

<p style="text-align: center;">INSTALLATION GUIDE</p> <p style="text-align: center;">Online Collection System for European Citizens' Initiatives</p>
--

1.	Application database preparation.....	2
1.1.	Oracle Database 11g	2
1.2.	MySQL 5.5+	3
2.	System initialisation	4
2.1.	Application file storage.....	4
2.2.	Cryptography tool installation	4
2.2.1.	Run-Time Performance	7
3.	Application deployment.....	9
3.1.	Prerequisites	9
3.2.	Configuration of JEE resources	9
3.3.	Deployment of the Online Collection System application.....	13
3.4.	Launching the Online Collection System application	13
	Glossary	15

1. Application database preparation

The Online Collection System (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). The shipped distributions support the Oracle Database Engine and MySQL and are respectively named `oct-db-oracle.zip` and `oct-db-mysql.zip`.

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 may be helpful: `oct-oracle-schema-drop.sql`

OCS supports Oracle Database 11g and higher¹. Although the system may also work on previous versions of the Oracle Database engine, this has not been tested and cannot be supported by the supplier of the system.

¹ At the time of writing (March 2012), the most recent stable release of the Oracle Database is version 11g Release 2.

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 may be helpful: `oct-mysql-schema-drop.sql`

OCS supports MySQL 5.5 and higher². Although the system may also work on previous versions of the MySQL engine, this has not been tested and cannot be supported by the supplier of the system.

² At the time of writing (March 2012), the most recent stable release of MySQL is version 5.5.21 .

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 objects, such as exported collected data. The server side file storage should be capable of accommodating exported data packages up to a minimum 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. Cryptography tool installation

The Cryptography 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 CAPTCHA phrase asked for in the login screen.

In order to install the application, a distribution archive needs to be unpacked into the file system. The Cryptography 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 and Linux and are respectively named `oct-crypto-win32.zip` and `oct-crypto-linux.zip`. The launcher script is located in the `bin` sub-directory and named `launcher.bat` for Windows and `launcher.sh` for Linux.

OCS Release Notes
Author: Kenneth Irvine
Publishing Date: 5 April 2012
Current Version at Publishing Date: 1.2.1

If for any reason the launcher script doesn't work, you can manually start the Cryptography tool from the command line, using the command shown below:

```
java -Xms256m -Xmx256m -jar ../../lib/oct-offline-1.0.0.jar
```

N.B.

- The above command should be run from the Cryptography tool's bin sub-directory.
- Running the above command also assumes that Java is accessible from the Cryptography tool's bin sub-directory.

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 **master password**. This password will be used to protect both the private key within the Cryptography tool and the password for the web application administrator account. It will also be used to access the Cryptography tool once it is initialised.

After entering and submitting the password in the Cryptography tool, the user is presented with a **public key** and a **hashed password** information. A new sub-directory called **data** is also created inside the cryptography tool's installation directory. This sub-directory contains three files:

- crypto.key – contains the encrypted private key

- oct.key – contains the public key in a hexadecimal form (the same as the one presented in the UI)

- crypto.salt – the file used for decrypting the private key

It falls within the responsibility of the administrator to remember the password entered at this phase and to avoid deleting/losing the data sub-directory. This data is 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.

Note: The data sub-directory is portable. As such, if you delete the Cryptography tool by mistake but you have a backup copy of the data sub-directory, all you need to do is to reinstall the Cryptography tool and to copy the backed-up data sub-directory into the installation directory. The Cryptography tool will recognise this data and consider the tool to be initialised with the password used when the original data sub-directory was created.

The password used for initializing the Cryptography tool will be used to access the Cryptography tool once it is initialised.

Public key must be inserted in the Online Collection System prior to deployment

The Cryptography 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, in the column PUBLICKEY.

Note: You have to make sure 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

The second value presented on the Cryptography tool initialisation output window is the *hashed password*.

This value is used for setting up the password of the initial administrator account in the web system. The account entries are stored in the database table OCT_ACCOUNT. The user must create a row for the initial account, where the USERNAME column value represents the account name (login), and the value for the PASSHASH column is the hashed password copied³ from the output window of the Cryptography tool. If the administrator password subsequently needs to be changed – or if the admin would simply like to use a different online-access password from the password used to access the Cryptography tool – this can be done via the Password hashing functionality of the Cryptography tool. Please note that this functionality does not affect the password set for the Cryptography tool itself.

In order to obtain a new *hashed password*, start the Cryptography tool and navigate to the Menu -> Hash password. This will open a simple window with one input field for the desired value and when clicking on the *hash password* button, the *hashed password* field will be filled with the corresponding *hashed password*.

³ You have to manually copy the hashed password. e.g. with copy & paste.

