Parallel V1.1

Multi−workspace GAUSS *plus* networking

September 20th, 2003

by Matthew Ford

A networked user interface for running multi−workspace GAUSS programs. Parallel allows you to run multiple GAUSS programs, on local or remote computers, at the same time and combine the results (i.e. parallel or concurrent programming). It saves your work as you go and provides extensive undo functions.

by **Forward Computing and Control Pty. Ltd.**
©Copyright 2002−2003, Forward Computing and Control Pty. Ltd
ACN 003 669 994   NSW Australia,   All Rights Reserved.
email:Parallel@forward.com.au

Table of Contents

Table of Contents

Menus and Toolbar

Table of Contents

Installation for Windows

To install Parallel you need to install the following:−

1. GAUSS V5.0.23 or later and its licence
2. Java JRE 1.4.1
3. Parallel
4. Parallel licence

You can then start Parallel. If you have problems starting Parallel there are some common problems and their fixes listed at the end of this section.

Before setting up a Parallel network, review the Security Issues.

Install GAUSS V5.0

Parallel is an interface for GAUSS V5.0.23 or later. Before trying to run Parallel, first install GAUSS V5.0.23 or later.
You will need to obtain a valid GAUSS licence from Aptech Systems Inc. (www.aptech.com) to run GAUSS.
Check that it is working by running the **gauss.exe** program.

Install Java JRE 1.4.1

Parallel needs the Java Runtime Enviroment (JRE) V1.4.1 or a later version in order to run. If you do not already have JRE 1.4.1 installed, you can download it for free from Sun at http://www.java.com. (Choose either English or all languages)

Note: The "Typical" JRE installation installs Sun Java as the plugin Java for Microsoft Internet Explorer. If you use websites that need the Microsoft JVM instead of Sun's JRE, you will need to open Internet Explorer and choose the menu items, `Tools, Internet Options.., Advanced` and **untick** the `Java (Sun)`, `Use Java 2 V1.4.1` option and then restart.
This option is not used by Parallel.

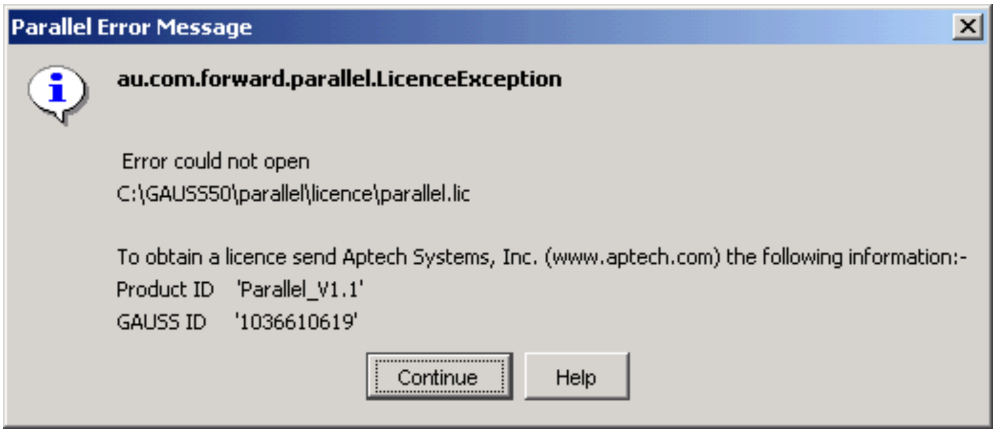
Install Parallel

On Windows machines, unzip the Parallel distribution zip to a temporary directory and then run the `ParallelInstall.exe` program. When the installer asks you to "Choose Install Location", browse to your GAUSS install directory.
You can delete the temporary directory when the installation is completed.

If you have not already installed Java JRE V1.4 you get an error message asking you to install it first.

Install Parallel Licence

When you first run Parallel (see Starting Parallel below) you will get an error message like:−
Error could not open C:\GAUSS50\parallel\licence\parallel.lic



au.com.forward.parallel.LicenceException

Error could not open
C:\GAUSS50\parallel\licence\parallel.lic

To obtain a licence send Aptech Systems, Inc. (www.aptech.com) the following information:-
Product ID 'Parallel_V1.1'
GAUSS ID '1036610619'

Contact Aptech Systems, Inc. with the **Parallel Product ID** and **GAUSS ID** shown in your error message so they can provide you with a Parallel licence. The Parallel Product ID and the GAUSS ID are also recorded in the **Parallel.log** file in the **parallel/logs** sub−directory of your home directory.

The Parallel licence will look like

```
 Key: 8165088      for GAUSS System: 1036610619
```

Place this text in a file called **parallel.lic** in the **parallel/licence** sub−directory of GAUSS.

Starting Parallel

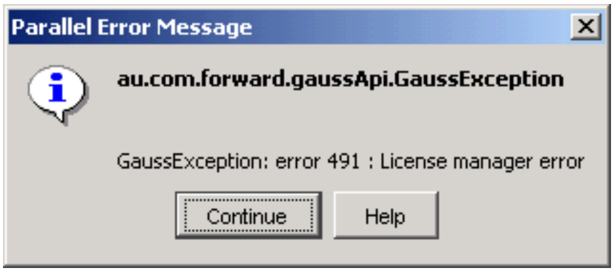
On Windows computers, if you installed the Desktop shortcut you can start Parallel by double clicking on the **Parallel V1.1** shortcut. Otherwise you can change to the GAUSS sub−directory and start Parallel by double clicking on **Parallel.exe** file.

Problems starting Parallel

If the Parallel main window does not appear, open a DOS Prompt window and run **parallel.exe** from there. This will show the error messages. The most likely cause is that you do not have Java JRE V1.4 installed or the **parallel.jar** file is not in the same directory as the **parallel.exe** file.

If Parallel starts but displays an error message, see the list below for common startup errors and their fixes.

Invalid GAUSS licence



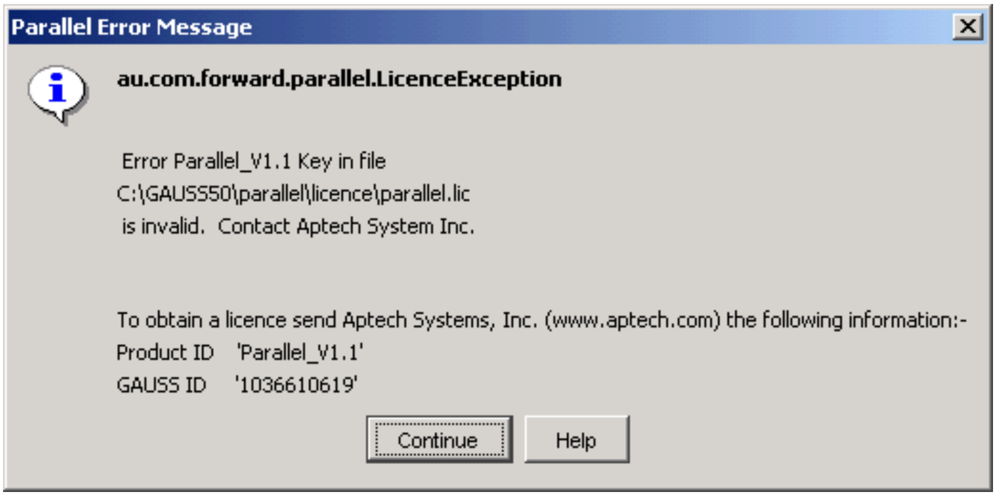
Parallel Error Message

au.com.forward.gaussApi.GaussException

GaussException: error 491 : License manager error

Check the installation of GAUSS V5.0. If you continue to have problems contact Aptech Systems, Inc.

Invalid Parallel licence



au.com.forward.parallel.LicenceException

Error Parallel_V1.1 Key in file
C:\GAUSS50\parallel\licence\parallel.lic
is invalid. Contact Aptech System Inc.

To obtain a licence send Aptech Systems, Inc. (www.aptech.com) the following information:-
Product ID 'Parallel_V1.1'
GAUSS ID '1036610619'

Check the installation of the Parallel licence. In particular check that the GAUSS System number in your **parallel.lic** file matches the GAUSS ID shown in the error message. If you continue to have problems contact Aptech Systems, Inc.

Parallel Software Licence Agreement

Version 1, May 2003

This LICENSE AGREEMENT covers Parallel only, and is in addition to the Aptech Systems, Inc, GAUSS licence agreement. This licence agreement is contained in the file **parallel_licence.html**

Parallel is owned by Forward Computing and Control Pty. Ltd. and distributed by Aptech Systems, Inc and its distributors.

Read Carefully Before Installing the Software Product

Customer: THE SOFTWARE PRODUCTS (AS DEFINED LATER BELOW) CONTAIN COPYRIGHTS, TRADE SECRETS AND OTHER INTELLECTUAL PROPERTY RIGHTS BELONGING TO FORWARD COMPUTING AND CONTROL PTY. LTD. ("FORWARD") and APTECH SYSTEMS, INC. ("APTECH"). THE SOFTWARE PRODUCTS MAY BE INSTALLED, COPIED, DISPLAYED, EXECUTED, OR USED IN ANY OTHER FASHION ONLY IN ACCORDANCE WITH THE TERMS OF THIS LICENSE AGREEMENT PURCHASED BY CUSTOMER. THE LICENSE AGREEMENT ARE HEREINAFTER COLLECTIVELY REFERRED TO AS THE "LICENSE AGREEMENT" OR "AGREEMENT". THE TERMS "YOU" OR "LICENSEE" MEAN CUSTOMER, AND THE TERM "YOUR" MEANS "CUSTOMER'S".

YOU MAY NOT INSTALL, COPY, DISPLAY, EXECUTE OR USE IN ANY OTHER FASHION THE SOFTWARE PRODUCTS EXCEPT IN ACCORDANCE WITH THE TERMS AND CONDITIONS OF THE LICENSE AGREEMENT. BY ANY USE OF ANY SOFTWARE PRODUCT, YOU ARE DEEMED TO HAVE ACCEPTED, AND YOU ARE BOUND BY, ALL THE TERMS AND CONDITIONS OF THE LICENSE AGREEMENT. IN ORDER TO USE SOFTWARE PRODUCTS, YOU MAY BE REQUIRED TO OBTAIN A UNIQUE LICENSE CODE OR CODES FROM FORWARD OR ITS DISTRIBUTOR TO ENABLE INSTALLATION, COPYING, DISPLAY, EXECUTION AND OTHER USE OF THE SOFTWARE PRODUCT.

IF YOU DO NOT ACCEPT AND AGREE TO THE TERMS AND CONDITIONS OF THE LICENSE AGREEMENT, YOU ARE PROHIBITED FROM INSTALLING, COPYING, DISPLAYING, EXECUTING OR USING IN ANY OTHER FASHION THE SOFTWARE PRODUCT. VIOLATION (I) CONSTITUTES INFRINGEMENT, MISAPPROPRIATION AND CONVERSION OF FORWARD'S COPYRIGHTS, TRADE SECRETS, AND OTHER INTELLECTUAL PROPERTY RIGHTS, (II) WILL SUBJECT THE INFRINGER TO SIGNIFICANT CIVIL LIABILITIES FOR DAMAGES (INCLUDING WITHOUT LIMITATION STATUTORY DAMAGES) AND OTHER REMEDIES AND LIABILITIES, AND (III) MAY CONSTITUTE VIOLATION OF CRIMINAL LAWS.

IF YOU DO NOT AGREE TO THE TERMS AND CONDITIONS OF THIS LICENSE

AGREEMENT, YOU MAY, WITHIN 30 DAYS OF PURCHASE AS EVIDENCED BY THE VENDOR'S RECEIPT AND/OR INVOICE, CONTACT THE VENDOR WHO SOLD YOU A SOFTWARE PRODUCT FOR ANY REFUND OR CREDIT AS MAY BE ALLOWED BY THE VENDOR'S PRODUCT RETURN POLICIES. IF, HOWEVER, YOU PURCHASED A SOFTWARE PRODUCT DIRECTLY FROM APTECH, YOU MAY RETURN THE SOFTWARE PRODUCT, EXCLUDING SOFTWARE UPDATES AND CONVERSIONS, AND THE PURCHASE PRICE WILL BE REFUNDED IN FULL ONLY IF CUSTOMER: (1) CONTACTS APTECH FOR A RETURN AUTHORIZATION NUMBER WITHIN 30 DAYS FROM DATE OF PURCHASE OF SUCH SOFTWARE PRODUCT AS EVIDENCED BY THE SOFTWARE PRODUCT RECEIPT/INVOICE; (2) RETURNS THE PACKAGE CONTAINING THE SOFTWARE PRODUCT AND ITS DOCUMENTATION, MEDIA AND ALL RELATED ITEMS TO APTECH SYSTEMS, INC. IN UNDAMAGED AND RESALABLE CONDITION.; AND CERTIFIES IN WRITING THAT CUSTOMER HAS NOT KEPT, DISTRIBUTED, SHARED, SOLD, LOANED ANY COPY OF THE SOFTWARE PRODUCT OR OTHERWISE VIOLATED THE LICENSE AGREEMENT.

IF CUSTOMER IS PURCHASING A SOFTWARE PRODUCT THAT IS AN UPGRADE OF AN EARLIER VERSION OF A FORWARD SOFTWARE PRODUCT, THEN THIS AGREEMENT SHALL SUPERSEDE THE TERMS AND CONDITIONS OF THE LICENSE THAT APPLIED TO THE EARLIER VERSION, AS OF THE DATE OF INSTALLATION OF THE UPGRADE.

Section 1. DEFINITIONS.

a. "Intellectual Property" means collectively the following: inventions, whether or not patented, copyrights, design rights, publicity rights, trade secrets, and any and all forms of intellectual property rights, whether tangible or intangible form, and all copies of Intellectual Property incorporated or embodied in any format or media, print, electronic, digital, magnetic, and all other forms, presently existing or that may be hereafter discovered.

b. The terms "Software Product" or "Software Products" mean ONLY the software program or software programs from those listed below that Customer purchased AND that is or are specifically identified in Aptech's invoice or invoices, including without limitation, applications, help files, libraries, source code, executables, applets, user manuals and documentation, technical manuals, systems manuals, keyboard function strips, all other products included in any package, box or container, and all associated Intellectual Property for all computer operating systems ("Platforms"):

− − Parallel, A user interface for GAUSS(TM) Products

c. "Person" means a natural Person or any form of organization or entity including, but not limited to, corporations, partnerships, associations, and governmental units.

d. "Customer" refers to the Person subject to this License Agreement and each Registered User to whom the Customer transfers rights in accordance with the terms of this Agreement.

e. "Initial Customer" means the Customer who initially Purchases (or otherwise obtains from Aptech) the Software Product.

f. To "Copy" a Software Product means to create a copy or representation of any portion of it in or on any medium or format, whether tangible or intangible, physical, electronic, magnetic, optical, or otherwise. A "Copy" refers to the representation thus created.

g. To "Disclose" a Software Product means to allow or authorize any Person to possess, use, or Copy any portion of it.

h. "Registration Form" means the user registration form accompanying the GAUSS System or GAUSS Engine Software Products, and any other user registration forms accompanying any of the Software Products.

i. "Licensee" or "Registered User" means the end−user that has submitted a Registration Form to APTECH for Parallel.

j. "Single User License" is a license for one individual end−user to install and use/access the Software Product on one stand−alone computer at any one time.

k. The terms "Payment", "Purchased", "Paid For" and/or "Paid−For" include both payments made in U.S. Dollars (or other authorized currency) in exchange for Software Products as well as Software Products otherwise lawfully obtained by Customer from FORWARD or its distributor. An invoice from FORWARD or its distributor describes each and every Software Products provided to Customer and any applicable License Fees due.

Section 2. LICENSE.

a. Subject to Payment of the license fee and other charges payable to FORWARD specified on the Software Product Invoice (or unless otherwise granted by FORWARD) and subject to other terms and conditions of the License Agreement, FORWARD grants to Customer a nonexclusive limited license to use the Software Product packaged with this License Agreement, as hereinafter described.

b. FORWARD reserves all rights to the Software Products not specifically granted herein.

c. THIS SECTION 2 DESCRIBES THE SINGLE USER LICENSE GRANTS.

(1) SO LONG AS NO TERM OR CONDITION OF THIS LICENSE AGREEMENT IS BREACHED, CUSTOMER MAY:

A. Only use the Software Product on a single computer at any one time. Parallel must be installed on the same Platform as the registered copy of GAUSS.

B. Transfer the Software Product to another Person provided that: (i) the transferor Customer provides APTECH with a written notice of change of Registered User; (ii) the transferor Customer has erased, deleted and destroyed all copies of the Software Product which remain in Customer's possession or control; (iii) the transferor Customer transfers the original Media and all documentation, including a copy of the License Agreement to the transferee Registered User; (iv) in any such transfer, although the transferor Customer's right to install, Copy or use the Software Product is terminated, the transferor Customer shall remain obligated and liable to FORWARD for any breaches of the License Agreement occurring prior to the transfer; and (v) the transferee Registered User agrees to all of the terms and conditions of the License Agreement;

C. Make a reasonable number of copies of the online documentation for personal use;

D. Make one Copy of the Software Product for use as a backup Copy provided that:

(i) the Copy includes all notices of copyright and all other proprietary rights appearing in and on the Software Product;

(ii) the Copy and the original Software Product are not in use at the SAME time;

(iii) Customer establishes a procedure for accounting for the Copy at all times; and (iv) Customer destroys or returns to APTECH the Copy when it is no longer required.

(2) YOU MAY NOT:

A. Electronically transfer the Software Product from one computer to another over a network.

B. Modify or translate those portions of any Software Product program that are provided in an object or source code form, or alter, merge, modify, reverse engineer or adapt the Software Product in any way including disassembling or decompiling and

C. Copy, loan, rent, lease or sublicense, assign, convey, sell, mortgage, pledge, or in any manner use or transfer all or any of Customer's rights and obligations under this Agreement or in any Software Product, except as provided in subsection 2.B.(1) above, without FORWARD's prior written consent.

Section 3. LIMITED WARRANTIES

Only the limited warranties expressly described in this Section 3 are made to the Initial Customer installing the software who has submitted a Registration Form to APTECH; and no warranty of any kind, whether express or implied, is made to any other Person or transferee Registered User:

a. FORWARD warrants that it has the right to grant the licenses contained in the License Agreement.

b. FORWARD warrants that the media in which the Software Product is recorded is free from defects in materials and workmanship under normal use, for a period of 90 days from the date of original delivery to Customer.

c. FORWARD makes no warranty of any kind, implied or express, that the Software Product will meet Customer's requirement or that the Software Product will be uninterrupted or error−free.

Section 4. LIMITATIONS ON LIABILITY AND REMEDIES.

a. General. FORWARD's only obligation, liability and/or remedy, with respect to the warranties set forth in Section 3 shall be one of the following, which FORWARD shall have sole discretion to choose: (1) to provide replacement media (whether in the form of the current release or otherwise) to Software Product; (2) to replace, without charge, Software with a functionally equivalent program; or (3) to refund the applicable license fees paid to FORWARD by Customer. These are the sole and exclusive remedies for any breach of warranty, as determined by FORWARD. FORWARD shall have no obligation, liability or any other kind of responsibility for the costs of restoring any destroyed or defective data, installation of replacement media, or software conversion work made necessary by replacement media, even if FORWARD had been specifically informed by Customer of the possibility or probability of any such consequences. During the ninety (90) day warranty period under Section 3.b above, FORWARD will replace, without charge, on an exchange basis, any media which is not as warranted. Customer must return the defective item postpaid to FORWARD, postmarked within the time period stated above. Customer must either insure the defective item being returned or assume the risk of loss or damage in transit. Any claim under the above warranty must include a copy of Customer's receipt or invoice or other proof of the date of delivery.

b. NOTICE: THE WARRANTIES, OBLIGATIONS AND LIABILITY OF FORWARD AND THE RIGHTS AND REMEDIES OF CUSTOMER SPECIFICALLY SET FORTH IN THIS AGREEMENT ARE THE CUSTOMER'S EXCLUSIVE REMEDIES. FORWARD DISCLAIMS AND CUSTOMER HEREBY WAIVES, DISCHARGES AND RELEASES FORWARD, ITS OFFICERS, DIRECTORS, EMPLOYEES, AGENTS AND DISTRIBUTORS, FROM ANY AND ALL (i) OTHER WARRANTIES, OBLIGATIONS, OR LIABILITIES AND (ii) RIGHTS, CLAIMS, CAUSES OF ACTIONS, EXPRESS OR IMPLIED, ARISING BY LAW OR OTHERWISE IN CONNECTION WITH, REGARDING, OR ARISING FROM ANY AND ALL SOFTWARE OR OTHER TANGIBLE OR INTANGIBLE ITEMS OR SERVICES PROVIDED UNDER THIS AGREEMENT (INCLUDING ALL INCORPORATED DOCUMENTS REFERENCED HEREIN). THE AFORESAID WAIVER, DISCHARGE AND RELEASE INCLUDES, BUT NOT LIMITED TO: (1) ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A SPECIFIC PURPOSE; (2) ANY IMPLIED WARRANTY ARISING FROM COURSE OF PERFORMANCE, COURSE OF DEALING, OR USAGE OF TRADE; AND (3) ANY OBLIGATION, LIABILITY, RIGHT, CLAIM, OR REMEDY, WHETHER IN TORT, CONTRACT, OR OTHERWISE. IN NO CASE SHALL FORWARD BE LIABLE FOR ANY INCIDENTAL, INDIRECT, SPECIAL, CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, LOSS OF USE, REVENUE, PROFIT, OR

DATA), PUNITIVE DAMAGES, EXEMPLARY OR OTHER STATUTORY DAMAGES, EXPENSES, COSTS (INCLUDING ATTORNEY FEES). IN ADDITION, IN NO CASE SHALL FORWARD'S LIABILITY EXCEED THE LICENSE FEES PAID BY CUSTOMER FOR THE SOFTWARE PRODUCT.

Section 5. TRADE SECRETS AND INTELLECTUAL PROPERTY.

a. Intellectual Property. This License Agreement neither shall be deemed to assign, transfer, or convey to, nor assigns, transfers or conveys to, Customer any right, title or interest to any part of the Software Product beyond the express terms of the grants of licenses described in this License Agreement. In addition, the Intellectual Property's trade secrets shall not be Disclosed, used, disseminated, or published other than in accordance with the terms and conditions of this License Agreement. FORWARD reserves to itself all rights, title and interest that are not expressly licensed herein.

b. Copyrights. The Software Product also contains copyrights owned by FORWARD under the copyright laws of Australia, the Universal Copyright Convention, and the Berne Convention.

Section 6. TERMINATION OF LICENSE GRANTS.

a. Except for the termination of the grants of license described herein, all other terms and conditions of this License Agreement (when Paid For or otherwise obtained from FORWARD) shall survive any such termination of the license rights granted to Customer hereby. All license grants made to Customer are automatically terminated upon any breach of any term or condition of the License Agreement. In the event of any such termination, the Customer shall immediately return to FORWARD (or any agent designated by FORWARD for such purpose) any and all existing copies of the Software Product in the possession or control of the Customer, and the Customer shall certify in writing Customer's compliance with this provision and any destruction of any copies of the Software Product not so returned.

b. No termination of the license grants made herein shall act to release, waive or otherwise discharge Customer from any and all claims and causes of action held by FORWARD against Customer for breach of the License Agreement, infringement of intellectual property rights, misappropriation of trade secrets, or any other claim or cause of action arising from Customer's wrongful acts and omissions.

