

**Simio Planning and Scheduling**

***Detailed Production Scheduling with Real-Time Data***

**AVEVA MES Integration Guide**





[www.simio.com](http://www.simio.com) Copyright © 2020 Simio LLC. All rights reserved.

Table of Contents

[Overview 3](#_Toc31391832)

[ERP Integration 3](#_Toc31391833)

[Interface Components 5](#_Toc31391834)

[Install Database Object 6](#_Toc31391835)

[New Simio Model Based On Template 7](#_Toc31391836)

[Import Entities 8](#_Toc31391837)

[Import Product Definition and Order Management Tables 12](#_Toc31391838)

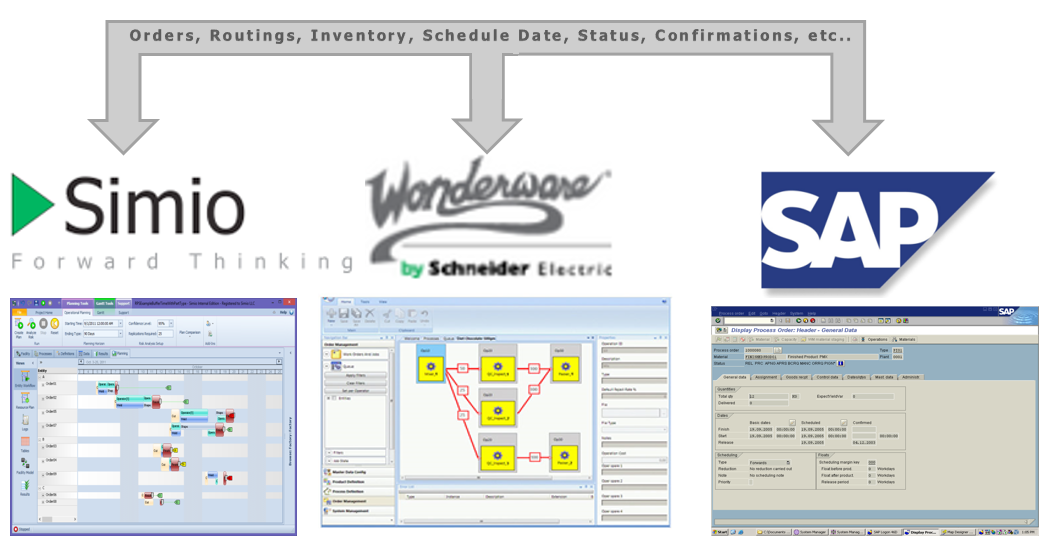
[Generating the Schedule 13](#_Toc31391839)

[Exporting Results 14](#_Toc31391840)

[Import Downtime 16](#_Toc31391841)

[Appendix A – Automatic Reschedule 19](#_Toc31391842)

[Appendix B – Publishing the Schedule using the Simio Portal 24](#_Toc31391843)



# Overview

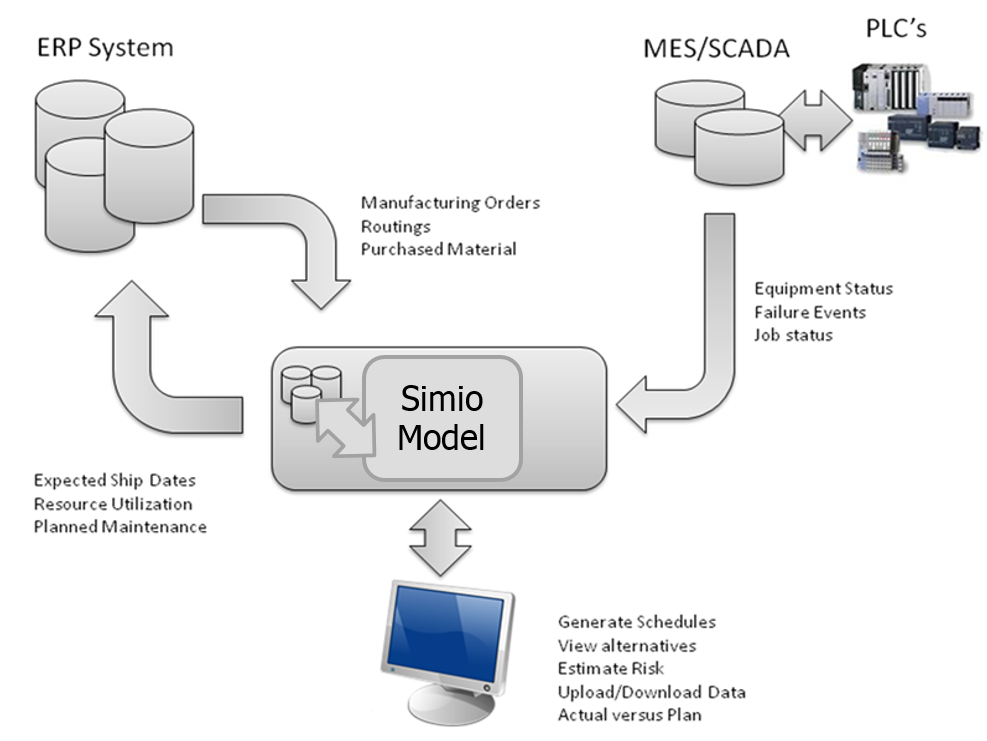
Welcome to the world of simulation. Simulation provides a unique way to examine the future and make intelligent decisions based on what you learn. While simulation technology has been around for decades, it is still rapidly evolving. Advances in object-oriented approaches provide rapid modeling and the flexibility to model complex systems that could not be modeled just a few years ago. And integrated 3D animation makes the creation of compelling 3D visualizations easy, and this in turn helps assure more robust, understandable models and better communication with stakeholders.

Simio extends that base technology to provide a unique solution for planning and scheduling. Simio provides AVEVA MES with Finite Capacity Scheduling and Advanced Planning and Scheduling functionality. Simio’s patented approach to planning and scheduling is first to automate comparing a deterministic schedule from MRP with a variation adjusted schedule to let you effectively deal with breakdowns, unplanned events and material shortages. You can then re-schedule resources to acceptable levels of risk. Data is brought into memory for fast execution. While Simio takes full advantage of your existing data to quickly and automatically build you scheduling model, that base model can also be enhanced as needed to accurately capture complexity unique to your system.

The team of architects behind Simio have been leaders in simulation since the early 1980’s, playing key roles in the design and development of four previous market-leading products. Simio is the result of this team applying their collective 180 years of simulation experience and using the very latest in technology and development techniques to create a new generation of simulation problem solving capability.

# ERP Integration

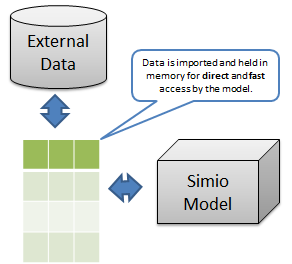
The Simio Production Scheduling model is typically integrated with the existing ERP and MES/SCADA system to provide the data on the current status of the system and the actual jobs to be processed through the system. This data is provided to Simio in the form of relational data sets that are imported and held in memory by Simio for fast execution. These data sets typically contain data such as a list of jobs to be processed, a bill of material for each job, job routings including setup and processing times, purchased material, etc., along with the status of all jobs in the system at the start of the planning period. Simio does not have a fixed data schema – the data sets used by Simio Scheduling Edition are configured as needed in the Simio Enterprise Edition to match the form of the external data. The flow of jobs as defined by the data sets is then simulated by Simio Scheduling Edition using the custom Simio model to generate a detailed schedule. A schematic of a typical implementation is shown below:



Integration between Simio Production Scheduling and AVEVA MES

This interface has been built to integrate data between Simio Production Scheduling and AVEVA MES to enable real-time event based scheduling. Utilizing Simio’s simulation-based scheduling capabilities, a model can be developed to depict the resource and material constraints within a manufacturing system. Then, interfacing Simio to the AVEVA MES enables the scheduling model to be run near real-time. This mean as events happen on the factory floor, the production schedule can be re-run quickly to re-route orders around possible bottlenecks.

The integration employs pulling of data from database views and stored procedures of the MES tables to get data into the model. The AVEVA MES as Web API Data Connector to call the AVEVA Web API to pull this data from the AVEVA MES data model. Inventory, work orders, routings, bill-of-materials, order status and machine status are all captured through this inbound integration to pull the data for creating the production schedule.



Once the schedule is generated, the scheduled is updated in the AVEVA MES using the AVEVA MES API. The Export Schedule to MES also uses a Web API Data Connector and the AVEVA Web API.

# Interface Components

There are a number of components that are used for this integration. These components can be downloaded using the following URL:

[http://www.simio.com/downloads/public/software/AVEVAMES.zip](http://www.simio.com/downloads/public/software/WonderwareMES.zip)

Unzip the file. The contents are as follows:

* AVEVAMESMSSQL2019Backup – A Microsoft SQL Server 2019 database backup is provided to recreate the demo if someone has AVEVA MES 2017 R2.
* DatabaseScripts - SQL views and stored procedures used to pull the MES data.
* DemoModels – There are 2 completed scheduling models. One model shows the integration with AVEVAMES and the other shows the integration with InBatch.
* Documents – This folder contains 2 documents. The first this SimioAPI-UsingTheSimioBackendDll.pdf that shows how to run Simio without the GUI. The MiddlewareExtensibility\_Jan2013.pdf is a reference on how to trigger evens from the WondewareMES Middleware.
* InBatchXMLStylesheets – This folder contains a sample InBatch recipe in BatchML format as well as 4 XML Stylesheets used to transform the BatchML into the format needed to import the data into the Simio tables.
* Node-REDFlows – This folder contains the flow used to automatically regenerate the schedule based on MQTT Topic and Simio Portal Web API.
* PublishMQTT – The folder contains the visual studio project that calls an MQTT topic called “mes/downtimeEvent” when configured to run via the AVEVAMES middleware. This is to be used with the Node-RedFlows.
* PublishMQTTTest – Test Client to run PublishMQTT
* RunSimioPortalExperiment – The folder contains the RunSimioPortalExperiment executable and supporting files to run Simio Portal Experiment using a Windows Service.
* RunSimioPortalExperimentCode – The folder contains the visual studio project used to generated the RunSimioPortalExperiment executable.
* TestHooks – The folder contains the visual studio project that produces a file when configured to run via the AVEVAMES middleware. This is to be used with the RunSimioPortalExperiment.

The DatabaseScript need to be run on the AVEVAMES database to setup the integration. The other folders are optional depending on the project. If needed, they will be referenced in other sections of this document.



