Design Room ONE

Installing Design Room ONE

This document describes how to setup and configure Design Room ONE server and client applications and integrations.

Table of Contents

Introduction

Design Room ONE consists of two components:

1. The Design Room ONE web server
   This is a Node.js server which uses a Mongo database.
2. The Design Room ONE Integration plugin for Eclipse-based modeling environments: Rational Software Architect Designer (RSAD), HCL RTist, and Rational Software Architect RealTime Edition (RSARTE). Among other things it provides an Export wizard for exporting RSAD/RTist/RSARTE models to the Design Room ONE web server.

If you already have installed Design Room ONE and just want to update it to the latest version go directly to Updating a Design Room ONE Installation.

Installing and Deploying the Web Server

Design Room ONE can be installed by means of scripts that take care of the installation and deployment of the web server. Alternatively, it can be installed using Docker.

Installation Using Scripts

Design Room ONE contains scripts which can be used for installation and deployment of the web server. These are Node.js scripts and work on both Windows and Linux. The examples in this chapter use Windows-style paths. On Linux, replace the backslashes (\) with forward slashes (/).

Before you can run the Node.js scripts you obviously must install Node.js:

1. Unzip the file DesignRoomONE-<version>_<platform>.zip to an empty folder `<dr-install-folder>`. For example, the name of the ZIP file may be
   `DesignRoomONE-2.1-2020.26_v20200623_1634_windows.zip` or
   `DesignRoomONE-2.1-2020.26_v20200623_1633_linux.zip`

> Note! The built-in “Extract All…” utility in Windows has a limitation on the length of the path names of the unzipped files. You can reduce the risk of running into this limitation by extracting the ZIP file into a folder where the full path name is short. If this doesn’t help, use a better

2. Go to `<dr-install-folder>` and unzip the web server ZIP file (called dr_server_<version>_<platform>.zip) to a folder `<dr-server-install-folder>`.
3. Install Node.js. You must have administrative or root access rights on the machine to perform the installation.
   On Windows double click on `<dr-server-install-folder>\DR_ReleaseManagement\server-sw\win32_x86_64\node-v8.11.3-x64.msi`.
   Accept the default installation options.

On Linux run the following command in a shell
`tar -Jxf <dr-server-install-folder>/DR_ReleaseManagement/server-sw/linux_x86_64/node-v8.11.3-linux-x64.tar.xz`

On Linux, you must manually update the PATH variable so that it contains the `bin` folder of the Node.js installation. The Windows installer automatically updates the PATH variable. Before you proceed, test that the `node` command is available from the command-line by typing
`node --version`
The version of Node.js should be printed.

You will use the `node` command for running the Design Room ONE installation and deployment scripts. The table below summarizes what these scripts do, and what they assume from the environment:

preinstall-dr.js	Downloads the Mongo database and the PM2 process management application from the internet. The license agreements of these applications prevent us from bundling them with Design Room ONE. This script must be run on a machine with internet access.
install-dr.js	Installs the Mongo database and the PM2 process management application. This script should be run on the machine where you want to install the Design Room ONE web server. Internet access is required, see the Note! section before step 5 to proceed with an offline installation.
deploy-dr.js	Starts the Mongo database and launches the Design Room ONE web server. This script should be run on the same machine where install-dr.js was run.

Note! If you do not have administrative or root access to your machine you may get errors when running the installation and deployment scripts. On Windows, you get administrative rights by right-clicking on the Command Prompt item in the Start menu and perform "Run as administrator". On Linux, you instead type `sudo` in front of the command.

4. Open a new command prompt and go to `<dr-server-install-folder>\DR_Install`. Execute the following command:
   `node preinstall-dr.js`

Note! On Linux, the scripts will not download and install the Mongo database automatically. The reason is that the recommended way to install Mongo on Linux is to use the package management system that is provided by each Linux distribution. See the Mongo documentation for more information.

The `preinstall-dr.js` script downloads the Mongo database (Windows only) and the PM2 process management application from the internet. The downloaded applications are placed in `<dr-server-install-folder>\DR_Install\downloaded`.

If you want to install the Design Room ONE web server on a machine without internet access, you should now copy the `<dr-server-install-folder>` to the installation machine and proceed with the remaining steps on that machine. Of course, you then also need to install Node.js on that machine (see step 3 above).

The next script to run is `install-dr.js` which installs the Mongo database (Windows only). The installation is controlled by the settings in the file `<dr-server-install-folder>\DR_Install\dr-config.json`. Review these settings and update them if needed before running the script. Settings marked with the
`// user option`
comment are those that you may want to change before installing. On Linux, pay particular attention to the setting mongo-linux-install-folder and ensure that it is set to the path where `bin/mongod` is located.

Besides editing the file `dr-config.json` directly, you can also customize the installation settings in one of the following ways:

- Set an environment variable or use a command line argument. For example:
  `set mongo-win-install-folder=C:\dbInstall`
  or
  `node install-dr.js --mongo-win-install-folder C:\dbInstall`
  This approach is the easiest if you just want to customize one or a few settings.
- Write installation settings in your own configuration file and use DR_CONFIG_FILE as an environment variable or command line argument to tell the script to read settings from that file. For example:
  `set DR_CONFIG_FILE=C:\temp\mysettings.json`
  or
  `node install-dr.js --DR_CONFIG_FILE C:\temp\mysettings.json`
  This approach is the easiest if you want to customize several settings.

If you use a mix of these approaches for customizing installation settings, the priority is that settings specified as command-line arguments take

precedence, followed by settings specified as environment variables, followed by settings specified using DR_CONFIG_FILE.

> Note! If the machine where the server is installed does not have internet connection the installation of PM2 will fail. In order to install PM2 in this case use another machine with the same operating system and internet access.
>
> On the online machine.
> 1.a. Install pm2 (skip this step if 'install-dr.js' script has been executed on this machine): `npm install -g pm2@3.2.2`
> 1.b. Get npm global dir: `npm config get prefix`
> 1.c. Change the directory to the directory from the step 1.b
> 1.d. Create an archive with the contents of `node_modules/pm2` subfolder
> 1.e. Add all `pm2*.*` files (e.g. pm2.cmd) from the folder in step 1.b to the archive
>
> On the offline machine.
> 2.a Copy the archive from the step 1.e
> 2.b Get the npm global dir: `npm config get prefix`
> 2.c Change the directory to the folder in the step 2.b
> 2.e Extract all `pm2*.*` files from the archive to the folder from the step 2.b
> 2.f Extract pm2 folder to `node_modules` subfolder of the folder in 2.b

5. Perform the installation by executing the command:
   `node install-dr.js`

The installation of the Design Room ONE web server is now complete, and it is ready to be deployed. Once again, the deployment settings are found in `<dr-server-install-folder>\DR_Install\dr-config.json` and you can customize the settings as described above.

