Imagine That Inc.

ExtendSim Cloud User Guide

Configuration, Usage, & Maintenance

May 19, 2022

Table of Contents

Overview

ExtendSim Cloud documentation

There are three PDF documents for the ExtendSim Cloud license:

1. This User Guide
2. ExtendSim Cloud Technical Reference
3. Cloud Installation/Activation

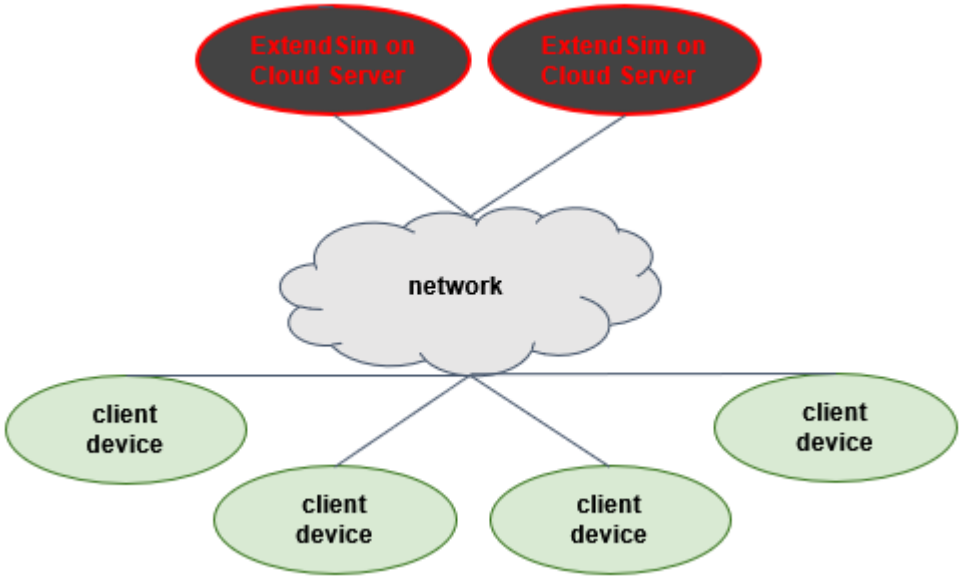
These documents get installed into the Documents/ExtendSim10Cloud/Documentation folder on the Cloud Server and are available for download from our website at www.ExtendSim.com.

ExtendSim Cloud product

An ExtendSim Cloud license gives the Licensee the right to:

- Host the ExtendSim application and model files on one or more servers, and
- Allow authorized internal or external "end user clients" to obtain access to those files to perform simulations over a LAN or over the internet or other WAN

A Licensee's clients can thus configure, run, and analyze ExtendSim models remotely, avoiding the need to install ExtendSim on local computers. Clients can be the Licensee's external customers or its employees and contractors.



The ExtendSim Cloud product uses a client-server architecture so that ExtendSim models located on a server can be executed from remote client devices. To provide the service, the Licensee installs the ExtendSim Cloud application, model file(s), ancillary files such as libraries and includes, and a continuously running Cloud Services Manager application on one or more "Cloud" servers. Client end users then use a frontend interface (html file, browser, or custom app) on their local computers or smart devices to request the Cloud services – simulation runs for the ExtendSim models they are authorized to access.

These requests push information required to run the ExtendSim models from the client machine to the Cloud Server and trigger simulation runs on the server. When simulation runs are completed,

requests are made at the frontend interface on the client to pull simulation results from the Cloud Server to the client device.

Advantages of the ExtendSim Cloud License

The main advantages of using an ExtendSim Cloud license are that the Licensee can:

1. Maintain centralized control over the ExtendSim application and model files, including updates and configurations
2. Provide end users with the ability to configure and run ExtendSim simulations without having to learn or even use the software
3. Reduce the complexity and costs of deployment onto individual computers
4. Present end users with a platform for performing sophisticated simulation analysis remotely
5. Allow ExtendSim simulations to be run from any computer or smart device

ExtendSim Cloud Architecture

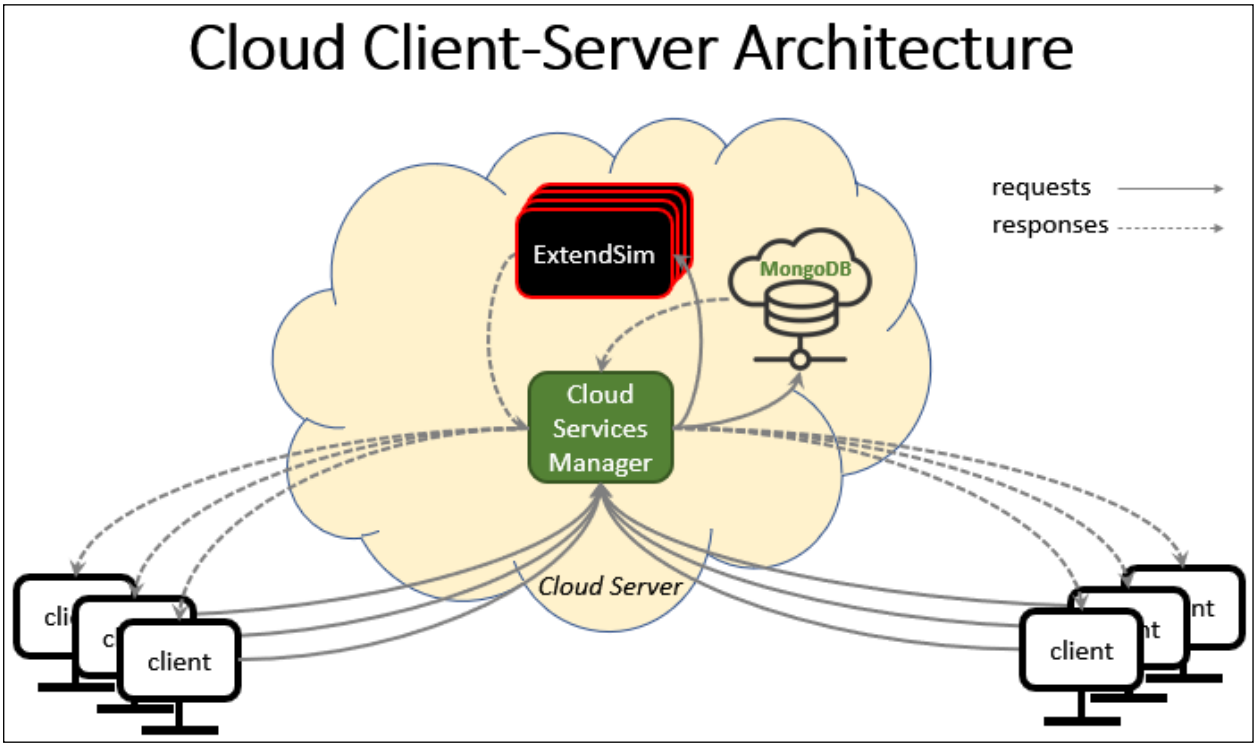


Figure 1: Network topology

The ExtendSim architecture has been implemented in a framework that consists of a *backend* in which responses are made to requests for Cloud services and a *frontend* from which these requests are made.

Backend

The backend of the ExtendSim Cloud framework consists of the:

1. Cloud Services Manager
2. ExtendSim Cloud application
3. ExtendSim reference models

4. MongoDB data repository

Cloud Services Manager

The Cloud Services Manager (CSM) gets installed once on each Cloud Server. As determined by settings in a configuration file, it is an HTTP or HTTPS server process that, once launched, runs in the backend of the Cloud framework. The CSM is effectively middleware that functions as the control point through which all interactions between a client device and the ExtendSim Cloud application are routed. Its purpose is twofold:

1. To respond to client requests made using Cloud Server API functions.
2. To orchestrate the launching, running, and utilization of one or more instances of ExtendSim running on the Cloud Server.

The CSM consists of a collection of JavaScript files that are executed as an application in node.js. It uses the JavaScript run-time environment to implement its functionality and provides RESTful API routes (endpoints) using the HTTP or HTTPS protocol as set in a configuration file.

A setting in a configuration file (ExtendSimCloudConfigurationManager) specifies how many cores the Cloud Server has. The CSM then spawns up to that many concurrent instances of ExtendSim, restricted by the maximum allowed by the License Manager. When all ExtendSim instances are busy running simulations, the CSM manages the queuing of requests for simulation runs.

Client requests

When the CSM runs it is listening to ports, mapping routes, and responding to client requests such as launching instances of ExtendSim and returning results. Client requests are routed by API functions (see the ExtendSim Cloud Technical Reference) that perform the following actions on the Cloud Server:

1. Login and authenticate users
2. Sign-up new users
3. Launch instances of the ExtendSim application
4. Create scenario folders to store models that have been run
5. Open models
6. Set up model inputs
7. Run simulations
8. Return simulation outputs
9. Store simulation inputs and outputs in the data repository
10. Retrieve information from the data repository

ExtendSim Cloud application

The ExtendSim Cloud product uses a specialized ExtendSim application configured to behave as an HTTP/HTTPS server, communicating over a port and servicing data routes.

