

Arena Grid Computing Utility

How to use the utility

This utility allows a user to run multiple replications of a model across several different machines on their network. In order to do this, every machine must have Arena installed on it. This utility will not work in run-time mode; every machine must have Arena with activation (at least Arena Basic Edition).

Because the utility runs discrete replications separate from each other, the following must be true of any model run with the utility:

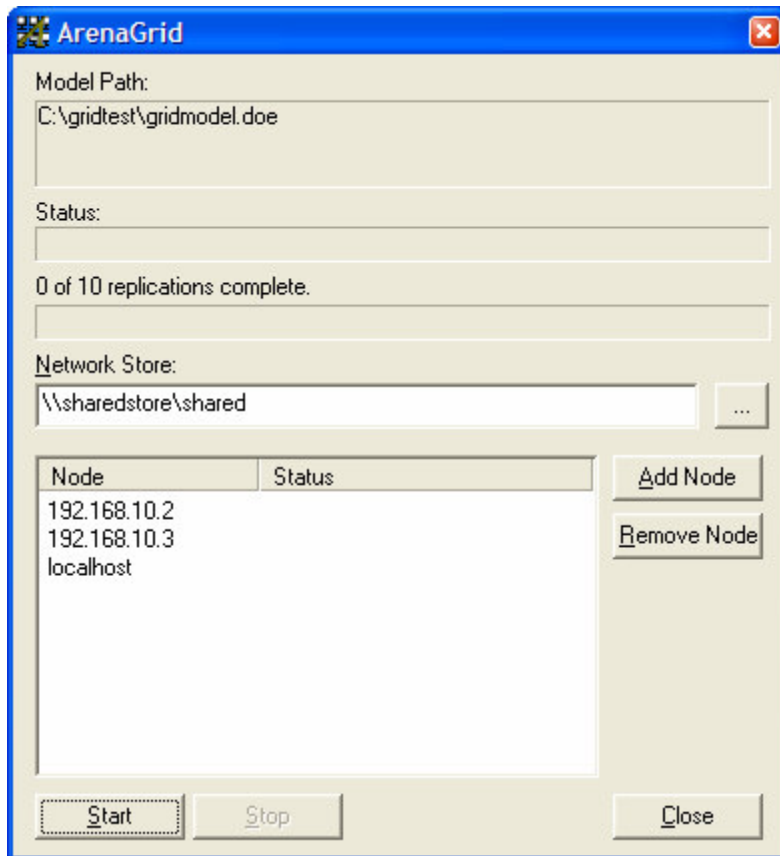
1. It must have a finite run time (either a finite replication length, or valid terminating condition)
2. It must not require information from previous replications (i.e. Statistics and the System should be initialized between replications, see Run->Setup->Replication Parameters tab).
3. In order to get the results of the simulation, the model must generate a report database.

Installation

1. Copy the *GridAddOn.dll* file into your Add-Ins directory (most likely c:\program files\rockwell software\arena [version number]\Add-ins)
2. Register the dll by running regsvr32.exe on it.
 - a. From the start menu, click "Run"
 - b. Type "cmd" and hit enter
 - c. At the command prompt type "c:"
 - d. Type "cd \windows\system32" for XP or "cd \winnt\system32" for 2000
 - e. type "regsvr32 "c:\program files\rockwell software\arena [version number]\add-ins\gridaddon.dll""

To run a model using the utility

1. Open the model you wish to run in Arena
2. Go to the *Tools* menu, then select *Arena Grid Computing Utility*
3. You will see a dialog like the one below:



4. The *Network Store* must be a shared directory on your network that all your nodes can read and write to. Enter a valid location in the edit box, or click the “...” button to browse for a folder. This location does not need to be changed for every model you wish to run with the utility. The utility will create a unique subdirectory for each model and replication run, and will delete that subdirectory when finished.
5. The list at the bottom of the dialog lists all the machines you want to run replications on. To add a new node to the list, click “Add Node” and then enter the machine name, and click “OK”. You may enter any valid network address (for example, the IP address of the machine, or the machine name). To remove a node from the list, click the node’s name in the list box, and then click “Remove Node”. See the *Remarks* section below for more information about specifying the local machine name and machine names twice.
6. When you are finished adjusting the list of nodes to run the simulation on, and have provided a valid location for the *Network Store*, click “Start” to start the run.
7. The dialog will provide feedback on the status of the run as it executes, it will display how many replications have finished, as well as the status of each node (what replication they are executing, if they have run into any errors, etc..)
8. To stop the run prematurely (because some error occurred, or the run is taking too long, or for some other reason). Click the “Stop” button. This will tell all nodes running Arena to stop their runs, and when they have all stopped, will return control of the dialog to you.

