Encrypt login and password for database and LDAP

Summary: this page describes how to encrypt logins and passwords for the CAST dashboards/RestAPI:

1. when connecting to CAST Storage Service/PostgreSQL
2. when configuring LDAP authentication

Introduction

When configuring CAST dashboard / RestAPI connections to **CAST Storage Service/PostgreSQL** (i.e. Measurement or Dashboard Service schemas) or to an **LDAP server for corporate login mode**, logins and passwords are defined in the relevant configuration files in **clear text**. This therefore represents a potential security risk. If your organization requires these logins and passwords to be **encrypted**, you can use the following instructions to do so.

Note that this document already assumes that you have a working connection to your deployed CAST dashboard or RestAPI.

Encrypting access to CAST Storage Service/PostgreSQL

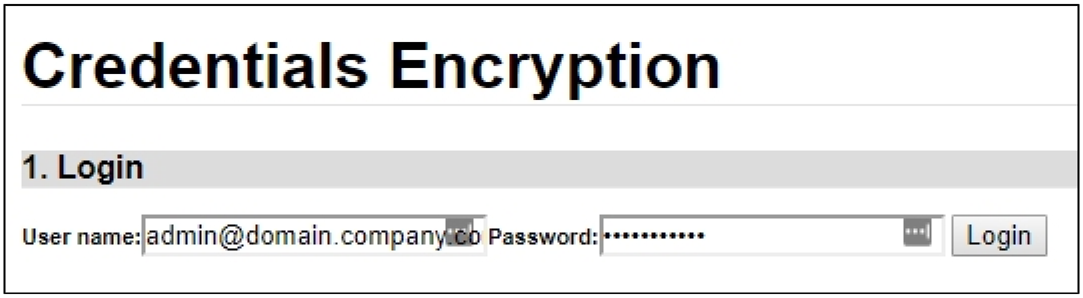
The ability to use encrypted CAST Storage Service/PostgreSQL credentials with WAR or ZIP files delivered in **CAST Dashboards 2.0** is currently not supported.

For CAST Dashboards 1.x, encrypted CAST Storage Service/PostgreSQL credentials are only supported for Dashboards deployed on **Apache Tomcat 8 or above**.

To encrypt the login and password that are defined when configuring access to the CAST Storage Service/PostgreSQL instance where your Measurement or Dashboard Service schemas are located, browse to the following **URL** to access the built in **login/password key generation** page:

```
http://<server>:[<port>]/<dashboard>/static/key.html
```

Login with a user (whether static list or Active Directory) that has the **ADMIN** role - by default no users have this role in either static list mode or in Active Directory mode - see **User authentication** for more information.