Section 7. GENERAL.

a. FORWARD's Benefit, Applicable Law, Jurisdiction, and Venue. This Agreement is made for the sole benefit of the parties hereto, and there are no third party beneficiaries of this Agreement, including without limitation any third−party users of any products using or derived from any of FORWARD's products. This License shall be interpreted under the laws of the State of New South Wales, Australia, notwithstanding the application of any jurisdiction's choice−of−law rules to the contrary. Customer and FORWARD consent to personal jurisdiction, subject matter jurisdiction, and venue in that state. Any action relating to this Agreement must be brought in the state of New South Wales, Australia.

b. Attorney's Fees. In the event legal action is brought by either party to enforce any of the provisions of this Agreement, the prevailing party shall recover its reasonable attorney's fees, costs, and expenses, including but not limited to fees, costs and expenses of collecting any judgment.

c. Waiver or Delay. No waiver of any provision of this Agreement, nor delay in enforcing any rights hereunder, shall be construed as a continuing waiver or create an expectation of non−enforcement of that or any other provision or right. Further, no waiver or modification of this Agreement or any covenant, condition or limitation herein contained shall be valid and no evidence of waiver or modification shall be offered or received in evidence in any proceeding, arbitration or litigation between the parties hereto arising out of or affecting this Agreement or the rights or obligations of the parties hereunder, unless such waiver or modification is in writing duly signed by both parties.

d. Severability. All agreements and covenants contained in this Agreement and any documents (tangible or intangible, electronic, magnetic, or in any other format or media) incorporated by referenced herein, are severable and in the event that any of them shall be held to be invalid by any competent court, this Agreement and referenced documents shall be interpreted as if such invalid agreement or covenant is not contained herein and all other provisions shall remain in full force and effect.

e. Entire Agreement. The License Agreement constitute the entire agreement between the parties relating to Software Product. Any and all prior negotiation or oral representations, and any contrary provisions in any form of Purchase orders from Customer, shall be superseded by the License Agreement. No later modification of the Agreement may be made unless in writing signed by both parties.

Forward Computing and Control Pty. Ltd.
27 Cottee Cr. Terrigal, NSW, Australia ACN 003 669 994

Introduction

Welcome to **Parallel V1.1**. The purpose of this introduction is to outline the features of Parallel and its user interface. This tutorial and reference manual assumes you are already familiar with GAUSS commands and GAUSS programming. If not, please consult the GAUSS documentation. The on−line help in Parallel contains the GAUSS commands reference.

What is Parallel V1.1?

It is a networked user interface for running multi−workspace GAUSS programs.
Parallel can run multiple GAUSS programs, on local or remote computers, at the same time and combine the results (i.e. parallel or concurrent programming). It saves your work as you go and provides extensive undo functions.

Boost GAUSS' productivity. Use Parallel to:−

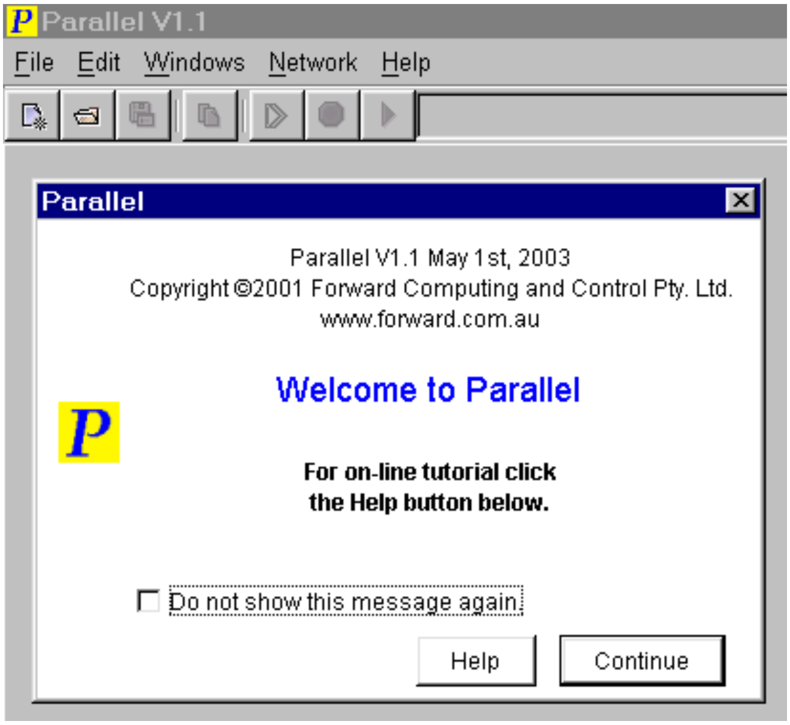
- Run GAUSS programs in multiple workspaces at the same time.
- Distribute these multiple GAUSS workspaces across networked computers.
- Write your own parallel programs without a computer science degree. You need to use just three extra commands. In most cases your existing GAUSS programs can be incorporated into a parallel program with no change.
- Make use of your multi−processor hardware to increase GAUSS' throughput.
- Continue preparing and queueing programs for execution while your program is running.
- Continue working in another workspace while your queued programs are running.
- View the Global Matrix and String variables in a workspace and drag and drop to copy them to another workspace.
- Type ahead to supply input GAUSS programs.
- Supply GAUSS input from your programs for un−attended execution.
- No need to save. All your work is saved as you do it.
- When you come back to Parallel, the workspaces you open are just as you left them.
- Extensive Undo/redo function that lets you recover from errors. Recover your workspace after GAUSS' autodelete removes all your globals. Even recover when you overwrite an existing workspace using the SaveAs function.

purpose other than the purchaser's personal use without the written permission of Forward Computing and Control Pty. Ltd.

Quick Start

Starting Parallel

After installing Parallel, on Windows machines you can start Parallel by running the **parallel.exe** file located in the Parallel sub−directory.



Parallel V1.1 May 1st, 2003
Copyright ©2001 Forward Computing and Control Pty. Ltd.
www.forward.com.au

Welcome to Parallel

For on-line tutorial click
the Help button below.

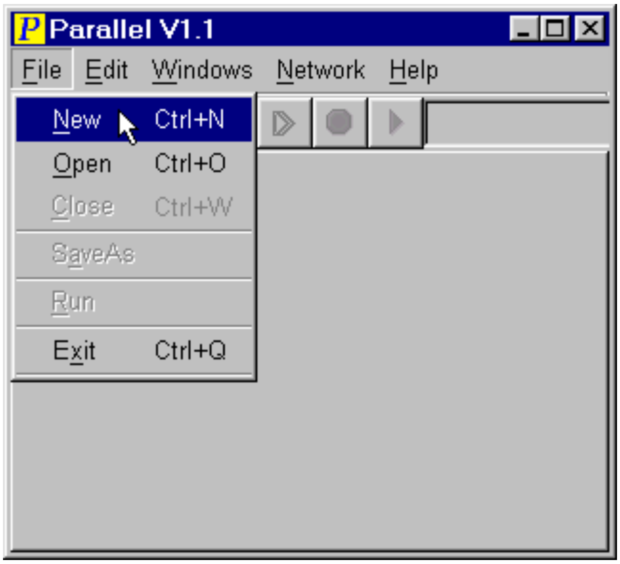
□ Do not show this message again

If Parallel does not start, check out these possible problems.

Create a New Workspace

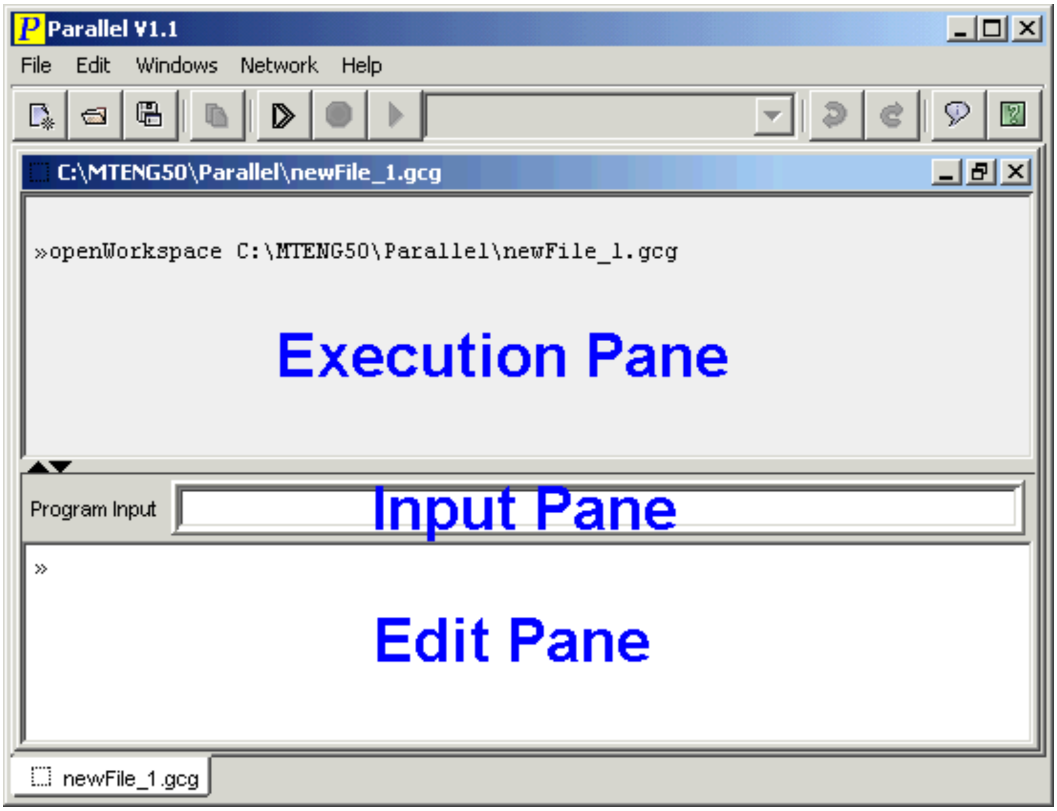
Select the **`File`, `New`** menu option to create a new workspace. This creates a new GAUSS workspace with the name **newFile_*n*.gcg** in the current directory, where ***n*** is replaced with the next available number.

If you have already created some new Workspaces in Parallel then your new files will have higher numbers than shown in these screen shots.



The Workspace Window

There are three sections to the workspace window. From the top they are:− the **Exection Pane**, the **Input Pane** and the **Edit Pane**.

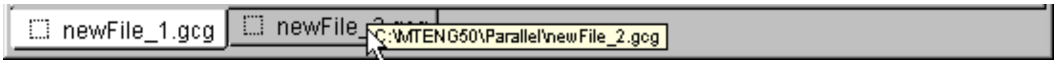


```
»openWorkspace C:\MTENG50\Parallel\newFile_1.gcg
```

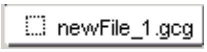
```
»
```

- The **Execution Pane** is where the execution of programs is displayed. You cannot edit any text in this pane but you can highlight text and copy and paste it to the Edit or Input Panes.
- The **Input Pane** is where you enter input requested by GAUSS programs.
- The **Edit Pane** is where you type the GAUSS programs that you are preparing to execute.

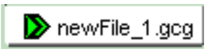
The currently open windows are shown as tabs at the bottom of the Parallel application window. You can click on these tabs to switch between workspaces. The tooltips give the full name of the workspace.



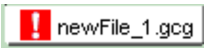
These tabs also show the current state of each workspace.



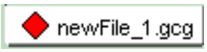
execution completed normally



a program is running



flashing when the program is waiting for user input



the program ended with an error

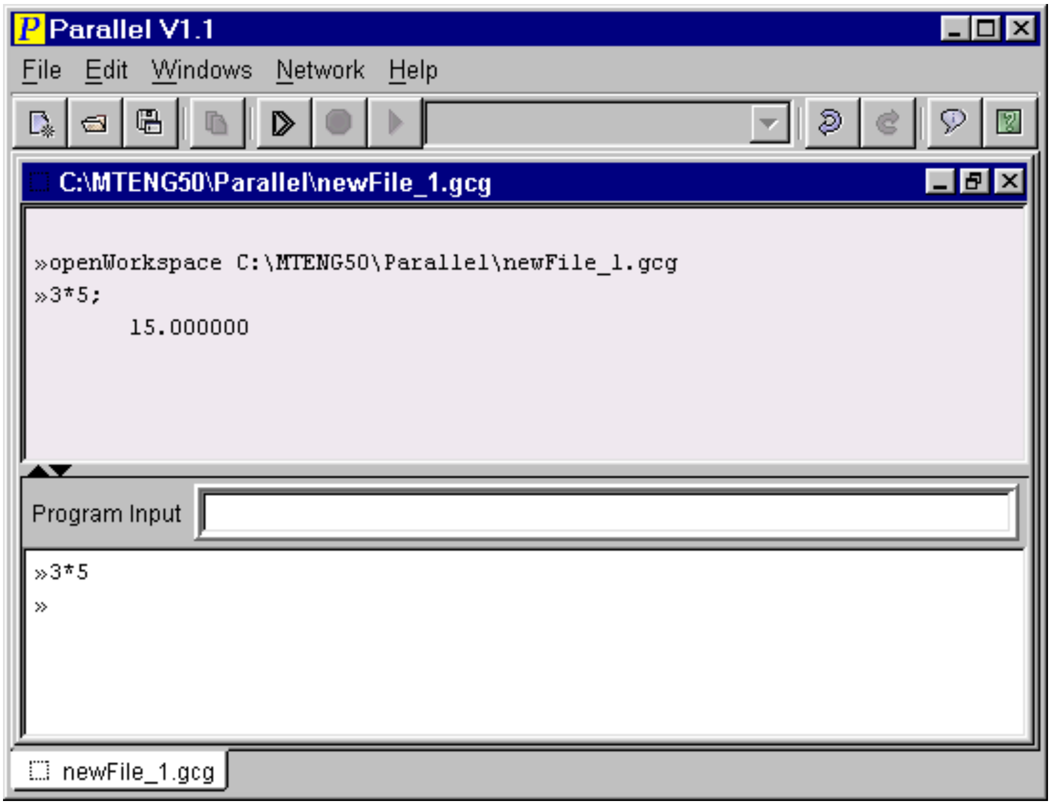
Execute a GAUSS command

In the Edit pane (bottom pane), type the following command. (No '`;`' at the end of this command.)

```
»3*5
```

To queue this program for execution by GAUSS, press the **Enter** key.

This will transfer all the commands between the two » symbols that enclose the cursor to the **Execution Pane** to be executed by GAUSS.


Parallel V1.1
File Edit Windows Network Help
C:\MTENG50\Parallel\newFile_1.gcg
Program Input
newFile_1.gcg

```
»openWorkspace C:\MTENG50\Parallel\newFile_1.gcg
»3*5;
       15.000000
```

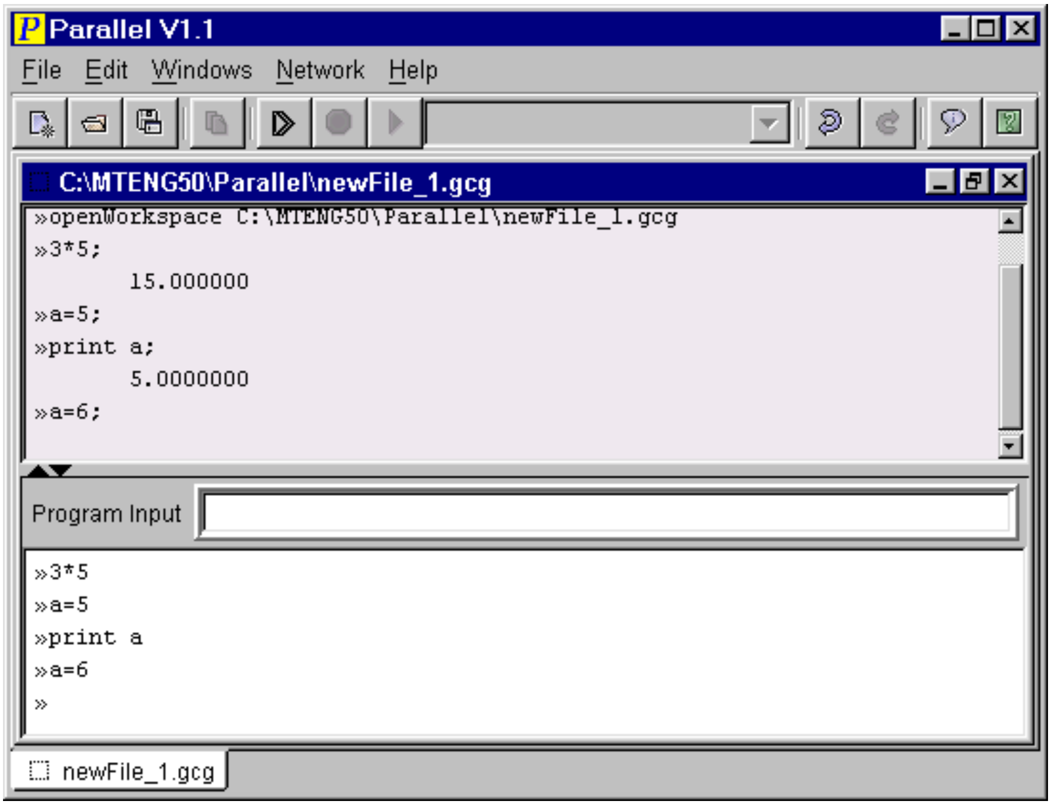
```
»3*5
»
```

Tip: If you type `;` and then **Enter**, the Edit Pane starts a new line without queuing the program for execution. This lets you type a multi−line program by terminating each command with `;` and then pressing **Enter**. If you press **Enter** anywhere other then immediately after the `;` character, the program will be queued for for execution.
You can also use **Shift+Enter** or **Ctrl+Enter** to insert a newline anywhere in the Edit Pane without queuing the program for execution.

Re−executing a command

In the Edit pane type, the following three single command programs (pressing **Enter** at the end of each line to submit each one for execution one at a time. Do **not** add a `;` at the end of these commands.)

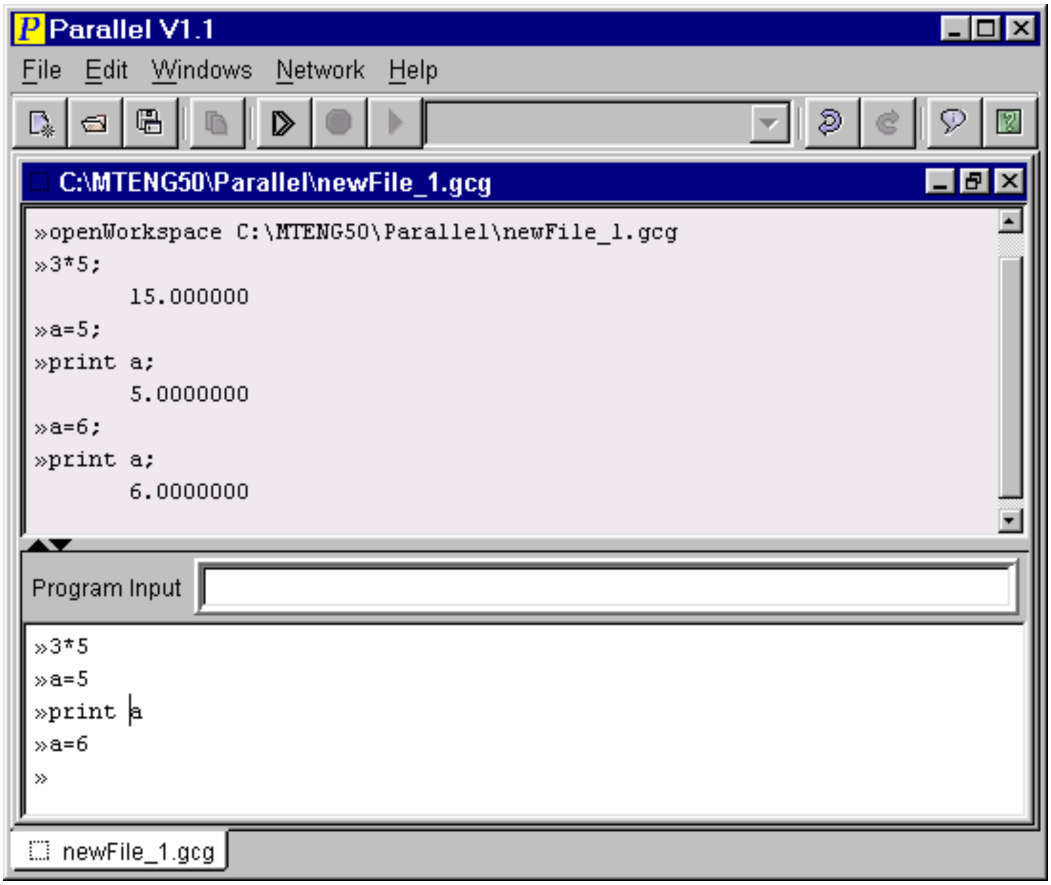
```
»a=5
»print a
»a=6
»
```



```
»openWorkspace C:\MTENG50\Parallel\newFile_1.gcg
»3*5;
       15.000000
»a=5;
»print a;
       5.0000000
»a=6;
```

```
»3*5
»a=5
»print a
»a=6
»
```

To re−execute **`print a`** , position the edit cursor in the edit pane anywhere between the two » symbols that enclose the **`print a`** command and press the **Enter** key. The **`print a`** program is re−submitted to GAUSS for execution.


Parallel V1.1
File Edit Windows Network Help
C:\MTENG50\Parallel\newFile_1.gcg
Program Input
newFile_1.gcg

```
»openWorkspace C:\MTENG50\Parallel\newFile_1.gcg
»3*5;
       15.000000
»a=5;
»print a;
       5.0000000
»a=6;
»print a;
       6.0000000
```

```
»3*5
»a=5
»print a
»a=6
»
```

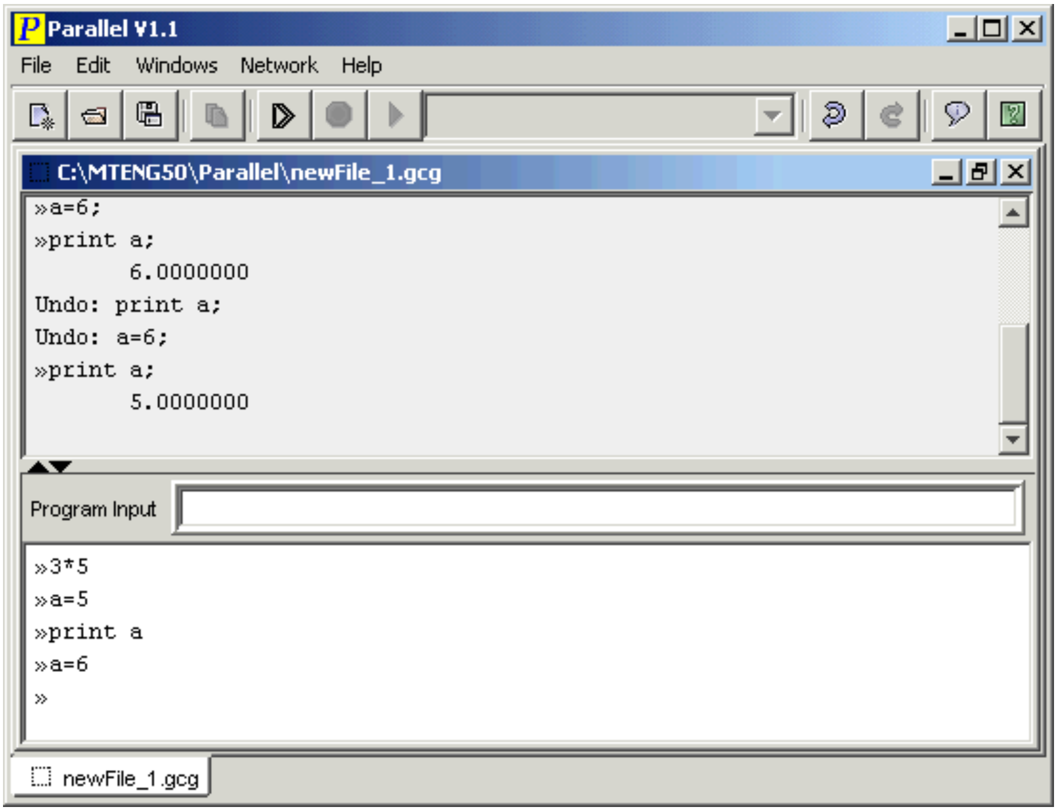
Tip: Multiple commands can be submitted for execution together. If you end the command with **;** and then press **Enter** you will get a new line in the Edit Pane without submitting the commands for execution. Then you can press **Enter**, anywhere other then immediately after the **;** character, to submit multiple commands at the same time. All commands between the two GAUSS command start symbols that surround the edit cursor position will be submitted.
That is if you want to add another command to this group of commands press **;** then **Enter** at the end of the current command, if this is the last command in the group, just press **Enter**.
This is easier to do then to describe. Try it.
Shift+Enter and **Ctrl+Enter** will insert a newline directly at the cursor position.