9. When the run has finished, the progress bar will show all replications complete, and the *Status* field will say “Run complete”
10. You can now close the dialog by clicking the “Close” button. Your settings for the *Network Store* and list of nodes will be saved to the registry, so you don’t have to re-enter them every time.
11. Back in Arena, go to the reports, and select the report you wish to see. The databases from each run were merged into a single database for the local model, and the reports should reflect the results for each replication.

Remarks

You may enter the local machine in the list of nodes to run the replication (either the IP address of the machine, “localhost”, or “127.0.0.1”). The utility will simply take over the instance of Arena you have running and run the replication.

The utility does not check for a user adding multiple machines using different identifiers. For example if server002 and 192.168.10.12 point to the same physical machine, the utility will not know this, and will run into problems when running, as only one Arena instance can be started and running on a machine at a time.

The utility does not close the Arena application on a remote machine once it has started it.

Special notes on DCOM setup

This utility works by using Arena’s automation interface, and thus uses DCOM (Distributed COM) to work with Arena instances on other machines across the network.

It is recommended you first try the utility without making any other changes to the machines on your network, and then try the suggestions below. You should try a very small example model first, before trying to run the utility on your production models.

Because of recent changes made to Windows XP, Service Pack 2, there are some general guidelines that need to be followed to get Arena to work with this utility on remote machines. These guidelines should also be noted even if you are running the utility across machines with Windows XP or 2000 as well.

1. It is highly recommended you run the utility as a user under the domain. Trying to run the utility as a local user can cause problems, and might possibly make the utility not work at all. For easier setup, it *may* be better for this user to have administrative privileges on each node.
2. You *may* need to modify DCOM settings, especially for Arena, on each machine you want to use as a node for the utility. To do this you can use the *dcomcnfg* utility. Go to Start->Run, and type “dcomcnfg”, then hit enter. Expand Component Services\Computers\My Computer\DCOM Config\. Find the entry in the tree called “Arena” and right click on it and select “Properties”. Here you can

- modify security settings for Arena in the security tab. You want to make sure whatever user you use to run the utility has Remote Launch and Remote Activation permissions under Launch and Activation Permissions, and has Remote Access permission under Access Permissions. You may also have to go to the properties of “My Computer” (right click->Properties), and change the security limits under the “COM Security” tab-> “Edit Limits”, such as allowing “Everyone” the “Remove Launch” and “Remove Activation” permission (These limits are only applicable under XP SP2).
3. To run Arena using the utility with no one logged into each node, you *may* need to go to dcomcnfg and under Arena’s properties, select the Identity tab, and then specify a specific user.

Technical Details

This utility was written in C++, using Microsoft Visual C++ 6.0.

To allow the utility to build on a machine, the following #define statements in stdafx.h may need to be modified:

1. ARENA_ADDON_IFC_TLB_LOCATION
2. ARENA_EXE_LOCATION
3. ADO_LOCATION

The general flow of the program when executing a run is as follows:

- 1.) Start the node controller thread
- 2.) The node controller thread
 - a. Creates new directories under the network store, using unique names for each comprising of the model name, current time, and replication number
 - b. The model .doe file is copied into each directory
 - c. While there are replications left to run and the user hasn’t hit “Stop”
 - i. Find a node that is free
 - ii. If we find a free node, start it running a replication, otherwise set the *wait for nodes to finish running* bit
 - d. If there are no more replications left to run (i.e. they are all currently running), or the *wait for nodes to finish running* bit is on, then wait for the nodes to finish running (wait on their threads)
 - e. When all nodes have finished running, then merge all the databases together, into the local model’s database
 - f. Finally, delete all the directories (and their contents) that were created

As noted above the program uses several threads in its execution:

- The UI Thread
 - The controller thread (dispatches replications to nodes)
 - Node thread 1 (actually starts Arena, and sends the open model and run commands)

- Node thread 2
- ...
- Node thread N

There are 3 main things that make this utility possible:

- 1.) Automation of Arena across the network via DCOM
- 2.) Setting the starting replication number via the Model.PreviousReplications property
- 3.) Merging the resulting databases together

To merge the model databases together the CArenaGridDatabaseMerger object is used. The bottom tables are merged first, those with common definitions for various objects (like data types and categories). Higher level tables are then merged, using a mapping object that maps id's from one database into id's of another when an object exists in both, but has a different id in each.