2.2.1. Run-Time Performance

Run-time performance can be greatly influenced by the choice of JVM and the characteristics of the CPU in use. On the same machine, the decryption time can be greatly reduced by using a different JVM vendor/version. The CPU frequency and the number of cores also considerably affect performance. The higher CPU frequency and the number of cores, the better will be the performance.

The Cryptography tool is designed to use all the available CPU cores. However – as the decryption process may take some time to complete – the decryption process runs with a lower CPU priority so that the computer remains usable for other tasks.

OCS Release Notes
 Author: Kenneth Irvine
 Publishing Date: 5 April 2012
 Current Version at Publishing Date: 1.2.1

As a guide to choosing your optimal configuration, benchmarking the decryption of a sample of 165000 signatures gave the following results:

JAVA 32 bit JVM SUN (1.6.0 U31)	
Core2 Quad Q9550 @ 2.83Ghz / Win7 64Bit	3h:20min
AMD 4450B X2 2.3 Ghz / Win XP 32 bit	9h:56min
i7 2600, 3.4 Ghz / Win7 64Bit	2h:20min
i5 2500k 3.4Ghz / Win7 64 bit	2h:25min
Notebook Core 2 Duo T8300, 2.4 Ghz / Win7 32Bit	10h:15min

JAVA 32 bit JVM BEA JRockit (1.6.0 U29 R28)	
Core2 Quad Q9550 @ 2.83Ghz / Win7 64Bit	1h:44min
AMD 4450B X2 2.3 Ghz / Win XP 32 bit	4h:32min
i7 2600, 3.4 Ghz / Win7 64Bit	1h:05min
i5 2500k 3.4Ghz / Win7 64 bit	1h:16min
Notebook Core 2 Duo T8300, 2.4 Ghz / Win7 32Bit	4h:49min

JAVA 64 bit JVM BEA JRockit (1.6.0 U29 R28)	
Core2 Quad Q9550 @ 2.83Ghz / Win7 64Bit	0h:56min
AMD 4450B X2 2.3 Ghz / Win XP 32 bit	n/a
i7 2600, 3.4 Ghz / Win7 64Bit	0h:33min
i5 2500k 3.4Ghz / Win7 64 bit	0h:37min
Notebook Core 2 Duo T8300, 2.4 Ghz / Win7 32Bit	n/a

JAVA 64 bit JVM SUN (1.6.0 U25)	
Core2 Quad Q9550 @ 2.83Ghz / Win7 64Bit	0h:55min
AMD 4450B X2 2.3 Ghz / Win XP 32 bit	n/a
i7 2600, 3.4 Ghz / Win7 64Bit	0h:33min
i5 2500k 3.4Ghz / Win7 64 bit	0h:37min
Notebook Core 2 Duo T8300, 2.4 Ghz / Win7 32Bit	n/a

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: Enterprise Java Beans 3.0, 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. Prerequisites

Other requirements for the successful deployment of OCS are:

- JEE5⁴ compliant application server
- Relational database, SQL 99 compliant
- File system
- Java 1.5⁵ JDK or higher⁶ to run the OCS Cryptography Tool

3.2. 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 on GlassFish, you can take advantage of the scripts provided with the shipped distribution in `oct-app-server-scripts.zip`, and which are discussed in the following sub-sections. Otherwise, the required resources will have to be created manually.

⁴ Although JEE6 already exists, JEE5 is used for reasons of compatibility with EJB 3.

⁵ Both version numbers 1.5. and 5 are used to identify this release of JDK. Version 5 is the *product version*, and version 1.5 is the *developer version*.

⁶ At the time of writing (March 2012), the most recent stable release of JDK is version 7.

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 1 "Application database preparation". In general these parameters include 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.

Transaction support (XA) must be enabled for the data source.

Please note that the application server may 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 in use. 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 configuration variables for:
 - the the WebLogic instance:
 - `SERVER_HOST`
 - `SERVER_PORT`
 - `WL_ADMIN_USER`
 - `WL_ADMIN_PASS`
 - `WL_INSTANCE`
 - 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` on Windows
 - `WL_HOME/common/bin/wlst.sh` on Linux
- 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 higher⁷. The configuration script is not guaranteed to work on any previous version of the Oracle WebLogic application server.

Configuration on GlassFish Open Source Edition 3

The GlassFish application server provides various ways of configuring JEE resources, one of which is XML configuration files which can be loaded by a server administration tool. This approach is 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 MySQL database engine
- oct-glassfish-3.1.1-oracle.xml Oracle database engine

For the selected database engine 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 the file with their 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),

⁷ At the time of writing (March 2012), the most recent stable release of WebLogic Server is version 12.1.1 .