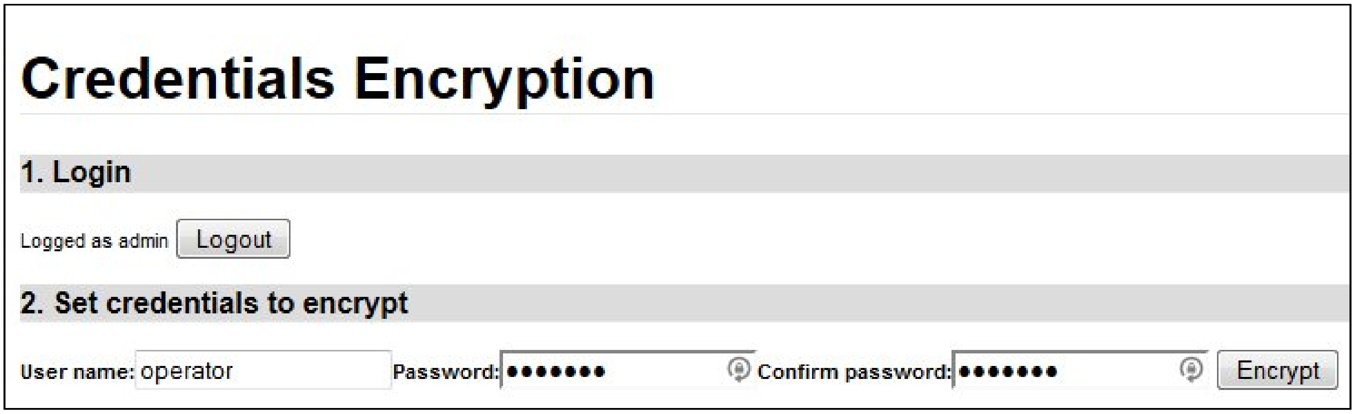


Credentials Encryption

1. Login

User name: admin@domain.company.co Password: ••••••••••••

When successfully authenticated, you now need to enter the **credentials (login and password)** for your target CAST Storage Service/PostgreSQL instance (that you would ordinarily enter into the **context.xml file** for configuring access to the Measurement or Dashboard Service) and that you wish to encrypt. In the example below, we have entered the default credentials for a CAST Storage Service/PostgreSQL instance:



Credentials Encryption

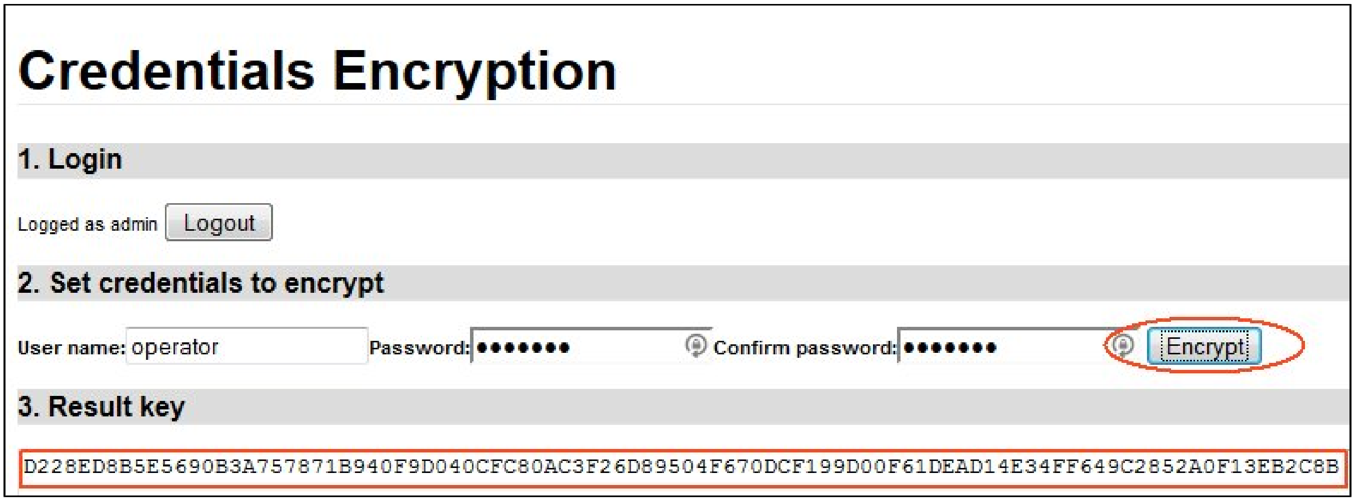
1. Login

Logged as admin

2. Set credentials to encrypt

User name: operator Password: ●●●●●●● Confirm password: ●●●●●●●

Now click the **Encrypt** button - CAST will then generate a key that relates to the credentials you entered:



Credentials Encryption

1. Login

Logged as admin

2. Set credentials to encrypt

User name: operator Password: ●●●●●●● Confirm password: ●●●●●●●

3. Result key

D228ED8B5E5690B3A757871B940F9D040CFC80AC3F26D89504F670DCF199D00F61DEAD14E34FF649C2852A0F13EB2C8B

You now need to copy this key to the clipboard or to a text file. To use the key in place of clear text database credentials, browse to the following file:

```
CATALINA_HOME\webapps\<dashboard>\META-INF\context.xml
```

