

SUMMARY

This Tech Note will demonstrate a procedure to easily create large numbers of instances from an existing template or templates to speed the process of developing large galaxies without the need to manually create each individual instance.

SITUATION

The Tech Note is presented in two parts. The first part is for people who want to use bulk-generation on simple templates – that is, templates which do not use containment. The second part is for templates that use containment.

It is recommended to follow both exercises to see the similarities and differences between them.

Frequently Asked Questions:

Who would want to do this?

Anyone creating a large enough number of instances from existing ArcestrA templates that doing so using the conventional method. In the first part of the example, you will create 100 instances from a simple template called “Reactor”.

In the second part of the example, you will create 100 instances of a complex template called “Tank”.

In what version(s) of Wonderware Application Server can this be done?

The screenshots for this TechTip are taken from Application Server 17.3.1. However, this procedure should work (possibly with some cosmetic differences) with any version of Application Server.

Are any special knowledge or skills required in order to use this technique?

An understanding of the Wonderware Application Server product. Also, a basic understanding of CSV files and the tools used to modify them is helpful.

Do I need to do anything special to prepare to use this technique?

This technique should work with any template built in the ArcestrA IDE. Note that the technique cannot be used to bulk-generate templates – only instances.

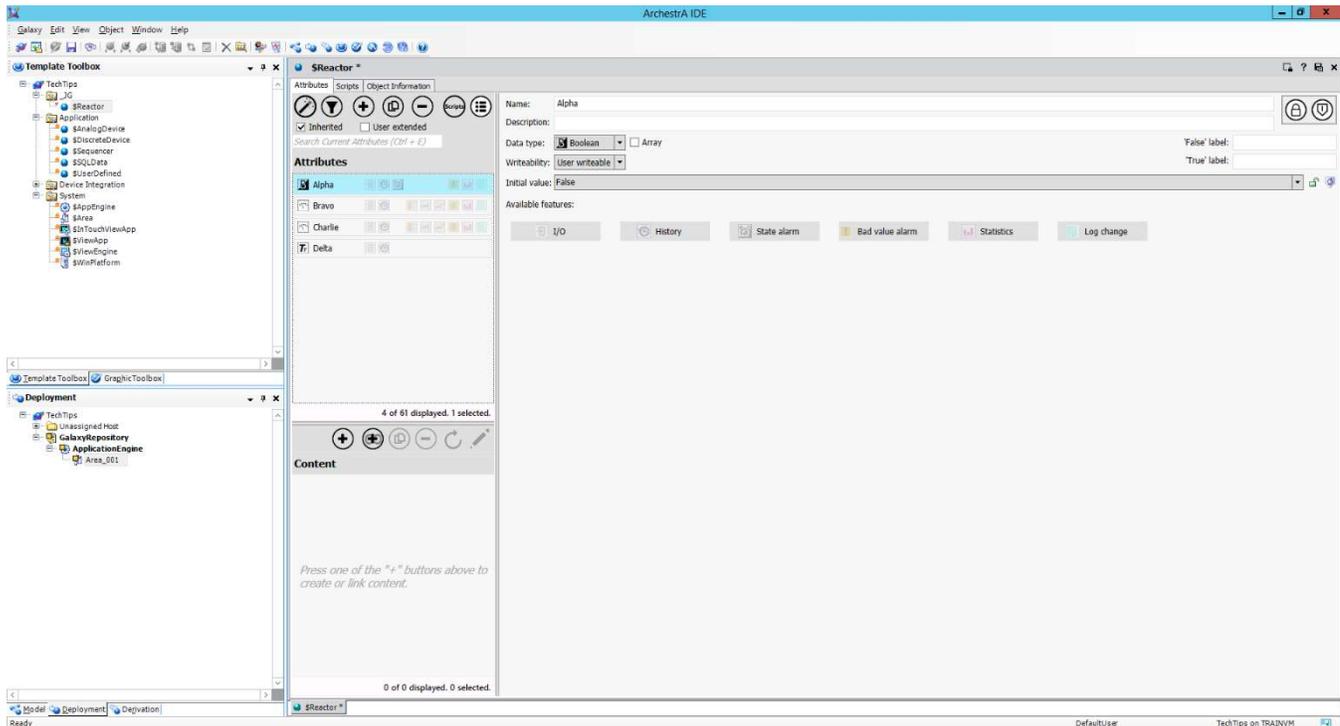
Bulk-generation with simple templates.

Scenario: There is a template called “Reactor” which contains four attributes and for which 100 instances must be created. The attributes serve no purpose in this example other than to illustrate that templates being used in this way may contain attributes without affecting the outcome. In this situation, the attributes have no additional features (I/O, History, Alarms, etc.) configured but, if they did, it would have effect on using this technique. The same is true for scripting. That is, the templates can have any number of scripts necessary for them to perform their needed functions and this procedure will still work.

Tech Note: Bulk Generation in Application Server

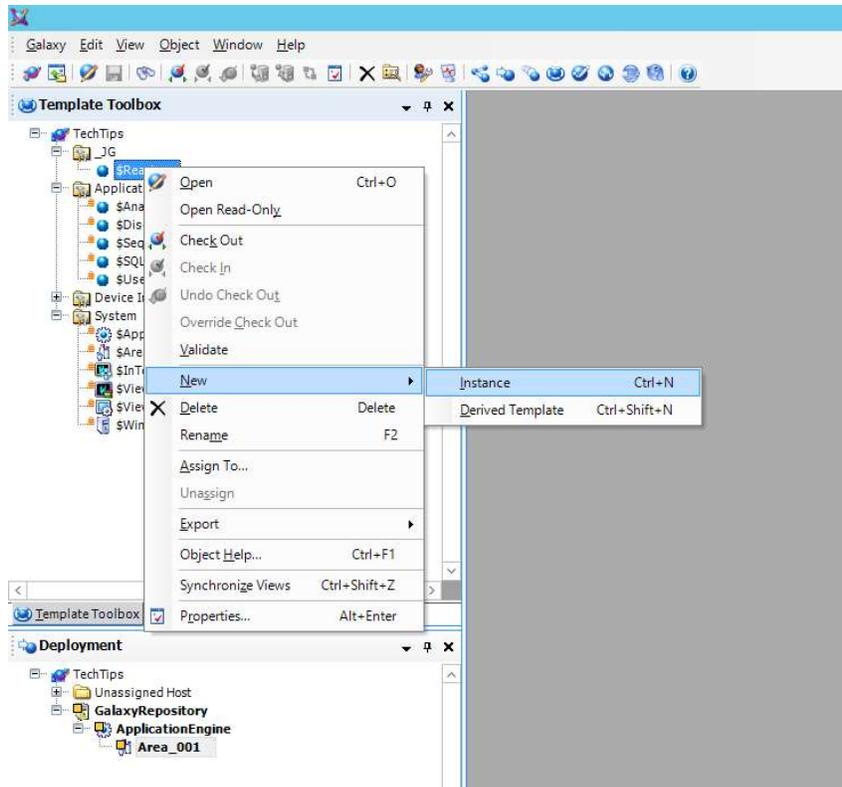
Procedure:

1. Here is the template to be used for the bulk-generation. In order to create it, right-click on the \$UserDefined base template and create a derived template called "Reactor".