6. Deploy the server by executing the command:
   `node deploy-dr.js`

This script starts up the Mongo database (unless it is already running).

Note! Design Room ONE by default uses the hostname 'localhost' to communicate with the Mongo database. In some rare cases (for example on a Windows machine with a disabled or missing network card) this hostname cannot be resolved. The workaround then is to replace 'localhost' with the IP address 127.0.0.1 in the value of the property `dr_db_url` in the file `<dr-server-install-folder>\OnPrem_Design_Room\config\server-config.json`.

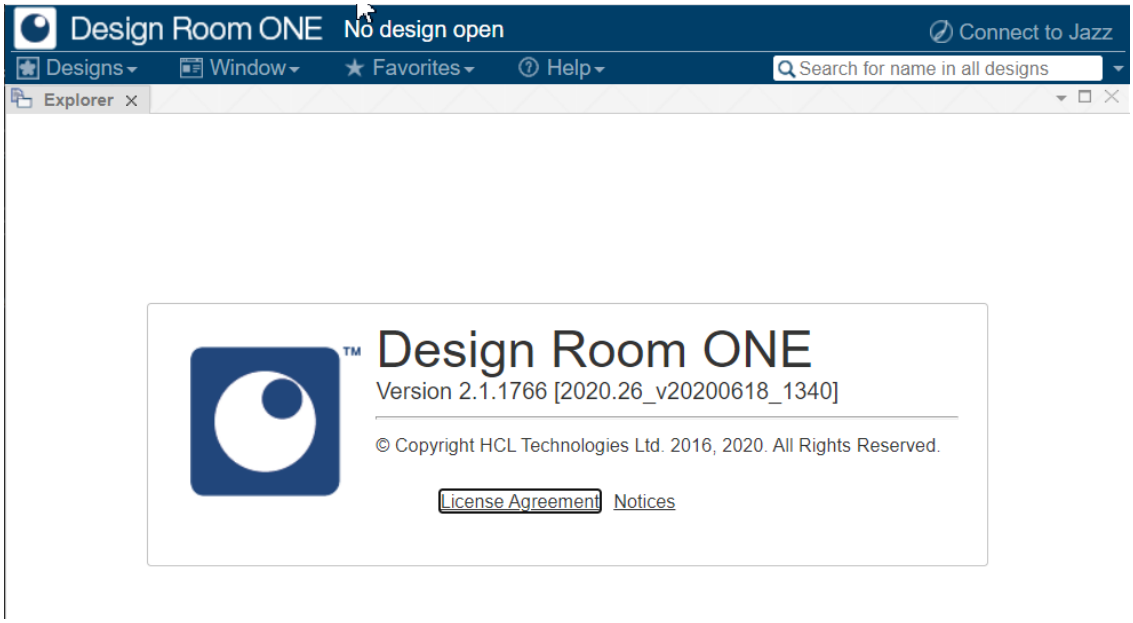
The deploy script then starts the Design Room ONE web server using the PM2 process management application. The settings for how to deploy the server are stored in the PM2 configuration file `<dr-server-install-folder>\DR_Install\pm2.config.js`. You can edit this file as needed (or write your own configuration file as described in Using PM2 for Managing the Web Server). For example, settings in this file control which SSL certificate to use (see SSL Certificate) and where to store the server log files (see Log Files). Remember that the syntax in this file is JavaScript which means that some characters, such as the backslash character (\), must be escaped. You can use forward slashes (/) to avoid escaping in path names. After you have changed a deployment setting you can just run `node deploy-dr.js` again to redeploy the server with the new settings.

PM2 provides several useful commands which you can use for managing the server. Read more about PM2 in Using PM2 for Managing the Web Server.

The deployment script finishes by printing the URL of the Design Room ONE web server. Test that the installation and deployment was successful by opening that URL in a web browser. For example, the default settings in `pm2.config.js` yields the following URL:

```
https://localhost:10101/dr/web/
```

If the web server uses a self-signed certificate (by default it does), you need to ignore the browser security warning. After that you should see the Design Room ONE application:



Note! If you run the web browser on a different machine than where the Design Room ONE web server runs, you may first need to configure the server machine so that it will accept incoming connections on the chosen port. It is common that firewalls by default block the default port 10101.

Installation Using Docker

Instructions for installing Design Room ONE using Docker can be found on Docker Hub: https://hub.docker.com/r/designroom/dr-one/. There you will also find information about how to customize the installation.

You need a working Docker environment before you can follow the instructions. The below links can be helpful:

- Download Docker: https://www.docker.com/get-docker
- How to install Docker: https://docs.docker.com/compose/install/

Customizing the Web Server

This chapter describes the various settings that control how the Design Room ONE web server runs. You can find all these settings in the configuration file `<dr-server-install-folder>\OnPrem_Design_Room\config\server-config.json`. Just like for the installation settings, you can configure the run-time settings in one of the following ways:

- Set an environment variable or use a command line argument. For example:
  ```
  set DR_PORT=10301
  ```

or

`node app.js --dr_port 10301`

This approach is the easiest if you just want to customize one or a few settings. Note that the name of an environment variable is the uppercase version of the run-time setting.

- Write installation settings in your own configuration file and use the setting `dr_config_file` to tell the DesignRoom ONE server to read settings from that file. For example:

  `set DR_CONFIG_FILE=C:\temp\mysettings.json`

  or

  `node app.js --dr_config_file C:\temp\mysettings.json`

  This approach is the easiest if you want to customize several settings.

If you use a mix of these approaches for customizing runtime settings, the priority is that settings specified as command-line arguments take precedence, followed by settings specified as environment variables, followed by settings specified using `dr_config_file`.