Undo/Redo

Parallel allows you to undo and redo programs executed by GAUSS.

The **Edit, Undo** menu item undoes the effects of the last command. Select it twice. The first undo reverses the last **print a** program and the next undo reverses the **a=6** program.

Now when you re−enter the **print a** command, GAUSS returns the result 5.0000000


Parallel V1.1
File Edit Windows Network Help
C:\MTENG50\Parallel\newFile_1.gcg
Program Input
newFile_1.gcg

```
»a=6;
»print a;
       6.0000000
Undo: print a;
Undo: a=6;
»print a;
       5.0000000
```

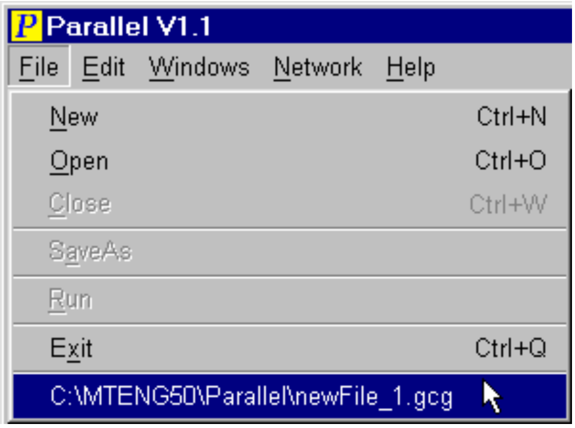
```
»3*5
»a=5
»print a
»a=6
»
```

The **`Edit`**, **`Redo`** menu item allows you to change your mind and redo programs if you have undone too may steps.

NOTE: Once you have executed a new program, you can not redo old programs you have undone. However all is not lost, see the section on the Command Log for information on recovering saved undo/redo workspaces.

Closing and Re−Opening a Workspace

To close the workspace, use the **`File`**, **`Close`** menu item. To re−open a workspace you can either use the **`File`**, **`Open`** menu item or select the workspace file name from the recently used list at the bottom of the **`File`** menu.



The undo/redo list is restored when the workspace is re−opened.

Tip: There is no File, Save command as all changes to workspaces are saved as they happen.

Exiting Parallel

When you want to exit Parallel use the **`File`**, **`Exit`** menu item.

What Next?

To set up a Parallel Network see Parallel Networking

For an introduction to Parallel Programming using Parallel see the next section.

Following the Parallel Programming section, there is a detailed Simple Example of running a parallel program in Parallel.

For more information on how to write your own Parallel command files, see Writing Parallel Programs in the Command Log section.

Parallel Programming

Parallel allows you to write and execute parallel (concurrent) GAUSS programs that pass results to each other. The simplest way to write your own Parallel programs is to first execute the programs by hand and then edit the resulting command log (see Writing Parallel Programs for details).

This section will show you how to execute parallel programs by hand and how to transfer the results between workspaces. It assumes you have read the Quick Start tutorial.

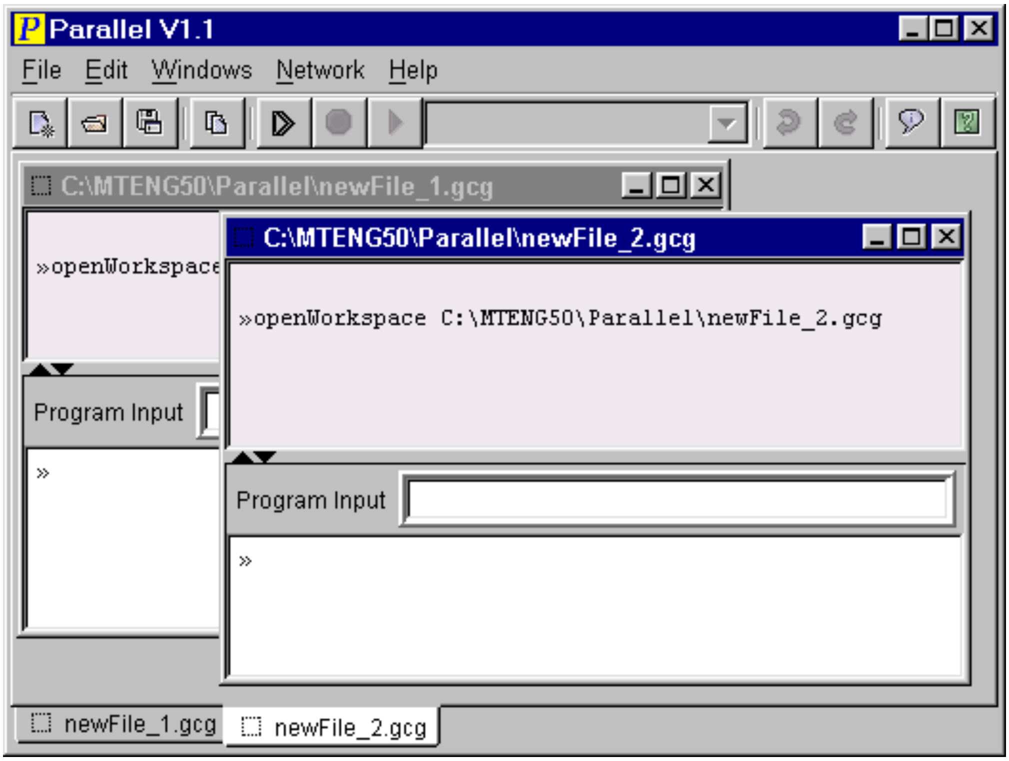
Three features of Parallel make parallel programing possible:−

i. You can have multiple GAUSS workspaces open and running GAUSS programs at the same time.
ii. You can queue up GAUSS programs in a workspace to be run in sequence while the workspace is busy running the previous ones.
iii. You can copy globals from one workspace to another to combine the results of the parallel programs into a single result.

In Parallel Networking you will see that you can also transfer files between computers under program control, just like copying globals between workspaces.

Multiple Workspaces

To get multiple workspaces in Parallel, either open or create more than one workspace. Each GAUSS workspace has its own independent set of global variables, procedures and user input. Each workspace is displayed in its own window within the Parallel application window.

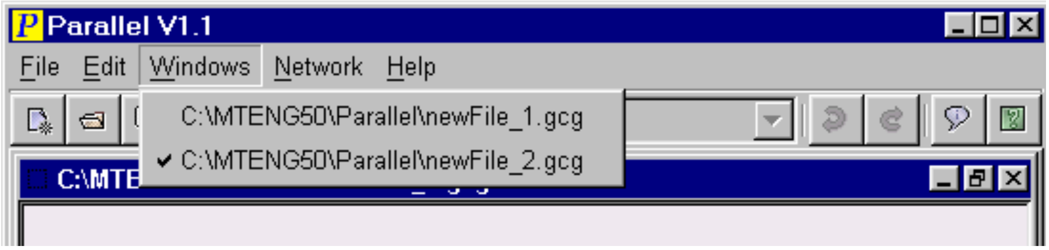

Parallel V1.1
File Edit Windows Network Help
C:\MTENG50\Parallel\newFile_1.gcg
C:\MTENG50\Parallel\newFile_2.gcg
Program Input
Program Input
newFile_1.gcg
newFile_2.gcg

```
»openWorkspace
```

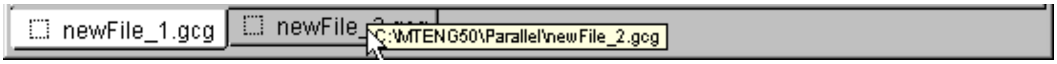
```
»openWorkspace C:\MTENG50\Parallel\newFile_2.gcg
```

Note: Some GAUSS commands effect **all** workspaces. The most important of these are **`chdir`** and **`ChangeDir()`**. These two GAUSS commands change the working directory for **all** workspaces on that computer. This effects where files are opened from and saved to.

To switch between workspaces you can use the **`Windows`** menu to select the workspace you want.


Parallel V1.1
File Edit Windows Network Help
C:\MTENG50\Parallel\newFile_1.gcg
✓ C:\MTENG50\Parallel\newFile_2.gcg
C:\MTE

The currently open windows are also shown as tabs at the bottom of the Parallel application window. You can click on these tabs to switch between workspaces. The tooltips give the full name of the workspace.



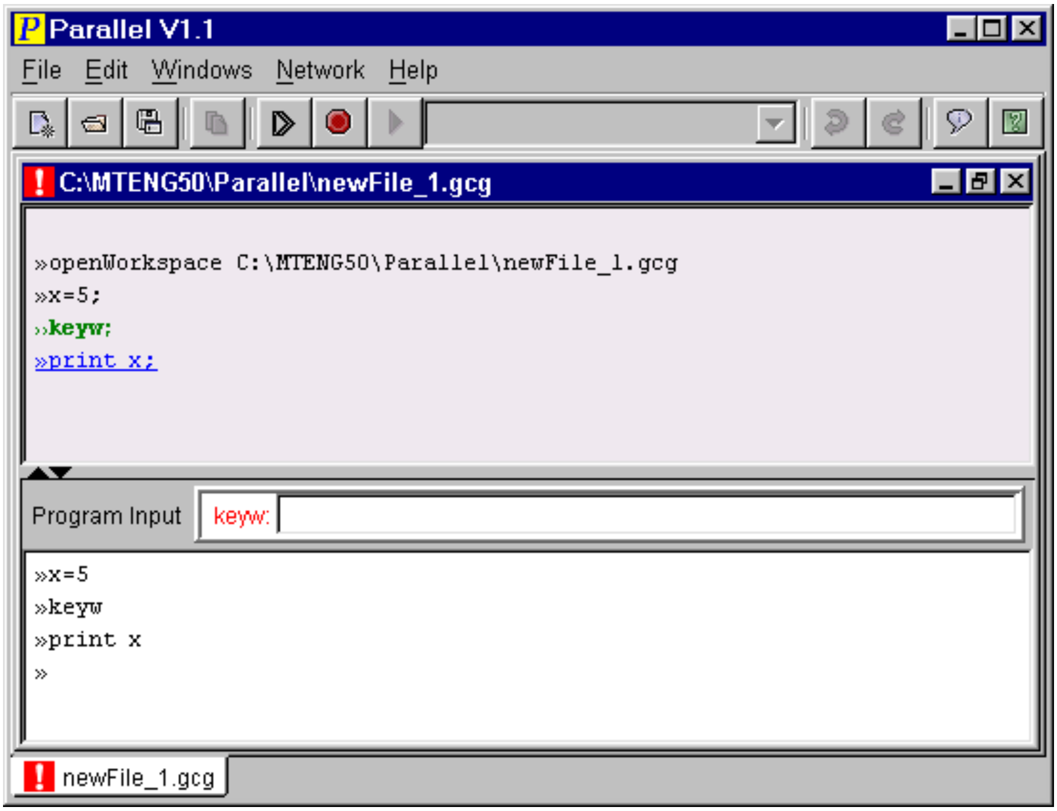
See the **`OpenWorkspace`** command for how to open and switch between workspaces while running a Parallel command program.

Parallel Program Queue

In Quick Start you have seen how you can enter a program for GAUSS to execute. However you do not have to wait for the GAUSS to finish executing the program before queueing more programs to be executed after the current one finishes.

Enter the following three single command GAUSS programs, pressing the **Enter** key at the end of each line

```
x = 5
keyw
print x
```



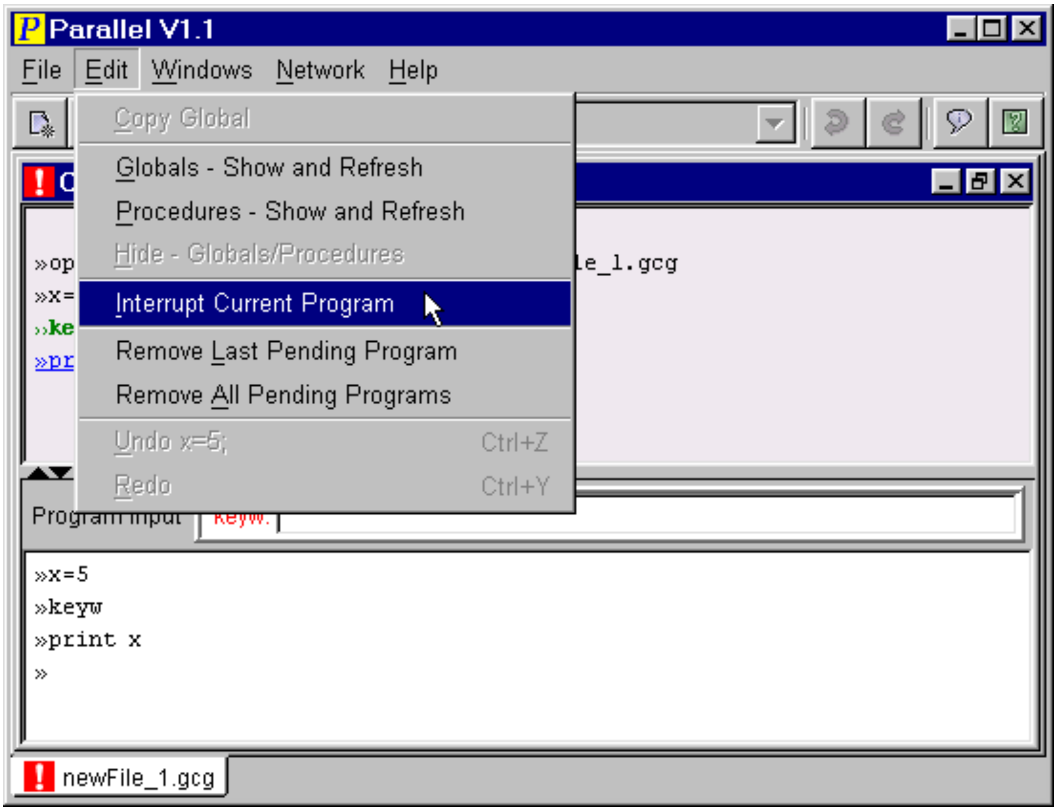
```
»openWorkspace C:\MTENG50\Parallel\newFile_1.gcg
»x=5;
»keyw;
»print x;
```

```
»x=5
»keyw
»print x
»
```

Your screen should now look like the above.

The **`x=5`** command has completed execution and so is shown in normal BLACK type. The **`keyw`** command is currently executing waiting for a key to be input, so it is shown in bold **GREEN**. The `print x` command is waiting to execute, and so it is shown in underlined BLUE.

Open the **`Edit`** menu and look at the options available before you type any keys into the **Input Pane** (the middle pane).

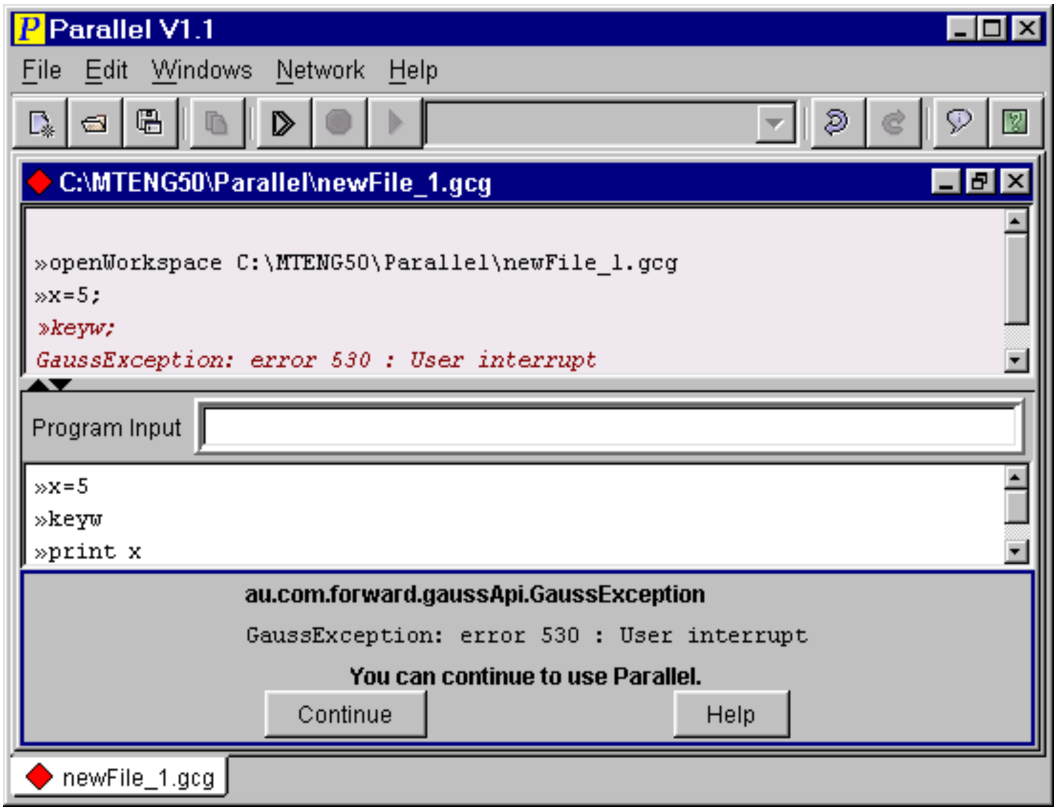

Parallel V1.1
File
Edit
Windows
Network
Help
Copy Global
Globals - Show and Refresh
Procedures - Show and Refresh
Hide - Globals/Procedures
Interrupt Current Program
Remove Last Pending Program
Remove All Pending Programs
Undo x=5;
Ctrl+Z
Redo
Ctrl+Y
Program input
keyw.
newFile_1.gcg

```
»op                le_1.gcg
»x=
»ke
»pr
```

```
»x=5
»keyw
»print x
»
```

When there are pending programs, you can remove either the last one or all the pending programs. When GAUSS is running a program, the **`Interrupt Current Program`** menu item is also available which prints an error message and removes all pending programs.

Select the **`Edit`** menu item, **`Interrupt Current Program`**. You screen should now look like this.



```
»openWorkspace C:\MTENG50\Parallel\newFile_1.gcg
»x=5;
»keyw;
GaussException: error 530 : User interrupt
```

```
»x=5
»keyw
»print x
```

au.com.forward.gaussApi.GaussException

GaussException: error 530 : User interrupt

You can continue to use Parallel.

When a GAUSS progam does not complete due to an error, all following programs are removed and the program in error is shown in *RED* italics together with the error message.

Tip: You can use **`errorlog()`** or **`end`** in your programs to force Parallel to end that program with an error and remove the pending programs.

CopyGlobal Command

The final feature of Parallel that makes parallel programming in GAUSS possible is the **`CopyGlobal`** command. This is used to copy the value of a global variable from one workspace to another. If the destination global variable does not exist it is created.

When you enter a **`CopyGlobal`** command it adds itself to the bottom of both workspaces' program queues and blocks the execution of any following programs in either workspace until both workspaces have completed execution of this **`CopyGlobal`** command.

This *synchronizes* the two GAUSS workspaces and ensures they are both in a know state when the value of the global variable is copied.

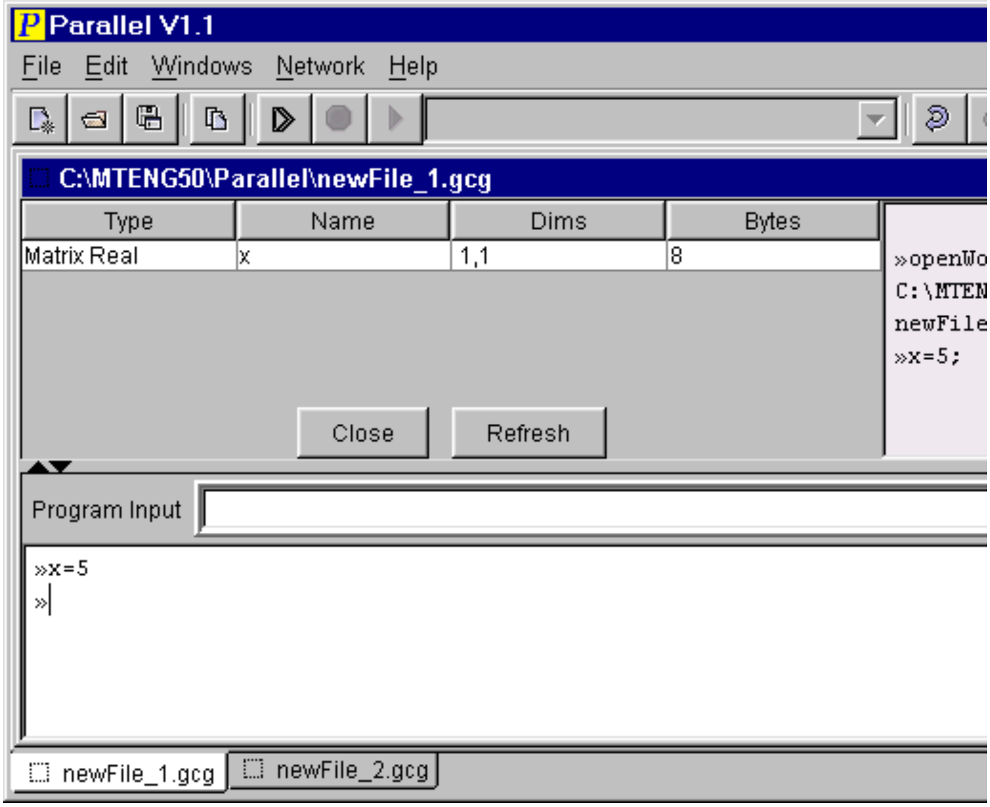
There are three ways to enter a **`CopyGlobal`** command:−

- Drag and drop a Global variable from the Show Globals table
- Use the Copy Global dialog box
- Type a CopyGlobal command into the Edit Pane or run one from a Parallel command file.

In order to copy a global from one workspace to another you need to have at least two workspaces open. Open a second workspace now using either **`File`** , **`New`** or **`File`** , **`Open`**

Copy Global Drag and Drop

To copy a global variable (a String, Matrix or String Array) using Drag and Drop, first select the workspace window you want to copy the global from, in this example newFile_1.gcg, and then open the Global viewer using **`Edit`**, **`Globals - Show and Refresh`**



Type	Name	Dims	Bytes
Matrix Real	x	1,1	8

```
»openWo
C:\MTEN
newFile
»x=5;
```

```
»x=5
»
```

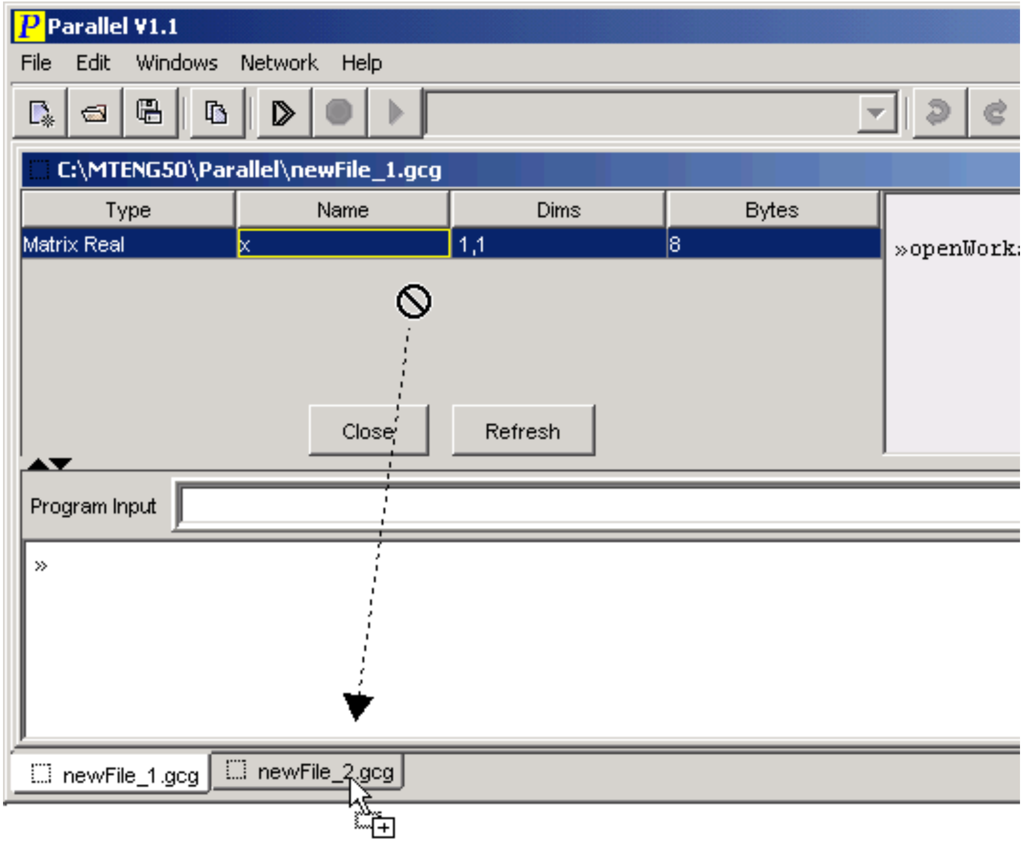
This displays all the String, Matrix and String Array globals that existed in this workspace at the end of the last GAUSS program that executed prior to choosing this menu item.

