with base64- prefix. In this case the command will treat the string after the prefix as a Base64 decided password.

## ADSReplication

This command manages internal user directory replication between multiple nodes. Each replication process in the local node is associated with the numerical index (1, 2, 3, …) used in this command. For example, in three-node deployment each node will maintain two replication processes indexed by numbers 1 and 2 on each node with each of the other nodes.

Read more about the internal user directory replication in high availability deployments in the help system.

Parameters

* catalina.home
* ads.remote.index is the replication process index on this node to apply below parameters to. Alternatively, this parameter might have a keyword **list**. This list command will list parameters of all configured replication processes on this node. When list command is specified, no other parameters are needed.
* ads.remote.server is the host of the replicating system. Alternatively, this parameter might contain keywork **delete** to delete this replication process from the local node.
* ads.remote.password is the service account (for the account DN: uid=admin,ou=system) password that might be a password itself or a dash to prompt a user to enter a password. The password might be started with base64- prefix. In this case the command will treat the string after the prefix as a Base64 decided password.

Examples

List configured replication slots

*./bin/PamDirectory.sh ADSReplication web list*

Delete configured replication slots number 3

*./bin/PamDirectory.sh ADSReplication web delete 3*

Configure replication slot number 3 requesting the command to prompt for the password

*./bin/PamDirectory.sh ADSReplication web 3 pam-node-d -*

## ADSExport

This command exports content of internal local user directory service to an external XML file.

Parameters

* catalina.home
* file is the file path
* encrypted is a true or false parameter indicating whether the export file should contain sensitive data encrypted

## ADSImport

This command imports content of the exported file to the internal user directory service.

Parameters

* catalina.home
* file is the file path

## DBConnect

This command changes connection to a new back-end database. The command expects the database server to exist with the appropriate instance and the data-storage with the name PamDB pre-created. A service restart is required after successful execution of this command.

Parameters

* catalina.home
* db.type is the vendor of the back-end database with possible values: Derby, MySql, MSSQL, Oracle, PostgreSQL
* db.server is the host name of the back-end database. The command will try the port default for the vendor in case the port is not specified. Otherwise, specify port in host:port notation
* db.user is the service account for the database integration
* db.password is the service account password that might be a password itself or a dash to prompt a user to enter a password. The password might be started with base64- prefix. In this case the command will treat the string after the prefix as a Base64 decided password.

## XTConnect

This command run on the remote PAM node establishes a connection with the master node. The communication between remote and master node is done using the https protocol.

Parameters

* catalina.home
* pam.server is the host of the master node in the form of host, host:port or https://host:port.
* pam.user is the service account with the Service role in the master node.
* pam.password is the service account password that might be a password itself or a dash to prompt a user to enter a password. The password might be started with base64- prefix. In this case the command will treat the string after the prefix as a Base64 decided password.

## ConfigureRealms

This command manages single server integration with MS Active Directory services. The PAM service should be restarted after successful execution of this command. The command is used by the installation process leaving the configuration complete after the successful installation. The command modifies properties in $PAM/web/conf/server.xml that could be otherwise managed directly without this command.

Parameters

* catalina.home
* auth.catalina.enable is an indicator with possible true or false values to manage combined user realm
* auth.ad.enable is an indicator with possible true or false values to manage Active Directory pool connector as a part of the combined realm.

## EnableSso

This command manages configuration of the Federated Sign-In module. The command is used by the installation process leaving the configuration complete after a successful installation. The command modifies properties in $PAM/web/conf/server.xml that could be otherwise managed directly without this command. A service restart is required after successful execution of this command.

Parameters

* catalina.home
* managed.path is the expected managed path of the PAM WEB Application it will be accessed by its WEB users.
* sso.enable enable is an indicator with possible true or false values to enable or disable the Federated Sign-In configuration

## GenerateSSL

This command generates a new self-signed SSL certificate to secure the PAM server WEB application traffic and applies it to the PAM WEB container. A service restart is required after successful execution of this command.

