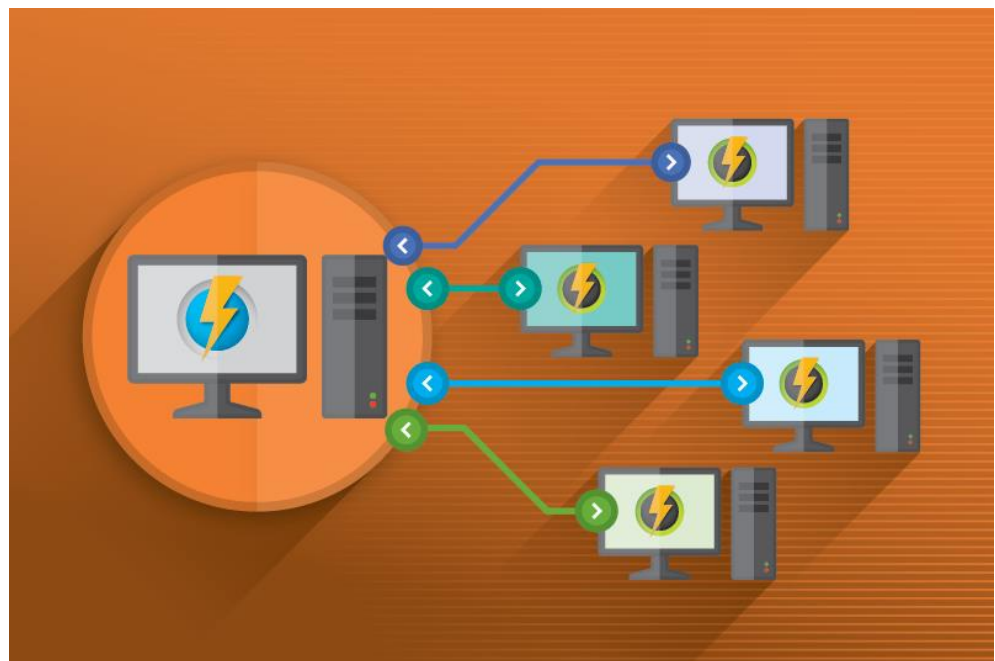


# GoldSim Distributed Processing Module





Copyright GoldSim Technology Group LLC, 2001-2018. All rights reserved.  
GoldSim is a registered trademark of GoldSim Technology Group LLC.

Version 12.1 (June 2018)



GoldSim Technology Group  
22500 SE 64<sup>th</sup> Place, Suite 240  
Issaquah, Washington 98027 USA

Visit us at our web site: [www.goldsim.com](http://www.goldsim.com)  
Email us at: [software@goldsim.com](mailto:software@goldsim.com)



---

# Contents

<b>Chapter 1: Introduction</b>	<b>1</b>
<b>Chapter Overview</b> .....	<b>1</b>
In this Chapter .....	1
<b>What is the Distributed Processing Module?</b> .....	<b>2</b>
<b>Which Version (DP or DP Plus) Do I Have?</b> .....	<b>2</b>
<b>Conventions Used in this Manual</b> .....	<b>3</b>
<b>Using Help</b> .....	<b>4</b>
<b>Technical Support, User Resources and Software Upgrades</b> .....	<b>4</b>
GoldSim Maintenance Program .....	4
Getting Technical Support .....	4
Other GoldSim Resources .....	5
<b>Chapter 2: Using the Distributed Processing Module</b>	<b>7</b>
<b>Chapter Overview</b> .....	<b>7</b>
In this Chapter .....	7
<b>Using the Distributed Processing Module</b> .....	<b>8</b>
<b>Overview of the Distributed Processing Module</b> .....	<b>8</b>
<b>Carrying out a Distributed Processing Simulation</b> .....	<b>8</b>
Running GoldSim in Slave Mode .....	9
Using the Slave Manager to Automate Launching of Slave Processes .....	14
Connecting the Master to the Slaves .....	17
Running a Distributed Processing Simulation.....	21
<b>Limitations of the Distributed Processing Module</b> .....	<b>28</b>
<b>Glossary of Terms</b>	<b>29</b>
<b>Index</b>	<b>31</b>



---

# Chapter 1: Introduction

Two heads are better than one.

Anonymous

## Chapter Overview

GoldSim is a user-friendly, highly graphical, object-oriented program for carrying out dynamic, probabilistic simulations to support management and decision-making in engineering, science and business.

The GoldSim Distributed Processing Module is a program extension to GoldSim which allows you to use multiple computers connected over a network to share the computational burden of a Monte Carlo simulation.

Each of the GoldSim modules has a separate User's Guide describing its capabilities and features. This short document provides a complete description of the features and use of the GoldSim Distributed Processing Module.



**Note:** This document only describes the Distributed Processing Module, and assumes that you are familiar with the basic capabilities of GoldSim (described in the **GoldSim User's Guide**).

---

### In this Chapter

This introductory chapter discusses the following topics:

- What is the Distributed Processing Module?
- Which Version (DP or DP Plus) Do I Have?
- Conventions Used in this Document
- Using Help
- Getting Technical Support

## What is the Distributed Processing Module?

GoldSim was designed to provide a highly efficient simulation framework for use on a single Windows personal computer. For most applications, running your GoldSim application on a single machine will be sufficient. For highly complex models, however, running Monte Carlo simulations on a single machine could become impractical. For example, if each realization took one minute to complete, then 1000 realizations would take over 16 hours.

The Distributed Processing Module allows you to overcome this problem by combining the power of multiple cores and/or computers to carry out a simulation involving multiple realizations.

This is accomplished by having a **Master** GoldSim executable connect to an unlimited number of **Slave** GoldSim executables. These Slave processes can be on a single computer or on computers connected over a network. The Master assigns and sends individual realizations to the Slave processes. Each Slave carries out its assigned realization, sends the results to the Master, and then makes itself available to work on another realization. Once all of the realizations have been completed, the Master process combines all of the results into a single GoldSim file (and enters Result Mode) just as if the entire simulation had been carried out on the Master's machine.

Although the Master and Slave processes will typically be connected via a Local Area Network (LAN), they can also communicate via a Wide Area Network (WAN) or even the Internet. In general, the speed of the simulation is proportional to the number of available Slave processes.

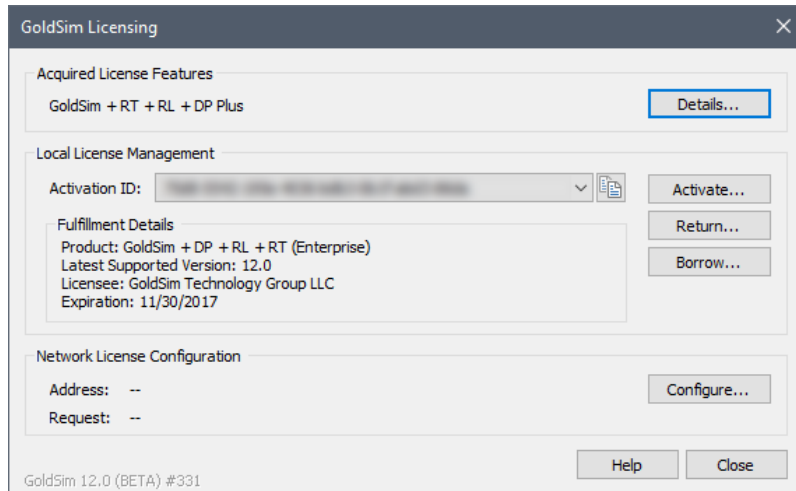
There are two versions of the Distributed Processing Module. **GoldSim DP** comes bundled with all versions of GoldSim, and limits the number of Slave processes to four. **GoldSim DP Plus** allows an unlimited number of Slave processes, and must be licensed separately.

## Which Version (DP or DP Plus) Do I Have?

As pointed out above, there are two different versions of the Distributed Processing Module, and your copy of GoldSim will have one or the other. You can determine which version of the Distributed Processing Module is activated on your machine by selecting **Help|Licensing** from the main menu.



At the top of the dialog, the "Acquired License Features" section will list either "DP" or "DP Plus":





## Conventions Used in this Manual

The following conventions are used in this manual:

Convention	Description
<b><i>Important Terms</i></b>	New and important terms are presented in <i>bold italics</i> . These terms all appear in the Glossary of Terms at the end of the document.
<b>File   Open...</b>	Menus and menu selections are separated by a vertical bar. <b>File   Open...</b> means "Access the File menu and choose Open"
CTRL+C	Key combinations are shown using a "+" sign.. CTRL+C means press the Control and C keys simultaneously.
	<b>Warning:</b> This means watch out! Warnings typically alert you to potential pitfalls and problems that may occur if you perform (or fail to perform) a certain action.
	<b>Note:</b> Notes highlight important information about a particular concept, topic or procedure, such as limitations on how a particular feature can be used, or alternative ways of carrying out an action.