Tip: You can sort on the various columns it this table and re−arrange their order. See Edit, Globals − Show and Refresh in the Menu section for details.

If the GAUSS program that creates the global to be copied has not finished executing yet, you may not see the global in the table. In that case, click Refresh when the program completes, or alternatively you can use the Copy Global dialog box (below).

Select the row describing the global variable you want to copy by clicking on it. Release the left mouse button. Then drag this row (by pressing and holding down the left mouse button and then moving the mouse) to the workspace tab of the workspace you want to copy it to. Release the left mouse button to drop. You can only drop on one of the unselected (grey) workspace window tabs. You cannot drop on the currently selected (white) window tab.

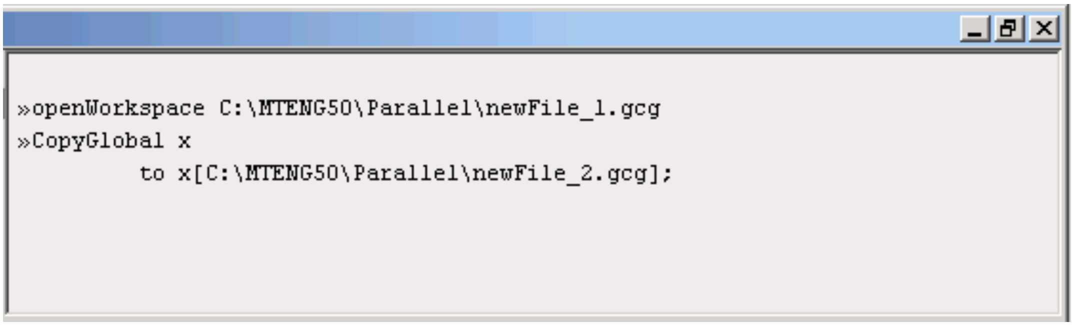
If you drop in the Edit Pane, you will just insert the name of the global at the current edit cursor position.



Type	Name	Dims	Bytes
Matrix Real	x	1,1	8

Note: Drag and Drop is a two step process. First you select the global variable you want to copy by clicking on it. Then release the left mouse button and next press and hold down the left mouse button and drag and drop. You will know you are in Drag and Drop mode because, when you move the mouse the cursor changes to ⊘

When you release the mouse button over the workspace tab, a CopyGlobal command will be inserted into both the source and the destination window. The copy will occur only when both workspaces are ready to execute this command.

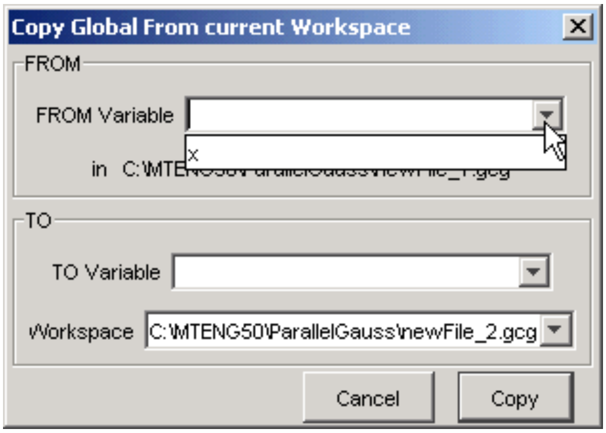


```
»openWorkspace C:\MTENG50\Parallel\newFile_1.gcg
»CopyGlobal x
         to x[C:\MTENG50\Parallel\newFile_2.gcg];
```

Copy Global Dialog Box

To copy a global variable using the Copy Global dialog box, first select the workspace window you want to copy the global from, in this example newFile_1.gcg and then choose the menu item **`Edit`**, **`Copy Global`**

The From Variable drop down list shows the same globals as the global table (above). However in the dialog box you can type in the name of the variable you want to copy if it is not in the list.



Copy Global From current Workspace

FROM

FROM Variable: x

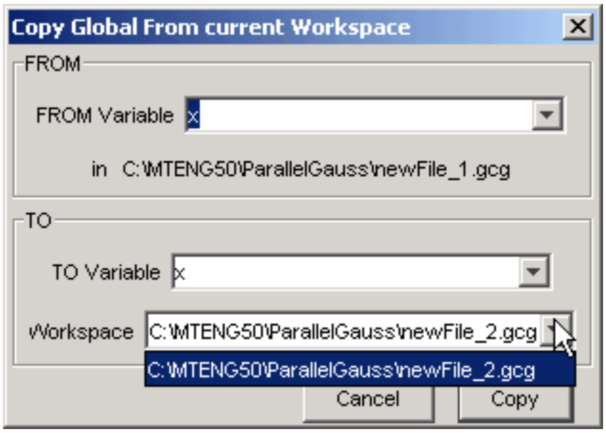
in C:\MTENG50\ParallelGauss\newFile_1.gcg

TO

TO Variable:

Workspace: C:\MTENG50\ParallelGauss\newFile_2.gcg

From the drop down list of workspaces at the bottom of the dialog box select the workspace you want to copy to.



Copy Global From current Workspace

FROM

FROM Variable x

in C:\MTENG50\ParallelGauss\newFile_1.gcg

TO

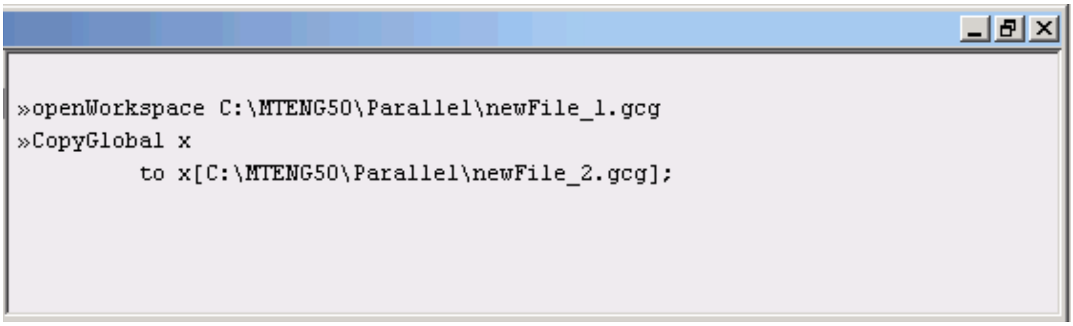
TO Variable x

Workspace C:\MTENG50\ParallelGauss\newFile_2.gcg

C:\MTENG50\ParallelGauss\newFile_2.gcg

Check that the name of the global you are copying to is correct, then click on **Copy**

This will insert a CopyGlobal command into both the source and the destination window. The copy will occur only when both workspaces are ready to execute this command.



```
»openWorkspace C:\MTENG50\Parallel\newFile_1.gcg
»CopyGlobal x
        to x[C:\MTENG50\Parallel\newFile_2.gcg];
```

Copy Global Command

The third way to copy a global is to type the **`CopyGlobal`** command into the Edit pane.

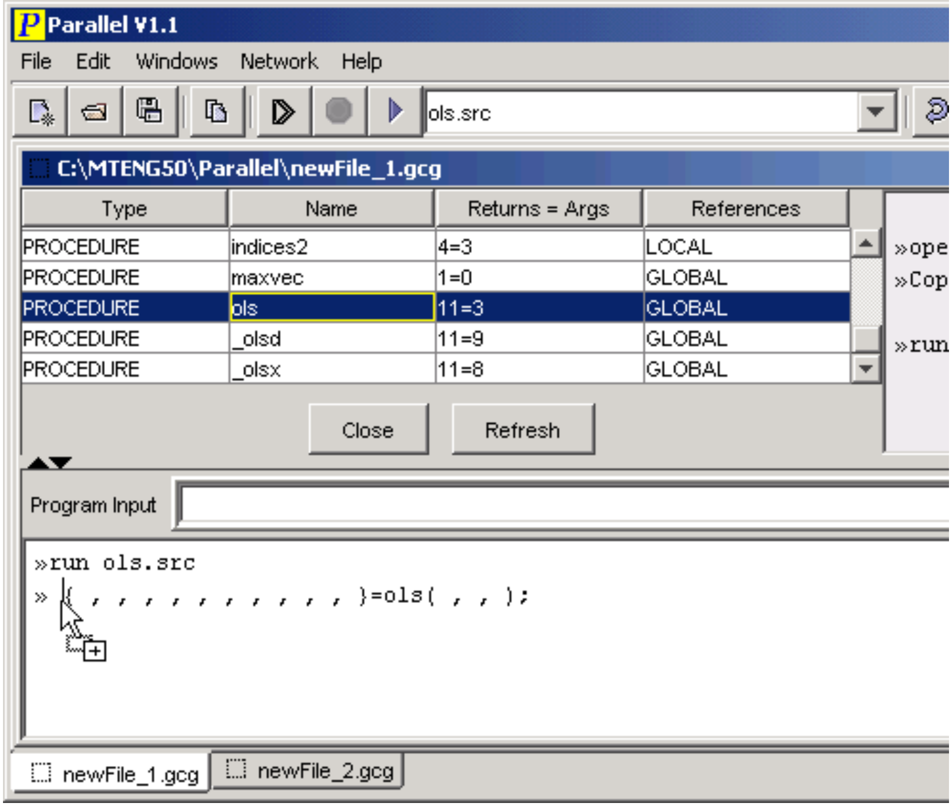
The following command copies the global **x** from the current workspace to a global variable, also called **x**, in workspace **newFile_2.gcg**

```
copyglobal x to x[newFile_2.gcg]
```

For the full syntax of the **`CopyGlobal`** command see the entry in the Commands section.

Other Uses of Drag and Drop

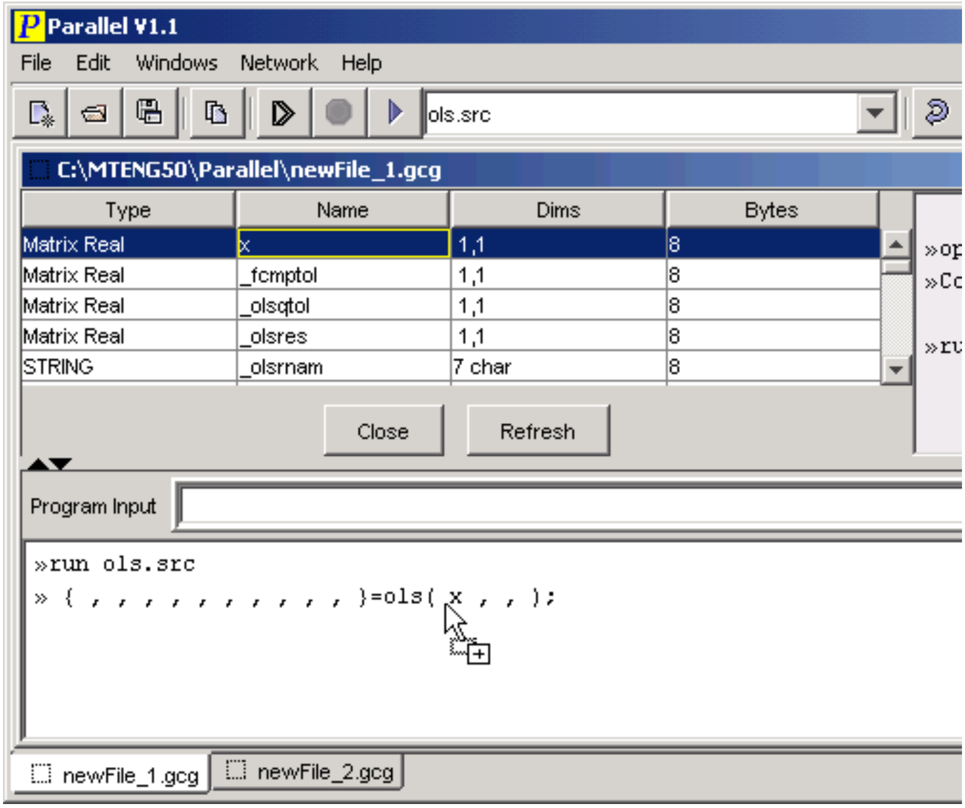
As well as using Drag and Drop to copy globals between workspaces, you can also use it to build GAUSS commands. If you open the Procedures display (**`Edit`**, **`Procedures – Show and Refresh`**), you can drag and drop a procedure into the Edit Pane to create a template call for that procedure.



Type	Name	Returns = Args	References
PROCEDURE	indices2	4=3	LOCAL
PROCEDURE	maxvec	1=0	GLOBAL
PROCEDURE	ols	11=3	GLOBAL
PROCEDURE	_olsd	11=9	GLOBAL
PROCEDURE	_olsx	11=8	GLOBAL

```
»run ols.src
» { , , , , , , , , , , }=ols( , , );
```

You can then open the Globals display (**`Edit`**, **`Globals – Show and Refresh`**) and drag and drop global variables into the argument and return positions to complete the command.



Type	Name	Dims	Bytes
Matrix Real	x	1,1	8
Matrix Real	_fcmptol	1,1	8
Matrix Real	_olsqtol	1,1	8
Matrix Real	_olsres	1,1	8
STRING	_olsrnam	7 char	8

```
»run ols.src
» { , , , , , , , , , , }=ols( x , , );
```

A Simple Example

The next section provides a simple example of running two programs in parallel and using CopyGlobal to combine the results.

A Simple Example

This section shows you how to run two GAUSS programs in parallel and how to use CopyGlobal to combine the results. It assumes you have read the Quick Start and Parallel Programming sections.

Let's assume you have coded two GAUSS programs that run independently of each other but whose results are both needed as input to a third program that produces the final result. One example might be running multiple statistical trials and averaging the results. Another example might be processing two datasets whose results are then used as input to a third GAUSS program. Alternatively you might have one big GAUSS program that you could split into two or more independent parts, that could then be run in parallel before combining the results.

As a simple example in the Parallel directory there are three GAUSS program files:− **prog1.src**, **prog2.src** and **result.src**

prog1.src sleeps for 30 seconds and then assigns **`x1=1;`**
prog2.src sleeps for 30 seconds and then assigns **`x2=2;`**
result.src takes **`x1`** and **`x2`** as inputs and produces the final result **`x1+x2;`**.

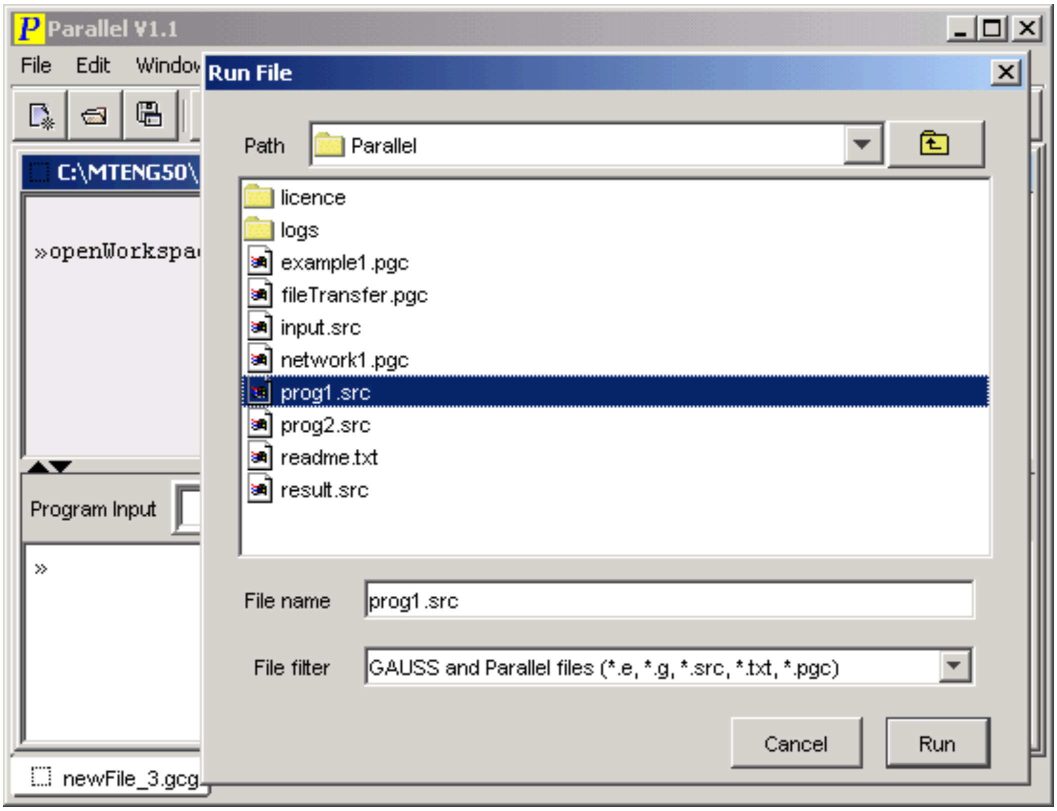
The following steps will run all three programs in Parallel to produce the result while letting **prog1.src** and **prog2.src** run in parallel.

Step 1) Start Parallel

Step 2) Create a new GAUSS workspace, using **`File`**, **`New`**.

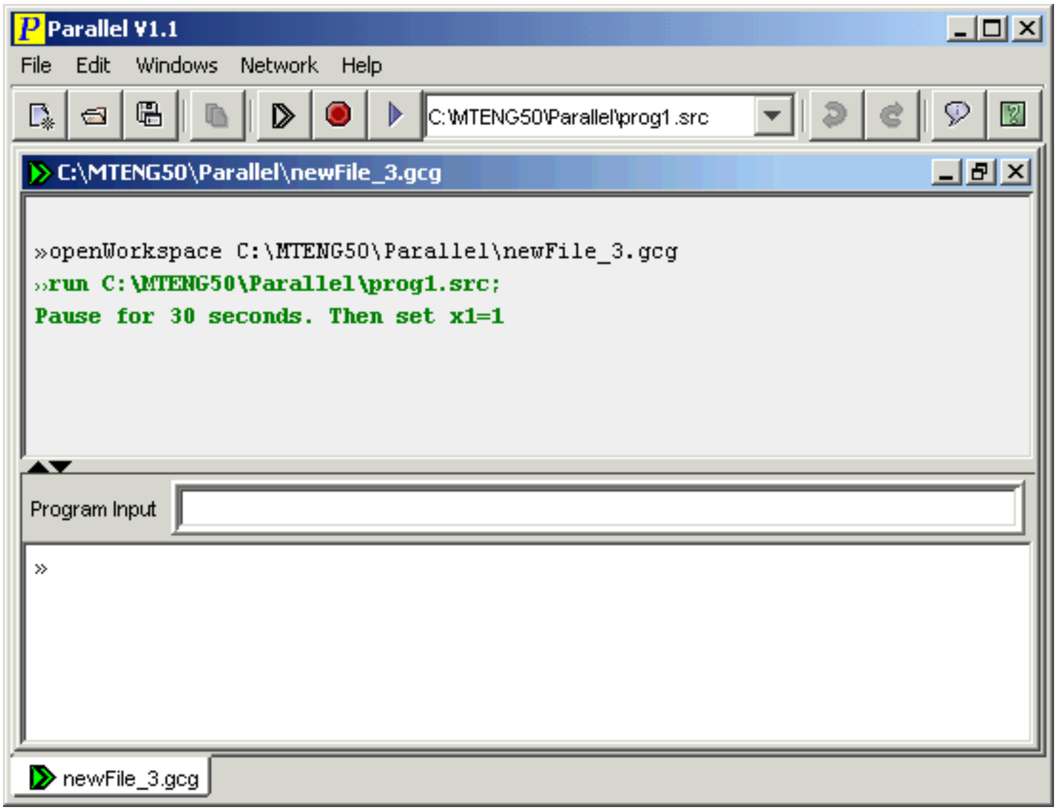
Note: Since two new files have already been created in the Quick Start tutorial, the new workspaces created here will have names newFile_3.gcg and newFile_4.gcg. If you did not do the Quick Start tutorial, create another two new workspaces now and then close them leaving newFile_3.gcg open. The section on Writing Parallel Programs assumes the workspaces created in this section exists.

Step 3) Use **`File`**, **`Run`** to start **prog1.src** running. (Don't wait for it to complete execution. Continue to the next step.)



File name: prog1.src

File filter: GAUSS and Parallel files (*.e, *.g, *.src, *.txt, *.pgc)

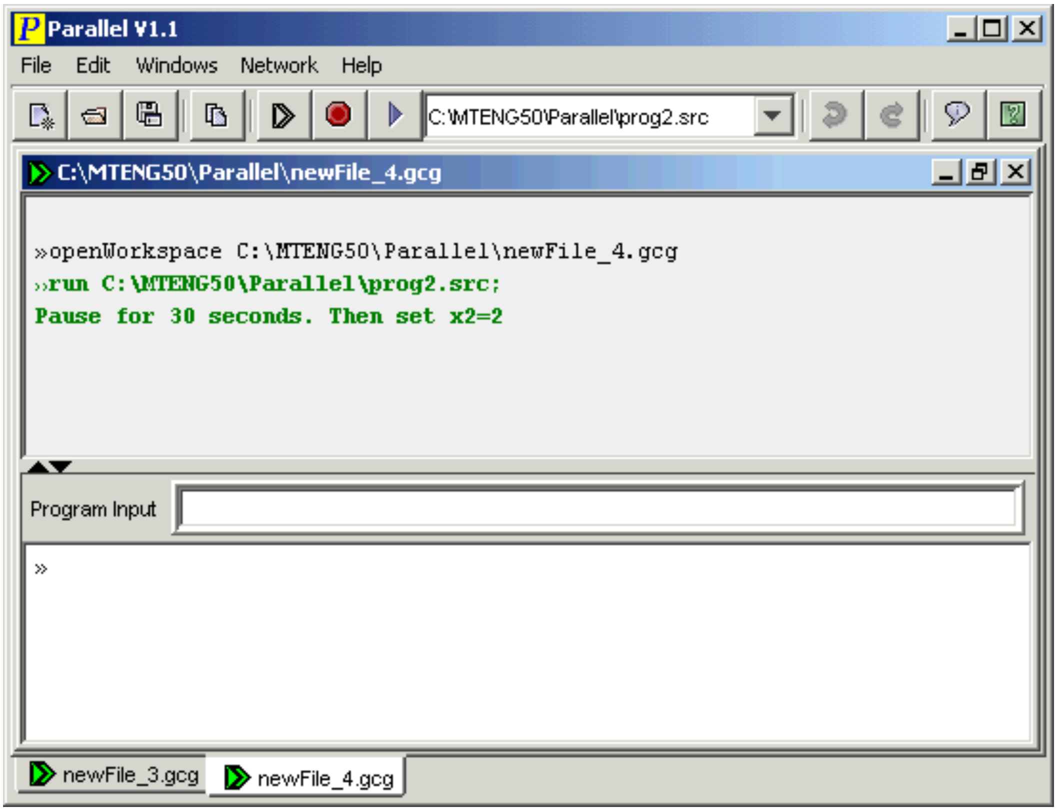


Parallel V1.1
File Edit Windows Network Help
C:\MTENG50\Parallel\prog1.src
C:\MTENG50\Parallel\newFile_3.gcg
Program Input
»
newFile_3.gcg

```
»openWorkspace C:\MTENG50\Parallel\newFile_3.gcg
»run C:\MTENG50\Parallel\prog1.src;
Pause for 30 seconds. Then set x1=1
```

Step 4) Create another GAUSS workspace, using **`File, New`**.