The best way to customize run-time settings using environment variables when you deploy the server using the scripts mentioned above, is to edit the property env in the file `<dr-server-install-folder>\DR_Install\pm2.config.js`. If you start the server manually from the command-line you can specify run-time settings on the command-line both in full and abbreviated forms. For a list of available command-line arguments and their abbreviations, go to `<dr-server-install-folder>\DR_Install\OnPrem_Design_Room` and run `node app.js --help.`

Change the Default Port or Host Name

By default the host name of the Design Room ONE web server is set to "localhost" and the port becomes 10101. If you want to change these defaults you should set the following run-time settings before starting the server:

`dr_port`	Port number to use. Note that the PORT environment variable overrides this setting, so make sure it is not set.
`dr_host`	Host name to use

Note that if you don't set-up the host name according to the name of the server where Design Room ONE is deployed, certain features (for example OSLC links) will not work.

Log Files

By default, the output printed by the Design Room ONE web server is directed to the file `<dr-server-install-folder>\DR_Install\dr-out.log` and the error messages are printed to the file `<dr-server-install-`

folder>\DR_Install\dr-err.log. You can use different log files by editing the properties out_file and error_file in <dr-server-install-folder>\DR_Install\pm2.config.js.

You can also control the format of the timestamps that are printed in these logs using the property log_date_format. Also, ensure that the run-time setting `dr_log_no_timestamps` is set. Otherwise you will see double timestamps in the log files (one timestamp printed by PM2 and another by the Design Room ONE web server itself).

Log messages are also produced by the Mongo database. By default, these messages are printed to <dr-server-install-folder>\DR_Install\mongodb.log. You can use a different log file by editing the property mongo-log-file in <dr-server-install-folder>\DR_Install\dr-config.json.

Change the Default Language

The language of the Design Room ONE web server affects the language that users see when using the web application. A user can specify a custom language to be used depending on his or her personal preference. If a custom language is not specified, the default language is English. You can change the default language by means of the run-time setting `dr_locale`. The following languages are currently supported:

Language	Code (argument to **`dr_locale`**)
English	en
German	de
Spanish	es
French	fr
Italian	it
Japanese	ja
Korean	ko
Portuguese (Brazilian)	pt-BR
Chinese (simplified)	zh-CN
Taiwanese	zh-TW
Hungarian	hu
Russian	ru

Authentication

By default, Design Room ONE does not require user authentication. This means that any user can access any data stored in Design Room ONE without logging in. However, if a design contains links to Jazz resources, such as a DOORS requirement, then the user needs to login to Jazz before the linked information can be shown.

You can configure the Design Room ONE server so that Jazz authentication also is required for accessing data stored in Design Room ONE. This is controlled by the run-time setting `dr_auth`. It can be set to the following values:

- "none"
  No authentication. This is the default.
  "jazz"
  Jazz authentication. When using Jazz authentication, several features in Design Room ONE will be blocked until the user has logged in to Jazz. For example, the following features require authentication before they can be used:
  - Opening or administrating designs
  - Searching in designs
  - Viewing or editing Jazz server preferences
- "keycloak"
  Keycloak server can be integrated to provide authentication and access control mechanisms for Design Room ONE. Please refer to Authentication Setup document on detailed configuration instructions.

Jazz Configuration Preferences

The following run-time settings allow you to set the configuration preferences that specify how Design Room ONE should connect to Jazz:

- `dr_jazz_root_url`
- `dr_jazz_consumer_key`
- `dr_jazz_consumer_secret`

Specifying them in the server configuration file has the same effect as setting them in the Design Room ONE web application (on the Preferences page). You can use this possibility if you have configured the server to use Jazz authentication (see Authentication) and prefer to register Design Room ONE with Jazz using the manual approach described in Manual Registration.

SSL Certificate

The Design Room ONE web server uses HTTPS, and you need to decide which SSL certificate it should use. You have two alternatives:

1) Use your own SSL certificate. This is the recommended approach for a Design Room ONE installation intended for a production environment. The connection to the server will be secure and users will not get any warnings from the browser.
   The location of the certificate files is controlled by these two run-time settings:

`dr_ssl_key`	Path to key file
`dr_ssl_cert`	Path to crt file

Also remember to set `dr_host` to the host name that is expected by the certificate you use (see Change the Default Port or Host Name).

Depending on which certificate authority (CA) that has issued your certificate you may have to import the certificate into your Java Runtime Environment (JRE) to allow RSAD/RTist/RSARTE to connect to the Design Room ONE server. The following command can be used for doing this:

```
<java path>\jre\bin\keytool -importcert -file <certfile> -alias
<certalias> -keystore <java path>\jre\lib\security\cacerts -storepass
changeit -noprompt
```

`<certfile>` specifies the certificate to import, `<certalias>` is an alias you chose, and `<java path>\jre` should specify the JRE that you use for running RSAD/RTist/RSARTE.

2) Use a self-signed SSL certificate. You can use this approach if you don't yet have your own SSL certificate to use. However, in this case users will get warnings from the web browser about the use of a self-signed certificate as the server will not be considered secure. And when exporting models to the Design Room ONE server from RSAD/RTist/RSARTE you must mark the "Allow insecure connection" checkbox on the first wizard page.

   The Design Room ONE installation contains a self-signed certificate in `<dr-server-install-folder>\OnPrem_Design_Room\config\cert`. The web server will use that certificate by default when `dr_ssl_key` and `dr_ssl_cert` have not been set.

Database Location

For best performance, it's recommended to run the Mongo database on the same machine where the Design Room ONE web server runs. However, it is also possible to run it on a different machine. In that case, you must set the following:

`dr_dr_url`	Database connection URI in the format mongodb://<DB_SERVER>:<DB_PORT>/<DB_NAME> for example mongodb://localhost:27017/dr

If you have more than one Design Room ONE installation that uses the same database you must ensure they use different database names.

Running Multiple Web Servers

You can start more than one instance of the Design Room ONE web server if you like to. The servers can either use separate databases or share the same database. You can specify an additional web server instance to be launched by adding an entry to the apps attribute in the file `<dr-server-install-folder>\DR_Install\pm2.config.js`.

For example, you can use a dedicated server when exporting models from RSAD/RTist/RSARTE, and another server when accessing the models from web browsers. Both these servers can share the same database. With this configuration users of the web application won't notice any performance degradation while a big model is exported to the server.

Normally it's necessary to use different ports if running multiple Design Room ONE servers on the same machine. However, if you run the server in PM2 cluster mode (see Using PM2 for Managing the Web Server) all server instances will share the same port.

