

On-Site WebCSD User Guide

- 1 On-Site WebCSD User Guide
 - 1.1 Conditions of Use
- 2 On-Site Lattice and WebCSD Installation
 - 2.1 Introduction
 - 2.2 Prerequisites and Setup
 - 2.2.1 System Requirements
 - 2.2.2 SELinux
 - 2.2.3 Authentication
 - 2.2.4 Docker Installation
 - 2.2.5 Postgres CSD Database
 - 2.2.6 CCDC User Account
 - 2.3 Basic Configuration And Installation
 - 2.3.1 GUI installation
 - 2.3.2 Command Line Installation
 - 2.3.3 Post-Installation Checks
 - 2.4 Further Configuration
 - 2.4.1 User Access Control
 - 2.4.2 In-house Database Configuration
 - 2.4.3 CSD-Theory Web Database Configuration
- 3 Known Issues
 - 3.1 Restart the message queue
 - 3.2 I get an 'Upload failed' message when trying to upload a CIF
 - 3.3 Viewing charts in CSD-Theory gives a 'WebGL is not supported by your browser' message
- 4 Alternative ways to setup On-Site WebCSD
 - 4.1 Running the On-Site WebCSD installer from the command line
 - 4.2 Storing the docker images in your local repository
 - 4.3 Kubernetes support
- 5 Setting Up A Self-Hosted PostgreSQL Server
 - 5.1 Install PostgreSQL server
 - 5.2 Getting the latest version of the CSD Database
 - 5.3 Restore database
 - 5.4 Point services to use the Postgres database
- 6 Updating Your In-House Databases
 - 6.1 In-house database updates from version 4.3.0
 - 6.1.1 Initial database setup
 - 6.1.2 Adding new structures
 - 6.1.3 Editing existing structures

- 6.2 Database updates for versions prior to 4.3.0
- 7 Access Control To In-House Databases
 - 7.1 Overview
 - 7.2 Use of the Manage users plugin
 - 7.3 Use of the Manage groups plugin
 - 7.4 Access control to in-house databases
 - 7.5 User management (Admin)
 - 7.5.1 Bulk user management (Admin)
 - 7.5.2 FAQs
 - 7.5.3 Disabled Local Authentication
 - 7.6 Changes from the 4.3.0 release
- 8 Single Sign-On (SSO)
 - 8.1 Setting up single sign-on (Admin)
 - 8.1.1 Prerequisites for version 4.3.1
 - 8.1.2 Microsoft Entra ID
 - 8.1.3 LDAP SSO
 - 8.1.4 Prerequisites for version 4.2.1
 - 8.1.5 Azure Active Directory (AAD)
 - 8.1.6 LDAP
 - 8.1.7 Google
 - 8.1.8 Enabling single sign-on
 - 8.2 Using single sign-on (All users)
 - 8.2.1 For Existing Users
 - 8.2.2 For New Users
- 9 Setting up CSD-Theory Web
 - 9.1 Introduction
 - 9.2 Adding the CSD-Theory Landscape Database
 - 9.2.1 Adding an example CSD-Theory database to the Server
 - 9.2.2 Creating or Updating a CSD-Theory Landscape Database
 - 9.2.3 Adding the CSD-Theory Landscape Database to the Server
 - 9.2.4 Mapping to the databases in docker-compose.database.yml
 - 9.3 Additional landscape analysis
 - 9.3.1 Similarity heatmap
 - 9.3.2 Cross reference to experimental structures
 - 9.4 Remove a landscape
 - 9.5 Structure links
- 10 WebCSD Configuration And Customisation
 - 10.1 Introduction
 - 10.2 Formatting conventions
 - 10.3 Adding system administrators
 - 10.4 Adding your own company logo
 - 10.5 Changing the Identifier text
 - 10.6 Changing the non-CSD entry title
 - 10.7 Customised Header and Information pages
 - 10.8 Associated Structure Links Configuration
 - 10.9 Customising The Header
 - 10.10 Only enable Structure Links for a specific role

- 10.11 Change the message displayed if Structure Links are disabled for a user
- 10.12 Report Downloads - Configuration
- 10.13 Change the message displayed if report downloads are disabled for a user
- 11 Updates and uninstallation
 - 11.1 Software updates
 - 11.1.1 File ownerships
 - 11.1.2 User management database
 - 11.1.3 Using the GUI maintenance tool
 - 11.1.4 Using the maintenance tool from the command line
 - 11.1.5 Manual Update With Docker
 - 11.2 Updating the SSL certificate
 - 11.3 Updating licence keys
 - 11.4 Uninstallation
- 12 Support And Troubleshooting
 - 12.1 Contacting CCDC Support
 - 12.2 Providing feedback
 - 12.3 Licensing Options
 - 12.4 Troubleshooting
 - 12.4.1 How can I check which version of the images I am currently running?
 - 12.4.2 Viewing logs
 - 12.4.3 Testing connectivity to a self-hosted Postgres server
 - 12.4.4 While the docker login command works I am unable to pull docker images and receive an error
 - 12.4.5 I am prompted to enter a licence key when accessing the Lattice GUI
 - 12.5 I see an 'error:invalid_request' page when trying to log in to Lattice for the first time
- 13 Technical Information
 - 13.1 WebCSD configuration files
 - 13.1.1 Introduction
 - 13.1.2 SSL
 - 13.1.3 Databases
 - 13.1.4 User access control
 - 13.1.5 Troubleshooting
 - 13.2 Configuring service ports
 - 13.3 Identity Gateway (User Management Database)
- 14 Identity Gateway (User Account Database)
 - 14.1 Introduction
 - 14.2 User Databases
 - 14.3 Upgrading User Account Database From Previous Versions
 - 14.3.1 Version 4.0 Changes
 - 14.3.2 User Database Migration
 - 14.3.3 FAQs

1 On-Site WebCSD User Guide

Copyright © 2026 Cambridge Crystallographic Data Centre

Registered Charity No 800579

1.1 Conditions of Use

The Cambridge Structural Database Portfolio (CSD Portfolio) including, but not limited to, the following: ConQuest, CSD-Editor, Decifer, Mercury, Mogul, IsoStar, CSD Conformer Generator, Hermes, GOLD, SuperStar, the CSD Python API, web accessible CSD tools and services, WebCSD, CSD sketchers, CSD data files, CSD data updates, the CSD database, sub-files derived from the foregoing data files, documentation and command procedures, test versions of any existing or new program, code, tool, data files, sub-files, documentation or command procedures which may be available from time to time (each individually a Component) encompasses database and copyright works belonging to the Cambridge Crystallographic Data Centre (CCDC) and its licensors and all rights are protected.

Any use of a Component of the CSD Portfolio, is permitted solely in accordance with a valid Licence of Access Agreement or Products Licence and Support Agreement and all Components included are proprietary. When a Component is supplied independently of the CSD Portfolio its use is subject to the conditions of the separate licence. All persons accessing the CSD Portfolio or its Components should make themselves aware of the conditions contained in the Licence of Access Agreement or Products Licence and Support Agreement or the relevant licence.

In particular:

- The CSD Portfolio and its Components are licensed subject to a time limit for use by a specified organisation at a specified location.
- The CSD Portfolio and its Components are to be treated as confidential and may NOT be disclosed or re-distributed in any form, in whole or in part, to any third party.
- Software or data derived from or developed using the CSD Portfolio may not be distributed without prior written approval of the CCDC. Such prior approval is also needed for joint projects between academic and for-profit organisations involving use of the CSD Portfolio.

- The CSD Portfolio and its Components may be used for scientific research, including the design of novel compounds. Results may be published in the scientific literature, but each such publication must include an appropriate citation as indicated in the Schedule to the Licence of Access Agreement or Products Licence and Support Agreement and on the CCDC website.
- No representations, warranties, or liabilities are expressed or implied in the supply of the CSD Portfolio or its Components by CCDC, its servants or agents, except where such exclusion or limitation is prohibited, void or unenforceable under governing law.

Licences may be obtained from:

Cambridge Crystallographic Data Centre 12 Union Road Cambridge
CB2 1EZ, United Kingdom

Web: <https://www.ccdc.cam.ac.uk/> Telephone: +44-1223-336408

Email: admin@ccdc.cam.ac.uk

2 On-Site Lattice and WebCSD Installation

2.1 Introduction

On-Site Lattice and WebCSD is currently only available to CCDC industrial customers and other invited groups.

Note, the 4.3.0 and 4.3.1 releases do not support Macromolecule Hub. Do not update past version 4.2.1 if you are currently using this component.

2.2 Prerequisites and Setup

2.2.1 System Requirements

A docker server is required to run On-Site WebCSD. In the standard configuration the postgres server for CSD and Identity databases will also run in a docker container but there is an option to host this separately.

Recommended requirements for docker server hosting postgres container without Macromolecule Hub:

- 130GB free HD space, 16GB RAM, 8 core CPU.

Recommended requirements for Docker server hosting On-Site Lattice and WebCSD containers only:

- 30GB free HD space, 16GB RAM, 8 core CPU.

If self-hosting the postgres server, see [Setting up a self-hosted Postgresql server](#) for system requirements.

On-Site Lattice and WebCSD should work with any OS that meets the requirements to run Docker, but official support is provided by CCDC on the following platforms. Note that these match the 2026.1 CSD Portfolio Desktop release.

- RedHat Enterprise Linux 8, 9 and 10
- Rocky Linux 8, 9 and 10
- Ubuntu LTS 22 and 24

The libraries required for running the installer are the same as for installing the CSD Portfolio.

On RHEL / Rocky Linux 8 or 9:

```
sudo dnf install nss libXScrnSaver libglvnd-opengl libgfortran  
libxkbcommon-x11 xcb-util-wm xcb-util-cursor xcb-util-  
keysyms libatomic tk
```

On RHEL / Rocky Linux 10:

```
sudo dnf install nss libXScrnSaver libglvnd-opengl libgfortran  
libxkbcommon-x11 xcb-util-wm xcb-util-cursor xcb-util-  
keysyms libatomic libquadmath xdg-utils libXmu tk
```

On Ubuntu:

```
sudo apt-get install libnss3 libxss1 libopengl0 libgfortran5  
libxkbcommon-x11-0 libxcb-xinerama0 libxcb-cursor0 libxcb-  
keysyms1 libatomic1 libtk8.6 tk8.6
```

2.2.2 SELinux

The Qt installer will not run correctly if SELinux is in enforcing mode. If using RHEL / Rocky Linux, temporarily set SELinux to permissive mode.

```
sudo setenforce 0
```

2.2.3 Authentication

- A valid CCDC activation key (a 36-character key in the format #####-#####-#####-#####-#####-#####) or local license server URL will be required to use the software.
- Connecting to your WebCSD server via https is now required. To configure this you will need a password-protected .pfx certificate.
- Ask your local IT staff to obtain a valid CA-signed certificate for you, or generate a self-signed certificate as shown below.
- Note that SSO authentication and CSD-Theory Web will not work with self-signed certificates.

To generate a self-signed certificate

Generate private key

```
openssl genrsa -out private.key 2048
```

Generate certificate signing request (CSR) (OpenSSL will prompt for a password)

```
openssl req -new -key private.key -out request.csr
```

Generate self-signed certificate (valid for 365 days)

```
openssl x509 -req -days 365 -in request.csr -signkey private.key -out certificate.crt
```

Bundle into .pfx

```
openssl pkcs12 -export -out certificate.pfx -inkey private.key -in certificate.crt
```

- On-Site WebCSD is installed via Docker, which requires access to the CCDC docker container registry. To obtain a username and password please contact CCDC Support.

2.2.4 Docker Installation

A standard Docker Server and Docker Compose installation is required for installation of WebCSD. If you are new to Docker we recommend installing the latest version of [Docker Desktop](#). Note that this is likely to [require a license](#) for most use-cases. Alternatively, [Rancher Desktop](#) is an open-source alternative.

Linux users also have the option to install [Docker Engine](#) and the [Docker Compose plugin](#).