In describing various mouse actions, the following conventions are used:

Mouse Action	Definition
Click	Press and release the left mouse button once.
Double-click	Press and release the left mouse button twice in rapid succession.
Right-click	Press and release the right mouse button once.
Drag	Press the left mouse button, and while keeping it depressed, move the cursor to another location, then release the button.

## Using Help

GoldSim has an extensive in-product help facility, which can be used to supplement this manual. The Help system can be accessed by selecting **Help | Help Topics** from the main GoldSim menu or the Help button (the question mark) on the standard toolbar. Pressing the **Help** button within any of the dialogs also provides access to Help (in a context-sensitive manner). All of the information in this document is accessible via GoldSim Help.

## Technical Support, User Resources and Software Upgrades

The GoldSim Technology Group is dedicated to providing complete solutions for our customers. We pride ourselves in providing prompt and extensive support and resources to our users, and are committed to ensuring that each installation of our software is successful and adds value to the customer.

### GoldSim Maintenance Program

When you purchase GoldSim software, you receive one year of Software Maintenance, entitling you to the following:

- Free software upgrades so that you always have the latest version of the GoldSim software.
- Basic Technical Support via email and phone. Basic support covers installation and licensing questions, as well as questions about GoldSim's features and capabilities.

After the first year, if you wish to continue to have access to new versions and technical support, Software Maintenance can be extended each year with payment of an annual fee.

Details regarding the GoldSim Maintenance Program can be found at [www.goldsim.com/Web/Products/BuyGoldSim/Pricing/MaintenanceProgram/](http://www.goldsim.com/Web/Products/BuyGoldSim/Pricing/MaintenanceProgram/).

### Getting Technical Support

Users with active Software Maintenance can submit questions directly to the GoldSim support team. Evaluation users are also welcome to contact us with questions on GoldSim functionality. The **GoldSim Help Center** (<https://goldsim.zendesk.com>) is the primary portal for technical support. You can submit your questions directly from the Help Center. If you register and log in through the Help Center, you will be able to check the status and view a history of all of your support requests.

The Help Center also includes:

- The **GoldSim Forum**, where you can post questions to the GoldSim community, or just browse existing messages;
- Articles on licensing questions and modeling tips; and
- An archive of past webinars (which demonstrate GoldSim features and capabilities).

Although using the Help Center is the preferred way to access support, you can also contact us by phone at 1-425-295-6985 between the hours of 8AM and 5PM Pacific Time.

Free Basic Technical Support does not include consulting, model troubleshooting or detailed assistance with applying GoldSim to a particular problem. Assistance of this nature is defined as Advanced Technical Support. Users may purchase Advanced Technical Support in pre-paid 10 hour blocks.

## Other GoldSim Resources

Details regarding Advanced Technical Support can be found at [www.goldsim.com/Web/Resources/TechnicalSupport/](http://www.goldsim.com/Web/Resources/TechnicalSupport/).

In addition to the GoldSim Help Center, additional resources are also available. These three resources can be accessed directly from the GoldSim website ([www.goldsim.com](http://www.goldsim.com)):

- A free **Online Training Course** that will provide you with a thorough understanding of the key concepts on which GoldSim is based and all of the fundamentals required to build complex models of nearly any kind of system.
- The **GoldSim Model Library**, which contains a collection of example models to allow you to see how specific features of GoldSim can be used and/or how GoldSim can be used for specific applications.
- The **GoldSim Blog**, which provides an informal mechanism for GoldSim staff to share their knowledge, point out some of the more advanced (and perhaps overlooked) GoldSim features, share and discuss common mistakes we see in GoldSim applications, discuss interesting applications, and keep you abreast of our plans for further GoldSim developments.

You can stay up to date on the latest GoldSim news through these resources:

- The GoldSim LinkedIn Group, which is primarily used for announcements (e.g., new versions, interesting applications). You can join the Group here: [www.linkedin.com/groups/1798413](http://www.linkedin.com/groups/1798413)
- Periodic email newsletters are sent two to three times per year. To be added to the newsletter list, contact us via the GoldSim Help Center (<https://goldsim.zendesk.com>).



**Note:** When you purchase GoldSim, you are entitled to a free one hour, live web-based training session in which one of our analysts provides an interactive training session via the Internet and telephone. You are strongly encouraged to take advantage of this free training.

---



---

# Chapter 2: Using the Distributed Processing Module

As I would not be a slave, so I would not be a master. This expresses my idea of democracy.

Abraham Lincoln

## Chapter Overview

This chapter provides the details about using the GoldSim Distributed Processing Module. After first providing an overview of how a distributed processing simulation is carried out, step-by-step procedures for carrying out such a simulation are described. The chapter closes with a discussion on the limitations of the Distributing Processing Module.

### In this Chapter

This chapter discusses the following topics:

- Using the Distributed Processing Module
- Overview of the Distributed Processing Module
- Carrying Out a Distributed Processing Simulation
- Limitations of the Distributed Processing Module

## Using the Distributed Processing Module

The topics below provide the details about using the GoldSim Distributed Processing Module. It is recommended that you first read the overview of how a distributed processing simulation is carried out. Step-by-step procedures for carrying out such a simulation can then be examined.

### Overview of the Distributed Processing Module

The Distributed Processing Module uses multiple copies of GoldSim running on multiple machines (and/or multiple processors or cores within a single machine) to share the computational burden of a Monte Carlo simulation.

A distributed processing simulation is controlled by a single *Master* GoldSim executable. The Master executable is a version of GoldSim running on one machine which has the Distributed Processing Module enabled. Through a dialog accessed via a menu item, the Master can create connections with multiple *Slave* GoldSim processes.

The Master assigns and sends individual realizations to the Slave processes. Each Slave process carries out its assigned realization, sends the results to the Master, and then makes itself available to work on another realization. Once all of the realizations have been completed, the Master combines all of the results into a single GoldSim file (and enters Result Mode) just as if the entire simulation had been carried out on the Master's machine.

The GoldSim model file (and other required files such as user-created DLLs or linked spreadsheets) need not be present on machines running Slave processes. The Master automatically transfers any necessary files to the Slave machines.

When running as a Slave process, GoldSim is run in a special mode. Rather than launching GoldSim in the traditional manner on the machine running the Slave, a GoldSim Slave is launched from the command line (this can be done using a Windows shortcut or the Run option on the Windows Start menu). Alternatively, the process of starting Slaves can be automated by installing the GoldSim Slave Manager on machines which will act as Slaves. When running in Slave mode, you cannot view or edit a GoldSim model. The only function that a Slave serves is to receive, process, and send back realizations sent to it by a Master.

In order for the Master process to access the Slave processes, they must be local to the computer or accessible over a network.

### Carrying out a Distributed Processing Simulation

The steps required to carry out a distributed processing simulation are as follows:

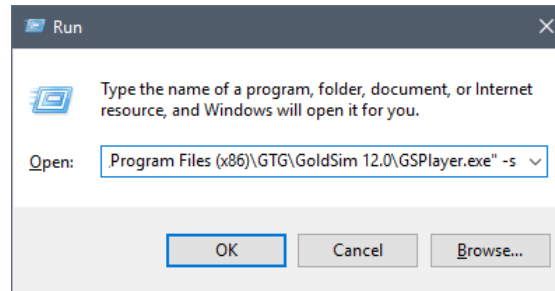
1. Launch one or more GoldSim processes on the machine(s) that you wish to utilize (or install the GoldSim Slave Manager on the Slave machines).
2. Open the GoldSim model file in a version of GoldSim with the Distributed Processing Module enabled (which will act as the Master), and connect the Master to the Slave processes.

## 3. Run the simulation.

Each of these three steps is discussed in detail below.

