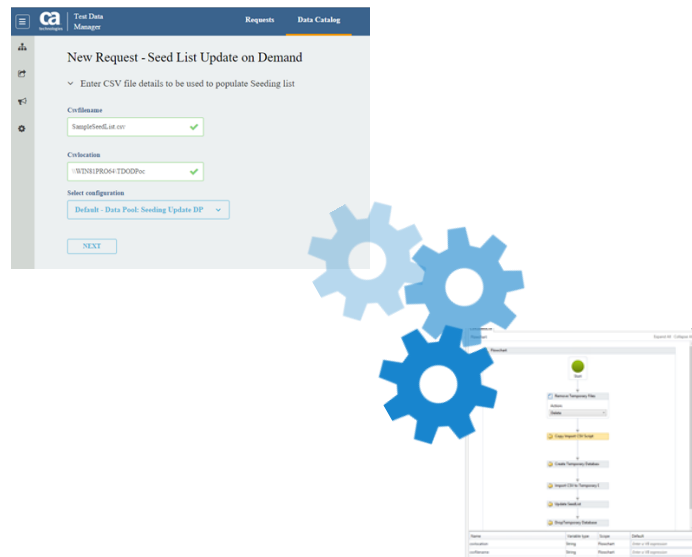


Introduction

I am relatively new to the Test Data Management products and I was recently asked to create a demonstration to show how a Test Data On Demand (TDoD) Self Service User could update the seed list table from a CSV. The scenario is that the Test Data On Demand (TDoD) Self Service User copies the CSV file to a known network location and then adds a request to the Test Data On Demand web UI to import this CSV file. The user enters the details on the location and name of a CSV and when the request is published the data is used to load the CSV file into the seed list table. The details for loading the data from a CSV to a seedlist have been published on the CA Docops web site at this URL <http://tinyurl.com/jbwa52u>.

Design

So my task was to create a form within CA Test Data On Demand and take the parameters entered into this form to trigger a javelin workflow which would automatically process the CSV file and load the contents into the seed list table.



In order to launch Javelin from TDM I utilised some new functionality that is currently available in the 3.2.2 Sandpit. The sandpit functionality I used enables an additional action which can be used to call Javelin and pass variables to the workflow. The workflow was started with the CSV file location and CSV filename and the Javelin workflow.

The following section explains the steps to recreate the demonstration.

If you are interested in taking part in the validation program and having access to the sand pit please email me keith.puzey@ca.com.

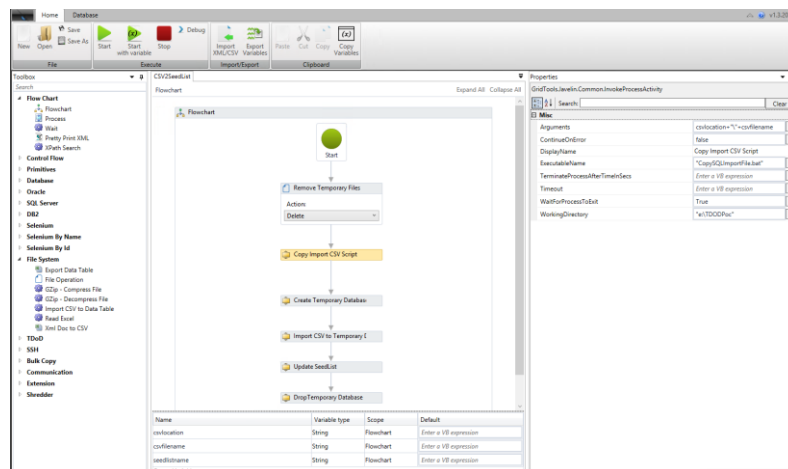
Implementation

The following section explains in detail how the demonstration was created .

To start we have to create a Javelin workflow and add variables required by the workflow that need to be exposed to the Tdod user. The workflow I created contained 6 steps as follows:

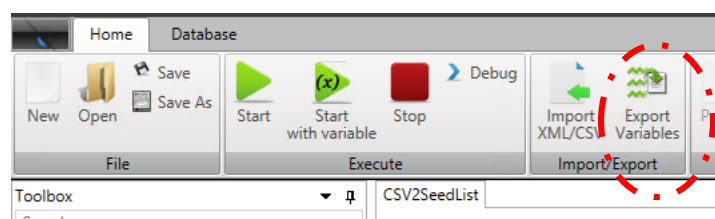
- 1 Remove Temporary Files - This step removed any temporary files before starting the import.
- 2 Copy Import CSV Script - The parameters for the CSV file name and location are passed from this step to a script that copies the CSV file its location to a temporary location and creates the import SQL script
- 3 Create Temporary Database - When importing from a CSV to the Seed List table you must first import the CSV data to a temporary table, this step creates a temporary staging table.
- 4 Import CSV to Temporary Database - This step calls the SQL import from the CSV to the temporary staging table.
- 5 Update SeedList - The data is imported into the seeding table from the temporary staging table.
- 6 DropTemporary Database - Maintenance task to drop the temporary staging table.