CHECK: Test your Docker installation as described at the above links. Also verify that you can log into the CCDC docker container registry:

```
docker login -u <user> -p <password> ccdcrepository.azurecr.io
```

```
# or to be prompted for the password  
docker login -u <user> --password-stdin ccdcrepository.azurecr.io
```

2.2.5 Postgres CSD Database

The postgres CSD database will be provided via a download link. To obtain this, please contact CCDC Support.

If self-hosting the postgres server please follow the instructions at [Setting up a self-hosted Postgresql server](#).

2.2.6 CCDC User Account

Containers run as the account with numeric uid 1397 and you will need to create this account and add it to the docker group. We suggest giving it the username ccdc.

```
# As of v2.0.0, containers run as non root users.  
# Because of this you will need to create the CCDC user if it does  
not already exist.  
sudo adduser ccdc --uid=1397  
  
# If you are upgrading from an older version to v4.0.0, ensure the  
user id is set to 1397  
sudo usermod -u 1397 ccdc  
  
# Create the docker group and add the CCDC user to it.  
sudo groupadd docker  
sudo usermod -aG docker ccdc  
  
# The pfx certificate created earlier must be owned by the ccdc  
user and group and should be only user readable  
sudo chown ccdc:ccdc certificate.pfx  
sudo chmod 400 certificate.pfx
```

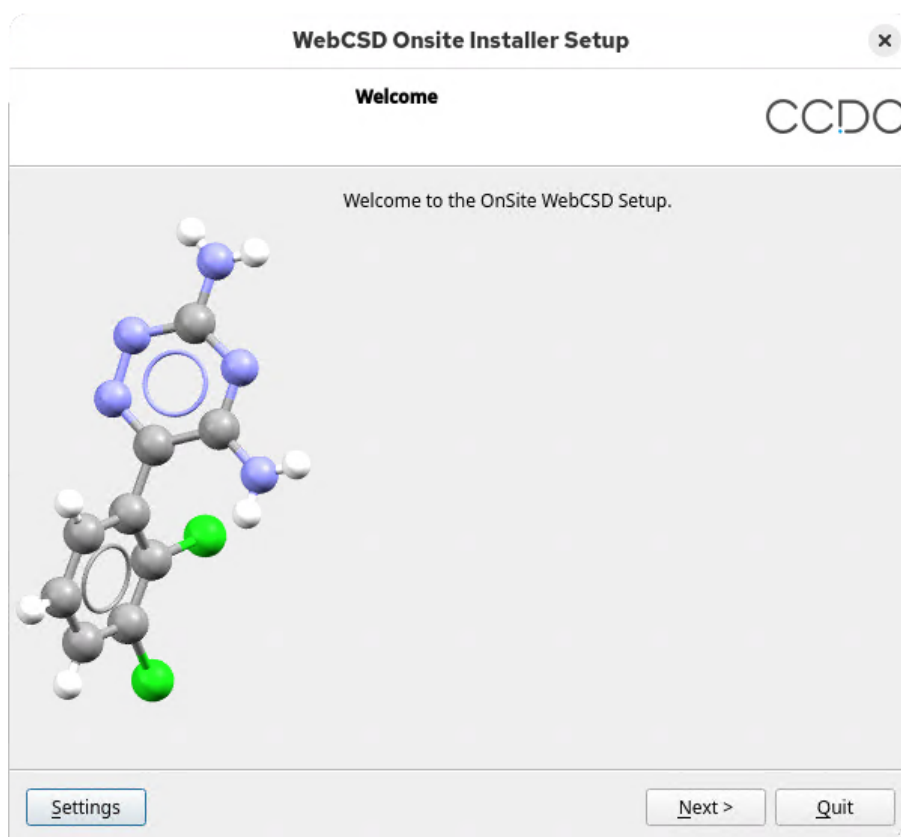
All installation methods (manual/installer) must be run from the ccdc account so that all file permissions are set up correctly.

2.3 Basic Configuration And Installation

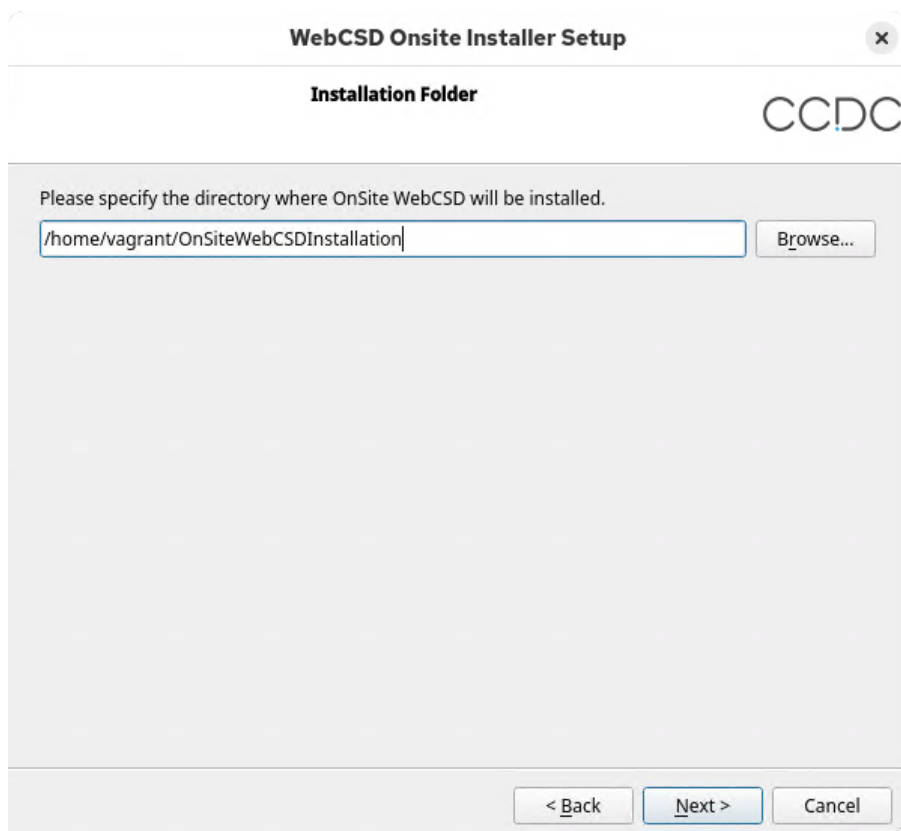
The installer is available from [GitHub](#) to run either via a GUI or command-line interface.

2.3.1 GUI installation

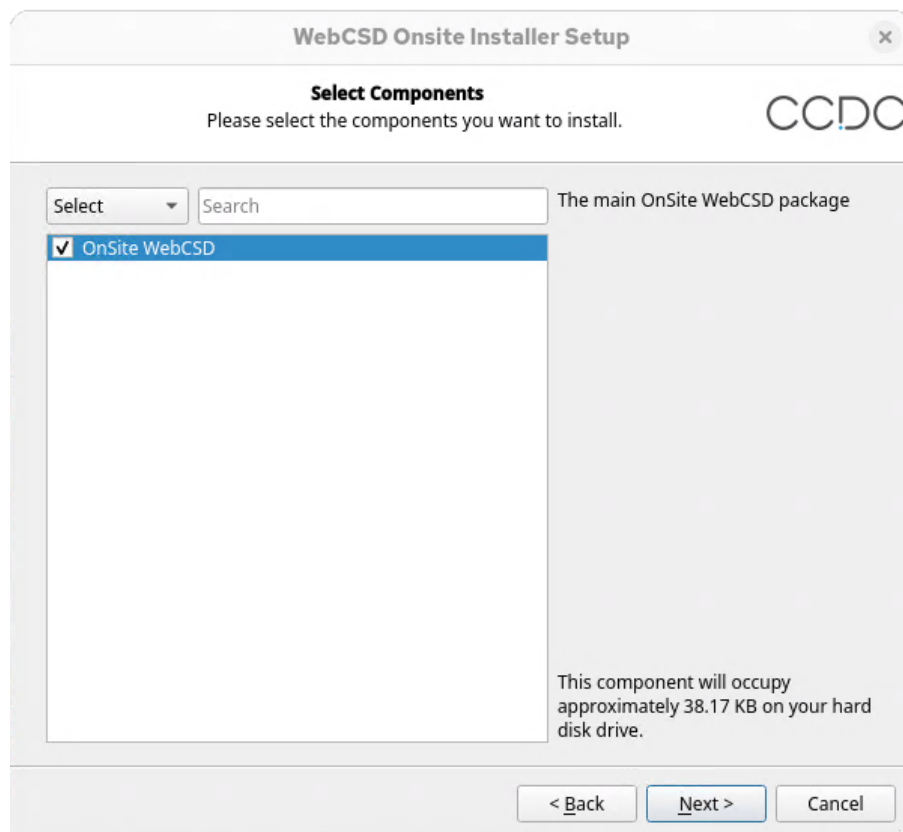
1. From the first screen, the **settings** button gives options for Network, Repositories and Local cache configuration. Click **Next** to continue with the setup



2. The next screen prompts for an installation directory (note system requirements)



3. The next screen details the components that will be installed, the current 4.3.1 release contains a single main package.



4. At the next stage, the configuration details required for setting up the server are required. This includes the location of the .pfx file and the associated password, the CCDC license key (or local license server URL) and the public URI for the server.
- If no .pfx file is provided a built-in self-signed certificate will be used. This is not recommended as it is insecure, and will not work with SSO authentication.
 - The public URI has the format `https://full.docker.hostname:PORT` where `full.docker.hostname` is your Docker host and `PORT` is a port number of your choice. If using the default 443 port this should not be included in the URI as it may be automatically stripped from the address by the browser.



The screenshot shows the 'Onsite Configuration' window of the 'WebCSD Onsite Installer Setup'. The window has a title bar with the text 'WebCSD Onsite Installer Setup' and a close button. Below the title bar, the text 'Onsite Configuration' is centered, and the 'CCDC' logo is on the right. The main area contains the instruction 'Please fill the following configuration fields:' followed by four labeled input fields: 'Certificate File (.pfx)' with a file selection button, 'Certificate File Password' with a text input containing 'Password to decrypt the certificate file', 'CCDC License Key' with a text input containing 'Contact CCDC support for a license key', and 'Public URI' with a text input containing 'https://your.server.hostname:port (leave port if default 443)'. At the bottom, there are three buttons: '< Back', 'Next >', and 'Cancel'.

WebCSD Onsite Installer Setup

Onsite Configuration

CCDC

Please fill the following configuration fields:

Certificate File (.pfx)

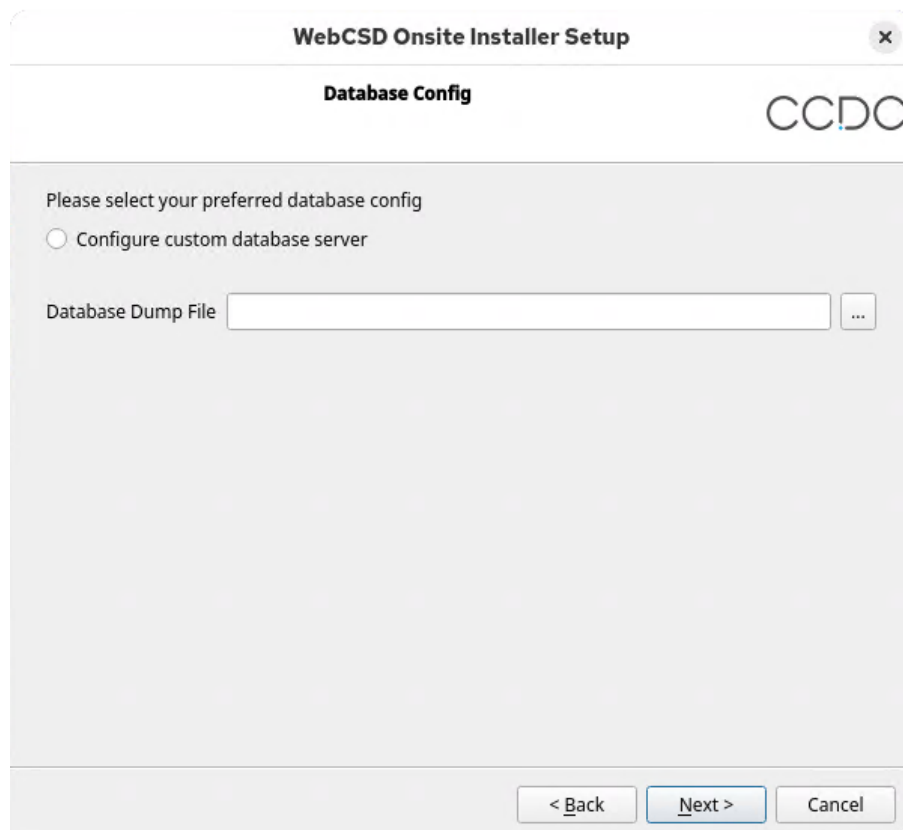
Certificate File Password

CCDC License Key

Public URI

< Back Next > Cancel

5. The following screen covers options for installing the CSD database. If the database has been downloaded as a Dump file there is an option to navigate directly to the location. Alternatively, a custom configuration for the database server can be specified by adding the Server, Port, User and Password.



