Elixys Database Schema

Contents

[Static initialization files 3](#_Toc300741375)

[System model 3](#_Toc300741376)

[Hardware map 3](#_Toc300741377)

[Robot positions 3](#_Toc300741378)

[MySQL tables 4](#_Toc300741379)

[Users 4](#_Toc300741380)

[Roles 4](#_Toc300741381)

[Sequences 4](#_Toc300741382)

[Reagents 5](#_Toc300741383)

[Sequence components 5](#_Toc300741384)

[Cassette 5](#_Toc300741385)

[Add 6](#_Toc300741386)

[Evaporate 6](#_Toc300741387)

[Transfer 6](#_Toc300741388)

[Elute 7](#_Toc300741389)

[React 7](#_Toc300741390)

[Prompt 7](#_Toc300741391)

[Move 8](#_Toc300741392)

[Install 8](#_Toc300741393)

[Comment 8](#_Toc300741394)

[Logs 8](#_Toc300741395)

[Status 8](#_Toc300741396)

[MySQL stored procedures 9](#_Toc300741397)

[Users 9](#_Toc300741398)

[LoadAllUsers 9](#_Toc300741399)

[LoadUser 9](#_Toc300741400)

[SaveUser 9](#_Toc300741401)

[DeleteUser 9](#_Toc300741402)

[LoadUserState 9](#_Toc300741403)

[SaveUserState 9](#_Toc300741404)

[Roles 9](#_Toc300741405)

[LoadAllRoles 9](#_Toc300741406)

[LoadRole 9](#_Toc300741407)

[SaveRole 9](#_Toc300741408)

[DeleteRole 9](#_Toc300741409)

[Sequences 9](#_Toc300741410)

[LoadAllSequences 9](#_Toc300741411)

[LoadSequence 9](#_Toc300741412)

[SaveSequence 9](#_Toc300741413)

[DeleteSequence 9](#_Toc300741414)

[Unit operations 10](#_Toc300741415)

[LoadAllUnitOperations 10](#_Toc300741416)

[LoadUnitOperation 10](#_Toc300741417)

[SaveUnitOperation 10](#_Toc300741418)

[DeleteUnitOperation 10](#_Toc300741419)

[Logs 10](#_Toc300741420)

[LoadLogMessages 10](#_Toc300741421)

[LogMessage 10](#_Toc300741422)

[Status 10](#_Toc300741423)

[LoadStatus 10](#_Toc300741424)

[SaveStatus 10](#_Toc300741425)

# Static initialization files

Plain-text INI files are used to store information that is instrument-specific and will not change throughout the life of the machine.

## System model

The system model file describes what hardware is present.

|  |  |  |
| --- | --- | --- |
| [root] | CoolingSystem | True if cooling system is present |
|  | VacuumSystem | True if vacuum system is present |
|  | ExternalSystems | True if the external systems (F-18 and HPLC) are present |
|  | PressureRegulators | Number of pressure regulators present |
|  | ReagentDelivery | True if the reagent delivery system is present |
| Reactor*N* | Stopcocks | Number of stopcocks on reactor *N* |

## Robot positions

The robot positions file contains the relative location of each name robot position and the absolute offsets for each reactor. See $/server/hardware/RobotPositions.ini for the current values.

|  |  |  |
| --- | --- | --- |
| [root] | ReagentXAxis | RoboNet axis for reagent robot X |
|  | ReagentZAxis | RoboNet axis for reagent robot Z |
|  | Reactor*N*Axis | RoboNet axis for reactor *N* |
| ReagentRobot | Reagent*N* | Relative X and Z coordinates for reagent *N* |
|  | ReagentDelivery*N* | Relative X and Z coordinates for reagent deliver position *N* |
| Reactors | Install | Relative Z coordinate for Install position |
|  | Transfer | Relative Z coordinate for Transfer position |
|  | React*N* | Relative Z coordinate for React*N* position |
|  | Add | Relative Z coordinate for Add position |
|  | Evaporate | Relative Z coordinate for Evaporate position |
|  | Radiation | Relative Z coordinate for Radiation position |
| Reactor*N* | CassetteXOffset | Absolute X offset for the cassette in reactor *N* |
|  | CassetteZOffset | Absolute Z offset for the cassette in reactor *N* |
|  | ReactorOffset | Absolute Z offset for reactor *N* |