Parameters

* catalina.home

## CheckWebStatus

This command checks PAM system logs to detect whether PAM service had started up.

Parameters

* catalina.home

## CheckAdsStatus

This command checks ADS system logs to detect whether ADS service had started up.

Parameters

* catalina.home
* ads.check.timeout.ms is the delay in milliseconds

## GenerateCASKeys

This command generates unique encryption and signature keys for use by the Federated Sign-In module. Each installation procedure runs this command to generate unique keys for each deployment. A service restart is required after successful execution of this command.

Read more about synchronizing encryption and signature keys for high availability deployments in the help system.

Parameters

* catalina.home

## GenerateCASCipher

This command encrypts provided sensitive password using the algorithm used by the Federated Sign-In module to put to the configuration file so that sensitive password would not be stored in the configuration files unencrypted.

Read more in the system help for the information about the technique to use encrypted Federated Sign-In module passwords in the configuration files.

Parameters

* catalina.home
* password might be a password itself or a dash to prompt a user to enter a password.

## GenerateCertificate

This command generates a unique certificate bundle for use by the WEB Session manager to protect traffic between the PAM master node and the WEB Session Manager. The command also uploads the bundle into the ADS storage.

Parameters

* catalina.home
* bundle.file.name is the file name for the generated new ZIP bundle

## ExportCertificate

This command extracts the WEB Session Manager certificate bundle from the ADS storage to the local file system.

Parameters

* catalina.home
* bundle.file.name is the file name for the extracted new ZIP bundle. For example, pamcert.zip

## File

This command allows one to replace all positional command line parameter with the instruction file that contains all these parameters named with the names gives in this guide. This option provides better control over the command execution by automated scripts and over special characters that can appear in the parameter.

See more details about the instruction file structure with the example in the Common Parameters File description in this guide.

## EnableNonOpenMode

This command enables non-open mode in Federated Sign-In module that forbids logins to the PAM WEB application with the unapproved destination service provided in the login URL.

*Note that the only method to change managed service on the application login is to edit the login URL in the browser. The attempt to login to the unapproved destination service is not a security risk but it may resolve some audit findings.*

Parameters

* catalina.home

# Tool Commands

This section describes the utility commands to test various connections, certificates, encryptions, to establish trusts, and help to understand deployment environments and relationships between system components.

Parameters for commands in this section could only be specified as positional command line parameters.

## Sign

This command signs a binary file with the signature key from the provided key store. PAM Server can verify the signature using the public key certificate included with every deployment.

Parameters

* KEYSTORE\_PATH – path to the keystore with the signature private key
* KEYSTORE\_PASSWORD – password to the keystore
* FILE\_PATH – binary file to sign

## Encrypt

This command encrypts a provided text string so it could be used in the configuration file $PAM/web/conf/catalina.properties for sensitive data. PAM Server internally decrypts passwords in the configuration files encrypted by this command.

Parameters

* SECRET is the secret sensitive data to decrypt. Alternatively, it is possible to type dash (-) instead of the data itself for the command to prompt user to type the data in the hidden prompt.

## TestCertificate

This command attempts to connect to the provider HTTPS URL to test the validity of the certificate securing this URL endpoint. The command issues a connection error in case of issues.

Parameters

* URL is the HTTPS URL to connect

## SSLPoke

This command attempts to connect to the TCP endpoint given by host and port parameters. After connection the command obtains a certificate from the connected endpoint and analyze its validity.

Parameters

* Host is the endpoint host
* Port is the endpoint port

## SSLImport

This command connects to the specified TCP endpoint, validates its SSL certificate and imports it to the PAM Server keystore. After successful connection, the command lists all certificates from the certificate chain retrieved from the connected endpoint allowing a user to import selected certificates into the PAM Server keystore. The command also generates errors of certificate validation and trust in case of the detected certificate or trust issues.

