Simio
Forward Thinking

# Simio Portal WebAPI Note: RunSimioPortalExpConsole

Last Update: 4 April 2023

## Contents

Simio
Forward Thinking

# RunSimioPortalExpConsole

## Overview

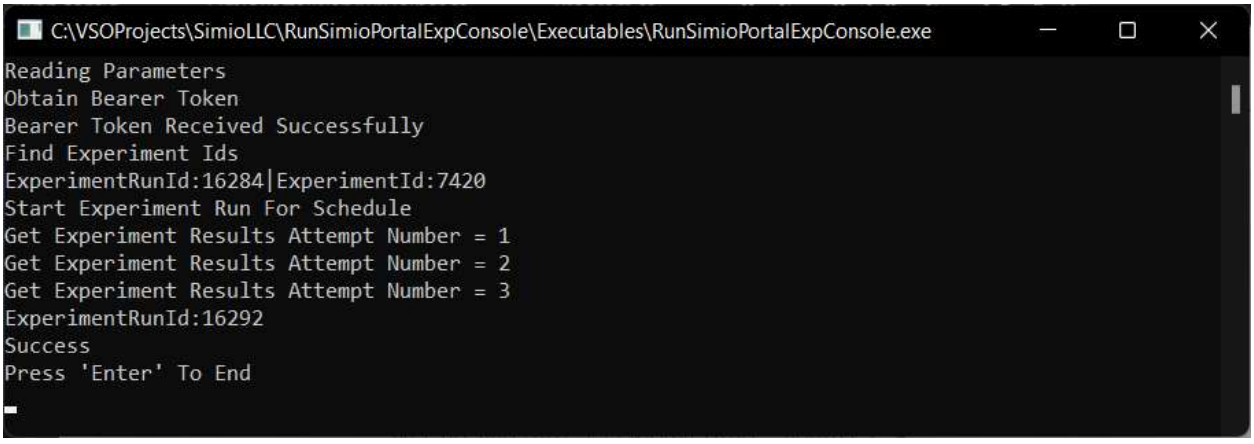
This .NET console application example illustrates external programs can use the Simio Portal WebAPI to create automated workflows for importing running, exporting, and publishing experiments and plans. Please see the Simio Portal API Help for more details on the Simio Portal Web API. See:

https://<yourSimioPortalURL>/Home/APIHelp

## Installation Of Executables and Running RunSimioPortalExpConsole

Download from the Executables folder and copy the files into a folder. You might need to right-click on every file, select properties and make sure the files are unlocked.

Next, open a command prompt. From the command prompt, navigate to the folder. To run the RunSimioPortalExpConsole, just type "RunSimioPortalExpColsole" and press Enter. The RunSimioPortalExpConsole will read the values in the RunSimioPortalExpConsole.exe.config when running the executable to determine what actions to take based on its settings. Here is an example of a successful schedule run.




Note: you will need to update the URL and Personal Access Token in the RunSimioPortalExpConsole.exe.config. See Settings section below. You will also need to upload the SchedulingLaborEfficiencies.spfx example from C:\Program Files\Simio LLC\Simio\Examples onto portal to have the RunSimioPortalExpConfig to run successfully.

