Name create-auth-realm– adds the named authentication realm

Synopsis
```
create-auth-realm   --classname realm_class
[--help]
[ --property (name=value)[:name=value]*]
[ --target  target_name] auth_realm_name
```

Description The `create-auth-realm` subcommand adds the named authentication realm. This subcommand is supported in remote mode only.

Options

`--help`
Displays the help text for the subcommand.

`--target`
Do not specify this option. This option is retained for compatibility with other releases. If you specify this option, a syntax error does not occur. Instead, the subcommand runs successfully and the option is silently ignored.

`--classname`
Java class which implements this realm.

`--property`
Optional attribute name/value pairs for configuring the authentication realm. Authentication realms require provider-specific properties, which vary depending on what a particular implementation needs.

You can specify the following properties regardless of the type of realm you are creating: `FileRealm`, `CertificateRealm`, `JDBCRealm`, `LDAPRealm`, or `SolarisRealm`.

`jaas-context`
Specifies the JAAS (Java Authentication and Authorization Service) context.

`assign-groups`
(optional) If this property is set, its value is taken to be a comma-separated list of group names. All clients who present valid certificates are assigned membership to these groups for the purposes of authorization decisions in the web and EJB containers.

You can specify the following properties for an authentication realm of class `FileRealm`:

`file`
Specifies the file that stores user names, passwords, and group names. The default is *domain-dir*`/config/keyfile`.

You can specify the following properties for an authentication realm of class `CertificateRealm`:

`clientAuth`
If `true`, specifies that client authentication is required for all applications that use the certificate realm. The default is `false`.

To require client authentication for a specific web application, set the method of authentication in the `web.xml` file to `CLIENT-CERT`.

You can specify the following properties for an authentication realm of class `JDBCRealm`:

`datasource-jndi`
: Specifies the `jndi-name` of the `jdbc-resource` for the database.

`user-table`
: Specifies the name of the user table in the database.

`user-name-column`
: Specifies the name of the user name column in the database's user table.

`password-column`
: Specifies the name of the password column in the database's user table.

`group-table`
: Specifies the name of the group table in the database.

`group-table`
: Specify the group table for an authentication realm of class `JDBCRealm`.

`group-name-column`
: Specifies the name of the group name column in the database's group table.

`db-user`
: (optional) Allows you to specify the database user name in the realm instead of the `jdbc-connection-pool`. This prevents other applications from looking up the database, getting a connection, and browsing the user table. By default, the `jdbc-connection-pool` configuration is used.

`db-password`
: (optional) Allows you to specify the database password in the realm instead of the `jdbc-connection-pool`. This prevents other applications from looking up the database, getting a connection, and browsing the user table. By default, the `jdbc-connection-pool` configuration is used.

`group-table`
: Specifies the name of the group table in the database.

`digest-algorithm`
: (optional) Specifies the digest algorithm. The default is `MD5`. You can use any algorithm supported in the JDK, or none.

`encoding`
: (optional) Specifies the encoding. Allowed values are `Hex` and `Base64`. If digest-algorithm is specified, the default is `Hex`. If `digest-algorithm` is not specified, by default no encoding is specified.

`charset`
(optional) Specifies the `charset` for the digest algorithm.

You can specify the following properties for an authentication realm of class `LDAPRealm`:

`directory`
Specifies the LDAP URL to your server.

`base-dn`
Specifies the LDAP base DN for the location of user data. This base DN can be at any level above the user data, since a tree scope search is performed. The smaller the search tree, the better the performance.

`search-filter`
(optional) Specifies the search filter to use to find the user. The default is `uid=%s` (`%s` expands to the subject name).

`group-base-dn`
(optional) Specifies the base DN for the location of groups data. By default, it is same as the `base-dn`, but it can be tuned, if necessary.

`group-search-filter`
(optional) Specifies the search filter to find group memberships for the user. The default is uniquemember=%d (%d expands to the user elementDN).

`group-target`
(optional) Specifies the LDAP attribute name that contains group name entries. The default is `CN`.

`search-bind-dn`
(optional) Specifies an optional DN used to authenticate to the directory for performing the search-filter lookup. Only required for directories that do not allow anonymous search.

`search-bind-password`
(optional) Specifies the LDAP password for the DN given in `search-bind-dn`.

Operands *auth_realm_name* Name of this realm.

Examples **EXAMPLE 1** Creating a New Authentication Realm

```
asadmin> create-auth-realm
--classname com.sun.enterprise.security.auth.realm.file.FileRealm
--property file=${com.sun.aas.instanceRoot}/config/
admin-keyfile:jaas-context=fileRealm file
Command create-auth-realm executed successfully
```

Where `file` is the authentication realm created.

Exit Status

0	command executed successfully
1	error in executing the command

See Also `delete-auth-realm`(1), `list-auth-realms`(1)

`asadmin`(1M)