## Hardware map

The hardware map file describes the location of the various system components in the PLC’s address space. See $/server/hardware/HardwareMap.ini for the current map.

|  |  |  |
| --- | --- | --- |
| **Section** | **Key** | **Description** |
| [root] | AnalogOutUnit | Unit number of the digital to analog PLC module |
|  | AnalogInUnit | Unit number of the analog to digital PLC module |
|  | Thermocontroller*N*Unit | Unit number of the Nth thermocontroller PLC module |
|  | DeviceNetUnit | Unit number of the device net PLC module |
|  | DigitalInOffset | Memory offset of the digital in PLC module |
|  | DigitalOutOffset | Memory offset of the digital out PLC module |
|  | PressureRegulatorSetSlope | Slope for converting between the set pressure in PSI and the corresponding analog out value |
|  | PressureRegulatorSetIntercept | Intercept for converting between the set pressure in PSI and the corresponding analog out value |
|  | PressureRegulatorActualSlope | Slope for converting between the analog in value and the corresponding pressure in PSI |
|  | PressureRegulatorActualIntercept | Intercept for converting between the analog in value and the corresponding pressure in PSI |
|  | VacuumGaugeSlope | Slope for converting between the analog in value and the corresponding vacuum in kPa |
|  | VacuumGaugeIntercept | Intercept for converting between the analog in value and the corresponding vacuum in kPa |
|  | RadiationDetectorSlope | Slope for converting between the analog in value and the corresponding radiation level in mCi |
|  | RadiationDetectorIntercept | Intercept for converting between the analog in value and the corresponding radiation level in mCi |
|  | VacuumPressure | Vacuum pressure (analog in) |
|  | CoolingSystemOn | Cooling system on (binary out) |
| PressureRegulator*N* | SetPressure | Set pressure (analog out) |
|  | ActualPressure | Actual pressure (analog in) |
| ReagentRobot | SetGripperUp | Set the gripper to up (binary out) |
|  | SetGripperDown | Set the gripper to down (binary out) |
|  | SetGripperOpen | Set the gripper to open (binary out) |
|  | SetGripperClose | Set the gripper to close (binary out) |
| F18 | Load | Turn on F-18 load gas (binary out) |
|  | Elute | Turn on F-18 elute gas (binary out) |
| HPLC | Load | Switch HPLC injection valve to load (binary out) |
| Reactor*N* | SetReactorUp | Set the reactor to the up position (binary out) |
|  | SetReactorDown | Set the reactor to the down position (binary out) |
|  | ReactorUp | Reactor up detection switch (binary in) |
|  | ReactorDown | Reactor down detection switch (binary in) |
|  | EvaporationNitrogenValve | Turn on evaporation gas (binary out) |
|  | EvaporationVacuumValve | Turn on evaporation vacuum (binary out) |
|  | TransferValve | Turn on transfer gas (binary out) |
|  | EluentValve | Turn on eluent transfer gas (Reactor 1 only, binary out) |
|  | Reagent1TransferValve | Turn on reagent 1 transfer gas (binary out) |
|  | Reagent2TransferValve | Turn on reagent 2 transfer gas (binary out) |
|  | Stopcock1ValvePosition1 | Turn on gas to move stopcock 1 to position 1 (binary out) |
|  | Stopcock1ValvePosition2 | Turn on gas to move stopcock 1 to position 2 (binary out) |
|  | Stopcock2ValvePosition1 | Turn on gas to move stopcock 2 to position 1 (Reactor 1 only, binary out) |
|  | Stopcock2ValvePosition2 | Turn on gas to move stopcock 2 to position 2 (Reactor 1 only, binary out) |
|  | Stopcock3ValvePosition1 | Turn on gas to move stopcock 3 to position 1 (Reactor 1 only, binary out) |
|  | Stopcock3ValvePosition2 | Turn on gas to move stopcock 3 to position 2 (Reactor 1 only, binary out) |
|  | StirMotor | Set stir motor speed (analog out) |
|  | RadiationDetector | Radiation detector (analog in) |
|  | TemperatureController*N* | Thermocontroller module and loop number for collet *N* |

# MySQL tables

All dynamic data is stored in a MySQL database in the tables that have the schema described below.