The screenshot shows the 'Database Config' window of the 'WebCSD Onsite Installer Setup'. The window has a title bar with the text 'WebCSD Onsite Installer Setup' and a close button. Below the title bar, the text 'Database Config' is centered, and the 'CCDC' logo is on the right. The main area contains the instruction 'Please select your preferred database config' followed by a radio button labeled 'Configure custom database server'. Below this is a 'Database Dump File' label and a text input field with a file selection button. At the bottom, there are three buttons: '< Back', 'Next >', and 'Cancel'.

WebCSD Onsite Installer Setup

Database Config

CCDC

Please select your preferred database config

☐ Configure custom database server

Database Dump File

< Back Next > Cancel

Note: Installation of the database may overwrite any existing user data, be sure to export any required data prior to running the installation

6. The installation is now ready to proceed. From the final screen, click the **Install** button to begin the process.

During the installation a .env file and some .yaml files will be created in the installation directory. These are docker compose config files and can be edited to customise your installation.

2.3.2 Command Line Installation

It is also possible to run the WebCSD installer from the command line. This process is documented under [Alternative ways to setup On-Site WebCSD](#).

2.3.3 Post-Installation Checks

CHECK: To check that the install has completed and that all the services are running, either view the containers in the Docker Desktop GUI or run:

```
docker compose ps
```

which should show the state of the services to all be Up. If any services have the state Up (unhealthy) or Exit then restart the stack. In most cases it is best to leave the postgres server running as in the commands below.

If the postgres server is kept running, the output from taking down the stack may include the warning !Network onsitewebcsdinstallation_default Resource is still in use - this is expected and does not require further action.

```
# Replace `INSTALLDIR` by your WebCSD installation directory
# Include docker-compose.insecure-configuration.yaml if using a
  self-signed certificate
cd INSTALLDIR
docker compose -f docker-compose.yaml -f docker-compose.ssl.yaml
  down
docker compose -f docker-compose.yaml -f docker-compose.ssl.yaml up
  -d
```

If the csd-database container is unhealthy then do a full restart of the stack and bring it up before the other containers.

```
# Follow the above instructions instead if using a self-hosted
  postgres server
# Replace `INSTALLDIR` by your WebCSD installation directory
```

```
# Include docker-compose.insecure-configuration.yml if using a
  self-signed certificate
cd INSTALLDIR
docker compose -f docker-compose.yml -f docker-compose.ssl.yml -f
  docker-compose.postgres.yml down

docker compose -f docker-compose.yml -f docker-compose.ssl.yml -f
  docker-compose.postgres.yml up -d csd-database
docker compose -f docker-compose.yml -f docker-compose.ssl.yml -f
  docker-compose.postgres.yml up -d
```

If the issues persist then please contact CCDC Support.

To access the On-Site Lattice and WebCSD service locally go to the WebCSD server URL you entered at install time.

To check that you can search the CSD you must create an account.

If you have problems check your `.env` file.

- The correct format for the `CCDC_LICENSING_CONFIGURATION` line is `CCDC_LICENSING_CONFIGURATION=la-code;ACTIVATION_KEY;` or `CCDC_LICENSING_CONFIGURATION=lf-server;http://SERVER_NAME:PORT_NO`
- `PUBLIC_URI` should be set to the full URL of your WebCSD server, including the port number unless this is the default 443. This must resolve to your docker host.
- `PLATFORM_PORT` must match the port number in `PUBLIC_URI` (default is 443).

2.4 Further Configuration

Your On-Site WebCSD server is now up and running but you will probably need to configure it, for example to set up user access control via SSO. This is done via docker compose files (see [WebCSD Configuration Files](#)). After making any changes to these files, restart the stack as described above, adding any docker compose files which you have customised into the `docker compose ... up -d` command.

2.4.1 User Access Control

User access control can be managed via local accounts or Single Sign-On (SSO). You must configure roles to allow access to in-house databases as by default even admin roles can only access the CSD. For details please see [Access Control To In-House Databases](#) and [Single Sign-On \(SSO\)](#).

2.4.2 In-house Database Configuration

On-Site Lattice and WebCSD can be configured to read from in-house databases.

Provided within the installation is a sample database teaching_subset.csdsq1 which can be found in the sample-data folder in the root of the installation.

For versions 4.3.0 and later, details on in-house database configuration are available in the documentation: [Updating Your In-House Databases](#)

For earlier releases (version 4.2.1 and earlier) the following information is still applicable:

To enable in-house databases:

1. Copy and rename the file docker-compose.sample-0n-Site-only-db-config.yml to docker-compose.db-config.yml
2. Edit the volumes section of that file to point to any in-house databases and edit the environment section to configure the application to recognise these databases. Ensure that all instances of 'database-1' or 'database-2' are replaced with the actual database name.

For example, to use the 'teaching_subset.csdsq1' database and have it appear as a database named 'Example', your docker-compose.db-config.yml should look like:

volumes:

- /path/to/webcsd/sample-data/teaching_subset.csdsq1:/csd-data/teaching_subset.csdsq1

environment:

- ServiceSettings__Databases__2__Name=Example
- ServiceSettings__Databases__2__ConnectionString=/csd-data/teaching_subset.csdsq1

More information is given in the notes & example sections of the sample file. This acts as an [override file](#) which you will have to include in the startup command.

2.4.3 CSD-Theory Web Database Configuration

Instructions on setting up CSD-Theory Web can be found in the documentation: [Setting Up CSD-Theory Web](#).

Provided within the installation is a sample CSD-Theory landscape database `CSPLandscape.csdsqlx` which can be found in the `sample-data` folder in the root of the installation, along with a blank `CSPDatabase.db` CSD-Theory metadata database.

3 Known Issues

3.1 Restart the message queue

With 4.3.0 and 4.3.1 rabbitmq needs to be rerun after installation. The installer will do this automatically. If you installed manually via docker please run the following:

```
# Replace `INSTALLDIR` by your WebCSD installation directory
cd INSTALLDIR
docker compose -f docker-compose.yml up -d
```

3.2 I get an 'Upload failed' message when trying to upload a CIF

When the On-Site server is first installed there can be situations where there is a name mismatch between a self-signed SSL certificate and the server ID. This can be resolved by restarting the Docker stack e.g.

```
# Replace `INSTALLDIR` by your WebCSD installation directory
# Include docker-compose.insecure-configuration.yml if using a
  self-signed certificate
cd INSTALLDIR
docker compose -f docker-compose.yml -f docker-compose.ssl.yml
  docker-compose.insecure-configuration.yml down
docker compose -f docker-compose.yml -f docker-compose.ssl.yml
  docker-compose.insecure-configuration.yml up -d
```

3.3 Viewing charts in CSD-Theory gives a 'WebGL is not supported by your browser' message

This issue can be resolved by changing browser settings.

For Chrome browsers:

- Go to `chrome://flags`

- Search for "Override software rendering list", set it to **Enabled** and relaunch

For Firefox:

- Go to `about:config`
- Search for `webgl.force-enabled`, and set it to **true**
- Ensure `webgl.disabled` is set to **false**

4 Alternative ways to setup On-Site WebCSD

4.1 Running the On-Site WebCSD installer from the command line

If installing On-Site WebCSD on a headless system you can run the installer non-interactively from the command line. There are two approaches to this. In both cases if the default port 443 is used `publicURI` must be set to just `https://<hostname>` to avoid redirection errors.

1. Specify parameters on the command line. Everything in angle brackets must be replaced by your own details (and the angle brackets removed), and `publicURI` must resolve to your Docker host.

```
# Install with automatic database server configuration
# (recommended)
./OnSiteInstallerExecutable --accept-licenses --confirm-
command install --root "<installation directory>"
licenseKey="<activation key>" username="<username for
CCDC docker container registry>" password="<password
for CCDC docker container registry>"
publicURI="https://<hostname>:<port>" port="<port>"
dbDumpFile="<path to database dump file>"
sslCertLoc="<path to pfx cert file>"
certPw="<pfx file password>"
```

```
# Install and connect to a pre-existing database server
./OnSiteInstallerExecutable --accept-licenses --confirm-command install --root "<installation directory>"
licenseKey="<activation key>" publicURI="https://<hostname>:<port>" port="<port>" csdDatabase="csd-database-only-development" identityDatabase="csd-identity-onsite-release-2" sslCertLoc="<path to pfx cert file>" certPw="<pfx file password>"
dbConnString="Server:<database server name>;Port:<database server port e.g 5432>;User Id:<database user>;Password:<database password>;"
```

2. Create a .env file and copy it into your installation directory. As above, replace everything in angle brackets. The .env file below is for automatic database server configuration. If self-hosting the postgres server, remove the POSTGRES_DUMP_LOC line and change Server=csd-database in DB_CONNECTIONSTRING to Server=<postgres_server_name>.

```
SSL_CERT_LOC=<path to pfx cert file>
CCDC_LICENSING_CONFIGURATION=la-code;<activation key>;
PUBLIC_URI=https://<hostname>:<port>
PLATFORM_PORT=<port>
POSTGRES_DUMP_LOC=<path to database dump file>
POSTGRES_DB=csd-database
POSTGRES_PORT=5432
POSTGRES_USER=postgres
POSTGRES_PASSWORD=<test123>
DB_CONNECTIONSTRING=Server=csd-database;Port=5432;User Id=postgres;Password=<test123>;
CSD_DATABASE=csd-database
IDENTITY_DATABASE=csd-identity
```

Now you can use a shorter command (answer Yes to the initial “Do you want to continue?” question).

```
./OnSiteInstallerExecutable --accept-licenses --confirm-command install --root <installation directory>
```

4.2 Storing the docker images in your local repository

If you want to store the docker images in your own repository follow the below steps:

1. Download and unzip a copy of the source code of the latest release from [GitHub](#). Change to the on-site-webscd-VERSION folder containing the docker compose files.
2. Run the below command to download the docker images

```
docker compose pull
```

3. For each docker image run "docker tag <oldrepo/service:version> <newrepo/service:version>" e.g.

```
docker tag ccdcrepository.azurecr.io/webcsd:0.1.6  
my.internal.registry/webcsd:0.1.6
```

4. Run "docker push <newrepo/service:version>" e.g.

```
docker push my.internal.registry/webcsd:0.1.6
```

5. Create a new docker-compose file to keep your image overrides separate and avoid them being reverted in updates e.g. "docker-compose.service-config.yml". The file will need to contain the new image location for each service copied into a new location. e.g.

```
version: '3.6'  
services:  
  webcsdbackend:  
    image: my.internal.registry/webcsdbackend:0.1.6  
  webcsd:  
    image: my.internal.registry/webcsd:0.1.6  
  database-server:  
    image: my.internal.registry/csd-database:2022.1.0.alpha1  
...  
etc
```

6. Include the new file in the startup command, so if you are also using local database configurations the command will be:

```
docker compose -f docker-compose.yml -f docker-compose.db-  
config.yml -f docker-compose.service-config.yml up -d
```

Please note that process will need to be repeated to copy further updates to your local repository.