Using PM2 for Managing the Web Server

Design Room ONE uses PM2 as process management application. It provides several useful features:

- It can automatically restart the server in case it terminates unexpectedly (to minimize the down time)
- It can manage the logs produced while running the server (writing logs to files in custom formats)
- It allows you to configure all aspects of server deployment in a single file (useful for example to automate the steps needed to deploy a new version of the server)
- It can distribute the deployment of the server onto multiple CPUs in a cluster to increase performance and perform load balancing
- …and much more

The default configuration file for PM2 is `<dr-server-install-folder>\OnPrem_Design_Room\pm2.config.js`. You can modify this file per your specific needs. If you instead prefer to create a different configuration file, you can update the pm2-config-file setting in `<dr-server-install-folder>\DR_Install\dr-config.json` to reference your own configuration file.

The deployment script `deploy-dr.js` starts the Design Room ONE web server using the specified PM2 configuration file. After that you can use PM2 commands directly to manage the web server application. Here are a few useful PM2 commands you can use for managing the Design Room ONE web server:

`pm2 list`	Display the status of the server (e.g. whether it is running or not).
`pm2 describe DesignRoomONE`	Print detailed information about the server.
`pm2 stop DesignRoomONE`	Stop the server.
`pm2 start pm2.config.js`	Starts the server with settings provided in the specified config file.
`pm2 delete DesignRoomONE`	Stop the server and remove it from the list of processes managed by PM2.

See the PM2 documentation for the full list of commands that are available.

Shutting Down the Design Room ONE Server

To shut down the Design Room ONE server you can just send the SIGINT signal to it. This happens automatically if you use the PM2 commands for stopping or restarting the server (see Using PM2 for Managing the Web Server).

When the server receives the SIGINT signal it will not shut down immediately. First it will try to complete the request it's currently processing. This ensures that when you run the server in PM2 cluster mode, you can perform a server restart without any downtime. However, if the current request takes a very long time to complete, the server will force a shutdown after a certain amount of time. You can configure this timeout using the property dr_shutdown_timeout. By default it is set to 5 seconds.

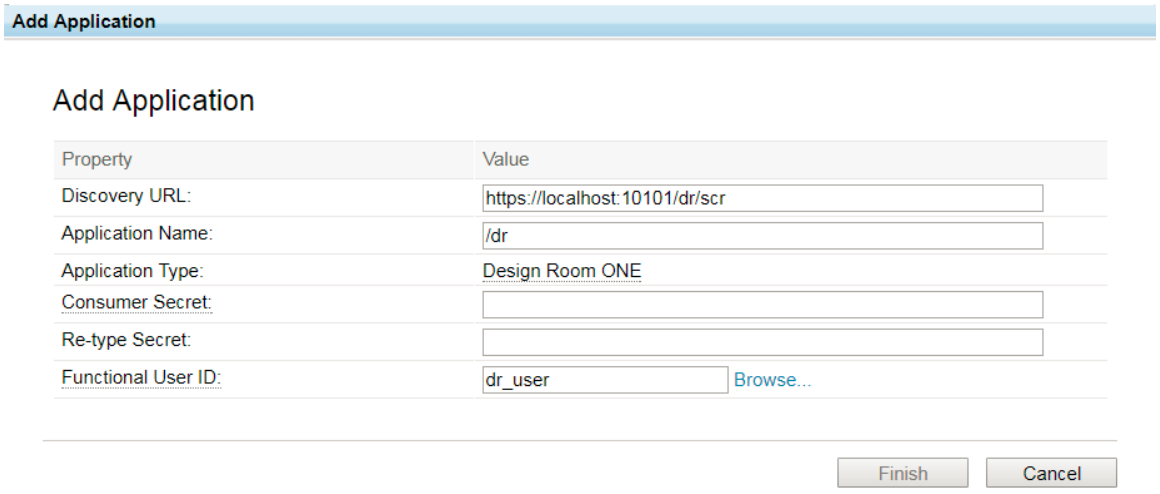
Registering Design Room ONE with Jazz

Several features of Design Room ONE requires that the application has been registered with a Jazz Team Server. One example of such a feature is the creation of links from Jazz CLM tools to Design Room ONE. Registering Design Room ONE as a Jazz application enables the Jazz CLM tools to call the Design Room ONE APIs and vice versa. The registration steps below need only be performed once, and it is therefore recommended that a Jazz administrator performs them right after the Design Room ONE installation is completed.

> Note! There are two ways to register Design Room ONE with Jazz. One quick and automatic way using the "Add Application" wizard in Jazz, and another more manual approach where several settings are made individually both in Jazz and Design Room ONE. The automatic approach is more convenient, but it may prevent some features in Jazz from fully working. For example, life cycle projects cannot be created. If you need to use such Jazz features, you have to instead register Design Room ONE using the manual approach.

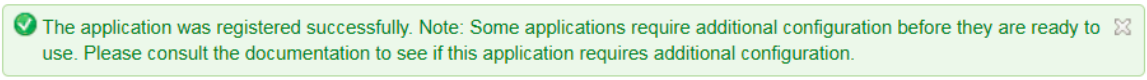
Automatic Registration

1. Make sure that the Design Room ONE server is running. The Jazz Team Server needs to communicate with it during the registration process.
2. Go to the Jazz Team Server administration page (https://<host>:9443/jts/admin) and click on the Manage Server link.
3. In the Configuration section to the left click on Registered Applications.
4. Click the Add… link to add a new registered Jazz application.
5. Enter the discovery URL (`/dr/scr`) of the Design Room ONE server. With default deployment settings it will look like this:

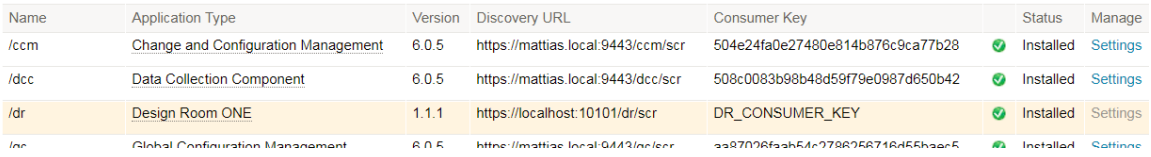


6. Enter a password as Consumer Secret and retype it in the next field. Also delete the Functional User ID (“dr_user”) and instead click Browse and select an existing administrator user id. Leave the default values for all other fields.
7. Press Finish.