## Running GoldSim in Slave Mode

In order to start a GoldSim Slave process on a machine, GoldSim (or the GoldSim Player) must be installed. The GoldSim Player is automatically installed when GoldSim is installed, and is also available as a free download. To launch the GoldSim Player in Slave mode, you must run the Player from the command line using the “-s” parameter. Typically, this would be done by accessing the Windows Run option, which displays the following dialog:

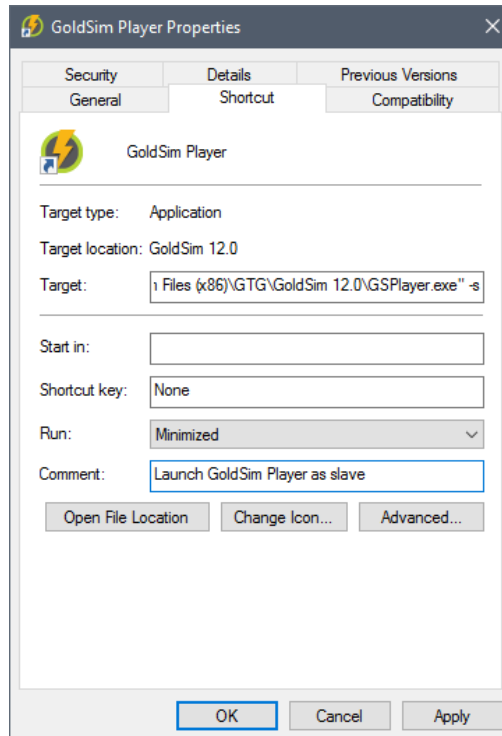


You must enter the full path to the GoldSim Player executable (which may, of course, be different from that shown above), followed by the “-s” command line parameter.



**Note:** In addition to using the GoldSim Player as a Slave, any GoldSim executable (even if it is unregistered) can also be accessed as a Slave. In this case, of course, the command line to run the Slave would reference the GoldSim executable (GoldSim.exe) rather than the Player executable.

You could accomplish the same thing by creating and running a shortcut to the GoldSim Player executable with the appropriate command line parameter:



*In this particular case, the GoldSim Player would be launched in Slave mode and immediately minimized.*

If you placed this shortcut in the StartUp directory of the machine, the GoldSim Player would be loaded in Slave mode whenever the machine was booted.



**Note:** When creating the shortcut, the pathname to the Player file must be enclosed in quotes, with the `-s` option (to start the Player in slave mode) following the pathname. For example, if the path to the GoldSim Player was `C:\Program Files (x86)\GTG\GoldSim 12.0\GSPlayer.exe`, the Target field of the shortcut dialog should be `"C:\Program Files (x86)\GTG\GoldSim 12.0\GSPlayer.exe" -s`.

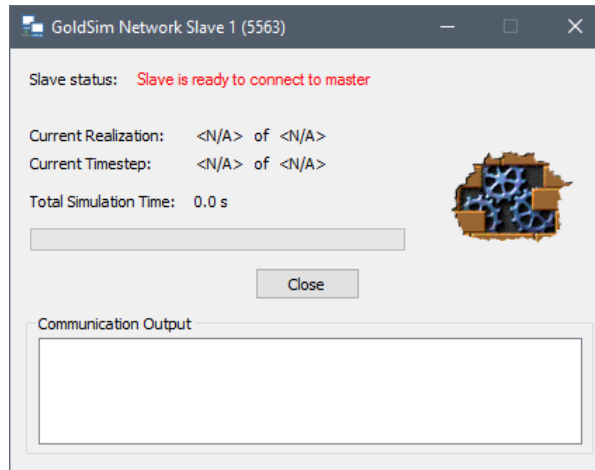
A Slave process requires approximately 6Mb of memory when it is loaded prior to being accessed by the Master process. Hence, for most machines, the act of loading the Slave process will not significantly impact anyone who is using that particular machine. Moreover, once a Master process accesses the Slave (and begins to send it realizations to process), the priority of the Slave is lower than that of other processes running on the machine. As a result, although an active Slave will slow the system down somewhat, the machine can still be used for other applications, with the GoldSim Slave process running in the background.



**Note:** You can run multiple Slave processes on a multi-core or multi-processor computer to increase performance. For a multi-processor computer, the recommended number of GoldSim processes (Master and Slave) is equal to the number of processors. For a multi-core computer, the recommended number of GoldSim processes (Master and Slave) is equal to the total number of cores minus one. You must launch each process separately (such that each process is present in the Windows Task Bar).



If you launch a Slave process from the command line, the following dialog will be displayed on the Slave machine:



Pressing the **Close** button will terminate the Slave process (close the GoldSim Slave executable) so that it can no longer be accessed by a Master. A Slave can also be terminated remotely by the Master.

**Read more:** [Terminating Slaves from the Master Process](#) (page 21).



**Note:** This dialog is only displayed for Slave processes that are manually started. Slave processes started automatically (using the GoldSim Slave Manager) will not display the dialog. These automatically-started Slave processes can be terminated from the Master process or ended using Task Manager.

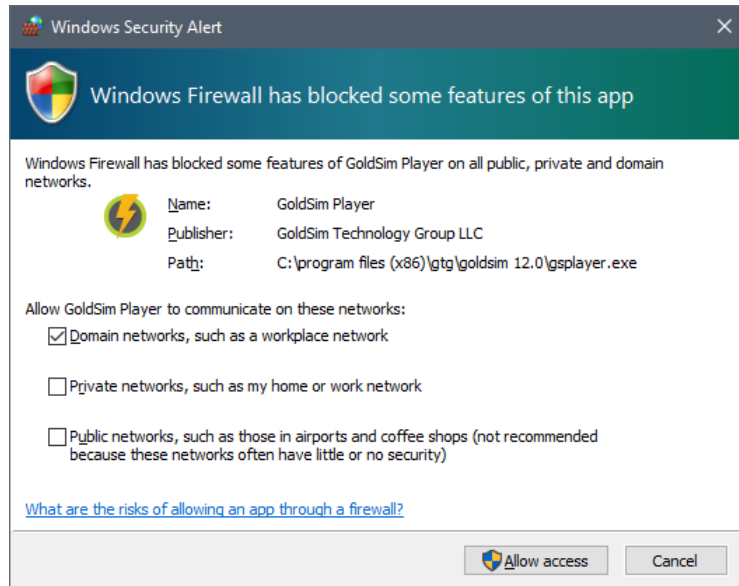
### **Configuring Windows Firewall to Permit Connections to Slave Processes**

**Read more:** [Using the Slave Manager to Automate Launching of Slave Processes](#) (page 14).

If a machine has the Windows Firewall enabled, GoldSim or the GoldSim Player must be added to the Windows Firewall's Exception List in order to manually start Slave Processes.

The first time that GoldSim or the GoldSim Player is started as a Slave process manually, the Windows Firewall will ask if you wish to unblock the program. The dialog for this differs slightly depending on the version of Windows you are running.

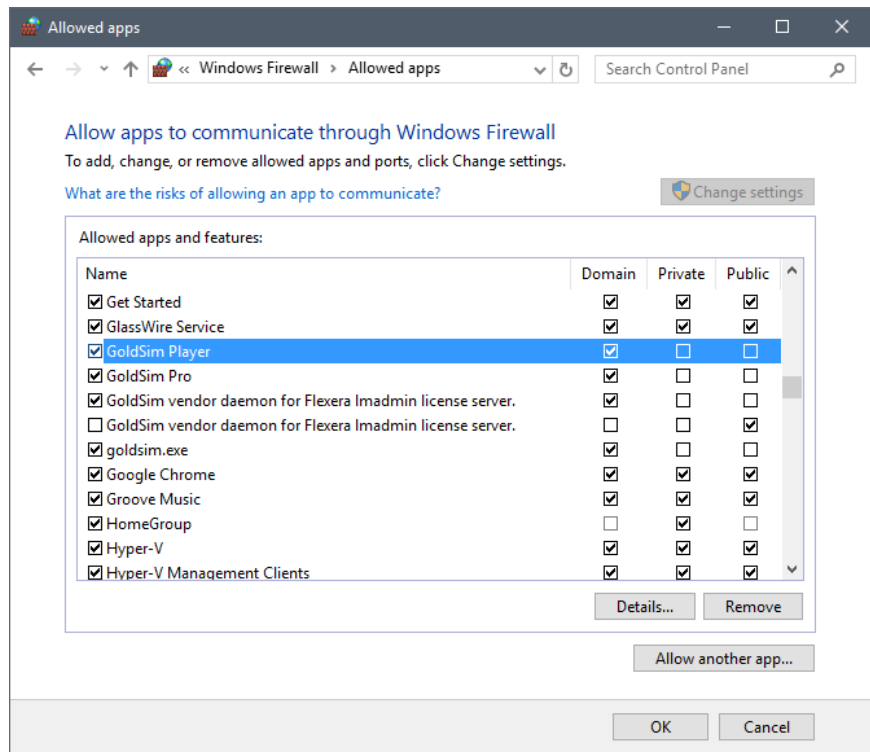
In Windows 10 and related versions you will see the following (the dialog is similar in Windows 7, 8 and 8.1):



For security purposes, you should ensure that only the Domain Networks option is checked. Then click the **Allow Access** button.

If you were not prompted to unblock the program, it may already have been added to the Windows Firewall Exception List.

