Elixys Core Server Interface

Contents

[Overview 2](#_Toc315285309)

[GetServerState() 2](#_Toc315285310)

[RunSequence() 3](#_Toc315285311)

[RunSequenceFromComponent() 4](#_Toc315285312)

[PauseSequence() 4](#_Toc315285313)

[ContinueSequence() 4](#_Toc315285314)

[WillSequencePause() 5](#_Toc315285315)

[IsSequencePaused() 5](#_Toc315285316)

[AbortSequence() 5](#_Toc315285317)

[PauseTimer() 5](#_Toc315285318)

[ContinueTimer() 6](#_Toc315285319)

[StopTimer() 6](#_Toc315285320)

[DeliverUserInput() 6](#_Toc315285321)

[CLIExecuteCommand() 7](#_Toc315285322)

[CLISendCommand() 7](#_Toc315285323)

[CLIAbortUnitOperation() 7](#_Toc315285324)

[CLIGetState() 7](#_Toc315285325)

# Overview

This document describes the interface implemented by the core server and consumed by the web server processes via RPC. This interface is used when the user is running the system or viewing a run that is in progress. All of the functions take the name of the current user as a parameter for logging purposes. Many of the function return a simple Boolean value to indicate success or failure. The system logs will contain the detailed error information.

GetServerState() – Returns the state of the server.

Parameters:

* Username – Name of the current user.

Returns:

* Run state:
  + Status – String describing the current system status. Possible values are:
    - “Idle” – The system is not in use. None of the additional run state fields will be set.
    - “Running” – The system is executing a unit operation.
    - “Paused” – The system has paused and is waiting on user input.
  + Username – Name of the user that is operating the system.
  + Sequence ID – ID of the sequence that the system is running.
  + Component ID – ID of the component that the system is currently on.
  + Prompt – Describes any prompt that is being displayed in association with the run
* Hardware state:
  + Pressure regulators – Details of each pressure regulator in the system:
    - Name – String describing the pressure regulator (e.g. “Pneumatic pressure”).
    - Pressure – The actual pressure in PSI.
  + Cooling – Boolean value that specifies if the cooling system is on.
  + Vacuum – The actual vacuum in kPa.
  + Reagent Robot – Details of the reagent robot:
    - Position:
      * Cassette – Number of the cassette the robot is over or zero if indeterminate.
      * Reagent position – Number of the reagent position the robot is over or zero if not over a reagent position.
      * Delivery position – Number of the delivery position the robot is over or zero if not over a delivery position.
    - Actuator – String that specifies the state of the actuator. Possible values are “up”, “down” and “indeterminate”.
    - Gripper – String that specifies the state of the gripper. Possible values are “open”, “closed” and “indeterminate”. Note that “closed” indicates a vial is between the fingers while “indeterminate” indicates no vial present.
  + Reactors – Details of each reactor:
    - Number – The reactor number.
    - Temperature – The actual temperature of the reactor in Celsius.
    - Position:
      * Horizontal – String describing the horizontal position, e.g. “Add”, “Evaporate”.
      * Vertical – String describing the vertical position. Possible values are “up”, “down” and “indeterminate”.
    - Activity – The last know radiation activity level of the reactor in millicuries.
    - Activity time – The time the activity was measured.
    - Evaporation – Boolean value that indicates if the evaporation values are open.
    - Transfer valve – Boolean value that indicates if the transfer valve is open.
    - Transfer position – String that describes the transfer stopcock position. Possible values are “Waste” and “Out”.
    - Reagent 1 transfer valve – Boolean value that indicates if the first reagent transfer valve is open.
    - Reagent 2 transfer valve – Boolean value that indicates if the second reagent transfer valve is open.
    - Stir speed – Speed of the stir motor.
    - Video – URL of the video stream.
  + Reactor 1 will have the following additional fields:
    - Column position – String that describes the position of the column stopcocks. Possible values are “Load” and “Elute”.
    - F18 transfer valve – Boolean value that indicates if the F18 transfer valve is open.
    - Eluent transfer value – Boolean value that indicates if the eluent transfer valve is open.

RunSequence() – Loads a sequence from the database and runs it.

Parameters:

* Username – Name of the current user.
* Sequence ID – Unique ID of the sequence to run.

Returns:

* Result – Boolean value (true on success, false otherwise).

RunSequenceFromComponent() – Loads a sequence from the database and runs it starting with the specified component.

Parameters:

* Username – Name of the current user.
* Sequence ID – Unique ID of the sequence to run.
* Component ID – Unique ID of the first component to run.

Returns:

* Result – Boolean value (true on success, false otherwise).

PauseSequence() – Flags the sequence to pause after the current unit operation.

Parameters:

* Username – Name of the current user.

Returns:

* Result – Boolean value (true on success, false otherwise).

ContinueSequence() – Continues the sequence run if paused or clears the pause flags if set but not yet paused.

Parameters:

* Username – Name of the current user.

Returns:

* Result – Boolean value (true on success, false otherwise).

WillSequencePause() – Determines if the sequence will paused after the current unit operation.

Parameters:

* Username – Name of the current user.

Returns:

* Result – True if the sequence will pause, false otherwise.

IsSequencePaused() – Determines if the sequence execution is paused.

Parameters:

* Username – Name of the current user.

Returns:

* Result – True if the sequence is currently paused, false otherwise.

AbortSequence() – Quickly turns off the heaters and terminates the run, leaving the system in its current state.

Parameters:

* Username – Name of the current user.

Returns:

* Result – Boolean value (true on success, false otherwise).

PauseTimer() – Pauses the timer if the unit operation has one running. This can be used as a mechanism to extend React or Evaporate or to increase the Add delivery time.

Parameters:

* Username – Name of the current user.

Returns:

* Result – Boolean value (true on success, false otherwise).

ContinueTimer() – Continues the timer if the unit operation has one paused.

Parameters:

* Username – Name of the current user.

Returns:

* Result – Boolean value (true on success, false otherwise).

StopTimer() – Cuts the timer short if the unit operation has one running. This can be used as a mechanism to shorten React or Evaporate or to decrease the Add delivery time.

Parameters:

* Username – Name of the current user.

Returns:

* Result – Boolean value (true on success, false otherwise).

DeliverUserInput() – Delivers user input to the waiting unit operation.

Parameters:

* Username – Name of the current user.

Returns:

* Result – Boolean value (true on success, false otherwise).

CLIExecuteCommand() – Executes the given command for the command line interface.

Parameters:

* Username – Name of the current user.
* Command – String containing the command to execute.

Returns:

* Result – Empty string if successful or a description of the error otherwise.

CLISendCommand() – Sends the given command to the PLC for the command line interface.

Parameters:

* Username – Name of the current user.
* Command – String containing the command to send to the PLC.

Returns:

* Result – Empty string if successful or a description of the error otherwise.

CLIAbortUnitOperation() – Aborts the current unit operation for the CLI.

Parameters:

* Username – Name of the current user.

Returns:

* Result – Empty string if successful or a description of the error otherwise.

CLIGetState() – Returns the current state of the system as a string.

Parameters:

* Username – Name of the current user.

Returns:

* Result – String describing the current state of the system if successful or the error otherwise.