Note: The variables are accessible via the “Variables” link at the bottom of the design panel.



Note: The variables are accessible via the “Variables” link at the bottom of the design panel. Variables can be selected for use within the workflow by using the keyboard shortcut Ctrl - Space

Once the flow has been created and tested. Create an export of the variables by clicking on the “Export Variables” button.



The variable export will create a CSV file in this format

```
Name,Value,Scope  
"csvlocation","","Flowchart"  
"csvfilename","","Flowchart"  
"seedlistname","","Flowchart"
```

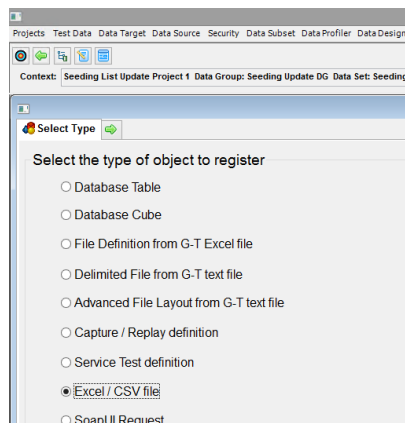
Use a text editor to create a second CSV file with column names that match the variables defined in the CSV that was exported from Javelin.

```
csvlocation,csvfilename,seedlistname
```

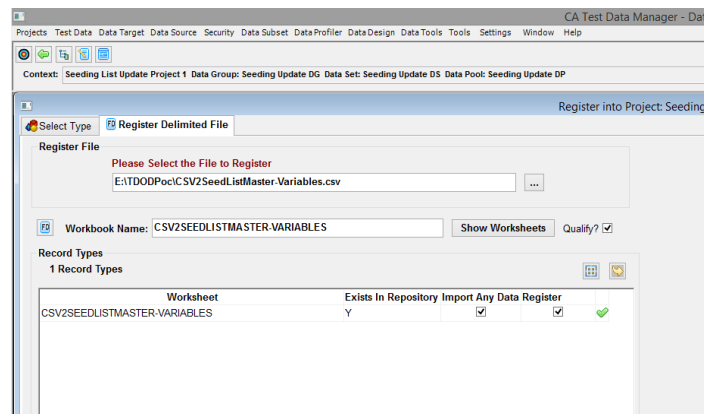
At this point we need to register the CSV file with TDM and create the javelin action within TDM

Login to CA Test Data Manager and the first step is to register the CSV file that contains the variables as column names to a data pool:

- I. Select Data Pool and from the Projects menu select the option “Register” and select the object type “Excel / CSV” to register

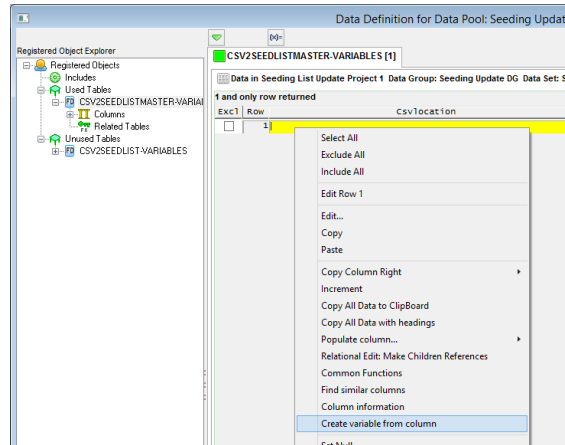


- II. Click green play button at the top of the screen and you will see the following screen