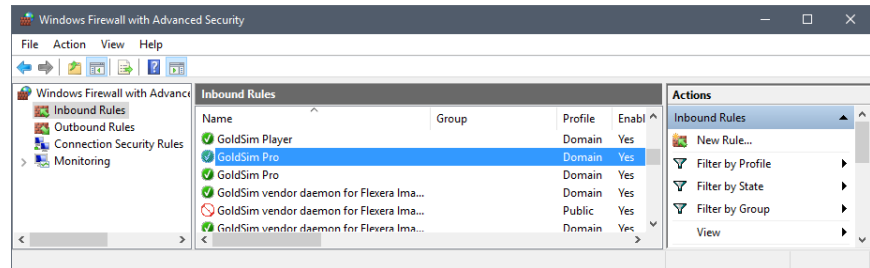
In Windows 10, you can check that the program has been correctly added by opening the Control Panel, clicking on Windows Firewall and then clicking (in the upper left) “Allow an app or feature through Windows Firewall”.



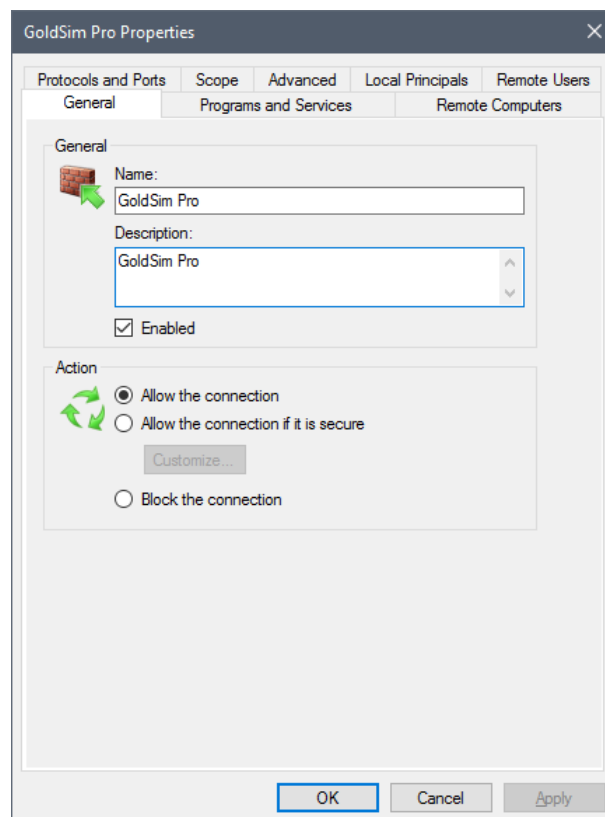
The procedure is similar for Windows 7, 8 and 8.1.

GoldSim or the GoldSim Player will be listed. If the box to the left is checked, the program is already unblocked. If it is unchecked, simply check the box to unblock the program.

You can check the program has been correctly added by clicking the Start button on the Desktop and then typing Windows Firewall into the Search field. Select *Windows Firewall with Advanced Security* from the Search results (note, to get this to appear in the search results, it may be necessary to type more than *Windows Firewall*). When the program opens, select Inbound Rules on the left-hand side. You'll see the following dialog:



GoldSim Pro and/or the GoldSim Player should show Yes under the Enabled column. If they are not enabled, double click the entry and check the Enabled box on the General tab of the dialog that appears:



**Note:** If you are automatically starting Slaves (using the GoldSim Slave Manager) there is no need to add GoldSim or the GoldSim Player to the Exceptions list. However, the appropriate ports must be added to the Exceptions List to permit automatically starting the Slaves.

## Using the Slave Manager to Automate Launching of Slave Processes

**Read more:** [Configuring Windows Firewall to Permit Use of the GoldSim Slave Manager](#) (page 15).

In addition to manually starting a GoldSim Slave process on a machine, you can also use the GoldSim Slave Manager to automatically launch Slave processes.

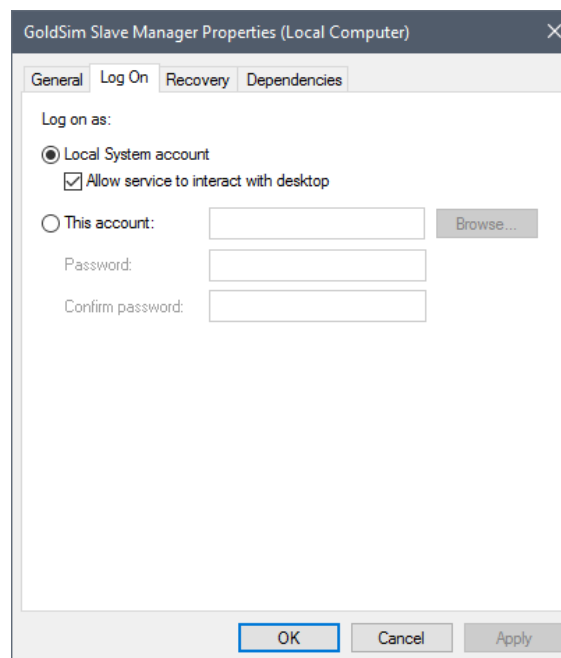
The GoldSim Slave Manager is a Windows Service that can be installed on a computer that permits a Master GoldSim process to remotely launch GoldSim in Slave mode.



**Note:** The Slave Manager is available as a separate installation file on the Downloads page of the GoldSim website.

When a Master GoldSim executable attempts to connect to a machine that has the GoldSim Slave Manager installed, the Slave Manager automatically launches the requested number of Slave processes. Once the firewalls of the machines have been properly configured and the Slave Manager installed, all actions (such as launching Slave processes, distributing realizations, and terminating Slave processes) can be controlled from the Master process.

Automatically-started Slave processes can only display the Slave dialog through the Interactive Services Detection if the user makes a change in the Services dialog to allow the Slave process to interact with the desktop. To do this you must open the Services dialog (click the Start button, then type *services* and then select the Services entry from the list of results). Double-click the GoldSim Slave Manager entry. Click the **Log On** tab and ensure the “Allow service to interact with the desktop” option is selected:



Click **OK**, then right-click the GoldSim Slave Manager entry and click Restart.

If you do not go through this process, automatically-started Slave processes can only be terminated by the Master, or by ending the process using the Task Manager.

**Read more:** [Terminating Slaves from the Master Process](#) (page 21).

## Configuring Windows Firewall to Permit Use of the GoldSim Slave Manager

In order for the Slave Manager to successfully launch Slave processes on a machine with Windows Firewall enabled, the ports used by the GoldSim Slave Manager and Slave processes must be added to the Windows Firewall's Exception List.



**Note:** If you are automatically starting slaves, there is no need to add GoldSim or the GoldSim player to the Exception List. These only need to be added to the Exception List if Slave processes will be started manually. However, even if these have been added, the *ports* required must also be added.

GoldSim uses the following ports on the slave for distributed processing:

Service	Port Number
GoldSim Slave Manager	5562
GoldSim Slave 1	5563
GoldSim Slave 2	5564
GoldSim Slave 3	5565
GoldSim Slave 4	5566
GoldSim Slave 5	5567
GoldSim Slave 6	5568
GoldSim Slave 7	5569
GoldSim Slave 8	5570
GoldSim Slave 9	5571
GoldSim Slave 10	5572

Therefore, if you wished to start two slaves on a machine, ports 5562 through 5564 would need to be opened. For five slaves on a machine, ports 5562 through 5567 would need to be opened.

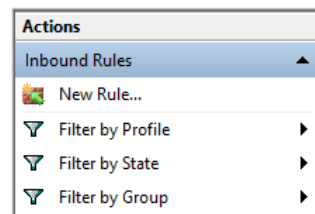


**Note:** There is no need to open ports for Slaves 5 through 10 on slave machines connected to a Master without Distributed Processing Plus (as standard Distributed Processing licenses can only support a maximum of 4 slaves).

### Opening Ports on Windows 10 and related versions:

In Windows 10 and related versions (e.g., Windows 7 and 8), to open the ports required for Distributed Processing, click the Start button and type *Windows Firewall* into the search box. Select *Windows Firewall with Advanced Security* from the search results (note, to get this to appear in the search results, it may be necessary to type more than *Windows Firewall*).

