OBEO
7 boulevard Ampère
44470 Carquefou
France
support@obeo.fr

Team for Capella

INSTALLATION GUIDE

This document is the property of THALES.

CONTENTS

LIST OF TABLES

1 SCOPE

1.1 TYPOGRAPHIC AND NOTATION RULES

To improve legibility, some text elements are identified by specific typographic rules in according to their tutorial purpose.

- **Emphasis** font is applied to emphasize words which designs controls (Click on **Cancel**)
- `Fixed` font is applied to expressions and texts of ASCII (`C:\My documents\Path`)
- *Terminology* Font is applied on expressions referenced in Terminology table.

📝	Points out- information useful for the user. Clarifies a detail.	⚠	Indicates warning information.
🚩	Indicates a potential pitfall or operating risks.	ℹ	Gives information in answer to expected user's question.
🛑	Stop and read before going on.	📕	Refers to an external document.

1.2 IDENTIFICATION

1.2.1 DESCRIPTION OF THE PRODUCT

Team for Capella is a collaborative solution to have several contributors working on the same model, with the granularity as fine as a model element.

It decouples the versioning issue which is ensured by the SCM tool from the concurrent accesses issue.
It introduces a shared repository which is populated from the SCM tool and which enables several users to work on the same model.
Team for Capella is composed of three parts:

- A server, so as to manage the models repository, and associated features (such as locks, etc.);
- An administration module, to schedule automatic backups of the models and the database, add/remove users, and handle existing locks;
- An add-on, packaged as an update site, to bring the multi-user functionalities on top of the standard Capella rich client.

Team for Capella is available in 64 bits versions.

1.2.2 APPLICABILITY

This guideline is applicable to following versions of Capella and Team for Capella:

Value name	Value
AppChk	[INSTALLDIR]\capella\eclipse\eclipse.exe;
Product Name	Capella x64 / Team for Capella x64
Product Version	1.3.2

1.3 DOCUMENT OVERVIEW

This document is intended for persons in charge of installing Team for Capella.

It describes the nominal installation, configuration and uninstallation of Team for Capella.

2 REFERENCED DOCUMENTATION

Title	Version
Embedded documentation within Team for Capella	1.3.2
Capella Online Installation Guide	1.0.1
Capella Release Notes	1.3.2
Team for Capella Release Notes	1.3.2

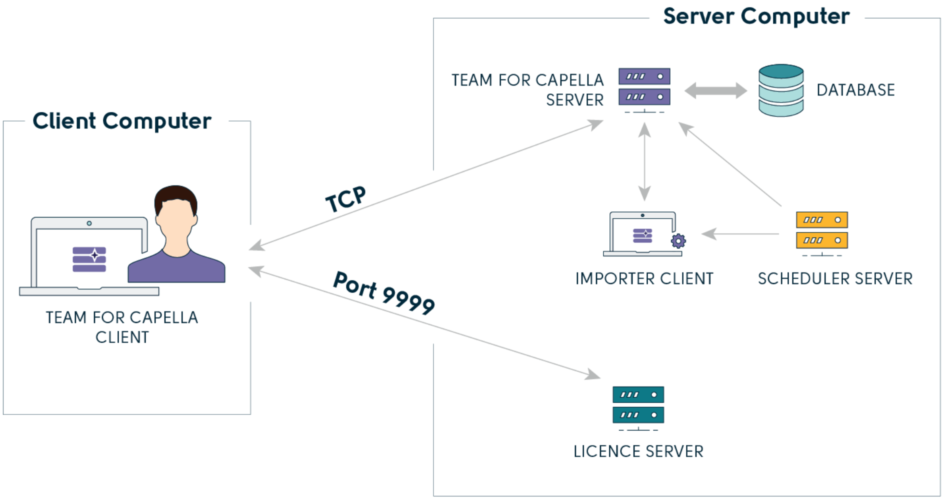
Table 1 : Reference Documents

3 TEAM FOR CAPELLA SERVER INSTALLATION

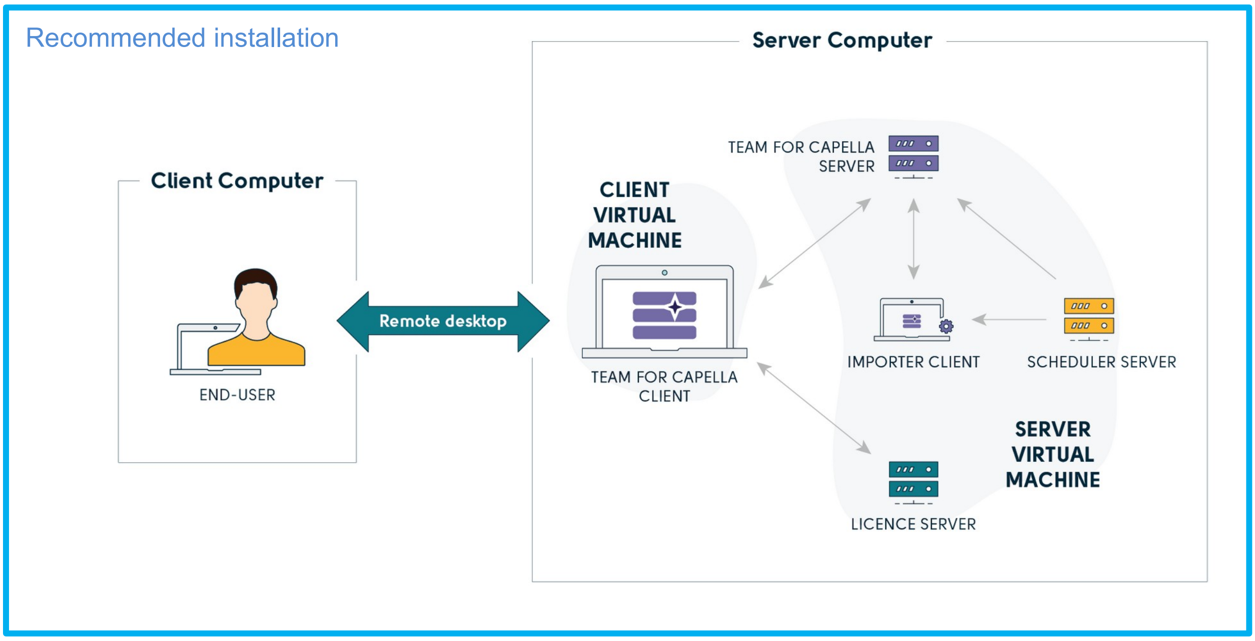
3.1 INSTALLATION ARCHITECTURE

There are two main ways to install Team for Capella:

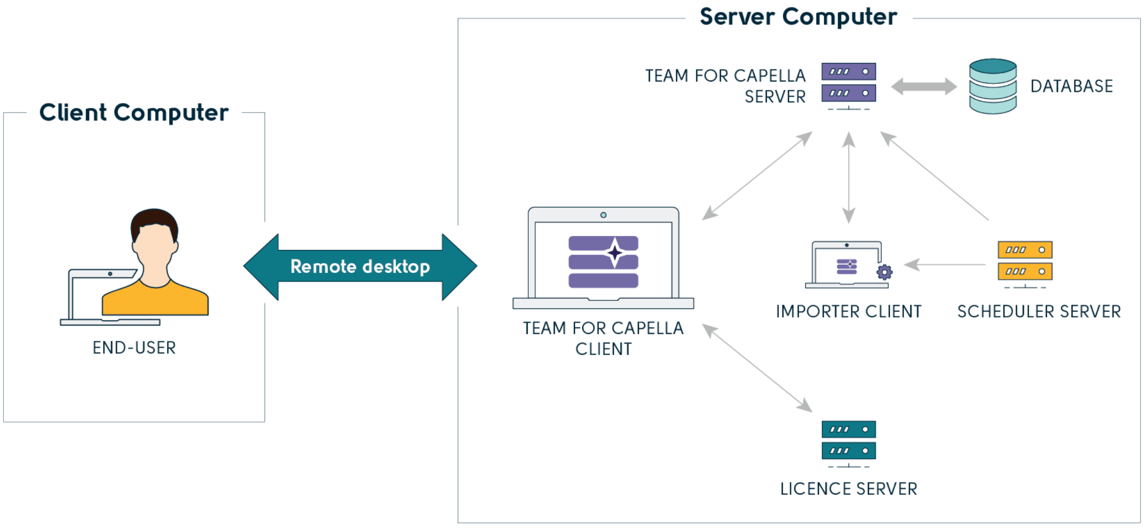
- Local clients:
  - 1 server computer runs the Team for Capella Server and Scheduler,
  - n client computers run the Team for Capella Clients,



- Remote clients: the same computer runs the Team for Capella Server and Scheduler and n Team for Capella Clients (users connect to this computer using RDP). This installation is recommended when the network's bandwidth is not very good. In this configuration, the client and server can either be installed on the server computer or separated in two virtual machines. The latter is the recommended installation
  - Client and server separated in two virtual machines



- Client and server on the same server machine



3.2 RECOMMENDED SYSTEM REQUIREMENTS

3.2.1 SERVER COMPUTER RECOMMENDED SYSTEM REQUIREMENTS

It is mandatory to run Team for Capella Server and Scheduler on the same computer.

For successful installation of Team for Capella Server, your computer must meet the following requirements:

- Local client installation (server side only):
  - 2 GHz processor,
  - RAM: 4 GB for the Team for Capella server + 3GB for the importer client
  - 15 GB of available hard disk space,
- Remote client installation:
  - 64 bits installation,
  - Multicore processor (2GHz)
    - 2 cores for Team for Capella Server, scheduler and license server
    - 1 core per running Team for Capella Client,
  - RAM:
    - 4 GB for the Team for Capella server + 3GB for the importer client
    - 3 GB per Team for Capella Client,
  - 15 GB of available hard disk space + 2 GB per Team for Capella Client,
  - 2 hard drives are recommended:
    - The first containing system files and software installation files (a SSD hard drive is mandatory if more than 8 users),
    - The second containing the Team for Capella Server files,

- System requirements: Windows 7/8/10, Windows Server 2008/2012/2016,
- Java Runtime Environment 8u121,
- Team for Capella database must be stored on a local hard drive,
- Security policies:
  - Virus scanner:
    - Team for Capella Server database files must not be scanned (*.db).
    - In addition, it should not be activated either on Capella models files: *.aird, *.melodymodeller, *.airdfragment, *.melodyfragment and *.afm,
    - The license server hosts a collection of licenses stored in several encrypted .ols files. Those licenses pools must not be scanned.
    - Periodic analyses should not be launched when users are working (launch them at night),
  - Firewall:
    - At least 2 ports must be opened, the Team for Capella Server port (by default 2036), the Scheduler port (by default 8036),
    - In addition, the license server port must also be opened (by default 9999).
- The computer should be fully dedicated to Team for Capella.

3.2.2 Client computer recommended system requirements (only for local client installation)

- 64 bits installation
- 2 GHz processor
- 3 GB for Team for Capella client
- Java Runtime Environment 8u121
- Security policies:
  - Virus scanner:
    - It should not be activated either on Capella models files: *.aird, *.melodymodeller, *.airdfragment, *.melodyfragment and *.afm,

3.3 DEPLOYMENT RECOMMENDATIONS

3.3.1 NETWORK

3.3.1.1 Latency: Client and Team Server

It is recommended to provide a network with the lowest possible latency between the client and the server: in the order of 1 to 10 ms for a round-trip.

3.3.1.2 Latency: Team server and DB server

It is strongly recommended that the Team server and the DB server are located on the same physical server as latency between the Team server and DB server will impact greatly the overall performances of the solution. As such the best performing deployment is achieved by using the H2 database in embedded mode with its .db database file located on the same disk than the Team server.

If there is a requirement on the database that prevents from using H2, make sure that the latency is as low as possible.

3.3.1.3 Network stability

VPN are not recommended (it is a latency factor) as well as other network elements that could drop connections which are more or less inactive. As such wireless connection are also not recommended as any loss of connectivity might lead to instability in the product and loss of data. However, if a network element of this kind is mandatory, an SSH tunnel could be used as a workaround to avoid client/server disconnections.

3.3.1.4 Server isolation

It is strongly discouraged to deploy the server on a public WAN. Team for Capella should be the only way to edit the information stored in the database.

3.3.2 SCALABILITY AND SIZE OF MODELS

Scalability and performances are highly dependent on the design of the domain metamodel, the implementation of this metamodel and the Viewpoint Specification Models. The following figures are given with an Ecore model and the EcoreTools tooling which applies the Sirius best practices.

The minimum physical memory dedicated to the Team server is 4 GB for a deployment where the expected model size is in the order of 300 000 model elements. The heap memory available for the server should be increased to support bigger models 8GB should support 600 000 model elements.

The memory usage of the clients will increase when the model which is shared among the clients grows as such these resources might need to be increased for larger models with 8GB being expected for models with 600 000 elements (the exact value might vary depending on the amount of information each model element holds).