If registration was successful you should see a message box

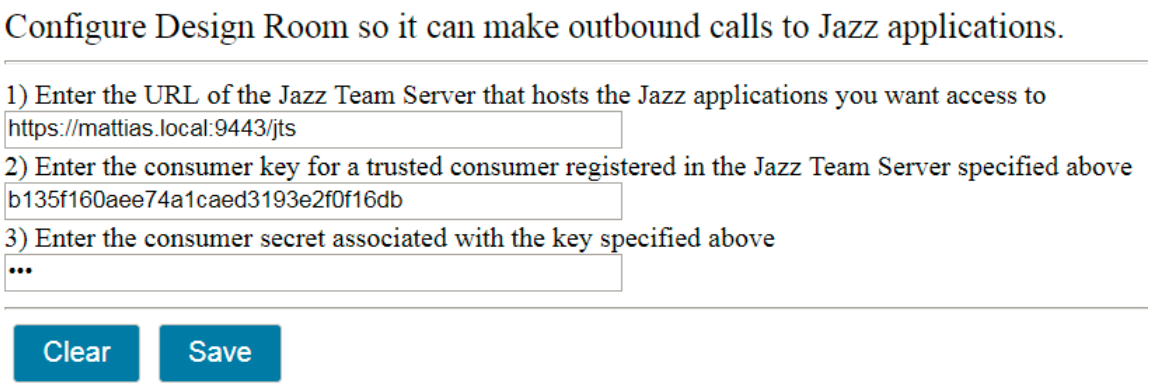


and an application with the name /dr should appear in the list of registered applications.



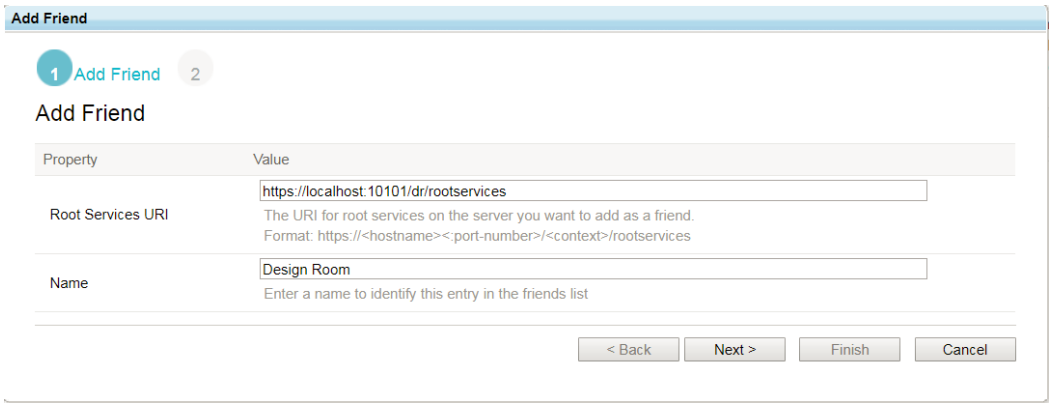
You can verify that Design Room ONE was correctly registered by looking at its preferences (the default URL is

https://localhost:10101/dr/web/html/jazzConfig.html). You should see populated values for the Jazz configuration preferences. For example:

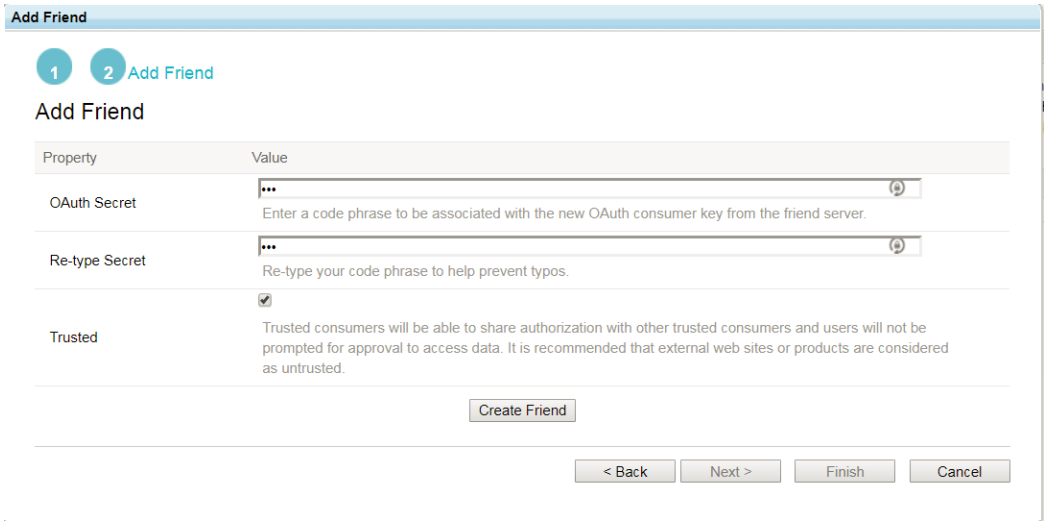


Manual Registration

1. Create a friend entry in the Friends (Outbound) list. Design Room ONE follows the convention of Jazz CLM products and exposes its root services document at /dr/rootservices.

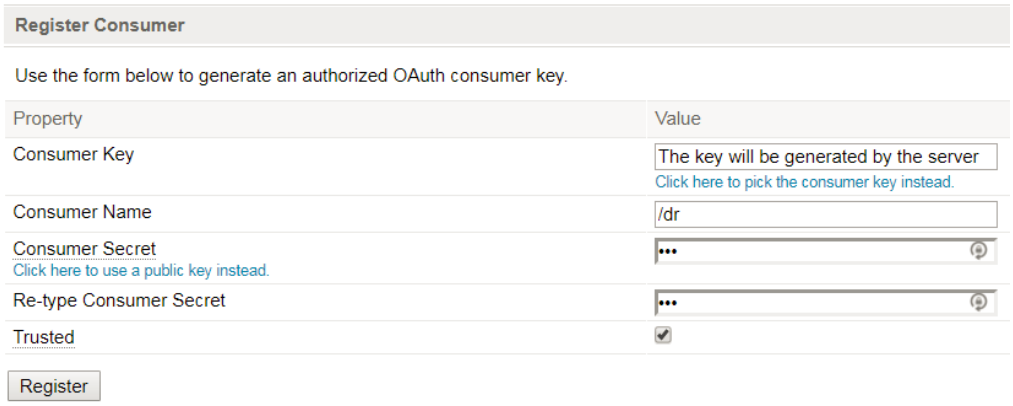


Enter a password for OAuth Secret and repeat it in the next field. Make sure the friend is marked as “trusted”.