# Install Database Objects

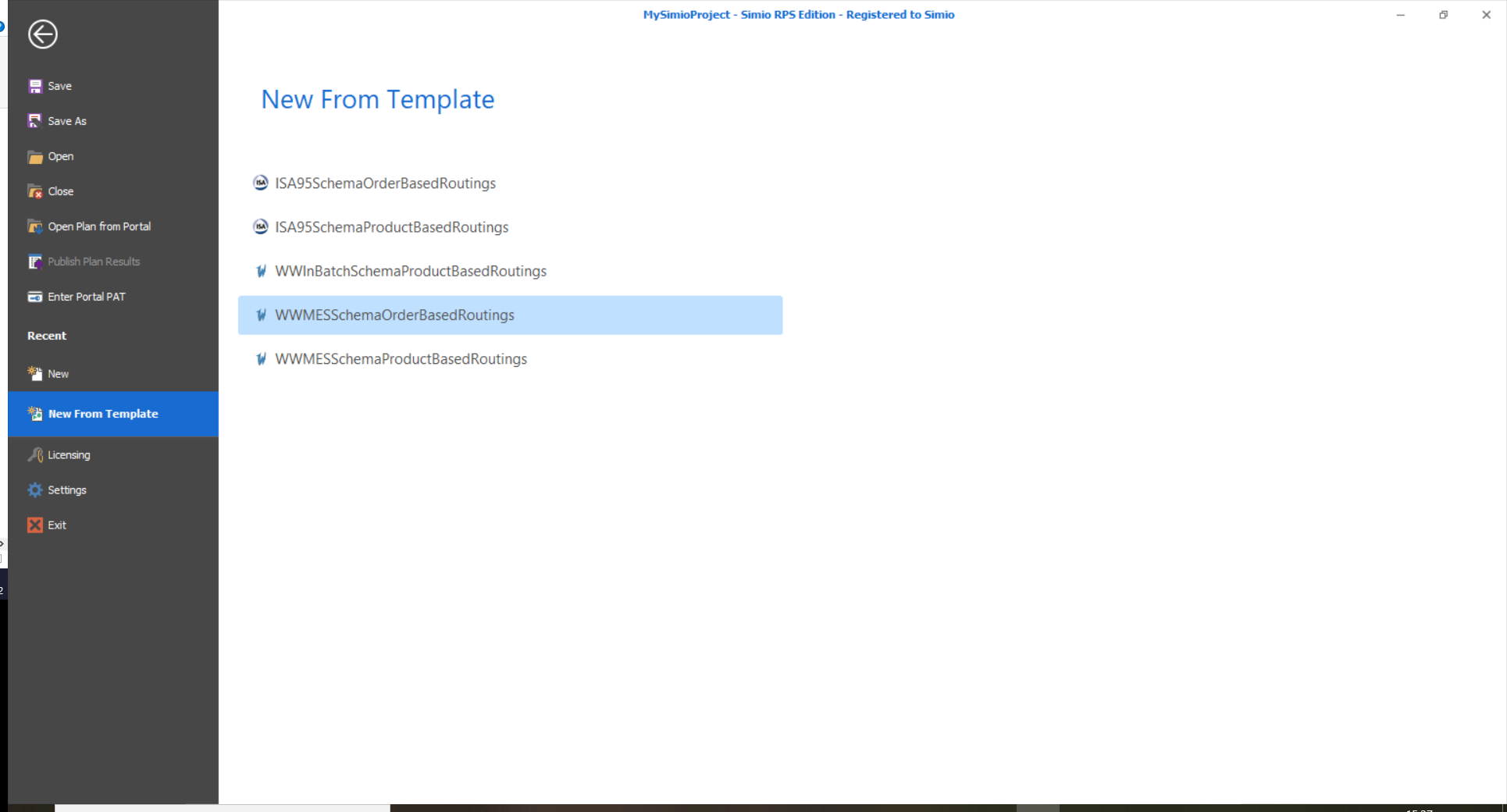
The database views and stored procedures are used to map the data between the AVEVA MES database and the Simio table. A database objects are in the SQL script provided. The scripts are to be run on the same database used by AVEVA MES.

Install the following scripts:

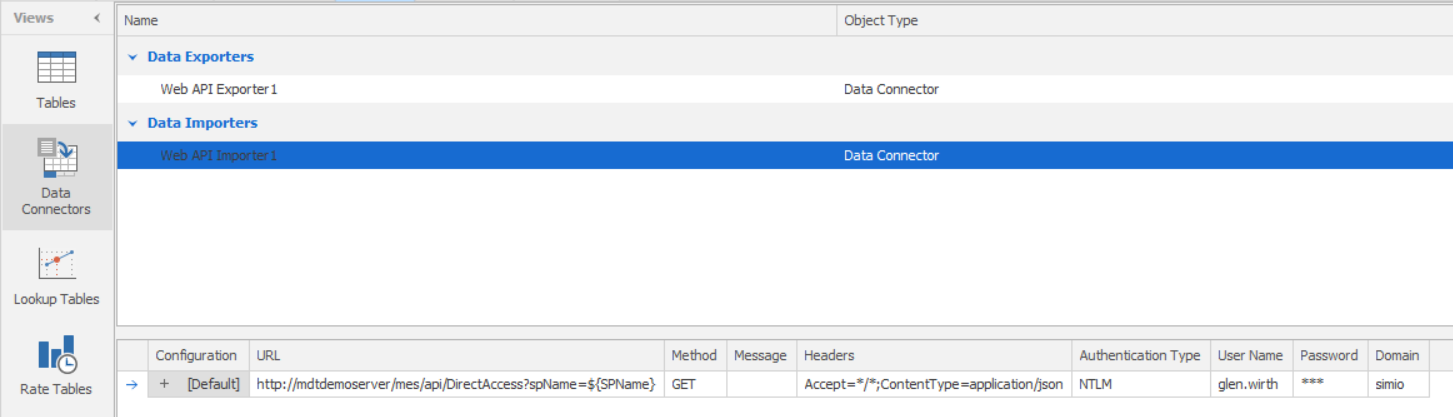
* DatabaseViews.sql – This script contains all the database views used in this integration.
* DatabaseStoredProcedures.sql - This script contains all the stored procedures used in this integration.

# New Simio Model Based On Template

Open Simio. The Select File…New From Template. Then select “WWMES Schema Order Based Routings”. This will create a new model that has a data schema, sub-classed object definitions, dashboards and reports defined. The new model can then easily be integrated with a existing AVEVAMES system that has order based routings.

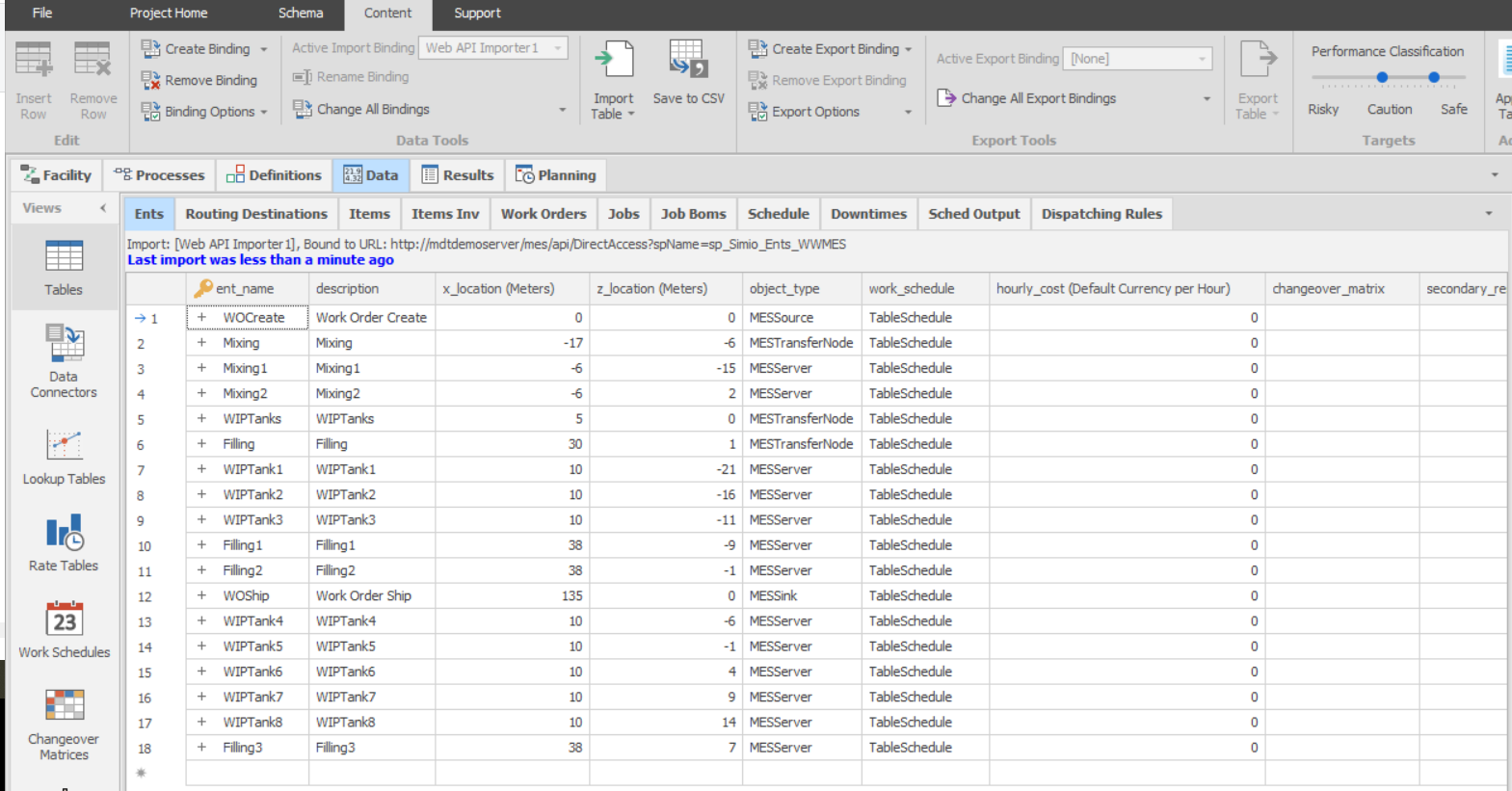


Next, select the Data tab and the Data Connectors. Configure URL, UserName, Password and Domain.



# Import Entities

In this step, we will import the entities (referred to as resources in Simio) from AVEVA MES into Simo. Select the ‘Data’ tab and then select the ‘Ents’ table. Then, select ‘Import Table’.