## Settings

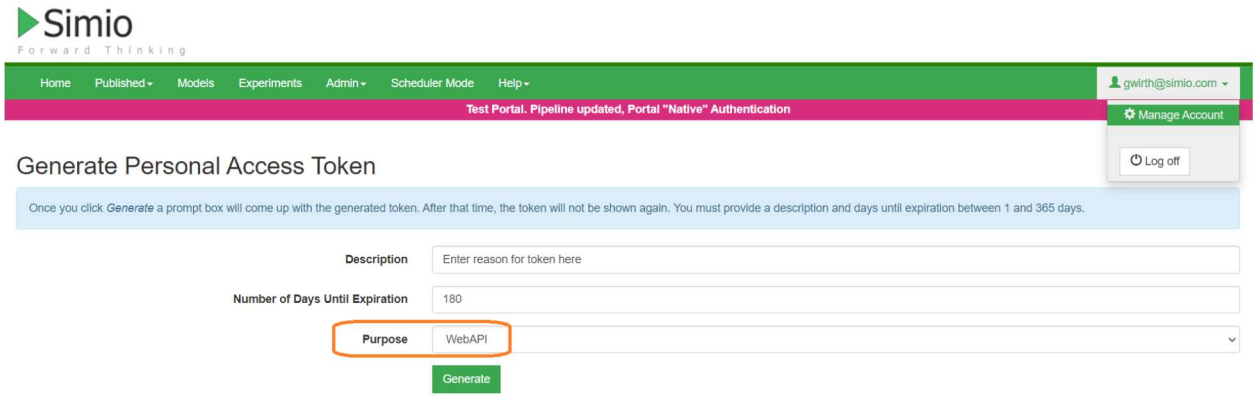
The settings are used to control the workflow of the RunSimioPortalExpConsole. These settings are configured in the RunSimioPortalExpConsole.exe.config file. The RunSimioPortalExpConsole.exe.config must be in the same folder as the RunSimioPortalExpConsole.exe.

Here is list of the settings contained in the RunSimioPortalExpConsole.exe.config. The values contained in the parentheses are command line overrides. The settings that are in *italic* require Simio Portal to be configured to allow private web API calls…. See Appendix A for steps to configure Simio Portal to allow private web API calls. The next section will cover how to use the command line overrides.

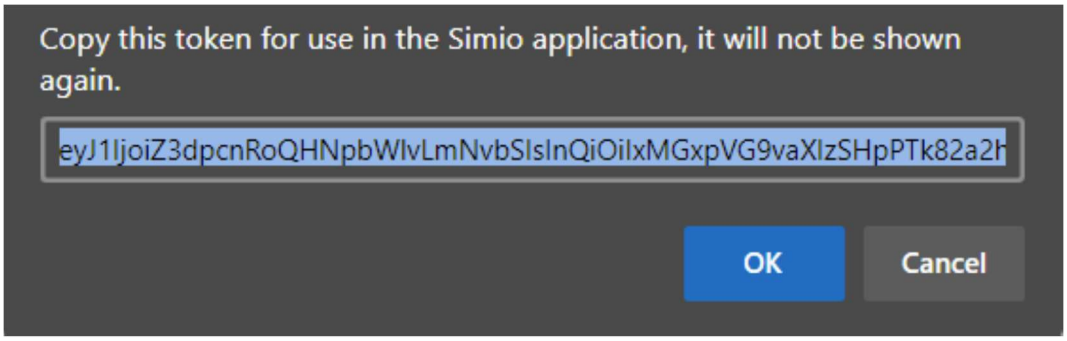
**General Settings**

URL (-u) – string – This is the URL so Simio Portal (e.g. https://<YourSimioPortalURL>/ ).

Personal Access Token (-pat) – string – This is the personal access token used to authenticate the calling of the Simio Portal Web API. The personal access tokens are setup under your username under "Manage Account". From there, select "Generate new token". Then, enter Description, Number of Days Until Expiration and Purpose. You must choose "WebAPI" as Purpose to use the RunSimioPortalExpConsole.




Once generated, copy the token text and copy it into this setting.




Retry Results Interval Seconds (-ris) – integer – The RunSimioPortalExpConsole will poll the Simio Portal Web API until a run completes. This parameter is the interval that the Simio Portal Web API is polled for results.

Retry Results Max Attempts (-rma) – integer – The number of attempts that the RunSimioPortalExpConsole will try to get results before it returns in error.

WaitAtEnd (-w) – boolean – used to make the console app wait at the end of the run.

**Authentication Settings**

Authentication Type (-at) – string – the valid types are None and Basic.

Domain (-d) – string – optional parameter to include the domain as part of the authentication.

UserName (-u) – string – optional parameter for user name.

Password (-p) – string – optional parameter for password. This value is not encrypted.

Bearer Token Refresh Interval Minutes (-btr) – integer – number of minutes that the Simio Portal Bearer Token will be used prior to refreshing. The default value is 15. The Simio Portal Bearer Token is valid for 30 minutes.

**Project Settings**

Project Name (-prj) – string – project name in portal.