The ExtendSim Cloud application is installed once on each Cloud Server. It is a run-time product and is not intended for model or block development. It is also not intended for any form of direct user interaction; all interactions with ExtendSim Cloud are managed and controlled by the Cloud Services Manager (CSM), using inputs and function requests from client-side interfaces.

In response to client requests, ExtendSim Cloud generates multiple instances of itself, running in memory on the Cloud Server. From the perspective of the Cloud Services Manager (CSM), each ExtendSim instance running on the Cloud Server is a "server".

Each instance of ExtendSim is assigned a specific simulation run with the inputs and outputs managed by the CSM. The CSM orchestrates the utilization of individual ExtendSim instances to maximize the throughput of client requests for simulation runs. The number of running instances is limited by the lesser of the ordering document's maximum number of concurrent instances and the number of cores on the Cloud Server.

Since no user interactions are required, the CSM can launch ExtendSim and run simulations in a headless mode. For debug purposes, it might be desirable to observe what is occurring in an ExtendSim model during a simulation run. This can be done by setting the EXTENDSIM_HEADLESS parameter in the configuration file to "No", forcing the Cloud Services Manager to run ExtendSim in a non-headless mode.

ExtendSim reference models

Reference models are the read-only models that are used to create a scenario model each time a user creates a new scenario. Reference models are stored in *User Group* folders, either within the default *ExtendSim Reference Models* folder in the UserName/Documents/ExtendSim10Cloud folder or elsewhere on the Cloud Server. For more information about User Groups, see page 10.

The ExtendSim reference models are accessible to users through API function requests made from client-side interfaces to the Cloud Services Manager. Access to these models is controlled using the *userGroups* collection in the *ExtendSimCloud_usersdb* database in the MongoDB repository on the Cloud Server.

Each user group in this collection is associated with a particular user group folder on the Cloud Server, and each folder contains models that can be accessed by that user group. When a valid user is created, that user is associated with a particular user group on the Cloud Server. The information for users (authorization code, username, password, and user group) is stored in the *users* collection in the MongoDB *ExtendSimCloud_usersdb* database.

Note 1: do not place libraries or extensions in the User Group folders. Libraries must be within the Libraries folder; extensions must go into the appropriate Extensions folder.

Note 2: by default, reference models aren't run as simulations. Instead, they are used as the starting point for creating Scenario Models. After a reference model has been configured with a specific collection of input values, it is saved as a scenario model in a separate scenario folder in the ExtendSim Scenario Folders folder. See the section on Scenario Folders on page 13 for more information.

Note 3: if you want to run a scenario, save the model, and then run another scenario using the model that was saved from the previous run, set the runusingcopy parameter of the submitsimulationscenario API function to True. This will allow the scenario model to be run with different scenarios and saved at the end of each run, rather than having each scenario model placed in a new folder.

MongoDB Data Repository

The MongoDB data repository is a document-oriented NoSQL database that uses MongoDB for high volume storage of user information and simulation scenario data. During the installation process, after configuring users and user groups, you ran the *MongoDB-user-groups setup.bat* file. Running that batch file automated the creation of user group information in the MongoDB database.

Through the Database port (by default, 27017), the Cloud Services Manager controls the flow of data into and out of the MongoDB repository using the database named *ExtendSimCloud_usersdb*.

This database organizes its data using documents and collections of documents. Documents consist of key-value pairs which are the basic unit of data in the repository. The *ExtendSimCloud_usersdb* database uses the following collections:

- authorizationCodes
- userGroups
- submittedScenarios
- userScenarios
- users

The *ExtendSimCloud_usersdb* database is embedded within the files in the folder Program Files/MongoDB/Server/Data. To access the database, use a MongoDB frontend utility program such as Robo 3T or MongoDB Compass.

Frontend

Client interfaces

There are two ways to provide a frontend client interface for ExtendSim Cloud:

1. Using the Cloud Scenario Manager.html file
2. Programming a custom web page in a browser or a standalone app

Cloud Scenario Manager

The ExtendSim Cloud product includes the *Cloud Scenario Manager.html* file, located by default at Program Files/ExtendSim10Cloud/Client Files. This frontend interface is implemented in HTML, CSS, and JavaScript and uses a web browser to communicate with the Cloud Services Manager running in the backend.

The Cloud Scenario Manager requires that end users obtain an authentication code and register to obtain access to the models. See page 12 for the process to create and use the authentication code.

The Cloud Scenario Manager can also be used to explore using JavaScript to execute all the Cloud Services Manager API functions including opening, saving, and closing ExtendSim Cloud models, sending data to and receiving data from ExtendSim models, running simulation models, and getting status information for simulation runs. Additionally, the Cloud Scenario Manager contains examples demonstrating how to use the *signup* and *login* API functions, as well as other API functions to retrieve information from the MongoDB repository.

You can use the Cloud Scenario Manager as is or customize it for your use with the Cloud Services Manager API functions. See the *Cloud Technical Reference* for more information and the API.

Custom client interface

Custom client interfaces for the Cloud framework can be created using any programming language (e.g., JavaScript, PHP, VBscript, C#, C++, or Python) that is capable of properly using the Cloud Services Manager API functions to communicate with a RESTful service.

Authentication to allow end user access to models is performed using the *signup* API function, as discussed on page 13.

Also see the *Cloud Technical Reference* for more information and the API.

Accessing the Cloud Server from the client device

Client-side access to the ExtendSim Cloud backend is predicated on having performed a valid user login. As discussed on page 12, how that access is controlled depends on the type of frontend interface.

User groups provide the Cloud framework with a mechanism for enabling authorized access to ExtendSim reference models on the server. Each end user can be associated with one or more user groups. User groups are created and stored in the *UserGroups* collection in the MongoDB repository on the Cloud Server.

See page 10 for a description of user groups and folders. For how to set up users and user groups, see the *Cloud InstallationActivation* document.

Client-based data transfer methods

There are two ways to transfer data from the client device to ExtendSim models on the Cloud Server:

1. Directly using Cloud Services Manager API functions. These API functions allow users to send data from the client:
   - Directly to targets within an ExtendSim model such as a database table or a dialog variable
   - As files directly to Scenario folders on the Cloud Server. This method requires the models being run on the server to explicitly import external data from the files being sent to the server using methods such as the *Data Import Export* block to import delimited text files. **NOTE**: While possible to do, it is strongly recommended to not use Excel workbooks to import and export model data on the server. The Excel application does not function robustly when multiple instances of the same workbook are required to be opened concurrently.
2. Indirectly as updates from client-side interfaces to database server applications, e.g., SQL Server, Oracle, MySQL, that are directly connected to ExtendSim reference models on the Cloud Server. The direct connections between ExtendSim models and these databases are made using ModL *import* and *export* functions that get triggered from within model blocks such as the *Data import Export* block during the opening or running of a model.

The client session

As discussed more in the ExtendSim Cloud Technical Reference, a client session is defined as the collection of user interactions that occur between the client application and the Cloud Services Manager from when the user logs in to the Cloud server until the point at which the user terminates the session. While not necessarily in the sequence that the user will encounter, these interactions fall into the following *categories*:

- User login. User access is restricted by User Group.
- Selecting a reference model and defining scenarios. User authentication can be used to control access to reference models. The client uploads input data from their device to the selected model.
- Saving scenario definitions.
- Selecting scenarios to run.
- Submitting simulation scenario runs. The client submits a request to run the configured simulation scenario. Each successfully submitted scenario is assigned an ID.

- Scenario status monitoring and results retrieval. Querying the scenario ID determines if the simulation has completed. The client app requests and receives output data from completed simulation scenarios.
- Session termination. The model is closed.

Starting and Stopping the Cloud Services Manager

The Cloud Services Manager (CSM) is the control point through which all interactions between Cloud components are routed. For ExtendSim Cloud to launch multiple instances, and for data to be transferred between the CSM and MongoDB, the CSM must be running. It can run continuously as a service or be launched only as needed.

Running the CSM as a service

The Cloud Services Manager process can be set up to run as a Windows Service, such that it will run continuously and will automatically restart if the Cloud Server is rebooted. To start the service, run the *install-windows-service.bat* file, located at Program Files/ExtendSimCloud/Server Files, with administrative permission. This starts a service named *ExtendSimCloud Services Manager.*

See the Cloud Installation & Activation document for instructions.

Stopping the CSM service

The easiest way to stop the *ExtendSimCloud Services Manager* service is to run the *uninstall-windows-service.bat* file, located at Program Files/ExtendSimCloud/Server Files, with administrative permission. This stops and removes the service.

You could also stop and then remove the *ExtendSimCloud Services Manager* service manually, using the process described on page 18.

Launching the CSM manually

To manually launch the CSM, use the start.bat file, located at Program Files/ExtendSim Cloud/Server Files. This opens the command prompt and runs the CSM, showing the log in the same window. When the command prompt is closed the CSM is stopped along with all the ExtendSim Cloud instances the CSM opened.

Configuring the Cloud Services Manager

The configuration file allows you to specify settings used by the Cloud Services Manager, such as how many cores the server has, the paths to specific directories, and whether you want ExtendSim to launch headless or not. A configuration file, *.env*, is located on the Cloud Server in the folder *Program Files/ExtendSim10Cloud/Server Files*.