- GLASSFISH_PORT – GlassFish port number,
- GLASSFISH_ADMIN_USER – GlassFish administrator account name

The supported GlassFish platform is 3.1.1 or higher⁸. The configuration files are not guaranteed to work on any previous version of the GlassFish application server.

3.3. Deployment of the Online Collection System application

When all application prerequisites are in place – including the database preparation, cryptography 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.

3.4. Launching the Online Collection System application

The OCS application is launched as shown in the table below:

User Type	OCS URL
Public interface	<i>http(s)://appserver:port/oct-web-public</i>
Administration interface	<i>http(s)://appserver:port/oct-web-admin</i>

Where:

- Text to be replaced in the OCS URL is shown in *italics*.
- Explicit text is shown in **bold**.
- The application protocol will be *http* or *https* in accordance with the Organiser's requirements.
- *appserver:port*: The Organiser will request the application server and port from their local system administrator.

⁸ At the time of writing (March 2012), the most recent stable release of GlassFish Server is version 3.1.2 .

- **oct-web-public** is the explicit string to indicate the public access of OCS.
- **oct-web-admin** is the explicit string to indicate the administrator access of OCS.

N.B.

- For legacy compatibility reasons the explicit access strings refer to *oct* rather than *ocs*.
- The actual domain names chosen by the Organiser to be used for the Citizens' initiative will be mapped by the local system administrator to the public and admin URLs.
- The OCS user manual therefore doesn't need to document the OCS launch procedure as the domain names to be used will be different for each Citizens' initiative, and these domain names will be publicised by the Organiser of that Citizens' initiative.
- The Organiser is the intended admin user.
- A signatory of a Citizens' initiative is the intended user of the public interface.

Glossary

Term	Description (EN)
API	Application Programming Interface Used by software components as an interface to communicate with each other.
CPU	Central Processing Unit The portion of a computer system that carries out the instructions of a computer program.
EAR	Enterprise Archive A file format used by Java EE for packaging one or more modules into a single archive so that the deployment of the various modules onto an application server happens simultaneously and coherently.
ECI	European Citizens' Initiative
EJB	Enterprise JavaBeans A Java API for a managed, server-side component architecture for the modular construction of enterprise applications.
HTTP	Hypertext Transfer Protocol An application protocol for distributed, collaborative, hypermedia information systems.
HTTPS	Hypertext Transfer Protocol Secure A combination of HTTP with the SSL/TLS protocol. It provides encrypted communication and secure identification of a network web server.
JDBC	Java Database Connectivity Java API that defines how a client may access a database.
JDK	Java Development Kit This by far the most widely used Java Software Development Kit.
JEE5	Java Enterprise Edition, version 5
JMS	Java Messaging System Java API for a Java Message Oriented Middleware, used for sending messages between two or more clients.
JNDI	Java Naming and Directory Interface Java API for a directory service that allows Java software clients to discover and look up data and objects via a name. <i>A directory service</i> is a software system that stores, organises and provides access to information in a directory. In software engineering, a directory is a map between names and values. Like a dictionary, it allows the lookup of values given a name.
JVM	Java Virtual Machine A virtual machine capable of executing Java bytecode.

Term	Description (EN)
MOM	Message Oriented Middleware A MOM is a software or hardware infrastructure supporting the sending and receiving of messages between distributed systems.
n/a	not applicable
OCS	Online Collection System (for ECI)
OCT	Online Collection Tool <u>N.B. This name has been deprecated and replaced by the name <i>OnLine Collection System</i>.</u>
SQL 99	Structured Query Language 99 SQL 99 – also known as SQL 1999 or SQL 3– is the fourth revision ⁹ of the SQL database query language.
SSL/TLS	Secure Sockets Layer / Transport Layer Security Transport Layer Security and its predecessor Secure Sockets Layer are cryptographic protocols that provide communication security over the Internet.
UCS	Universal Character Set A standard set of characters containing almost one hundred thousand abstract characters, each identified by an unambiguous name and an integer number called its code point.
UI	User Interface
URL	Uniform Resource Locator A specific character string that constitutes a reference to an Internet resource.
UTF-8	UCS Transformation Format 8-bit A variable-width encoding that can represent every character in the Unicode character set.
WLST	WebLogic Scripting Tool A command-line scripting interface, using scripts to automatically configure and manage changes to WebLogic Server domains in a simple way.
XA	eXtended Architecture XA is an X/Open group standard for executing a "global transaction" that accesses more than one back-end data-store.
XML	Extensible Markup Language A markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.

⁹ At the time of writing (March 2012), the latest revision of the standard is SQL 2008.

OCS Release Notes
Author: Kenneth Irvine
Publishing Date: 5 April 2012
Current Version at Publishing Date: 1.2.1