When the Windows Firewall dialog opens, click **Inbound Rules** on the left-hand side of the dialog, then click **New Rule...** in the Actions column at the right of the dialog:



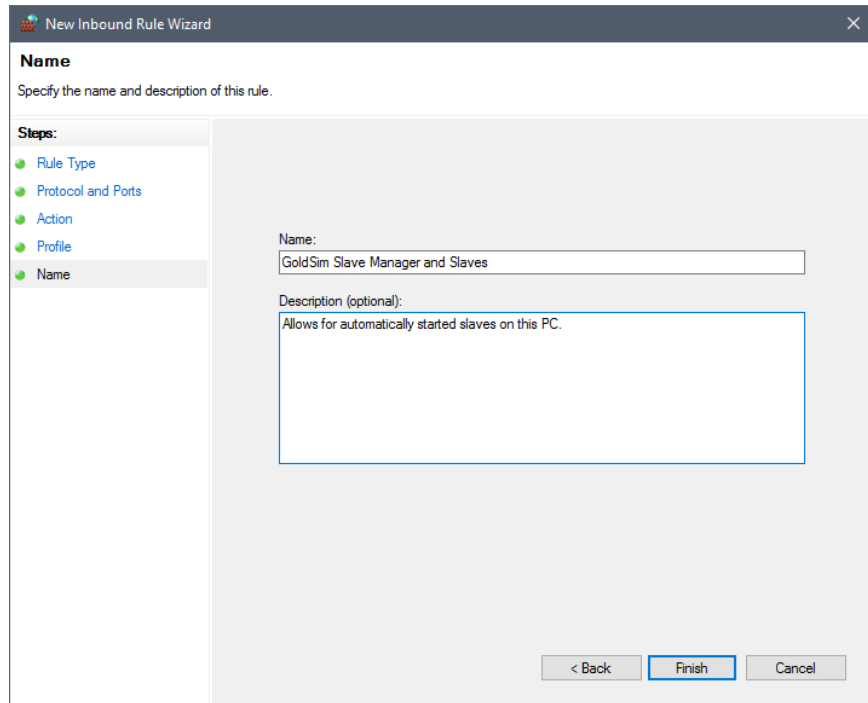
This brings up a Wizard. Select *Port* for the Rule Type and click **Next**.

The screenshot shows the 'New Inbound Rule Wizard' dialog box, specifically the 'Protocol and Ports' step. The title bar reads 'New Inbound Rule Wizard'. The main heading is 'Protocol and Ports' with the instruction 'Specify the protocols and ports to which this rule applies.' On the left, a 'Steps:' list includes 'Rule Type', 'Protocol and Ports' (highlighted), 'Action', 'Profile', and 'Name'. The main area contains two questions: 'Does this rule apply to TCP or UDP?' with radio buttons for 'TCP' (selected) and 'UDP'; and 'Does this rule apply to all local ports or specific local ports?' with radio buttons for 'All local ports' and 'Specific local ports:' (selected). Below the 'Specific local ports:' option is a text input field containing '5562-5572' and an example 'Example: 80, 443, 5000-5010'. At the bottom right are buttons for '< Back', 'Next >', and 'Cancel'.

Select *TCP* and then the *Specific local ports* option. Specify a range of ports: *5562-5566* if you have Distributed Processing and *5562-5572* if you have Distributed Processing Plus and then click **Next**. Select *Allow the connection* in the Action step of the Wizard and click **Next**.

The screenshot shows the 'New Inbound Rule Wizard' dialog box, specifically the 'Profile' step. The title bar reads 'New Inbound Rule Wizard'. The main heading is 'Profile' with the instruction 'Specify the profiles for which this rule applies.' On the left, a 'Steps:' list includes 'Rule Type', 'Protocol and Ports', 'Action', 'Profile' (highlighted), and 'Name'. The main area contains the question 'When does this rule apply?' with three checked options: 'Domain' (checked), 'Private' (unchecked), and 'Public' (unchecked). Each option has a brief description: 'Domain' applies when connected to a corporate domain, 'Private' applies when connected to a private network location, and 'Public' applies when connected to a public network location. At the bottom right are buttons for '< Back', 'Next >', and 'Cancel'.

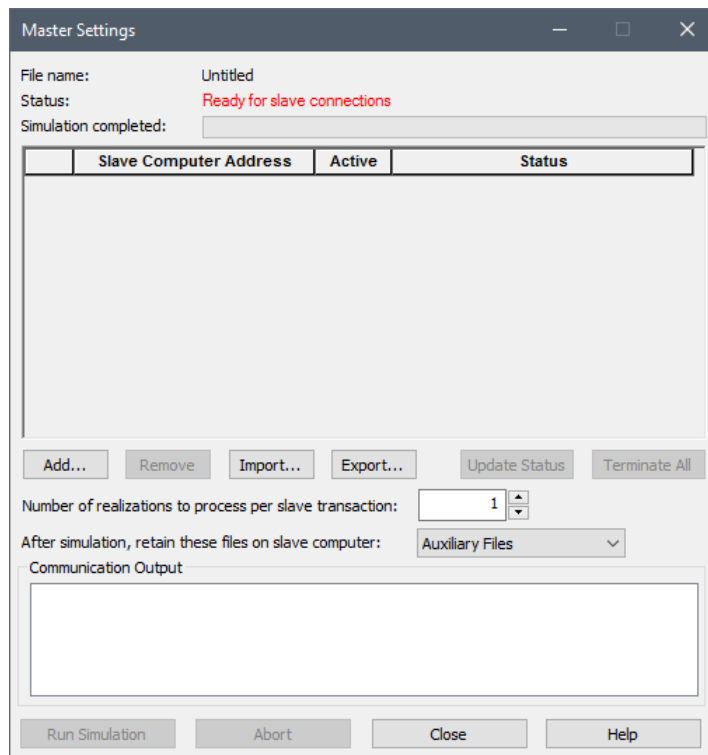
Uncheck the Public option (we also recommend leaving the Private option unchecked) in the Profile step and click **Next**.



Finally, add a meaningful name and description in the Name step and click **Finish**. You can then close the Windows Firewall.

## Connecting the Master to the Slaves

To start a distributed processing simulation from a Master GoldSim executable, start GoldSim, load the model file which you wish to run, and select **Run | Run on Network...** from the main menu. The following dialog will be displayed:





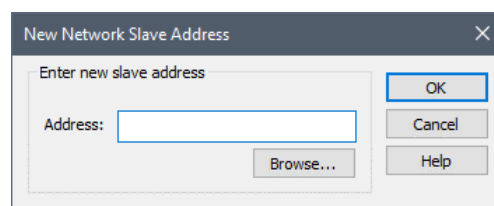
**Note:** The option to **Run | Run on Network...** is only available if multiple realizations have been specified. That is, a distributed processing simulation cannot be carried out on a single realization.

---

The next step is to specify the names of the machines that will host the Slave processes. If multiple Slave processes will be run on a single PC, the name of that PC should appear as many times as the desired number of Slave processes. You can add machine names manually, or by importing a list of Slaves that you have previously entered and saved.

**Read more:** [Importing and Exporting Slave Lists](#) (page 21).

To add Slaves manually, press the **Add...** button, which will display the following dialog:



You can use the **Browse...** button to find a computer on the network (e.g., by exploring My Network Places), or you can enter a network ID or IP address directly.

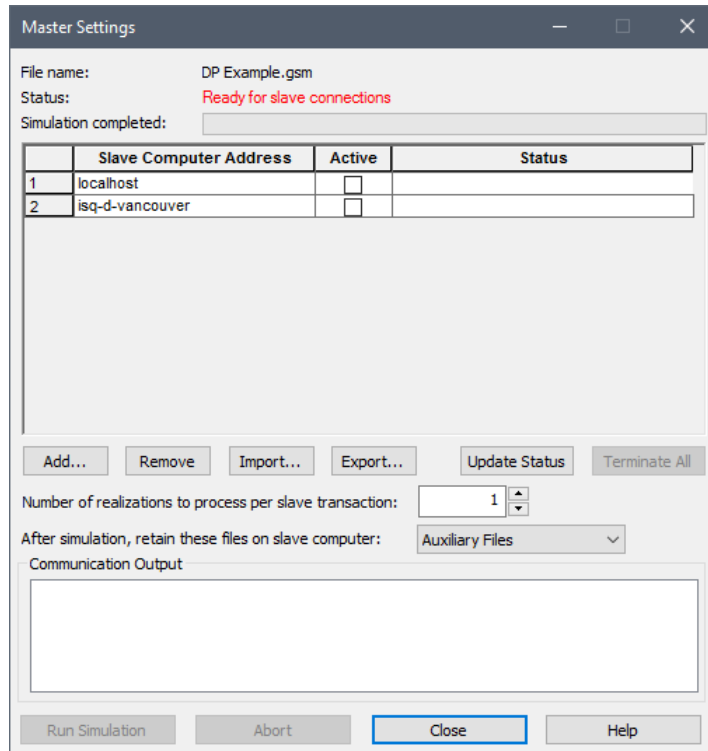


**Note:** If a Slave Process is to be run on the local machine, you can simply type *localhost* for the computer name. If you are running multiple slaves on your computer (i.e., using multiple processors) you should enter *localhost* for each slave you have launched (i.e., if you are running 6 slaves, *localhost* should be entered six times).