As shown below, the Cloud product also contains a user-friendly implementation of the .env file, named *ExtendSimCloudConfigurationManager.exe*. When you install the ExtendSim Cloud files on a Cloud Server, the installer can place a shortcut (ES Cloud Config) to the configuration application on the Cloud Server desktop. On startup/launch, the .env file is consumed by the node.js application to configure the Cloud Services Manager with the parameter values you enter.

Each of the settings in the configuration file specifies some aspect of your Cloud situation. The explanations and options for these settings is given in the *Cloud InstallationActivation* document.

Parameter Name	Parameter Value
EXTENDSIM_APPLICATION_PATH	C:/Program Files/ExtendSim10Cloud/
EXTENDSIM_AVAILABILITY_CHECK_DELAY_AFTER_START	2000
EXTENDSIM_AVAILABILITY_CHECK_DELAY_WHEN_BUSY	1000
EXTENDSIM_HEADLESS	no
EXTENDSIM_INSTANCES	2
EXTENDSIM_INSTANCE_ADDRESS	localhost

NOTE: To make a change to the configurations after you have saved the settings, stop the ExtendSim Cloud Services Manager service if it is running. Enter the new settings in either the .env file or in the ExtendSimCloudConfigurationManager app. Then restart the service or run the start.bat file as Administrator. Both methods reload the configuration parameters.

User Groups and Folders

The ExtendSim Cloud product requires that *User Groups* and *User Group* folders be set up. They provide the mechanism ExtendSim Cloud uses to control which end-user clients have access to which models on the Cloud Server. Each user group is associated with a particular folder on the server and each authenticated user will be associated with a particular user group. Thus, each user will be granted access to the models contained in the folder associated with their user group.

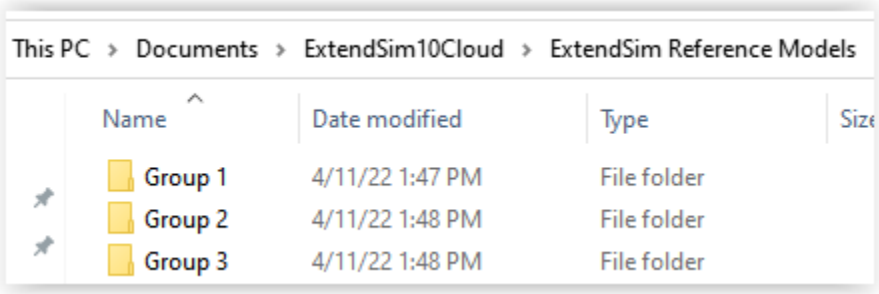
User Group folders

The *Cloud InstallationActivation* document describes how to create User Group folders. Each User Group folder stores the reference model files that a specific group of users will be given access to.

To provide a default location for User Group folders, the Cloud installer created an *ExtendSim Reference Models* folder in the Documents folder. However, you can create User Group folders anywhere on the Cloud Server you want, as long as the user has Read/Write permissions to those folders and the paths to the folders are specified when you set up the User Groups.

A user associated with a particular user group will have access to all the models residing inside that group's folder.

Note: User Group folders are only for models and ancillary files such as spreadsheets. Do not place libraries or extensions in the User Group folders. Libraries must be within the Libraries folder; extensions must go into the appropriate Extensions folder.



Creating User Groups

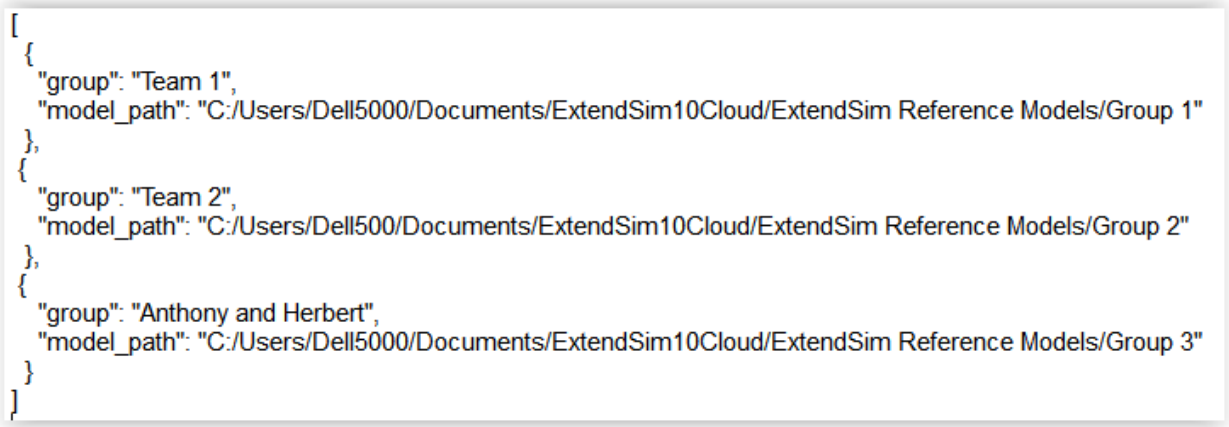
After the User Group folders have been created, User Groups are configured in the *import-user-groups.json* file that is located in the folder *Program Files/ExtendSim10Cloud/Server Files/MongoDB*.

Launch Notepad or something similar with administrator privileges, and use its File menu to Open the *import-user-groups.json* file.

Edit the file as follows:

1. The "group" name can be any alphanumeric or numeric name you want as long as it not the same as any other group name.
2. "model_path" specifies the path to one of the User Group folders that you have already created.
    a. Verify that the user group folder names are correct and each path ends with a ".
    b. If the model path has backslashes \ instead of forward slashes/, you must use a double backslash\\.
3. To specify each new User Group after the first one, put a comma after the right brace (},). Then enter a left brace to specify a new *group name* and *model path*.
4. Repeat as needed, as shown below. However, do not put a comma after the final right brace.
5. Save and close the file.

The following screenshot shows how to set up 3 user groups to use 3 of the User Group folders:



After the group/model_path pairs have been created, they must be entered into the MongoDB database, as discussed below.

Using the batch file

The *import-user-groups.bat* file can be used to automate the creation of user groups in MongoDB for your Cloud Server. By default, it is configured to use the *import-user-groups.json* file to setup user groups in MongoDB.

With elevated privileges, launch the *import-user-groups.bat* file at Program Files/ExtendSim10Cloud/Server Files/mongodb. The number of documents that are imported should equal the number of User/User Group pairs you've created.

The *ExtendSimCloud_usersdb* database stores user information; it is embedded within the files in the folder Program Files/MongoDB/Server/4.2/Data. To access the database, use a MongoDB GUI such as Robo 3T.

User Authentication

To control who has access to the models on the Cloud Server, users must be authenticatable. This is accomplished differently depending on which frontend interface end users will use: the Cloud Scenario Manager or a custom website or application built with the ExtendSim Cloud API.

Note: Since setting up user authentication requires the transfer of data between the Cloud Services Manager and MongoDB, the Cloud Services Manager must be running, as discussed on page 9.

Cloud Scenario Manager user authentication

The Cloud Scenario Manager is discussed on page 7; it requires that the user log in. To enable that, you must create an authorization code that is linked to the user group folder(s) the user will have access to. The end user then uses the authorization code to register, with a username and password, to get access to the Cloud Scenario Manager.

Create the authorization code

1. With administrator privileges, launch Notepad or something similar and use its File menu to open the *import-authorization-codes.json* file. The file is located in the Program Files/ExtendSim10Cloud/Server Files/MongoDB folder; it has skeleton information you can customize.
2. Edit the file as follows:
    a. The "code" can be any alphanumeric or numeric name as long as it is not the same as any other code.
    b. For "usergroups", enter the name of one or more valid existing User Groups you created in the "import-user-groups.json" file. (You can also find a group using the route getServerUserGroups, which returns all existing user group names.)
    c. Put each group name inside quotation marks; separate multiple names with a comma.
    d. Note: while one authorization code allows an end user to have access to multiple user group folders, there is a one-to-one relation between the user's credentials and the user group name. Upon sign in, users must choose one of the groups; to access a different group they must register a different set of credentials.

```
[
  {
    "code": "A55667",
    "usergroups": [
      "Team 2",
      "Anthony and Herbert"
    ]
  },
  {
    "code": "A112248",
    "usergroups": [
      "Team 1"
    ]
  }
]
```

3. Separate multiple credentials by placing a comma after the right brace (}), as shown.
4. Repeat as needed. However, do not put a comma after the final right brace.
5. Save the file using the File menu, then close the file.