The latency of end user operations requiring a full analysis of the model increase as the model grow, this includes: opening and closing a project, deleting model elements and representations, launching a transformation or a code generation. Opening a project (and hence collecting the model from the network) might take up to 1 min for a model with 500K elements.

Models having 1 000 000 model elements are the considered the upper limit for a Collaborative Server deployment.

A given server is expected to be used by 10 to 20 users simultaneously depending on their level of activity.

3.3.3 Disclaimer

Notwithstanding what was stated previously, Team for Capella product is not warranted to run without any error or interruption. Obeo does not make any warranty regarding the statements that are under the chapter «Deployment Recommendations», this chapter is provided for information purposes.

You acknowledge and accept the risks involved in using these products which could include without limitation, down time, loss of connectivity or data, system crashes, bad performances or performance degradation.

3.4 INSTALLATION PROCEDURE

Java must be setup properly and added to the system PATH.

The date/time and the time zone of the server must be correct to make the scheduler work as expected.

To use existing models in a new version of Team for Capella, copies of these models have to be kept (in files format) before removing the old version. Once the new version is installed, the migration procedure will be performed on these models.

3.4.1 Team for Capella Server installation procedure

3.4.1.1 Installation

Preparation steps:

1. Extract the archive `TeamForCapella-1.3.2-win32.win32.x86_64.zip` in a given directory. It contains a `TeamForCapella` directory with 4 sub-folders and 2 files:
   - `capella`: contains an eclipse folder with some batch and properties files
   - lic-server: contains a floating license server which allows several users to share the same product licenses. Each license can be used by only one user simultaneously.
   - `scheduler`: contains Jenkins and pre-configured jobs
   - `server`: the Team for Capella server
   - `updateSite`: the Team For Capella update site for the client
   - `license.html`
2. Download Capella 1.3.2 Bundle (windows 64 bits, https://www.eclipse.org/capella/download.html)

Client Installation

1. Unzip Capella bundle and rename the folder to `capella`.
2. Move the `capella` folder into the `TeamForCapella` folder.
3. Structure of `TeamForCapella\capella\eclipse`
   - `configuration`
   - `eclipse.exe`
   - `installTeamForCapellaInCapella.bat`
   - …
4. Launch `installTeamForCapellaInCapella.bat`

The installation script will install the Team for Capella features in Capella, update the splash screen and update some properties in `eclipse.ini`.
It is configured by default to retrieve the Team For Capella update site in folder `TeamForCapella\updateSite`, the -*`repository`* property can be updated in the script to reference it from another location.

This Capella client (`capella` folder) should be used only for the Scheduler jobs: it must not be moved or renamed as the .exe and .bat scripts are referenced from the pre-configured jobs.

It can also be zipped and provided to user in case of local client installation, see #4.2. Installation Procedure.

In remote clients installation, you need to **copy the full** `capella` **folder** and **rename it into** *`capella_client`.* Then this client can be started on Windows Server and accessed with Remote Desktop. If you want to install additional functionality, it will have to be done on `capella_client` and will not impact the *`capella`* folder.

3.4.1.2 Extensions installation

If meta-model extensions or add-ons are needed, use **one** of the following ways to install them:

- Either unzip/copy their binary files in the folder `TeamForCapella\capella\eclipse\dropins`
- Or:
    - Unzip/copy them in any folder (it can be a shared folder between this server installation and client installations)
    - Modify the configuration file `TeamForCapella\capella\eclipse\eclipse.ini` by adding the following parameter, **after -vmargs** :

```
-Dorg.eclipse.equinox.p2.reconciler.dropins.directory=<ExtensionFolder>
```

Exactly the same extensions have to be installed on the server (importer client) **and on all clients**.

3.4.1.3 Checks on environment variables

Before starting the scheduler, the following check has to be done on environment variables:

- **At least one** of the two following variables has to be defined and valid:
    - The `JRE_HOME` variable must point to a JRE installation,
    - The `JAVA_HOME` variable must point to a JDK installation.

If both variables are defined, `JRE_HOME` is used.

3.4.2 How to install Scheduler as a Windows Service (recommended)

1. Go to the `scheduler` directory and launch `winservice.bat`, it will install the service `TeamForCapellaScheduler`,

Do not forget to check environment variables (see part 3.2.1.3) before launching “winservice.bat”.

2. Run `services.msc` from a command prompt, the service `TeamForCapellaScheduler` should be visible,
3. Double click on it to open the service configuration dialog,
4. Set the `Startup type:` parameter to `Automatic` (the service will be started with the computer),
5. In the `Log On` tab, configure the Windows account that will run the service. This account must have correct access rights to allow the Scheduler and the Team for Capella Server to access their data files,
6. If needed, start the service.

3.4.3 LICENSE SERVER INSTALLATION

3.4.3.1 *Server installation*

1. The license server is provided in the `TeamForCapella-1.3.2-win32.win32.x86_64.zip` archive. After the preparation steps (see section 3.4.1.1), it is located in `TeamForCapella\lic-server`
2. Unzip the `OLS.zip` archive in `TeamForCapella\lic-server\OLS`, the `OLS` folder should contain 4 `.ols` files.
3. You can choose to:
   - either launch the license server from the sheduler’s job `Start license server` (disabled by default)
   - or directly launch the tool using the comand `lic-server -keys ./OLS -verbose`

Additional parameters and documentation can be found in `DOCUMENTATION.txt` (located in `TeamForCapella\lic-server\`).

The server configuration and more especially the `.ols` files should not be modified, moved or even accessed while the server is operating. Stop the license server before doing such operation.

3.4.3.2 *Client configuration*

In order to connect Team for Capella instances with the license server, a connection key must be used to retrieve its address and port.

You should have received this key with the license server bundle and your `.ols` files. If not please contact the support and provide the IP address and port to use to connect to the machine that will host the server in your local network.

Add the following line to define the configuration of the server to your `capella\eclipse\eclipse.ini` file:

```
-DOBEO_LICENSE_SERVER_CONFIGURATION=<connection key>
```

This must be done on all clients.

A license token is retrieved at the first connection attempt done by Team for Capella. It is then revoked when the last connected Capella project is closed or when Team for Capella closes. This license is verified from times to times with the server while you are using Team for Capella.

Clients are programmatically throttled and will not send requests to the server more frequently than once every minute. This throttling means some of the client actions might have a delay before being distributed to the server, for instance if one user stops using a given feature, 2 minutes of delay at most can be necessary for another user to be able to get the token.