## Users

The Users table contains an entry for each user in the system.

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| UserID | int, unique, required, auto-increment | Number that is used by other tables in the database to uniquely identify the user |
| Username | string[30], unique, required | User’s login name |
| Password | string[30], required | Hash of the user’s password |
| FirstName | string[20], required | User’s first name |
| LastName | string[20], required | User’s last name |
| RoleID | int, required | Reference to the user’s role in the Roles table |
| ClientState | string[64], required | Current client state for the user |

## Roles

The roles table contains an entry for each user role that is recognized by the system.

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| RoleID | int, unique, required, auto-increment | Number that is used by other tables in the database to uniquely identify the role |
| RoleName | string[30], unique, required | Role name (e.g. “Administrator”) |
| Flags | int, required | Number that describes the role’s permissions. Each bit determines if the user had the ability to perform a certain action |

## Sequences

The sequence table contains an entry for each sequence in the system.

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| SequenceID | int, unique, required, auto-increment | Number that is used by other tables in the database to uniquely identify the sequence |
| Name | string[64], required | Sequence name (e.g. “FAC (high temp)”) |
| Comment | string[255] | Sequence comment |
| CreationDate | date, required, automatic | Date and time of the sequence creation |
| UserID | int,required | The ID of the user that created the sequence |
| FirstComponentID | int, required | The ID of the first component in this sequence or zero if the sequence does not contain any components |

## Reagents

The reagents table describes the reagent configuration for each sequence.

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| ReagentID | int, unique, required, auto-increment | Number that is used by other tables in the database to uniquely identify the reagent |
| SequenceID | int, required | The ID of the reagent sequence |
| ComponentID | int, required | The ID of the reagent component (should always be a cassette) |
| Position | string[2], required | Reagent position in the cassette (currently 1-10, A and B) |
| Available | boolean, required | Flag that indicates if a reagent is present in this position |
| Name | string[64], required | Reagent name (e.g. “MeCN”) |
| Description | string[255] | Reagent description (e.g. “acetonitrile”) |

## Components

The components table contains an entry for each cassette or unit operation of each sequence in the system.

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| ComponentID | int, unique, required, auto-increment | Number that is used by other tables in the database to uniquely identify the sequence component |
| SequenceID | int, required | The ID of the reagent sequence |
| NextComponentID | int, required | The ID of the next component in this sequence or zero if this is the last component |
| Type | string[20], required | String that describes the component type. Possible values are the names of the following subsections |
| Name | string[20] | Name of the component. The system will use the default name if none is provided by the user |
| Details | string[2048], required | JSON string that contains the details of the sequence component. See the subsections that follow for the contents of this field by component type |

### CASSETTE

The JSON string that describes each cassette contains the following information:

1. Reactor – Reactor associated with this cassette.
2. Available flag – Boolean value that indicates if this cassette is used.
3. Reagents – Array of reagent IDs.

**{**

**"type":"CASSETTE",**

**"reactor":1,**

**"available":True,**

**"reagents":**

**[**

**847,**

**848,**

**849,**

**850**

**]**

**}**

### ADD

The JSON string that describes the reagent addition unit operation contains the following information:

1. Reactor – Reactor where the reagent will be added.
2. Reagent – ID of the reagent to add.
3. Delivery position – Position where the reagent will be delivery (currently 1 or 2).
4. Delivery pressure – Pressure of the reagent delivery gas in PSI. A default value will be used if this field is empty.
5. Delivery time – Time to wait while delivering the reagent in seconds. A default value will be used if this field is empty.

**{**

**"type":"ADD",**

**"reactor":1,**

**"reagent":344,**

**"deliveryposition":2,**

**"deliverypressure":10,**

**"deliverytime":4**

**}**

### EVAPORATE

The JSON string that describes the evaporate unit operation contains the following information:

1. Reactor – Reactor where the evaporation will be performed.
2. Duration – The length of the evaporation in seconds.
3. Evaporation temperature – The evaporation temperature in Celsius.
4. Evaporation pressure – The pressure of the evaporation gas in PSI. A default value will be used if this field is empty.
5. Final temperature – The final temperature in Celsius.
6. Stir speed – The stir speed in arbitrary units.