Open this file with a text editor and scroll down to the location of a database access resource you have previously configured, for example:

```
<Resource name="jdbc/domains/AAD" url="jdbc:postgresql://localhost:2280/postgres"
    initConnectionSqls="SET search_path TO CAST_MEASURE;"
    username="operator" password="CastAIP"

    auth="Container" type="javax.sql.DataSource" driverClassName="org.postgresql.Driver"
    validationQuery="select 1"
    initialSize="5" maxActive="20" maxIdle="10" maxWait="-1"/>
```

Replace the line containing "**username**" and "**password**" with your generated key using the following syntax:

```
key="D228ED8B5E5690B3A75"
```

Add a new line directly underneath the line containing the "key" as follows - take note of the line that is specific to your release of CAST AIP and Apache Tomcat:

```
WARs delivered in CAST AIP  8.3.4 and all standalone CAST Dashboard Packages:

Tomcat  8 only: factory="com.castsoftware.adg.webservice.security.BasicDataSourceFactory"

WARs delivered in CAST AIP 8.3.0 - 8.3.3:

Tomcat 7: factory="com.castsoftware.adg.webservice.security.BasicDataSourceFactory"
Tomcat 8/8.5/9: factory="com.castsoftware.adg.webservice.security.BasicDataSourceFactory2"
```

Your database access resource should now look like this (this is an example for Tomcat 8 in CAST AIP 8.3.4 and all standalone CAST Dashboard Packages):

```
<Resource name="jdbc/domains/AAD" url="jdbc:postgresql://localhost:2280/postgres"
    initConnectionSqls="SET search_path TO CAST_MEASURE;"
    key="D228ED8B5E5690B3A75"
    factory="com.castsoftware.adg.webservice.security.BasicDataSourceFactory"

        auth="Container" type="javax.sql.DataSource" driverClassName="org.postgresql.Driver"
    validationQuery="select 1"
    initialSize="5" maxActive="20" maxIdle="10" maxWait="-1"/>
```

Save the file, **reload the cache** (see **Reload the cache**) and then reload your CAST dashboard / RestAPI and ensure you can login and view the data you need to.

ⓘ You may need to repeat the above for each database server resource you have configured in the **context.xml** file.

Encrypting access to an LDAP server

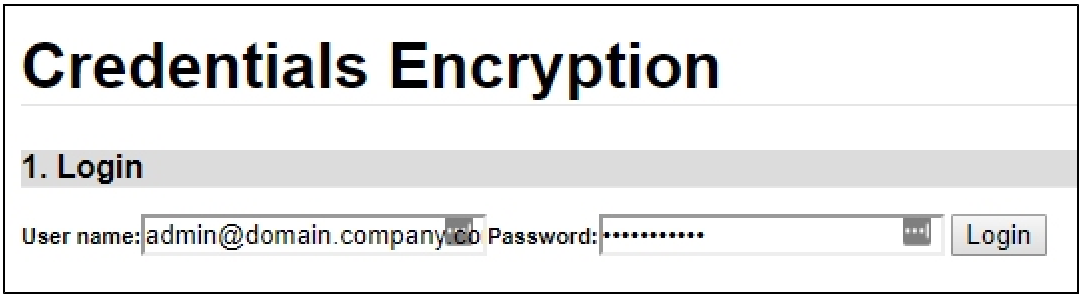
When configuring access to an LDAP server for authentication, an LDAP **service account login** and **password** must be specified in the **.properties** file in clear text as described in **User authentication**:

```
WAR 1.x
security.ldap.account.dn=cn=serviceaccount,dc=example,dc=com
security.ldap.account.password=password

WAR and ZIP  2.x
security.ldap.manager.dn=CN=serviceaccount,OU=RESOURCES,OU=FR,DC=example,DC=com
security.ldap.manager.password=password
```