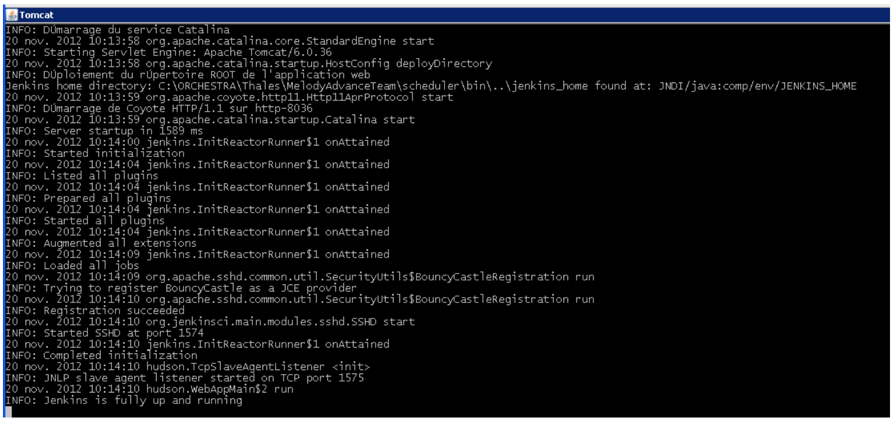
The client/server communication is request based, no connection is kept alive for longer than just a request.

3.4.4 INSTALLATION VERIFICATION

1. Launch `scheduler.bat` from the scheduler folder,

Do not forget to check environment variables (see part 3.2.1.3) before launching “scheduler.bat”.

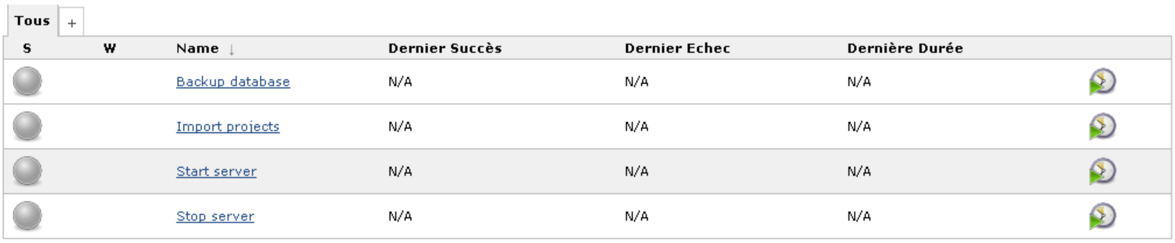
Once “scheduler.bat” file has been launched, you should get the following command window:



```
INFO: DÚmarrage du service Catalina
20 nov. 2012 10:13:58 org.apache.catalina.core.StandardEngine start
INFO: Starting Servlet Engine: Apache Tomcat/6.0.36
20 nov. 2012 10:13:58 org.apache.catalina.startup.HostConfig deployDirectory
INFO: DÚploiement du rÚpertoire ROOT de l'application web
Jenkins home directory: C:\ORCHESTRA\Thales\MelodyAdvanceTeam\scheduler\bin\..\jenkins_home found at: JNDI/java:comp/env/JENKINS_HOME
20 nov. 2012 10:13:59 org.apache.coyote.http11.Http11AprProtocol start
INFO: DÚmarrage de Coyote HTTP/1.1 sur http-8036
20 nov. 2012 10:13:59 org.apache.catalina.startup.Catalina start
INFO: Server startup in 1589 ms
20 nov. 2012 10:14:00 jenkins.InitReactorRunner$1 onAttained
INFO: Started initialization
20 nov. 2012 10:14:04 jenkins.InitReactorRunner$1 onAttained
INFO: Listed all plugins
20 nov. 2012 10:14:04 jenkins.InitReactorRunner$1 onAttained
INFO: Prepared all plugins
20 nov. 2012 10:14:04 jenkins.InitReactorRunner$1 onAttained
INFO: Started all plugins
20 nov. 2012 10:14:04 jenkins.InitReactorRunner$1 onAttained
INFO: Augmented all extensions
20 nov. 2012 10:14:09 jenkins.InitReactorRunner$1 onAttained
INFO: Loaded all jobs
20 nov. 2012 10:14:09 org.apache.sshd.common.util.SecurityUtils$BouncyCastleRegistration run
INFO: Trying to register BouncyCastle as a JCE provider
20 nov. 2012 10:14:10 org.apache.sshd.common.util.SecurityUtils$BouncyCastleRegistration run
INFO: Registration succeeded
20 nov. 2012 10:14:10 org.jenkinsci.main.modules.sshd.SSHD start
INFO: Started SSHD at port 1574
20 nov. 2012 10:14:10 jenkins.InitReactorRunner$1 onAttained
INFO: Completed initialization
20 nov. 2012 10:14:10 hudson.TcpSlaveAgentListener <init>
INFO: JNLP slave agent listener started on TCP port 1575
20 nov. 2012 10:14:10 hudson.WebAppMain$2 run
INFO: Jenkins is fully up and running
```

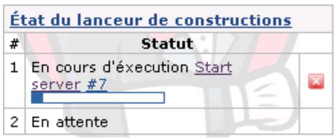
2. Connect to the scheduler admin page using the default URL http://localhost:8036,

3. Launch the job “Start server”,



S	W	Name ↓	Dernier Succès	Dernier Echec	Dernière Durée	
●		Backup database	N/A	N/A	N/A	
●		Import projects	N/A	N/A	N/A	
●		Start server	N/A	N/A	N/A	
●		Stop server	N/A	N/A	N/A	

On the scheduler admin page, once the job “Start server” is launched, you should get:



État du lanceur de constructions

#	Statut	
1	En cours d'éxecution Start server #7	☒
2	En attente	

4. Once the server is started, it can be used to check the Team for Capella Client installations (see the Team for Capella Client Installation verification procedure),

5. Launch the `Backup database` job,

6. Launch the `Import projects` job,

7. Stop the server using the `Stop Server` job → the `Start server` job should be stopped,

8. Stop the Scheduler by closing its command window.

3.5 TEAM FOR CAPELLA SERVER CONFIGURATION

This part describes the basic configuration of the Team for Capella Server and Scheduler.

Chapter #3.3.3.How to Reuse server configuration when configuring v1.3.x will help you save time when installing a new version a Team for Capella server.

3.5.1 DEFAULT SERVER CONFIGURATION

For the complete server configuration documentation, refer to the Capella client *Help Contents* in chapter ***Team for Capella Guide/Server Configuration***

By default, the Team for Capella Server has the following configuration:

- Port: **2036**,
- Repository: **repoCapella**

3.5.2 DEFAULT SCHEDULER CONFIGURATION

For the complete scheduler configuration documentation, refer to the Capella client *Help Contents* in chapters:

- ***Team for Capella Guide / Jenkins Configuration***
- ***Team for Capella Guide / Server Administration / Team for Capella Scheduler***
- ***Team for Capella Guide / Guidelines***

Team for Capella server embeds a Jenkins installation used as a Scheduler, its default port is:

- **8036** (to connect to Jenkins from the host running it, use http://localhost:8036).