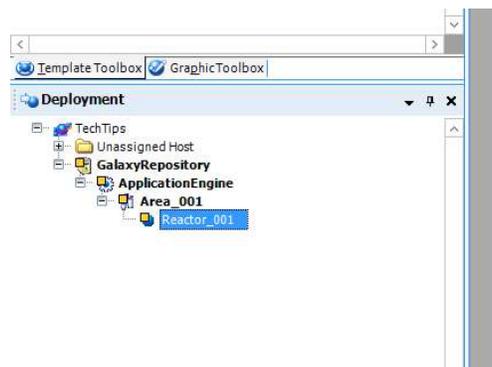


Note that there is already a Platform, Application Engine and Area instance created to receive the bulk-generated instances of Reactor. While it is not strictly necessary to create these things prior to the bulk-generation, it is preferable.

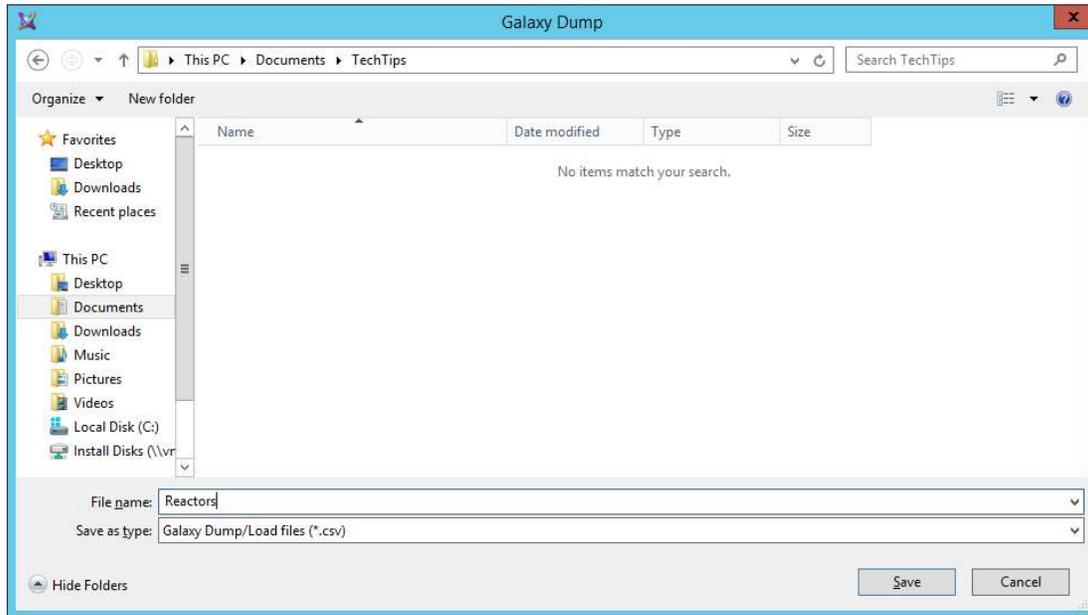
2. Create the first instance manually using traditional methods. This step is also not entirely necessary but it is helpful.



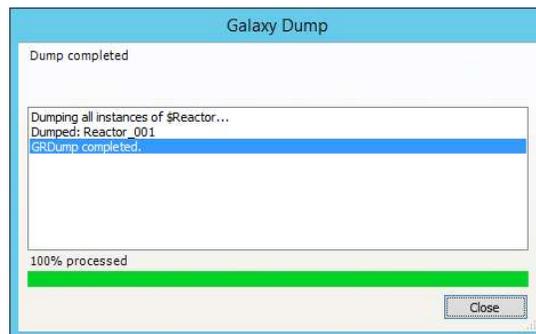
The result will be the creation of an initial instance as shown below.



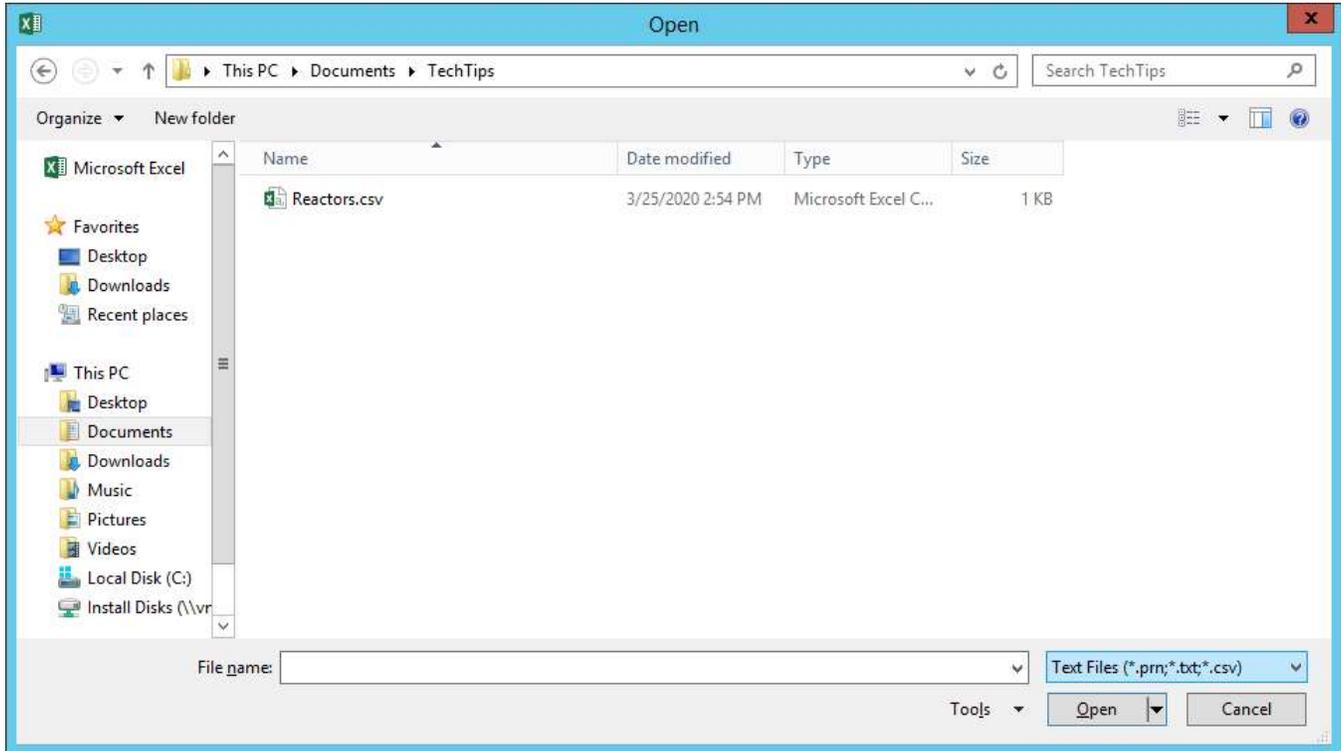
3. Next, right-click on the instance and chose the “Export > Galaxy Dump” option. In this case, the file is saved as “Reactors.csv”.