4.3 Kubernetes support

We do not currently offer Kubernetes support; please contact us to discuss your preferred installation options.

5 Setting Up A Self-Hosted PostgreSQL Server

5.1 Install PostgreSQL server

With the release of version 4.3 of the On-Site platform, by default the database server runs in a docker container. However, there is also the option to use your own instance of Postgres to host the database.

Recommended requirements for a Postgres server:

- Postgres version 14 or newer
- 100GB free hd space

For supported operating systems the native version of Postgres should be sufficient. To install on RHEL / Rocky Linux:

```
sudo yum install postgresql-server
```

To install on Ubuntu:

```
sudo apt-get install postgresql
```

N.B. The PostgreSQL server must also be configured to allow network connections from the WebCSD server.

5.2 Getting the latest version of the CSD Database

Contact support@ccdc.cam.ac.uk for assistance in getting the postgres database / databases you need. They will be able to provide you with a link to download the databases to your infrastructure.

5.3 Restore database

Restore the database you have downloaded into your new environment. If you are replacing an existing database first delete it:

```
psql -c "DROP DATABASE \"csd-database\""
```

Then install the new database:

```
psql -c "CREATE DATABASE \"csd-database\""  
pg_restore --no-owner -d csd-database <  
downloaded_database_name.dump >
```

If you are unable to perform a database backup as described above, please contact support@ccdc.cam.ac.uk for assistance.

5.4 Point services to use the Postgres database

The best way to do this is at install time.

- If running the GUI installer, select the option “Configure custom database server” on the Database Config screen.
- If running the CLI installer, set the `databaseConnectionString` parameter.

```
./OnSiteInstaller  
databaseConnectionString="DB_CONNECTIONSTRING=Server=database-  
server;Port=5432;User Id=postgres;Password=< passwordhere >"
```

Alternatively, after installation you can set the `DB_CONNECTIONSTRING` variable in the `.env` file as above.

6 Updating Your In-House Databases

6.1 In-house database updates from version 4.3.0

Updates to in-house databases are performed via the 'Manage databases' plugin accessible to all users with administrator rights.

6.1.1 Initial database setup

- From a new installation, a new, empty database must be created by clicking the 'Add new database` button from within the plugin, this will automatically be configured for searching with WebCSD.
- You will be prompted to add a name for the database, and optionally a description and colour.
- If you have an existing in-house database, the `.csdsqllx` database file created by Decifer or CSD-Editor can be imported directly by

clicking on the upload icon next to the database. This navigates to an Add structures page where the existing database can be uploaded.

6.1.2 Adding new structures

- Additional .csdsq1x databases can be added to an existing database following the same procedure as above, therefore any existing database creation workflows can be maintained.
- New CIF files can be added to the database in a similar method, however multiple CIFs can be added simultaneously. An important difference to note is that new structures in CIF format will be processed using the Decifer algorithm that assigns chemistry, resolves any disorder modelled in the structure and adds metadata from the CIF into the database entry. Additions from an existing database are not altered from the original database file.

6.1.3 Editing existing structures

- By selecting the Review structures icon next to the database, a list of all entries in the database is shown on a new Review structures page. Individual entries can be selected from this list; clicking on the Edit structures button will then create a .csde2 file which can be downloaded from the browser. This file should be automatically opened by the CSD-Editor software available from the CSD Portfolio (versions from CSD Portfolio release 2025.3 onwards) to allow edits to the database entries. Once edits have been completed, clicking the Save button within the editor will upload the entries back into the database hosted on your On-Site server.

6.2 Database updates for versions prior to 4.3.0

To update an existing database, first check the location of the database that you want to update. This will be specified in the webcsdbbackend volumes section of your docker-compose or in-house database configuration file. Copy the new version of the database to the required location, then do the following:

```
docker-compose down
```

```
docker-compose -f docker-compose.yml -f docker-compose.db-  
config.yml up -d
```

7 Access Control To In-House Databases

7.1 Overview

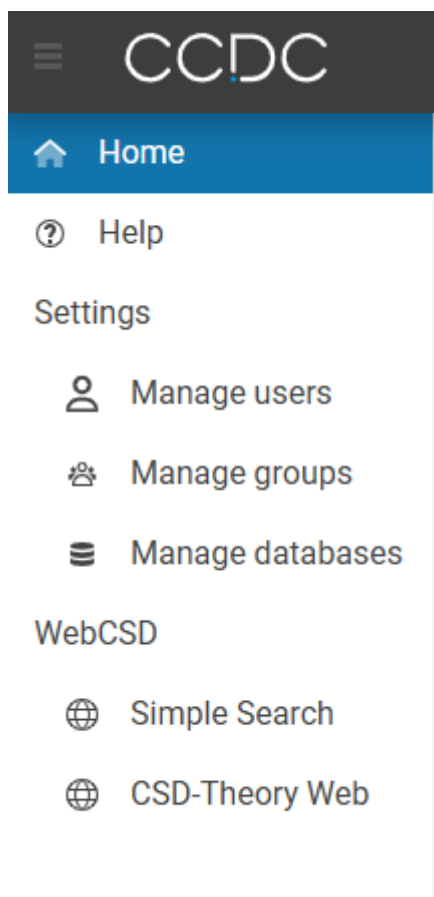
These instructions relate to the 4.3.0 and later releases

To create accounts users have two options:

1. Using single sign-on - see [Single Sign On \(SSO\)](#).
2. Using form based registration - documented below.

The first user to register will be set as the admin and they have the ability to grant admin rights to other users; there is no limit on the number of admin accounts allowed.

Users with administrator rights will have the ability to see functionality in the **Settings** section: Manage users, Manage groups and Manage databases. To access this functionality go to the Lattice Home page. Click the Lattice tab on any WebCSD search page or click Lattice Home in the sidebar on your profile page. If the sidebar on the Lattice Home page shows only icons then click the hamburger menu at the top left to expand it.



Users without admin rights will only see options for searching databases in the **WebCSD** section: Simple Search and CSD-Theory Web (if applicable).

By default all non-admin users start with no access to any in-house databases but will be able to see and search the CSD; by contrast users with admin rights are able to see all databases. Users who are not logged in cannot access any databases.

Note: Users will be required to log in to WebCSD by clicking the **Sign In** button at the top-right of the WebCSD page after they have logged in to the initial Lattice home page. This second log in will not require the user to re-add their username and password, but by clicking the Sign In button, the user's credentials are passed to the WebCSD search engine.

7.2 Use of the Manage users plugin

The Manage users plugin gives the ability to reset a password, grant administrator rights or delete the user from the system.

7.3 Use of the Manage groups plugin

To make it easier to grant access rights to users, the Manage groups plugin allows an administrator to add many individual users to a single group. Access to in-house databases is then granted to this group of users, rather than having to assign permissions individually. A new group can be created by selecting the **Add new group** button, at which point there are options to give a group name and description. Users can then be added or removed from this group by selecting it in the Manage groups plugin in future.

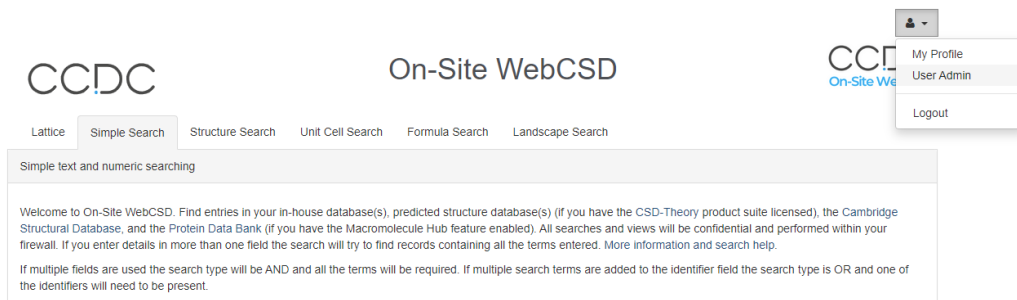
7.4 Access control to in-house databases

Having admin rights gives the ability to add, update or delete any in-house databases from the Manage databases plugin. To grant access to other users, the Manage databases plugin contains the option to add groups to each database (using the padlock icon next to the database). All users are automatically included in an 'Everyone' group; should user-level access control not be required, ensure each database has this group selected. Alternatively, different user groups can be selected on a per-database level, allowing the ability to control access to all structures.

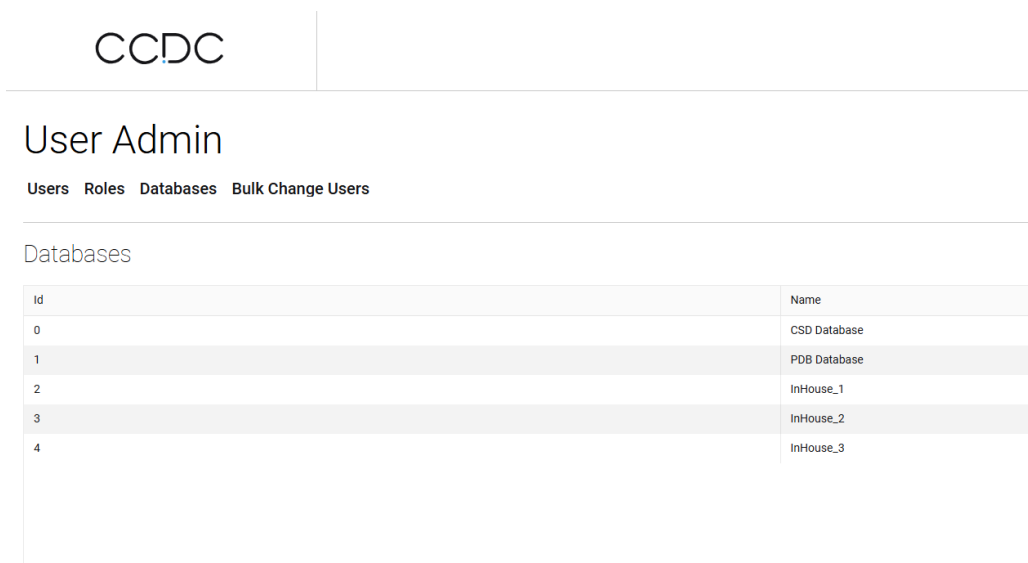
The User management instructions below refer to the earlier 4.2.1 release

7.5 User management (Admin)

1. Navigate to the User Admin section via WebCSD or the New Lattice Platform.



2. Go to Databases tab. Here you can see an index assigned to each database which is taken from the container configuration.



3. Go to Roles tab. Here you can see the name of available roles and indices of the databases accessible for the users with this role.

User Admin

[Users](#) [Roles](#) [Databases](#) [Bulk Change Users](#)

Roles

Role Name

Add New Role

Name	Assigned Databases	Id
Basic	0	Edit
StructureLinks		Edit
ReportDownload		Edit
Admin		Edit
AllDBs	0,1,2,3,4	Edit

4. You can add a new role by entering its name into the relevant field and clicking Add New Role.

User Admin

[Users](#) [Roles](#) [Databases](#) [Bulk Change Users](#)

Roles

Role Name

Add New Role

5. You can click on the Edit button to the right of each role and tick those databases which you want to make available for the users with this role. Click on Submit when all the relevant databases are selected.

User Admin

Users Roles Databases Bulk Change Users

Edit Role: NewRole

Assign which databases this role can use:

- ☐ CSD Database
- ☒ PDB Database
- ☒ InHouse_1
- ☒ InHouse_2
- ☐ InHouse_3

Submit

Cancel

6. You can assign roles separately for each user or in a bulk manner (described in the next section). To assign role(s) to one user, go to Users and click Manage roles to the right of their email.

