

Prescription Mashup: Setup and Use

1 Introduction

The Prescription Mashup is designed to demonstrate making a Prescription (i.e. optimizing a result) using Thingworx Analytics.

The mashup uses a simple dataset showing different inputs to concrete (e.g. water, coarse aggregate, fine aggregate, age etc) and the resulting compressive strength. A prediction model predicts compressive strength based on different inputs entered. Furthermore an optimized score is calculated based on adjusting some values. The prescribed adjustments are also shown.

The dataset was taken from this site:

<https://archive.ics.uci.edu/ml/datasets/Concrete+Compressive+Strength>

2 Thingworx Version

The entities were created and tested with TWX version 8.3

3 Resources

The following files are needed for this scenario:

Entities.twx	<i>Allows import of Thingworx entities</i>
concrete.json	<i>Needed to create an Analytics Dataset</i>
concrete.csv	<i>Needed to create an Analytics Dataset</i>

Optional files:

scriptSourceCode.zip *Source code for services in concreteHelper. Note that this file is only included for convenience as the source code is contained in Entities.twx anyway*

4 Import Entities

In Thingworx Composer, import ... From File ... Type=Entity, Source=Single File, File Name= Entities.twx (or Entities.xml)

5 Create Dataset

Open Analytics Builder ... Data ... New:

Dataset Name = concrete

File Containing Dataset Field Configuration (JSON format) = concrete.json

File Containing Dataset Data (CSV format) = concrete.csv

Dataset has Header = True

Submit

6 Create Model

Open Analytics Builder ... Models ... New:

Model Name = concrete1

Dataset = concrete

Goal = CompressiveStrength

Filter = all_data

Excluded Fields from Model = none

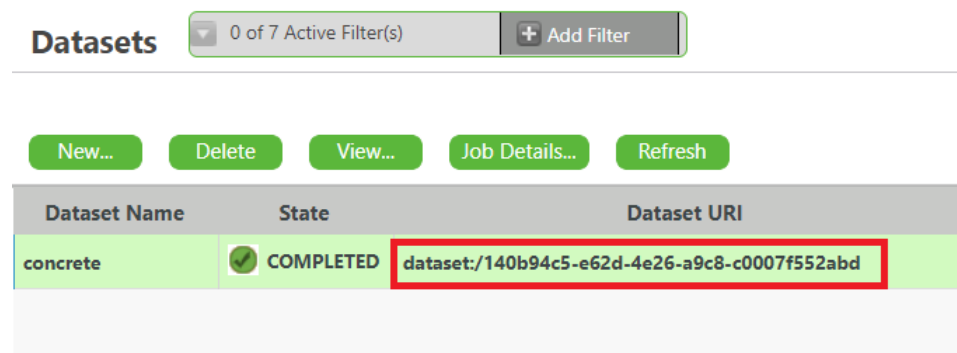
Advanced Model Configuration tab = leave default settings

Submit

7 Set Dataset URI

Open Analytics Builder ... Data ...

Copy Dataset URI to the clipboard as shown below:



The screenshot shows the 'Datasets' section of an analytics tool. At the top, there are buttons for 'New...', 'Delete', 'View...', 'Job Details...', and 'Refresh'. Below these is a table with the following data:

Dataset Name	State	Dataset URI
concrete	COMPLETED	dataset:/140b94c5-e62d-4e26-a9c8-c0007f552abd

Note that you can copy the dataset URI from Chrome DevTools using right-mouse-click ... Inspect ... Copy ... Copy Element and then locating the dataset URI within the HTML code.

Dataset Name	State	Dataset URI	Number of Rows	Number of Columns
concrete	COMPLETED	dataset:/140b94c5-e62d-4e26-a9c8-c0007f552abd	1020	0

Back	Alt+Left Arrow
Forward	Alt+Right Arrow
Reload	Ctrl+R
Save as...	Ctrl+S
Print...	Ctrl+P
Cast...	
Translate to English	
View page source	Ctrl+U
View frame source	
Reload frame	
Inspect	Ctrl+Shift+I

This saves having to type out the dataset URI manually. Probably other browsers have a similar option.