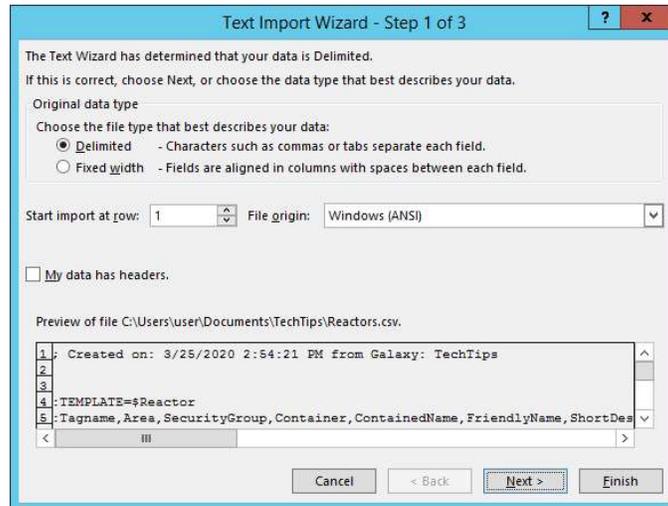
Note the progress of the export:



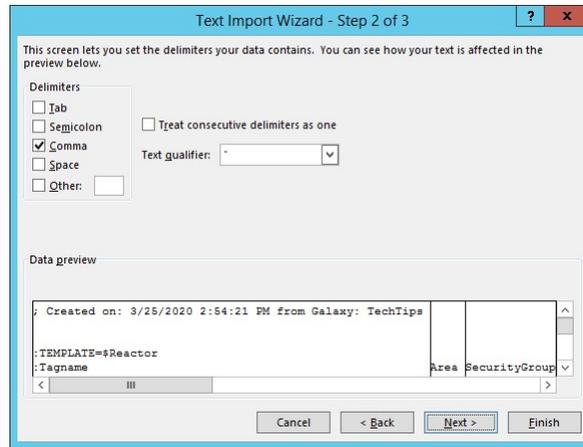
4. Any CSV file editor can be used to modify the file. In this example, Excel will be used. In order for Excel to properly load a “Galaxy Dump” CSV, the file should be loaded from within Excel rather than by double-clicking on the CSV file directly.



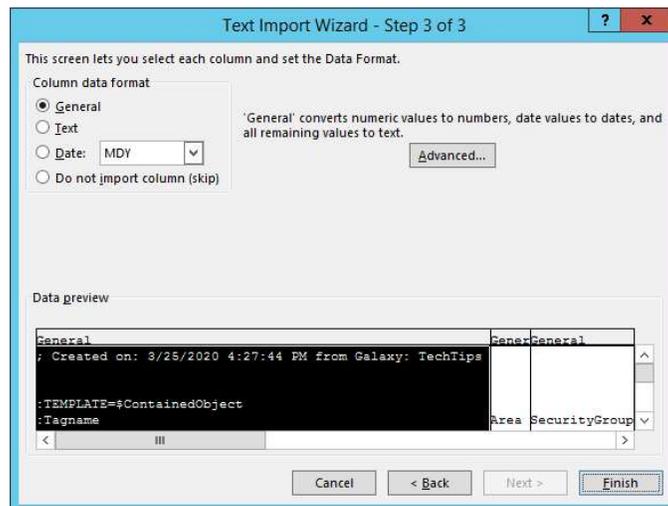
5. On the first screen of the Text Import Wizard, leave everything at the default and click “Next”. On the second screen, make sure that only the “Comma” delimiter is selected and no others.



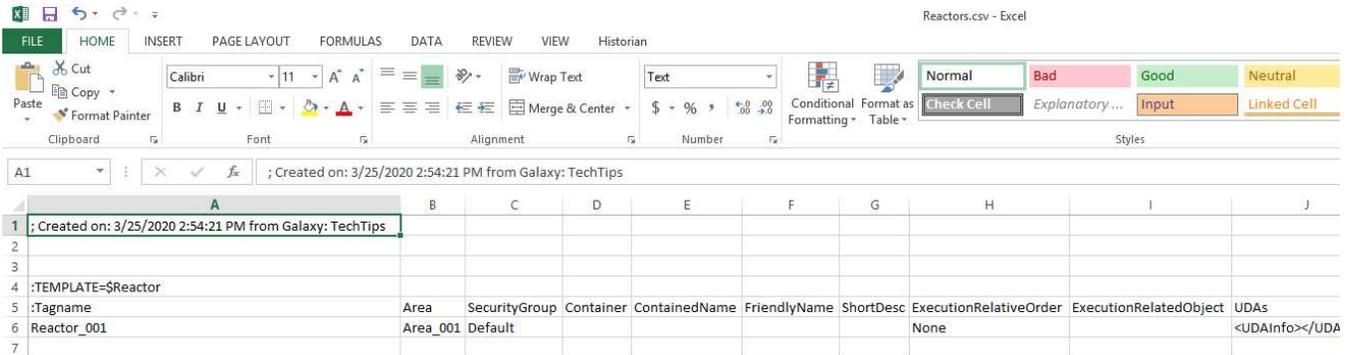
6. On the second screen, make sure that only the “Comma” delimiter is selected.