---

When you add a Slave and press the **OK** button, the Slave is added to the list:

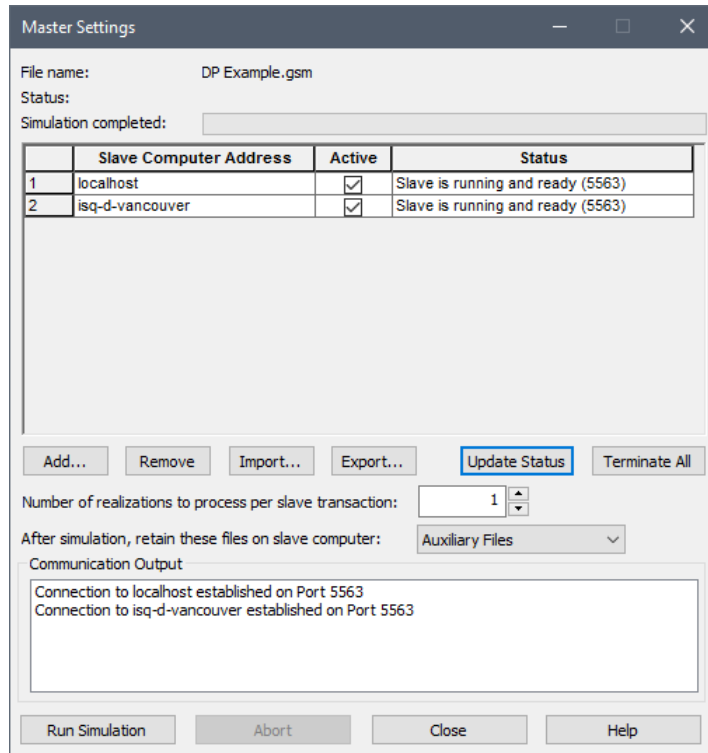




If you then press **Update Status**, the Master does the following:

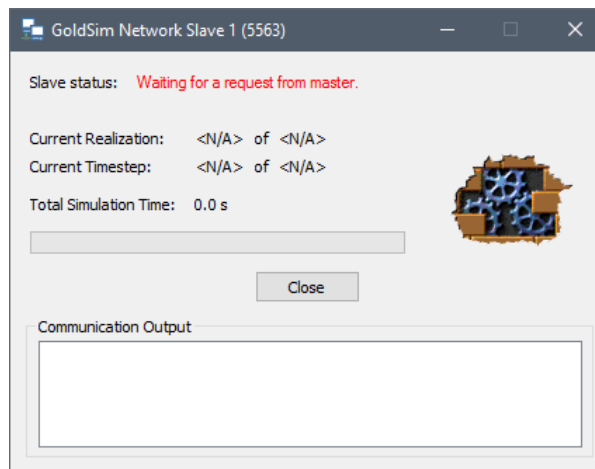
1. Finds the specified computers on the network;
2. Checks to see if a GoldSim Slave process is running and available on the computers; and if not, then
  - If the Slave Manager is running on a listed machine that has no available Slave processes, the Master will attempt to launch a new GoldSim Slave process on that machine.

GoldSim then reports the status of the Slaves in the dialog:



**Note:** Unless you have licensed the Distributed Processing Plus (DP Plus) module, you will be limited to using four slave processes at a time. If you have listed more than four slave processes, when you press the **Update Status** button, GoldSim will only activate the first four, and will notify you (in the Status column of the dialog) that the other Slave processes will be inactive.

The dialog for manually-started Slave processes indicates when a connection with a Master has been made (and that it is waiting for a request):



You can subsequently remove Slave process entries from the list in the Master Settings dialog by selecting an address and pressing the **Remove** button. If necessary, existing entries in the Slave process list can also be edited directly.

### Importing and Exporting Slave Lists

The list of computer addresses for Slave processes is saved to the Registry when you close the Master Settings dialog, so that the addresses you used the last time you ran a distributed processing run will be available the next time the Run on Network dialog is opened.

As discussed above, the list of Slave computer addresses is saved to the Registry when you close the Master Settings dialog, so that the addresses you used the last time you ran a distributed processing run will be available again the next time you wish to carry out a distributed processing run (and you will not have to reenter them).

In some cases, however, you may use different sets of slaves for different runs, or you may want to copy a list of Slaves from one computer to another. To facilitate this, GoldSim allows you to export and import lists of Slaves.

Once you have entered a list of Slaves, pressing the **Export** button prompts you to specify a filename for the list of Slaves. The file has a .SLV extension. You can subsequently import this list of Slaves by pressing the **Import** button and specifying a .SLV filename. When you do so, GoldSim will ask if you want to remove any existing Slave processes that are already in the dialog and will ask if you want to update the status of all Slave processes immediately.

### Terminating Slaves from the Master Process

You can terminate (close) a Slave process from the machine on which the Slave is running so that it can no longer be accessed by the Master.

**Read more:** [Running GoldSim in Slave Mode](#) (page 9).

You can also terminate Slaves from the Master machine. To terminate all Slaves, press the **Terminate All** button. This is typically what you would do when you complete your simulation. You can terminate an individual Slave process by clearing the **Active** button. When you do this, you will be presented with a dialog asking if you want to terminate the Slave process or simply make it inactive (i.e., stop using it, but keep it open).

**Read more:** [Activating, Deactivating and Terminating Slave Processes During a Simulation](#) (page 24).



**Warning:** Although the Master can terminate Slave processes, it cannot launch them remotely (unless the Slave machine is running the Slave Manager). On remote machines that do not have the Slave Manager installed and configured, a Slave process can only be restarted by opening the process locally on the remote machine.

### Updating the Status of Slave Processes

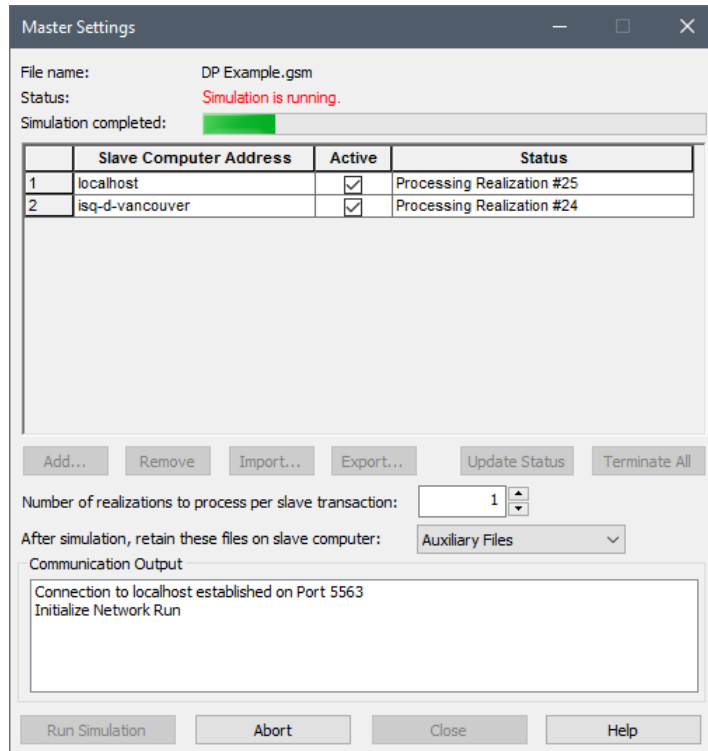
In some situations, the status of a Slave could change in such a manner that the Master would not be aware of the change (until it tried to send realizations to the Slave). This could happen, for example, if a Slave machine was turned off, or if it was rebooted prior to the beginning of the simulation.

The **Update Status** button provides a mechanism by which the Master can recheck the status of the Slaves prior to starting a simulation (in case some action needs to be taken, for example, if one of the Slave machines has been disabled). Pressing the **Update Status** button in the Master Settings dialog causes the Master to update the status of all Slaves.

### Running a Distributed Processing Simulation

Once you have connected to at least one Slave process, you can run a simulation. To do so, press the **Run Simulation** button.

During the simulation, the Master dialog will display the status of each Slave process:



The overall status of the simulation (based on the fraction of realizations that have been completed) is displayed at the top of the dialog. The status of each Slave process (i.e., the realization currently being processed) is displayed under Status.