In the Scheduler, 4 jobs are built-in and already configured:

- **Start server**: this job starts the server **every Saturday at 06:00.**
  - This job never stops except if “Stop server” is launched,
- **Stop server**: this job stops the server every Saturday at 05:00.
- **Backup database**: this job is launched automatically 3 times a day (07:30, 12:30 and 20:30) from Monday to Friday. Its purpose is to:
  - Do a dump of the database in a file that could be used later to restore the database in exactly the same state (e.g.: existing locks will be restored),

- **Import projects**: this job is launched automatically every hour from 07:00 to 21:00 from Monday to Friday. Its purpose is to:
    - Check periodically the database consistency. If the job is not able to get all projects, it will fail and **stop the server (in this case the database must be reinitialized)**,
    - Keep a copy of projects in file format to be able to reinitialize the database if a corruption happens.

For the 2 last jobs, generated data (database files and projects) are kept as build's artefacts and can be retrieved later using the Jenkins interface (by default, 100 builds are kept for each jobs).

If you have many repositories, you ought to have as many "import projects" jobs that may start at the same time. So you need to configure the number of job executors.

Go to `Manage Jenkins > configure systems` menu if number of T4C repository have been extended: # of executors ≥ =nb of repo +3

3.5.3 How to Reuse previous server configuration when configuring v1.3.x server

When installing a new version of Team for Capella, some parts of the server/client configurations of the previous version can be reused.

3.5.3.1 Procedure

- End task `server.exe` and `lic_server.exe` in task manager (processes sheet) and stop `TeamForCapellaScheduler` (service sheet)
- Keep the entire Team For Capella installation which contains all the file you may reuse.
- Choose among the following configuration you want to reuse
- Once installation is done, in task manager services sheet or in service.msc, restart `TeamForCapellaScheduler` service

3.5.3.2 Server CDO

- Copy part of the `C:\xxx\T4C\server\eclipse\configuration\`**`cdo-server.xml`** file which contains
    - the acceptor tag can be reused
    - the repository configurations can be reused except for *userManager*, *securityManager* tags.
- if the server was configured with the authenticated configuration, the `C:\xxx\T4C\server\eclipse\configuration\`**`users.properties`** file content can be kept as is.
- If the server was configured with the user profile configuration, the user profile model can be reused. Follow the steps described in *Team for Capella Guide/Access Control/User Profiles/Import/Export User Profile Model.*
- You may also consider all properties or parameters you may have set in the server.ini file.
- If you are doing a new installation with the same version, you can reuse the database. Copy the `C:\xxx\T4C\server\eclipse\db` folder. Note that the database cannot be reused if the *Audit mode* activation changes between both installations.

3.5.3.3 Scheduler

- In the `C:\xxx\T4C\scheduler\jenkins_home` folder
  - `config.xml` contains the global jenkins configuration which can be reused
  - `jobs\` contains the jobs definition with the build history. `jobs\<xxxjob>\config.xml` can be reused modulo some parameter changes that may be needed (for example for importer executable call).

3.5.3.4 Importer

- You may consider all system properties or parameters you may have set in the **importer.ini** file. That information have to be transfered to `C:\xxx\T4C\capella\eclipse\`**`importer.bat.`**

3.5.3.5 Client (in remote clients installation)

- You may consider all properties or parameters you may have set in the eclipse.ini file. That information have to be transfered to `C:\xxx\T4C\capella_client\eclipse\`**`eclipse.ini.`**
- If the ip of the server did not change, you can keep the `OBEO_LICENSE_SERVER_CONFIGURATION` property system (just copy `-DOBEO_LICENSE_SERVER_CONFIGURATION=116168566473385529985114841….`)

3.5.3.6 License server

- The license server version is now provided in the Team for Capella archive.
- The OLS file may have been thought up to be reusable for your new Team For Capella version. In this case, you can keep your OLS file.

3.6 BASIC ADMINISTRATION TASKS

3.6.1 User Accounts Management

Several modes of access control can be used for each repository on the server:

- Identification (default mode): each user defined in the file user.properties is authorized to read and/or modify all models present on the repository.
- User Profiles: discriminating user rights are defined in a User Profiles model.
- LDAP Authentication: this mode allows to authenticate with a LDAP server. It can be also used with authenticated or with user profiles.
- Not Authenticated Access: anyone can read and/or modify all models on the repository.

For the complete access control configuration documentation, refer to the Capella client *Help Content* in chapters:

- ***Team for Capella Guide / Access Control***
- ***Team for Capella Guide / Server Configuration***
- ***Team for Capella Guide / Server Configuration / Not Authenticated Configuration***
- ***Team for Capella Guide / Server Configuration / Authenticated Configuration***
- ***Team for Capella Guide / Server Configuration / LDAP Authentication***

- ***Team for Capella Guide / Server Configuration / Activate SSL Connection***
- ***Team for Capella Guide / Server Configuration / User Profile Configuration***

In the current version, Team for Capella is configured with the “Identification” access control mode, ie. passwords are not encrypted. Refer to ***Team for Capella Guide / Access Control*** if you need Authentication or Authorization mechanisms.

There is two ways to manage user accounts (Identification mode) in Team for Capella:

- On Server side:

Edit the **users.properties** file from the **configuration** folder using the following syntax:

```
login=password
```

Example :

```
admin=admin

user1=user1

user2=user2

user3=user3

T123456=T123456

T654321=T654321
```

If the Team for Capella Server is running when this file is changed, it has to be restarted (using the “Stop Server” job and then the “Start Server” job) to load the new accounts.

- On Client side:

Use the “User Management” view available in all Team for Capella clients. When using this view, the server does not need to be restarted after changes in the user accounts.