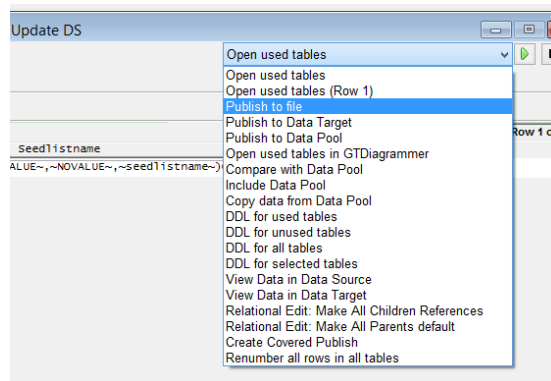


- III. Register CSV file that was created with variable names as columns

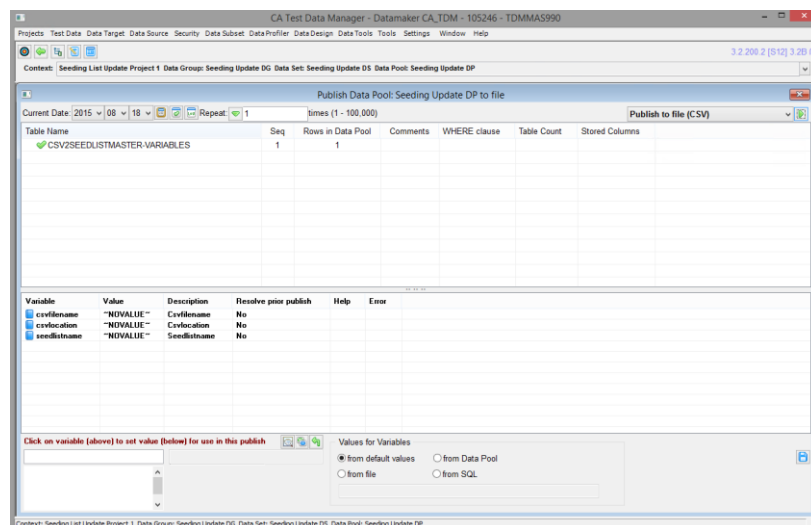
- IV. Click the button Show Worksheets and select the worksheet that was created in the CSV
- V. Click Register / Import
- VI. Click on the Data Pool and select the option to Edit Data
- VII. Add an empty row of data and In each column right click and select “Create variable from Column” for each value and click the save icon in the top right hand corner of the screen.



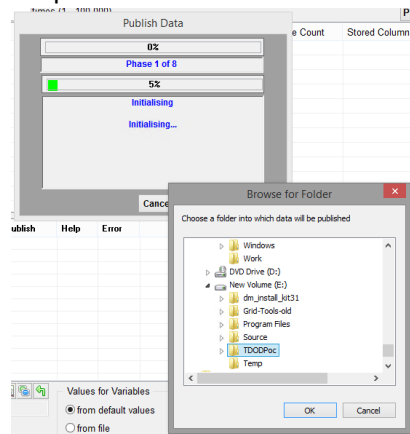
- VIII. Now we are going to publish the CSV file by selecting “publish to file” from the menu in the top right hand corner of the screen and pressing the green action arrow



- IX. From the drop down in the top right hand corner select the option to “publish to file (CSV)”



X. Select a location to publish the file to

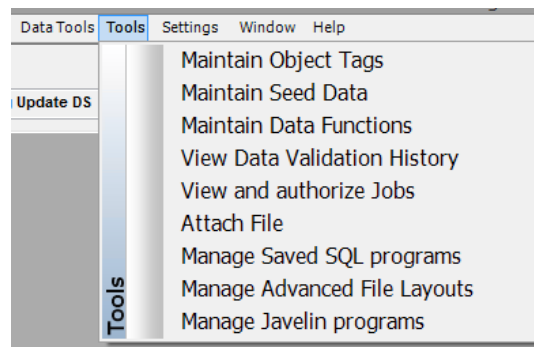


i.

XI. After Publishing the log file will be displayed

The data pool preparation is now complete and the next step is to create a post publishing action to trigger the javelin workflow. In version 3.2. The Javelin action is only currently available in the sandbox, and will be available in an upcoming release

Click on tools and select “Manage Javelin Program” at the bottom of the drop down.