User Admin

[Users](#) [Roles](#) [Databases](#) [Bulk Change Users](#)

Manage User Roles

Add/Remove Roles for User **user2@ccdc.cam.ac.uk**.

- ☒ Basic
- ☐ StructureLinks
- ☐ ReportDownload
- ☐ Admin
- ☐ AllDBs
- ☒ NewRole

Submit

Cancel

7.5.1 Bulk user management (Admin)

1. Navigate to the User Admin section via WebCSD or the New Lattice Platform.

The screenshot displays the 'On-Site WebCSD' interface. At the top left is the CCDC logo. To its right is the text 'On-Site WebCSD'. Below this is a navigation bar with tabs: 'Simple Search', 'Structure Search', 'Unit Cell Search', 'Formula Search', and 'Landscape Search'. The 'Simple Search' tab is active. Below the navigation bar is a section titled 'Simple text and numeric searching'. This section contains a welcome message and a 'Databases' section with checkboxes for 'CSD', 'Main RP DB', 'RP Test 2', 'CSP database' (which is checked), and 'Solubility'. At the bottom of the 'Databases' section, there is a note about access to databases and contact information for System Administrators.

CCDC On-Site WebCSD

Simple Search Structure Search Unit Cell Search Formula Search Landscape Search

Simple text and numeric searching

Welcome to On-Site WebCSD. Find entries in your in-house database(s), predicted structure database(s) (if you have the CSD-Theory product suite licensed), the Cambridge Structural Database, and the Protein Data Bank (if you have the Macromolecule Hub feature enabled). All searches and views will be confidential and performed within your firewall. If you enter details in more than one field the search will try to find records containing all the terms entered. [More information and search help](#).

If multiple fields are used the search type will be AND and all the terms will be required. If multiple search terms are added to the identifier field the search type is OR and one of the identifiers will need to be present.

Databases

☐ CSD ☐ Main RP DB ☐ RP Test 2 ☒ CSP database ☐ Solubility

If you require access to databases that are not shown. Firstly check that you are logged in, if you still do not have access then please contact any of your System Administrators: adminuser@company.com, adminuser2@company.com

hudson@ccdc.cam.ac.uk

My Profile

User Admin

Logout

Manage

Lattice

Welcome to Lattice, CCDC's new web platform. We are actively building new applications and functionality, so please keep in touch and check our [What's New?](#) page regularly.

You can still access On-Site WebCSD using the menu to the left.

Search

2. Select Bulk Change Users.

CCDC

My Account User Admin

User Admin

Users Roles Databases Bulk Change Users

Bulk Change Users

Upload

To insert or change user roles in bulk, please take the following steps:

1. Create a CSV file and populate it with the information you wish to bulk update. Please note, it needs to be in a specific format, to download a template file click below.
2. Upload the file
3. Review validation messages and make any required changes
4. Submit final changes

Please upload your bulk change CSV file below.

Select Bulk Change CSV File

Download Template File

Upload Bulk Change CSV File

Cancel

3. Create a CSV file with user data in the following format.

```
FirstName,LastName,EmailAddress,Roles
Joe,Bloggs,joe@joe.com,"Basic"
```

You should then have a file that looks something like this:

```
1 FirstName,LastName,EmailAddress,Roles
2 Blakelee,Santarelli,bsantarelli3@washington.edu, "Basic"
3 Mile,Cornfield,mcornfield3@abc.net.au, "Basic"
4 Chilton,Voak,cvoak8@yale.edu, "Basic"
5 Karen,Haldenby,khaldenby1@economist.com, "Basic"
6 Thomasin,Walles,twalles2@gmpg.org, "Basic"
```

4. Select Bulk Change CSV File to upload the populated CSV file and hit the Upload Bulk Change CSV File button to upload and validate the provided user details.

5. Validation results will be displayed. If any errors are found, you'll be alerted to make the necessary corrections and re-upload the file.

File with failed validation:

User Admin

[Users](#) [Roles](#) [Databases](#) [Bulk Change Users](#)

Bulk Change Users

Validate & Save

← Back

CSV Uploaded successfully with 5 rows identified.

Some errors have been found. Please go back and upload a new CSV with corrections before proceeding.

First Name	Last Name	Email Address	Password	Roles	Valid	Notes
Blakelee	Santarelli	bsantarelli3@washington.edu		Basic	✓	
Mile	Cornfield	mcornfield3@abc.net.au		NewRole	✗	Role: NewRole Does not exist
Chilton	Voak	cvoak8@yale.edu		Basic	✓	
Karen	Haldenby	khaldenby1@economist.com		Basic	✓	
Thomasin	Wallis	gmpg.org		Basic	✗	Email Address gmpg.org is not valid

File with successful validation:

User Admin

[Users](#) [Roles](#) [Databases](#) [Bulk Change Users](#)

Bulk Change Users

Validate & Save

← Back

CSV Uploaded successfully with 5 rows identified.

No errors found. Review your data below and click 'Submit User Changes' at the end to apply the changes to the system.

First Name	Last Name	Email Address	Password	Roles	Valid	Notes
Blakelee	Santarelli	bsantarelli3@washington.edu		Basic	✓	
Mile	Cornfield	mcornfield3@abc.net.au		Basic	✓	
Chilton	Voak	cvoak8@yale.edu		Basic	✓	
Karen	Haldenby	khaldenby1@economist.com		Basic	✓	
Thomasin	Wallis	twallis2@gmpg.org		Basic	✓	

✓ Submit User Changes

- After carefully reviewing the user data, click Submit User Changes to finalize the bulk registration/change process.
- Download the generated report by clicking on the Export User List to Excel button. **This contains the initial passwords for the created accounts. You will not be able to access this later.**

7.5.2 FAQs

What fields are required on the CSV template file? Only the Email address and Roles are required.

What happens when a user already exists? The User's role(s) would be updated (see below)

User Admin

Users Roles Databases Bulk Change Users

Bulk Change Users

Validate & Save

← Back

CSV Uploaded successfully with 5 rows identified.

No errors found. Review your data below and click 'Submit User Changes' at the end to apply the changes to the system.

First Name	Last Name	Email Address	Password	Roles	Valid	Notes
Blakelee	Santarelli	bsantarelli3@washington.edu		Basic	✓	User: bsantarelli3@washington.edu Already Exists in the system. This will reset the users roles.
Mile	Cornfield	mcomfield3@abc.net.au		Basic	✓	User: mcomfield3@abc.net.au Already Exists in the system. This will reset the users roles.
Chilton	Voak	cvoak8@yale.edu		Basic	✓	User: cvoak8@yale.edu Already Exists in the system. This will reset the users roles.
Karen	Haldenby	khaldenby1@economist.com		Basic	✓	User: khaldenby1@economist.com Already Exists in the system. This will reset the users roles.
Thomasin	Walles	twalles2@gmpg.org		Basic	✓	User: twalles2@gmpg.org Already Exists in the system. This will reset the users roles.

✓ Submit User Changes

I forgot to download the final report containing the user passwords, what do I do? Admin would have to reset/change each user's password manually.

Can I update existing users' roles in bulk as well? Yes, populate the CSV file with the users Email address and their respective updated Roles, upload and the system will automatically update their roles.

7.5.3 Disabled Local Authentication

Your site admin, based on company policy, may decide to restrict users to sign in to the application using only SSO, and hence can disable the option to sign in locally with a username and password.

7.6 Changes from the 4.3.0 release

For earlier versions of the On-Site platform it was possible to disable all user access control. This ability has now been removed due to additional functionality granted to users with administrator rights.

8 Single Sign-On (SSO)

8.1 Setting up single sign-on (Admin)

These instructions apply to the 4.3.1 release

8.1.1 Prerequisites for version 4.3.1

To use OpenID Connect (SSO), SSL must be enabled. please see Updating the SSL certificate for how to set that up.

The 4.3.1 release supports both LDAP providers (such as Windows server Active Directory) and Microsoft Entra ID (formerly Azure AAD). If your organisation has other preferred options, please contact our support team at support@ccdc.cam.ac.uk to discuss requirements.

8.1.2 Microsoft Entra ID

8.1.2.1 Azure App Registration

- Sign in to the [Azure portal](#)
- Search for the **Microsoft Entra ID** service, and from that page select **App Registration** from the '+ Add' dropdown option near the top of the page.
- Provide an application name
- Select the option **Multiple Entra ID tenants** - Note: this option can be configured to use a single tenant
- This option gives an additional radio button selection - choose **Allow only certain tenants (Preview)**
- As no tenants have currently been specified, click on the **Manage allowed tenants (0)** link to add your information
- The Tenant ID can be found by searching 'Tenant ID' and selecting the [Tenant properties](#) service
- The Tenant ID should be added to the allowed list and click **Apply**
- Set the Redirect URI:
 - From the 'Select a platform' dropdown pick the **Web** option
 - The URI will have the format `https://{yourserver}/identity/callback/login/microsoft`
- Note the **Application (client) ID** from the overview page as this is required for your server set up as the <Microsoft Client ID> (see below)
- Navigate to **Certificates & secrets** and click on **+ New client secret**

- Provide a description for the secret and copy the value for your server set up as the <Microsoft Client Secret> (see below)

8.1.2.2 Configure Entra on your On-Site installation

- Your On-Site installation comes with a sample file `docker-compose.sample.entra.yml`, create a new version `docker-compose.entra.yml`
- Update the file with your specific information as follows:

Enable Microsoft Entra single sign-on

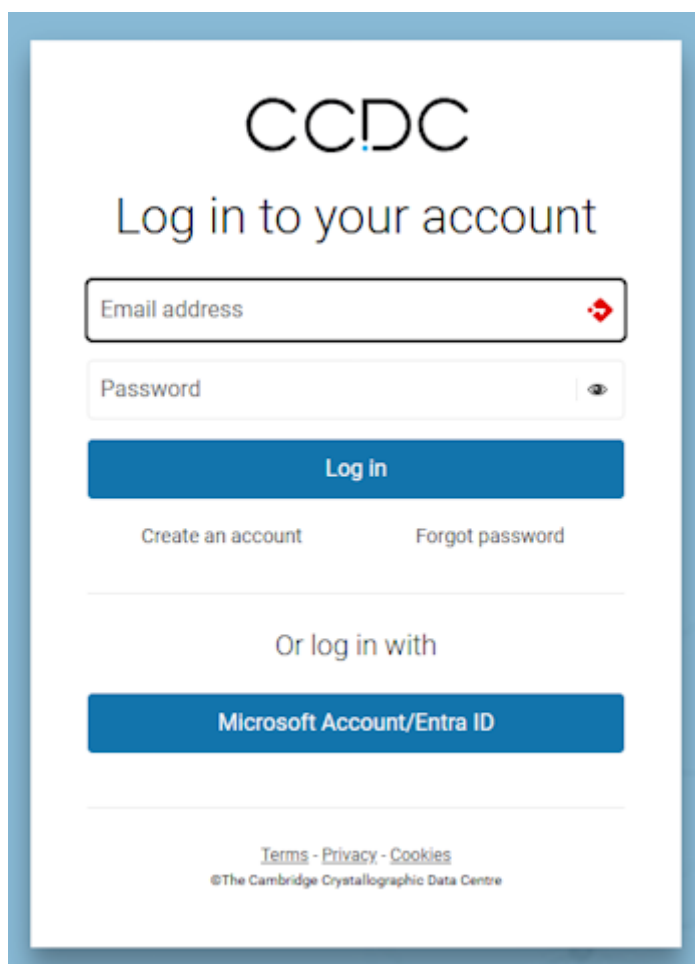
services:

ccdc-identity:

environment:

- `OpenIdDict__EnableExternalLogin=true`
- `OpenIdDict__MicrosoftClientId=<Microsoft Client ID>`
- `OpenIdDict__MicrosoftClientSecret=<Microsoft Client Secret>`

- Restart the stack with a command including all the necessary yml files e.g.: `docker compose -f docker-compose.yml -f docker-compose.ssl.yml -f docker-compose.entra.yml up -d`
- The login page for your On-Site installation should now include an Entra ID option



CCDC

Log in to your account

Email address

Password

Log in

Create an account Forgot password

Or log in with

Microsoft Account/Entra ID

[Terms](#) - [Privacy](#) - [Cookies](#)

©The Cambridge Crystallographic Data Centre

8.1.3 LDAP SSO

A sample LDAP yml configuration file `docker-compose.sample.ldap.yml`, which should allow for use of LDAP for user authentication, can be found in the top level of the github repository. This can be used as the basis for a file `docker-compose.ldap.yml` that contains custom settings for your organisation. However as there are various types of LDAP service available this may require some modification based on the type in use. For additional help with configuration, please contact support@ccdc.cam.ac.uk for assistance but please note that due to the breadth of potential systems that could be used, the help we can offer for setup may be limited.

