Elixys Database Schema

Contents

[Static initialization files 2](#_Toc313864237)

[System model 2](#_Toc313864238)

[Robot positions 2](#_Toc313864239)

[Hardware map 3](#_Toc313864240)

[MySQL tables 5](#_Toc313864241)

[Users 5](#_Toc313864242)

[Roles 5](#_Toc313864243)

[Sequences 5](#_Toc313864244)

[Reagents 6](#_Toc313864245)

[Components 6](#_Toc313864246)

[System log 7](#_Toc313864247)

[Run log 7](#_Toc313864248)

[Status log 8](#_Toc313864249)

[Status 8](#_Toc313864250)

# 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 |
|  | Valves | True if the valves (gas transfer, 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 |
|  | ReagentYAxis | RoboNet axis for reagent robot Y |
|  | Reactor*N*Axis | RoboNet axis for reactor *N* |
| ReagentRobot | Reagent*N* | Relative X and Y coordinates for reagent *N* |
|  | ReagentDelivery*N* | Relative X and Y coordinates for reagent deliver position *N* |
|  | Elute | Relative X and Y coordinate for elute position |
| Reactors | Install | Relative Y coordinate for Install position |
|  | Transfer | Relative Y coordinate for Transfer position |
|  | React*N* | Relative Y coordinate for React*N* position |
|  | Add | Relative Y coordinate for Add position |
|  | Evaporate | Relative Y coordinate for Evaporate position |
| Reactor*N* | CassetteXOffset | Absolute X offset for the cassette in reactor *N* |
|  | CassetteYOffset | Absolute Y offset for the cassette in reactor *N* |
|  | ReactorOffset | Absolute Y 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 |
|  | CoolingSystemOn | Cooling system on (binary out) |
|  | VacuumPressure | Vacuum pressure (analog in) |
|  | VacuumSystemOn | Vacuum 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) |
|  | SetGasTransferUp | Set the gas transfer to up (binary out) |
|  | SetGasTransferDown | Set the gas transfer to down (binary out) |
|  | GripperUp | Gripper up sensor (binary in) |
|  | GripperDown | Gripper down sensor (binary in) |
|  | GripperOpen | Gripper open sensor (binary in) |
|  | GripperClose | Gripper close sensor (binary in) |
|  | GasTransferUp | Gas transfer up sensor (binary in) |
|  | GasTransferDown | Gas transfer down sensor (binary in) |
| Valves | GasTransferValve | Turn on gas transfer valve (binary out) |
|  | F18Load | Turn on F-18 load valve (binary out) |
|  | HPLCLoad | 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 sensor (binary in) |
|  | ReactorDown | Reactor down sensor (binary in) |
|  | Stopcock1ValveCCW | Set stopcock 1 to CCW (binary out) |
|  | Stopcock1ValveCW | Set stopcock 1 to CW (binary out) |
|  | Stopcock2ValveCCW | Set stopcock 2 to CCW (Reactor 1 only, binary out) |
|  | Stopcock2ValveCW | Set stopcock 2 to CW (Reactor 1 only, binary out) |
|  | Stopcock3ValveCCW | Set stopcock 3 to CCW (Reactor 1 only, binary out) |
|  | Stopcock3ValveCW | Set stopcock 3 to CW (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 |
| Type | string[20] | Type of sequence (“Saved” or “History”) |
| 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 |
| ComponentCount | int, required | Number of component in this sequence |
| Valid | boolean, required | Flag that indicates if the sequence is valid |
| Dirty | boolean, required | Flag that indicates if the Valid flag is up to date |

## 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 |
| PreviousComponentID | int, required | The ID of the previous component in this sequence or zero if this is the first component |
| 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[64] | 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 ElixysHTTPProtocol.docx for the contents of this field by component type |

## System log

The system log table contains log messages from the core and web server regarding system operation.

|  |  |  |
| --- | --- | --- |
| **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 generated the log message |
| Message | string[1024], required | Log message |

## Run log

The run log table contains messages about the progress of each run.

|  |  |  |
| --- | --- | --- |
| **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 log

The status log table contains detailed information about the state of the system throughout each run.

|  |  |  |
| --- | --- | --- |
| **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 |
| SystemState | string[4096], required | JSON string describing the state of the system |

## 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.