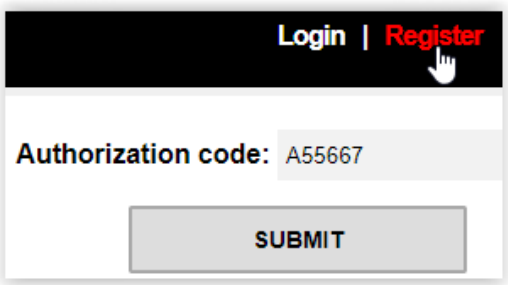
Run the batch file to store the information in MongoDB

1. With administrator privileges, launch the *import-authorization-codes.bat* file at Program Files/ExtendSim10Cloud/Server Files/MongoDB

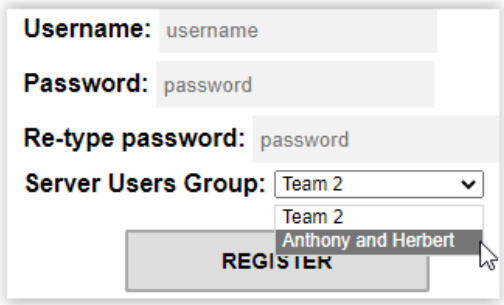
```
Importing import-authorization-codes.json

2022-04-28T16:13:38.144-0700    connected to: mongodb://localhost/
2022-04-28T16:13:38.155-0700    2 document(s) imported successfully. 0 document(s) failed to import.

Press any key to continue . . .
```

2. If successful, it should report a number of documents imported successfully, where all of the codes are imported and each document represents one of the codes you created.
3. The information is stored in MongoDB's *ExtendSimCloud_usersdb* database.

Registering in the Cloud Scenario Manager



1. Give the end user the appropriate authorization code and the Cloud Scenario Manager.html file, located at Program Files/ExtendSim10Cloud/Client Files.
2. In the Cloud Scenario Manager.html file, the end user clicks the *Register* button.
3. In the window that appears, the end user enters the authorization code and clicks Submit.
4. In the next window, they enter the information as follows:
    a. *Username* can be any numeric or alphanumeric name they want as long as it has not already been signed up.
    b. *Password* can be any numeric or alphanumeric name they want. It will be stored in encrypted form on the Cloud Server.
    c. Since there is a one-to-one relation between the username and the user group, *Server Users Group* is whichever group they want these credentials to apply to.



Custom frontend application user authentication

For a custom frontend application or web page, authentication is created in the Cloud framework using the *signup* API function. This function adds a new instance to the *users* collection in the MongoDB repository. Each *user* instance consists of a unique username, password, and the associated user group, as specified in the function's 3 arguments:

1. Username • Any numeric or alphanumeric name that has not already been signed up.
2. Password • Anything you want. It will be stored in encrypted form on the Cloud Server.
3. User Group • The group must be a valid existing User Group. To locate a group, use the route getServerUserGroups, which returns all existing user groups. User Group specifies the path to the folder to which the user will have access; permission applies to all models within the folder. Note that there is a one-to-one relation between the Username and the User Group.

Once a valid user has been created using the *signup* API, that user can be given access to the Cloud Server using the *login* API function.

Postman is an excellent tool for testing Cloud Server API calls and can readily be used to call the signup API to create new users. In addition, the Cloud Scenario Manager.html template demonstrates how to use the *signup* and *login* API functions. Those functions are fully described in the ExtendSim Cloud Technical Reference.

To provide security *during* a user session see Jason Web Tokens (JWT) on page 14.

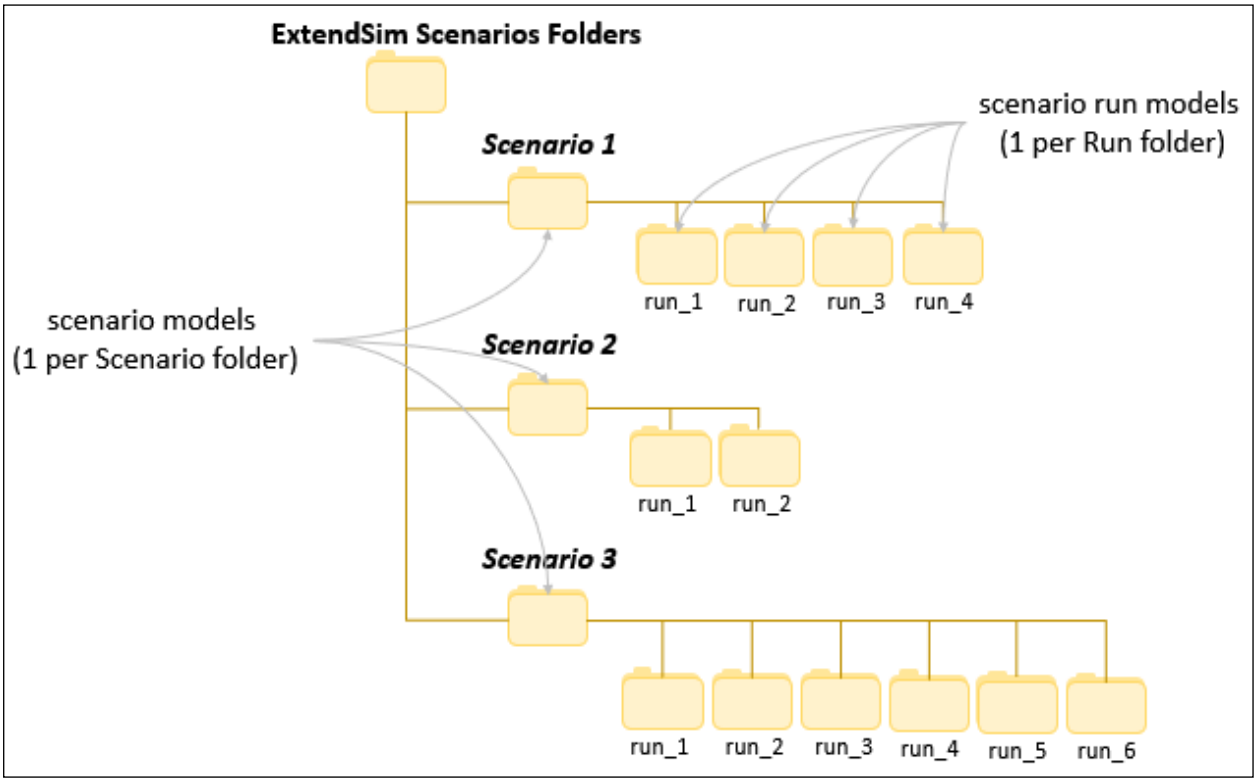
Scenario Models and Folders

As described on page 6, reference models are the read-only models used as the starting points for scenario models. A scenario model is a reference model that has been configured and then saved with a specific set of input values, ready to be run.

Where scenario models are stored

When ExtendSim Cloud was installed, an empty folder named ExtendSim Scenario Folders was created at Documents/ExtendSim10Cloud; its location is referenced in the configuration file using the EXTENDSIM_SCENARIOS_DIRECTORY_PATH parameter. By changing the setting of that parameter, the ExtendSim Scenario Folders directory can be relocated to any location on the Cloud Server.

As shown in the diagram, each scenario model is saved in a separate scenario folder in the Scenario Folders folder, such that the folder will have one scenario folder for each scenario created by users.



Simulation runs and scenario run models

Simulation runs never directly use the scenario models at the root level of a scenario folder. Instead, to enable multiple concurrent instances of a scenario to be run on the Cloud Server, the Cloud Services Manager creates a new subfolder for each run for a scenario and places a copy of the base scenario model into that subfolder. These copies of scenario models are referred to as *scenario run models*. Each "Run" subfolder represents one and only one simulation run. Thus, if a given scenario was run 25 times, there would be 25 subfolders in the root folder for that scenario.

Using JSON Web Tokens (JWT) for Additional User Security

ExtendSim Cloud has a built-in user authentication mechanism that is activated by setting the SERVER_AUTH_JWT parameter in the configuration file to "yes". JWT (JSON Web Tokens) is a standard client-server mechanism that is used to provide security during a user session. It does this by ensuring that, following the authentication of a user via a valid login, all queries to the Cloud Server are being made by the user that logged in at the beginning of the session.

This authentication mechanism uses the ***signup*** and ***login*** API functions to create and authenticate users for your ExtendSim Cloud license.

- Use the **signup** API function to create new users for your ExtendSim Cloud license. This function requires the specification of a new username, a password, and a valid user group. The signup API stores the user information in the MongoDB database that is installed with your ExtendSim Cloud license. Passwords are stored in an encrypted form.
- Once a valid user has been created using the signup function, users can log in to the Cloud Server using the **login** API function. The login function uses the MongoDB database to authenticate the username specified in the API call. The successful execution of the login API returns a dynamically created web token.

If the SERVER_AUTH_JWT parameter in the configuration file is set to "yes", the Cloud Services Manager responds to a valid login API call from the client with a dynamically generated web token. The client application MUST use this web token in the header of every API call that it makes to the Cloud Services Manager. If the Cloud Services Manager does not find a valid web token in the header of an API call, it rejects the request from the client application.

There is also a SERVER_AUTH_JWT_EXPIRATION parameter in the configuration file. If SERVER_AUTH_JWT is set to "yes", the SERVER_AUTH_JWT_EXPIRATION specifies how long a given client-server session is valid. The default setting is 10h (hours). After the period expires, the token created at the beginning of the session is no longer valid and the user must login again.

Debugging

ExtendSim Cloud generates log files and messages that are useful for debugging:

1. The Cloud Services Manager records transactions between the ExtendSim application and client devices executed on a Cloud Server.
2. ExtendSim generates application-level and user function messages.
3. The License Manager, which is installed on the License Server, can generate a RLMDIAG.txt file for diagnosing server or network issues.