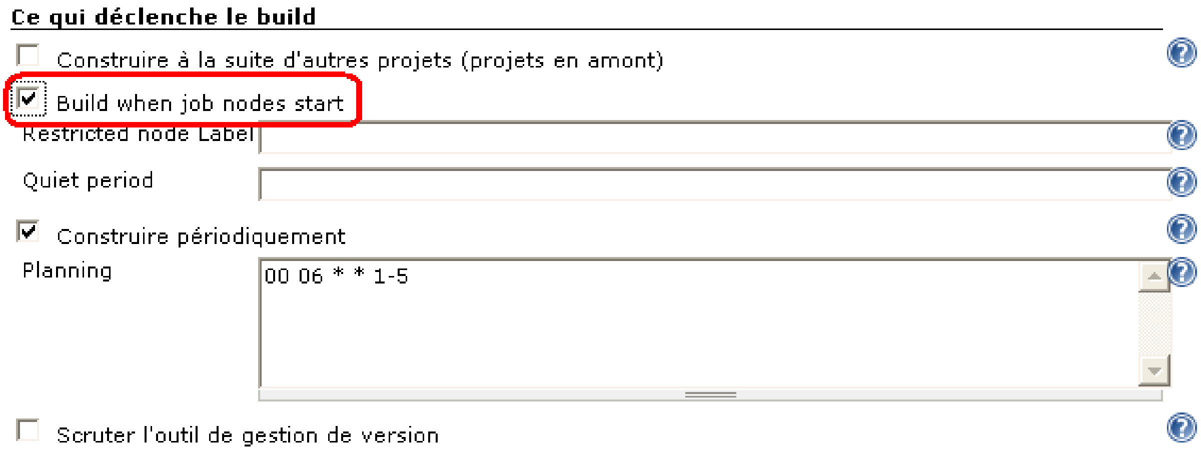
3.6.2 How to start the Scheduler

There are two ways to launch the Scheduler:

- As a classic process:
  - Launch “scheduler.bat” in the Scheduler folder (<TeamForCapellaInstallationFolder>\scheduler\scheduler.bat),
  - This method is not recommended because if the user running the Scheduler logs off or if the computer is restarted, the Scheduler will not be started automatically,
- As a Windows service, see 3.4.2 How to install Scheduler as a Windows Service.

3.6.3 How to start the Server when Scheduler starts

To start the Team for Capella Server automatically when the scheduler starts (i.e.: launch the “Start server” job), go to the configuration page of the “Start server” job and then check the box “Build when job nodes start”:



Ce qui déclenche le build

□ Construire à la suite d'autres projets (projets en amont)

☑ Build when job nodes start

Restricted node Label

Quiet period

☑ Construire périodiquement

Planning

```
00 06 * * 1-5
```

□ Scruter l'outil de gestion de version

3.6.4 How to change job scheduling

Every job contains in its configuration page a text field called “Planning”. Use this field to change the Job’s scheduling configuration.

3.6.5 How to stop the Server

The main methods to close the server are the following:

1) Launch the dedicated Scheduler job: “Stop server” (recommended method),

2) If Jenkins has been stopped, use the OSGI console :

   - `telnet localhost 12036`
   - at the osgi prompt:
     - `close`

To avoid database corruptions, the server **must in no way be closed these ways:**:

1) **Using** the “Abort” button on the “Start server” job of the Scheduler,
2) Especially on Windows 2008 Server 64 bits platforms:
   a. **Closing** the command prompt running the server (if any) by clicking on the Windows close button,
   b. **Leaving** the server close when the user logs out or the computer stops (to avoid this problem, it is advised to launch the Scheduler as a service so the server is not closed on log out).

3.6.6 How to stop the Scheduler

- If the Scheduler was launched as a classic process (scheduler.bat), the opened command dialog can be closed,
- If the Scheduler was launched as a Windows service, open the service configuration dialog (see 3.4.2 How to install Scheduler as a Windows Service) and click on “Stop”,
- An alternative method is to use the command `scheduler.bat stop`.

3.6.7 How to reset the Server

To restart with a clean server or after a database corruption, it can be useful to reset the server:

1. Stop the server using the Scheduler,

2. Remove the folder **db** and **workspace** from the server folder,

3. Start the server,

4. Export the needed models from a Team for Capella Client (using the “Import Job” artifacts for example).

3.7 UNINSTALLATION PROCEDURE

To uninstall Team for Capella:

1. Do a backup of projects stored on the Team for Capella Server,

2. Stop the server (see How to stop the Server),

3. Stop the scheduler (see How to stop the Scheduler),

4. If a Windows Service was created, remove it:

5. Using a Windows Command Line, go to the scheduler directory,

6. Execute the following command: winservice.bat uninstall

7. Remove the installation directory.

4 TEAM FOR CAPELLA CLIENT INSTALLATION

4.1 REQUIREMENTS

4.1.1 RECOMMENDED SYSTEM REQUIREMENTS

For successful installation of Team for Capella Client, your computer must meet the following requirements:

- System requirements: 64 bits, Windows XP/7/8/10, Windows Server 2008/2012/2016
- 2 GHz processor,
- 3 GB RAM (for big models, 4 GB RAM),
- 10 GB of available hard disk space,
- Java Runtime Environment 8u121,
- Virus scanner should not be activated on Capella models files: *.aird, *.melodymodeller, *.airdfragment, *.melodyfragment.

4.2 INSTALLATION PROCEDURE

4.2.1 TEAM FOR CAPELLA CLIENT INSTALLATION PROCEDURE

You may take your Team For Capella client as it has been installed for the server installation. In that case, you simply need to make an archive of the `capella` folder and distribute it to users.

Before archiving the Capella client that will be deployed on others computers, don't forget to remove the workspace used for the installation from the recent workspaces list (`Window > Preferences > General > Startup and Shutdown > Workspaces`).

It is also possible to manually install Team for Capella but note that the installation with the `capella/eclipse/installTeamForCapellaInCapella.bat` is recommended.

Manual Team for Capella installation

1. Extract the archive TeamForCapella-1.32-win32.win32.x86_64.zip and keep only the `updateSite` folder.
2. Download and unzip Capella 1.3.2 bundle (windows 64 bits, https://www.eclipse.org/capella/download.html).
3. Launch Capella (eclipse.exe)
4. Click Menu *Help/Install New Software…* , add the archive file in the `updateSite` folder and select features in the *Team for Capella* category.
5. Then you need to do the additional tasks performed by the installation script:
   - `(optional)` to change the splashscreen: in `capella\eclipse\eclipse.ini` replace the value of `osgi.splashPath` by `platform\:/base/plugins/com.thalesgroup.mde.melody.collab.ui`

6. Add the license server configuration key as described in #3.3.3.2.Client configuration

Steps 4 and 5 are done by the installation script.