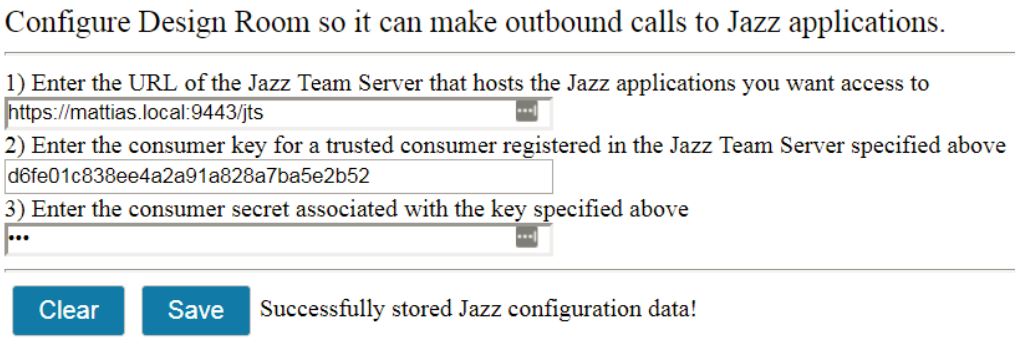
2. Press the Create Friend button, then Next and finally Finish.

3. Create an authorized keys entry in the Consumers (Inbound) list. Give it the name '/dr' and enter a password in Consumer Secret (you can use the same as before). Retype the password in the next field. Also mark the Trusted checkbox.



4. Press the Register button to register the consumer. Mark the consumer key that appears in the popup and copy it to the clipboard.

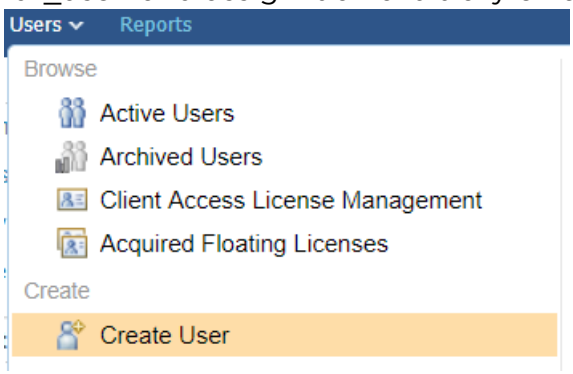
5. Open the Preferences page of Design Room ONE and enter the URL of the Jazz Team Server in the first field. Then paste the consumer key you just created in the Jazz Team Server into the second field on the preference page. In the third field type the consumer secret you chose for the key.



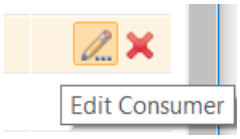
Press Save to store the entered data in Design Room ONE.

Note that if Jazz authentication is enabled (see Authentication) you cannot edit the Jazz preferences in Design Room ONE. Then you may instead specify them in the server configuration file (see Jazz Configuration Preferences).

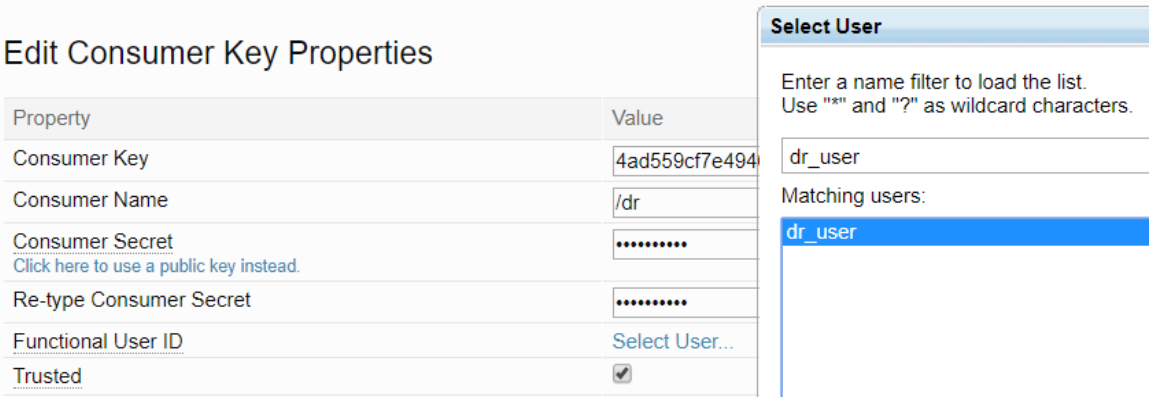
6. Create a functional user for Design Room ONE. Use the command Users – Create User in the Jazz Team Server. Give the user the name ‘dr_user’ and assign it an arbitrary email address.



7. Go back to the Consumers (Inbound) list and click the Edit Consumer button for the consumer you just created.



Click Select User and select ‘dr_user’ from the list of users and press the Add and Close button followed by Finish.



Installing the Design Room ONE Integration Plugin in Eclipse-based Modeling Environment

Design Room ONE can be integrated with the following Eclipse-based products:

- Rational Software Architect Designer (RSAD)
- HCL RTist

- Rational Software Architect Real-Time Edition (RSARTE)

Note! Before proceeding with installation in RSAD (but not in RTist/RSARTE) make sure it does not have the following components installed:
   a. Development modeling
   b. Design Room ONE and Requirements Management Integration
   c. Requirements Management Integration.

If you have any of these components they must be uninstalled before installing Design Room ONE Integration. The reason for this is that Design Room ONE Integration plugin includes a Requirements Management Integration feature that allows you to visualize DOORS requirements in RSAD/RTist/RSARTE, and to create links from model elements to such requirements. An older version of this feature is also included as a part of listed optional components in the RSAD and it is

Follow the steps below for installing the Design Room ONE Integration plugin:

1. Start Eclipse-based modeling environment with sufficient administration rights (so that you are allowed to modify the its installation directory).
2. Perform *Help – Install New Software*
3. Click the "Add" button. The "Add Repository" dialog appears. Type a name for the installation repository in the "Name" field, for example the current date.