Transaction log from the Cloud Services Manager

The Cloud Services Manager (CSM) is installed on each Cloud Server; that is, on each server that has ExtendSim installed. Each CSM generates transaction logs of all the interactions between client devices and the ExtendSim application on that server.

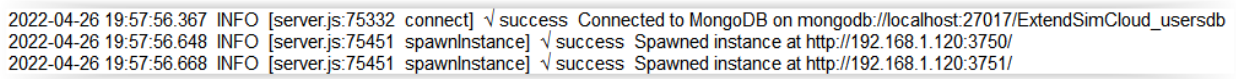
The generated files (YYYMMDD-HHmm-SEQ-Server.log) are located in the Program Files/ExtendSim Cloud/Server Files/Log folder on each Cloud Server. A new log file will be generated each time one of the following occurs:

1. ExtendSim Cloud is started or restarted
2. The date changes such that the day is different from the current or last log file day
3. The current log file reaches 20 MB in size

The log file reports the version of the entire collection of JavaScript files used by the Cloud Services Manager application, the API functions called during ExtendSim launch and execution, and any

errors that have been found. It also reports the raw .env file content and the parsed environment so you can compare what has been configured to what the CSM is actually using.

Each line in the log report uses a three-part output format: {date/time} - function name: information, as shown below.



In the report:

1. "ExtendSim Server" is the specialized version of ExtendSim that behaves as a server application, communicating over a port and servicing routes
2. The controller files contain the code that executes all of the routes provided by the Cloud Services Manager. ESSrv and ESCfcn are prefixes for different classes of functions used in the JavaScript files.
    - ESSrv is for functions that communicate directly with the ExtendSim Server
    - ESCfcn is for ancillary ExtendSim controller functions

This transaction log is the primary method for determining why something went wrong when running ExtendSim models. If you can't figure out the problem from reviewing the log file, initiate a support ticket by going to https://extendsim.com/support/support-help. **Submit the log file along with your ticket, and make sure that the log is associated with the date on which the error occurred**.

ExtendSim messages

ExtendSim can be installed on one or more Cloud Servers; it will launch multiple instances of itself on each server depending on requests for model runs and the number of cores available. Two features help when debugging ExtendSim model runs:

1. Running in headless mode to generate a log file
2. Running in non-headless mode so you can see modeling issues

When running ExtendSim in *headless mode*, any messages generated by the application that would normally be directed to the user interface are instead directed to a log file. That includes user API functions such as UserError, UserParameter, UserPrompt, as well as application-level messages. If the log file (HeadlessLogFile.txt, placed by default in the Program Files/ExtendSim10Cloud/Server Files folder) is not present, it will be created and placed in the ExtendSim10Cloud folder that is in the Documents folder. If the log file has already been created, new information will be appended.

Switching to the *non-headless* running mode is useful if the model run encounters an unexpected state that causes the run to abort. For the EXTENDSIM_HEADLESS parameter in the CloudConfigurationManager file (located by default in the Program Files/ExtendSim10Cloud/Server Files folder), changing the setting to *No* forces spawned instances of ExtendSim to be launched in non-headless mode.

When starting the Cloud Server for the first time, it is recommended that you run ExtendSim non-headless. This will allow you to respond to any firewall messages that might appear the first time ExtendSim Cloud is launched. After that, set EXTENDSIM_HEADLESS to "Yes" *unless you want to run in non-headless mode to debug problems.*

License Manager and License Server issues

The License Manager is installed on the License Server. It runs as a service and monitors requests transmitted through TCP/IP ports to manage the Cloud license and control the number of concurrently running instances of ExtendSim.

If the license stops working or presents server error messages, generate a debug log on the License Server:

1. Open the Reprise License Server Administration (RLSA) window in a browser. (See the Cloud Installation & Activation for instructions.)
2. On the left side of the RLSA window, click the Diagnostics command
3. In the main window, click the Run Diagnostics button and wait for the window to report that the diagnostics were written.
4. Locate and open the text file named RLMDIAG.txt; it will be placed where the License Manager is installed (by default, C:/ExtendSimFloatingLicenseServer).
5. This report contains information about License Manager activities over time. Examine the report to see if there's any discrepancies or anomalies that might indicate the cause of your issues. If you can't readily find the issue, we are happy to help. Create a support ticket and send the debugging log file to us with a description of your issues.

If you can't figure out the problem from reviewing the debug log file, initiate a support ticket by going to https://extendsim.com/support/support-help. **Be sure to submit the diagnostic file along with your ticket**.

Installing or Uninstalling the ExtendSim Cloud Product

General Information

The ExtendSim Cloud product has two installers:

1. An installer for the License Manager, which is discussed on page 19
2. A separate installer for the Cloud application and files, as discussed below.

The License Manager and the Cloud application and files can be installed on separate devices (on a "License Server" and on a "Cloud Server", respectively) or on the same device.

Installing the Cloud Application and Files

The ExtendSim Cloud Installation/Activation document describes how to install the files and configure the settings for the Cloud product.

You may install ExtendSim Cloud on multiple Cloud Servers but only install ExtendSim Cloud one time on each Cloud Server.

Removing ExtendSim Cloud Files from a Cloud Server

Since you should not install the ExtendSim Cloud applications and files more than once on a single Cloud Server, there are only two situations where you might need to uninstall Cloud files:

- To completely remove all Cloud applications and files from a Cloud Server, for example because you want to install Cloud on a different server
- To change where ExtendSim Cloud is installed from one folder on a Cloud Server to another folder on that same Cloud Server

Completely removing all of ExtendSim Cloud from a Cloud Server

The ExtendSim Cloud Installation & Activation document lists all the applications and files that get installed. After installation, you might create other files and data.

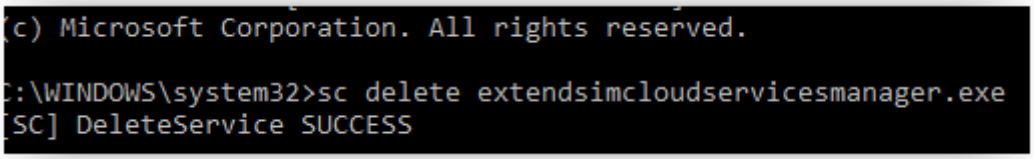
Save the files you've added

If you've been using Cloud and you're removing the Cloud application and files from one server so you can install Cloud on another server, save the following files so they can be used at the new location:

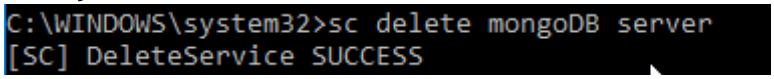
1. From the Program Files/ExtendSim10Cloud folder:
    a. From its Client Files folder, save the "Cloud scenario Manager.html" file, if you have made changes to it.
    b. From its Server Files folder, save:
        i. The ".env" file; it contains your configuration settings.
        ii. The "mongodb" folder; it contains the user group and authorization data.
2. From the Program Files/MongoDB/Server/4.2 folder, save the "data" file; it contains the user sign-in credentials.
3. From the Documents/ExtendSim10Cloud folder, save the:
    a. "ExtendSim Reference Models" folder
    b. "ExtendSim Scenario Folders" folders (if you want to save the model runs)
    c. Libraries folder (if you have put custom libraries there)
    d. Extensions folder (if you've added includes, etc.)

Then remove the Cloud applications from the Cloud Server:

1. Stop the services (instructions for Windows 10; others differ)
    a. Right-click on the Server's Start command and choose Computer Management
    b. In the list that appears, double-click Services and Applications
    c. Double-click Services
    d. Right-click on each of the two services below and select Stop:
        i. ExtendSimCloud Services Manager
        ii. MongoDB Server
    e. Close the Computer Management window
2. Remove the services
    a. Open Command Prompt as Administrator (enter CMD in the Search box and simultaneously click Ctrl+Shift+Enter, then click *Yes* in the User Account Control window to enable the Windows Command Processor)
    b. At the command line, enter *SC delete ExtendSimCloudservicesmanager.exe*. Then press the Enter key.



    c. Repeat the above instructions to remove the MongoDB service (*SC delete MongoDB Server)*



    d. If successful, the services will be removed and you can close the window
3. Uninstall these program files using Windows Settings or Control Panel
    a. ExtendSim10 Cloud
    b. MongoDB
    c. Node.js

4. Optional: delete these folders
    a. In Program Files: ExtendSimCloud and MongoDB
    b. In Documents: ExtendSimCloud

NOTE 1: to restart the ExtendSim Cloud Services Manager as a service on the same Cloud Server where you previously deleted the service, first delete the files that are in the Daemon folder located at Program Files/ExtendSimCloud/Server Files.

NOTE 2: this process removes the ExtendSim Cloud application and files from the Cloud Server. It does not remove the License Manager from the License Server; see below for that information.

Installing ExtendSim Cloud to a different folder on the same Cloud Server

Since multiple installations will conflict with each other, ExtendSim Cloud should only be installed one time on each Cloud Server. If you install ExtendSim Cloud again on a Cloud Server that already as an ExtendSim Cloud installation, the folder name for the second installation will be different than the original. In this case, the service for Cloud Services Manager won't run for the second installation, since it points to the location of the original *Cloud Server Files* folder.