7. On the third screen, make sure the “Column data format” is set to “General”.



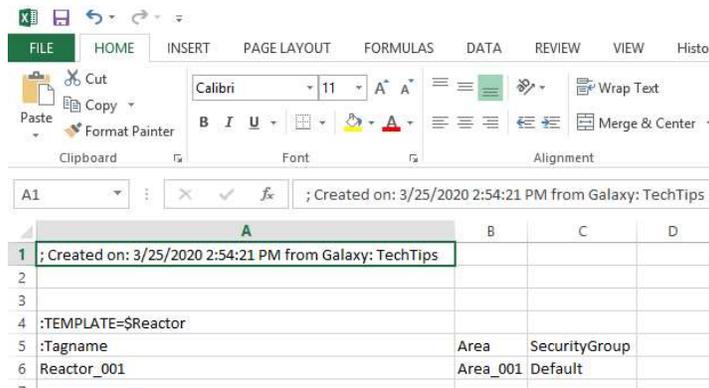
8. Click “Finish” and then arrange the column widths to clearly display the data.



	A	B	C	D	E	F	G	H	I	J
1	; Created on: 3/25/2020 2:54:21 PM from Galaxy: TechTips									
2										
3										
4	:TEMPLATE=\$Reactor									
5	:Tagname	Area	SecurityGroup	Container	ContainedName	FriendlyName	ShortDesc	ExecutionRelativeOrder	ExecutionRelatedObject	UDAs
6	Reactor_001	Area_001	Default					None		<UDAInfo></UDA
7										

9. For bulk-generation to succeed, not all of these columns are required. For this procedure, it is fine to delete the unnecessary ones – it makes working with the file much easier. In this situation delete every column after “SecurityGroup” (Column C). When working with contained objects/templates, “Container” (Column D) and “ContainedName” (Column E) are also necessary.

After deleting the unnecessary columns, the CSV should look like this:



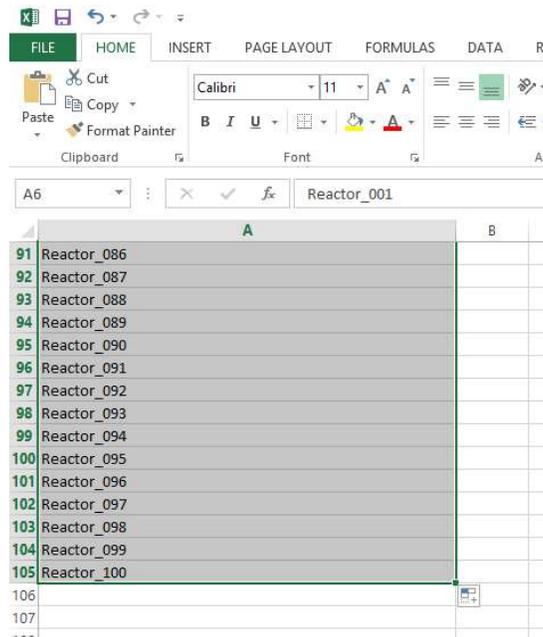
	A	B	C	D
1	; Created on: 3/25/2020 2:54:21 PM from Galaxy: TechTips			
2				
3				
4	:TEMPLATE=\$Reactor			
5	:Tagname	Area	SecurityGroup	
6	Reactor_001	Area_001	Default	
7				

Tech Note: Bulk Generation in Application Server

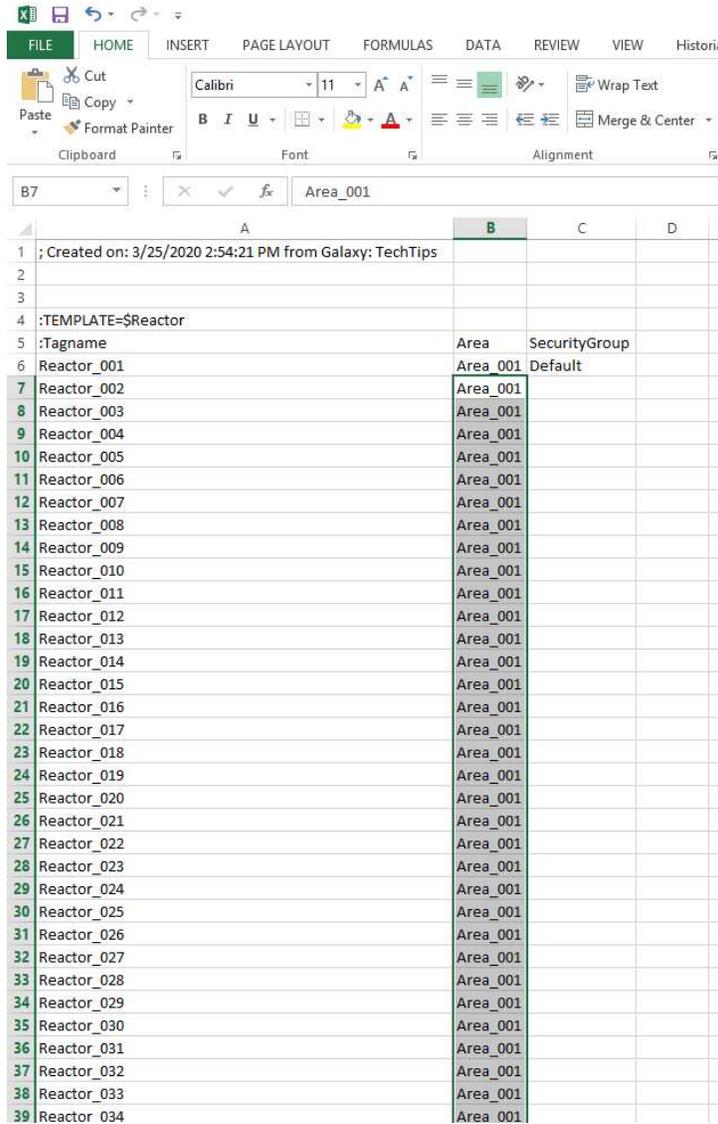
10. To create the additional instances, all that is required is to add the additional names below “Reactor_001”.