Navigate to Thing: ConcreteHelper and set datasetURI value to the value in your clipboard as shown below:

Thing: ConcreteHelper

Save Cancel More

General Information Properties and Alerts Services Events Subscriptions Permissions Change History View Relationships

Properties Alerts

My Properties Add Duplicate Delete Manage Bindings Refresh

Name	Actions	Source	Default Value	Value	Alerts	Category	Add
- datasetURI				dataset:/33e6f688-3eab-4a10...	+	0	
- modelID				3818935e-d78c-421e-bb9c-4...	+	0	

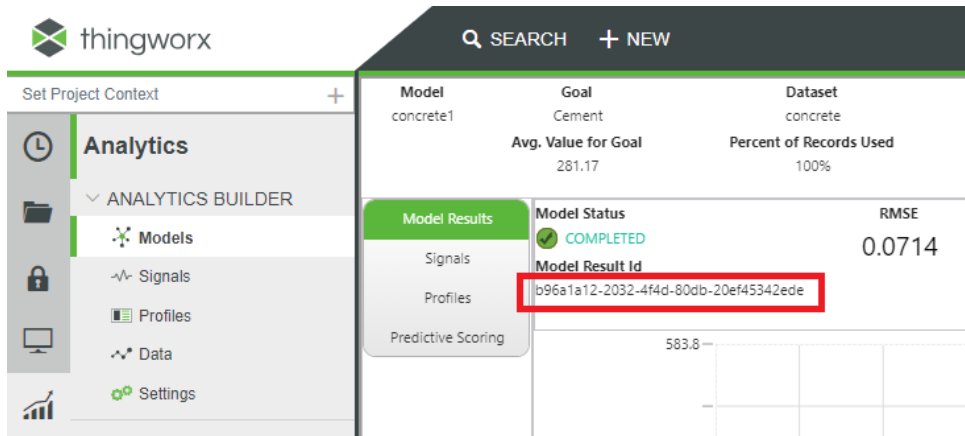
An example value for datasetURI is **dataset:/140b94c5-e62d-4e26-a9c8-c0007f552abd**

Save changes to Thing: ConcreteHelper.

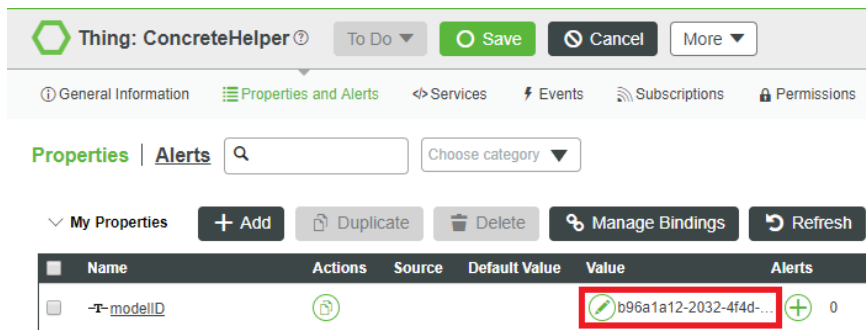
8 Set Model ID

Open Analytics Builder ... View Model ...

Copy Model Result Id to the clipboard as shown below:



Navigate to Thing: ConcreteHelper and set modelID value to the value in your clipboard as shown below:



An example value for modelID is **db17b49d-43a6-4685-8aa4-bc1944df0ae3**

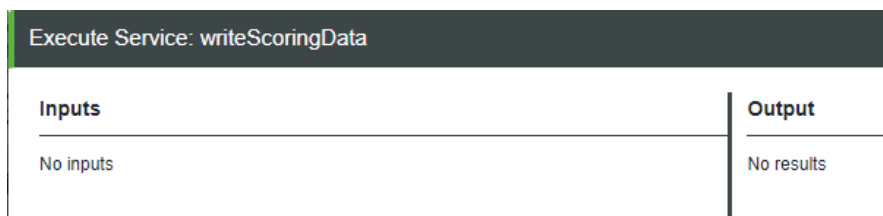
Save changes to Thing: ConcreteHelper.

9 Write Scoring Data

We now need to write some records for scoring to a data table to make it easy to use the mashup (by loading pre-set values). In order to do this you need to execute the following service:

ConcreteHelper.writeScoringData

The service does not display any outputs after completion, see below:



10 Using the Mashup

You can open the mashup by adapting this URL and then pasting into your browser:

<https://<your-servername:port>/Thingworx/Mashups/concretePrescMashup>

or

<http://<your-servername:port>/Thingworx/Mashups/concretePrescMashup>

(depending whether you're using HTTPS or HTTP)

You should see the blank mashup load as shown below:

Prescribe Concrete Mixture

The screenshot shows the 'Prescribe Concrete Mixture' interface. On the left, there is an 'Identifier' input field containing '0' and a blue 'Load Values' button. Below this are eight input fields for 'Water', 'Coarse Aggregate', 'Cement', 'Fine Aggregate', 'Slag', 'Superplasticizer', 'Fly Ash', and 'Age', each containing '0'. The middle section, 'Prescription Fields', features a list box with 'Cement' selected, and other options include 'Slag', 'FlyAsh', 'Water', 'Superplasticizer', and 'CoarseAggregate'. On the right, the 'Prescription Results' section is empty, with a blue 'Prescription' button positioned above it.