Step 5) Use **`File, Run`** to start **prog2.src** running in the second workspace.

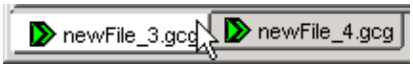


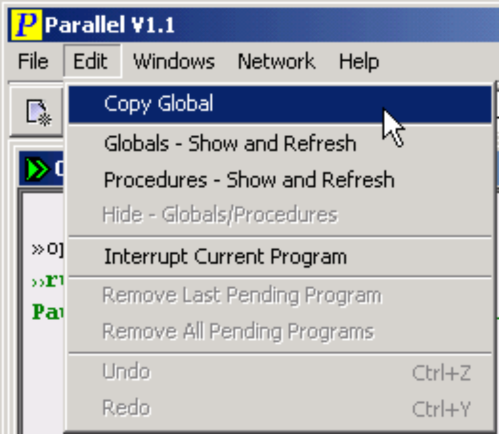
```
»openWorkspace C:\MTENG50\Parallel\newFile_4.gcg
»run C:\MTENG50\Parallel\prog2.src;
Pause for 30 seconds. Then set x2=2
```

```
»
```

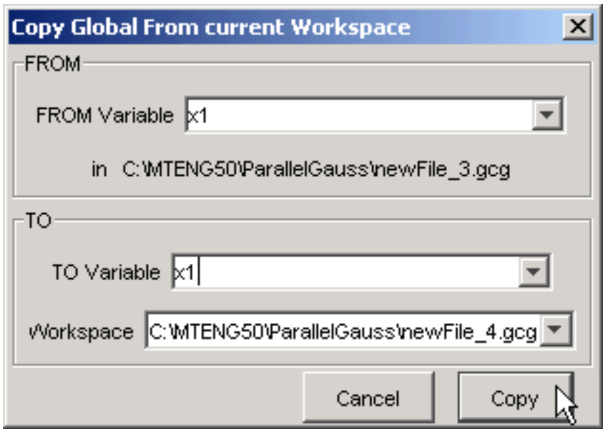
prog2.src runs concurrently with **prog1.src** but in different workspaces.

Step 6) Switch back to newFile_3 and use **`Edit, Copy Global`** to copy the global variable **`x1`** from workspace newFile_3 to **`x1`** in workspace newFile_4.




Parallel V1.1
File
Edit
Windows
Network
Help
Copy Global
Globals - Show and Refresh
Procedures - Show and Refresh
Hide - Globals/Procedures
Interrupt Current Program
Remove Last Pending Program
Remove All Pending Programs
Undo Ctrl+Z
Redo Ctrl+Y
»o
»r
Pa

This opens the Copy Global dialog box. Since the global variable **`x1`** has not been created yet, it does not appear in the From Variable drop down list. So you have to type it in. Check the other entries are as shown below and click **Copy**.



FROM

FROM Variable x1

in C:\MTENG50\ParallelGauss\newFile_3.gcg

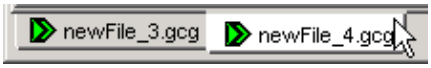
TO

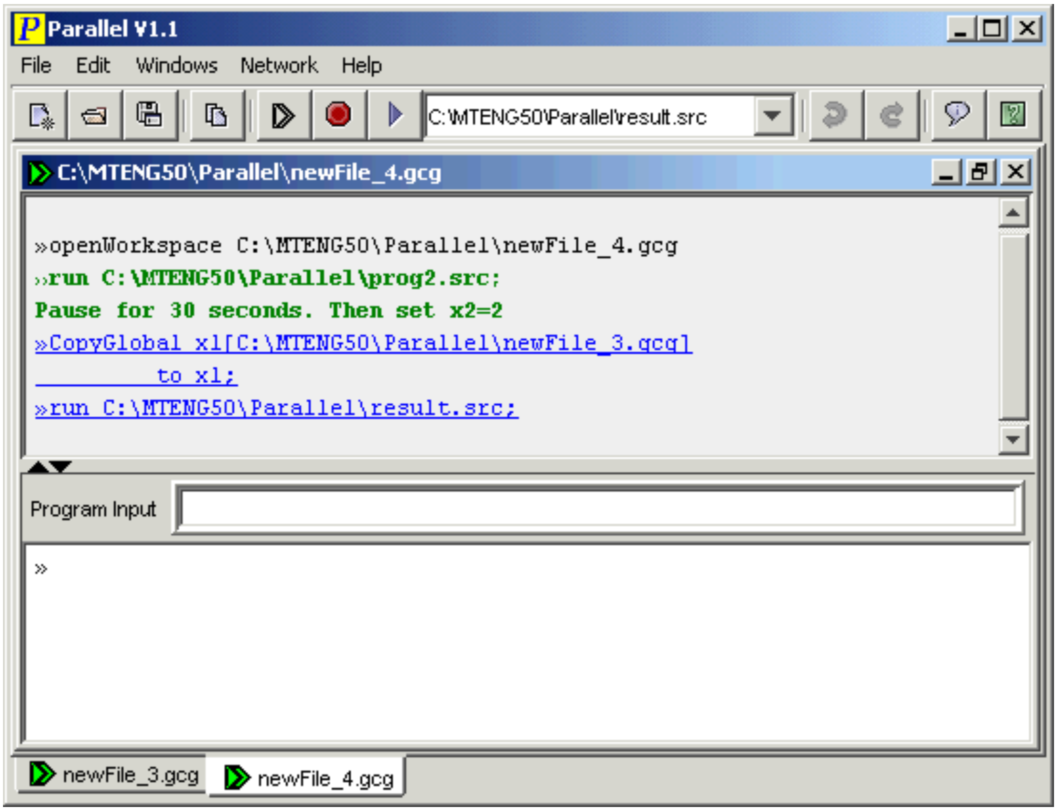
TO Variable x1

Workspace C:\MTENG50\ParallelGauss\newFile_4.gcg

This adds a **`CopyGlobal`** program to both workspaces. The **`CopyGlobal`** command waits for both **prog1.src** and **prog2.src** to finish before it copies **`x1`**.

Step 7) Finally switch back to newFile_4 workspace, where **`x1`** will be copied to, and then run the **result.src** program to produce the final answer.

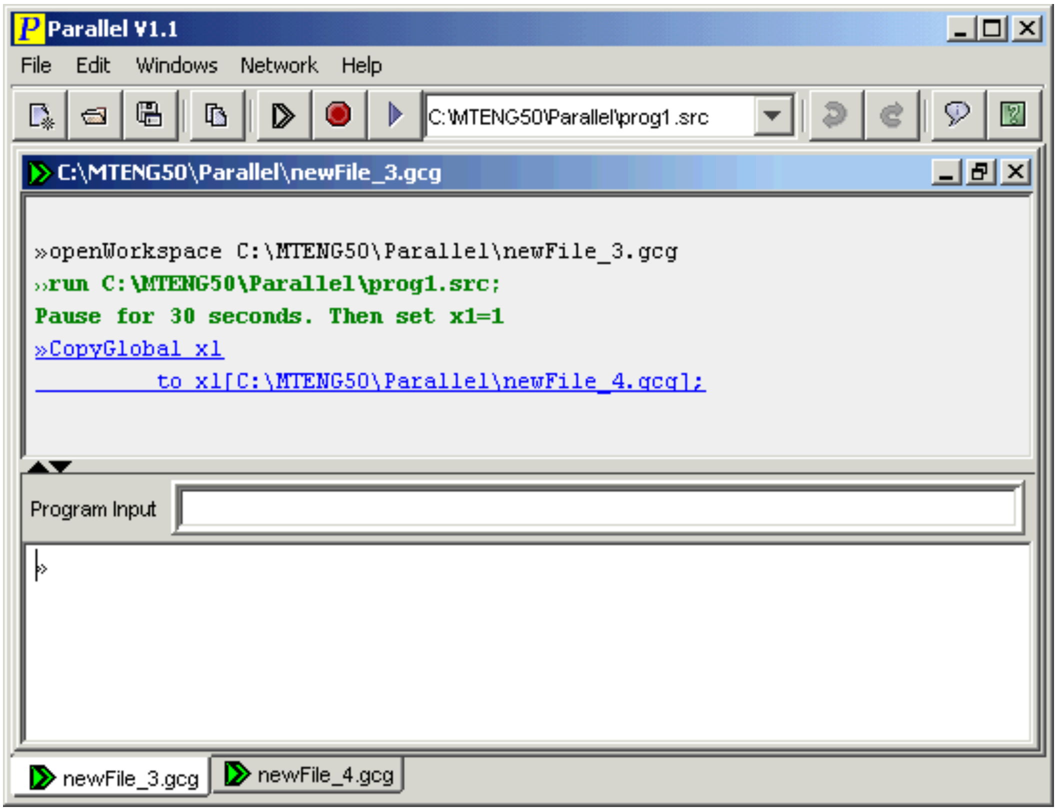




```
»openWorkspace C:\MTENG50\Parallel\newFile_4.gcg
»run C:\MTENG50\Parallel\prog2.src;
Pause for 30 seconds. Then set x2=2
»CopyGlobal x1[C:\MTENG50\Parallel\newFile_3.gcg]
         to x1;
»run C:\MTENG50\Parallel\result.src;
```

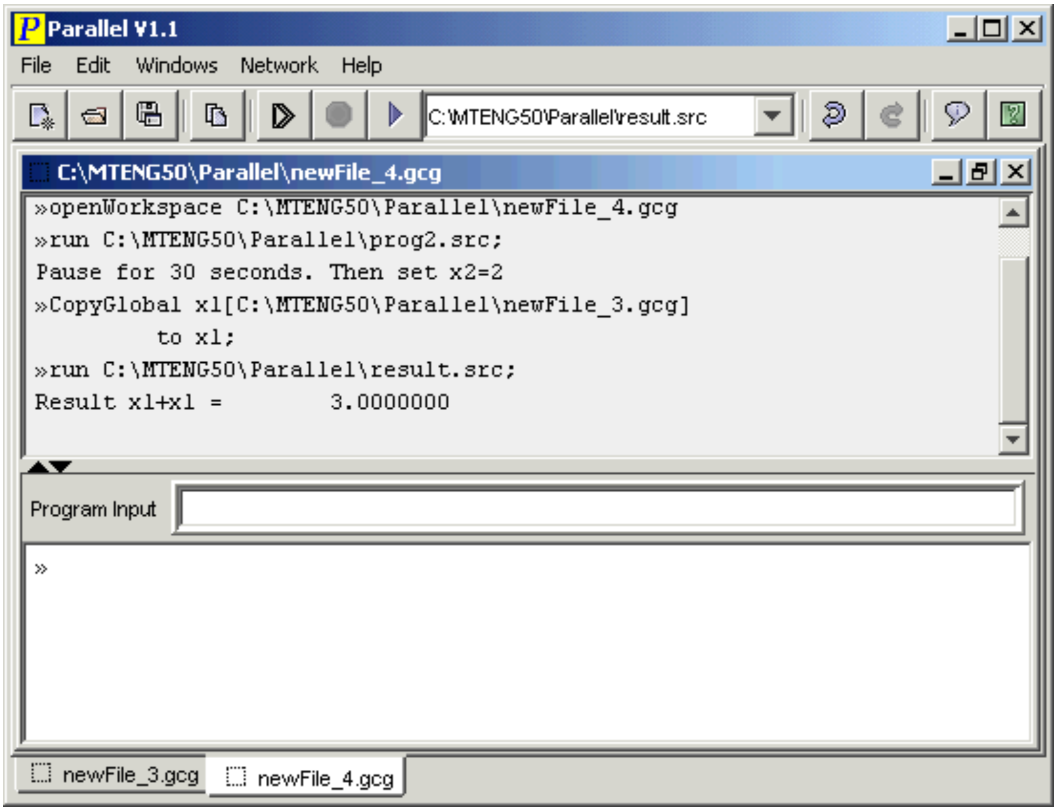
```
»
```

If you switch back to newFile_3.gcg you will see the CopyGlobal command waiting to execute.


Parallel V1.1
File Edit Windows Network Help
C:\MTENG50\Parallel\prog1.src
C:\MTENG50\Parallel\newFile_3.gcg
Program Input
newFile_3.gcg
newFile_4.gcg

```
»openWorkspace C:\MTENG50\Parallel\newFile_3.gcg
»run C:\MTENG50\Parallel\prog1.src;
Pause for 30 seconds. Then set x1=1
»CopyGlobal x1
          to x1[C:\MTENG50\Parallel\newFile_4.gcg];
```

The GREEN programs are currently executing. The BLUE (underlined) programs are waiting to execute. When the **`CopyGlobal`** program is running in both workspaces, **`x1`** is copied from newFile_3 to the newFile_4 workspace. Only then is the **result.src** program executed in the newFile_4 workspace.



```
»openWorkspace C:\MTENG50\Parallel\newFile_4.gcg
»run C:\MTENG50\Parallel\prog2.src;
Pause for 30 seconds. Then set x2=2
»CopyGlobal x1[C:\MTENG50\Parallel\newFile_3.gcg]
         to x1;
»run C:\MTENG50\Parallel\result.src;
Result x1+x1 =        3.0000000
```

```
»
```

Summary

This simple example ran two GAUSS programs in parallel, in seperate workspaces. While they were still running, a CopyGlobal command was added to copy the result from newFile_3.gcg to newFile_4.gcg. This CopyGlobal command synchronized the execution of the two workspaces.

The **result.src** program was queued after the CopyGlobal command to produce the final result.

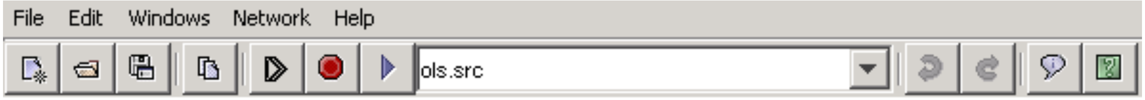
What Next?

Once you have run a set of parallel programs by hand you can automate the process using a Parallel command file. This command file is created by copying the entries in the command log.

See Writing Parallel Programs for more details.

Menus and Toolbar

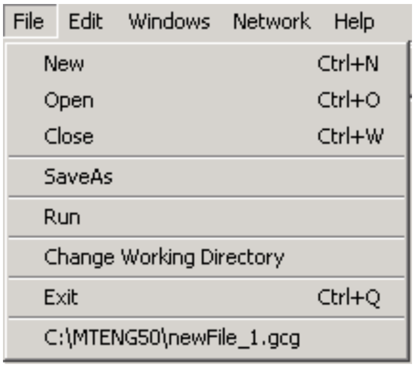
Parallel has the following main menus and toolbar:−



The toolbar icons are shown below next to their respecive menu items.

File Menu

The **`File`** menu has the following sub−menus:−



File, New

Creates a new GAUSS workspace in the current directory and and gives it a unique name.

File, Open

Opens an existing GAUSS workspace.

File, Close

Closes the currently selected workspace window. When the workspace is closed it is added to the bottom of the **File** menu and can be selected from there to re−open it. When a workspace is re−opened, the **Edit, Undo** / **Edit, Redo** list is restored also. You can also manually recover previous workspace states. See the Command Log section for more details.

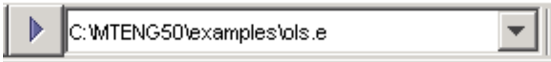
File, SaveAs

Save the currently selected workspace under another name. If you choose a name that already exists you will replace the existing workspace with this current workspace. However you can use **Edit Undo** to reverse this if you change your mind. You can also consult the Command Log to find the name of the copy of the overwritten workspace.

File, Run

Opens a Run File dialog box which displays the GAUSS program files and Parallel command files. (**.pgc**). Selecting one of these files and clicking the Run button in the dialog box, runs the file in the current workspace.

On the toolbar there is a drop down list of the recently run files for current selected workspace.



C:\MTENG50\examples\ols.e

To re−run one of these files select the file from the drop down list and then click the run button, ▶.

File, Change Working Directory

Opens a File dialog box from which you can choose the new working directory for GAUSS on the computer running this workspace.
Warning: This command executes **chdir**. The GAUSS commands **chdir** and **changedir** are not multi−workspace safe. Both these commands change the current working directory for **all** workspaces running on this computer as soon as they are executed in any one of the workspaces. This can cause programs running other workspaces to fail if they are trying to open file in the current directory. A work around is to always use full path names when reading and writing files in GAUSS programs. When running files use the File, Run menu command as this will include the full path of the file to be run.

Parallel incorrectly adds these commands to Edit Undo/Redo lists but the undo/redo does not change the current directory. When GAUSS is updated to support a seperate current directory for each workspace, these problems will go away.

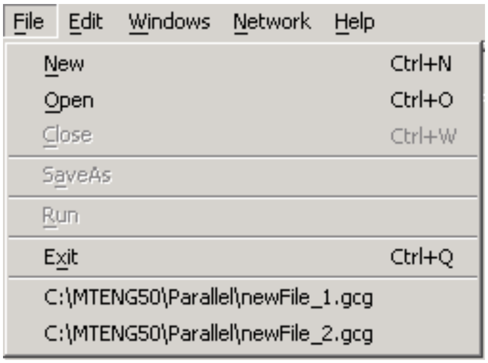
File, Exit

Exits Parallel. All open workspaces are closed.

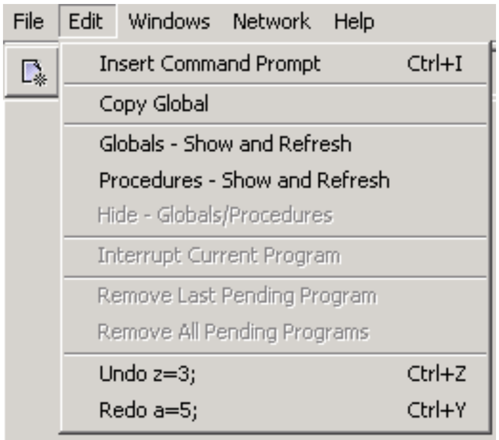
File, *filename*

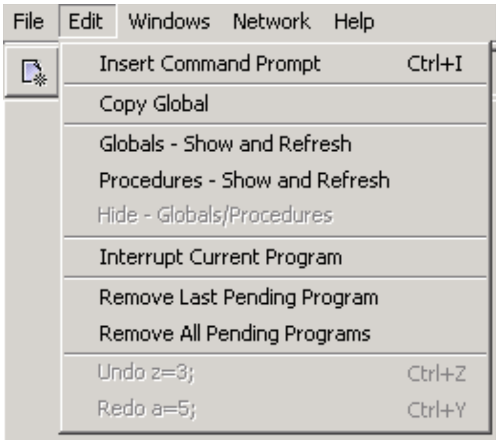
When you close a workspace, it is added here on the bottom of the File menu. When you select one of these files, the workspace is reopened and the undo/redo list is restored.

When there are no workspaces open, the commands **`Close`**, **`SaveAs`** and **`Run`** are not available



Edit Menu


File
Edit
Windows
Network
Help
Insert Command Prompt Ctrl+I
Copy Global
Globals - Show and Refresh
Procedures - Show and Refresh
Hide - Globals/Procedures
Interrupt Current Program
Remove Last Pending Program
Remove All Pending Programs
Undo z=3; Ctrl+Z
Redo a=5; Ctrl+Y


File
Edit
Windows
Network
Help
Insert Command Prompt Ctrl+I
Copy Global
Globals - Show and Refresh
Procedures - Show and Refresh
Hide - Globals/Procedures
Interrupt Current Program
Remove Last Pending Program
Remove All Pending Programs
Undo z=3; Ctrl+Z
Redo a=5; Ctrl+Y

Edit, Insert Command Prompt

Inserts the command start symbol (») on a new line in the Edit Pane, at the current cursor location.

Edit, Copy Global

Copies the value of a GAUSS global variable from the current workspace to another workspace. See the Commands section for detailed examples.

Edit, Globals − Show and Refresh

Opens a table of GAUSS global Strings, Matrices and String Arrays that exist in this workspace. If a GAUSS program is currently running, Show Globals will not show any globals that are currently being created. To see them, wait until the program completes and choose this menu item again to refresh the table of globals.

This table can be use to copy global variables from one workspace to another by draging from this table and dropping on one of the workspace tabs. See Copy Global Drag and Drop for a detailed example. You can also drop a global name from this table into the command edit pane to assist in building a command or to fill−in the arguments or returns for a procedure.

You can sort the columns of this table by clicking on the column heading/s (**`Type`**, **`Names`**, **`Dims`** and/or **`Bytes`**). Each time you click you reverse the sorting order for that column.

Note: Sorting the **`Dims`** column is not very useful as it contains a mixture of numbers and words and is sorted as characters. This means dimensions 4,1 will sort below 301,300.

You can do a multi−column sort by sorting the required columns in **opposite** sequence. For example, if you want to sort by **`Type`**, then within the same **`Type`** sort the rows by **`Bytes`** and then within the same number of bytes, sort by **`Name`**. You first sort the **`Name`** column, then sort the **`Bytes`** column and finally sort the **`Type`**. This ensures that the final table is sorted by **`Type`** but within **`Type`**, the sorting is by **`Bytes`** and then by **`Name`**.

In the table, you can re−arrange the posititon of the columns by dragging the column headings left or right to move them in relation to the other columns.

Edit, Procedures − Show and Refresh

Opens a table of GAUSS Procedures, Functions and Keywords that exist in this workspace. If a GAUSS program is currently running, Show Procedures will not show any procedures that are currently being created. To see them, wait until the program completes and choose this menu item again to refresh the table of procedures.

This table can be used to drag and drop a procudure name from this table into the command edit pane to assist in building a command. When it is dropped a template for calling that procedure will be created in the edit pane. You can then either type in the arguments and returns or you can open the globals table and drag and drop them from there.

You can sort the columns of this table by clicking on the column heading/s (**`Type`**, **`Name`**, **`Retruns`**

`= Args` and/or **`References`**). Each time you click you reverse the sorting order for that column.

Note: Sorting the **`Returns = Args`** column is not very useful as it contains a mixture of numbers and words and is sorted as characters. This means dimensions 1=0 will sort below 11=3.

You can do a multi−column sort by sorting the required columns in **opposite** sequence. For example, if you want to sort by **`Type`**, then within the same **`Type`** sort the rows by **`Name`**, you first sort the **`Name`** column, then sort the **`Type`** column. This ensures that the final table is sorted by **`Type`** but within **`Type`**, the sorting is by **`Name`**.

In the table, you can re−arrange the posititon of the columns by dragging the column headings left or right to move them in relation to the other columns.

Edit, Hide − Globals/Procedures

Closes the table of GAUSS Globals or Procedures.

Edit, Interrupt Current Program

While a program is running in the current workspace, this command is available to interrupt it..

Edit, Remove Last Pending Program

While here are programs queued for execution in the current workspace, this command is available to remove the last one.

Edit, Remove All Pending Programs

While here are programs queued for execution in the current workspace, this command is available to remove all programs queued that have not started to execute.

Edit, Undo

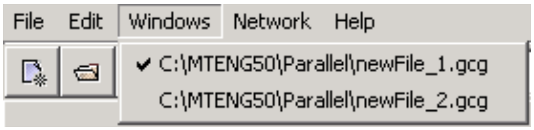
If there are no programs executing in the current workspace, this command is available to undo the program indicated. The workspace is restored to the state it was in prior to the execution of the program. There are multiple levels of undo available.