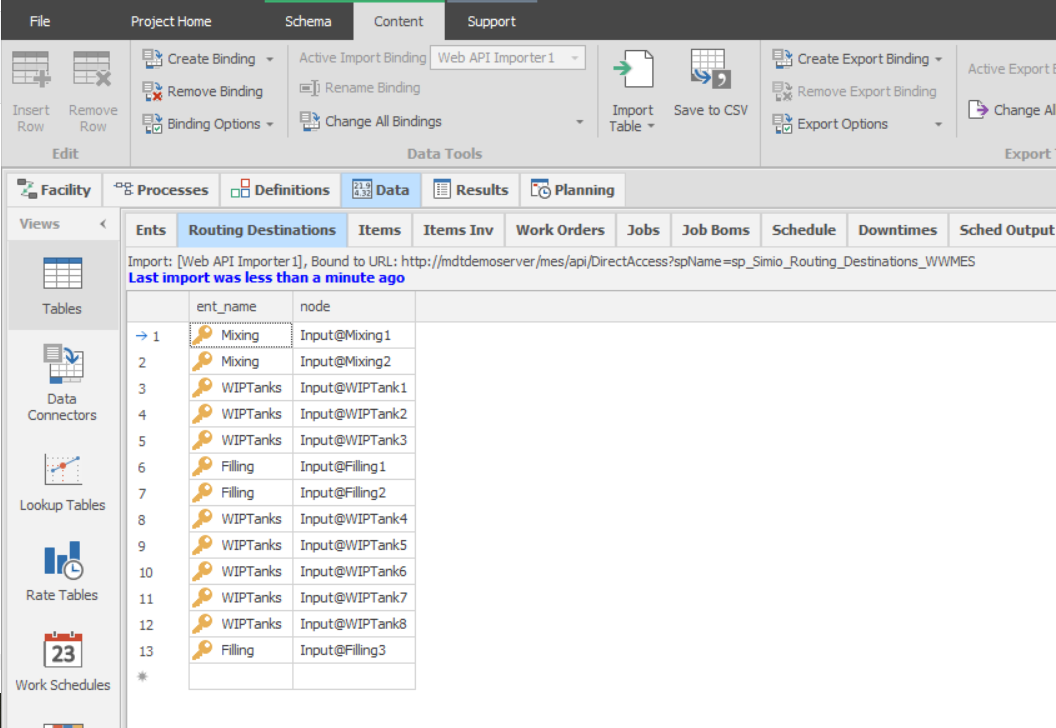
As the entities are added to the table, they are automatically added to the model. Select the ‘Facility’ tab. The entities are positioned at the location defined in the Ents table. If you choose to move the entities in the facility, the position will be updated in the Ents table.



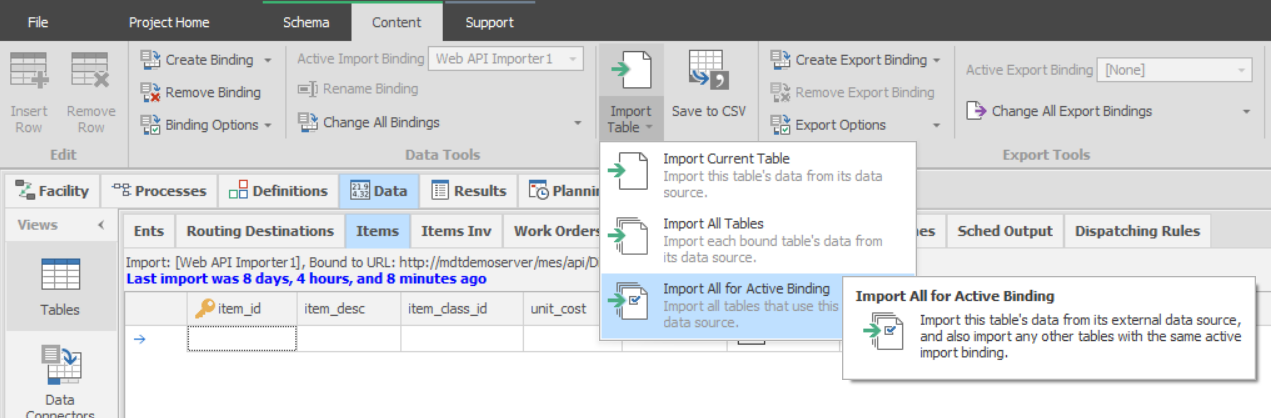
Your Facility view is ready to use. It is good to save your model at this time. You can choose File…Save As to choose a different name for the model.

# Import Product Definition and Order Management Tables

Follow the same database binding steps described in the Import Entities section to import the rest of the data. On each table that has an import binding, press the to import the data into the model.

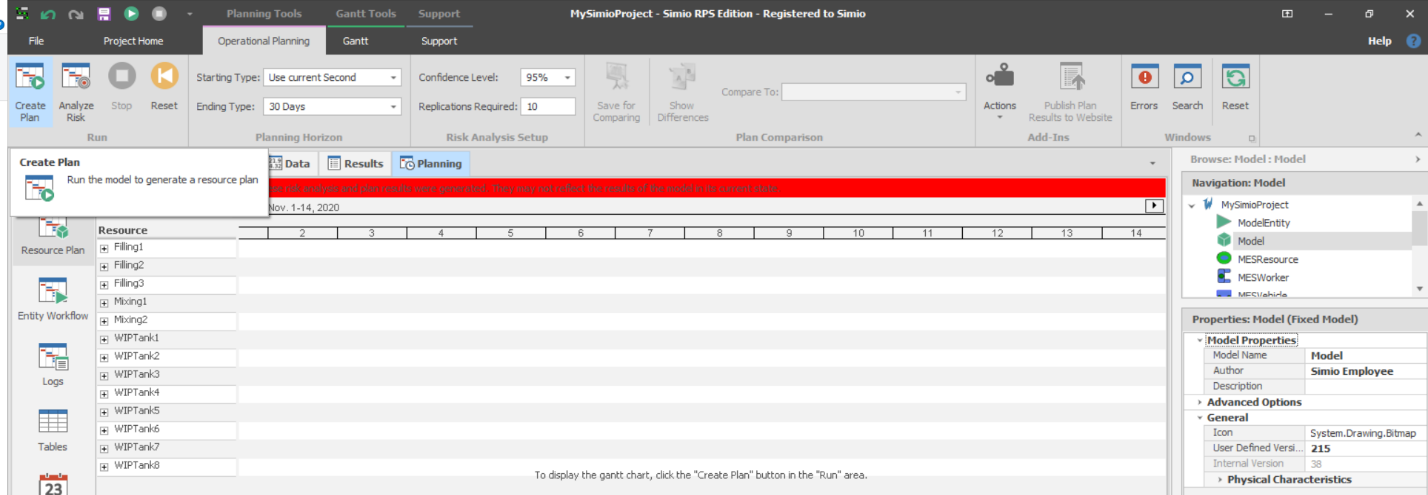


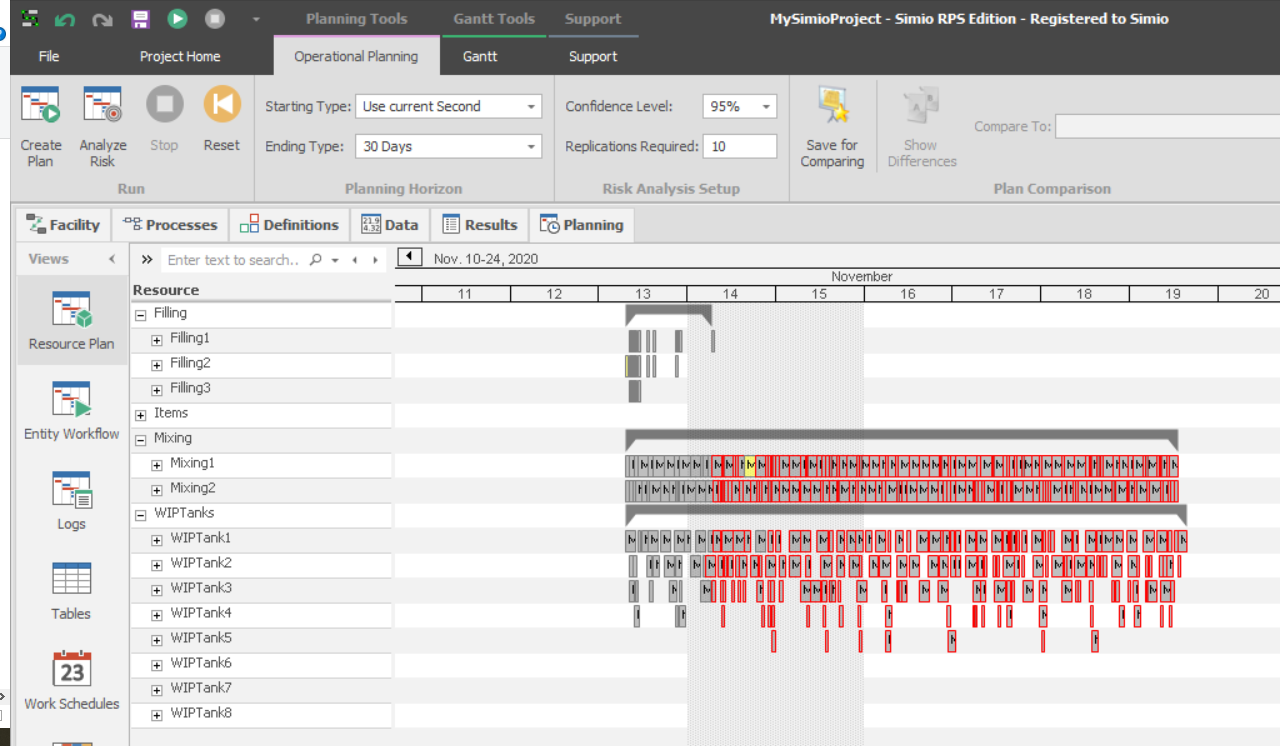
You can also press ‘Import All for Active Binding’ to import all the data into the model with a single button press.



