

Guidance Development Environment

User Manual

Part Number 609737, Revision A



Brooks Automation

Information provided within this document is subject to change without notice, and although believed to be accurate, Brooks Automation assumes no responsibility for any errors, omissions, or inaccuracies.

AcuLigner™, Advan Tag™, AutoTeach™, ATR™, AXM™, BiSymmetrik™, CenterSmart™, Crate to Operate™, CrossingConnect™, DARTS™, Enerta™, e-RMA™, e-Spares™, e-Volution™, Falcon™, FIXLOAD™, FrogLeg™, GuardianPro™, Independent Twin Linear Exchange™, InCooler™, InLigner™, Isoport™, ITLX™, Jet™, Jet Engine™, LEAP™, LeapFrog™, LowProfile™, LPT™, M2 Nano™, Marathon 2, Marathon Express, PASIV™, Pathway™, PowerPak™, PowerTools™, PuroMaxx™, QuadraFly™, Radius™, Radiant™, Radiant Express™, Reliance™, Reliance ATR™, RetroEase™, SCARA™, SmartPM™, SMIF-INX™, SMIF-LPT™, SPOTLevel™, The New Pathway to Productivity™, Time Optimized Trajectory™, Time Optimal Trajectory™, Time Optimized Path™, TopCooler™, TopLigner™, VacuTran™, VersaPort™, WaferEngine™, LEAP™, Pathway™, GIO, GSB, Guidance 6600, Guidance 6430, Guidance 6420, Guidance 6410, Guidance 6000, Guidance 3400, Guidance 3300, Guidance 3200, Guidance 2600, Guidance 2400, Guidance 2300, Guidance 2200, Guidance 1400, Guidance 1300, Guidance 1200, Guidance 0200 Slave Amplifier, Guidance 0006, Guidance 0004, Guidance Controller, Guidance Development Environment, GDE, Guidance Development Suite, GDS, Guidance Dispense, Guidance Input and Output Module, Guidance Programming Language, GPL, Guidance Slave Board, Guidance System, Guidance System D4/D6, PreciseFlex™ 300, PreciseFlex™ 400, PreciseFlex™ 3400, PreciseFlex™ 1300, PreciseFlex™ 1400, PreciseFlex™ DD4, PreciseFlex™ DD6, PreciseFlex™ DDR, PreciseFlex™ G5400, PreciseFlex™ G5600, PreciseFlex™ G6400, PreciseFlex™ G6410, PreciseFlex™ G6420, PreciseFlex™ G6430, PreciseFlex™ G6600, PreciseFlex™ GSBP Slave Amp, PreciseFlex™ PFD0, PrecisePlace 100, PrecisePlace 0120, PrecisePlace 0130, PrecisePlace 0140, PrecisePlace 1300, PrecisePlace 1400, PrecisePlace 2300, PrecisePlace 2400, PrecisePower 300, PrecisePower 500, PrecisePower 1000, PrecisePower 2000, PreciseVision, and RIO logos are trademarks of Brooks Automation.

Fusion®, Guardian®, MagnaTran®, Marathon®, Razor®, Spartan®, Vision®, Zaris®, and the Brooks and design logo are registered U.S. trademarks of Brooks Automation.

All other trademarks are properties of their respective owners.

© 2024 Brooks Automation. All rights reserved. The information included in this manual is proprietary information of Brooks Automation, and is provided for the use of Brooks customers only and cannot be used for distribution, reproduction, or sale without the express written permission of Brooks Automation.

This technology is subject to United States export Administration Regulations and authorized to the destination only; diversion contrary to U.S. law is prohibited.

<p>Brooks Automation 15 Elizabeth Drive Chelmsford, MA 01824-2400 Tel: +1 978-262-2400 Fax: +1 978-262-2500</p>	<p>Brooks Automation, PreciseFlex Collaborative Robots 201 Lindbergh Avenue Livermore, CA 94551 Tel: +1-408-224-2838</p>
---	---



Worldwide Headquarters

15 Elizabeth Drive
Chelmsford, MA 01824 U.S.A.

**Brooks Automation,
PreciseFlex Collaborative Robots**

201 Lindbergh Avenue
Livermore, CA 94551 U.S.A

Technical Support

Location	Contact	Website
North America	+1-800-447-5007 (Toll-Free) +1-978-262-2900 (Local) +1-408-224-2838 (PreciseFlex™)	http://www.brooks.com/
Europe	support_preciseflex@brooksautomation.com	
Japan	+81 120-255-390 (Toll Free) +81 45-330-9005 (Local)	
China	+86 21-5131-7066	
Taiwan	+886 080-003-5556 (Toll Free) +886 3-5525258 (Local)	
Korea	1800-5116 (Toll Free)	
Singapore	+65 1-800-4-276657 (Toll Free) +65 6309 0701 (Local)	