To fix this, you must remove the service (ExtendSimCloud Services Manager), reconfigure it to point to the location of the new *Cloud Server Files* folder, and start the service again. NOTE: this explanation assumes that the Cloud Services Manager is running as a service on the Cloud Server.

1. Stop the *ExtendSimCloud Services Manager* service (instructions for Windows 10; others differ)
    a. Right-click on the Server's Start command and choose Computer Management
    b. In the list that appears, double-click Services and Applications
    c. Double-click Services
    d. Right-click on the service named *ExtendSimCloud Services Manager* and select Stop
    e. Close the Computer Management window
2. Remove the service
    a. With elevated privileges, run the *uninstall-windows-service.bat* file, located at Program Files/ExtendSim10Cloud/Server Files folder
    b. If successful, you should get a notice that the service has been removed
3. Point to the new directory
    a. Locate the install-windows-service.js script in the Program Files/ExtendSim10Cloud/Server Files folder
    b. Run the Windows Command Prompt program and change the directory to be the location of the **new** *Server Files* directory.
    c. At the command prompt, type **nodeCloud-install-windows-service.js** and click return.
    d. Click OK for each Windows prompt that occurs until the script finishes running.

You don't need to uninstall any applications, stop other services, or delete other files. You also don't need to remove, change, or reactivate the License Manager.

License Manager

General information

ExtendSim Cloud uses a License Manager (RLM.exe) to manage the Cloud license and control the number of instances of ExtendSim that are running. The License Manager is installed and activated on a License Server and runs as a service. It can be installed on a physical or virtual machine and it can be installed on the same device where ExtendSim Cloud is installed.

The License Manager is bound to the Server it is installed on, either through the hard disk hardware serial number, the BIOS id, or some other hostID. You must contact us before trying to "move" the License Manager to a different Server. See below for how to uninstall the License Manager before installing it on a new Server.

After activation, the License Manager runs as a service process that remains running as long as the system is up and restarts automatically when the License Server is rebooted.

When first launched and at midnight local time each day, the License Manager triggers a reread of its extendsim.lic license file, then opens port 5053 to receive requests from the Cloud Services Manager.

Moving the License Manager from the License Server

You must remove the License Manager from the current License Server in order to install and use it on a different License Server.

Before changing License Servers, you must get advance approval by contacting us at Sales@ExtendSim.com. We can then revoke the activation so you can install and activate on the new server.

After we've notified you that the license has been revoked, remove the License Manager from the original License Server:

1. Stop the service (instructions for Windows 10; others differ)
    a. Right-click on the Server's Start command and choose *Computer Management*
    b. In the list that appears, double-click *Services and Applications*
    c. Double-click *Services*
    d. Stop the service named *ExtendSimLicServer* (right-click and select Stop)
    e. Close the Computer Management Window
2. Remove the service
    a. Open Command Prompt as Administrator (enter CMD in the Search box and simultaneously click Ctrl+Shift+Enter), then click *Yes* in the User Account Control window to enable Windows Command Processor
    b. At the command line, enter *SC delete ExtendSimLicServer*. Then press the Enter key.
    c. If successful, the Service will be removed and you can close the window
3. Uninstall the application named *ExtendSim Floating License Manager* using Windows Settings or Control Panel.
4. You can also delete the folder *ExtendSimFloatingLicenseServer*, which is located by default at the root of the C drive.

After we've notified you that the license has been revoked, you can install and activate the License Manager on the new server, following the instructions in the Cloud Installation & Activation document.

NOTE: to remove ExtendSim Cloud from the Cloud Server, see Removing ExtendSim Cloud Files from a Cloud Server *on page 17*

Reprise License Manager (RLM) already installed on License Server

The Floating license of ExtendSim uses the Reprise License Manager (RLM.exe). If RLM is already running on the Server where you want to install the License Manager for an ExtendSim license, it would be best to use a different License Server.

If that isn't possible, there are two options to install on a Server that already has RLM running:

- If the currently running version of RLM is release 12 or greater, use **Case 1**, below, to add ExtendSim files to the existing installation.
- If the currently running version of RLM is earlier than release 12, use **Case 2**, below, to stop and remove the existing Service and instead use the RLM.exe that ships with ExtendSim.

Case 1: Do this if the currently running version of RLM is release 12 or greater

Step 1 • Installing the ISV Settings File on the Server

1. DO NOT INSTALL THE EXTENDSIM LICENSE MANAGER ON THE SERVER.
2. Obtain an ISV settings file (extendsim.set) from Imagine That Inc.
3. Place the extendsim.set file in the same directory where the other RLM-using product has installed RLM.exe as well as their license and settings files.

Step 2 • Activating the License

1. Follow the instructions for *Part 1b • Activate License Manager on the License Server* in the separate document ExtendSim Cloud InstallationActivation.
2. Activation will either be automatic or manual.
3. Make a copy of the extendsim.lic file for use in activating ExtendSim on each Cloud Server. Since RLM was already running as a service, the extendsim.lic license file will be located in the same folder where you placed the extendsim.set file.

Step 3 • Installing and Activating ExtendSim on the Cloud Server

1. Follow the instructions for *ExtendSim Cloud Installation and Activation*, in the document ExtendSim Cloud InstallationActivation
2. With this option, the License Manager (RLM) will run as a Service under the existing Service's name. The Service will not be named ExtendSimLicServer.

Case 2: If the currently running version of RLM is earlier than release 12

Step 1 • Stop and remove the currently running RLM Service

1. Determine the name of the existing RLM Service (the Service used by the other RLM-managed products).
2. Stop that Service. (Instructions for stopping that Service under Windows 10; others differ:)
   - Right-click on the Server's Start command and choose Computer Management.
   - In the list that appears, double-click Services and Applications.
   - Double-click Services.
   - In the list that appears, scroll to the name of the identified RLM Service.
   - Stop that Service (right-click select Stop).
   - Close the Computer Management window.
3. Remove the currently running RLM Service.
   - Open Command Prompt as Administrator (Enter CMD in the Search box and simultaneously click Ctrl+Shift+Enter).
   - Click Yes in the User Account Control window to enable Windows Command Processor.
   - At the command line, enter SC delete xyz (where xyz is the name of the identified RLM Service) and press Enter.
   - If successful, the Service will be removed and you can close the window.

Step 2 • Installing the License Manager (RLM.exe) used by ExtendSim

1. Use the ExtendSimServerLicenseManager installer (obtained from an email, a download, or an ExtendSim 10 USB drive) and follow its instructions to install the files on the Server.
2. By default, the installer creates a folder named ExtendSimFloatingLicenseServer that is placed at the root of the Server's C drive (C:/ExtendSimFloatingLicenseServer).
   - The folder can be installed on a drive other than the C drive, but it must be installed at the root of the drive and it must be named ExtendSimFloatingLicenseServer.

- The folder contains the RLM Svc folder and three files: extendsim.set, logfile, and RLM.exe.

3. After installation, the License Manager (RLM.exe) silently runs in the background as a Service named ExtendSimLicServer.
4. Before proceeding, we suggest you verify that ExtendSimLicServer is running as a Service.
5. Copy all the settings and license files (but not RLM.exe) from the previous RLM server folder to the ExtendSimFloatingLicenseServer folder, which by default will be located at the root of the C drive.

Step 3 • Activating the License Manager

1. Follow the instructions for *Part 1b* • *Activate License Manager on the License Server* in the separate document ExtendSim Cloud InstallationActivation.
2. Activation will either be automatic or manual.
3. Make a copy of the extendsim.lic file for use in activating ExtendSim on each Cloud Server. Since RLM was already running as a service, the extendsim.lic license file will be located in the same folder where you placed the extendsim.set file.

Step 4 • Installing and Activating ExtendSim on the Cloud Server

1. Follow the instructions for *ExtendSim Cloud Installation and Activation*, in the document ExtendSim Cloud InstallationActivation
2. With this option, the License Manager (RLM) will run as a Service named ExtendSimLicServer.

FAQs

Changing the port for the ISV Server

When the License Manager was activated, a high-number port (51006) was assigned as the ISV Server port, as shown in the Status command. If the ISV port is blocked you can either unblock it (preferred) or specify a different unblocked port as the ISV port.

We strongly suggest you do not change the ISV port from the default. If you specify a different ISV port, you might need to add it to the license file (extendsim.lic) every time you reactivate the license, such as after every Maintenance & Support renewal.

To set a different port than the default, use the RLSA window to edit your already-activated extendsim.lic license file:

1. Using a browser, open the RLSA window as indicated in the separate document Cloud Installing & Activating.
2. On the left side of the RLSA window, click the *Edit License Files* command. (Note: The "License File to Edit" should be extendsim.lic.)
3. In the window that appears, click the *Edit License File* button.
4. In the license file field, add to the "ISV extendsim" line the phrase "port=nnnnn", where "nnnnn" is the unblocked port number you've chosen. It should now read "ISV extendsim port=nnnnn".
5. In the RLSA window, click the *Update License File* button. The window should report "license file extendsim.lic written".
6. On the left-hand side of the RLSA window, click the *Reread/Restart Servers* command (this restarts the License Manager, not your Server).
7. For the ISV, select *extendsim*, then click the *Reread/Restart* button to restart the License Manager.
8. If successful, the window will display the message "Reread command sent to extendsim". This indicates that the new port will be added as an alternate port.
9. Click the Status command on the left side of the RLSA window to verify that the new ISV port is running. If it isn't listed or isn't running, and you've followed the directions correctly, you will need to restart the Server.
10. You do not need to copy the updated extendsim.lic file to each Cloud Server, but each Cloud Server does need to have that ISV port open for outbound access.