Once the configuration is complete, restart the stack with a command including all the necessary yml files e.g.

```
docker compose -f docker-compose.yml -f docker-compose.ssl.yml -f docker-compose.ldap.yml up -d
```

These instructions apply to the 4.2.1 release

8.1.4 Prerequisites for version 4.2.1

To Enable SSO for On-Site users, site admin must first register an application with a preferred OpenID Connect identity provider, and obtain these open id access credentials: Client id, Client secret, Authority and CallbackPath. The following sub-sections provide guidelines on how to register an application with major identity providers such as Microsoft - Azure Active Directory (AAD), Google and LDAP providers such as Windows server Active Directory.

To use OpenID Connect (SSO), SSL must be enabled. please see Updating the SSL certificate for how to set that up.

8.1.5 Azure Active Directory (AAD)

- Sign in to the [Azure portal](#).
- If you have access to multiple tenants, use the Directories + subscriptions filter in the top menu to switch to the tenant in which you want to register the application.
- Search for and select Azure Active Directory.
- Under Manage, select App registrations > New registration.
- Enter a Name for your application, for example CCDC-WebCSD. Users of your app might see this name, and you can change it later.
- Add the URL `https://{webcsd-url}/identity/signin-oidc` in Redirect URI as a Web platform.
- Select Register.
- Under Manage, select Authentication.
- In the implicit grant and hybrid flows section, select ID tokens, and then select save.
- Under Manager, select Certificates & Secrets.
- Select +New client secret, enter a description, select secret expiry date and select Add to generate a client secret. Copy the generate Client secret value as it can't be retrieved after this point.
- On the Overview page, also copy the client id (Application (client) ID) and tenant id (Directory (tenant) ID).

- **Replace in this endpoint `https://login.microsoftonline.com/{tenant id}/v2.0` to form the Authority credential.**

8.1.6 LDAP

A sample LDAP yml configuration file which should allow for use of LDAP for user authentication can be found in the top level of the github repository. However as there are various types of LDAP service available this may require some modification based on the type in use. For additional help with configuration, please contact support@ccdc.cam.ac.uk for assistance but please note that due to the breadth of potential systems that could be used, the help we can offer for setup may be limited.

8.1.7 Google

Sign in as a [Google developer](#).

- From the Credentials tab, select Create Project, provide a project name (could be name of your organization), select a parent organization, and click Create.
- Select Configure Consent Screen to configure the OAuth consent screen with information about the application to be registered.
- On the OAuth Consent Screen, select the Internal user type and click Create.
- Complete the customization options as desired and select Back to Dashboard on the summary page.
- Return to the Credentials tab and select +Create Credentials. Choose OAuth client ID.
- Select Web Application for the Application Type.
- Enter WebCSD as the client name and select Create.
- Copy the generated Credentials and/or download the generated JSON file which includes all the necessary credentials.
- **Use `https://accounts.google.com/` as the Authority and `/signin-google` as CallbackPath**

8.1.8 Enabling single sign-on

Before proceeding with this section, you must have obtained the required credentials: Authority, ClientId and ClientSecret.

Update the environment section of the **docker-compose.enable-sso.yml** file with the credentials.

services:

```
  ccdc-identity:
    environment:
      - OnSiteLoginSetup__EnableSingleSignOn=true
      - OnSiteLoginSetup__EnableFormBasedLogin=true
      - ThirdPartyCredentials__CallbackPath=/Identity/signin-oidc

      # Single Sign On Configuration
      # The following configuration is required to enable single
      # sign on

      # The authority URL of the identity provider
      - ThirdPartyCredentials__Authority=https://
        login.microsoftonline.com/{Directory (tenant) ID}/v2.0
      # The client id of the application registered with the
      # identity provider
      - ThirdPartyCredentials__ClientId={Application (client) ID}
      # The client secret of the application registered with the
      # identity provider
      - ThirdPartyCredentials__ClientSecret={Client Secret Value}
      # The display name of the identity provider
      - ThirdPartyCredentials__DisplayName={Name of SSO provider.
        Eg. Microsoft}
```

Note: Admin can disable local authentication and require users to strictly sign in with only SSO by setting the `OnSiteLoginSetup__EnableFormBasedLogin` variable to false.

Finally, include the `docker-compose.enable-sso.yml` in the start-up command:

```
docker-compose -f docker-compose.yml -f docker-compose.enable-
sso.yml up -d
```

8.2 Using single sign-on (All users)

8.2.1 For Existing Users

Users who already have a local account would need to first link their local account to their third party account (Microsoft, Google, etc.) in their profile before using the sign-in button on the login page. This is necessary to avoid changes in user roles and permissions or loss of user's data.

First, sign in with your local account.

- Navigate to the My Profile area.

- Select External Logins.
- Select the configured SSO provider (Microsoft, Google, etc.) to link your local account. You'll be redirected to the provider's page for sign in and authorisation.
- If login succeeds, you'll be redirected back to the application and be notified that the accounts have been linked successfully.

My Account Logout

My Account

Profile Email Password External logins Two-factor authentication Personal data

External Logins

The external login was added.

Registered Logins

Azure AD

Info: This process is only required on first login with SSO. Subsequent logins can be done straight away from the login page by clicking on the Continue with.

8.2.2 For New Users

From the Log in page, click Continue with you'll be redirected to the SSO provider to sign in (if you're not already signed in).

On successful authentication, the Identity Gateway will initiate an account registration process which will create a local profile and link it with the third party account you just signed into.

You will be presented with the Register page as shown below. Enter your email in the space provided if it was not pre-populated, and click Register.

Register

Associate your Azure AD account.

You've successfully authenticated with **Azure AD**. Please enter an email address for this site below and click the Register button to finish logging in.

Email Address

Register

You can now proceed to login.

9 Setting up CSD-Theory Web

9.1 Introduction

Note: The 4.3.0 release does not support CSD-Theory Web.

Note: Restricting access to the CSD-Theory database is not possible with the 4.3.1 release. All users with an account will have the option to search the CSP landscapes.

Ensure that your licence has the CSD-Theory feature enabled. If this is not the case, please contact our support team at support@ccdc.cam.ac.uk who will be happy to assist you. Once an updated licence is available, if necessary update the licence in the installation using the instructions on [changing licence keys](#).

This section of the documentation contains information on the setup options for CSD-Theory Web. CSD-Theory Web requires two databases:

1. A CSD-Theory metadata database containing additional theoretical data. This must be named `CSPDatabase.db`. This is populated during the ingestion script.
2. A CSD-Theory landscape database for initial searching of landscape structures, e.g. `CSPLandscape.csdsq1x`. This is populated during the ingestion script.

These two databases are used in conjunction to create the CSD-Theory charts and reports.

We have provided both a `CSPDatabase.db` and `CSPLandscape.csdsq1x` with an example landscape. You can use this to set up CSD-Theory before adding your own landscapes. Once you have ingested your landscapes you will need to remove the example landscape following the instructions further down the page.

9.2 Adding the CSD-Theory Landscape Database

9.2.1 Adding an example CSD-Theory database to the Server

- Ensure you have copied the sample [CSPDatabase.db](#) and [CSPLandscape.csdsq1x](#) from the `/sample-data/` to your `/csd-theory/` folder.
- Containers run as the `ccdc` user, so ensure the folder and associated files are owned by the `ccdc` user. Run e.g. `sudo chown -R ccdc:ccdc csd-theory` on the folder, and also database files which the server will edit when the ingestion script is run.
- The WebCSD server should be restarted for the new database to become searchable by users.

9.2.2 Creating or Updating a CSD-Theory Landscape Database

To create or update a CSD-Theory Landscape Database, you will need the Python Utilities. These can be downloaded from the [CCDC Downloads page](#) under CSD Python API > Python CSD Python API Utilities, or by filling in the form to [request download links](#).

Note it is not currently possible to run the scripts if using a self-signed SSL certificate.

Relevant scripts can be found under `ccdc > utilities > csp`.

Use the `import_csp_landscape.py` script to import the CIFs. This script can build a new database or update an existing one. Run the script as a user who can write to the `csd-data` directory.

argument	description	examples
landscape_database	URL of the landscape database server (the	https://csd-theory.ccdc.cam.ac.uk https://csd-theory.ccdc.cam.ac.uk:8080

argument	description	examples
	same as the URL you use in your web browser)	
structure_database	crystal structure database (.csdsqlx). Either point to an existing database via the filepath, or give it a name to build it directly.	E:\CSP_Upload\existingLandscapeDatabaseName newLandscapeDatabaseName.csdsqlx
cif_files	CIF file(s) for structure(s) to be imported to the landscape and structure databases (.cif)	E: \CSP_Upload\CSP_enriched_CIFs\singlestructure E:\CSP_Upload\CSP_enriched_CIFs*.cif
-n LANDSCAPE_NAME	The name of the CSP landscape. If not included, landscape name will be generated from the CIFs	
-v, --validate	Validate, rather than add, that the CIF files have been added to a	

argument	description	examples
	CSP landscape	

Some sample commands include:

Validate importing a single CIF into an existing database at <https://csd-theory.ccdc.cam.ac.uk>:

```
import_csp_landscape.py https://csd-theory.ccdc.cam.ac.uk E:
\CSP_Upload\existingLandscapeDatabaseName.csdsqlyx E:
\CSP_Upload\CSP_enriched_CIFs\singlestructurelandscape.cif -v
```

Import a folder of CIFs into a new database called "NewDatabase.csdsqlyx" at <https://csd-theory.ccdc.cam.ac.uk:8080/>, with landscape name "NewLandscape":

```
import_csp_landscape.py https://csd-theory.ccdc.cam.ac.uk:8080
NewDatabase.csdsqlyx E:\CSP_Upload\CSP_enriched_CIFs\*.cif -n
NewLandscape
```

9.2.3 Adding the CSD-Theory Landscape Database to the Server

The CSD-Theory Landscape database needs to be added to the /csd-theory/ folder.

If you are copying the database via an SCP command to transfer the file between a local host and a remote host, you may need to run chown on the /csd-theory/ folder to change it to the owner who is running SCP, such as onsiteadmin.

Ensure once the SCP command or ingestion script run is complete, you change the owner of /csd-theory/ and all databases inside back to ccdc. The user "ccdc" must have read access to the CSP databases.

The following section applies only to versions 4.2.1 and earlier:

9.2.4 Mapping to the databases in docker-compose.database.yml

Start by taking down the instance of the WebCSD docker container and update the docker-compose.database.yml file. You can start from docker-compose.sample-csd-theory-db-config.yml. Updates should include:

Update locations and names in the the webcsdbackend volumes and environment sections, e.g.

```
services:
  webcsdbackend:
    volumes:
      - ./csd-data/LandscapeDatabaseName.csdsq1x:/csd-data
    environment:
      - ServiceSettings__Databases__2__Name=LandscapeDatabaseName
      - ServiceSettings__Databases__2__ConnectionString=/csd-data/
        LandscapeDatabaseName.csdsq1x
      - ServiceSettings__Databases__2__Speciality__0=CSP
```

the volumes of the webcsd-theory section and optional settings in the environment section (described below), e.g.

```
webcsd-theory:
  volumes:
    - ./csd-data/CSPDatabase.db:/app/CSPDatabase.db
    - ./csd-data/LandscapeDatabaseName.csdsq1x:/app/databases/
      CSPLandscape.csdsq1x

  environment:
    - ReportSettings__CspStructureDatabaseConnectionString=/
      app/databases/CSPLandscape.csdsq1x
    - ReverseProxy__Clusters__csdtheory__Destinations__csdtheory/
      destination1__Address=http://webcsd-theory:${WEBCSD_PORT}
```

Once you have edited and saved the docker-compose.database.yml file, restart the docker container. You should now be able to access all the basic CSD-Theory functionality.