For this exercise, select cell A6 containing “Reactor_001”, release the mouse and then rest the pointer on the lower-right corner of the cell so the mouse cursor changes to a cross.

Left-click and drag from the corner of the cell down to row 105 so that Excel fills the column with consecutively-numbered reactor names and the last one reads “Reactor_100”.

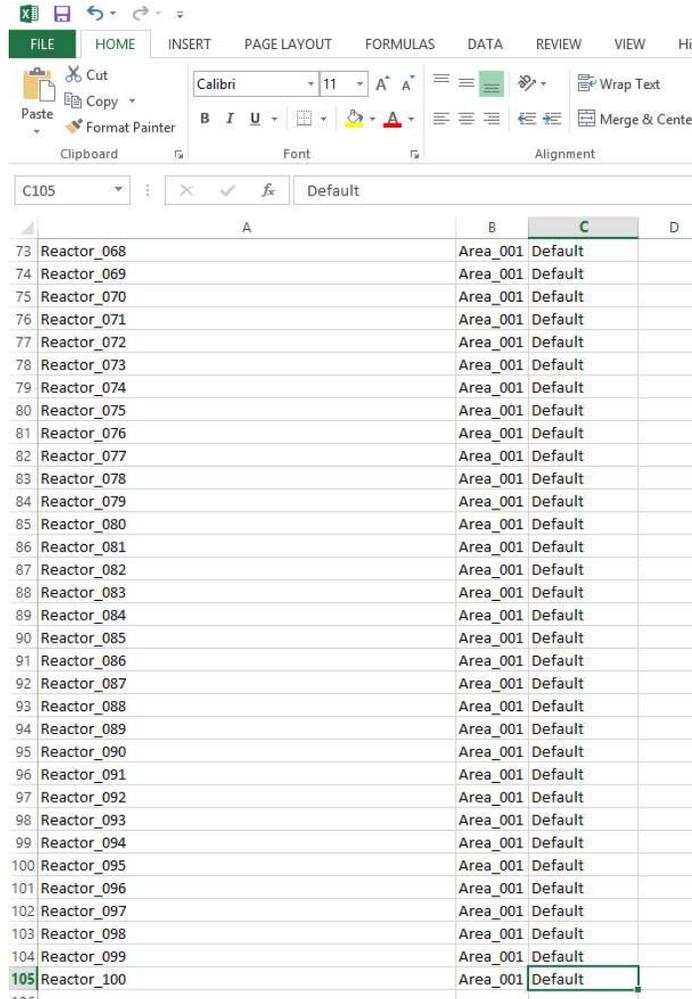


11. The “Area” column (B) determines where the instance created will be assigned. If the area specified in this column does not exist, the instance will be placed in the “Unassigned Host” folder in the Deployment view. In this simple example, they will all be made the same but they could be modified individually if desired. Copy and paste “Area_001” into the B column cells beside the Reactors.



	A	B	C	D
1	; Created on: 3/25/2020 2:54:21 PM from Galaxy: TechTips			
2				
3				
4	:TEMPLATE=\$Reactor			
5	:Tagname	Area	SecurityGroup	
6	Reactor_001	Area_001	Default	
7	Reactor_002	Area_001		
8	Reactor_003	Area_001		
9	Reactor_004	Area_001		
10	Reactor_005	Area_001		
11	Reactor_006	Area_001		
12	Reactor_007	Area_001		
13	Reactor_008	Area_001		
14	Reactor_009	Area_001		
15	Reactor_010	Area_001		
16	Reactor_011	Area_001		
17	Reactor_012	Area_001		
18	Reactor_013	Area_001		
19	Reactor_014	Area_001		
20	Reactor_015	Area_001		
21	Reactor_016	Area_001		
22	Reactor_017	Area_001		
23	Reactor_018	Area_001		
24	Reactor_019	Area_001		
25	Reactor_020	Area_001		
26	Reactor_021	Area_001		
27	Reactor_022	Area_001		
28	Reactor_023	Area_001		
29	Reactor_024	Area_001		
30	Reactor_025	Area_001		
31	Reactor_026	Area_001		
32	Reactor_027	Area_001		
33	Reactor_028	Area_001		
34	Reactor_029	Area_001		
35	Reactor_030	Area_001		
36	Reactor_031	Area_001		
37	Reactor_032	Area_001		
38	Reactor_033	Area_001		
39	Reactor_034	Area_001		

12. Do the same for the “SecurityGroup” column. In this example, all of the instances will be assigned to the value “Default”. Make sure that all of the Reactors have their Area and SecurityGroup assigned for this example as shown below.

