OCS INSTALLATION GUIDE

1. Application database preparation

The Online Collection Software (OCS) uses a relational database for application configuration and collected data storage. The database engine is accessed by the Java Database Connectivity standard protocol.

OCS supports and provides scripts for the Oracle Database Engine and MySQL (see the following sections).

Please note that the SQL scripts containing schema definitions and application data do not contain any database creation statements - the **database must be created before executing the script**, using the native tools of the underlying engine.

OCS requires the database to use *UTF-8* as a character encoding.

1.1. Oracle Database 11g

To create the OCS database for Oracle, the script `oct-oracle-schema-create.sql` must be executed. It creates the database schema including tables, sequences and indexes, and populates the tables with initial data.

In order to drop the OCS database, the following script might be helpful: `oct-oracle-schema-drop.sql`

OCS supports Oracle Database 11g. Although the system might also work on previous versions of the Oracle Database engine, this has neither been tested, nor can it be supported by the supplier of the system.

1.2 MySQL 5.5+

To create the OCS database for MySQL, the script `oct-mysql-schema-create.sql` must be executed. It creates the database schema including tables, sequences and indexes, and populates the tables with initial data.

In order to drop the OCS database, the following script might be helpful: `oct-mysql-schema-drop.sql`

OCS supports MySQL 5.5 and greater. Although the system might also work on previous versions of the MySQL engine, this has neither been tested, nor can it be supported by the supplier of the system.

2. System initialisation

In order to configure the system, the application database must be populated with a small number of environment specific settings, which are described in the following parts of this section.

2.1. Application file storage

The application uses a server side file system to output application artefacts, such as exported collected data. The server side file storage should be capable of accommodating exported data packages up to a recommended capacity of 10 GB.

The **absolute path to the file system storage** on the application server must be inserted into the only row of the `OCT_SYSTEM_PREFS` table, column `FILE_STORE`.

The value can simply be inserted into the row that has been created by the database initialisation script.

Please note that **only one row** is allowed in the `OCT_SYSTEM_PREFS` table.

2.2. Security tool installation

The Security tool is a standalone application responsible for decrypting collected data. It is also required for the web application authentication process, as it provides the functionality for decrypting the authentication challenge phrase which is asked in the login screen.

In order to install the application, a distribution archive needs to be unpacked into the file system. The Security tool is a Java application and thus platform-independent. However the package provides operating system specific distributions, which contain a convenient launcher script for the underlying operating system. Only one of these distributions needs to be unpacked.

The shipped distributions support Windows or Linux and are named `oct-crypto-win32.zip` or `oct-crypto-linux.zip`, respectively. The launcher script is located in the `bin` directory and named either `launcher.bat` or `launcher.sh`, depending on the platform.

2.2.1. *Security* tool initialisation

When the application is run for the first time, the only available menu option is *Initialize*. Upon selection, the application prompts the user to create a **security tool password**. This password will be used to protect both the private key within the Security tool and the password for the web application administrator account. It will also be used to access the Security tool once it is initialised.

This functionality, available only when you access the tool for the first time, allows to define a password to further access the Security Tool (security tool password) and to generate the Security Keys for encryption and decryption of statements of support collected via the online collection software.

After entering and creating a password protection henceforth the access to the Security Tool a folder is automatically created inside the OCS Security Tool installation folder called "**data"**. This folder can be accessed directly anytime via the Security Tool main menu functionality "Credentials Folder".

The **data** folder will at first contain the following subfolders and files:

- Files to Keep
    - private.key (contains the encrypted private key)
    - crypto.salt (the file that is used for decrypting the private key)
- Files to Send
    - public.key (contains the public key in a hexa-decimal form)

Moreover, the user credential files (defining credentials to access the admin interface of the online collection software) which have to be subsequently created by the user (see section 2.2.2. below for details) will be also automatically stored in the "Files to send" subfolder.

Notes:

It falls within the responsibility of the administrator to not forget the security tool password entered at this phase or delete/lose the data folder. These data are stored with a very strong encryption mechanism making it impossible to provide a password recovery or a password reset mechanism. As such it is mandatory for these files to be kept safely.