Edit, Redo

If there are no programs executing in the current workspace and something has been undone, this command is available to redo the program indicated. The workspace is restored to the state is was in after the execution of the program. The program in **not** re−executed.

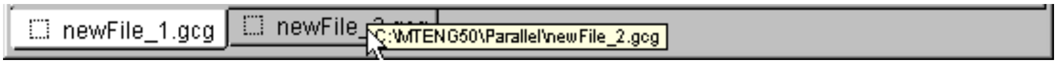
Once you execute a new program the redo stack is cleared. You cannot redo other programs you have undone. However see Command Log for how to recover these other workspaces.

Windows Menu

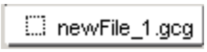

File
Edit
Windows
Network
Help
✔ C:\MTENG50\Parallel\newFile_1.gcg
C:\MTENG50\Parallel\newFile_2.gcg

This menu shows the currently open workspace windows and allows you to select which workspace you want to work with.

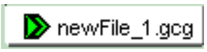
The currently open windows are also shown as tabs at the bottom of the Parallel application window. You can also click on these tabs to switch between workspaces. Tooltips give the full name of the workspace.



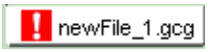
These icons in the tabs show the current state of each workspace.



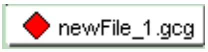
execution completed normally



a program is running

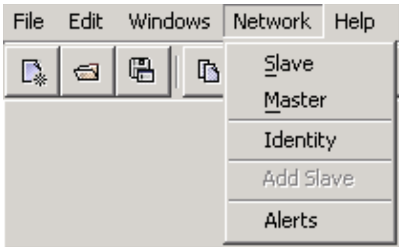


flashing when the program is waiting for user input



the program ended with an error

Network Menu



This menu allows you to set this computer as the Master in a Parallel network or as a Slave. You also use this menu to add remote slave computers to a Parallel network, to display the network identity of this computer and to display the network alerts window.

See the Parallel Networking for a more detailed discussion on setting up a Parallel network.

Network, Master

This sets this computer as the Master computer in a Parallel network. There is only one Master per Parallel network. When you select this menu item the Add Slave dialog box is automatically opened.

Network, Slave

This sets this computer as a Slave computer in a Parallel network. There can be more then one slave computer in Parallel network. Slave computers are controlled by the Master computer.

When you select this menu item you are prompted for the name or IP of the Master computer that will control this slave. Slave computers can only be controlled by one Master computer at a time.

If the Master computer is already running, this slave computer will notify the Master that it is ready to be added to the network. The Master computer will then automatically add this slave computer to the network.

If the Master computer is not in Master mode yet, after attempting to contact the master, this slave computer will just wait for the Master to contact it.

Network, Identity

This displays a dialog box showing the network name and IP address of this computer.

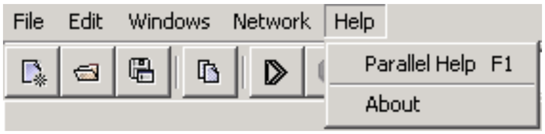
Network, Add Slave

When this computer is in Master mode, this menu item is available to add remote slave computers to the network. To add a remote computer to the network, it must already be in Slave mode and it must have specified this computer as its Master.

Network, Alerts

This displays the Network Alerts window which shows the network status messages.

Help Menu



This menu gives you access to the on−line help, **`Parallel Help`** ?,

and to information about this version of Parallel and this computer's network identity, **`About`** 💬.

Commands

In addition to all the standard GAUSS commands, there are a few new commands in Parallel. These are:−

- **`OpenWorkspace`**
- **`CopyGlobal`**
- **`TransferFile`**
- **`WorkspaceInput`**
- **`WorkspaceSaveAs`**

Each of these commands can only be executed by themselves. They cannot be combined with each other or with other GAUSS commands in the same program. The GAUSS **`run`** command also cannot be combined with any other command in the same program.

A program in Parallel is defined as all commands between any two » symbols.

OpenWorkspace

Purpose: Open an existing GAUSS workspace. If the workspace is already open, just switch to its window.

Format: **`openworkspace`** *filename***;**

Input: *filename* − literal. The GAUSS workspace to be opened. If the *filename* does not end in **.gcg** then **.gcg** is automatically added. The name of the file on the disk must end in **.gcg**

Output: **None.** The workspace is opened in a Parallel workspace window.

Remarks: **OpenWorkspace** must be the only command in the current program. **OpenWorkspace** first tries to match the *filename* given to that of one of the workspaces already open. If there is a unique partial match, then Parallel switches to that workspace window. See the examples below.

Otherwise this workspace file is opened from disk and its undo/redo list is restored.

For remote workspaces, add the remote computer name or IP followed by // to the front of the workspace name.

Examples:
In the following examples assume these workspaces are currently open in Parallel:−

C:\MTENG50\Parallel\newFile_1.gcg
C:\MTENG50\Parallel\logs\newfile_1.gcg
C:\MTENG50\Parallel\newFile_2.gcg

Then the command

```
openworkspace newFile_1;
```

will switch to the open GAUSS workspace, C:\MTENG50\Parallel\newFile_1.gcg. The exact case is matched first.

The command

```
openworkspace newfile_1;
```

will switch to the open GAUSS workspace, C:\MTENG50\Parallel\logs\newfile_1.gcg. The exact case is matched first.

The command

```
openworkspace newfile_2;
```

will switch to the open GAUSS workspace, C:\MTENG50\Parallel\newFile_2.gcg. If there is no exact match, a case insensitive search is done.

The command

```
openworkspace NewFile_1;
```

will return the error

```
»openworkspace NewFile_1;
Could not decide between workspaces
'C:\MTENG50\Parallel\logs\newfile_1.gcg'
'C:\MTENG50\Parallel\newFile_1.gcg'
```

as there is no exact case match, and a case insensitive match produces more then one result.

The command

```
openworkspace logs\NewFile_1;
```

will switch to the open GAUSS workspace, C:\MTENG50\Parallel\logs\newfile_1.gcg.

The command

```
openworkspace NewFile;
```

will try and open the existing GAUSS workspace, C:\MTENG50\Parallel\NewFile.gcg. If this file is not found using the normal operating system search, then an error is returned by GAUSS.

```
»openWorkspace C:\MTENG50\Parallel\NewFile.gcg
(0) : error G0014 : 'C:\MTENG50\Parallel\NewFile.gcg' : File not found
```

The command

```
openworkspace forward//newFile_1;
```

will open the remote GAUSS workspace, C:\MTENG50\Parallel\newFile_1.gcg on the Parallel networked computer named **forward**.

The command
`openworkspace 192.168.0.1//newFile_1;`
will open the remote GAUSS workspace, C:\MTENG50\Parallel\newFile_1.gcg on the Parallel networked computer with the IP address **192.168.0.1**.

CopyGlobal

Purpose: To copy the value of a GAUSS global variable from one workspace to another.

Format:
`copyglobal` *fromVar* **`to`** *toVar*;
`copyglobal` *fromVar*`[`*fromWk*`]` **to** *toVar*`[`*toWk*`];`
`copyglobal` *fromVar* **to** *toVar*`[`*toWk*`];`
`copyglobal` *fromVar*`[`*fromWk*`]` **to** *toVar*`;`

Note: the **[ ]** are part of the command. They enclose the workspace name.

Input:
fromVar − literal, the global variable to copy from.
toVar − literal, the global variable to copy to.
fromWk − literal, the GAUSS workspace to copy from.
toWk − literal, the GAUSS workspace to copy to.

Output: None. The *toVar* is set to the value of the *fromVar*. The *toVar* global is created if it does not exist.

Remarks: **`CopyGlobal`** must be the only command in the current program. **`CopyGlobal`** tries to match the *fromWk* and the *toWk* workspace names using the same matching process as **`OpenWorkspace`**. If either `[`*fromWk*`]` or `[`*toWk*`]` are missing, then the current workspace is assumed for the missing workspace. See the examples below.

When the **`CopyGlobal`** command is entered in a workspace window, it first determines which workspaces are referenced by the command and then a paired **`CopyGlobal`** command is added to program queue in both the from and to workspaces.

In order for the **`CopyGlobal`** command to execute, both the from and to workspaces must be executing the same **`CopyGlobal`** command. This forces one workspace to wait until the other one has reached the paired **`CopyGlobal`** command before the value of the global variable is copied.

To copy globals to, from or between remote workspaces, add the remote computer name or IP followed by // to the front of the workspace name.

Examples:

```
copyglobal x to y;
```

This command is replaced with the equivalent GAUSS statement `x = y;`

```
copyglobal x[newFile_1] to y;
```

This command copies the value of the global variable `x` in the GAUSS workspace **newFile_1.gcg** to the global variable `y` in the current workspace. **Note:** the **[ ]** are part of the command. They enclose the workspace name.

```
copyglobal x to x[newFile_2];
```

This command copies the value of the global variable `x` in the current workspace to the global variable `x` in the GAUSS workspace **newFile_2.gcg**.

```
copyglobal x[newFile_1] to x[newFile_2];
```

This command copies the value of the global variable `x` in the GAUSS workspace **newFile_1.gcg** to the global variable `x` in the GAUSS workspace **newFile_2.gcg**. If the current workspace is neither **newFile_1.gcg** nor **newFile_2.gcg**, this command will not display anything in the **Execution Pane**. Switch to either newFile_1.gcg or newFile_2.gcg to see the queued `CopyGlobal` commands.

```
copyglobal x[forward//newFile_1] to x[192.168.0.3//newFile_1];
```

This command copies the value of the global variable `x` in the remote GAUSS workspace **forward//newFile_1.gcg** to the global variable `x` in the remote GAUSS workspace **192.168.0.3//newFile_2.gcg**.

If the current workspace is neither **forward//newFile_1.gcg** nor **192.168.0.3//newFile_2.gcg**, this command will not display anything in the **Execution Pane**. Switch to either forward//newFile_1.gcg or 192.168.0.3//newFile_2.gcg to see the queued `CopyGlobal` commands.

TransferFile

Purpose: To copy a file from one computer to another.

Format:

`transferfile` *fromFile* **`to`** *toFile*;
`transferfile` *fromFile*`[`*fromWk*`]` **to** *toFile*`[`*toWk*`];`
`transferfile` *fromFile* **to** *toFile*`[`*toWk*`];`
`transferfile` *fromFile*`[`*fromWk*`]` **to** *toFile*`;`

Note: the **[ ]** are part of the command. They enclose the workspace name.

Input:
fromFile − literal, the file to copy.
toFile − literal, the file to copy to.
fromWk − literal, the from workspace.
toWk − literal, the to workspace.

Output: None. The *toFile* on the computer running the *toWk* contains a copy of the *fromFile* from the computer running the *fromWk*. The *toFile* is created if it does not exist. If *toFile* exists it is overwritten. You can use undo to recover the original *toFile*

Remarks: This command is most commonly used to copy data and program files to and from remote computers. To specify a workspace on a remote computer add the remote computer's name or IP followed by // to the front of the workspace name.

`TransferFile` must be the only command in the current program. **`TransferFile`** tries to match the *fromWk* and the *toWk* workspace names using the same matching process as **`OpenWorkspace`**. If either `[`*fromWk*`]` or `[`*toWk*`]` are missing, then the current workspace is assumed for the missing workspace. See the examples below.

When the **`TransferFile`** command is entered in a workspace window, it first determines which workspaces are referenced by the command and then a paired **`TransferFile`** command is added to program queue in both the from and to workspaces.

In order for the **`TransferFile`** command to execute, both the from and to workspaces must be executing the same **`TransferFile`** command. This forces one workspace to wait until the other one has reached the paired **`TransferFile`** command before the file is copied.

If the complete path is not given for the *fromFile* or the *toFile* then the current Parallel directory is used as the starting point. Since there is no way to change the current Parallel directory, complete paths are usually given for the files. See the examples below.

The *fromFile* and the *toFile* paths can contain spaces and \ or /. If the file path includes the keyword **to** in it you can use the following form of the command

transferfile *fromFile***[***fromWk***]**
to *toFile***[***toWk***];**

That is split the command into two lines and start the second line with the keyword **to**. See the examples below.

Examples:

```
transferfile x.src to y.src;
```

This command copies the file `x.src` to the file `y.src` in the current directory of the local computer. **Note:** the current directory for Parallel is usually the GAUSS directory.

```
transferfile c:\My Documents\x.src to y.src;
```

This command copies the file `c:\My Documents\x.src` to the file `y.src` in the current directory of the computer which is running currently active workspace.

```
transferfile c:\files to copy\x.src
to x.src[forward//newFile_1];
```

This command copies the file `c:\files to copy\x.src` from the computer running currently active workspace to the file `x.src` in the current directory of the computer `forward`.

WorkspaceInput

Purpose: To supply keyboard input to GAUSS programs.

Format:
workspaceinput *input*;
workspaceinput *input*;;

Input: *input* − literal or ^string. If the ^string form is used then string must be the name of a global string variable. Its contents will be used as the input.

Output: **None.** The input is added to the **Input Pane** for use by GAUSS programs requiring keyboard input and a newline (\n) is then added to the **Input Pane**. If the command is terminated in ;; then a newline is not added.

Remarks: **WorkspaceInput** must be the only command in the current program.
The characters in the **Input Pane** are accessed by the GAUSS commands, **key**, **keyw**, **keyav**, **cons**, **con**, **wait** and **waitc**.
All carriage return characters (\r) are removed. There is no support for entering a carriage return

character (\r) into the **Input Pane**. Any newlines (\n) are displayed in the **Input Pane** as the symbol **»**. There are no escape sequences and any `"` in the input are just copied directly to the **Input Pane**. Use the ^string version of the command if you need to input leading or trailing blanks.

Examples:

```
workspaceinput     this is the    input     ;
```

adds the following characters to the **Input Pane**

```
this is the    input»
```

the leading and trailing blanks are removed but the internal spaces are retained. A terminating newline character (\n) is added. This newline character is diplyed as the **»** symbol in the **Input Pane**.

```
workspaceinput some more input;;
```

adds the following characters to the **Input Pane**

```
some more input
```

no terminating newline character (\n) is added.

```
Workspaceinput ;
```

adds the following characters to the **Input Pane**

»

Just a terminating newline character (\n).

```
Workspaceinput ;;
```

adds a single space character to the **Input Pane**

```
»a=" test ";
»Workspaceinput ^a;
```

When these two programs are run, they add the following characters to the **Input Pane**

```
 test »
```

The leading and trailing blanks in the string **`a`** are retained.

```
»a=" test ";
»Workspaceinput ^a;;
```

When these two programs are run, they add the following characters to the **Input Pane**

```
 test
```

The terminating newline is not added.

```
»Workspaceinput first line;
»Workspaceinput second line;
```

After these two programs are run the **Input Pane** contains:

```
first line»second line»
```

These two lines of input can then be used by two GAUSS **`cons`** calls.

WorkspaceSaveAs

Purpose: Save the current GAUSS workspace under a new name. If that workspace name already exists, just overwrite it with this one.

Format: `workspacesaveas` *filename*;

Input: *filename* − literal. The new filename for the GAUSS workspace. If the *filename* does not end in **.gcg** then **.gcg** is automatically added.

Output: **None.** The new workspace replaces the current workspace in this Parallel workspace window.

Remarks: **WorkspaceSaveAs** must be the only command in the current program. If the new *filename* already exists, a copy is taken of the existing workspace before replacing it with this one. You can use Undo to recover the workspace that was overwritten or you can consult the command.log in the Parallel/logs sub−directory to find the name of the copy.

Examples:

```
workspacesaveas myWork;
```

will save the current workspace under the file name **myWork.gcg** in the current working directory.

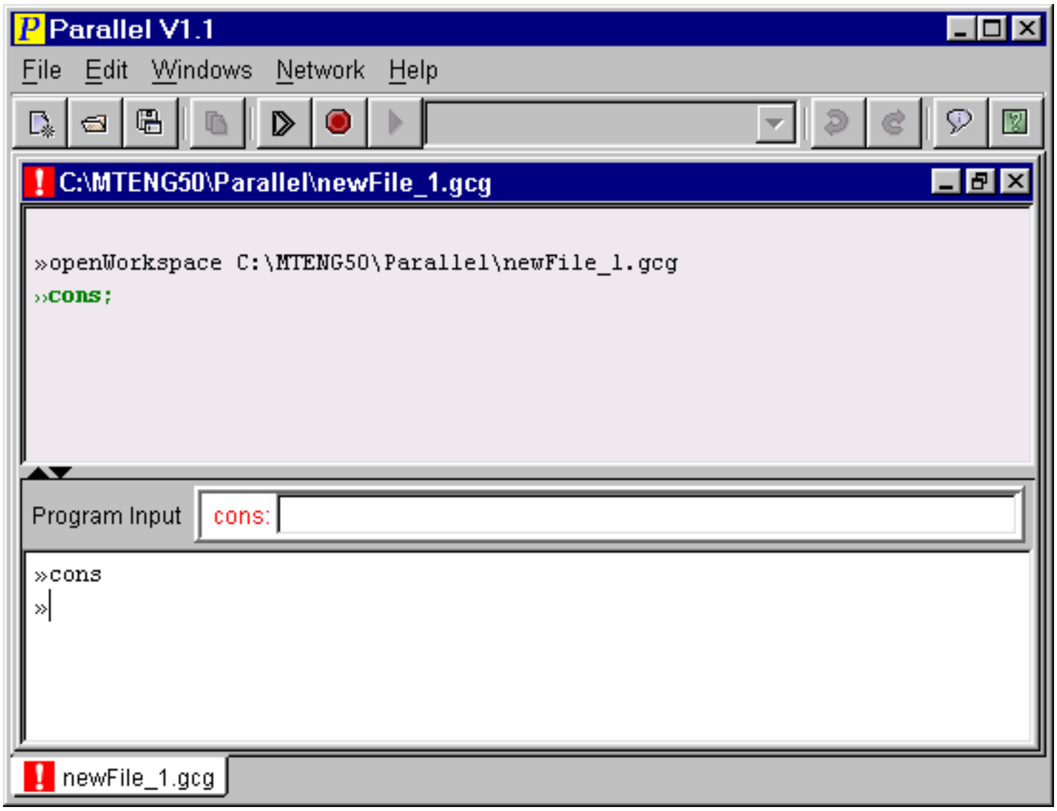
Program Input

When your need to supply keyboard input to a GAUSS program running in Parallel, you enter the input into the **Input Pane**.

Create a new workspace and enter the following GAUSS command.

»cons

The GAUSS **cons** command waits for a line of input from the user.



Parallel V1.1
File Edit Windows Network Help
C:\MTENG50\Parallel\newFile_1.gcg

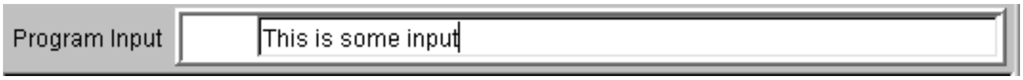
```
»openWorkspace C:\MTENG50\Parallel\newFile_1.gcg
»cons;
```

```
»cons
»
```

Your screen should look like the one above.

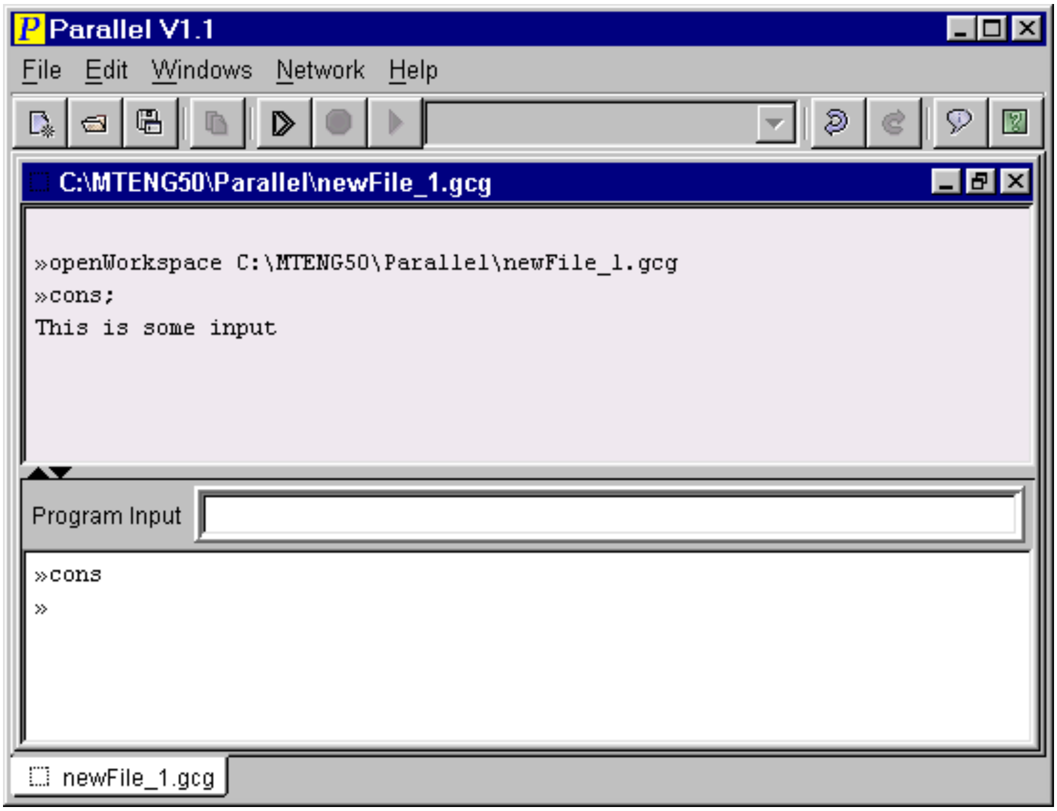
The **BOLD GREEN** **»cons;** in the **Execute Pane** indicates this is the program currently being executed by GAUSS. The flashing cons: in the **Input Pane** indicates that the GAUSS program is waiting for user input to a **cons** function.

Type some characters into the **Input Pane**.



This is some input

Press **Enter** to terminate the **cons** input.



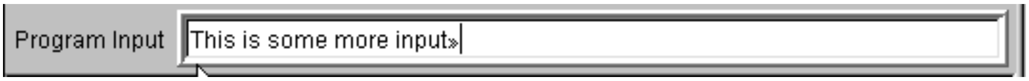
Parallel V1.1
File Edit Windows Network Help
C:\MTENG50\Parallel\newFile_1.gcg

```
»openWorkspace C:\MTENG50\Parallel\newFile_1.gcg
»cons;
This is some input
```

```
»cons
»
```

The GAUSS program in the **Execute Pane** goes BLACK when it has completed execution without errors.

You don't have to wait for the GAUSS program to prompt you for input. You can pre−enter the input in the **Input Pane**. To pre−load the program input, type some more text into the **Input Pane** and press the **Enter** key.