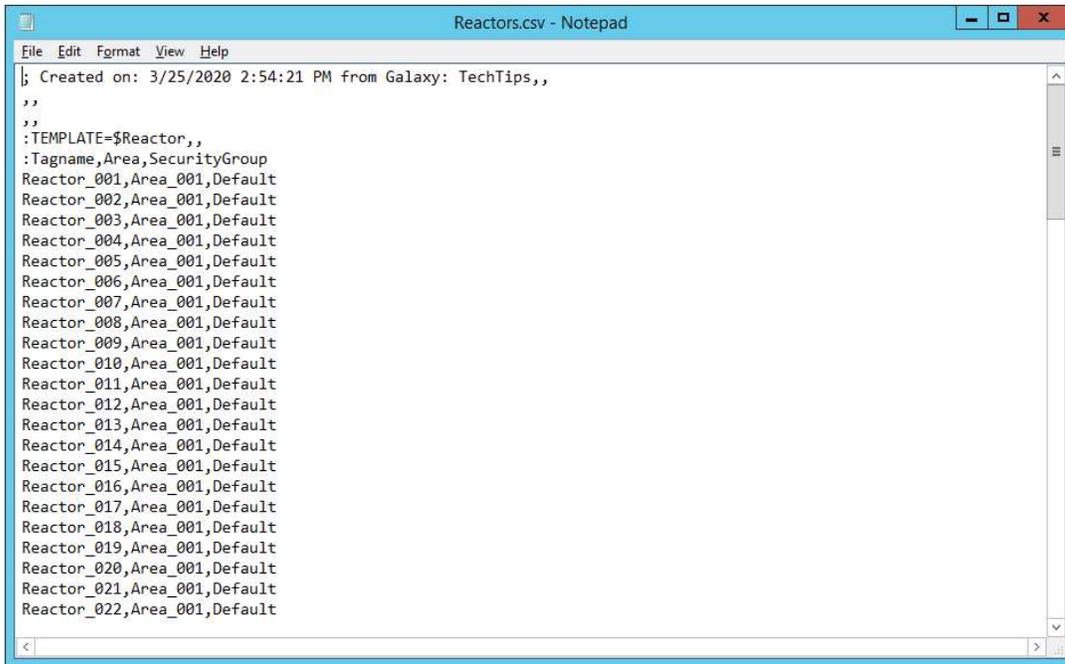


	A	B	C	D
73	Reactor_068	Area_001	Default	
74	Reactor_069	Area_001	Default	
75	Reactor_070	Area_001	Default	
76	Reactor_071	Area_001	Default	
77	Reactor_072	Area_001	Default	
78	Reactor_073	Area_001	Default	
79	Reactor_074	Area_001	Default	
80	Reactor_075	Area_001	Default	
81	Reactor_076	Area_001	Default	
82	Reactor_077	Area_001	Default	
83	Reactor_078	Area_001	Default	
84	Reactor_079	Area_001	Default	
85	Reactor_080	Area_001	Default	
86	Reactor_081	Area_001	Default	
87	Reactor_082	Area_001	Default	
88	Reactor_083	Area_001	Default	
89	Reactor_084	Area_001	Default	
90	Reactor_085	Area_001	Default	
91	Reactor_086	Area_001	Default	
92	Reactor_087	Area_001	Default	
93	Reactor_088	Area_001	Default	
94	Reactor_089	Area_001	Default	
95	Reactor_090	Area_001	Default	
96	Reactor_091	Area_001	Default	
97	Reactor_092	Area_001	Default	
98	Reactor_093	Area_001	Default	
99	Reactor_094	Area_001	Default	
100	Reactor_095	Area_001	Default	
101	Reactor_096	Area_001	Default	
102	Reactor_097	Area_001	Default	
103	Reactor_098	Area_001	Default	
104	Reactor_099	Area_001	Default	
105	Reactor_100	Area_001	Default	

Note: if you are not using the SecurityGroup functionality in Application Server (or aren’t sure what it does), be certain that all instances are set to “Default”. Failure to do so may make the instance behave in an unexpected manner.

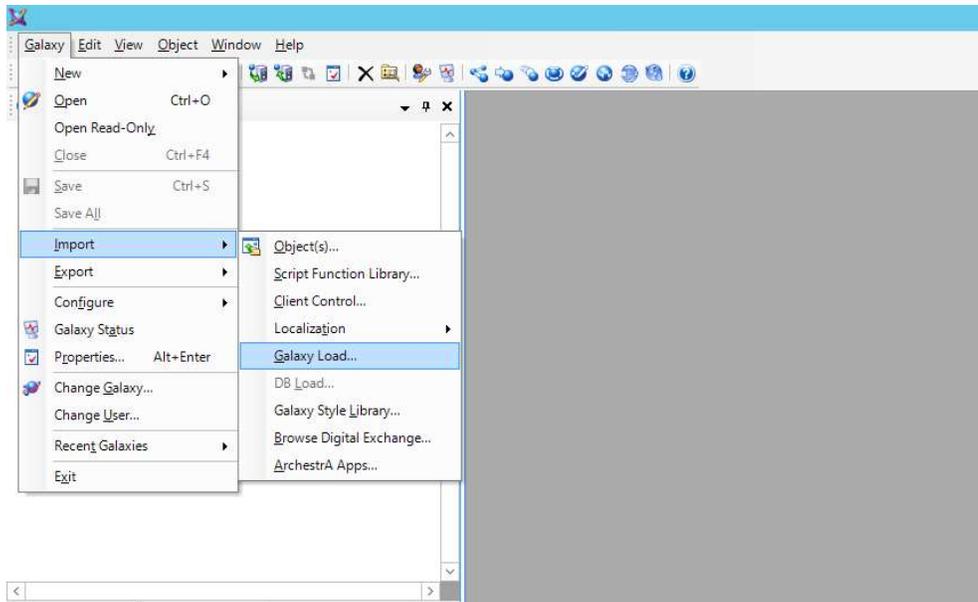
13. Save the CSV file – make sure it is saved as a CSV file and not any other format. If you want to confirm this, it is advisable to do it BEFORE you close Excel.

To confirm: open Windows File Explorer, right-click on the file and select “Edit”. This should open the file in Notepad where it will be obvious if the CSV format has been preserved...



```
Reactors.csv - Notepad
File Edit Format View Help
; Created on: 3/25/2020 2:54:21 PM from Galaxy: TechTips,,
,,
,,
:TEMPLATE=$Reactor,,
:Tagname,Area,SecurityGroup
Reactor_001,Area_001,Default
Reactor_002,Area_001,Default
Reactor_003,Area_001,Default
Reactor_004,Area_001,Default
Reactor_005,Area_001,Default
Reactor_006,Area_001,Default
Reactor_007,Area_001,Default
Reactor_008,Area_001,Default
Reactor_009,Area_001,Default
Reactor_010,Area_001,Default
Reactor_011,Area_001,Default
Reactor_012,Area_001,Default
Reactor_013,Area_001,Default
Reactor_014,Area_001,Default
Reactor_015,Area_001,Default
Reactor_016,Area_001,Default
Reactor_017,Area_001,Default
Reactor_018,Area_001,Default
Reactor_019,Area_001,Default
Reactor_020,Area_001,Default
Reactor_021,Area_001,Default
Reactor_022,Area_001,Default
```