# Generating the Schedule

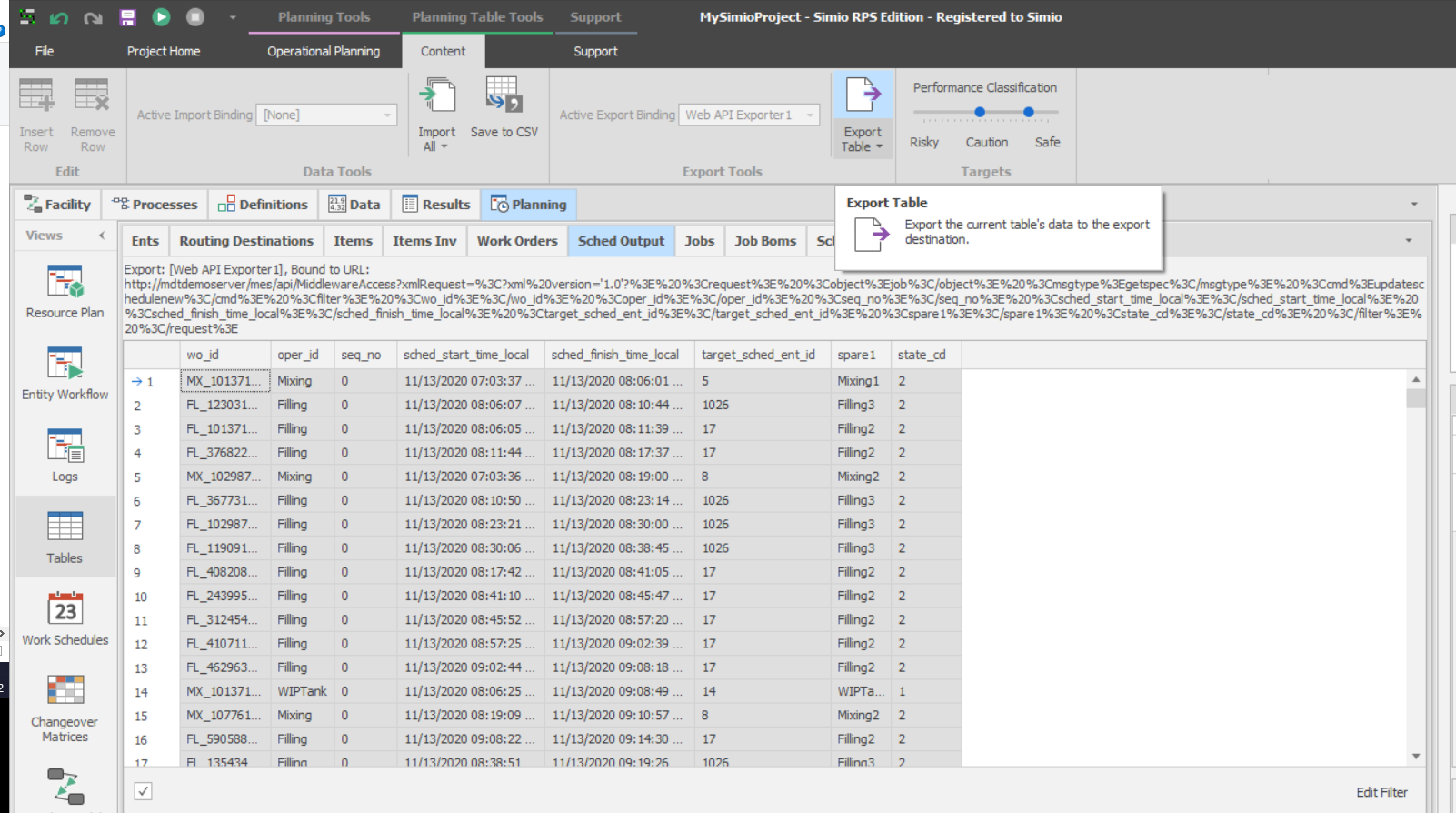
Select the ‘Planning’ tab. Once selected, the ‘Resource Plan” should be populated with the AVEVA MES Entities. From here, the schedule can be generated. By default, the work orders will be scheduled based on priority. To create the schedule, select “Create Plan”.



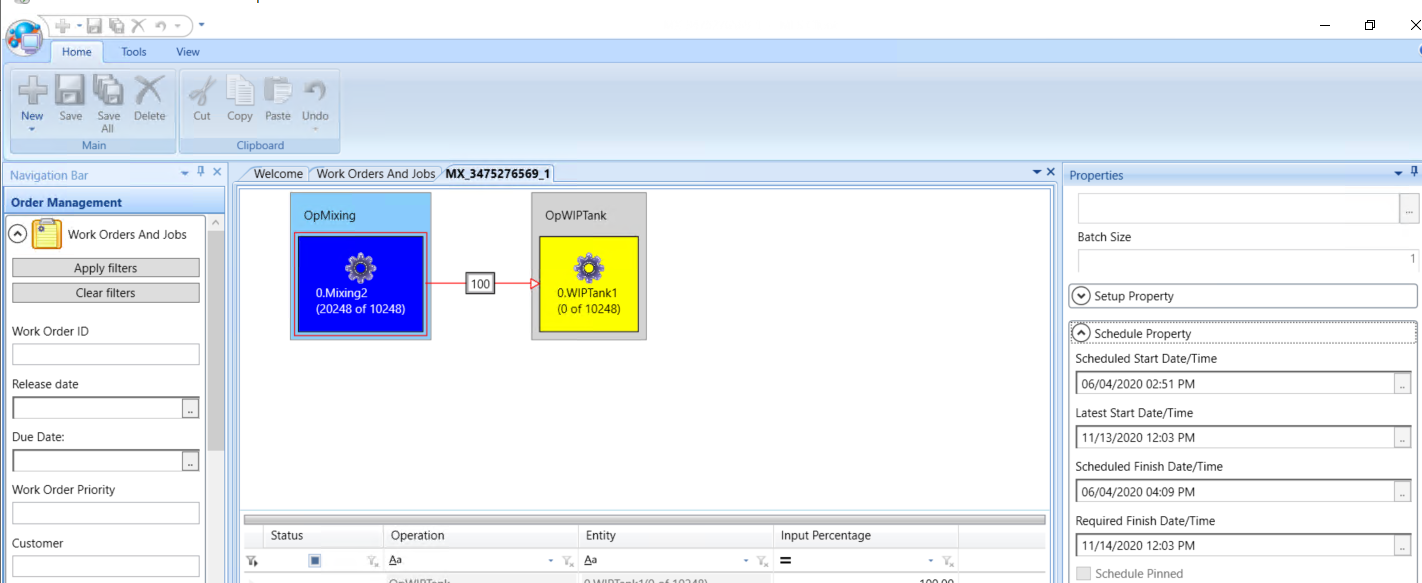
This will create the schedule. From the Gantt tab, select “All” to see the entire schedule. 

# Exporting Results

By default, the template is setup to automatically export the schedule back to AVEVA MES at the end of RunPlan. You can also do it manually by selecting the Tables from the Planning tab. Then, select the SchedOutput table and press ‘Export Table’.

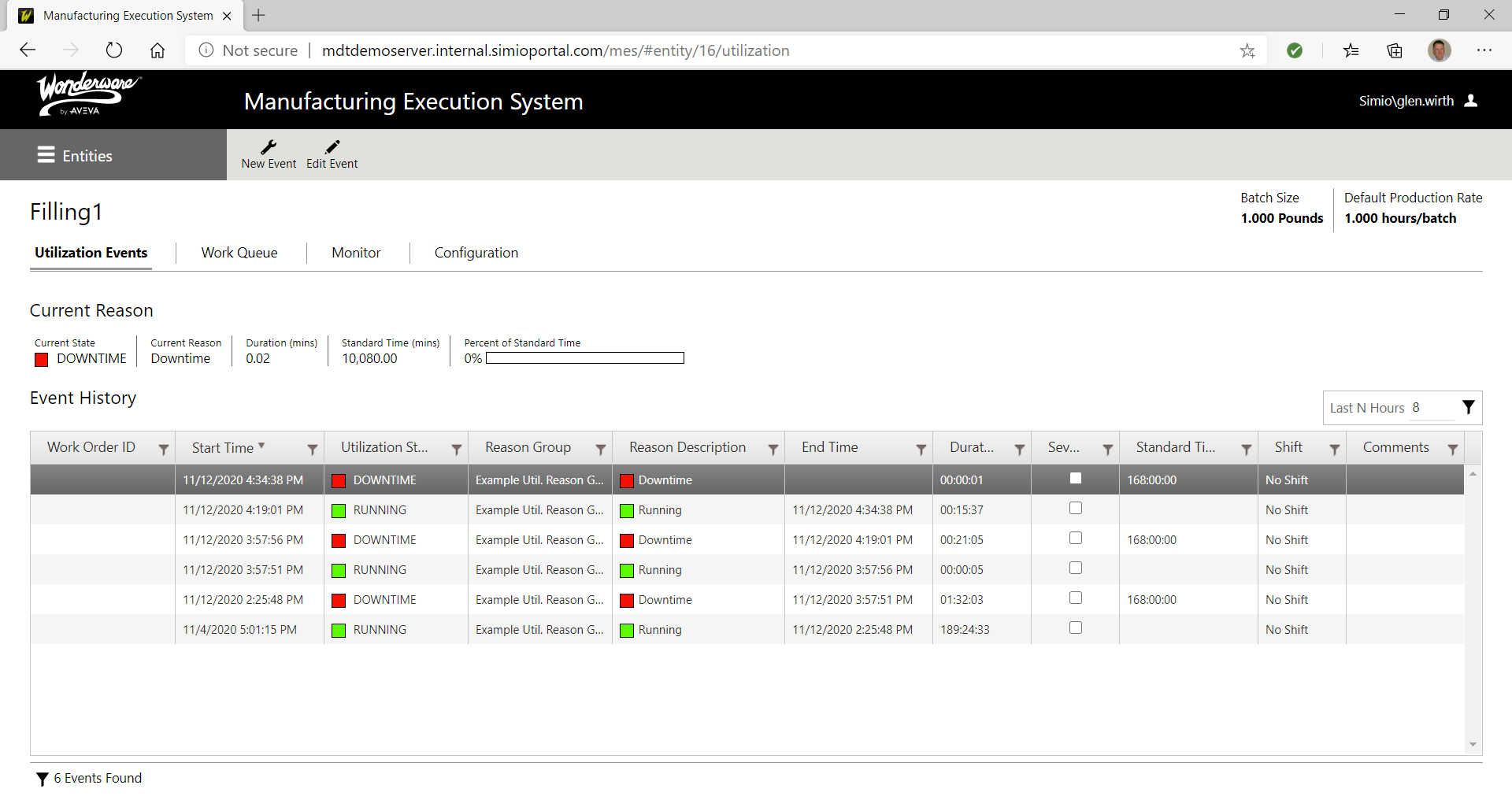


In AVEVA MES, you will see that the jobs have been updated with the Simio scheduling information.