9.2.4.1 Reports

If you wish to run reports, the Landscape Database needs to be mapped to CSPLandscape.csdsq1x and you need to include the ReportSettings environment information.

9.2.4.2 Ports other than 80

If you are running on a port other than 80, ensure the ConnectionString environment detail (ReverseProxy) includes the port information. Do not include a backslash on the CSP url.

9.3 Additional landscape analysis

Once a CSP landscape has been ingested into CSD-Theory there are additional steps that can be taken to add extra information.

9.3.1 Similarity heatmap

To compare structures within a landscape, the script `csp_similar_structures.py` will produce a heatmap showing the similarity between (up to) the 25 lowest energy structures. The script has a syntax

```
python csp_similar_structures.py http://my.landscape.server:12345  
wonder_drug.csdsq1x wonder_drug --pxrd_threshold 0.98
```

This will run similar structures calculations in the CSP landscape `wonder_drug`. The optional argument `--pxrd_threshold` sets the PXRD similarity threshold.

9.3.2 Cross reference to experimental structures

Structures within a landscape can also be compared to experimental data from an in-house database. This requires the script `csp_cross_reference.py` with the syntax

```
python csp_cross_reference.py http://my.landscape.server:12345  
wonder_drug.csdsq1x my_in-house_database.csdsq1
```

This will cross reference all structures in the CSP database "wonder_drug" with observed structures in the `my_inhouse_database` of experimental structures. If no in-house database is referenced the script will run a comparison against the CSD.

9.4 Remove a landscape

You can remove a landscape by running the following Python script.

```
python delete_csp_landscape.py http://my.landscape.server:12345/  
wonder_drug.csdsq1x wonder_drug
```

More information on this script can be found in the CSD Python API documentation.

9.5 Structure links

Additionally, it is possible to add links from theoretical structures to another web server. To do this create a CSV file named `StructureLinks.csv` with the following column headings, and as many rows as you have links.

Identifier	Key	Value
structure identifier of a CSP landscape structure in the CSPLandscape.csdsq1x file	StructureLink	Any URL or file path

Copy `StructureLinks.csv` into the `csd-theory` folder and restart the server.

10 WebCSD Configuration And Customisation

10.1 Introduction

This section details a number of customisations you can make to WebCSD. In order to keep these settings from being accidentally over-written, we recommend putting any customisations in a separate docker-compose file, e.g. `docker-compose.customisations.yml`. This acts as an override file which you will have to include in the startup command. Alternatively you could add these settings into the aforementioned `db-config` file to keep all of your configuration in one place.

10.2 Formatting conventions

- Any file location beginning `/path/to` should be replaced by the file location on your docker host.
- Any filename or text in angle brackets should be replaced by your own content and the angle brackets removed.
- In lines with two pathnames separated by a colon, the first is generally the location on your docker host and needs to be customised. The second is the location in the container and should be left alone.

For example, you might change the line

- /path/to/logo/<CompanyLogo.png>:/app/wwwroot/images/OnSite-logo.png

to

- /usr/local/logos/MyCompanyLogo.png:/app/wwwroot/images/OnSite-logo.png

10.3 Adding system administrators

WebCSD allows you to display local system administrators who can provide support. To add one or more email addresses of system administrators, add them to the webcsd section of your own docker-compose file.

For example:

- Customisations__SystemAdministrators__0=TestAdmin1@ccdc.cam.ac.uk
- Customisations__SystemAdministrators__1=TestAdmin2@ccdc.cam.ac.uk

10.4 Adding your own company logo

Replace the standard "CCDC On-Site WebCSD" logo at the top right of the page, with your own company logo, by adding the following to the webcsd volumes section of your own docker-compose file.

- /path/to/logo/<CompanyLogo.png>:/app/wwwroot/images/OnSite-logo.png

As there is no height or width specified when displaying this image, please ensure the replacement is of a reasonable size, to integrate with the layout of the page. For example, the CCDC logo, at the top left of the page, is 142 x 50 pixels.

10.5 Changing the Identifier text

Change the label used for entry identifiers in the search results list, by adding the following settings to the webcsd environment section of your own docker-compose file.

- Customisations__Identifier__Full=<Identifier text>
- Customisations__Identifier__Short=<Identifier text short version>

10.6 Changing the non-CSD entry title

Change the text displayed before the identifier in the main title, when viewing a non-CSD entry, by adding the following setting to the webcsd environment section of your own docker-compose file.

- `Customisations__NonCsdEntryTitle=<In-house entry text>`

10.7 Customised Header and Information pages

To add custom HTML above the search form on the home page, create a file containing a HTML snippet and add the following setting to the webcsd volumes section of your own docker-compose file.

- `/path/to/<Custom_Header.html>:/app/wwwroot/static/HomeHeader.html`

Clicking on About This Service in the footer or accessing /access-structures-information brings up the information page. To add custom HTML for this page, create a file containing a HTML snippet and add the following setting to the webcsd volumes section of your own docker-compose file.

- `/path/to/<Custom_Information.html>:/app/wwwroot/static/Information.html`

PLEASE NOTE: The content of these pages is rendered by the site as is, without encoding or validation. Any embedded scripts in HomeHeader.html or Information.html will run. It is the site administrator's responsibility to ensure that these pages are properly secured.

10.8 Associated Structure Links Configuration

On-Site WebCSD can be configured to display links associated with a structure, e.g DOI links or File links. To do this you will need a CSV file containing structures with their associated data and any files you may want to download, with a separate CSV file for each database hosted on the server.

The CSV file should have three headers - Identifier,Key,Value. The Identifier refers to the database identifier for the individual structure in the database. The Key is StructureLink - other keys have been used in previous version of the On-Site platform but are no longer used. The value is the link required e.g. DOI links or File links.

Note: URL links require a full path including both a root domain and page to resolve correctly. For example, a link to `https://www.ccdc.cam.ac.uk/support-and-resources/` will resolve correctly, whereas a link to just the root domain `https://www.ccdc.cam.ac.uk/` will not.

The following information refers to the 4.3.0 release

Users with admin rights can upload the CSV file directly via the **Manage databases** plugin. Each database has a 'three dots' button that links to the **Import additional data** page. From this page the CSV file can be uploaded.

The following information refers to the 4.2.1 release

The CSV file is used by webcsdbbackend and should be configured in the environment section by adding the following line to the environment section under webcsdbbackend in your in-house database yml file:

```
services:
  webcsdbbackend:
    volumes:
      - /path/to/database-1:/csd-data/database-1
      - /path/to/database-2:/csd-data/database-2
      - /path/to/structure-links.csv:/csd-data/structure-links.csv:RO
    environment:
      - ServiceSettings__Databases__2__Name=database-1
      - ServiceSettings__Databases__2__ConnectionString=/csd-data/database-1
      - ServiceSettings__Databases__3__Name=database-2
      - ServiceSettings__Databases__3__ConnectionString=/csd-data/database-2
      - ServiceSettings__StructureInfoLocation=/csd-data/structure-links.csv
```

This mounts the `structure-links.csv` in the `csd-data` directory. Change the location to where the csv file you are using is located.

The related files themselves are used by webcsd and their location is configured in the volumes section of the in-house database yml file, i.e.

```
webcsd:
  volumes:
    - ./path/to/structure/files:/structure-files
```

Here the structure files folder `./path/to/structure/files` is mounted as `/structure-files`, which is where WebCSD will expect to find them.

The related files can be stored within a folder structure. It is important to note that any file links set up in the CSV file must point to the correct location of these files, otherwise the file will not download when the link is clicked. For example, given the folder structure:

```
\structure-files
  \structure1
    \rawfiles
      myfile.raw
    \xyefiles
      myfile.xye
```

In the CSV file, the above would be referenced by:

```
Identifier,Key,Value
STRUCT01,StructureLink,/structure-files/structure1/rawfiles/
      myfile.raw
STRUCT01,StructureLink,/structure-files/structure1/xyefiles/
      myfile.raw
```

10.9 Customising The Header

The structure links are displayed when viewing a structure. By default these are shown in a box labelled 'File Locations'. To change this for your installation, add the following setting to the webcsd environment section of your own docker-compose file:

```
- Customisations__StructureLinksTitle=<title text>
```

■

10.10 Only enable Structure Links for a specific role

The following information refers to the 4.2.1 release

You can enable structure links downloads just for specific users, using a new Role that is automatically created.

When enabled, users will only be able to download structure links if they have been specifically assigned the StructureLinks role. If they have not been given access, the buttons will be greyed out and a message appears on hover.

Structure links will be **enabled** in the following circumstances:

- Structure links role checking is disabled
- Structure links role checking is enabled and the user has the StructureLinks role

Structure links role checking is disabled by default. To enable this for your installation, add the following setting to the webcsd environment section of your own docker-compose file:

- `Customisations__StructureLinksDisableByRole=true`

10.11 Change the message displayed if Structure Links are disabled for a user

If Structure links role checking is enabled, when a user does not have the required role, a message will display when attempting to click a download button. This message is customisable, to change it, add the following setting to the webcsd environment section of your own docker-compose file:

- `Customisations__StructureLinksDisabledText=<message text>`

10.12 Report Downloads - Configuration

The following information refers to the 4.2.1 release

You can restrict the download of the reports generated by the PXRD Match and Report buttons to specific users by adding them to the reportDownload role (which is automatically created).

Users with the reportDownload role have both the PXRD Match and Report buttons on the landscape results page enabled and available for use. For users without the reportDownload role, both buttons are disabled and greyed out with a message appearing on hover.

The reportDownload role checking is disabled by default. To enable this for your installation, add the following setting to the webcsd environment section of your own docker-compose file:

- `Customisations__ReportDownloadDisableByRole=true`

10.13 Change the message displayed if report downloads are disabled for a user

If report download by role is enabled, a user without the required role sees the default message This functionality has not been enabled. Please contact the system administrator to add this role to your account. when attempting to click any of the report download buttons. This message is customisable. To change it, add the following setting to the webcsd environment section of your own docker-compose file:

- `Customisations__ReportDownloadDisabledText=<message text>`

11 Updates and uninstallation

11.1 Software updates

11.1.1 File ownerships

For updating from pre-4.3.0 versions, the ccdc account needs access to the necessary docker volumes.

```
sudo chown -R ccdc:ccdc lic
sudo chown -R ccdc:ccdc userdata
```

11.1.2 User management database

For updating from pre-4.0.0 versions, please see [Identity Gateway](#) if you wish to keep your user management database.

11.1.3 Using the GUI maintenance tool

To update your installation, run the maintenance tool and select "Update components". Click Ok when a warning message pops up. This will automatically pull the latest versions of all containers and restart the stack.

For major releases a new database dump file is shipped. Contact CCDC Support for the database download link. The maintenance tool will prompt for the location. It will then overwrite the csd database volume with the update, please export any in-house databases beforehand. Once the update has completed they will need to be reimported within Lattice -> Database Management.