The aim of the mashup is to allow you to enter inputs on the left, select fields for optimizing in the middle, and then use the Prescription button on the right to trigger prescription results. Additional hints / details are shown in the subsections below.

10.1 Load Values

The Load Values button allows you to load a pre-defined set of input data (to avoid having to type out all fields)¹. To do this, type in an identifier from 1 to 20 and click Load Values as shown below:

Prescribe Concrete Mixture

Identifier	<input type="text" value="15"/>	Prescription Fields	Prescription Results
	<input type="button" value="Load Values"/>		
Water	<input type="text" value="218"/>	<input checked="" type="checkbox"/> Cement	<input type="button" value="Prescription"/>
Coarse Aggregate	<input type="text" value="949"/>	<input type="checkbox"/> Slag	
Cement	<input type="text" value="280"/>	<input type="checkbox"/> FlyAsh	
Fine Aggregate	<input type="text" value="671"/>	<input type="checkbox"/> Water	
Slag	<input type="text" value="72"/>	<input type="checkbox"/> Superplasticizer	
Superplasticizer	<input type="text" value="0"/>	<input type="checkbox"/> CoarseAggregate	
Fly Ash	<input type="text" value="3"/>		
Age	<input type="text" value="41"/>		

This will save you time, as you don't need to type input values manually, however you are free to do so, or to edit the input values if you wish.

10.2 Prescription Fields

In this list you can select 1 or more fields which you would like the system to optimize. This means that the system will determine, based on the inputs you provided, optimal values for these fields to maximize the Compressive Strength of the concrete.

10.3 Prescription Results

Once you've input some values and hit the "Prescription" button, you should get some results as shown below:

Prescribe Concrete Mixture

The interface is divided into three main sections:

- Input Fields:** Identifier (15), Water (218), Coarse Aggregate (949), Cement (280), Fine Aggregate (671), Slag (72), Superplasticizer (0), Fly Ash (3), and Age (41). A 'Load Values' button is located below the Identifier field.
- Prescription Fields:** A dropdown menu showing selected fields: Cement, Slag, FlyAsh, Water, Superplasticizer, and CoarseAggregate.
- Prescription Results:** Contains an 'originalScore' (28.59734936224) and an 'optimizedScore' (73.85705336663). A 'Prescription' button is present. Below it is a table of lever fields.

originalScore	leverFields		
28.59734936224	FieldName	OriginalValue	OptimalValue
73.85705336663	Water	218.00	121.80
	CoarseAggregate	949.00	931.72
	Cement	280.00	540.00
	FlyAsh	3.00	200.10

errorMessage

Here's an explanation of the fields shown in the Prescription Results:

10.3.1 Original Score

This is the prediction for Compressive Strength based on the input values you provided

10.3.2 Optimized Score

This is the increased Compressive Strength score which would result from making changes as recommended in the Lever Fields table.

10.3.3 Lever Fields

This is a table showing some fields with their original values (i.e. the values you provided on the far left of the screen) and the optimal values (the results of prescription). If you change the input values to these optimal values then you will maximise the outcome you are predicting: Compressive Strength.

Fields which you selected in the middle of the screen for optimization are included in the Lever Fields output table.

10.3.4 Error Message

This field displays any error from the Prediction microservice.

11 Main Services

In order to use this functionality in your own environment you may want to copy the services to use as code samples. Below are overviews of the services called to enable the functionality in the mashup.

11.1 ConcreteHelper.writeScoringData

Writes static data into a data table for loading later into the mashup (to avoid typing out all values).

11.2 ConcreteHelper.listFields

Runs when the mashup loads. It retrieves all field names from the dataset excluding field CompressiveStrength (which is the goal). These fields are used in the mashup to populate a list so that the user can select which fields to use for optimization.

11.3 ConcreteHelper.prescribeStrength

Gets the input values for prediction as well as the selected fields for optimization as inputs. Then calls RealtimeScore on the Prescriptive Microservice and returns all results including optimization values.