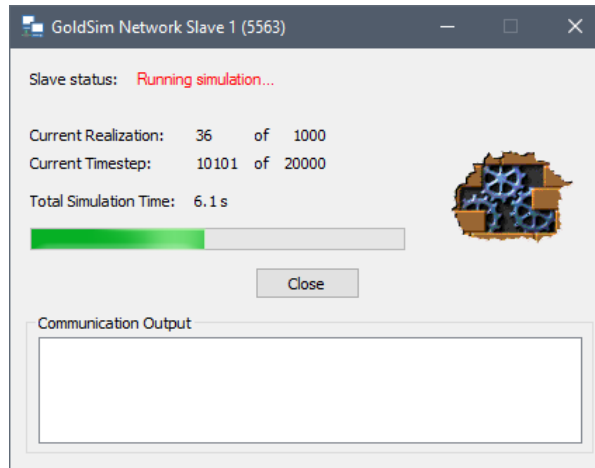
**Note:** When running a distributed processing simulation, the GoldSim Run Control Toolbar and Status bar are not used. The progress of the simulation is monitored from the Master Settings dialog rather than the main application status bar.

When a distributed processing simulation is started, the Master process first transfers the model file (and any other required files) to each of the Slave processes.

**Read more:** [Distributed Processing Auxiliary Files and Working Folders](#) (page 25).

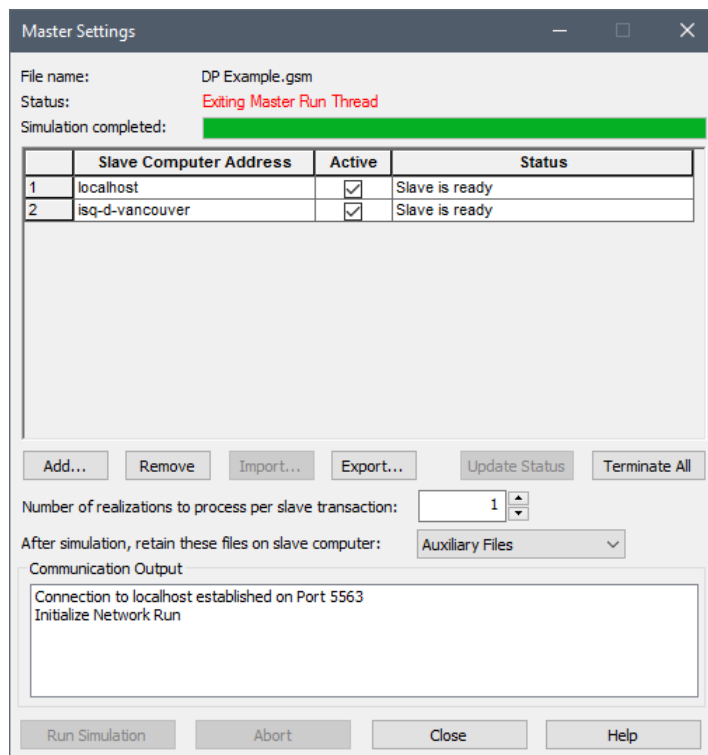
Once the Slave processes have received the required files, they can begin to receive and process realizations. The Master assigns and sends individual realizations to each Slave process, which then carries out its assigned realization, and then sends the results to the Master process. The Slave process is then assigned a new realization by the Master. This continues until the specified number of realizations have been run.

On machines with manually launched Slave processes, the Slave dialog indicates the current status of the Slave:



Once all of the realizations have been completed, the Master process combines all of the results into a single GoldSim file (and displays a “Simulation Complete” dialog) just as if the entire simulation had been carried out on the Master’s machine.

When you close the Simulation Complete dialog, the Master Settings dialog is still open, and looks like this:



To browse the GoldSim model and view the results, press the **Close** button.

### **Assigning Multiple Realizations to Each Slave Transaction**

When modeling systems with low-probability, high-consequence outcomes it is sometimes necessary to run a large number of realizations. When a realization can be run quickly the amount of time spent transferring information to Slave processes for each realization individually can negate the performance benefits of distributed processing. As a result, GoldSim has an option to assign a block of realizations to a Slave process. This option is controlled by the **Number of**

### Aborting a Distributed Processing Simulation

**Realizations to process per Slave transaction** field on the Master Settings dialog.

By default the value is 1 (indicating a single realization will be assigned each time). The value can be increased to any number that is desired (e.g. specifying 5 means that the first Slave transaction will assign realizations 1-5, the second 6-10 and so on).

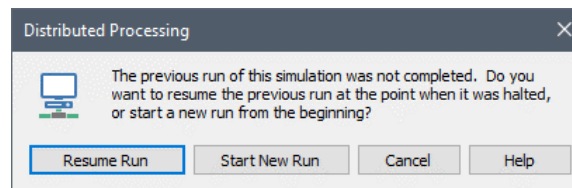
While a distributed processing simulation is running, you may choose to abort the simulation before it is complete (i.e., in order to select a new group of computers to host Slave processes). You can do this by pressing the **Abort** button in the Master Settings dialog.

When the simulation is aborted, all Slave processes will be deactivated (but left running). Completed realizations are retained. In addition, the **Run Simulation** button is renamed **Run/Resume**.

Once the simulation is aborted, the Master Settings are fully editable; the Slave process list can be edited, and the **Number of Realizations per Slave Transaction** setting can be changed.

In addition to the Help button, the **Run/Resume** and **Close** buttons are active at the bottom of the dialog.

If the **Run/Resume** button is pressed, GoldSim displays this dialog:



There are three buttons (in addition to Help):

**Resume Run.** This option makes use of realizations completed prior to the simulation being aborted, and runs the remaining realizations using the current list of Slave processes.

**Start New Run.** This option discards all completed realizations and re-runs the entire simulation.

**Cancel.** Returns to the Master Settings dialog.

If the **Close** button is pressed in the Master Settings dialog, GoldSim assembles completed realizations up to the first gap in the realization sequence, and places the model into Result Mode. For example, if realizations 1,2,3,4, 6 and 7 were completed, only realizations 1,2,3, and 4 would be assembled and available in Result Mode.

### Activating, Deactivating and Terminating Slave Processes During a Simulation

Once you have started a distributed processing simulation, you can choose to activate an inactive Slave process (e.g., if a machine becomes available) or deactivate a Slave process that is currently active (e.g., if a user requires the machine for other work, or the machine's performance is not sufficient).

You activate an inactive Slave process by simply clicking on the **Active** checkbox. The GoldSim Master will automatically try to connect to the Slave, and if successful, will begin to send it realizations.

**Read more:** [Connecting the Master to the Slaves](#) (page 17).

You can deactivate or terminate an active Slave process by clearing the **Active** checkbox. The Master will redirect the realization that the Slave process was

working on to another Slave process, and will send no additional realizations to the Slave process (unless it is subsequently reactivated).

Similarly, if a Slave process is terminated during a simulation (i.e., if the Slave process is closed, or if the machine hosting the process is turned off or rebooted), the Master will redirect the realization that that Slave process was working on to another Slave process, and will send no additional realizations to the terminated process.



**Note:** If an active Slave process is terminated (by first deactivating the slave in the Master Settings and then closing the slave dialog or turning off the slave machine) and then restarted during a simulation, the GoldSim Master will not automatically reconnect to the Slave process and start to use it again. To reestablish the connection, you must check the **Active** checkbox for that Slave process in the Master Settings dialog.

---

### **Distributed Processing Auxiliary Files and Working Folders**

When a distributed processing simulation is started, the first thing that the Master does is to transfer the model file (and any other required files, such as spreadsheets referenced by Spreadsheet elements, DLLs referenced by External elements, and all File elements) to each of the Slave processes.

The files are copied to a working folder on the machine hosting the Slave process. The location of the working folder varies depending on the operating system. The best way to access the folder is to use variables provided by the operating system. Enter the following directly into the address bar for Windows Explorer:

%ALLUSERSPROFILE%\GTG

Within this folder will be a working folder called "Slaveiiii", where iiii is the port number. (If you run multiple Slave processes on a single system, each port will have its own working folder). An index file (index.gst) is generated by GoldSim inside each working folder. This file stores information about any auxiliary files that have been saved in the folder.



**Note:** This is the default location for the folder. You can manually change this by editing the registry and creating a key called GoldSimDataFolder with the folder location under  
HKEY\_CURRENT\_USER\Software\GTG\GoldSim\DistributedProcessing.

---

If the GoldSim model uses External elements or External Pathway elements, copies of their DLL files will be created in the Slave's local working folder.

If the GoldSim model uses Spreadsheet elements, then:

- If the associated .XLS file has a full pathname (e.g., "C:\test\file.xls") then it will be copied to the same path on the Slave system under the working folder created by GoldSim (e.g., Slaveiiii\test\file.xls).
- If the associated .XLS file has a local pathname (e.g., "testfile.xls") then it will be copied into the Slave's working folder.
- If the associated .XLS file has a relative pathname (e.g., ".\xyz\testfile.xls") then it will be copied to a path on the Slave system under the working folder created by GoldSim (e.g., Slaveiiii\xyz\test\file.xls).