General Emails

Division	Email Address
Sales	sales_preciseflex@brooksautomation.com
Technical Support	support_preciseflex@brooksautomation.com
Technical Publications	Technical.Publications@brooksautomation.com

Revision History

Revision	ECO	Date	Action	Author
A	EC147365	08/04/2023	Released manual at Rev. A to follow standard Brooks technical publication styles.	M. Ashenfelder

Table of Contents

1. Safety	1
Safety Setup	1
Authorized Personnel Only	1
Explanation of Hazards and Alerts	2
Safety Text	2
Safety Icons	2
Signal Words and Color	2
Alert Example	3
General Safety Considerations	4
Mechanical Hazards	6
Electrical Hazards	7
Ergonomic Hazards	8
Emergency Stop Circuit (E-Stop)	10
Recycling and Hazardous Materials	10
2. Guidance Development Environment	11
GDE Introduction	11
3. Installation and Startup	13
Installing GDE/GDS on a PC	13
Configuring the Controller for GPL Execution	14
Activating GDS Components	15
Connecting to the Controller	17
4. GDE Windows and Tools	19
GPL Projects and Libraries Overview	19
GDE Screen Layout	20
Editor and Debugger Window	22
Main Menu Bar	26
Main Toolbar	30
Debug Toolbar	32
Project Manager Window	33
Object Browser Window	37
Robot Control Window	38
Virtual Pendant Window	39
GPL Output Window	40
Threads Window	41
Program Stack Window	42
File Manager Window	43
Console Window	44
Preferences Panels	45
5. GDE Programming Examples	47
Hello World Example	47
Pick and Place Example	50
Safety Zones	57

Types of Safety Zones	57
Safety Zone Violation Detection and Clearing	59
Certified Speed Restrict Safety Zones	59
Configuring Safe Zones	60
Appendices	62
Appendix A: Frequently Asked Questions	62
Appendix B: Guidance Update Wizard	64

1. Safety

Safety Setup

Brooks uses caution, warning, and danger labels to convey critical information required for the safe and proper operation of the hardware and software. Read and comply with all labels to prevent personal injury and damage to the equipment.

 DANGER Read the Safety Chapter	
<p>Failure to review the <i>Safety</i> chapter and follow the safety warnings can result in serious injury or death.</p> <ul style="list-style-type: none">• All personnel involved with the operation or maintenance of this product must read and understand the information in this safety chapter.• Follow all applicable safety codes of the facility as well as national and international safety codes.• Know the facility safety procedures, safety equipment, and contact information.• Read and understand each procedure before performing it.	

Authorized Personnel Only

This product is intended for use by trained and experienced personnel. Operators must comply with applicable organizational operating procedures, industry standards, and all local, regional, national, and international laws and regulations.

Explanation of Hazards and Alerts

This manual and this product use industry standard hazard alerts to notify the user of personal or equipment safety hazards. Hazard alerts contain safety text, icons, signal words, and colors.

Safety Text

Hazard alert text follows a standard, fixed-order, three-part format.

- Identify the hazard
- State the consequences if the hazard is not avoided
- State how to avoid the hazard.

Safety Icons

- Hazard alerts contain safety icons that graphically identify the hazard.
- The safety icons in this manual conform to ISO 3864 and ANSI Z535 standards.

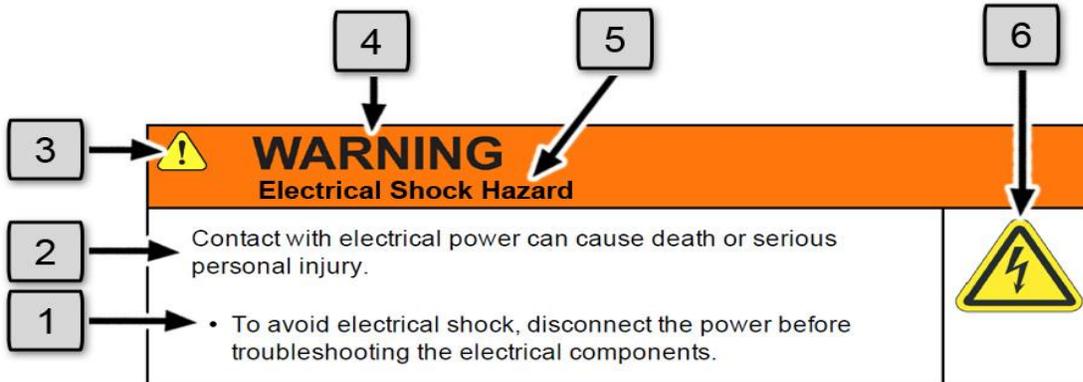
Signal Words and Color

Signal words inform of the level of hazard.