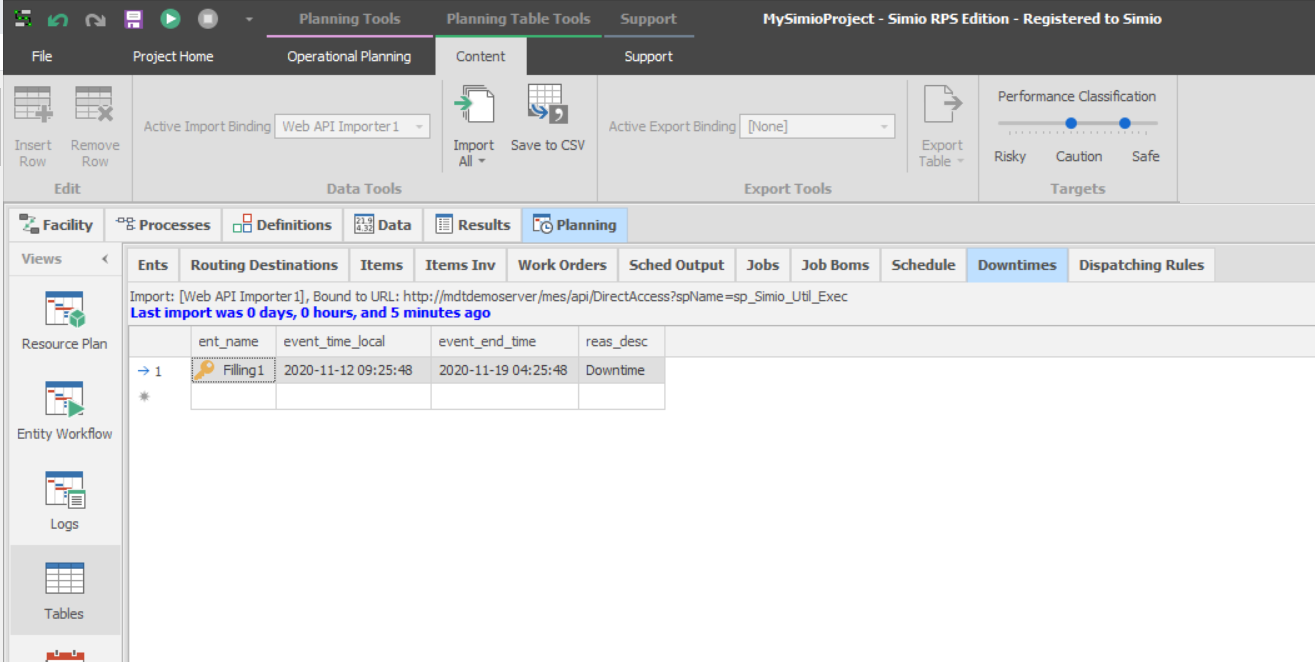


# Import Downtime

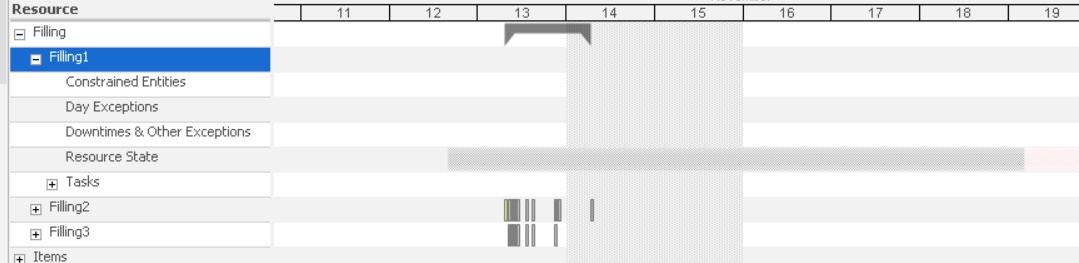
Add downtime on an entity in AVEVA MES Portal



Within Simio, select the Tables from the Planning tab. Then, select the Downtimes table and press ‘Import Table’.



Re-plan around downtime. You will see that jobs have been shifted from Filler1 to the other mixer resources.



# Appendix A – Automatic Reschedule Using TestHooks and RunSimioPortalExperiment

To automate reschedule, the AVEVA MES Middleware Extensibility Hooks and the Simio Portal Web API are used. The AVEVA MES Middleware Extensibility Hooks are used to capture the events in AVEVA MES. Once the events are captured, the RunSimioPortalExperiment service will see the event and automatically re-run the schedule using the Simio Portal Web API. After the scheduling run, the schedule will automatically be sent back to MES and published on Simio Portal.

To setup the AVEVA MES Middleware Extensibility Hooks, the ‘Custom Assembly That Is Not in the GAC’ example that start on page 26 of the MiddlewareExtensibility\_Jan2013.pdf guide was referenced. The example was modified slightly to change the name of the output file and to capture any exceptions. The method used is :

public void TestHooksMethod(string xmlSource)

{

// Compose a string that consists of three lines.

string lines = string.Format("DateTime: {0}, XMLSource: {1}", DateTime.Now.ToString(), xmlSource);

// Write the string to a file.

try

{

using (System.IO.StreamWriter file = new System.IO.StreamWriter("C:\\Temp\\Event.txt", true))

{

file.WriteLine(lines);

}

}

// Catch Exception

catch (Exception ex)

{

using (System.IO.StreamWriter file = new System.IO.StreamWriter("c:\\Temp\\Error.txt", true))

{

file.WriteLine(ex.Message);

}

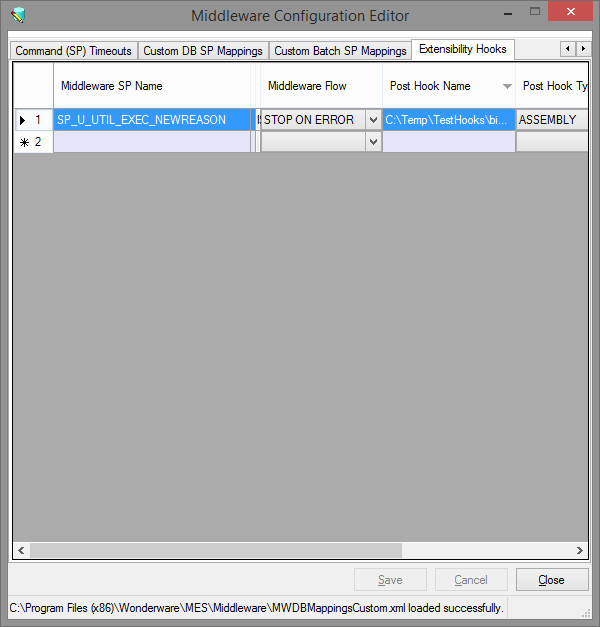
}

}

This assembly was compiled using the same structure defined in the ‘Custom Assembly That Is Not in the GAC’ example.

C:\Temp\TestHooks\bin\Debug\TestHooks.dll;TestHooksNamespace.TestHooksClass;TestHooksMethod

Once Within the Middleware Configuration Editor, a post hook is defined to call the assembly every time a new downtime is entered or updated ‘SP\_U\_UTIL\_EXEC\_NEWREASON’ . When the new downtime is entered / updated, the Event.txt is added / updated.

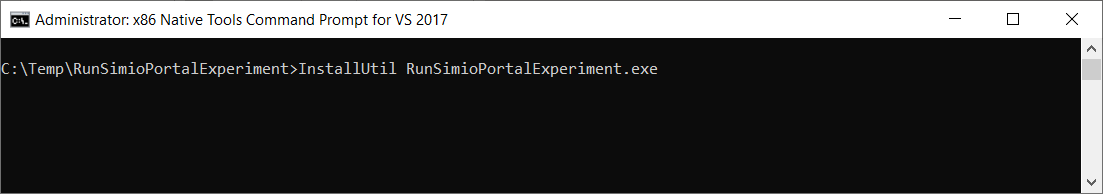


Once configured, the middleware server needs to be restarted before this event will be captured.

Copy the RunSimioPortalExperiment folder from the AVEVAMES.zip into the ‘C:\Temp’ folder. Modify the RunSimioPortalExperiment.exe.config.

Next, register the RunSimioPortalExperiment.exe as a windows service. Open a command prompt as an administrator (e.g. right click and select ‘Run as Administrator’).

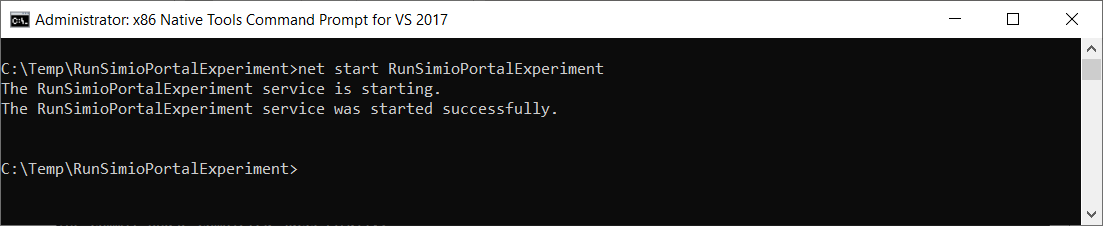
Navigate to ‘C:\Temp\RunSimioPortalExperiment’ From the command prompt, enter command ‘InstallUtil RunSimioPortalExperiment.exe’



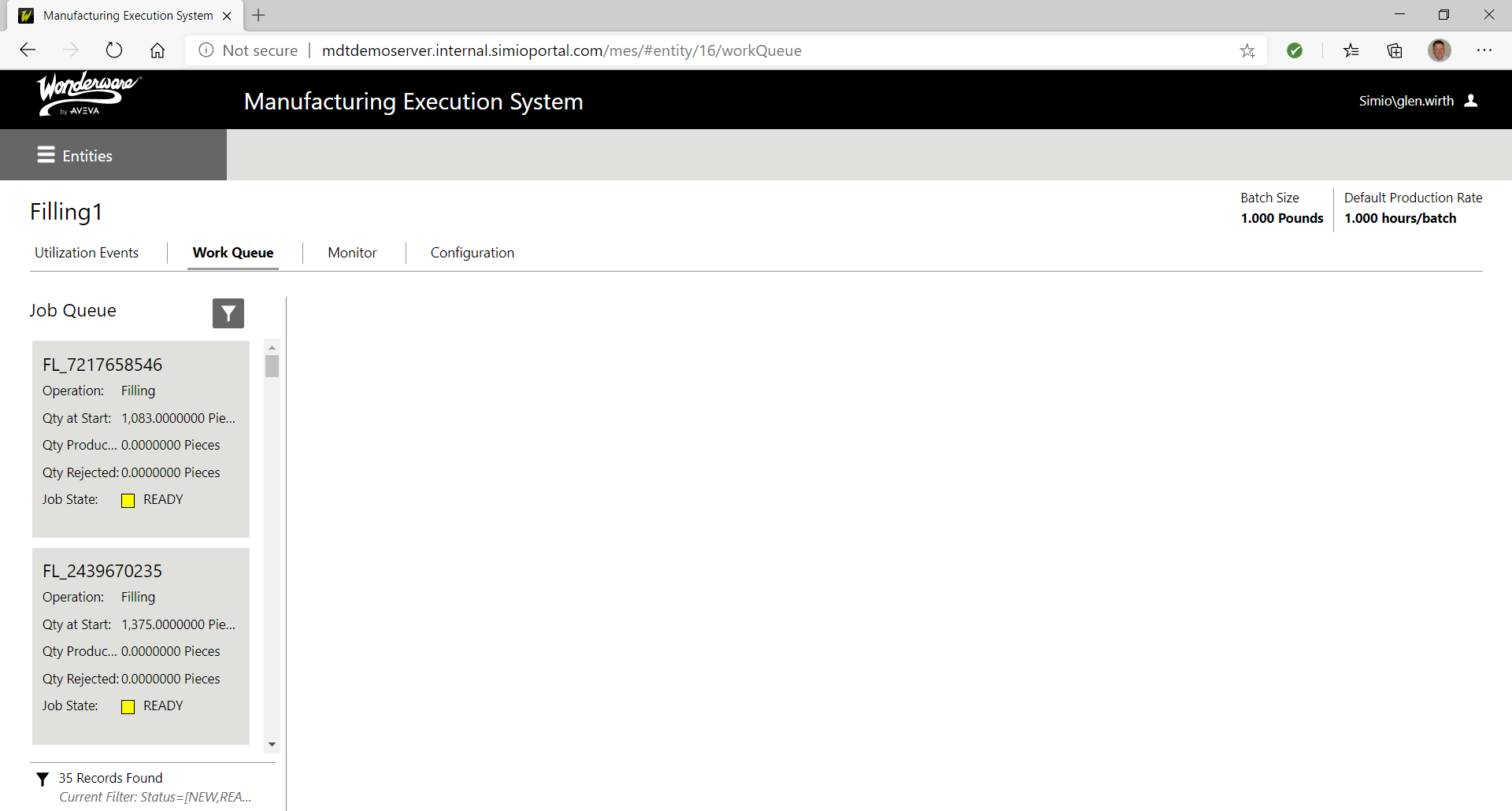
This will install the RunSimioPortalExperiment as a windows service.