**Warning:** When doing distributed processing simulations, any referenced files (e.g., spreadsheets and DLLs) must be stored in the same folder as the GoldSim model file (or subfolders within that same folder). Relative paths to folders outside of the model folder should not be used.

---

If you use File elements, the same rules discussed above for .XLS files apply to the File Element's local file name.

You can control whether or not the files in the working folders are saved at the end of the simulation. This is determined by the Master Settings (run) dialog using the drop-box **After simulation, retain these files on client computer**.

There are three options:

**None:** All files generated by the Slave process (as well as the working folder itself) are deleted.

**Auxiliary Files (default):** Any auxiliary files transferred to the Slave process (e.g., DLLs, spreadsheets) are saved, but any intermediate files that might have been generated independently by the Slave process (e.g., via DLLs) are not saved.

**All Files:** Any auxiliary files transferred to the Slave process (e.g., DLLs, spreadsheets) are saved, *and* any intermediate files that might have been generated independently by the Slave process (e.g., via DLLs) are also saved.

---



**Note:** The gsm file itself is never saved in the working folder.



**Note:** If a Slave machine is turned off during a simulation, its working folders are not deleted.

---

While a distributed processing simulation is running, results sent back to the Master are stored in temporary result files in the same directory as the model file. One file is created for each realization (or set of realizations), and named ModelFilename-RealX.gsr, where ModelFilename is the name of the model file, and X is the realization number (if a set of multiple realizations are saved, the filename becomes ModelFilename-RealX-Y.gsr, where X is the first realization in the set and Y is the last realization in the set). By default, these files are deleted when the Master assembles all of the results into the model file.

---



**Note:** In some cases it is necessary to retain the temporary results files (e.g., to allow for recovery if the Master process crashes). GoldSim provides a mechanism to do this by modifying a registry setting.

---

**Read more:** [Recovering a Simulation When the Master Process Fails](#) (page 27).

### **Assigning Realizations to Multiple Slaves at the End of the Simulation**

In some cases, when running a simulation with multiple realizations, it is possible that one or more of the realizations may “stall” or run very slowly for unanticipated reasons (e.g., due to hardware issues on the Slave machine, or just because that particular Slave is slow). In this situation, it is possible for all of the realizations except a small number to be complete. As a result, the



simulation may take an extremely long time to finish (as it is waiting for the last few realizations to complete on the “slow” Slaves).

To address this situation, toward the end of a simulation, if one or more Slaves have no realizations assigned to them, you can instruct GoldSim to assign any incomplete realizations to the unused Slaves. As a result, a realization can be assigned to multiple Slaves. As soon as one of the Slaves completes the realization, the other Slaves working on the same realization are told to stop.

You can enable this behavior so that unfinished realizations are assigned to idle slaves by creating a registry DWORD called EnableReassignRealization in HKEY\_CURRENT\_USER\Software\GTG\GoldSim\Settings\ and setting it equal to a nonzero value. A value of 0 (or the absence of the the key) disables the behavior.



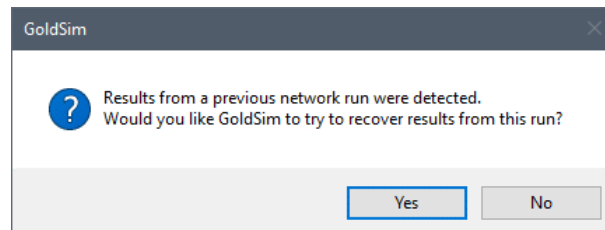
**Note:** This key is only meaningful if defined on the computer acting as DP master.

### Recovering a Simulation When the Master Process Fails

Occasionally there will be situations where the computer hosting the Master process crashes. If temporary result files are retained on the computer completed realizations can be recovered.

By default the temporary result files are processed (and deleted) as the simulation is running (to conserve disk space). To retain these files until the completion of the simulation, create a DWORD called DelayDeleteGSR in HKEY\_CURRENT\_USER\Software\GTG\GoldSim\Settings\ and set it equal to a value of 1.

When GoldSim is restarted and the model file that was being worked on is reopened, GoldSim will search for these temporary files when the **Run|Run on Network** option is selected. The following dialog will be displayed:



Pressing **Yes** will allow use of the completed realizations from the run where the Master process crashed. When you enter the Master Settings dialog the Run Simulation button will be labeled **Run/Resume**. Clicking the **Run/Resume** button and then the **Resume Previous Run** option will run the remaining realizations needed to complete the simulation.

Pressing **No** will erase the temporary files from the interrupted run.

### Distributed Processing Log Files

When carrying out a distributed processing simulation, GoldSim creates ASCII log files for the Slave processes. Each Slave process has a log file named "SlaveLog*iiii*.log", where *iiii* is the port number used by the Slave. This file is located one level up from the Slave's working folder (typically, C:\ProgramData\GTG).

Log files document the communication that takes place between the Master and Slave processes, and can be useful for debugging purposes.

## Limitations of the Distributed Processing Module

For the most part, if you need to run Monte Carlo simulations for a complex model, the use of the Distributed Processing Module is advantageous and strongly encouraged. There are some situations, however, in which use of the Distributed Processing Module may not be feasible.

In particular, because a distributed processing simulation requires that potentially large quantities of data (i.e., the GoldSim model file and other associated files, the results of each realization) be passed between the Master and Slave processes over a network, effective use of the Distributed Processing Module requires fast network connections (and fast computers). If the connections between the processes are slow, a distributed processing simulation may be impractical.

In addition, there are two situations in which distributed processing is not supported at all:

- You cannot run multiple scenarios (using the Scenario Manager) and utilize distributed processing. That is, the Scenario Manager provides no access to distributing processing capabilities.
- Because distributed processing can not be used for a single realization, and high resolution results (in which all unscheduled timesteps are displayed) are only available for single realizations runs, high resolution results can never be viewed for distributed processing simulations.

Scenarios and high resolution results are discussed in detail in Chapters 7 and 8, respectively of the **GoldSim User's Guide**.

---

# Glossary of Terms

## **DP Plus**

A premium version of the Distributed Processing Module that allows an unlimited number of Slave processes to be used.

## **Master**

Also referred to as a Master process. A GoldSim executable that manages Slave processes in order to carry out distributed processing of a Monte Carlo simulation using the Distributed Processing Module.

## **Slave**

Also referred to as a Slave process. A GoldSim executable running on a client machine that can be used to carry out distributed processing of a Monte Carlo simulation using the Distributed Processing Module.

## **Slave Manager**

A Windows service that permits a Master process (on another machine) to remotely launch a GoldSim Slave process.



---

# Index

## A

- Aborting
  - distributed processing simulation 24
- Activating
  - Slaves 24
- Adding
  - Slaves for distributed processing 17

## C

- Closing
  - Slaves 9
  - Slaves remotely 21
- Connecting to GoldSim Slaves 17
- Context-sensitive Help 4
- Conventions
  - to describe key combinations 3
  - to describe mouse actions 3

## D

- Deactivating
  - Slaves 24
- Distributed processing
  - aborting a simulation 24
  - activating/deactivating Slaves 24
  - assigning multiple realizations 23
  - auxiliary files 25
  - closing a Slave 9
  - connecting to Slaves 17
  - importing/exporting Slaves 21
  - limitations 28
  - log files 28
  - overview 8
  - recovering simulation after failure 27
  - running a simulation 22
  - starting a Slave 9
  - terminating Slaves remotely 21
  - updating Slave status 21
  - using the Slave Manager 14
- Distributed Processing Module
  - description 2
  - overview 8
  - steps for using 8
  - using 7
  - versions 2

DP Plus 2

## E

- Email
  - support 4
- Exporting
  - Slave lists 21

## F

- Files
  - used by Slaves 25

## G

- GoldSim
  - blog 5
  - context-sensitive Help 4
  - Maintenance program 4
  - model library 5
  - online Help 4
  - running in Slave mode 9
  - training 5

## H

- Help 4

## I

- Importing
  - Slave lists 21

## K

- Key combinations
  - conventions to describe 3

## M

- Maintenance program 4
- Manual
  - Notes in 3
  - Warnings in 3
- Master
  - defined 8
- Model library 5
- Mouse actions
  - conventions to describe 3

## R

- Realizations per Slave transaction 23
- Running

---

distributed processing simulation  
22

## **S**

Slave

- closing 9
- configuring firewall 11
- connecting to 17
- defined 8
- importing and exporting 21
- running GoldSim as a 9
- terminating 9
- terminating remotely 21
- updating status 21

Slave Manager 14

- configuring Windows Firewall 15

Support

- email 4
- GoldSim user forum 4
- model library 5

## **T**

Technical support 4

Terminating

- Slaves 9
- Slaves remotely 21

Training course 5

## **U**

Updating

- Slave status 21