**Scheduling Only Settings**

Run Schedule Plan Scenario Name (-rsn) – string – existing plan name that is used to run the schedule. You need to have an existing plan name to run the plan. The user must be the owner of the plan to access it.

*Run Schedule Control Values Array (-rsc) – string – name values pairs that will set the control value prior to running the plan. The control value settings are defined as follows "ABC=123|EFG=456" where equals "=" separates the name from the value and pipe "|" separates the control value settings from one another.*

Run Schedule (-rs) – boolean -determines whether to run plan.

Run Risk Analysis (-rra) – boolean – determine whether to run risk analysis.

Publish Schedule Run (-psr) – boolean – determines when to publish plan. This can be run independently from running the schedule.

Publish Schedule Name (-psn) – string – name of the publish plan.

Publish Schedule Description (-psd) – string – description of the publish plan.

*Run Length Days (-rdl) – integer – number of days to run the model. This value will only be used if the Start Time Selection is set to a valid string and this value is greater than zero.*

*Start Time Selection (-sts) – string – determines when to start the schedule based on current time type. The valid strings are Second, Minute, Hour, Day, Week, Month and Year. A valid string needs to be entered and the Run Length Days need to be greater than zero*

*Import All Tables (-iat) – boolean – determines whether to import all tables before scheduling run. Run schedule must be set to true to use this setting.*

*Export All Tables And Logs (-eat) – determine whether to export all tables. This can be run independently from running the schedule.*

**Experiments Settings**

Run Experiment Run Description (-erd) – string – description of the experiment run. Currently, the RunSimioPortalExpConsole is only setup to run from an existing experiment run.

NOTE: In the RunSimioPortalExpConole.exe.config, the experiment run name is OptQuestRun. This was created from the OptQuest experiment SchedulingLaborEfficiencies example. When run from the RunSimioPortalExpConsole, it will not run OptQuest. This capability is currently not available from this app. It will run the experiment based on the detail in the Run Experiment Scenarios JSON setting.

Run Experiment Scenarios JSON (-rej) – string – JSON string containing the experiment run detail.

Run Experiment (-re) – boolean - determines whether to run experiment.

*Publish Experiment Run (-psr) – boolean - determines whether to publish experiment. This can be run independently from running the experiment.*