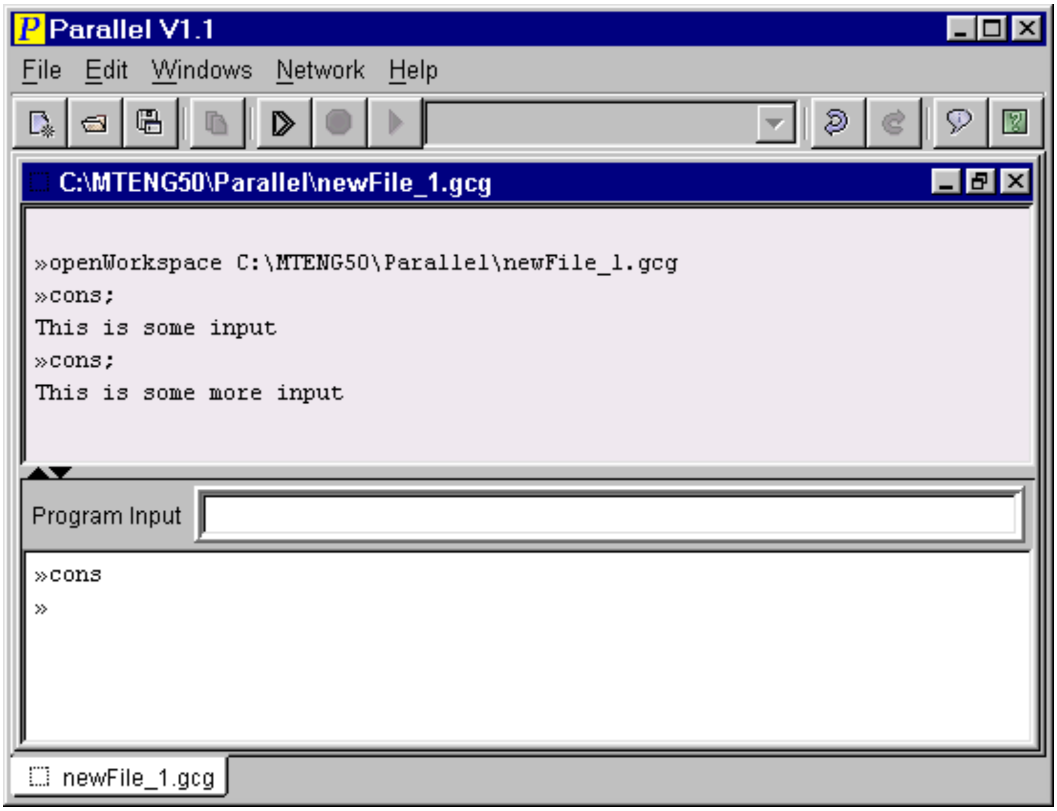
Program Input: This is some more input»

Now re−run the

»cons

»

program by clicking anywhere between the two » symbols and pressing the **Enter** key.



```
»openWorkspace C:\MTENG50\Parallel\newFile_1.gcg
»cons;
This is some input
»cons;
This is some more input
```

```
»cons
»
```

The program is re−run and takes the pre−entered input.

The newline character ('\n') is displayed as a » symbol in the **Input Pane**. You can pre−enter more than one line of input. The » symbols show you where the line breaks are.

Differences between GAUSS for Windows and Parallel

Under GAUSS for Windows, when your GAUSS program is waiting for the user to press **Enter** you use code like

```
print /flush "Press Enter to continue";;
 inkey = keyw;  /* wait for user to Press Enter key */
 if inkey $== vals("\r");    /* if key == Enter, continue */
   .. do something
 else;
   .. do something else
 endif;
```

In Parallel, the Input Pane removes all '\r' characters and instead inserts '\n' when the Enter key is pressed. So you will need to change the above code to

```
print /flush "Press Enter to continue";;
 inkey = keyw;  /* wait for user to Press Enter key */
 if inkey $== vals("\n") or inkey $== vals("\r");    /* if key == Enter, continue */
```

```
      .. do something
    else;
      .. do something else
    endif;
```

Programmed Input

See the WorkspaceInput command for details on how to supply input from within a Parallel command file.

Command Log

All the programs executed by GAUSS in the **Execute Pane** are recorded in a **command.log** file. This log file is found in the **parallel/logs** sub−directory of the user's home directory.

Here is the log file for the Simple Example of parallel programming.

```
 /* Thu 8 May 2003 3:41 PM started Parallel V1.1 May 1st, 2003 */

»openWorkspace C:\MTENG50\Parallel\newFile_3.gcg;

»run C:\MTENG50\Parallel\prog1.src;
 /* C:\MTENG50\Parallel\_P_UNDO\_P_41111.gcg */

»openWorkspace C:\MTENG50\Parallel\newFile_4.gcg;

»run C:\MTENG50\Parallel\prog2.src;
 /* C:\MTENG50\Parallel\_P_UNDO\_P_17074.gcg */

»CopyGlobal x1[C:\MTENG50\Parallel\newFile_3.gcg] to x1;
 /* C:\MTENG50\Parallel\_P_UNDO\_P_38310.gcg */

»run C:\MTENG50\Parallel\result.src;
 /* C:\MTENG50\Parallel\_P_UNDO\_P_29794.gcg */
```

The first line tells you when Parallel was started and the version and date of Parallel. The following lines show the commands that were exectuted. Commands grouped between » » symbols are executed as a single GAUSS program. The comment following the command tells you the name of the undo file for this command. If there is no comment, then there is no additional file required to undo this command.

Writing Parallel Programs

The simplest way to write parallel programs for Parallel is to first execute the commands and programs by hand in Parallel and then edit the resulting command log. In order to run a file as a Parallel program it must have the extension **.pgc**. The **example1.pgc** file contains the lines listed above.

To run this Parallel program (assuming newFile_3.gcg and newFile_4.gcg already exist in the current directory):−

i) Open Parallel
ii) Open or create a GAUSS workspace. It need not be either newFile_3 or newFile_4, any GAUSS workspace will do.
iii) Use **`File`**, **`Run`** to run **example1.pgc** file.

This will open the appropiate workspaces and queue the programs for execution by GAUSS. If **newFile_3.gcg** or **newFile_4.gcg** are not already open in Parallel the **`openWorkspace`** commands will open them. If they are already open in Parallel, **`openWorkspace`** will just switch to that window and continue queuing GAUSS programs to run in that workspace. For more details on the **`openWorkspace`** command see Commands section.

If you have not done the Quick Start and the Simple Example, you may not have the two new workspaces **newFile_3.gcg** and **newFile_4.gcg**. In this case Parallel will stop with an error when it tries to open the non−existent workspaces.

Handling Keyboard Input

The Parallel **`WorkspaceInput`** command allows you to pre−enter keyboard input that your GAUSS programs will read using the GAUSS commands **`key`**, **`keyw`**, **`keyav`**, **`cons`**, **`con`**, **`wait`** and **`waitc`**. In the command log you will see comments indicating the keys that were read by your GAUSS program when it executed. For example:−

```
»/* keyav 't' */
»/* key 't' */
»/* key 'e' */
»/* cons 'st»' */
»run input.src; /* C:\MTENG50\Parallel\_P_UNDO\_P_36491.gcg */
```

This extract shows that the GAUSS program called the **`keyav`** function and received the character **`t`** from the Input Pane. The program then retrieved two keys **`t`** and **`e`** and then called **`cons`** to retrieve the rest of the input line. The **»** indicates that a newline character was returned.

When writing your parallel program you can use these command log comments as a guide to what your GAUSS program requires for keyboard input. You then need to put the appropiate characters in a **`WorkspaceInput`** command. In this example you would use the command
`»workspaceInput test;`

Undo Workspace Files

Parallel provides a multi−level undo/redo facility so that you can reverse the effects of GAUSS programs. The redo history is cleared when you execute a new command, but you can use the information in the command log to redo these commands.

In the command log, at the end of each GAUSS program, there is a comment which identifies the name of the file that holds the undo workspace for this program execution. That is, if you want to restore the GAUSS workspace to the state it was in before the program was run, you would open the workspace shown in the comment. However Parallel will not open these temporary files directly. You first need to rename the file.

For example, to go back to the state of the GAUSS workspace before the **result.src** program was run,

```
 »run C:\MTENG50\Parallel\result.src;  /*
C:\MTENG50\Parallel\_P_UNDO\_P_29794.gcg */
```

you would rename **_P_29794.gcg** to some other file name not starting with **_P_** and then open that workspace in Parallel. Parallel will not open any file that starts with **_P_** or is in a sub−directory called **_P_UNDO**, so you need to copy and rename the saved workspace before you can open it.

For `TransferFile` commands, if the destination file (the toFile) exists, a copy of its original contents is saved before it is replaced with the source file (the fromFile). The file name of that copy is shown in the comment for this command. If there is no comment, then the destination file did not exist and was not overwritten.

To recover from a `TransferFile` command by hand, first check that the file mentioned in the comment exits, then delete the destination file (the toFile) and rename the file mentioned in the comment to name of the destination file.

Parallel Networking

Parallel also allows you to control multiple GAUSS workspaces, spread over multiple computers, from a single Master computer. This allows you to distribute your GAUSS programs over multiple machines and workspaces. Each of these GAUSS workspaces appears in its own window on the Master computer. You can execute GAUSS commands and programs on the remote machines as well as copy globals between any of the workspaces (either local or remote). You can run Parallel command files that automatically copy source and data files to remote computers and then execute GAUSS commands in any of these workspaces and copy back the resulting globals or data files.

Security Issues in a Parallel Network

In a Parallel network the Master computer has complete control of GAUSS running on the Slaves. Since GAUSS on the slave computers can run programs and create, replace and delete files on the slave computers, any computer the slave accepts as its Master has essentially unlimited power over the slave computer. This raises security issues which you should review before setting up a Parallel network.

The Master and Slave computers run as "trusted" hosts for each other and are identified by their IP address.

On networks with poor security and/or running insecure operating systems, Parallel in Network mode is vulnerable to "IP Spoofing" attacks which, if sucessful, could allow an attacker to transfer files to either the Master or Slave computer and run arbitrary GAUSS commands.

Refer to http://www.sans.org/rr/threats/intro_spoofing.php for an introduction to IP Spoofing and to http://www.cert.org/advisories/CA−1996−21.html for a CERT advisory on IP Spoofing.

Proper design of network routers and firewalls and operating systems can guard against such attacks.

It is your responsibility to ensure your network and operating systems have been upgraded to prevent this type of attack.

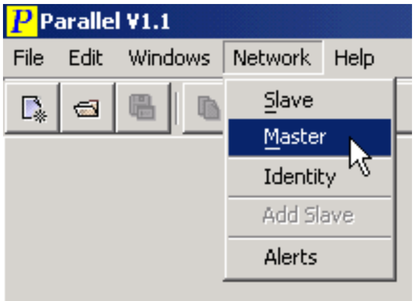
Parallel may also be vulnerable to other forms of attack. Parallel is **NOT** warranted to be secure against any form of attack. You are advised to only use Parallel on secure networks running secure operating systems.

Setting up a Parallel Network

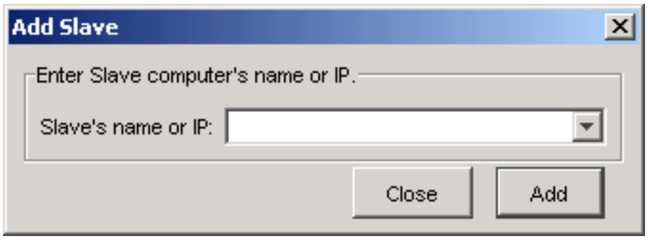
This section will take you through setting up a Parallel Network.

To set up a Parallel network:−

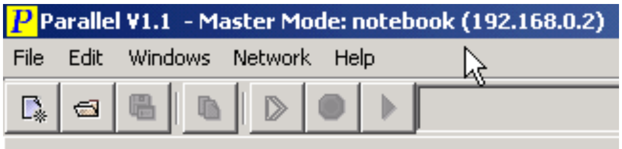
1. Install GAUSS and Parallel on two or more computers that are connected via a network. Check that each installation of Parallel is running correctly as a stand alone version.
2. Choose one of the computers to be the Master computer. The other computers will all be Slaves to this Master. There is only one Master computer in a Parallel network. The Master computer has complete control over the copy of Parallel running on each of the Slave computers.
3. On the Master computer select the **`Network`**, **`Master`** menu option.



This will then display the Add Slave dialog box

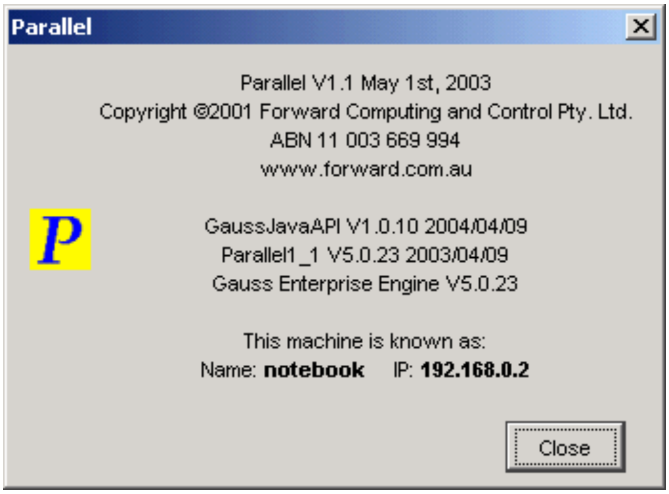


Since none of the Slave computers are running Parallel in Slave mode yet just close this dialog box for now. The Parallel Application title bar now shows the name and IP of this Master computer.



In this example the computer's network identity is **notebook** and its IP address in **192.168.0.1**

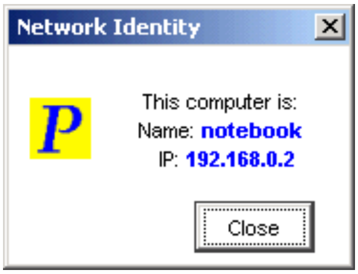
The network identity of any Parallel computer can be seen in either the **`Help`**, **`About`** box or the **`Network`**, **`Identity`** box.



Parallel V1.1 May 1st, 2003
Copyright ©2001 Forward Computing and Control Pty. Ltd.
ABN 11 003 669 994
www.forward.com.au

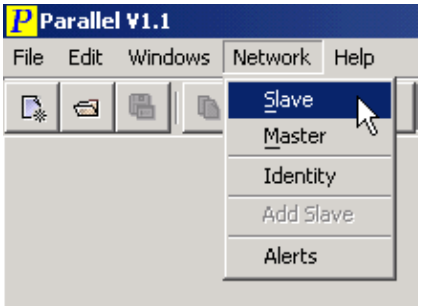
GaussJavaAPI V1.0.10 2004/04/09
Parallel1_1 V5.0.23 2003/04/09
Gauss Enterprise Engine V5.0.23

This machine is known as:
Name: **notebook** IP: **192.168.0.2**

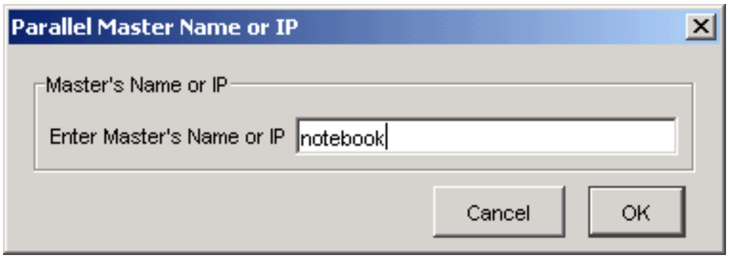


This computer is:
Name: **notebook**
IP: **192.168.0.2**

4. Having set the local computer as the Master, go to each of the remote computers and select the **`Network`**, **`Slave`** menu item.

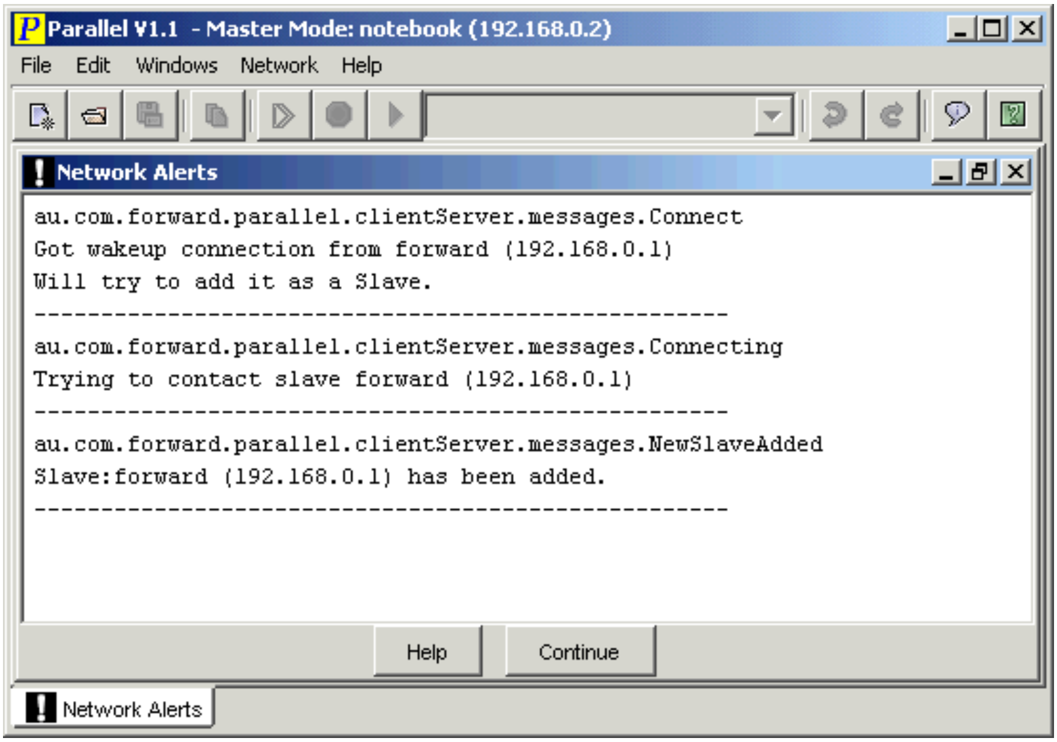


This will open the **Parallel Master Name or IP** dialog box where you specify which computer is the master in this network. In this example the master computer is **notebook**. You could also enter the master computer's IP number (**192.168.0.2**) instead of its name if you wish.



Enter Master's Name or IP: notebook

When you press OK, the remote computer contacts the master and alerts the master that it is now available to be added as a slave to the Parallel network. The master computer then automatically runs the **`Network`**, **`Add Slave`** menu command to add this new slave to the Parallel network. Both the local and remote computers display a Network Alerts window to keep you informed of changes in the network status. For example on the master computer this looks like



```
au.com.forward.parallel.clientServer.messages.Connect
Got wakeup connection from forward (192.168.0.1)
Will try to add it as a Slave.
----------------------------------------------------
au.com.forward.parallel.clientServer.messages.Connecting
Trying to contact slave forward (192.168.0.1)
----------------------------------------------------
au.com.forward.parallel.clientServer.messages.NewSlaveAdded
Slave:forward (192.168.0.1) has been added.
----------------------------------------------------
```

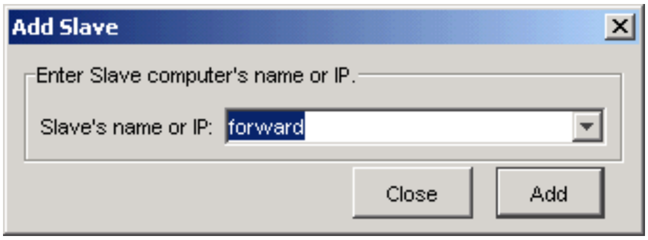
You can click on Continue to dismiss this window and stop the flashing alert.

You now have a functioning Parallel network. If you get error messages about not being able to contact the master or slave, or if there is a long delay in establishing the connection (especially on a Window's network) then check that your network IP naming service is correctly set up (refer to IP Networkng Tips for some basic hints).

Adding Existing Slaves

In the above example the master computer was first put into Parallel Master mode and then Parallel on the remote computers was put into Parallel Slave mode. This automatically added that slave computer to the network.

However if the remote computer is already in Parallel Slave mode when the master computer is put into Parallel Master mode, you have to manually add the slave computer to the Parallel network. To do this select the **`Network`**, **`Add Slave`** menu item and enter the network name or IP of the remote computer you wish to add.



Slave's name or IP: forward

The drop down list in the Add Slave dialog box contains the names or IPs of remote computers you have previously added.

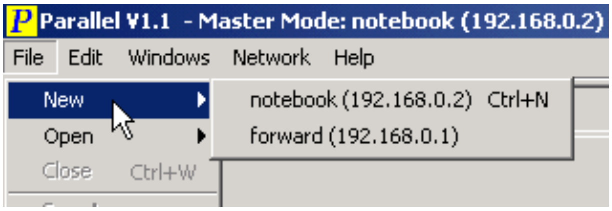
You can find a remote computer's name and IP by selecting its **`Network`**, **`Identity`** menu item or selecting its **`Help`**, **`About`** menu item.

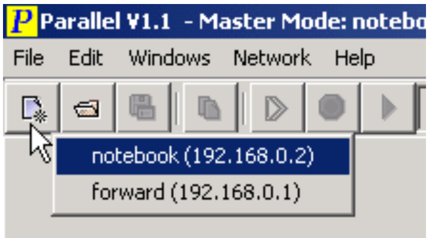
Creating New Remote Workspaces and Openning Existing Ones

Now that you have a Parallel network running, from the local (master) computer you can create new workspaces on the remote (slave) computers or open workspaces that already exist on the remote (slave) computers.

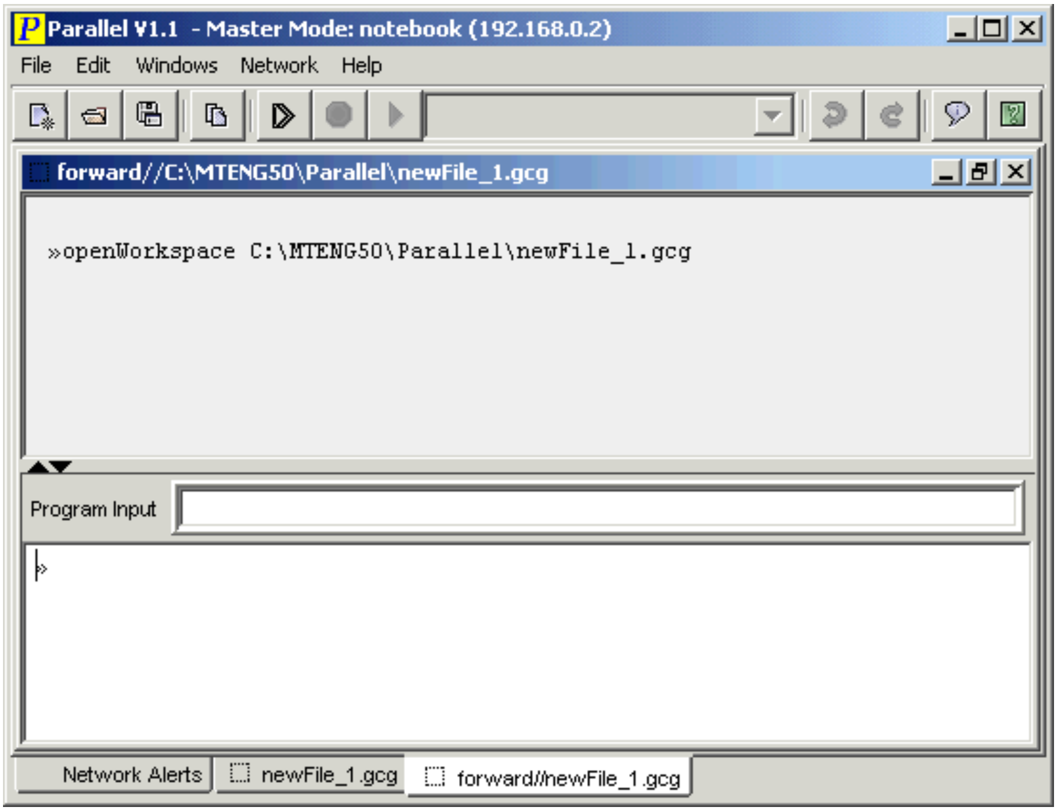
Creating New Remote Workspace

To create a new workspace you use the **`File`**, **`New`** menu item or click on the New Workspace icon, 🗋. This displays a list of the computers currently in the Parallel network. The top computer in this list is always the local (master) computer. That is the computer you are using. The other computers in the list are remote (slave) computers.


Parallel V1.1 - Master Mode: notebook (192.168.0.2)
File
Edit
Windows
Network
Help
New
Open
Close Ctrl+W
notebook (192.168.0.2) Ctrl+N
forward (192.168.0.1)



To create a new workspace on a remote (slave) computer, just select it from the list. For example selecting **`forward (192.168.0.1)`** creates a new GAUSS workpspace on **forward**.