11.1.4 Using the maintenance tool from the command line

The maintenance tool can be run non-interactively from the command line too.

```
# To upgrade (ensure in-house database have been exported and if  
    using the installer setup postgres server  
# that the database dump file has been replaced with the update in  
    the same location originally specified -  
# can be found in docker-compose.postgres.yml)  
./maintenancetool update dbDumpFile="<path to database dump  
    file>" overWriteDb="True"
```

11.1.5 Manual Update With Docker

For major releases export any in-house databases, recreate the csd-database and restore it with a new database dump provided by Support.

```
# Replace `INSTALLDIR` by your WebCSD installation directory  
# Include docker-compose.insecure-configuration.yml if using a  
    self-signed certificate  
cd INSTALLDIR  
docker compose -f docker-compose.yml -f docker-compose.ssl.yml  
    down  
docker compose pull  
docker compose -f docker-compose.yml -f docker-compose.ssl.yml up  
    -d
```

Now from Lattice -> Database Management, reimport your in-house databases.

11.2 Updating the SSL certificate

SSL is required for your On-Site server, if it is not configured the server will use a built-in self-signed certificate which is insecure. Normally, SSL will be configured as part of the installation process. To reconfigure it manually, you will need to do the following:

1. Obtain a .pfx certificate.
2. Update the SSL_CERT_LOC variable in your .env file to point at this file then restart the stack e.g.

```
SSL_CERT_LOC=/ccdc/on-site-webcsd/ssl_cert.pfx
```

11.3 Updating licence keys

To change your licence key you must update licence information in the .env file, delete the licence file with the current licence details, and restart the stack:

An example of the process:

- Bring down the server e.g.

```
# Replace `INSTALLDIR` by your WebCSD installation directory
# Include docker-compose.insecure-configuration.yml if using a
  self-signed certificate
cd INSTALLDIR
docker compose -f docker-compose.yml -f docker-compose.ssl.yml
down
```

- Edit the .env file to enter the new license key e.g.

```
vi .env
```

Change the content after CCDC_LICENSING_CONFIGURATION= to include the new key (press i to insert new text, Esc to leave editing mode and :wq to save edits)

- Remove the current licence information e.g.

```
cd lic followed by rm data.conf to remove the file
```

- Finally, bring up the server to use the new licence information

```
# Replace `INSTALLDIR` by your WebCSD installation directory
# Include docker-compose.insecure-configuration.yml if using a
  self-signed certificate
cd INSTALLDIR
docker compose -f docker-compose.yml -f docker-compose.ssl.yml up
-d
```

To use a local licence server, the server URL should have the prefix lf-server; in the CCDC_LICENSING_CONFIGURATION= field, as opposed to la-code; as used for a licence key.

11.4 Uninstallation

To uninstall On-Site WebCSD run the GUI maintenance tool and select "Remove all components". Or run it from the command line with the purge option.

```
# To uninstall, this will give a Yes/No prompt to confirm the  
uninstall  
./maintenancetool purge
```

12 Support And Troubleshooting

12.1 Contacting CCDC Support

If you experience any difficulties with installing or using On-Site WebCSD, please contact our support team at support@ccdc.cam.ac.uk who will be happy to assist you.

12.2 Providing feedback

If you would like to provide some feedback our support team would be happy to hear from you at the above email address.

12.3 Licensing Options

The environment variable `CCDC_LICENSING_CONFIGURATION` needs to be set in the `.env` file. This will normally happen automatically when you enter your licence key at install time. You may also need to edit it after installation, for example if your licence key has changed.

For best performance we recommend that this variable is set to use our standard node-locked licensing using this format (where 123456-123456-123456-123456-123456-123456 should be your node-locked licence activation key):

```
CCDC_LICENSING_CONFIGURATION=la-code;  
123456-123456-123456-123456-123456-123456;
```

It is also possible to connect to your own local licence server:

```
CCDC_LICENSING_CONFIGURATION=lf-server;http://myserver:1234
```

More details on the options available with the `CCDC_LICENSING_CONFIGURATION` environment variable can be found in the Installation Notes for the latest CSD Portfolio release in the [Documentation](#) section of the [CCDC website](#).

12.4 Troubleshooting

12.4.1 How can I check which version of the images I am currently running?

```
docker compose images
```

12.4.2 Viewing logs

To view the container logs, use the following command:

```
docker logs [OPTIONS] <CONTAINER>
```

For a list of containers, use the following command:

```
docker compose ps
```

Support may ask for the full set of logs to be sent when attending a ticket, use the following command:

```
# collect logs creates tar.gz file which can be sent to support.
cd <installation directory>
./collect_logs.sh
```

For more information see the Docker documentation: <https://docs.docker.com/engine/reference/commandline/logs/>

12.4.3 Testing connectivity to a self-hosted Postgres server

To test whether your WebCSD server can connect to your Postgres server, run the following on the docker host.

```
# Pull postgres image
docker pull postgres:14.13
# Run container
docker run --name debug-postgres -e POSTGRES_PASSWORD=password -d
    postgres:14.13
# Shell into container
docker exec -it debug-postgres sh
# Attempt to establish a connection to the database server
# Replace my.postgres.server by the hostname of your postgres
    server
psql -h my.postgres.server -p 5432 -U postgres -d csd-database
```

If all is well the response to the final command will be similar to

```
psql (14.13 (Ubuntu 14.13-1.pgdg20.04+1))
SSL connection
    (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits:
    256, compression: off)
Type "help" for help.
```

12.4.4 While the docker login command works I am unable to pull docker images and receive an error

If you receive an error like

```
ERROR: pull access denied for //ccdcrepository.azurecr.io/onsite/
csd-database, repository does not exist or may require 'docker
login': denied: requested access to the resource is denied"
```

but the docker login command was successful then it is likely that you are using the main branch rather than a release. Please ensure you use the latest official release from <https://github.com/ccdc-opensource/on-site-webcsd/releases>.

If you are unable to pull the public RabbitMQ image, you may be using an old version of Docker.

- To check your docker version type `docker version`
- To install the latest version of docker see [the Docker documentation](#)

12.4.5 I am prompted to enter a licence key when accessing the Lattice GUI

The licence key should now only need to be entered via the installer or edits to the server configuration - see [Updating licence keys](#). If the front-end GUI is prompting for a licence key, this means the details entered in the back-end configuration are invalid and must be updated as described in the linked page.

12.5 I see an 'error:invalid_request' page when trying to log in to Lattice for the first time

Logging in to the On-Site installation for the first time may result in an error page displaying a message similar to:

```
error:invalid_request
error_description:The specified 'redirect_uri' is not valid for
    this client application.
error_uri:https://documentation.openiddict.com/errors/ID2043
```

This is likely to indicate a mismatch between the `PUBLIC_URI` field set in the `.env` file and the server URL specified in the browser. A common cause is specifying the default port `:443` in the `PUBLIC_URI` field; the port should only be specified if it is non-standard (not the default `443`).

To resolve the issue, first edit the `.env` file to ensure the `PUBLIC_URI` is the same as the URL given in the browser. Once the `.env` file has been updated, restart the docker containers for the change to take effect.

13 Technical Information

13.1 WebCSD configuration files

13.1.1 Introduction

WebCSD configuration is handled via a number of `.yaml` files which can be included as part of the `docker compose` command when bringing up the stack e.g.

```
docker compose -f docker-compose.yaml -f file1.yaml -f file2.yaml up
-d
```

`docker-compose.yaml` is the main configuration file. We do not recommend editing it - instead put your changes in a separate "override file" and add it into the `docker compose` command. Example changes are listed at [WebCSD Configuration And Customisation](#).

Configuration files other than the main file may be renamed, particularly those with "sample" in the name. Which of these files you need to use depends on your WebCSD configuration. The installer should take care of including the correct files but the information below is helpful if you need to restart the stack.

13.1.2 SSL

To secure connections via SSL which is now compulsory:

- Set up a pfx SSL certificate
- Ensure that `CERTIFICATE_PASSWORD` and `SSL_CERT_LOC` are set correctly in `.env` (the installer should take care of this)
- Include `docker-compose.ssl.yaml` in the `docker compose` command
- If your certificate is self-signed, also include `docker-compose.insecure-configuration.yaml`

13.1.3 Databases

In version 4.3.0:

- Include `docker-compose.postgres.yml` if running the PostgreSQL server in a docker container (the default configuration)

In version 4.2.1 and earlier:

- If using in-house databases without CSP, edit and include `docker-compose.sample-onsite-only-db-config.yml`
- If using in-house databases with CSP, edit and include `docker-compose.sample-csd-theory-db-config.yml`

13.1.4 User access control

- For SSO (single sign-on), edit and include `docker-compose.sample-enable-sso.yml`.
- For LDAP, also edit and include `docker-compose.sample.ldap.yml`.

For earlier versions of the On-Site platform it was possible to disable all user access control. This ability has now been removed due to additional functionality granted to users with administrator rights.

13.1.5 Troubleshooting

CCDC support staff may ask you to include `docker-compose.debug.yml` to obtain more informative logs.

13.2 Configuring service ports

The port on which users connect to the On-Site Lattice and WebCSD services is controlled by the `PLATFORM_PORT` variable in your `.env` file which is set to port 443 by default but can be set to a different value at install time.

By default the On-Site Lattice and WebCSD services are configured to run using port 8080 for all inter-service communication.

13.3 Identity Gateway (User Management Database)

If you are upgrading from version 3.0 or earlier and want to keep your user management database, please see instructions in the next chapter.

14 Identity Gateway (User Account Database)

14.1 Introduction

The identity gateway has been implemented since version 4.0. It provides a way to authenticate and manage users signing into On-Site Lattice and WebCSD. It holds the following responsibilities:

- Manage my user profile - Allow users to manage their passwords, and change aspects of their profile.
- User Administration - Administrators have access to an area to manage users, update passwords and set group permissions.
- Authentication endpoints - Users in On-Site Lattice and WebCSD will be redirected to the identity gateway to sign in.

14.2 User Databases

The identity gateway needs a postgres database to manage user data. This is similar to the userdata sqlite database used in previous versions.

The identity gateway will create this database on initial startup if one does not exist.

To configure database connections, update the following section of the docker-compose file:

```
ccdc-identity:
  environment:
    - "ConnectionStrings__DefaultConnection=Server=database-server;Port=5432;Database=identity;UserId=postgres;Password=${CSD_DB_PASSWORD};"
    - "ConnectionStrings__SqlitePath=userdata/UserLogin.db"
```

The sqlfilepath setting is required if you have a previous userdata sqlite database that you would like to retain user data from. See below the section on Upgrading Identity for more information.

14.3 Upgrading User Account Database From Previous Versions

14.3.1 Version 4.0 Changes

The system from version 4.0 onward has introduced significant changes in how users are managed. A new Identity gateway service manages users for all services. This means:

- The new Identity gateway has its own new postgres user database
- Users are now administered via the Identity Gateway
- Users sign in and out via the Identity Gateway
- Users can manage their own profile (change password, etc) via Identity Gateway

14.3.2 User Database Migration

Up to version 3.0, users were held in a Sqlite database managed by WebCSD, users were also managed via WebCSD.

Version 4.0 includes an automatic migration mechanism that undertakes the following functions:

- Creates a new postgres database
- Generates a schema for user management
- Reads the previous Sqlite database (if one exists) and migrates all users and roles to the new database

14.3.3 FAQs

14.3.3.1 Will users need to re-register?

No. Any users that had registered (or been registered) for version 3.0 or previous will be automatically migrated.

14.3.3.2 Will users need a new password?

No. Passwords will also be migrated as part of the upgrade process.

14.3.3.3 What will happen to the old Sqlite database?

This will be preserved. The database will not be deleted, but it will no longer be used.

14.3.3.4 When will migration run?

When the identity gateway starts up for the first time after the upgrade, it will create the database and then attempt a migration if it can find a previous user database.

14.3.3.5 How does this work with Single Sign On?

Single sign on user data will be migrated and users should be able to sign on as they did before.

14.3.3.6 Can I re-run the migration at a later date?

Yes. Just remove the identity postgres database and restart the service. On startup, it will attempt another migration.

14.3.3.7 How does the migrator know where my Sqlite database is?

The internal configuration for the identity gateway contains a connection string, this can be configured as detailed above.

14.3.3.8 Can I start afresh with a blank user database?

Yes, either remove or rename the Sqlite database, or change the identity gateway connection string to point elsewhere.