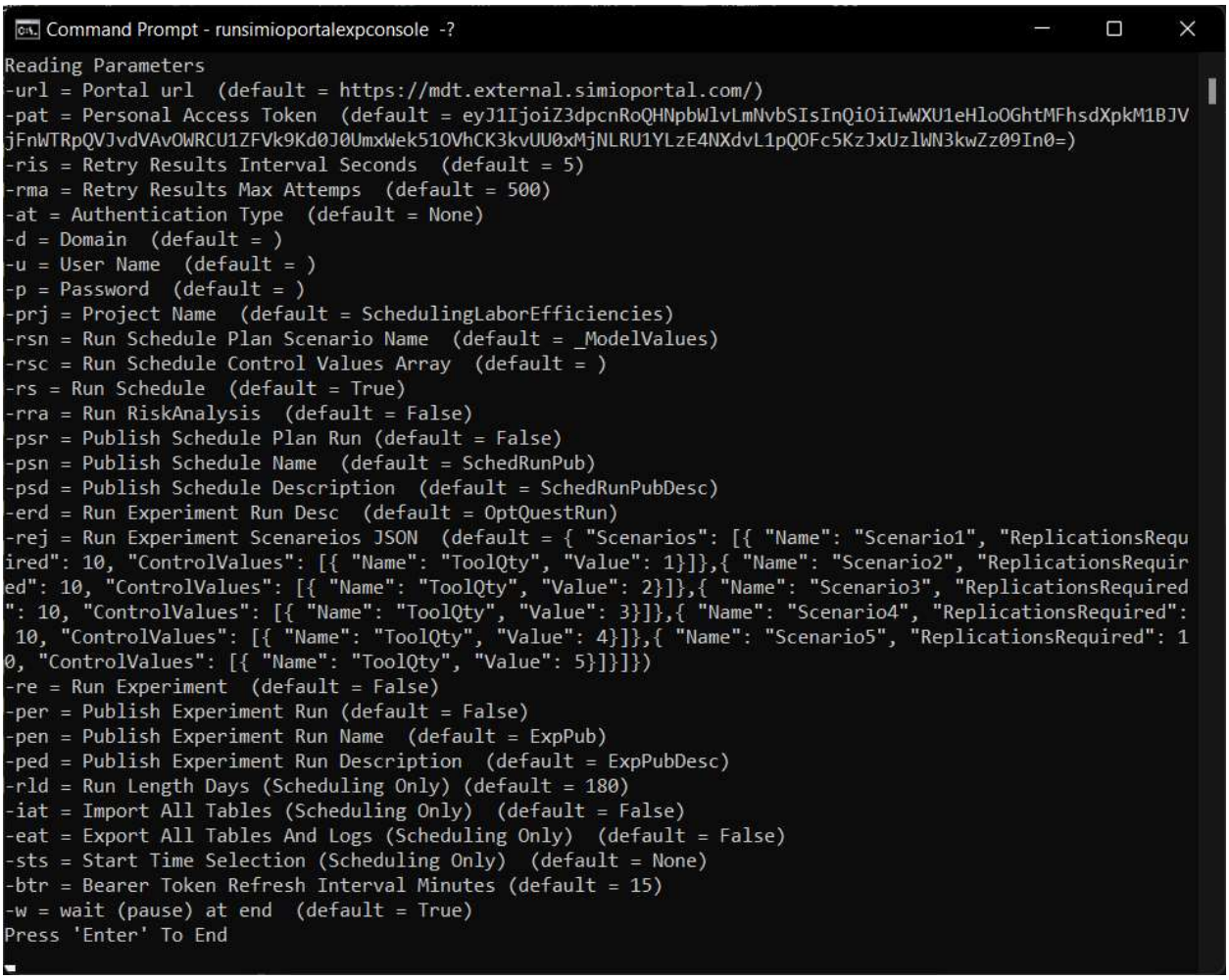
*Publish Experiment Run Name (-pen) – string - name of the published experiment.*

*Publish Experiment Run Description (-ped)- string – description of the published experiment.*

## Specifying Command Line Settings

Any of the setting can also be set by specifying them as part of the command line run. In addition to the Settings section, you can find the setting from running “RunSimioPortalExpConsole -?”. Here are

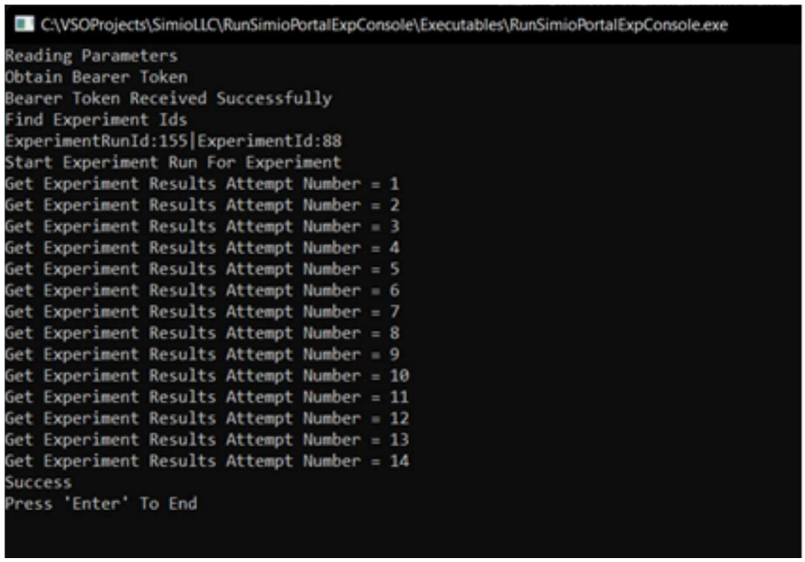
the results. The results also show the current setting values in the RunSimioPortalExpConsole.exe.config.