The data folder is portable. As such, if you deleted by mistake the Security Tool but you have a back-up copy of the Files to keep folder, all you need to do is to reinstall the Security Tool and to copy in the data folder the backed-up "Files to keep" sub folder. The Security Tool will recognise those data and consider the tool initialised with the password used when the original data folder was created.

The security tool password used for initializing the Security Tool will be used to access the Security Tool once it is initialized.

The Public key as stored on the "Files to send" subfolder must be inserted in the Online Collection Software prior to its deployment.

Attention, in case you use a live DVD, your data files will not be saved in your computer after the end of your session, so you have to keep a copy thereof on a USB drive. You will also have to copy the content of the "Files to keep" subfolder back into your computer each time you use the decryption functionalities.

The Security tool initialisation output window presents the newly generated public key within the text area labelled *Public key*.

The value in this text area must be inserted into the single row of the `OCT_SYSTEM_PREFS` table, column `PUBLICKEY`.

Note: One must pay attention to insert the value into the already existing row, as **only one row** is allowed in the `OCT_SYSTEM_PREFS` table.

Web account login and password

2.2.2. Set up of the System Credentials

In order to set up and further manage credentials, allowing to further configure and access the Admin interface of your online collection system, you should use the functionality "System Credentials" in the main menu of the Security Tool.

This functionality allows to create a set of credentials for each administrator of the online collection system.

- In order to set up credentials for a new user:
  - Define a username
  - Define a password corresponding to this user (must be different from the Security Tool password)

Once the set of credentials is successfully defined the following confirmation message will appear on the screen: "Your System Credentials have been successfully set up".

You can repeat this operation in order to create several sets of system credentials, for several administrators.

Each set of credentials will be stored in a system credential file with a name including the username of the administrator concerned (following a model : username_system_credentials.wac).

The respective system credential files will be created in the "Files to Send" subfolder of your Data folder which can be accessed directly anytime via the Security Tool main menu functionality "Credentials Folder".

The actual configuration of the access rights to your online collection system, will only take place after you have updated the database table OCT_ACCOUNT.

The "System Credentials" functionality can be used subsequently to modify the System Credentials allowing to access the Admin Interface of your online collection system, by adding Credentials for a new user or by modifying password for an existing user.

- If you want to set up Credentials for a new user without changing the existing one(s):
  - Define a new username
  - Define a password corresponding to this user (must be different from the Security Tool password)

  An additional file will be created in the Credentials Folders.

- If you want to modify password for an existing user:
  - Define the already existing username
  - Define a new password (must be different from the Security Tool password)

  A modified file will be created in the Credential Folders (overwriting the previous one)

The actual change in the access rights of your online collection system, will only take place after you have updated the database table OCT_ACCOUNT.

In case you wish to remove an existing user, you can directly delete the corresponding row from the OCT_ACCOUNT table.

Notes:

username_system_credentials.wac file content example:

admin::3875034eab17855bac03a3cc9e107b1d28a9b4c33355885cd25b4e70b55b

- The value ‘admin’ must be inserted in the USERNAME column of the OCT_ACCOUNT table.

- The value ‘3875034eab17855bac03a3cc9e107b1d28a9b4c33355885cd25b4e70b55b’ must be inserted in the PASSHASH column of the OCT_ACCOUNT table.

Attention, In case you use a live DVD, the content of your Credentials Folder as created or modified by the Security Tool will not be stored in your computer after the end of your session, so you have to make a copy thereof on a USB drive.

3. Application deployment

The OCS application is fully compliant with Java Enterprise Edition (JEE) version 5. It can be hosted on any middleware software providing the following services: Servlet 2.5, JMS and Java Persistence API 2.0.

The application requires JEE resources to be configured within the application server, which are discussed in detail in the following sections.

3.1. Configuration of JEE resources

The following list of resources needs to be created within the application server. For a system running on Oracle Weblogic Application Server or GlassFish, one can take advantage of the scripts provided alongside the application, which are discussed in the next subsections. Otherwise the required resources will have to be created manually.

Java Messaging System

The following Java Messaging System (JMS) items need to be created:

Item type	JNDI name
JMS connection factory	`OctExportQueueConnectionFactory`
JMS Queue	`OctExportQueue`
JMS Queue	`OctExportDispatcherQueue`

The JMS queues are being used for an asynchronous export feature. In order to optimise system performance, one should configure them for 20 threads.