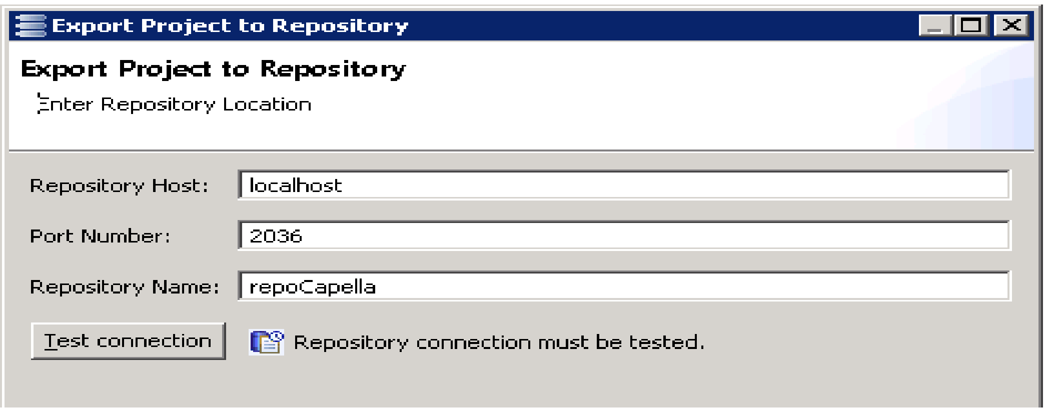
4.2.2 TEAM FOR CAPELLA CLIENT CONFIGURATION

For the complete client configuration documentation, refer to the Capella client *Help Content* in chapter ***Team for Capella Guide/Client Preferences***

1. Launch Team for Capella (“eclipse/eclipse.exe”),
2. Optional: clean the user’s Secure Storage (it contains the save Login/Password, if “Remember Me” option was used):
   a. Go to menu Window > Preferences > General > Security > Secure Storage,
   b. Open the “Contents” tab,
   c. Select “[Default Secure Storage]”,
   d. Click on “Delete”,
   e. Upon request, restart the Team for Capella client.
3. Go to the menu Window > Preferences > Sirius > Team collaboration
   a. Set the server location with the hostname or the IP address of the Team for Capella Server (`localhost` if the server is setup on the same machine),
   b. Click on “Apply”

4.2.3 VERIFICATION INSTALLATION PROCEDURE

1. Create a new project → Right click in the Capella Explorer → New → Capella Project
   a. Call it Test for example → Finish
2. Export the project to the remote repository
   a. Right click on the project → Export → Team for Capella → Export model to remote repository→Next,
   b. Click on Test Connection,



Export Project to Repository

Export Project to Repository

Enter Repository Location

Repository Host: localhost

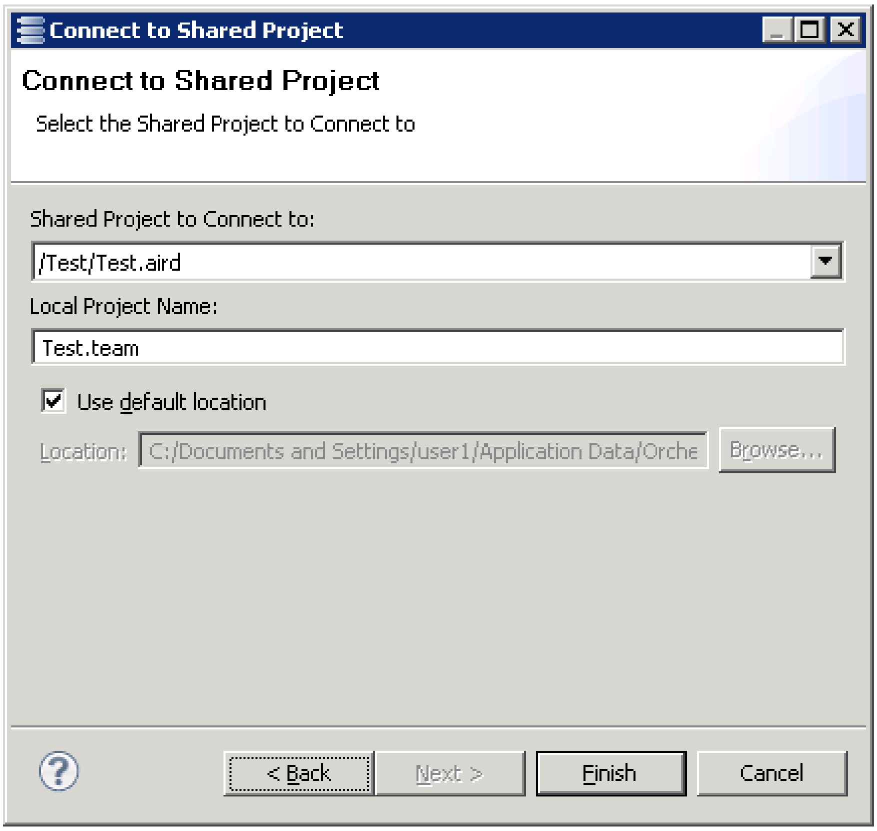
Port Number: 2036

Repository Name: repoCapella

Repository connection must be tested.

   c. Provide the user name and password (by default : user1/user1, user2/user2, user3/user3, admin/admin) → See chapters Default Server Configuration and User Accounts Management to customize users/passwords,
   d. Click on Finish.

3. Connect to the remote project previously exported,
   a. Right click in the Capella Explorer → New → Connect to remote repository,
   b. Test Connection,
   c. Select the Shared project from the list,