14. To import the CSV into Application Server, click on “Galaxy > Import > Galaxy Load”

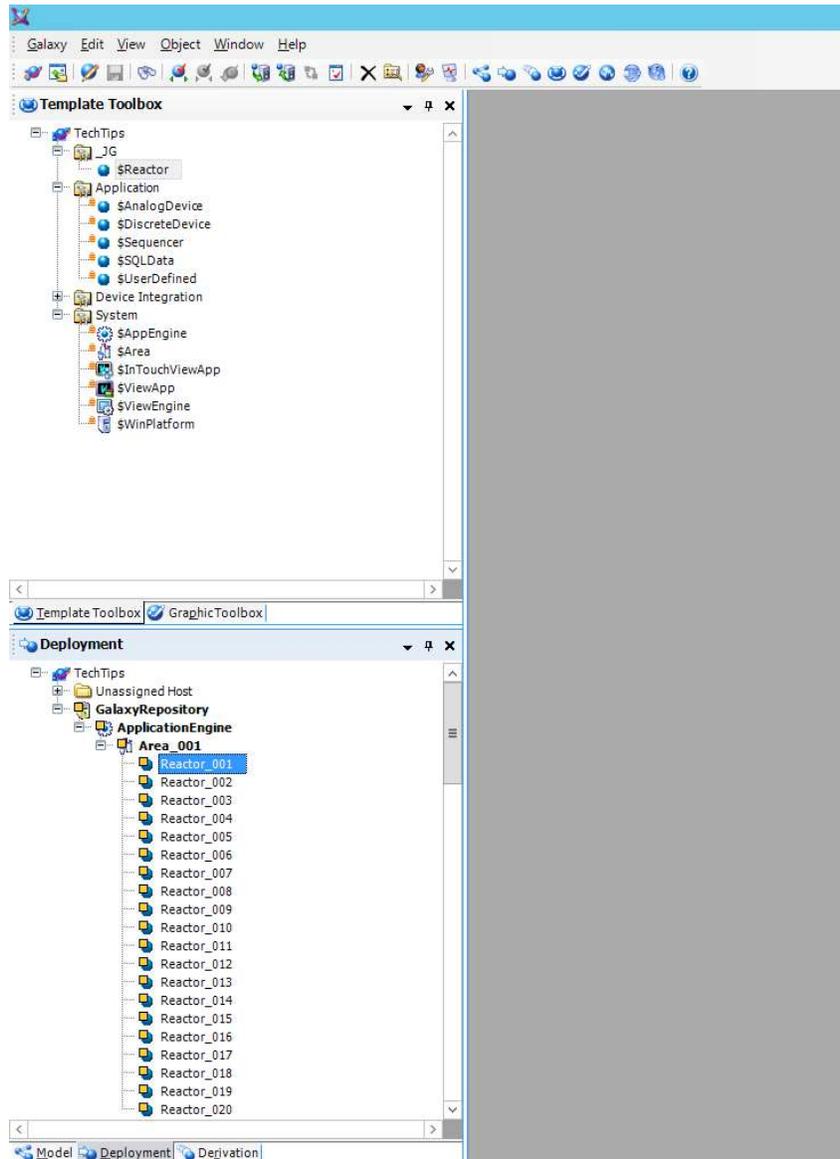


Load the CSV file and the “Conflict Resolution” window will appear:



In this example, the default “Skip” will be selected. This will skip over any existing instances whose names match what’s in the CSV file.

When the load is complete, check the Deployment view and check that all the instances have been imported...



Using this method, large numbers of instances can easily be created. The example above assumed the creation of a simple template – that is, one with no contained child templates.

The procedure can also be used with templates using containment, although the procedure is slightly more complex.

Bulk-generation with complex templates.

Scenario: There is a template called “Tank” which contains multiple child templates that requires multiple instances to be generated.

This example will use the same Galaxy as the previous one. However, a new area (“Area_002”) will be created to show the new objects separately from the Reactors in the previous example.

Create the following derived templates...

From \$UserDefined: \$Level, \$Pump, \$Tank, \$Temperature and \$Valve

From \$Pump: \$OutletPump

From \$Temperature: \$PumpTemperature and \$TankTemperature

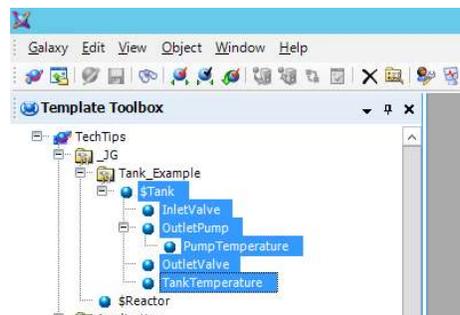
From \$Valve: \$InletValve and \$OutletValve

Next, drag-and-drop templates as follows...

Into \$OutletPump: \$PumpTemperature

Into \$Tank: \$InletValve, \$OutletPump, \$OutletValve, \$TankTemperature

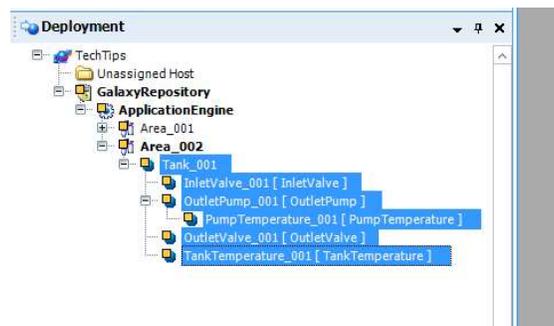
The results should look like this:



In order to create multiple instances of this template, the same basic procedure as before will be followed with some minor differences...