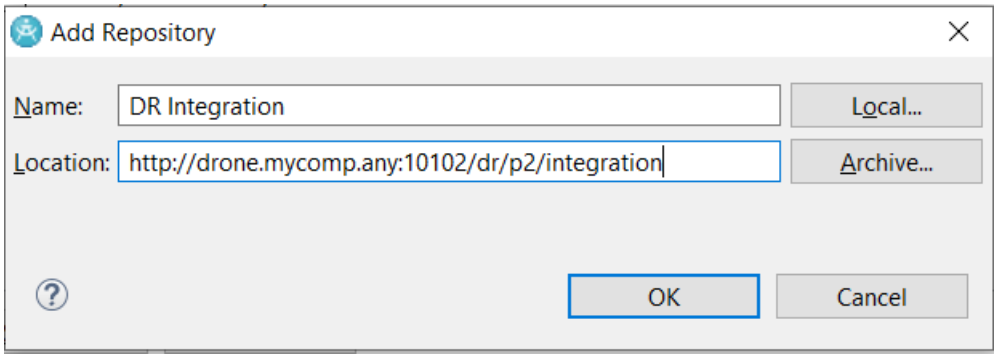
You now have two alternatives for how to specify the location of the Design Room ONE Integration plugin. The recommended and easiest approach is to install the plugin by using the Design Room ONE web server:

4a. In the "Add Repository" dialog, type one of the following URL in the "Location" field:

    `http://<host_name>:<http_port>/dr/p2/integration`

    Change the host name and port to match the `dr_host` and `dr_port` run-time settings used by the Design Room ONE web server. Note that this installation URL uses http instead of https. Also, note that the port number used in this URL is one above the regular port

number used by the Design Room ONE web server. Then press Add.



The other alternative is to install from a ZIP file containing the plugin.

4b. The ZIP file should be created from the contents of the following folder in the Design Room ONE installation:

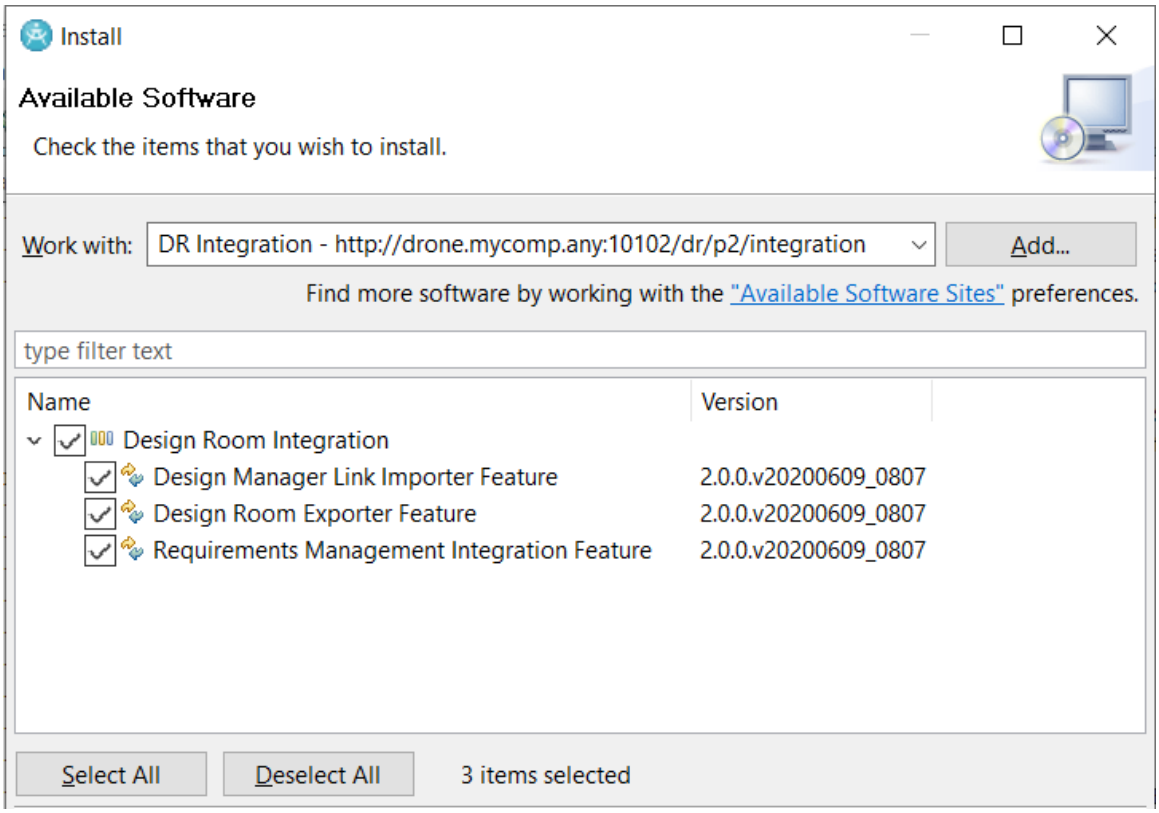
    `<dr-server-install-folder>\OnPrem_Design_Room\p2\integration`

    This ZIP file then needs to be made available to each modeling environment user, for example by placing it in a shared location on your intranet.

    To install from such a ZIP file, in the "Add Repository" dialog, click the "Archive" button and browse to the ZIP file that was created. Then press Add.