Connect to Shared Project

Select the Shared Project to Connect to

Shared Project to Connect to: /Test/Test.aird

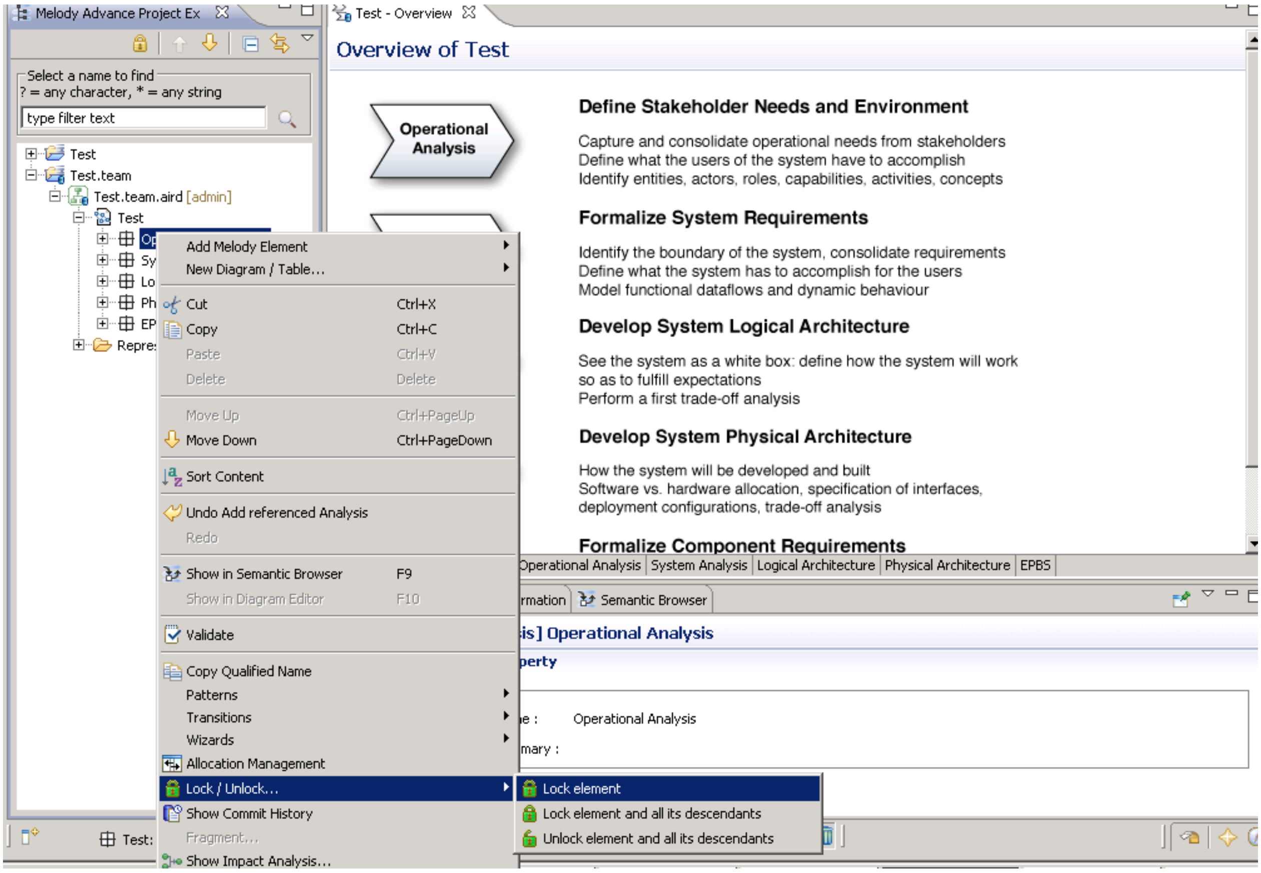
Local Project Name: Test.team

☑ Use default location

Location: C:/Documents and Settings/user1/Application Data/Orche

   d. Click on Finish,

4. You should now be able to work on the project on the remote repository.



Overview of Test

Define Stakeholder Needs and Environment

Capture and consolidate operational needs from stakeholders
Define what the users of the system have to accomplish
Identify entities, actors, roles, capabilities, activities, concepts

Formalize System Requirements

Identify the boundary of the system, consolidate requirements
Define what the system has to accomplish for the users
Model functional dataflows and dynamic behaviour

Develop System Logical Architecture

See the system as a white box: define how the system will work
so as to fulfill expectations
Perform a first trade-off analysis

Develop System Physical Architecture

How the system will be developed and built
Software vs. hardware allocation, specification of interfaces,
deployment configurations, trade-off analysis

Formalize Component Requirements

ie : Operational Analysis

5 RELEASE NOTES

The Team for Capella 1.3.2 release notes are available at:

https://www.obeo.fr/en/team-for-capella-releases-1-3#1.3.2

6 MIGRATION OF EXISTING PROJECTS

6.1 VERSION COMPATIBILITY

Team for Capella	(based on) Capella	(third party software) Sirius
V1.0.1 (32bits & 64bits)	V1.0.1	Sirius V3.1.2
V1.0.3 (32bits & 64bits)	V1.0.3	Sirius V3.1.6
V1.1.0 (32bits & 64bits)	V1.1.0	Sirius V4.1.1
V1.1.1 (32bits & 64bits)	V1.1.1	Sirius V4.1.4
V1.1.2 (32bits & 64bits)	V1.1.2	Sirius V4.1.6
V1.1.3 (32bits & 64bits)	V1.1.3	Sirius V4.1.7
V1.2.0 (64bits)	V1.2.0	Sirius V5.1.0
V1.1.4 (32bits & 64bits)	V1.1.4	Sirius V4.1.9
V1.2.1 (64bits)	V1.2.1	Sirius V5.1.2
V1.3.0 (64bits)	V1.3.0	Sirius V6.1.1
V1.2.2 (64bits)	V1.2.2	Sirius V5.1.4
V1.3.1 (64bits)	V1.3.1	Sirius V6.1.3
V1.3.2 (64bits)	V1.3.2	Sirius V6.1.4

Table 1 : Version Compatibility (sorted by delivery dates)

6.2 MODEL MIGRATION FROM PREVIOUS VERSION TO V1.3.X

To use a previous version model in Team for Capella V1.3.x, a migration must be done to be compliant with the new metamodel. This model migration is provided by Capella and must be done between each minor version (from V1.2.x to V1.3.x for example).

The process to migrate a model to V1.3.x from a shared repository follows the following steps:

1) If Team for Capella users have created local diagrams (i.e.: diagrams stored in the local .aird file), they have to move all diagrams they want to keep to a remote .aird or .airdfragment ("**Move Diagrams**" action),

2) Import locally the model to migrate ("**Import model from remote repository**") using previous version (V1.2.x) of Team for Capella. Make sure that associated team server is running before importing from remote repository. Another way is to use the last valid import of the Scheduler.

   ➔ Make a baseline of the imported model,

3) Migrate the previously imported model ("**Migrate Project toward current version**") in a Team for Capella V1.3.x. (The migration can also be done in Capella V1.3.x). To do so, please refer to *Migration* dedicated chapter of Capella online installation guide,

   ➔ Make a baseline of the migrated model,

4) Export the migrated model to the Team for Capella server V1.3.x (“**Export model to remote repository**”) using Team for Capella V1.3.x. Make sure that associated team server is running before exporting to remote repository.

After performing these steps, the model in the shared repository is in right version. Team for Capella has to be upgraded on client’s computers. Then users can connect and work on the model. They do not need to do any migration.

6.3 MODEL MIGRATION FROM AN OLDER VERSION

The model migration is necessary between each version considering only minor and major version change.

- For the first model migration, you need to reproduce only steps 0, 1 and 2 described above.
- For the following intermediary model migrations, you need to reproduce only step 2.
- For the last model migration, you need to reproduce only steps 2 and 3

For example is you start from v0.8.x model

- do steps 0,1 and 2 with previous version = V0.8.x and new version= V1.0.3
- do step 2 with previous version = V1.0.3 and new version= V1.1.4
- do step 2 with previous version = V1.1.4 and new version= V1.2.2
- do steps 2 and 3 with new version= V1.3.2