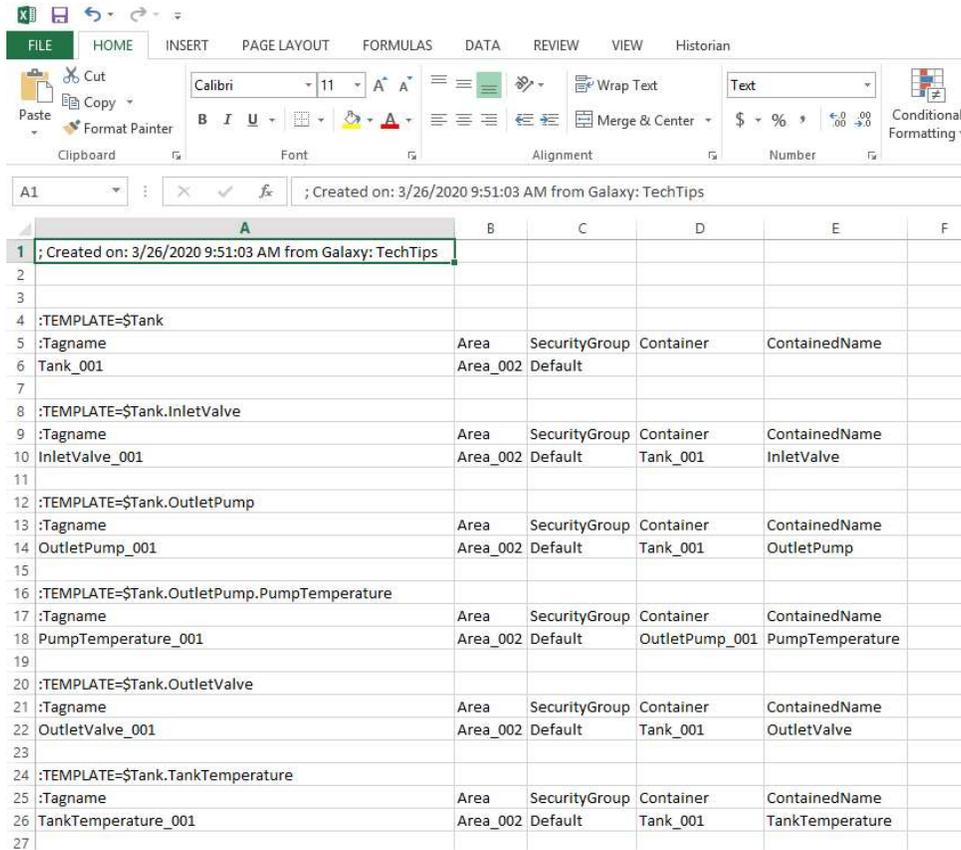
1. Manually create a single instance and then select all the elements as shown below. Note, that it is necessary to expand the "OutletPump_001" instance to reveal the child object. In more complex structures, it is useful to select the top-level object and then press the asterisk "*" key on the numeric keypad.

Once expanded, select all the instances as shown below, right-click and export using the "Galaxy Dump" option



2. Open Excel and load the CSV file using the same options as in the previous section (comma delimited and "General" column data format).

3. Delete all the columns to the right of “ContainedName” (Column E) so that it looks like this:



	A	B	C	D	E	F
1	; Created on: 3/26/2020 9:51:03 AM from Galaxy: TechTips					
2						
3						
4	:TEMPLATE=\$Tank					
5	:Tagname	Area	SecurityGroup	Container	ContainedName	
6	Tank_001	Area_002	Default			
7						
8	:TEMPLATE=\$Tank.InletValve					
9	:Tagname	Area	SecurityGroup	Container	ContainedName	
10	InletValve_001	Area_002	Default	Tank_001	InletValve	
11						
12	:TEMPLATE=\$Tank.OutletPump					
13	:Tagname	Area	SecurityGroup	Container	ContainedName	
14	OutletPump_001	Area_002	Default	Tank_001	OutletPump	
15						
16	:TEMPLATE=\$Tank.OutletPump.PumpTemperature					
17	:Tagname	Area	SecurityGroup	Container	ContainedName	
18	PumpTemperature_001	Area_002	Default	OutletPump_001	PumpTemperature	
19						
20	:TEMPLATE=\$Tank.OutletValve					
21	:Tagname	Area	SecurityGroup	Container	ContainedName	
22	OutletValve_001	Area_002	Default	Tank_001	OutletValve	
23						
24	:TEMPLATE=\$Tank.TankTemperature					
25	:Tagname	Area	SecurityGroup	Container	ContainedName	
26	TankTemperature_001	Area_002	Default	Tank_001	TankTemperature	
27						

4. Next, it will be necessary to insert 100 empty rows for each of the instances (Tank_001, InletValve_001, etc.)

The easiest way to do this is to select the row where you want the insert to begin (for example: row 7 directly beneath "Tank_001") by left-clicking on the row number. Then, scroll down and shift-left-click on the row where you want the insert to end (in this example, row 106). Finally, click the "Insert" button on the right edge of the "Home" tab in Excel. This will insert 100 empty rows. Technically, only 99 are necessary since the first instance already exists but a few extra empty rows don't hurt anything.

Repeat this process for each of the other instances except the last one since there is already sufficient space under it.

5. Go back to the top of the file and select the cell containing "Tank_001". Then, rest the mouse on the lower left corner of the cell so the cursor changes to a "+" and then drag down to fill all the cells down to row 105 ("Tank_100").

6. For the Tank_xxx section, copy and paste the "Area" and "SecurityGroup" values ("Area_002" and "Default") into the empty cells underneath so they fill all the rows down to Tank_100 with the same values.

Leave the "Container" and "ContainedName" empty.

7. For the InletValve_001 section, repeat the procedure used to fill the Tagname column with consecutively-numbered inlet valves. Also, copy/paste the "Area" and "SecurityGroup" as before.

Since "InletValve" is a child of the "Tank" object, it is also necessary to specify the "Container" and "ContainedName".

For "Container", use the same drag-to-fill technique as with the "Tagname". This will result in consecutively-numbered containers.

For "ContainedName", simply copy/paste as was done with "Area" and "SecurityGroup".

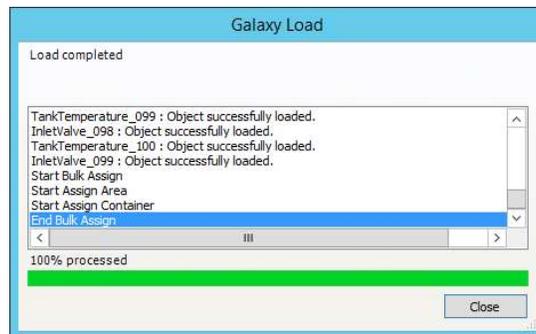
Note that the "ContainedName" remains the same for all instances but the "Container" must be different.

8. Use "Save As..." to save the results as a CSV file. Do ***NOT*** save as a "Unicode Text" file (the default), this will not import properly.

9. In the ArchestrA IDE, use “Galaxy > Import > Galaxy Load” to import the file. Choose the “Skip” option for Conflict Resolution as before.

Note: The import process with complex structures and/or large numbers of instances can take some time. Just because the load reaches 100% processed doesn’t mean it is complete.

Be sure to watch the status window and the “Bulk Assign”, “Assign Area” and “Assign Container” procedures are complete and the “End Bulk Assign” message appears.



Once the Bulk Assign has completed, the Galaxy Load window may be closed and the instances will be present in the galaxy.

Rev 1.0 – J. Gwynne – 3/26/20