Error messages when installing, activating, or launching

These are the most common error messages. For complete information about these and other error messages, see the FAQs at ExtendSim.com/support/FAQ.

Bad HTTP transaction.

Your system is blocking receipt of the activation data. Manual activation is required.

Bad server hostname or bad hostname for network connect

These messages indicate that ExtendSim could not resolve an issue with the name of the License Server and you will need to use the IP address instead.

ComponentTransferData or Codebase failed to register

ExtendSim was not installed with administrative privileges/permission.

RightClickConnect database is missing error message.

A needed DLL was not registered in the Windows Registry due to inadequate privileges.

Documents folder not found or lp_solve(0) Error_system error code:2 error messages

Required files cannot be located or the end user does not have the correct permission for those files. The error code is a Windows system error; search online for more information.

The Activation Dialog appears after ExtendSim has already been activated

The license file (extendsim.lic) has been moved or removed or you have multiple installations of the same product (which is not allowed).

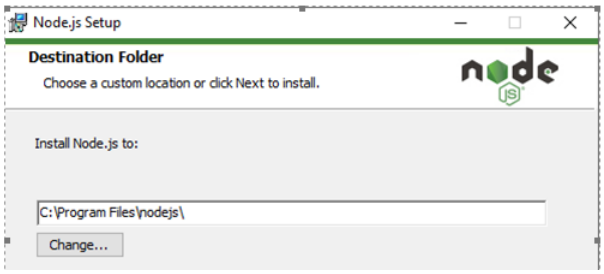
Connection refused at server (-111)" message

At least one of the two required ports (5053 or the ISV port) is blocked for outbound access on the Client device.

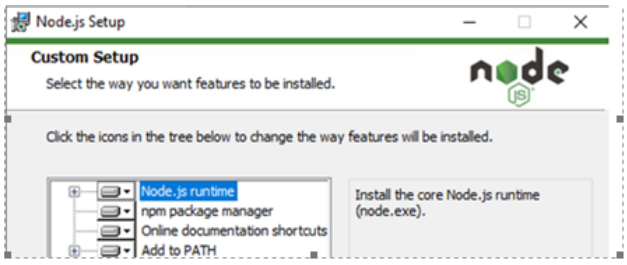
Screenshots

Installing Node.js

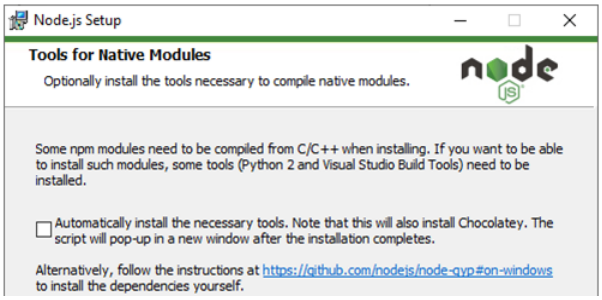
The default location is shown below



For the Custom Setup, choose Node.js runtime, as shown below

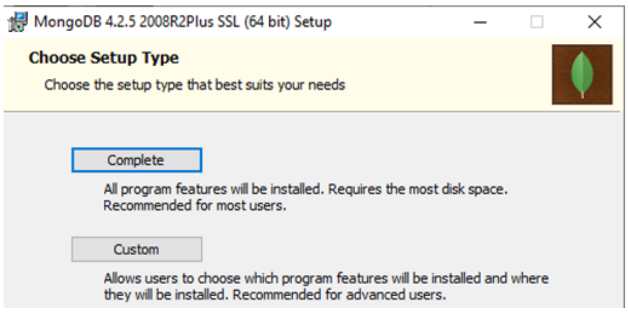


Do not install the additional tools; they are not needed for Cloud.

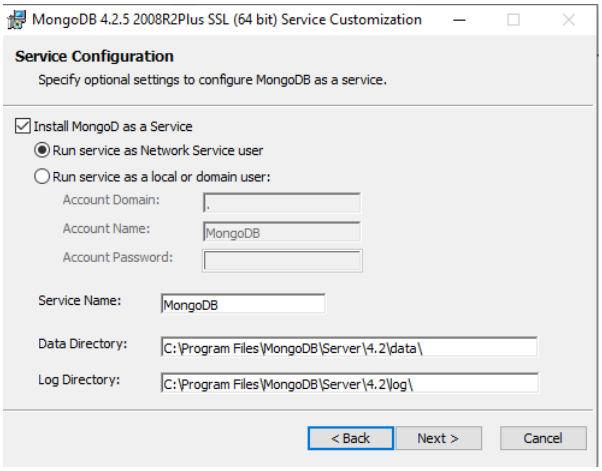


Installing MongoDB

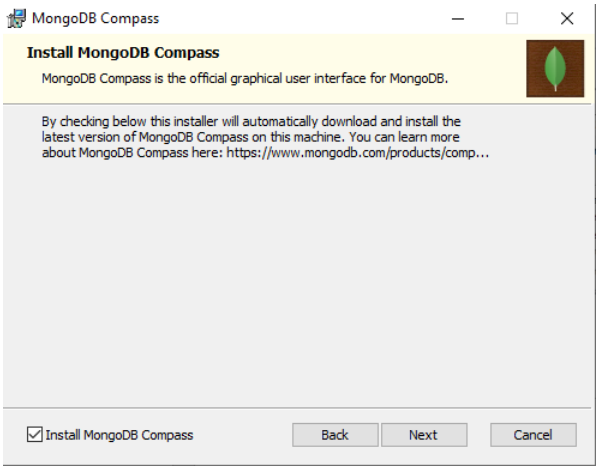
Click to choose the **Complete** setup type