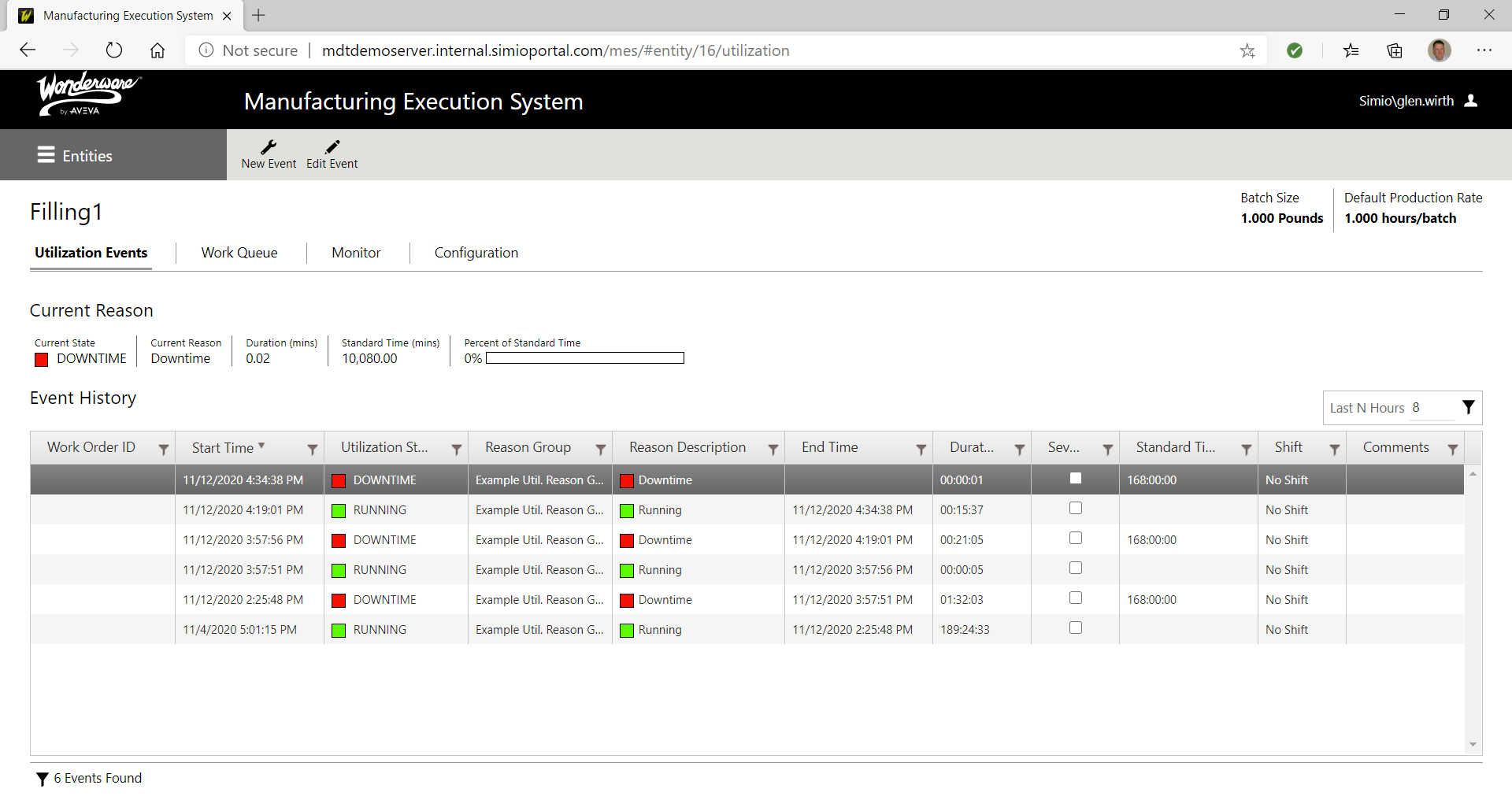
Next, start the service. Enter ‘net start RunSimioPortalExperiment’ to start the service.



To test the automatic generation of the schedule, check the work queue to verify that the current dispatch list on an entity.



Enter a downtime on the entity….



Verify that the schedule has changed.



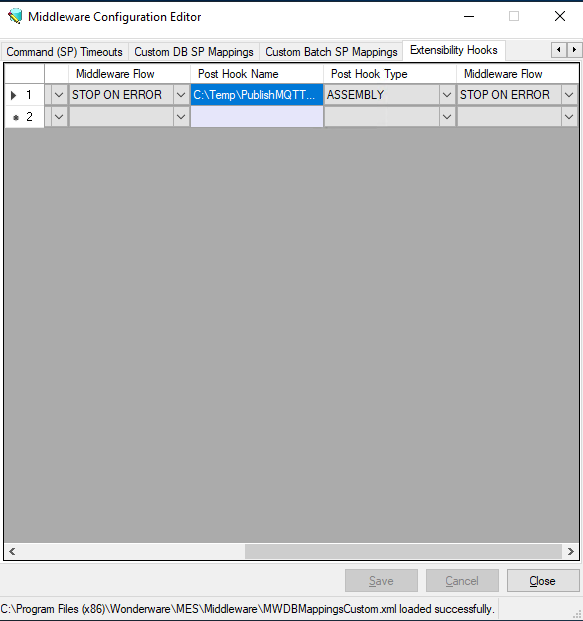
# Appendix B – Automatic Reschedule Using PublishMQTT and Node-RED flow.

To automate reschedule, the AVEVA MES Middleware Extensibility Hooks and the Simio Portal Web API are used. The AVEVA MES Middleware Extensibility Hooks are used to capture the events in AVEVA MES. Once the events are captured, the Node-Red flow will see the event and automatically re-run the schedule using the Simio Portal Web API. After the scheduling run, the schedule will automatically be sent back to MES and the schedule will be published on Simio Portal.

Instead of the TestHooks (previous appendix), we are going to use PublishMQTT instead.

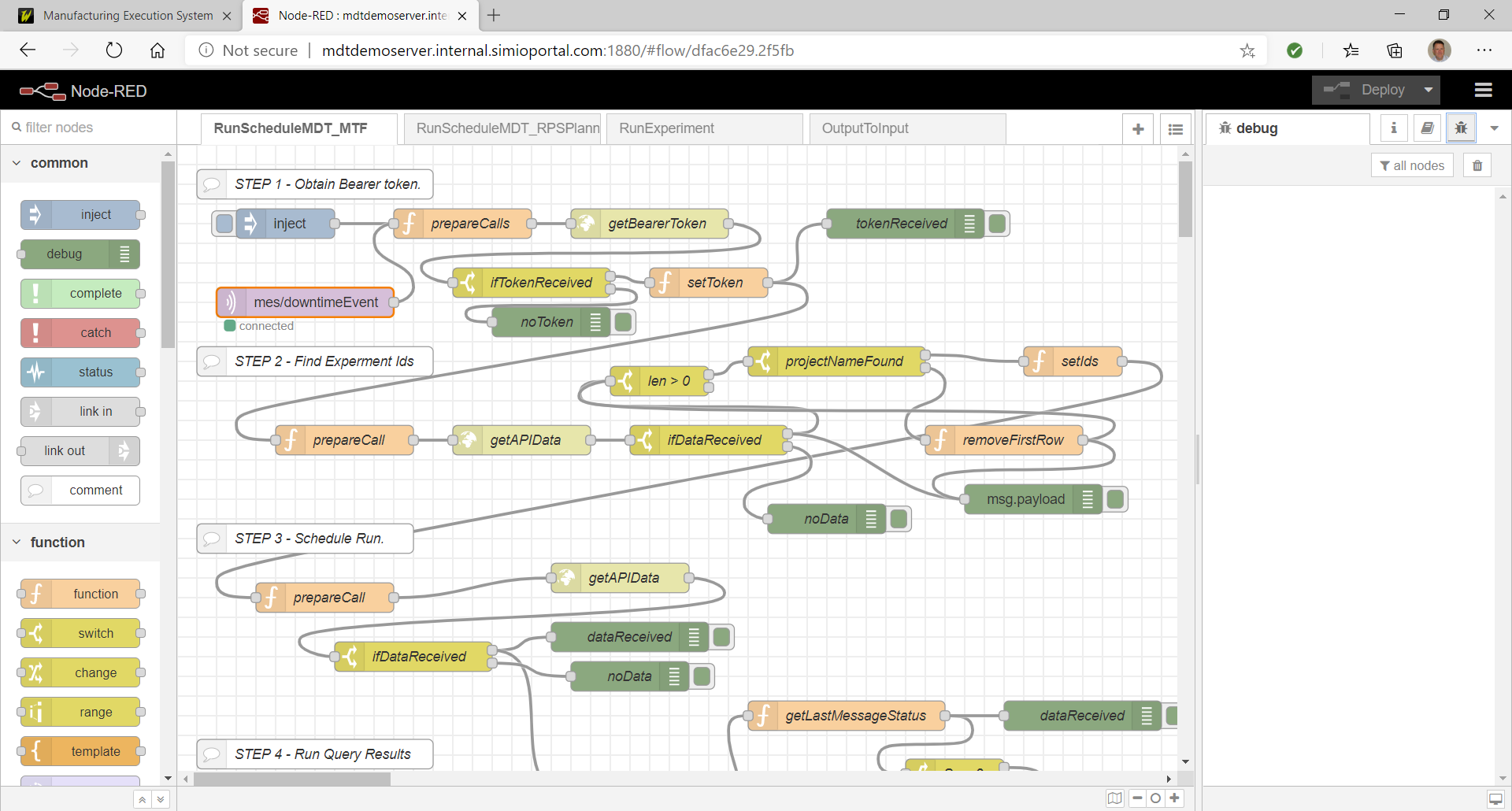
C:\Temp\PublishMQTT\bin\Debug\PublishMQTT.dll;PublishMQTTNamespace.PublishMQTTClass;PublishMQTTMethod

Once Within the Middleware Configuration Editor, PublishMQTT event is defined to call the assembly every time a new downtime is entered or updated ‘SP\_U\_UTIL\_EXEC\_NEWREASON’ . When the new downtime is entered / updated, the MQTT Topic ‘mes/downtimeEvent’ is notified.