Parallel V1.1 - Master Mode: notebook (192.168.0.2)
File
Edit
Windows
Network
Help
forward//C:\MTENG50\Parallel\newFile_1.gcg
Program Input
Network Alerts
newFile_1.gcg
forward//newFile_1.gcg

```
»openWorkspace C:\MTENG50\Parallel\newFile_1.gcg
```

As you can see from the screen shot above, the names of workspaces that reside on remote computers are prefixed by the name of the computer followed by // and then the file name on the remote computer, in this example **forward//C:\MTENG50\Parallel\newFile_1.gcg**

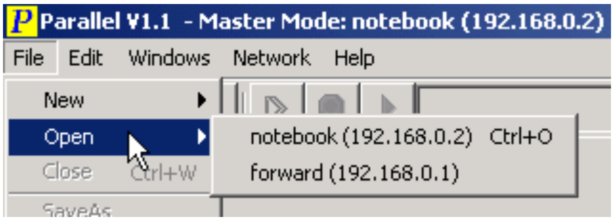
Workspaces that reside on the local computer do not have a *computername//* in front of the filename. That is local workspaces look exactly the same as they do when you are not in networking mode.

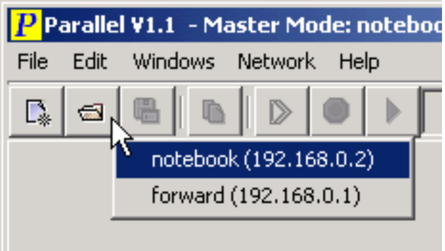
If you go to the remote (slave) computer you will see **C:\MTENG50\Parallel\newFile_1.gcg** open there, but without the **forward//** in front of the workspace name because the workspace is local to that slave computer.

Now that you have a workspace open on the remote computer you can run GAUSS commands and programs on the remote computer by typing them into the command pane for that workspace on the master computer.

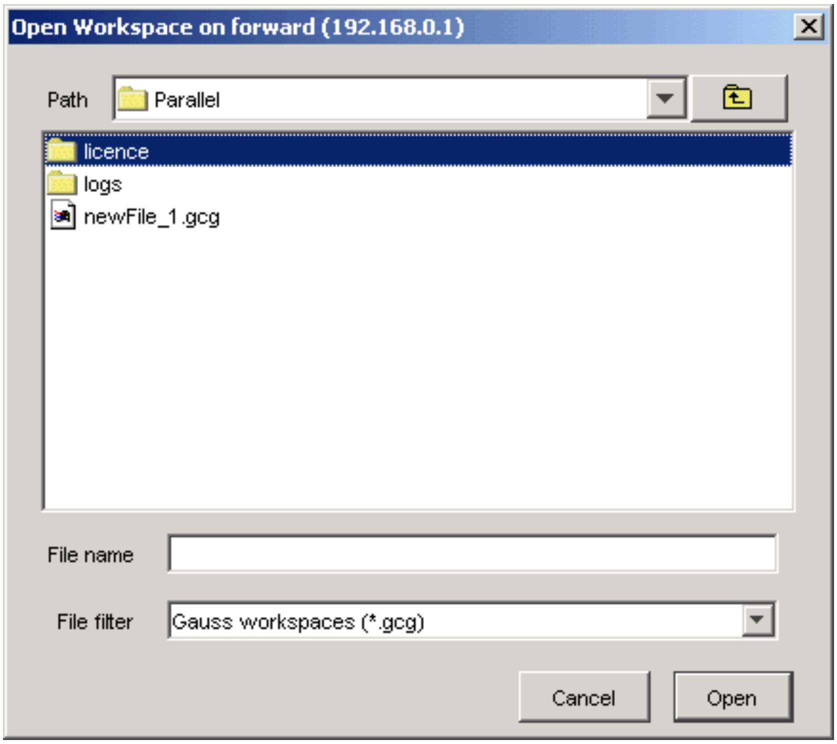
Opening an Existing Remote Workspace

To open an existing workspace on a remote (slave) computer you select the **`File`**, **`Open`** menu item or click on the Open Workspace icon, 📂. This displays a list of the computers currently in the Parallel network. The top computer in this list is always the local computer. That is the computer you are using. The other computers in the list are remote computers.


Parallel V1.1 - Master Mode: notebook (192.168.0.2)
File
Edit
Windows
Network
Help
New
Open
Close Ctrl+W
SaveAs
notebook (192.168.0.2) Ctrl+O
forward (192.168.0.1)



When you choose one of the computers in the list an Open Workspace dialog will be display a view of the files available on that computer.



Path: Parallel

licence
logs
newFile_1.gcg

File name:

File filter: Gauss workspaces (*.gcg)

Choose the workspace you want to open from the list.

Running GAUSS Programs

When you select the **File**, **Run** menu item or click on the run icon, ▷, a file browser is opened for the files available on the computer that is running the currently active workspace window. You can then choose the GAUSS program to run.

Note: You can only run programs that already reside on the computer that is running the workspace. Use the TransferFile command to transfer programs (and data files) between computers in a Parallel network. See the Parellel command program transferFile.pgc (in the Parallel directory) for an example.

Note: When you run a GAUSS program, or any GAUSS command, it is run completely and independently on the computer where the workspace is running and only has access to files that already exist on that computer. If the program you intend to run needs to load another file, either data or a support program, then that file must already exist on the computer. The master computer displays the program's output and can enter user input and interrupt the program, but none of the files on the master computer are available for use by the remote computer's GAUSS program.

Running Parallel Command Files

You can only run Parallel command files from the master computer. The one you are using. When you select the **`File`**, **`Run`** menu item or click on the run icon, ▷, with a local (master) workspace window active (that is a workspace that does not have a computer name// added to the front of its file name), you will also be shown the **.pgc** files, Parallel command files.

Once you start the Parallel Command file, it can open remote workspaces and run remote GAUSS commands and programs. As in the non−network mode, the easiest way to write Parallel Command files is to exectue the commands by hand first and then cut and past them from the **command.log** file in the Parallel/logs subdirectory.

```
/*
 * File transferFile.pgc
 *
 * This command file transfers a source file from the master computer to
 * the computer forward.
 * The transfer synchronizes newFile_1.gcg and
 * forward//newFile_2.gcg so that the source file is not
 * executed until it has finished being copied.
 * It then runs the source file in the workspace
 * forward//newFile_2.gcg and copies the result,
 * xResult, back to the Master
 *
 * You will need to modify this file to suit your
 * computer names and workspace and program names.
 */


/* open local workspace here */
»openWorkspace newFile_1.gcg;

/* open remote workspace here */
»openWorkspace forward//newFile_2.gcg;

/* copy the program file to remote machine */
»TransferFile c:\gauss50\parallel\remoteProgram.src[newFile_1.gcg]
         to c:\gauss50\remoteProgram.src[forward//newFile_2.gcg];

/* run the file and copy the result back to the Master computer */
»run c:\gauss50\remoteProgram.src;
»CopyGlobal xResult
         to xResult[newFile_1.gcg];

/* switch back to local workspace where results have been copied to */
»openWorkspace newFile_1.gcg;
```

IP Networkng Tips

Network Names and Addresses

Parallel in Network mode needs to know the IP address of its Master or Slaves, respectively, in order to be able to communicate with them. It is usually more convenient to assign computers meaningful names rather then refer to them by their IP address. On small independent networks, that do not have a Domain Name Server (DNS), hosts files provide the map between computer names and IP addresses.

Parallel tries to display meaningful computer names instead of the IP addresses it uses internally. To do this, Parallel tries to lookup the computer name given its IP and vise versa. If your Parallel network is slow to make connections between computers, slow to open remote windows, or if Parallel is returning network connection errors, then it is possible Parallel is having trouble doing the lookups.

In this case you need to check your DNS is working correctly. For small independent networks, check the hosts files. For larger networks see your administrator.

On Windows networks, being able to transfer files between computers using Window's Explorer does not mean the lookup is working correctly. Windows has its own proprietary method of naming computers on a network.

Making a hosts File

Assuming you are setting up a small independent network to run Parallel on, here is a sample **hosts** file.

```
127.0.0.1          localhost
192.168.0.1        forward
192.168.0.2        notebook
192.168.0.3        fcc
```

This file is a plain text file, which you can use a simple text editor (like Notepad) to create or modify.

This hosts file identifies three computers on the network called **forward**, **notebook** and **fcc** and assigns a unique IP address to each. It also names the loopback address, 127.0.0.1 as **localhost**. The loopback address is how the computer sends messages to itself. All computers use the same loopback address of 127.0.0.1 **localhost** is the common name for this address.

For more details on hosts files see http://networking.earthweb.com/netsysm/article.php/991281 (for Windows hosts files) and http://publibn.boulder.ibm.com/doc_link/en_US/a_doc_lib/files/aixfiles/hosts.htm for the general specifications.

Once you have created a **hosts** file that contains the names and IP addresses for all the computers you want to use in your Parelle network, copy that file to each computer so that they all know the address and name of very other computer.

Where you put the **hosts** file varies with operating system :−
UNIX operating systems typically use the file **/etc/hosts**,
Windows 2000 uses the file **c:\WINNT\system32\drivers\etc\hosts**,
Windows XP uses the file **c:\WINDOWS\system32\drivers\etc\hosts** and
Windows 98 uses just **C:\WINDOWS\hosts**.

Note: The **hosts** file name does not have any extentsion. If you text editor adds a .txt extension, rename the file to just **hosts**

Error Messages

au.com.forward.parallel.exceptions.CopyGlobalException :
An error occurred while processing a CopyGlobal command. The message displayed with this error will give more information about the cause of the error.

au.com.forward.parallel.exceptions.FileLockAquireException:
An error occurred while Parallel was trying to lock a file. The message displayed with this error will give more information about the cause of the error.
This error is often caused by either a system error or a program error.

au.com.forward.parallel.exceptions.FileLockException:
An error occurred while Parallel was trying to lock a file. The message displayed with this error will give more information about the cause of the error.
This error is often caused by either a system error or a program error.

au.com.forward.parallel.exceptions.FileLockReleaseException:
An error occurred while Parallel was trying to release the lock on a file. The message displayed with this error will give more information about the cause of the error.
This error is often caused by either a system error or a program error.

au.com.forward.parallel.exceptions.GaussHomeDirException :
GAUSS cannot find its home directory (where the gauss.cfg file is located).

au.com.forward.parallel.exceptions.ProgrammingException:
This is due to a program error. Please save the Parallel.log file and email it to Parallel@forward.com.au together with the circumstances that produced this error.

au.com.forward.parallel.exceptions.MultiHomedException:
This computer has two or more network connections (see the Identity dialog in the Network menu or the About dialog in the Help menu for details of the connections).
Computers with more then one network connection cannot be used as a Master in a Parallel network.
However this computer can be used as a Slave.
If you must use this computer as the Master, disable the extra connections until you have put Parallel into Master mode.

au.com.forward.parallel.exceptions.NetworkException:
A network error occured. The error message will give more information about this error.

au.com.forward.parallel.exceptions.TransferFileException :
An error occurred while processing a TransferFile command. The message displayed with this error will give more information about the cause of the error.

au.com.forward.parallel.LicenceException:
There is a problem with the licence for this copy of Parallel. The message displayed with this error gives more details on the error that occurred.

To obtain a Parallel licence, send Aptech Systems, Inc (www.aptech.com) the **Parallel Product ID** and **GAUSS ID** displayed in the error message. This information can also be found at the bottom of the **Parallel.log** file in the **parallel/logs** sub−directory of your home directory.

Place the licence obtained from Aptech Systems, Inc. in a file called **parallel.lic** in the **parallel/licence** sub−directory of the GAUSS directory.

au.com.forward.parallel.exceptions.ParallelUndoRedoException:
An error occurred while trying to Undo or Redo. You may be able to recover by hand. Please save and refer to the command.log file for details of the undo file names.

au.com.forward.parallel.exceptions.TempFileException:
The name of the file you are trying to open or saveAs starts with '_P_' or is in a sub−directory called '_P_UNDO'. Parallel uses this for temporary files. If you are trying to undo a GAUSS program please make a copy of of this file and rename it.

au.com.forward.parallel.exceptions.WorkspaceWindowException :
An error occurred while processing a command. The message displayed with this error will give more information about the cause of the error.

au.com.forward.parallel.clientServer.messages.Connect:
There was an error trying to connect to a Slave or to the Master. The message displayed with this error will give more information of the cause of this error.
Common causes of this error are:−
i) Parallel could not find the machine you asked it to connect to.
ii) The machine you are trying to connect to is not running Parallel as a Master or a Slave. The connection will be established when Parallel is started on the other machine and set as a Master or Slave.
iii) Your machine has the same Master/Slave setting as the machine you are trying to connect to. You can only connect a Slave to a Master OR connect a Master to a Slave. Change the Master/Slave setting on one of the machines and try again.

au.com.forward.parallel.clientServer.messages.Connecting:
This computer is trying to contact the other computer mentioned in the message. Either it will respond or the attempt will timeout. In either case there will be another message displayed in the Network Alerts window to tell you what happened.

au.com.forward.parallel.clientServer.messages.ConnectionFromMaster:
The Master computer has established a connection to this Slave computer.

au.com.forward.parallel.clientServer.messages.ConnectionToMasterClosed:
The connection to the Master computer has been closed. Waiting for Master to make a new connection.

au.com.forward.parallel.clientServer.messages.MasterClosedConnection:
The master has closed this connection. Wait for Master to make contact again.

au.com.forward.parallel.clientServer.messages.MasterContacted:
The master has been notified that this Slave is ready to receive commands. Wait for Master to contact us when it is ready.

au.com.forward.parallel.clientServer.messages.MasterModeDisabled:
This computer is not longer in Master mode, either because of some connection error or because you deselected the Master menu item in the Network menu.
All slave windows will be closed as this computer can not longer contact or control the Slave computers.

au.com.forward.parallel.clientServer.messages.NewSlaveAdded:
A new Slave has been found and added to the control of this Master. You can now create new workspaces or open existing workspaces on this slave.

au.com.forward.parallel.clientServer.messages.RemoteFileSystemException:
An error occurred while trying to retrieve information about the remote file system.
The error message will give more details of the error that occurred.

au.com.forward.parallel.clientServer.messages.RemoteTextFieldReaderException:
An error occurred while trying to read GAUSS user input from the Master computer.
The error message will give more details of the error that occurred.

au.com.forward.parallel.clientServer.messages.SlaveRemoved:
The connection to this Slave has been closed. Use the AddSlave menu item in the Network menu to re−open a connection to this Slave.
Re−opening a connection may be delayed while the Slave cleans up from the last connection (closes all windows). If the Slave has been taken out of Slave mode or Parallel has been shut down you will not be able to re−open the connection.

au.com.forward.parallel.clientServer.messages.SlaveAlreadyAdded:
This Slave has been already been added. You can create new workspaces or open existing workspaces on this slave.

au.com.forward.parallel.clientServer.messages.SlaveClosedWindow:
This Slave has closed this window.
Someone at the slave computer has closed this window or has closed Parallel thereby closing all this

slave's windows.

au.com.forward.parallel.clientServer.messages.SlaveClosedConnectionException:
This Slave has closed this connection.
Someone at the slave computer has closed Parallel or turned off the Slave mode
or the may have been a programming error which has closed the connection.

au.com.forward.parallel.clientServer.messages.SlaveModeDisabled:
This computer is not longer in Slave mode, either because of some connection error or because you deselected the Slave menu item in the Network menu.
The Master computer can not longer contact or control this computer.

au.com.forward.parallel.clientServer.messages.SlaveReady:
This Slave is ready to receive commands. You can create new workspaces or open existing workspaces on this slave.

au.com.forward.parallel.clientServer.messages.WindowsOpen:
Some open windows are preventing you from changing the Master or Slave setting of Parallel.
In order to select Slave mode in Parallel there must be no windows open (apart from the Network Alerts window).
In order to exit the Master or Slave setting of Parallel, there must be no remote windows open.
Close the windows and try again.

au.com.forward.parallel.gui.messages.AddTextInterrupted:
Some output was being written to this window when you interrupted the process.

au.com.forward.parallel.gui.messages.CloseDelayed:
This window is waiting for the current program to complete before it closes.
You can interrupt the program using the **`Edit`**, **`Interrupt Current Program`** menu item.

au.com.forward.parallel.gui.messages.CommandFile :
The Parallel command file (**.pgc**) was stopped. The message displayed will give more information about why it was stopped.

au.com.forward.parallel.gui.messages.CommandFileRunning:
A Parallel command file (**.pgc**) is currently running. To stop it, use the **`STOP`** button,🛑

au.com.forward.parallel.gui.messages.CopyGlobal:
Some of the input to the Copy Global From dialog box was not correct. The message gives more details on the problem. Choose **`Edit`**, **`Copy Global`** again to correct the problem.

au.com.forward.parallel.gui.messages.FileInUse:

The file you are trying to access is currently in use. It may be already open in another Parallel window or being used by another process.>

au.com.forward.parallel.gui.messages.FileNotFound:
This file could not be found. Did you delete it?

au.com.forward.parallel.gui.messages.FileTransferStopped :
A file transfer has been stopped. This is not an error.
The file transfer was stopped because it was no longer needed due to some other error or interruption.

au.com.forward.parallel.gui.messages.InvalidFileExtension:
The extension used is not valid for this type of file. The valid extensions are:

.gcg	Workspaces
.pgc	Parallel command files

au.com.forward.parallel.gui.messages.NonLocalCommandFile :
Only Parallel Command files (*.pgc) that exist on this local computer can be run. You cannot run a Parallel command file that resides on a remote Slave computer.

au.com.forward.parallel.gui.messages.UnknownWindow:
Parallel could not match up this remote window. Please exit Parallel on both the Master and Slave and then restart Parallel and try again.
If you get this error repeatedly, please save the Parallel.log file and email it to Parallel@forward.com.au together with the circumstances that produced this error.

au.com.forward.gaussApi.GaussException:
GAUSS has returned an error. The message displayed with this error gives more details on the error that occurred.

java.lang.ArrayIndexOutOfBoundsException:
This is probably due to a program error. Please save the Parallel.log file and email it to Parallel@forward.com.au together with the circumstances that produced this error.

java.lang.AssertionError:
This is probably due to a program error. Please save the Parallel.log file and email it to Parallel@forward.com.au together with the circumstances that produced this error.

java.lang.ClassNotFoundException:
This is probably due to an installation error. Please check the installation instructions. If the error persists please save the Parallel.log file and email it to Parallel@forward.com.au together with the

circumstances that produced this error.

java.lang.ClassCastException:
This is probably due to a program error. Please save the Parallel.log file and email it to Parallel@forward.com.au together with the circumstances that produced this error.

java.lang.Error:
This error can occur when you interrupt a GAUSS program. In these cases just continue.
If this error occurs in any other situation it is most likely due to a program error. Please save the Parallel.log file and email it to Parallel@forward.com.au together with the circumstances that produced this error.

java.lang.IndexOutOfBoundsException:
This is due to a program error. Please save the Parallel.log file and email it to Parallel@forward.com.au together with the circumstances that produced this error.

java.lang.IllegalArgumentException:
This is probably due to a program error. Please save the Parallel.log file and email it to Parallel@forward.com.au together with the circumstances that produced this error.

java.lang.IllegalStateException:
This is probably due to a program error. Please save the Parallel.log file and email it to Parallel@forward.com.au together with the circumstances that produced this error.

java.lang.IllegalThreadStateException:
This is probably due to a program error. Please save the Parallel.log file and email it to Parallel@forward.com.au together with the circumstances that produced this error.

java.lang.InterruptedException:
This is probably due to you interrupting an operation by clicking the STOP button 🛑 or choosing 'Interrupt Current Program' from the Edit menu. While Parallel tries to recover gracefully from these interrupts, the operation did not get a chance to complete its work and so some parts of the window may not be useable.
Try closing the window and re−opening it.

java.lang.NoClassDefFoundError:
This is probably due to a program or installation error. Please save the Parallel.log file and email it to Parallel@forward.com.au together with the circumstances that produced this error.

java.lang.NoSuchMethodException:
This is probably due to a program error. Please save the Parallel.log file and email it to Parallel@forward.com.au together with the circumstances that produced this error.

java.lang.NullPointerException:
This is probably due to a program error. Please save the Parallel.log file and email it to Parallel@forward.com.au together with the circumstances that produced this error.

java.lang.OutOfMemoryError:
This is probably due to a trying to display a large amount of GAUSS output.
To recover close Parallel and restart it. (You may need to kill the Parallel application)

java.lang.reflect.InvocationTargetException:
This is probably due to a program error. Please save the Parallel.log file and email it to Parallel@forward.com.au together with the circumstances that produced this error.

java.lang.RuntimeException:
This is probably due to a program error. Please save the Parallel.log file and email it to Parallel@forward.com.au together with the circumstances that produced this error.

java.lang.UnsatisfiedLinkError:
This is probably due to an installation error which is preventing the Parallel from finding the shared library. Please check that the Parallel shared library is in a directory that is on your library path. See the installation instructions for more details.

java.io.IOException:
An error occurred while trying to read or write a file or while communicating to another computer. The message displayed with this error will give more information of the cause of this error.

Common causes of this error when reading and writing files are:−

- the file could not be found,
- the disk is full,
- the file is already in use,
- the file is set to ReadOnly.

When communicating to another computer:−

- java.io.IOException: Socket input closed.
  means the other computer closed the connection. If you were trying to Add a slave computer at the time, then a possiable cause could be that the remote computer will not accept a connection from this computer because the remote computer does not recognize this computer as its master. Check that the remote computer has the correct Master computer setting. Try IP's instead of names.
  See the Network, Idenity menu item for the IP's of each computer.
- java.net.SocketException: Connection reset
  means the connection to the other computer has been closed. A possiable is that Parallel on

the other computer has been shut down.

java.net.BindException:
This computer could not use this port. This port, 4898, is registered for use by Parallel. If another application is using this port, close the other application before starting Parallel.
This error is also reported if the computer has lost its network connection. In this case normal operation will be restored when the network connection is restored.

java.net.ConnectException:
An error occurred while trying to connect to another machine. The message displayed with this error will give more information of the cause of this error.
Common causes of this error are:−
i) The machine you are trying to connect to is not running Parallel as a Master or a Slave. The connection will be established when Parallel is started on the other machine and set as a Master or a Slave.
ii) Your machine has the same Master/Slave setting as the machine you are trying to connect to. You can only connect a Master to a Slave OR connect a Slave to a Master. Change the Master/Slave setting on one of the machines and try again.

java.net.SocketException:
An error occurred while trying to communicate to another computer. The message displayed with this error will give more information of the cause of this error.
If the message is "Connection Reset", this means the connection has been closed.
If the message is "Connection Reset by Peer" this means the conneciton has been closed by the other computer.
If the message is "Software caused connection abort: socket write error", this can be caused by trying to add a remote (slave) computer, whose master is not you. Check that the remote computer is set in Slave mode and that you are set as the master. Try using IP numbers instead of names in case there is a DNS lookup problem. You should then try and re−connect to the Slave computer again. This error can also be just a transient error that can occur when the master tries to add a slave which is trying to contact the master at the same time. Just try and add the slave again.

java.net.UnknownHostException:
Could not find the IP address for this host name
OR
this IP is invalid. (IPs should be of the form *nnn.nnn.nnn.nnn* Where nnn is 0 to 255. Eg. 192.168.0.1)

java.util.NoSuchElementException:
This is most likely due to a program error. Please save the Parallel.log file and email it to Parallel@forward.com.au together with the circumstances that produced this error.

javax.swing.text.BadLocationException:
An error occured while trying to update the screen. This is most likely due to a program error. Please save the Parallel.log file and email it to Parallel@forward.com.au together with the circumstances that produced this error.

javax.swing.undo.CannotUndoException:
An error occured while trying to Undo. This is most likely due to a program error. Please save the Parallel.log file and email it to Parallel@forward.com.au together with the circumstances that produced this error.

javax.swing.undo.CannotRedoException:
An error occured while trying to Redo. This is probably due to a program error. Please save the Parallel.log file and email it to Parallel@forward.com.au together with the circumstances that produced this error.

javax.swing.text.StateInvariantError:
An error occured in the Java runtime. This sometimes occurs while trying to display the Welcome Dialog box while at the same time trying to display other error messages. Close Parallel and re−start it.

If you can reliably reproduce the error, please save the Parallel.log file and email it to Parallel@forward.com.au together with the circumstances that produced this error and we will report it to Sun.