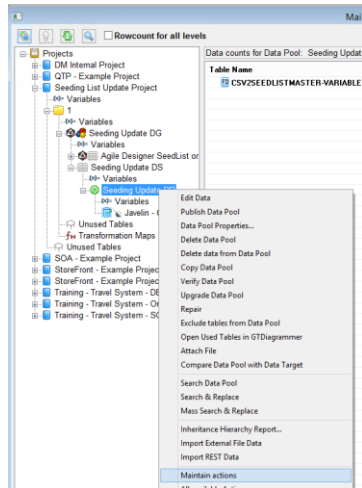


Click + to add a new javelin program

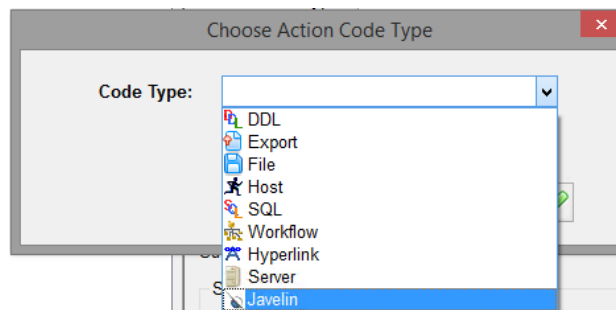


The Javelin file is the javelin workflow that the action will call (vwf)

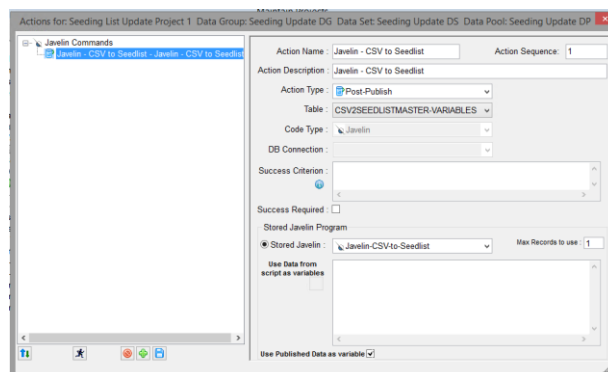
The variable file is the CSV file that is required for the javelin workflow and was exported from Javelin



Now Click on Pool and select option maintain actions, click on the green plus sign to create a new action.

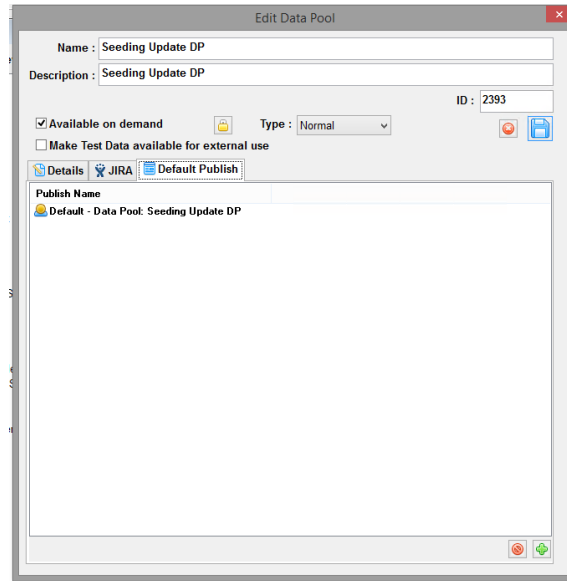


In the “Code Type” drop down select “Javelin” and enter a name for the action



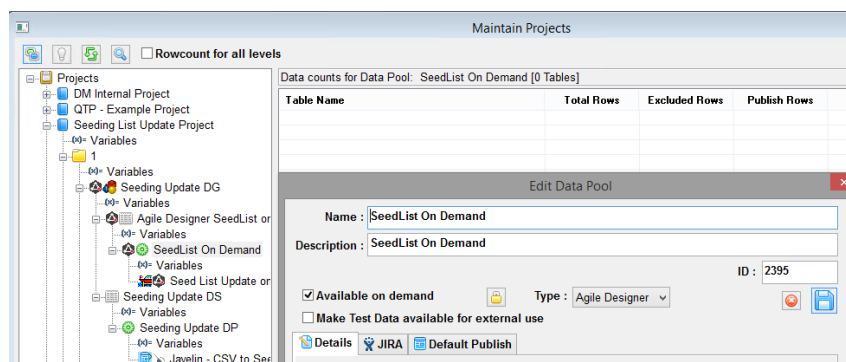
Change the action type to “Post-Publish” and in the stored javelin drop down select the workflow program that was registered in the previous step.