To avoid the need to do this, browse to the following **URL** to access the built in **login/password key generation** page:

```
http://<server>:[<port>]/<dashboard>/static/key.html
```

Login with a user (whether Default Authentication or LDAP) that has the **ADMIN** role - by default no users have this role in either mode - see **User authentication** for more information:

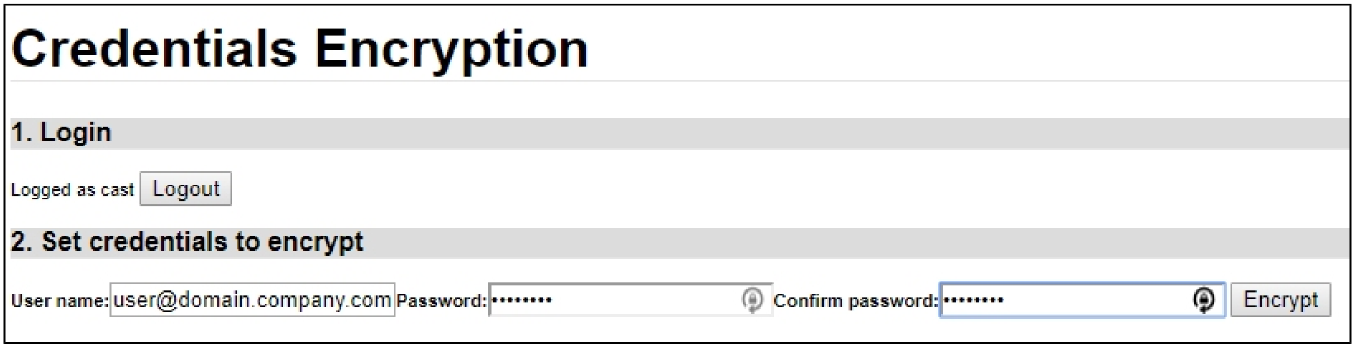


Credentials Encryption

1. Login

User name: admin@domain.company.co Password: ••••••••••••

When successfully authenticated, you now need to enter the **credentials (service account login and password)** for your LDAP server that you would ordinarily enter into the **.properties** file for configuring LDAP mode, and that you wish to encrypt. In the example below, we have entered the required LDAP credentials:



Credentials Encryption

1. Login

Logged as cast

2. Set credentials to encrypt

User name: user@domain.company.com Password: •••••••• Confirm password: ••••••••

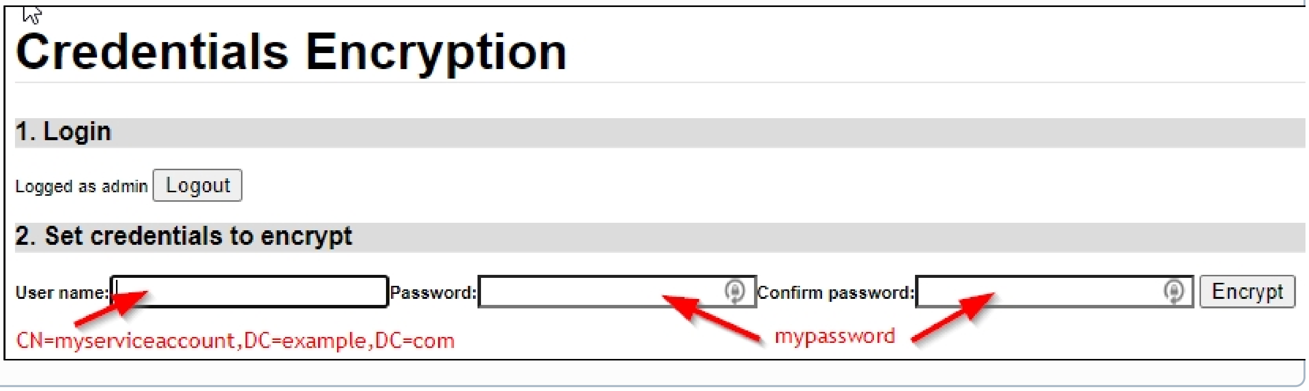
Note that the encryption key combines the values assigned to the following lines in the **.properties** file:

```
WAR 1.x
security.ldap.account.dn=cn=serviceaccount,dc=example,dc=com
security.ldap.account.password=password

WAR and ZIP  2.x
security.ldap.manager.dn=CN=serviceaccount,OU=RESOURCES,OU=FR,DC=example,DC=com
security.ldap.manager.password=password
```