The command is useful to establish trust for PAM Server with various components connected using various protocols in case the protocols are secured by the self-signed SSL certificate generated in-house as oppose to well-known Internet certificate authorities. Examples of such components include: MS Active Directory, external LDAP server (eDirectory, OpenLDAP), PAM Remote Session Manager, PAM ADS, PAM WEB Container, PAM Load Balancer, etc

Parameters

* Host is the endpoint host
* Port is the endpoint port
* in-place is an optional parameter with true (default) or false values that allows to import the certificate into the temporary local file instead of directly to the PAM Server keystore

Example

*./bin/PamDirectory.sh SSLImport remote-session-manager-host 4822*

## ADTest

This command tests connection to the provided LDAP server without making configuration changes in the system.

Parameters

* LDAP\_SERVER is an LDAP server host or URL
* LDAP\_USER is an LDAP server user
* LDAP\_PASSWORD is and LDAP server password or a dash to prompt a user to enter a password. The password might be started with base64- prefix. In this case the command will treat the string after the prefix as a Base64 decided password.

## DecryptFile

This command decrypts a file (such as session recording stored on the file system) encrypted by PAM Server using provided master password.

Parameters

* encrypted.file is path to the source encrypted file
* unencrypted.file is path to the destination decrypted file

## PublishedVersion

This command downloads the last published PAM Server version from PAM binaries content distribution network. The command allows to test the connection from the server hosting PAM deployment to the PAM CDN. The command also allows to test connection established through HTTP proxy in case direct traffic to PAM CDN is blocked but proxy traffic is allowed.

Parameters

* pam.edition is the optional parameter with the possible values product (default) for production version or qa for QA version
* proxy is the optional parameter to specify proxy host and port in the notation proxy:port to test indirect proxied connection to PAM CDN. Alternatively, specify dash (-) in this parameter to indicate that direct connection should be user instead of proxy to allow to use next positional parameter to test protocol security
* security.level is the optional parameter to specify different levels of protocol security with possible values SSLv3, TLSv1, TLSv1.1, TLSv1.2.

# Break Glass Commands

This section describes summary to break glass commands. Refer to the PAM system help for more information about break glass scenarios and workflows.

Parameters for commands in this section could only be specified as positional command line parameters.

## ListExport

This command lists records from the system export that match the provided criteria.

Parameters

* export.file is the full path to the export file or to the export file base: a file name without extension and export index to include all export files into the archive search.
* match is the criteria to search for records. The command will perform a case insensitive search for a match criterion as a substring of record name, description or host. Alternatively, a criterion might be a record ID.

## Extract

This command extracts sensitive information of the specified records from the system export and prints it on the screen. The command requires the master password.

Parameters

* export.file is the full path to the export file or to the export file base: a file name without extension and export index to include all export files into the archive search.
* name is the criteria to search for records. The command will perform a case insensitive search for a match criterion as a substring of record name, description or host. Alternatively, a criterion might be a record ID.
* master is a master password to decrypt the records sensitive data. The master password might be a password itself or a dash to prompt a user to enter a password. The password might be started with base64- prefix. In this case the command will treat the string after the prefix as a Base64 decided password.

# Database Configuration Commands

This session describes commands that allow admins to manipulate database configuration usually performed using the application WEB GUI. However, there are situations that require direct intervention to the database mostly caused by configuration mistakes that are difficult to recover. Commands in this section require master password as the authentication method.

## DBReleaseLockedAdmins

This command removes Administrator block from all workflow bindings in the system to unlock system administrators who accidentally locked themselves by applying workflow bindings they could not pass (blocking, interactive approval with missed required approvers). The command requires a valid master password.

Parameters

* catalina.home
* master is a master password to decrypt the records sensitive data. The master password might be a password itself or a dash to prompt a user to enter a password. The password might be started with base64- prefix. In this case the command will treat the string after the prefix as a Base64 decided password.

## DBListAdmins

This command lists system users with the System Administrator node. The command requires a valid master password.

Parameters

* catalina.home
* master is a master password to decrypt the records sensitive data. The master password might be a password itself or a dash to prompt a user to enter a password. The password might be started with base64- prefix. In this case the command will treat the string after the prefix as a Base64 decided password.