	<p>Danger indicates a hazardous situation which, if not avoided, will result in serious injury or death.</p> <p>The Danger signal word is white on a red background with an exclamation point inside a yellow triangle with black border.</p>
	<p>Warning indicates a hazardous situation which, if not avoided, could result in serious injury or death.</p> <p>The Warning signal word is black on an orange background with an exclamation point inside a yellow triangle with black border.</p>
	<p>Caution indicates a hazardous situation or unsafe practice which, if not avoided, may result in minor or moderate personal injury.</p> <p>The Caution signal word is black on a yellow background with an exclamation point inside a yellow triangle with black border.</p>
	<p>Notice indicates a situation or unsafe practice which, if not avoided, may result in equipment damage.</p> <p>The Notice signal word is white on blue background with no icon.</p>

Alert Example

The following is an example of a Warning hazard alert.



Number	Description
1.	How to Avoid the Hazard
2.	Source of Hazard and Severity
3.	General Alert Icon
4.	Signal Word
5.	Type of Hazard
6.	Hazard Symbol(s)

General Safety Considerations

 WARNING Software	
<p>Software is not safety rated. Unplanned motion can occur as long as power is supplied to the motors. Maximum torque could be momentarily applied that may cause equipment damage or personal injury.</p> <ul style="list-style-type: none">• Only operate the robot with its covers installed.• Guarantee that safety controller features are in place (for example, an emergency stop button and protective stop).• Regularly test safety components to prove that they function correctly.	 

 WARNING Robot Mounting	
<p>Before applying power, the robot must be mounted on a rigid test stand, secure surface, or system application. Improperly mounted robots can cause excessive vibration and uncontrolled movement that may cause equipment damage or personal injury.</p> <ul style="list-style-type: none">• Always mount the robot on a secure test stand, surface, or system before applying power.	

 WARNING Do Not Use Unauthorized Parts	
<p>Using parts with different inertial properties with the same robot application can cause the robot's performance to decrease and potentially cause unplanned robot motion that could result in serious personal injury.</p> <ul style="list-style-type: none">• Do not use unauthorized parts.• Confirm that the correct robot application is being used.	

 WARNING Magnetic Field Hazard	
<p>This product contains magnetic motors that can be hazardous to implanted medical devices, such as pacemakers, and cause personal harm, severe injury, or death.</p> <ul style="list-style-type: none">• Maintain a safe working distance of 30 cm from the motor when with an energized robot if you use a cardiac rhythm management device.	

 CAUTION Unauthorized Service	
<p>Personal injury or damage to equipment may result if this product is operated or serviced by untrained or unauthorized personnel.</p> <ul style="list-style-type: none">• Only qualified personnel who have received certified training and have the proper job qualifications are allowed to transport, assemble, operate, or maintain the product.	

 CAUTION Damaged Components	
<p>The use of this product when components or cables appear to be damaged may cause equipment malfunction or personal injury.</p> <ul style="list-style-type: none">• Do not use this product if components or cables appear to be damaged.• Place the product in a location where it will not get damaged.• Route cables and tubing so that they do not become damaged and do not present a personal safety hazard.	

 CAUTION Inappropriate Use	
<p>Use of this product in a manner or for purposes other than for what it is intended may cause equipment damage or personal injury.</p> <ul style="list-style-type: none">• Only use the product for its intended application.• Do not modify this product beyond its original design.• Always operate this product with the covers in place.	

 CAUTION Seismic Restraint	
<p>The use of this product in an earthquake-prone environment may cause equipment damage or personal injury.</p> <ul style="list-style-type: none"> The user is responsible for determining whether the product is used in an earthquake prone environment and installing the appropriate seismic restraints in accordance with local regulations. 	

Mechanical Hazards

 CAUTION Pinch Point	
<p>Moving parts of the product may cause squeezing or compression of fingers or hands resulting in personal injury.</p> <ul style="list-style-type: none"> Do not operate the product without the protective covers in place. 	

 WARNING Automatic Movement	
<p>Whenever power is applied to the product, there is the potential for automatic or unplanned movement of the product or its components, which could result in personal injury.</p> <ul style="list-style-type: none"> Follow safe practices for working with energized products per the facility requirements. Do not rely on the system software or process technology to prevent unexpected product motion. Do not operate the product without its protective covers in place. While the collaborative robotics system is designed to be safe around personnel, gravity and other factors may present hazards and should be considered. 	