**{**

**"type":"EVAPORATE",**

**"reactor":1,**

**"duration":300,**

**"evaporationtemperature":165.0,**

**"evaporationpressure":15,**

**"finaltemperature":35.0,**

**"stirspeed":500**

**}**

### TRANSFER

The JSON string that describes the transfer unit operation contains the following information:

1. Reactor – Reactor whose contents will be transferred.
2. Target – The target ID.

**{**

**"type":"TRANSFER",**

**"reactor":1,**

**"target":321,**

**}**

### ELUTE

The JSON string that describes the elute unit operation contains the following information:

1. Reactor – Reactor where the eluted contents will be delivered.
2. Reagent – The ID reagent used to elute.
3. Target – The target ID.

**{**

**"type":"ELUTE",**

**"reactor":1,**

**"reagent":12,**

**"target":321,**

**}**

### REACT

The JSON string that describes the react unit operation contains the following information:

1. Reactor – Reactor where the reaction will be performed.
2. Position – The react position (currently 1 or 2).
3. Duration – The length of the reaction in seconds.
4. Reaction temperature – The reaction temperature in Celsius.
5. Final temperature – The final temperature in Celsius.
6. Stir speed – The stir speed in arbitrary units.

**{**

**"type":"REACT",**

**"reactor":1,**

**"position":1,**

**"duration":600,**

**"reactiontemperature":165.0,**

**"finaltemperature":35.0,**

**"stirspeed":500,**

**}**

### PROMPT

The JSON string that describes the prompt unit operation contains the following information:

1. Message – Text to display to the user.

**{**

**"type":"PROMPT",**

**"message":"Please take a sample for analysis",**

**}**

### MOVE

The JSON string that describes the move unit operation contains the following information:

1. Reactor – Reactor that will be moved.
2. Position – Position to which the reactor will be moved.
3. State – The state of the reactor.

**{**

**"type":"MOVE",**

**"reactor":1,**

**"position":"Evaporate",**

**"state":"Open",**

**}**

### INSTALL

The JSON string that describes the install unit operation contains the following information:

1. Reactor – Reactor that will be moved to the install position.
2. Message – Text to display to the user.

**{**

**"type":"INSTALL",**

**"reactor":1,**

**"message":"Please take a sample for analysis",**

**}**

### COMMENT

The JSON string that describes the comment unit operation contains the following information:

1. Comment – User-specified comment.

**{**

**"type":"COMMENT",**

**"comment":"Bromination and cytosine coupling",**

**}**

## System log

The system log table contains log messages about system operation from the beginning of the machine’s life to present.

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| LogID | int, unique, required, auto-increment | Number that uniquely identifies the log message |
| Date | date, required, automatic | Date and time when message was logged |
| Level | int, required | Integer that indicates the log level:   * 0 – Error * 1 – Warning * 2 – Information * 3 – Debug |
| Message | string[1024], required | Log message |

## Run log

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| LogID | int, unique, required, auto-increment | Number that uniquely identifies the log message |
| Date | date, required, automatic | Date and time when message was logged |
| Level | int, required | Integer that indicates the log level:   * 0 – Error * 1 – Warning * 2 – Information * 3 – Debug |
| UserID | int, required | ID of the user that was operating the system when the message was logged |
| SequenceID | int, required | ID of the sequence that was being run when the message was logged |
| ComponentID | int | ID of the sequence component that was being executed when the message was logged |
| Message | string[1024], required | Log message |

## Status

The status table is used to store the last know state of the system. This information is only used if the program is terminated unexpectedly. This table will contain only a single row. The contents and format of this table will be determined at a later point in the system development.

# MySQL stored procedures

All data in the MySQL database is accessed through stored procedures as described below. Actual implementation of the stored procedures should be on an as-needed basis since not all of the procedures listed below may be necessary.

## Users

### LoadAllUsers

### LoadUser

### SaveUser

### DeleteUser

### LoadUserState

### SaveUserState

## Roles

### LoadAllRoles

### LoadRole

### SaveRole

### DeleteRole

## Sequences

### LoadAllSequences

### LoadSequence

### SaveSequence

### DeleteSequence

## Unit operations

### LoadAllUnitOperations

### LoadUnitOperation

### SaveUnitOperation

### DeleteUnitOperation

## Logs

### LoadLogMessages

### LogMessage

## Status

### LoadStatus

### SaveStatus