Therefore, you must enter in the "**username**" and "**password**" fields in the encryption tool EXACTLY what is entered in the "**dn=**" and "**password=**" lines in the **.properties** file. For example, if the **.properties** file contains:

```
WAR 1.x
security.ldap.account.dn=CN=myserviceaccount,DC=example,DC=com
security.ldap.account.password=mypassword

WAR and ZIP  2.x
security.ldap.manager.dn=CN=myserviceaccount,DC=example,DC=com
security.ldap.manager.password=mypassword
```

...then you need to enter exactly the same in the following fields:



Credentials Encryption

1. Login

Logged as admin

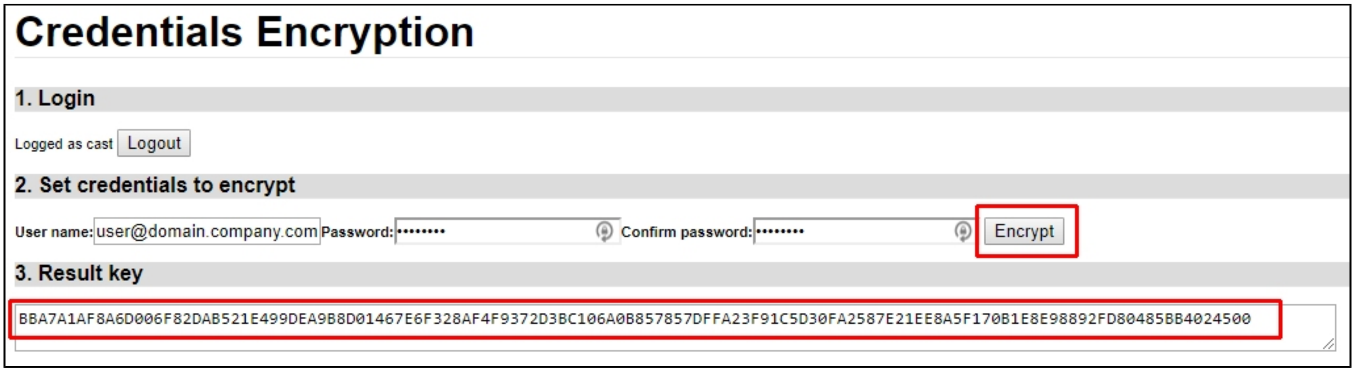
2. Set credentials to encrypt

User name: Password: Confirm password:

CN=myserviceaccount,DC=example,DC=com

mypassword

Now click the **Encrypt** button - CAST will then generate a key that relates to the credentials you entered:



Credentials Encryption

1. Login

Logged as cast

2. Set credentials to encrypt

User name: user@domain.company.com Password: •••••••• Confirm password: ••••••••

3. Result key

BBA7A1AF8A6D006F82DAB521E499DEA9B8D01467E6F328AF4F9372D3BC106A0B857857DFFA23F91C5D30FA2587E21EE8A5F170B1E8E98892FD80485BB4024500

You now need to copy this key to the clipboard or to a text file and then open the following file with a text editor:

```
WAR 1.x
CATALINA_HOME\webapps\<dashboard>\WEB-INF\security.properties

WAR  2.x
CATALINA_HOME\webapps\<dashboard>\WEB-INF\classes\application.properties

ZIP  2.x
<unpacked_zip>\application.properties
```

