Copying a production database to a test system (sadmin login)

This page describes what to do when the database of a production system is copied into an existing test or integration system (called "test system" below), and you then want to log in as the system administrator sadmin again. It applies to Webdesk 4.56 and later and is meant for system administrators and consultants. For background on the configuration file and the key, see Configuration file webdesk.properties and environment variables[1].

Why the login fails after the copy

Webdesk encrypts passwords stored in the database with a key (AES-256). This includes the password of sadmin: it is not an ordinary user but a system parameter stored in encrypted form. On every login, Webdesk decrypts the stored value and compares it with the password entered.

The key, however, is not stored in the database but with the installation: in the entry webdesk.encryption.key of the file webdesk.properties, or in the file enc.key. The test system has its own key. After the copy, the database contains values encrypted with the production system's key, but the test system still decrypts them with its own key:

```
Production system: database (encrypted with key P) + webdesk.properties with key P
Test system:       copied database (key P)          + webdesk.properties with key T
                   # decrypting with T fails
```

Consequences:

- Logging in as sadmin fails, even with the correct (production) password. The log contains a message with javax.crypto.AEADBadTagException: Tag mismatch.
- All other passwords stored in encrypted form in the production database cannot be read on the test system either, for example those of connectors, document management repositories or groupware access.

The solution is to give the test system the production system's key.

Before the first start: mind the production configuration

Attention: With the production database, the test system also takes over the production system's configuration, for example time and attendance connections, connectors, e-mail sending and scheduled jobs. As soon as the passwords can be decrypted again, the test system can access production systems, write data or send e-mails. Check and deactivate or change these settings before you put the test system into operation with the production key.

Note: With the production key, the test system can read all encrypted production data. From then on, protect the test system's configuration as carefully as the production system's.

Login through a reverse proxy (SSO via HTTP header)

Many installations log users in through a reverse proxy that passes the username to Webdesk in an HTTP header (system parameter AuthenticationOptions.ssoHeaderNameRemoteUser, default X-Remote-User). Webdesk only accepts the header for requests whose URL starts with the system parameter AuthenticationOptions.ssoHeaderAuthenticationValidationUrl, for example https://<prod-host>/webdesk/.

These system parameters are stored in the database and are copied along. On the test system, the production system's validation URL then does not match the test system's address. Result: on every request carrying the header, the automatic login fails with a java.lang.NullPointerException in AuthenticationUtils.getNormalizedUser, each time with a full stack trace in the log and in catalina.out. The log files grow quickly until the disk is full and the server fails. A lower log level does not prevent this.

Therefore change the validation URL to the test system's address before the test system's first start. Since logging in is not possible at that point, do this directly in the test system's database, with Webdesk stopped and after backing up the tables PoBeanProperty and PoBeanPropertyValue:

```
SELECT p.propertyName, v.property
FROM PoBeanProperty p JOIN PoBeanPropertyValue v ON v.BEAN_UID = p.BEAN_UID
WHERE p.beanName = 'AuthenticationOptions'
  AND p.propertyName IN ('authenticationMode', 'ssoHeaderNameRemoteUser',
 'ssoHeaderAuthenticationValidationUrl');

UPDATE PoBeanPropertyValue SET property = 'https://<test-host>/webdesk/'
WHERE BEAN_UID = (SELECT BEAN_UID FROM PoBeanProperty
                  WHERE beanName = 'AuthenticationOptions'
                    AND propertyName = 'ssoHeaderAuthenticationValidationUrl');
```

- Enter the address as Webdesk sees the requests, in the same form as on the production system (protocol, host name, path).
- If the test system should not use SSO, set authenticationMode to DB instead.
- An entry in webdesk.properties is not enough: at startup, the values from the database take precedence.
- Once logging in works again, the value can be maintained in the "System parameters" screen.

If the test system has already been started with the wrong validation URL: stop Webdesk, have the reverse proxy temporarily stop sending the header (or access Webdesk directly without the proxy), remove log files to free up space, and then change the validation URL as described above.

Procedure

1. Stop Webdesk on the test system (on all nodes if there are several).
2. Back up the test system's webdesk.properties (and enc.key, if present).
3. **Find the production system's key.** In the production system's webdesk.properties, the entry webdesk.encryption.key.source tells you where the key is:
   - webdesk_properties: the key is the value of webdesk.encryption.key in the same file.
   - file: the key is in the file enc.key next to webdesk.properties.

   Transfer the key over a secure channel, not by e-mail or in a ticket.
4. **Enter the key on the test system**, in one of two ways:

   ```
   webdesk.encryption.key.source = webdesk_properties
   webdesk.encryption.key = <key of the production system>
   ```

   or replace enc.key with the production system's file and set webdesk.encryption.key.source = file (the entry webdesk.encryption.key is then not used). With several nodes, do this on every node.
5. **Re-encrypt the database credentials.** The username and password of the test database in webdesk.properties are encrypted with the test system's previous key and no longer match after the key change. Without this step, the test system can no longer connect to its database. You only need this step if webdesk.credentials.encrypted = true is set; with false (plain text), or with the old entry webdesk.connection.encrypted, the values do not depend on the key. Two ways:
   - **Let Webdesk encrypt them at startup (recommended):** enter the usernames and passwords of the Webdesk and Shark databases in plain text and **delete** the lines webdesk.credentials.encrypted and shark.credentials.encrypted (do not set them to false):

     ```
     webdesk.connection.username = <database user>
     webdesk.connection.password = <database password>
     shark.connection.username = <database user>
     shark.connection.password = <database password>
     ```

At the next start, Webdesk encrypts the values with the new key, adds the entries ...credentials.encrypted = true again and saves the file. The user Webdesk runs as needs write access to the file. With false, Webdesk would leave the values as intentional plain text.

- **With the tool:** on Windows installations, start manageWebdeskDBProps.bat in Tomcat's bin folder, choose "Basic" mode and use the menu option for entering new values. The tool encrypts the values entered with the key from the file. Do not use the "Expert" mode parameters webdeskLogin= / webdeskPassword= for this: on files that are already encrypted, they are saved unencrypted. The tool only supports webdesk.encryption.key.source = webdesk_properties.

6. **With SSO through a reverse proxy:** change the validation URL ssoHeaderAuthenticationValidationUrl in the database to the test system's address (see section "Login through a reverse proxy").
7. Start Webdesk. Check that the system starts and that the credentials in webdesk.properties are encrypted again (as Base64 text).
8. Log in as sadmin with the production system's password.
9. Back up the test system's changed webdesk.properties (or enc.key) again.

Troubleshooting

- **The log fills up with java.lang.NullPointerException in AuthenticationUtils.getNormalizedUser until the disk is full:** the validation URL for login via the reverse proxy header is still the production system's (see section "Login through a reverse proxy").
- **The test system does not start after the key change, the database connection fails:** the database credentials are still encrypted with the test system's old key (step 5).
- **Logging in as sadmin still fails, the log contains AEADBadTagException:** the test system does not use the production system's key. Check the key source and the key or enc.key on all nodes.
- **After the start, the credentials are still in plain text in the file:** Webdesk could not write the file (write access), or the lines ...credentials.encrypted are still in the file with false.
- **The production system's key can no longer be found:** the encrypted values from the production database, including the sadmin password, then cannot be recovered. Contact Webdesk support.

1. /daisy/personalwolke-admin/11598-dsy.html?language=4