Transaction support (XA) must be enabled for these resources.

Data Source

The following data source needs to be created:

Item type	JNDI name
Data Source	`jdbc/oct`

The data source must be set up with all required parameters for connecting to the database already configured in section “Application database preparation”. These include, in general, the name **and port of the database server**, the **name of the database**, as well as a **user name and password** for connecting to the database.

The data source must support transactions (XA).

Please note that the application server might also need a suitable JDBC driver to be installed, in order to connect to the selected database engine. Whether this is necessary or not depends on the actual combination of application server and database engine. Please refer to the technical manuals of these products, in order to determine if and how a JDBC driver needs to be installed, and where to obtain it.

Configuration on Oracle Weblogic Application Server 10.3.4+

Oracle Weblogic uses an API called WLST for administering system resources of the application server. WLST is implemented as a Python environment running on a Java platform. OCS provides a Python script which can be executed within the WLST engine, and which initialises all required JEE resources.

The name of the script is `oct-weblogic-10.3.4.py`.

The procedure for creating a Weblogic server configuration is as follows:

- customise the Python script by providing valid values for the following variables describing the configuration of the Weblogic instance: `SERVER_HOST`, `SERVER_PORT`, `WL_ADMIN_USER`, `WL_ADMIN_PASS`, `WL_INSTANCE` and the underlying database engine: `DB_DIALECT`, `DB_HOST`, `DB_PORT`, `DB_NAME`, `DB_USER`, `DB_PASS`

- launch WLST by executing `WL_HOME/common/bin/wlst.cmd` or `.sh`

- execute the script within the WLST console:
    - `execfile('<PATH_TO_SCRIPT>')`
- exit the WLST console:
    - `exit()`

See also the Oracle Weblogic documentation for more information on WLST.

The supported Oracle Weblogic platform is 10.3.4 or later. The configuration script is not guaranteed to work on any previous version of the Oracle Weblogic application server.

Configuration on GlassFish Open Source Edition 4

The GlassFish application server provides various ways of configuring JEE resources, one of which are XML configuration files which can be loaded by a server administration tool. This approach seems simple and convenient, and was therefore chosen for the OCS project.

OCS provides two separate XML configuration files for the initialisation of required JEE resources:

- oct-glassfish-3.1.1-mysql.xml
- oct-glassfish-3.1.1-oracle.xml

For a database engine among the ones above, the corresponding configuration file needs to be customised and loaded into the GlassFish server. The detailed configuration procedure is as follows: customise the selected XML file by replacing all placeholders (`$DB_USER`, `$DB_PASSWORD`, `$DB_HOST`, `$DB_PORT`, `$DB_NAME`) in the *** CUSTOMISE *** section of this file with the actual values

Launch the GlassFish administration tool with the option to create resources based on the selected XML file:

```
GLASSFISH_HOME/bin/asadmin
 -H GLASSFISH_HOST -p GLASSFISH_PORT
-u GLASSFISH_ADMIN_USER
add-resources <PATH_TO_XML_FILE>
```

- where: GLASSFISH_HOME – GlassFish installation directory,
- GLASSFISH_HOST – GlassFish host name (usually localhost),
- GLASSFISH_PORT – GlassFish port number,
- GLASSFISH_ADMIN_USER – GlassFish administrator account name

The supported GlassFish platform is 4.1 (not 4.1.1). OCS and the configuration files are not guaranteed to work on any other version of the GlassFish application server.

3.2. Deployment of the Online Collection Software application

Once all application prerequisites are in place, including the database preparation, Security initialisation and configuration of JEE resources, the OCS application can be deployed on the application server.

The OCS application is bundled as a single Enterprise Application Archive: `oct-ear.ear`.

The application archive must be installed on the application server using one of the facilities provided by the middleware. The most common ways of deployment are through an administration console, by copying the application file to an auto-deploy directory of the server, or by using a command-line tool from the application server distribution.

Annex: List of requirements for using the software

- J2EE5 compliant application server
- Relational database, SQL 99 compliant
- File system
- Last version of Java 1.7 to run the OCS Security tool and the web application