Locate the following configuration in the file:

```
WAR 1.x
# Parameters for ldap mode
# ------------------------
security.ldap.url=ldap://directory.example.com/
security.ldap.account.dn=cn=serviceaccount,dc=example,dc=com
security.ldap.account.password=password
security.ldap.account.key=
security.ldap.usersearch.base=dc=example,dc=com
security.ldap.usersearch.filter=(&(objectClass=user)(sAMAccountName={0}))
security.ldap.groupsearch.base=dc=example,dc=com
security.ldap.groupsearch.filter=(&(objectClass=group)(member={0}))

WAR and ZIP  2.x

## SPRING SECURITY LDAP CONFIG
# LDAP url, in the form ldap://HOST:PORT
security.ldap.url=ldap://directory.example.com/
# The ldap base where users and groups can be found
security.ldap.base=dc=example,dc=com
# The DN for accessing the LDAP repository
security.ldap.manager.dn=CN=serviceaccount,OU=RESOURCES,OU=FR,DC=example,DC=com
# The associated password. You can encrypt this using the aip encryption tool
security.ldap.manager.password=password
```

For CAST Dashboards 1.x

First remove the two lines with the `security.ldap.account.dn` and `security.ldap.account.password` parameters. Then enter the key generated previously into the line containing "**key**". This should give you the following:

```
# Parameters for ldap mode
# ------------------------
security.ldap.url=ldap://directory.example.com/
security.ldap.account.key=A9762B77F8A5B6C0A885BABD58DFA1438D77A51B94ECA09
security.ldap.usersearch.base=dc=example,dc=com
security.ldap.usersearch.filter=(&(objectClass=user)(sAMAccountName={0}))
security.ldap.groupsearch.base=dc=example,dc=com
security.ldap.groupsearch.filter=(&(objectClass=group)(member={0}))
```

Save the file, **restart the web application** and ensure you can login and view the data you need to.

For CAST Dashboards 2.x

Add a new line underneath `security.ldap.manager.password` called `security.ldap.manager.key` and enter the key generated previous into this new line. In a development deployment you do not need to remove the `security.ldap.manager.dn` or `security.ldap.manager.password entries` - if the `security.ldap.manager.key` is present it will be used. **However, you SHOULD remove both lines in a live production environment so that the DN and password are not present in clear text:**

```
## SPRING SECURITY LDAP CONFIG
# LDAP url, in the form ldap://HOST:PORT
security.ldap.url=ldap://directory.example.com/
# The ldap base where users and groups can be found
security.ldap.base=dc=example,dc=com
# The DN for accessing the LDAP repository
security.ldap.manager.dn=CN=serviceaccount,OU=RESOURCES,OU=FR,DC=example,DC=com
# The associated password. You can encrypt this using the aip encryption tool
security.ldap.manager.password=password
security.ldap.manager.key=A9762B77F8A5B6C0A885BABD58DFA1438D77A51B94ECA09
```

Save the file, **restart the web application** and ensure you can login and view the data you need to.

What happens if the LDAP credentials change (new password)?

If your LDAP credentials change, for example a new password is generated on the LDAP server, then access to the the CAST Dashboard for any LDAP user will fail. As such the encryption key for the new credentials will need to be regenerated in the **key.html** page, however, this page requires authentication therefore it will not be accessible in order to generate a new key. This can only be resolved by:

- temporarily restoring access using a **login** and **password**, i.e. removing the `security.ldap.account.key / security.ldap.manager.key` line from the **.properties** file and (for 1.x WAR files only) re-adding the `security.ldap.account.dn` and `security.ldap.account.password` lines.
- accessing **key.html** and encrypting the new login/password into a key.
- re-adding the `security.ldap.account.key / security.ldap.manager.key` line with the new key and (for 1.x WAR files only) removing the `security.ldap.account.dn` and `security.ldap.account.password` lines.