Tick option use published data as variable and in the table drop down select the data pool that was registered with the CSV file. Click save and close the action screen

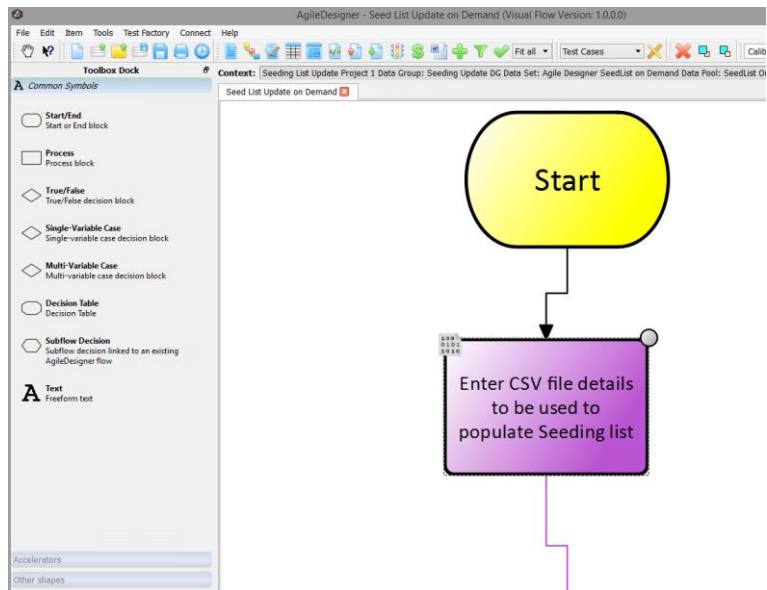


Go to data pool properties and ensure that default publish tab contains the successful publish

We are now going to create and agile designer flow and publish this to Test Data on Demand. The first step is to create a data pool with the type Agile designer as shown in the example below.

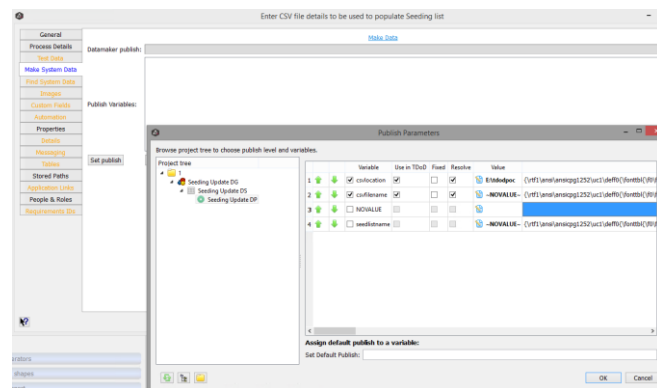


Now we can launch agile designer and create Agile designer flow and create a simple flow like the following example.

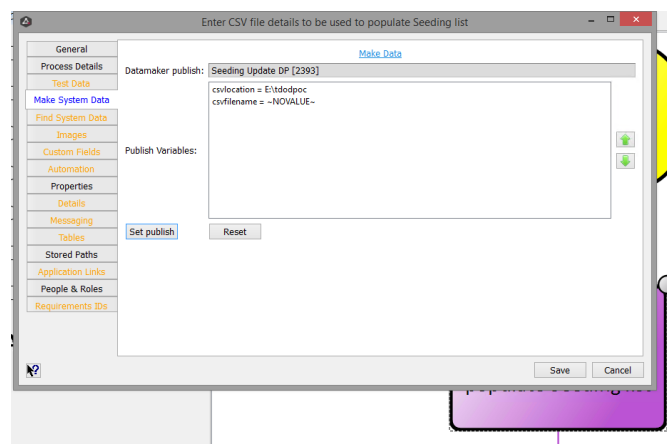


Double click on the square box (Process) and click on the link “Make System Data

Click on the Set Publish and navigate in the Project tree to find the data pool which was used to publish the CSV file.

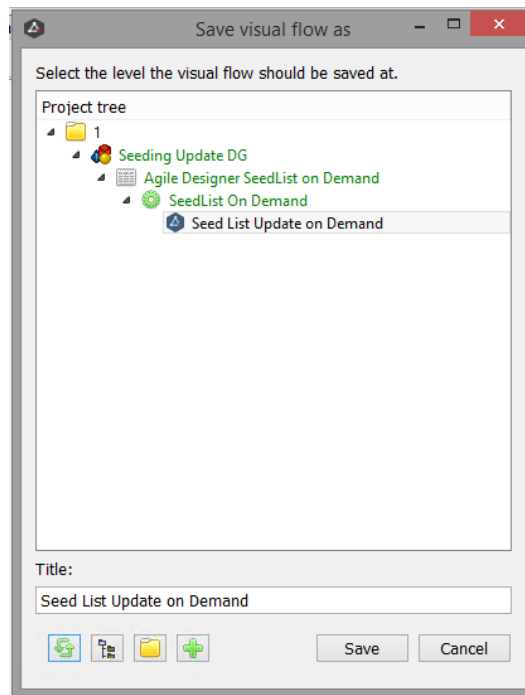


For any variables that are required to be entered into the Test Data On Demand UI ensure the corresponding tick boxes have been selected in the variable and “Use in TDod”. Default values can also be entered in the value section. Click OK to update the process