To run the existing configuration as an experiment instead of schedule, first you need to create an experiment run from the OptQuest experiment called “OptQuestRun”. Next, the following command line setting would be applied. “-rs False” will disable the schedule run and “-re True” will enable the experiment run. The existing configuration already has the Run Experiment Scenarios JSON that will define the scenarios.

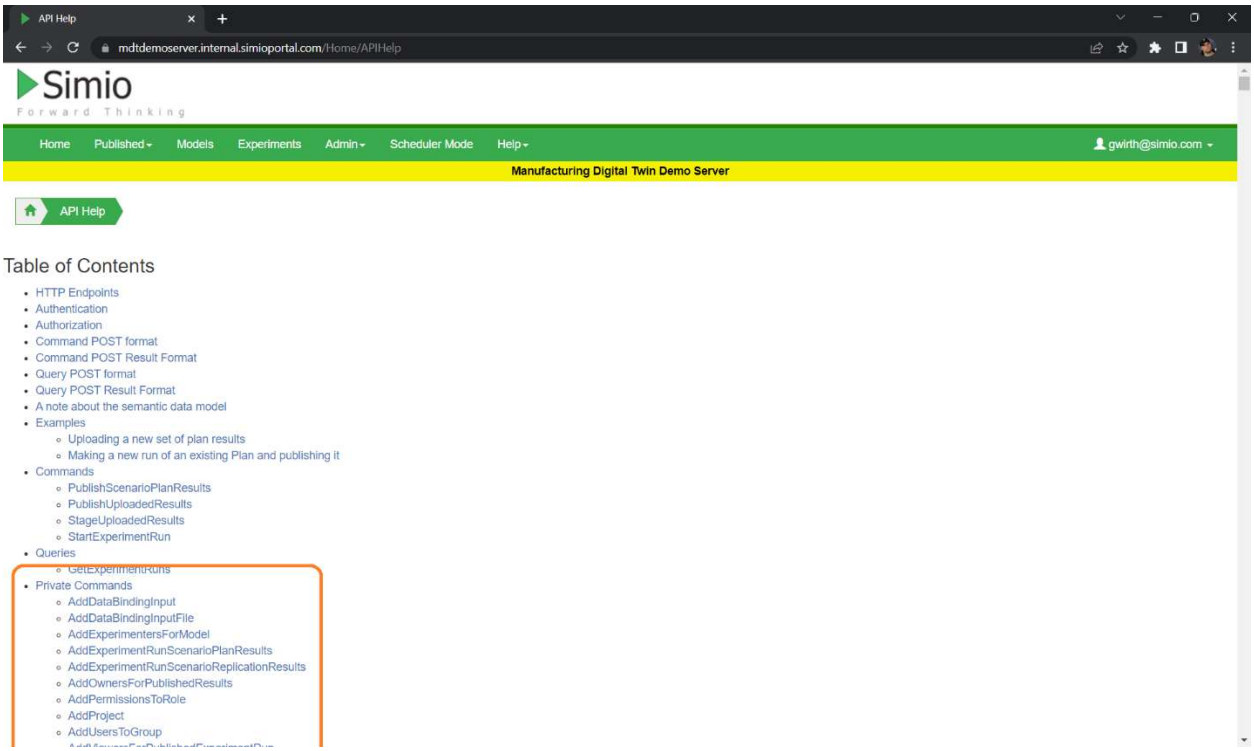
RunSimioPortal -re True -rs False

Here are the results from the run.




## Appendix A: Allow Private API Calls

Some of the methods used in the RunSimioPortalExpConsole use private Simio Web API call. The method that required the private API available are noted in the Settings section of this document. If you plan to use these methods, make sure that the Simio Portal has been set up to expose its private commands and queries. To verify that Simio Portal has been setup with these private methods, navigate to Help…API Help. Make sure the Private Commands and Private Queries are shown on the screen.




If the private methods are not available, add the following key in the appSettings to the Simio Portal Web.config.

<add key="allowPrivateAPICallsKnowingTheAPIMayChangeAndCallersWillThenFail" value="true"/>

Once added, just save the Web.Config and the private Web API calls should be available. See screenshot below for the location where to add this key:

Simio

Forward Thinking