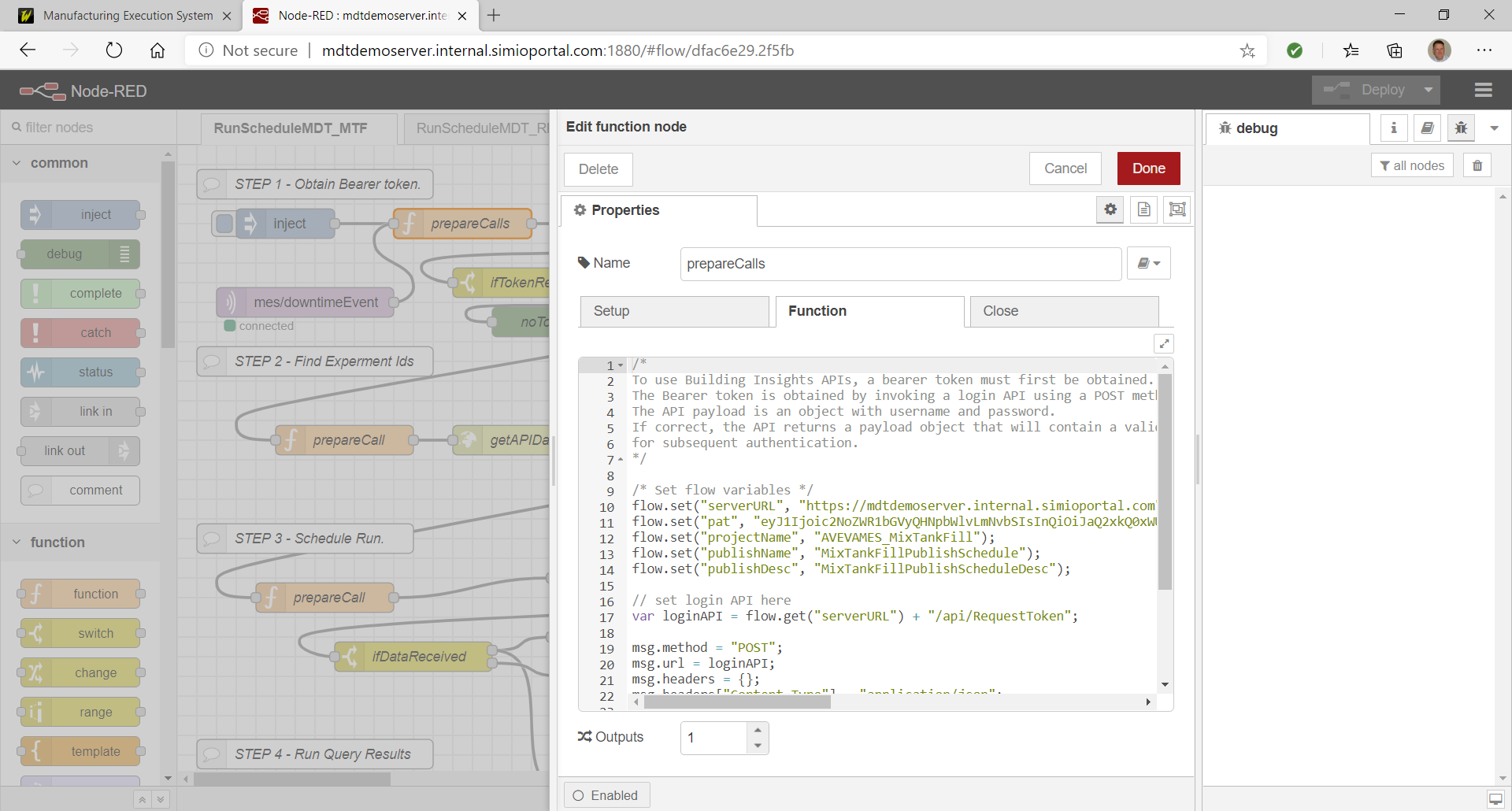


Once configured, the middleware server needs to be restarted before this event will be captured.

Then a Node-RED flow is deployed to capture the event from MQTT. Once captured, the schedule is rerun and published on Simio portal.



The first prepareCall function is used to configure the running and publishing of the schedule.

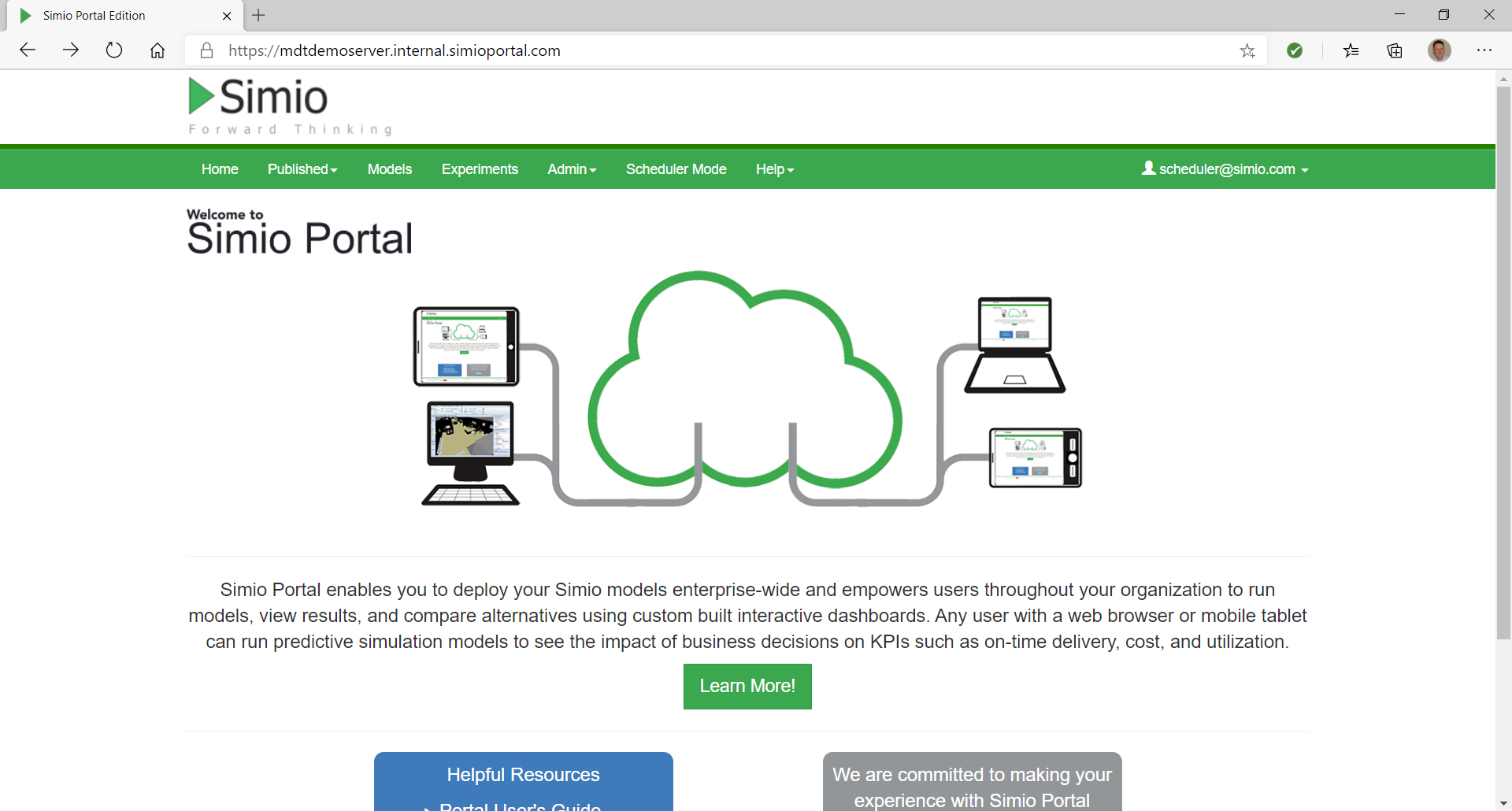


Once configured and deployed, the downtime works the same as in Appendix A.

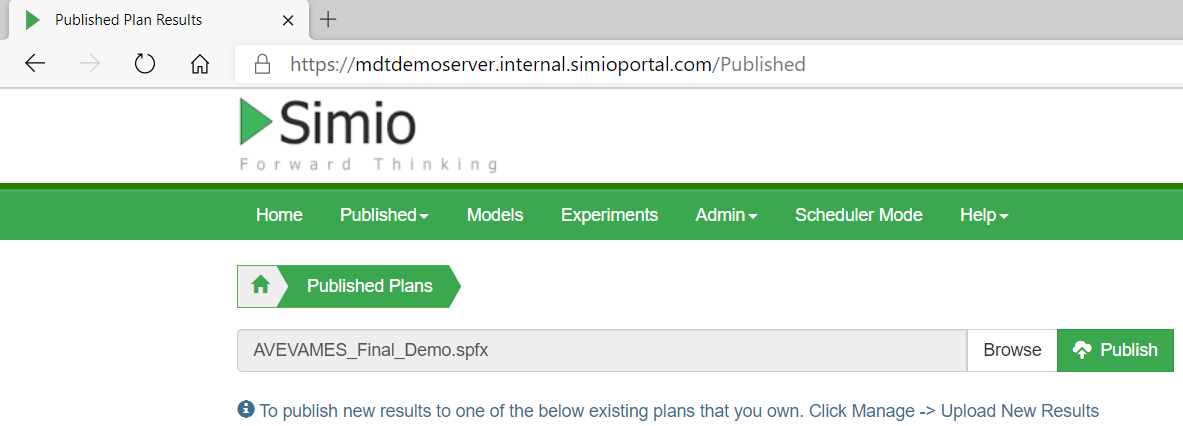
# Appendix C – Publishing the Schedule using the Simio Portal

An alternative method for publishing the Simio schedule uses the Simio Portal. The Simio Portal is a web based model repository used to publish the production schedule to be viewed throughout the organization. The Simio Portal can be hosted within an enterprise or hosted on Microsoft Azure.