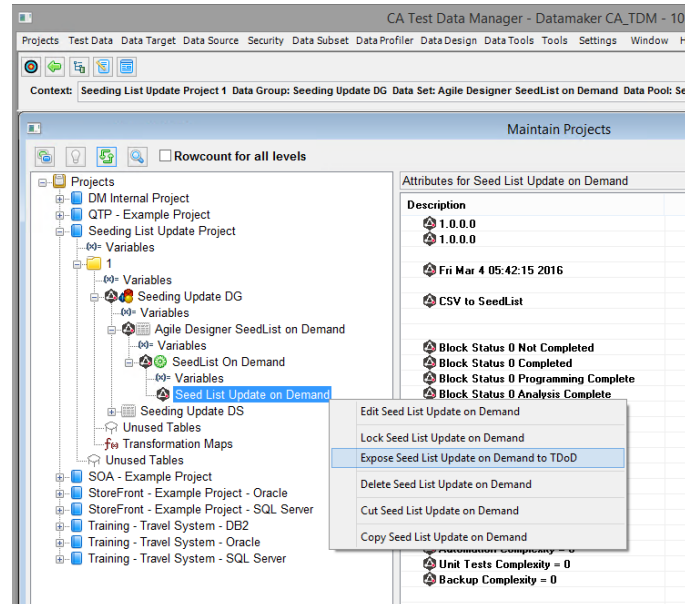
Confirm the order of the variables and click save.

To save the visual flow click on the file menu and select the option “Save as to Repository”

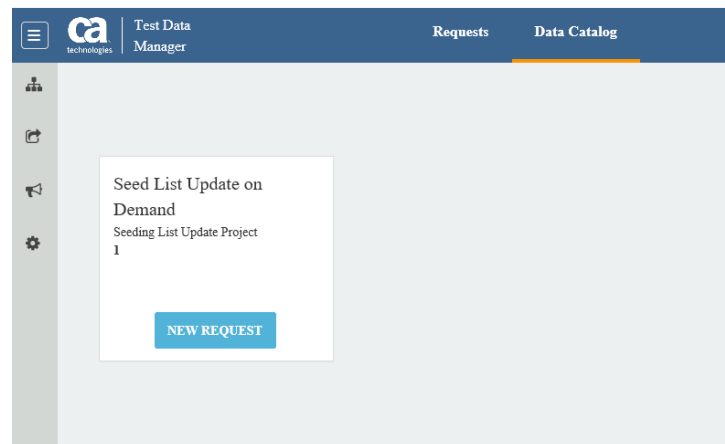


Navigate the project tree to find the agile designer data pool and enter a name for the form.

The final step to make the form available to TDod is to right click on the Visual flow object that was created in agile designer and select the option “Expose To TDod”



The final step is to access the Test Data on Demand screen and enter the details for the name and location of the CSV file. When you login to the TDoD UI the form will be available in the data catalog screen.



As you can see the form now shows the variables that were selected in Agile Designer. Enter the relevant details for this form and press next to submit the job.

Two screenshots of the 'New Request - Seed List Update on Demand' form. The left screenshot shows the 'Enter CSV file details to be used to populate Seeding list' section. It has two text input fields: 'Csvfilename' with the value 'SampleSeedList.csv' and a green checkmark, and 'Csvlocation' with the value '\\WIN81PRO64\\TDODPoc' and a green checkmark. Below these is a 'Select configuration' dropdown menu showing 'Default - Data Pool: Seeding Update DP'. At the bottom is a blue 'NEXT' button. The right screenshot shows the 'Configure request' section. It has an 'Email' text input field with a green checkmark and a 'Schedule' text input field with the value '03/08/2016'. Both screenshots have the same top navigation bar and sidebar as the first image.

Once the publish has completed the post action will run and the variables that were entered in the form will be passed to the workflow.

As you can see from this example the published zip file contains the log folder and details on the javelin workflow execution.