## DBMakeAdmin

This command adds an existing account to the System Administrator role. The command requires a valid master password.

* catalina.home
* login is the account of the user in PAM to assign global System Administrator role
* master is a master password to decrypt the records sensitive data. The master password might be a password itself or a dash to prompt a user to enter a password. The password might be started with base64- prefix. In this case the command will treat the string after the prefix as a Base64 decided password.

## DBListUsers

This command lists cached users for further analysis. The command requires a valid master password.

Parameters

* catalina.home
* master is a master password to decrypt the records sensitive data. The master password might be a password itself or a dash to prompt a user to enter a password. The password might be started with base64- prefix. In this case the command will treat the string after the prefix as a Base64 decided password.

## DBUnblock

This command unblocks the user by provided login. The command requires a valid master password.

The command is useful to unlock locked administrators in cases of single system administrator present in the system. Note that the user block might happen because of various circumstances such as in the result of a behavior analytics rules.

* catalina.home
* user is the user login to unlock
* master is a master password to decrypt the records sensitive data. The master password might be a password itself or a dash to prompt a user to enter a password. The password might be started with base64- prefix. In this case the command will treat the string after the prefix as a Base64 decided password.

# Appendix A: Summary of commands

* SetAdminPassword catalina.home admin.password|<generate>|pam-generate
* SetMasterPassword catalina.home master.password|-|<generate>|pam-generate
* SetDBPassword catalina.home db.home db.password|<generate>|pam-generate
* Init catalina.home
* CreateUser catalina.home user.login user.firstName user.lastName user.password|GENERATE
* CreateGroup catalina.home group.name group.description [group.member]
* RenameGroup catalina.home group.name group.newName
* SetUserPassword catalina.home ads.password user.login user.password
* ADConnect catalina.home ldap.server ldap.user ldap.password
* ADQuery catalina.home ldap.query [-v]
* LdapConnect catalina.home ldap.name ldap.server ldap.user ldap.password
* LdapConnect catalina.home ldap.name DISABLE
* ADSConnect catalina.home ads.server ads.password
* ADSReplicate catalina.home ads.remote.server ads.remote.password
* ADSReplication catalina.home ads.remote.index|list ads.remote.server|delete ads.remote.password
* ADSExport catalina.home file encrypted:{true|false}
* ADSImport catalina.home file
* DBConnect catalina.home db.type:{Derby|MySql|MSSQL|Oracle|PostgreSQL} db.server db.user db.password
* XTConnect catalina.home pam.server pam.user pam.password
* ConfigureRealms catalina.home auth.catalina.enable:true|false auth.ad.enable:true|false
* EnableSso catalina.home managed.path sso.enable:true|false
* GenerateSSL catalina.home
* CheckWebStatus catalina.home
* GenerateCASKeys catalina.home
* GenerateCASCipher catalina.home {SECRET|-}
* CheckAdsStatus catalina.home ads.check.timeout.ms
* GenerateCertificate catalina.home bundle.file.name
* ExportCertificate catalina.home bundle.file.name
* File PATH\_TO\_INSTRUCTION\_FILE
* Sign KEYSTORE\_PATH KEYSTORE\_PASSWORD FILE\_PATH
* Encrypt SECRET
* TestCertificate URL
* SSLPoke host port
* SSLImport host port [in-place:true|false]
* ADTest LDAP\_SERVER LDAP\_USER LDAP\_PASSWORD
* Extract export.file name|id master|-
* ListExport export.file match
* DecryptFile encrypted.file unencrypted.file
* PublishedVersion [product|qa] [proxy:port|-] [SSLv3|TLSv1|TLSv1.1|TLSv1.2]
* DBReleaseLockedAdmins catalina.home master|-
* DBListAdmins catalina.home master|-
* DBMakeAdmin catalina.home login master|-
* DBListUsers catalina.home master|-
* DBUnblock catalina.home user master|-
* EnableNonOpenMode catalina.home