The remaining steps are the same regardless of which of the above two alternatives you choose:

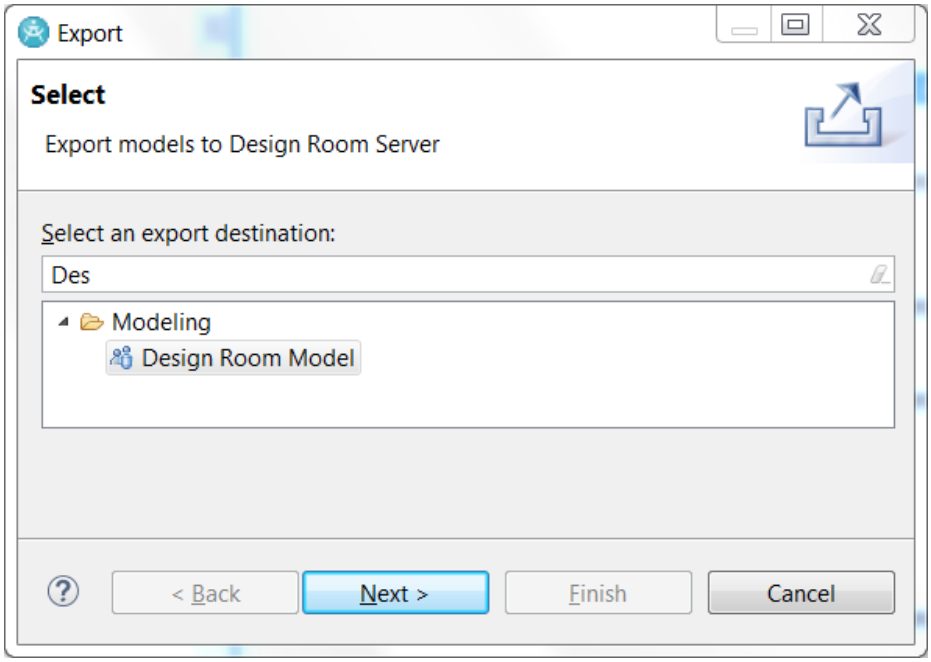
5. You should now see "Design Room Integration" in the feature list (the version will be newer). Select it.



In case of problems with update site you can try re-running installation after starting your modeling environment with clean option:

```
<path-to-RSAD/RTist/RSARTE>\eclipse -clean
```

6. Finish the wizard by going through all steps. You need to accept the license agreement.
7. When the wizard is finished, the installation will start. Please wait a couple of minutes.
8. When the installation is finished, you will be prompted to restart the Modeling environment. Do so.
9. When Modeling environment has started up again, verify that the installation was successful by performing *File – Export*. You should see *Modeling – Design Room Model* in the list of exporters.



Updating a Design Room ONE Installation

To update an existing Design Room ONE installation to the latest version, follow these steps:

1. Perform steps 1 and 2 mentioned in Installing and Deploying the Web Server to unzip `<dr-server-install-folder>`. Make sure to extract the files into an empty folder. Do not overwrite an existing Design Room ONE installation, especially not if it is currently running.
2. If you want to delete the old installation folder you need to perform this step to ensure that no process is running that will prevent that folder from being deleted.
   Open an administrative command shell and perform
   `pm2 kill`
   Enter the Mongo shell by running `<mongo-install>\bin\mongo`
   where `<mongo-install>` is specified either by **`mongo-win-install-folder`** or **`mongo-linux-install-folder`** in the file `<dr-server-install-folder>\DR_Install\dr-config.json`.
   Run the following commands in the Mongo shell for stopping the Mongo database:
   ```
   use admin
   db.shutdownServer()
   ```
3. Ensure that the version of Node.js is the same in the new version. Go to a command prompt and perform the command:
   `node --version`
   Compare the output of this command with the version of Node.js that

is in the new installation. On Windows the Node.js installer is located in `<dr-server-install-folder>\DR_ReleaseManagement\server-sw\win32_x86_64\`. On Linux it's located in `<dr-server-install-folder>\DR_ReleaseManagement\server-sw\linux_x86_64\`. If the version is not the same, you should uninstall Node.js (using the normal procedure for uninstalling installed programs) and then install the new Node.js version.

4. If you have changed properties in `<OLD_dr-server-install-folder>\OnPrem_Design_Room\config\server-config.json` file apply the same changes to `<dr-server-install-folder>\OnPrem_Design_Room\config\server-config.json`

   > Note! If you are updating Design Room ONE version 2.0 or earlier to Design Room ONE version 2.1 or later consider that in newer versions `dr_db_url` property is used to control database connection by default. If you want to continue using `dr_dbhost, dr_dbport` and `dr_db` properties you need to delete `dr_db_url` property from `server-config.json` file.

5. Open an administrative command shell and perform
   ```
   cd <dr-server-install-folder>\DR_Install
   node deploy-dr.js
   ```

The deployment script will first stop the currently running Design Room ONE web server. However, it does not terminate the pm2 daemon process nor the Mongo database. That's the reason why you need to perform step 2 above if you want to avoid lingering processes that prevent the old installation folder from being deleted.

The deployment script then starts the new version of the server. The version of Node.js, Mongo and PM2 will remain the same.

Repeat the steps mentioned in Installing the Design Room ONE Integration plugin in order to update the Design Room ONE Integration plugin to the latest version. Eclipse will detect that an older version of the plugin was already installed and update it as needed. Note that Eclipse provides preferences in *Install/Update – Automatic Updates* which allow you to automatically update the Design Room ONE Integration plugin when a new version becomes available.

Uninstalling Design Room ONE

There is currently no script for uninstalling the Design Room ONE web server, so you need to perform the following steps manually (from an administrative command prompt):

1. `pm2 delete DesignRoomONE`
   Stops the Design Room ONE web server and deletes it from PM2 list of managed processes.
2. `pm2 kill`
   Stops the PM2 daemon process.
3. `npm uninstall -g pm2`
   Uninstalls PM2. If it is still available in the PATH after this command, you have to manually remove the PM2 files. Use `where pm2` (Windows) or `which pm2` (Linux) to find out where it is located.
4. Enter the Mongo shell by running `<mongo-install>\bin\mongo`
   where `<mongo-install>` is specified either by **`mongo-win-install-folder`** or **`mongo-linux-install-folder`** in the file `<dr-server-install-folder>\DR_Install\dr-config.json`.
   Run the following commands in the Mongo shell for stopping the Mongo database:
   ```
   use admin
   db.shutdownServer()
   ```
5. On Windows you can uninstall Mongo using the command
   `msiexec /uninstall <dr-server-install-folder>\DR_Install\downloaded\mongo-win.msi`
   On Linux the commands for uninstalling Mongo depend on your Linux distribution; see the Mongo documentation.
6. Now you can remove the Design Room ONE server by deleting the folder `<dr-server-install-folder>`.

To uninstall the Design Room ONE integration plugin run the command *Help – Installation Details* in RSAD/RTist/RSARTE. In the "Installed Software" tab mark the features you want to uninstall:

- Design Manager Link Importer Feature
- Design Room Exporter Feature
- Requirements Management Integration Feature

Then press the Uninstall button. You will be prompted to restart RSAD/RTist/RSARTE afterwards.