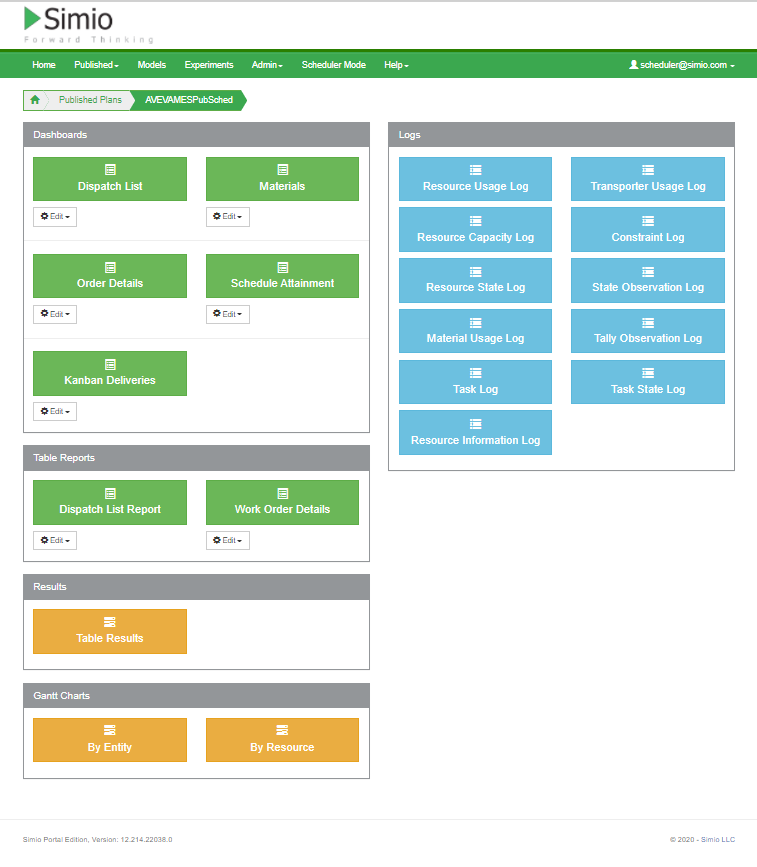
In this example, we will publish the Simio schedule to Microsoft Azure. Once the user logs into the Simio portal, they will be provided a number of capabilities based on their permissions. This user has been given full access in the portal.



The user selects Published to upload a model.



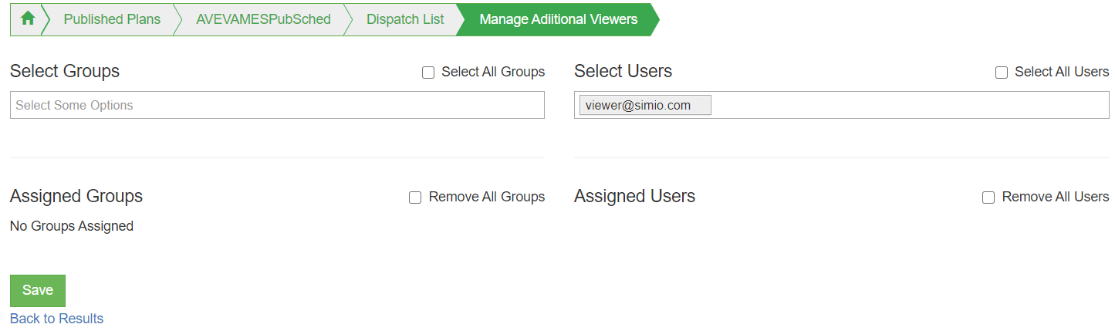
Once the model is uploaded to the Portal, the user will have the option to display Dashboards, Table Reports, Results, Gantt Charts and Logs.



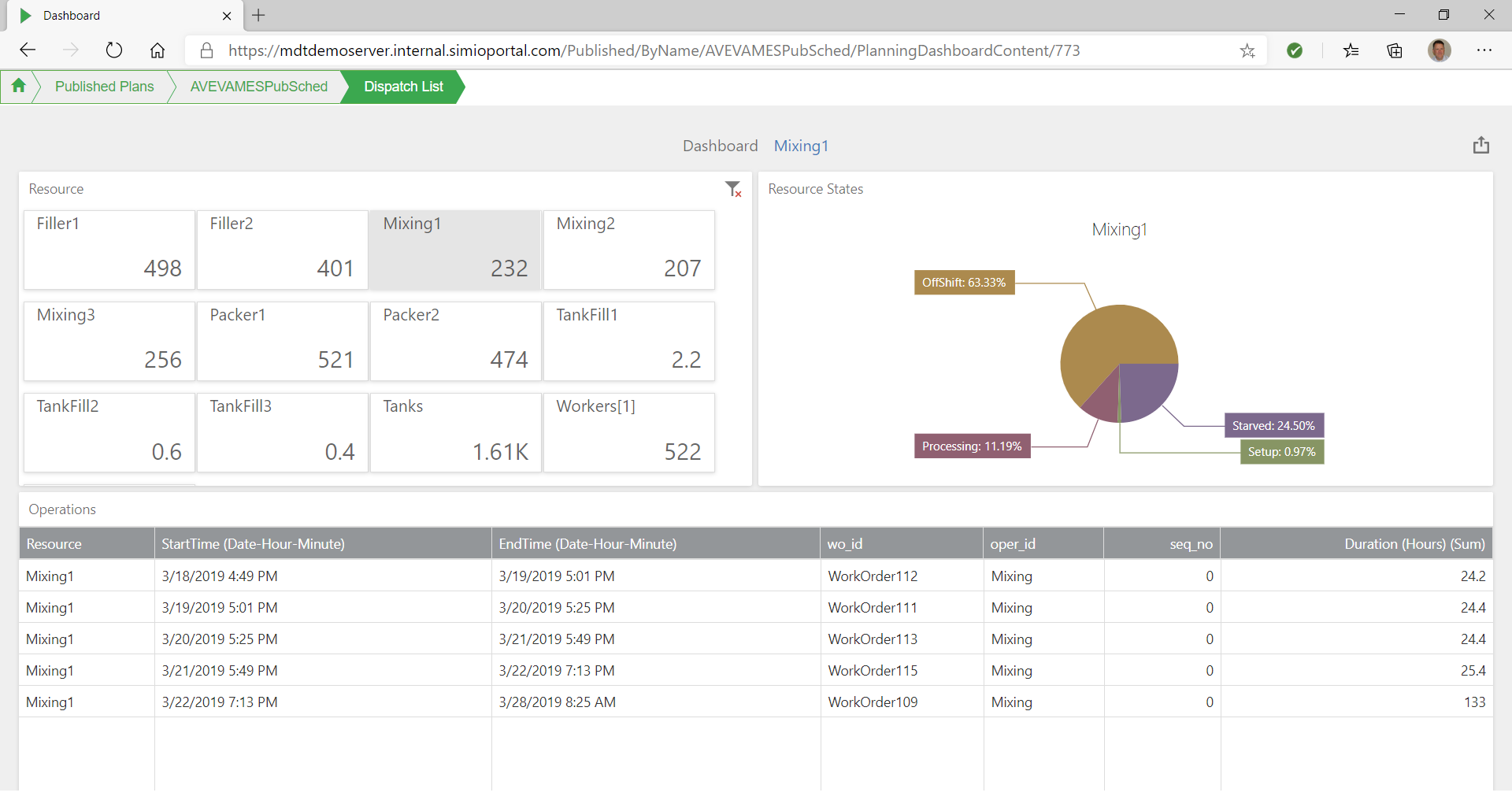
Under Gantt Charts, select ‘By Resource’. This will display the Resource Gantt chart



Dashboards and Reports can be configured so only certain users can view them. When the [viewer@simio.com](mailto:viewer@simio.com) user access the schedule, they only view the Dispatch List



Here is an example of the Dispatch List Dashboard provided to the [viewer@simio.com](mailto:viewer@simio.com) user.



**The Unique Solution from Simio LLC**

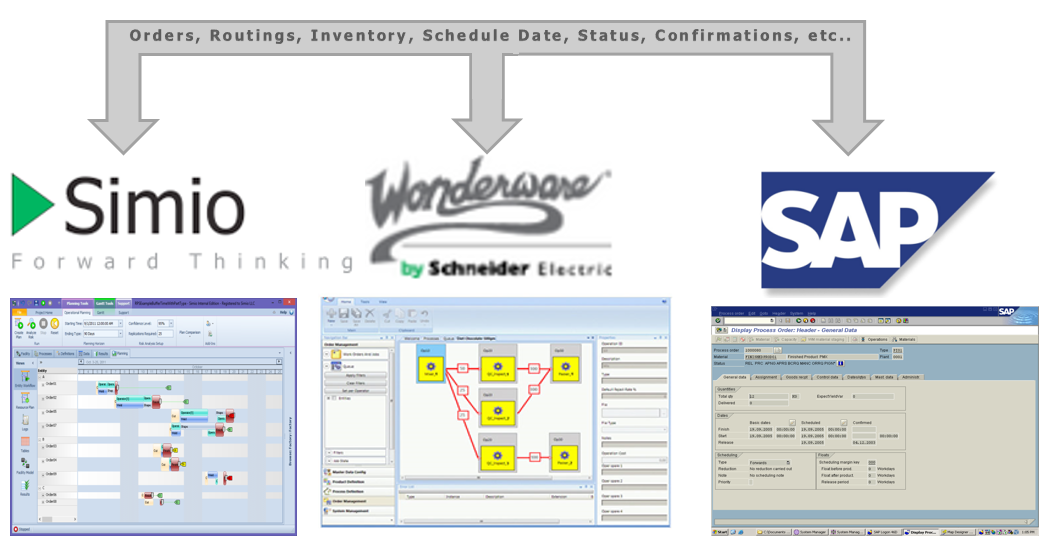
Interface AVEVA to planning and scheduling

Achieve fast deployment

Effectively deal with system variability

Accurately capture system complexity

Provide near instantaneous schedule updates



**For more information:**

Simio LLC

504 Beaver Street, Sewickley, PA, 16033

+1(412)-528-1576. sales@simio.com