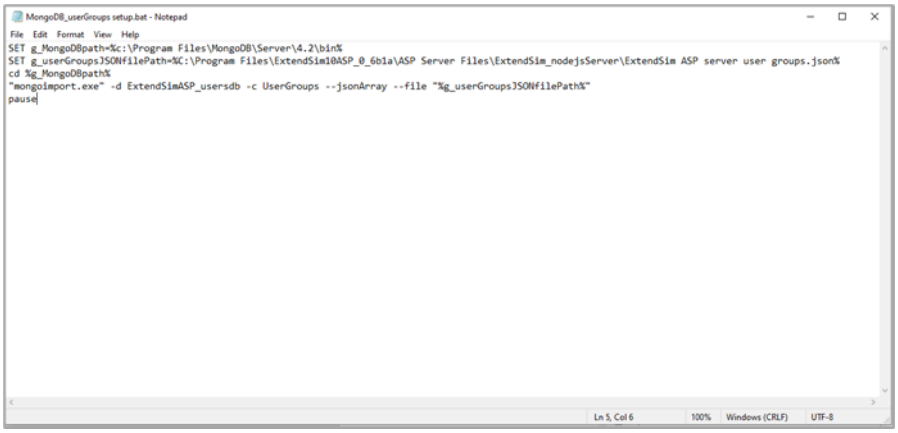
Use the default System Configuration

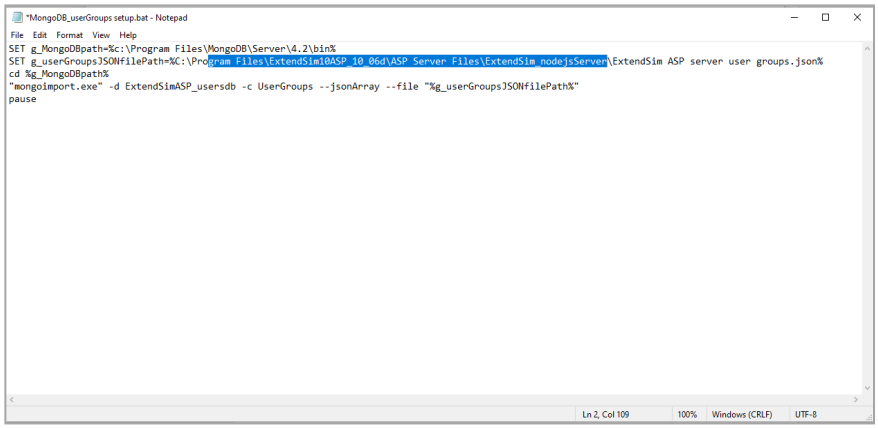


Uncheck the box so you do not download and install MongoDB Compass

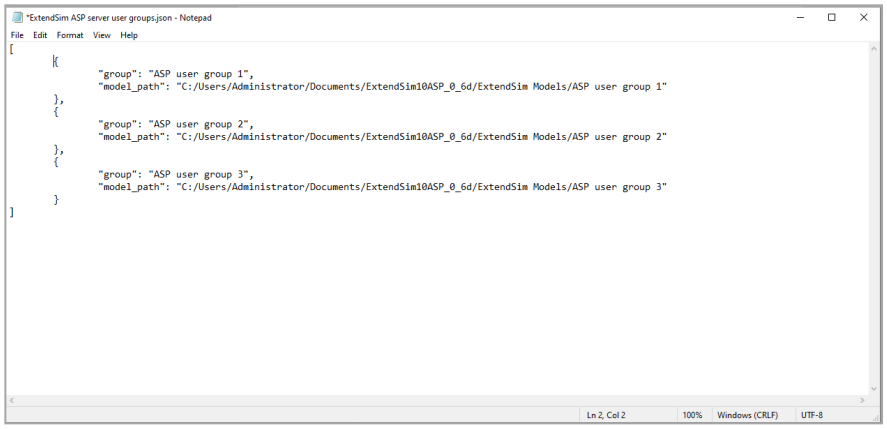


Setting up ExtendSim Cloud user groups in MongoDB

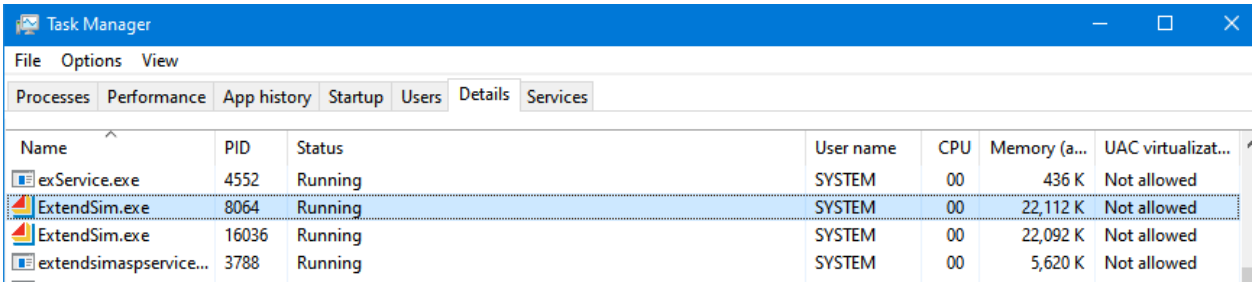




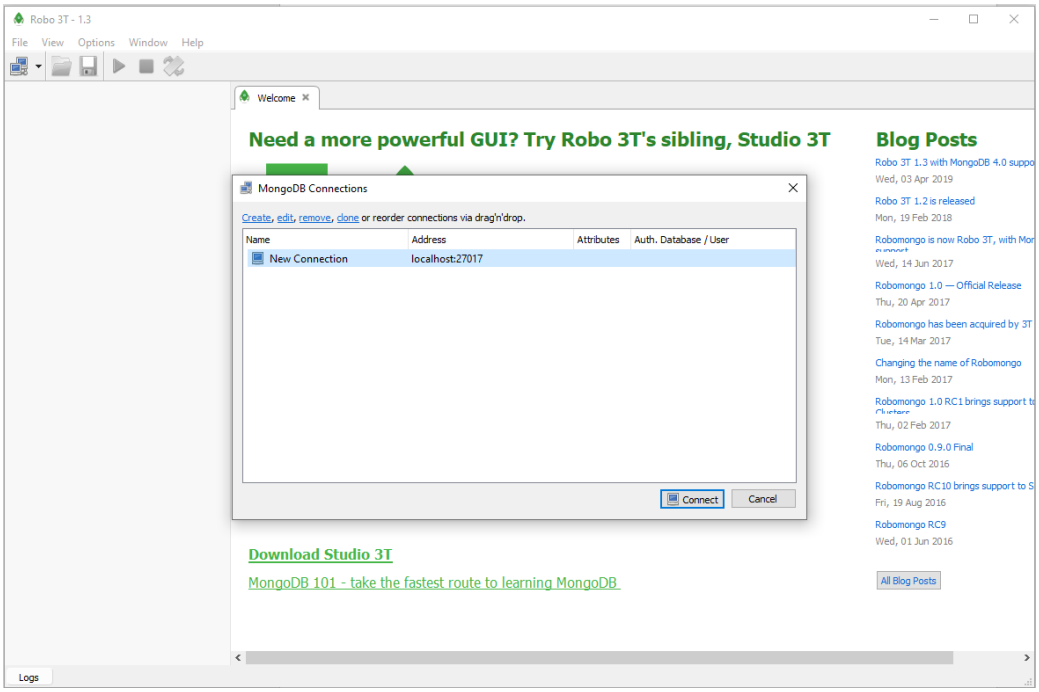
-

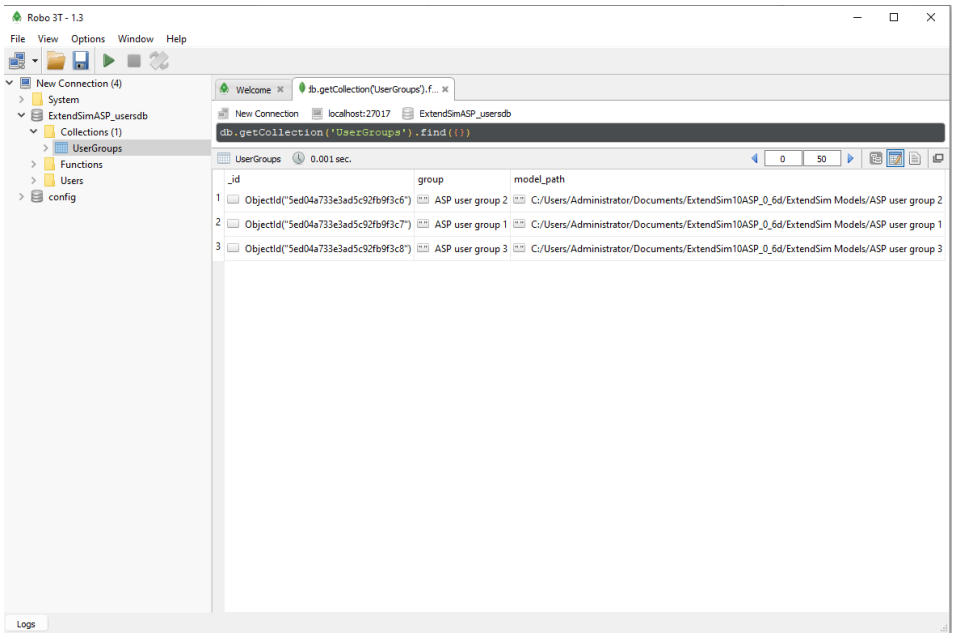


Using the Task Manager's Details tab to view services

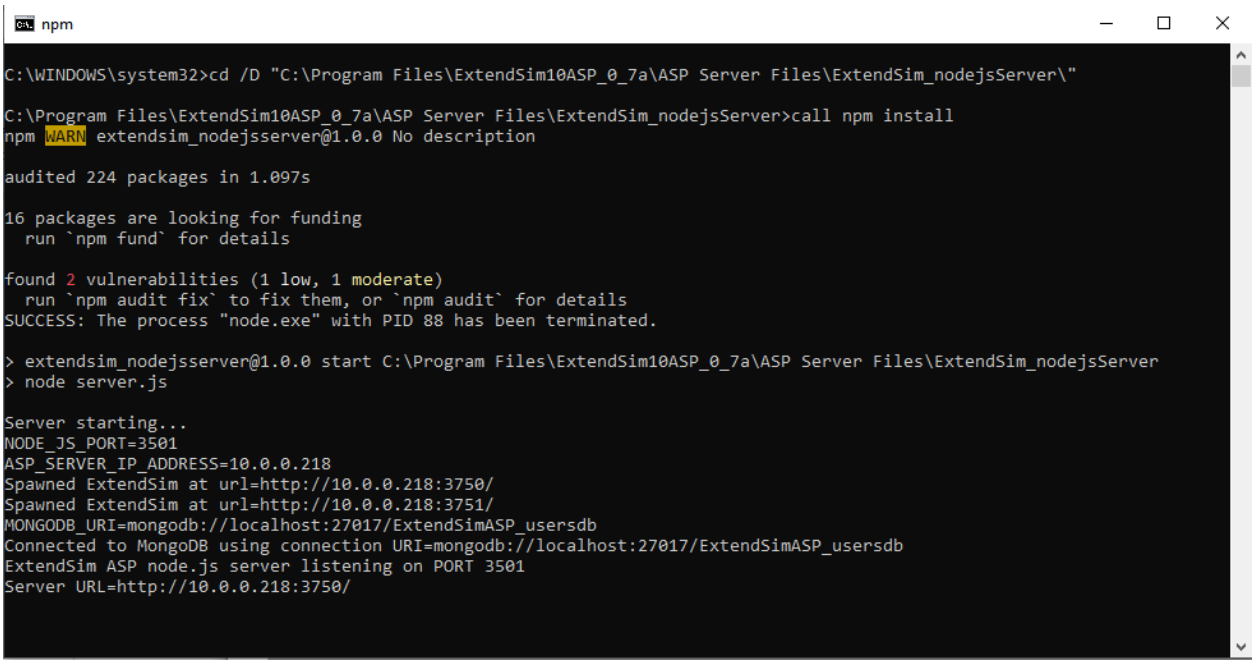


Using Robo 3T to view the ExtendSim_UsersDB database stored by MongoDB





Node.js server



Index