 CAUTION Vibration Hazard	
<p>As with any servo-based device, the robot can enter a vibratory state resulting in mechanical and audible hazards. Vibration indicates a serious problem. Immediately remove power.</p> <ul style="list-style-type: none">• Before energizing, ensure the robot is bolted to a rigid metal chamber or stand.	

Electrical Hazards

Refer to the specifications of the *Guidance Controller Quick Start Guide* for the electrical power.

 DANGER Electrical Shock Hazard	
<p>Contact with electrical power can cause personal harm and serious injury.</p> <ul style="list-style-type: none">• To avoid electrical shock, disconnect the power before troubleshooting the electrical components.• Check the unit's specifications for the actual system power requirements and use appropriate precautions.• Never operate this product without its protection covers on.	

 WARNING Electrical Burn	
<p>Improper electrical connection or connection to an improper electrical supply can result in electrical burns resulting in equipment damage, serious injury, or death.</p> <ul style="list-style-type: none">• Always provide the robot with the proper power supply connectors and ground that are compliant with appropriate electrical codes.	

 WARNING Electrical Fire Hazard	
<p>All energized electrical equipment poses the risk of fire, which may result in severe injury or death. Fires in wiring, fuse boxes, energized electrical equipment, computers, and other electrical sources require a Class C extinguisher.</p> <ul style="list-style-type: none"> Use a fire extinguisher designed for electrical fires (Class C in the US and Class E in Asia). It is the facility's responsibility to determine if any other fire extinguishers are needed for the system that the robot is in. 	

NOTICE
<p>Improper handling of the power source or connecting devices may cause component damage or equipment fire.</p> <ul style="list-style-type: none"> Connect the system to an appropriate electrical supply. Turn off the power before servicing the unit. Turn off the power before disconnecting the cables.

Ergonomic Hazards

 CAUTION Heavy Lift Hazard	
<p>Failure to take the proper precautions before moving the robot could result in back injury and muscle strain.</p> <ul style="list-style-type: none"> Use a lifting device and cart rated for the weight of the drive or arm. Only persons certified in operating the lifting device should be moving the product. 	

 CAUTION Tipover Hazard	
<p>This product has a high center of gravity which may cause the product to tip over and cause serious injury.</p> <ul style="list-style-type: none"> Always properly restrain the product when moving it. Never operate the robot unless it is rigidly mounted. 	

 CAUTION Trip Hazard	
<p>Cables for power and communication and facilities create trip hazards which may cause serious injury.</p> <ul style="list-style-type: none">• Always route the cables where they are not in the way of traffic.	

Emergency Stop Circuit (E-Stop)

The integrator of the robot must provide an emergency stop switch.

 WARNING Emergency Stop Circuit	
<p>Using this product without an emergency stop circuit may cause personal injury.</p> <ul style="list-style-type: none">• Customer is responsible for integrating an emergency stop circuit into their system.• Do not override or bypass the emergency stop circuit.	

Recycling and Hazardous Materials

Brooks Automation complies with the EU Directive 2002/96/EU Waste Electrical and Electronic Equipment (WEEE).

The end user must responsibly dispose of the product and its components when disposal is required. The initial cost of the equipment does not include cost for disposal. For further information and assistance in disposal, please email Brooks Automation Technical Support at support@brooksautomation.com.

2. Guidance Development Environment

GDE Introduction

The Guidance Development Environment (GDE) is a software package that allows you to develop and debug Guidance Programming Language (GPL) software projects. GDE runs on a Windows PC. To execute and debug programs, the PC must be connected to the controller via Ethernet either locally or remotely. However, for offline development, the PC need not be connected to a Guidance Controller.

In this document, instructions for installing and executing GDE are provided. This is followed by a description of each of the components of GDE, their basic operation, and the functions that they provide. Finally, the step-by-step process of developing and executing the traditional “Hello World” program is presented along with a simple robot pick-and-place application.

In order to execute the examples in this document, the following are required:

- A 500 MHz or faster PC running a Windows XP or later system
- Microsoft Internet Explorer version 8.x or later
- A 10/100 Ethernet card for the PC
- At least 20 MB of space on the PC's disk
- A 2x CD-ROM drive interfaced to the PC or connection to the Internet
- The Guidance Development Suite distribution disk (P/N: PDS0-DA-00010) or access to the Precise Support website
- A Guidance Controller interfaced to a robot, e.g. a PreciseFlex™ 400

 WARNING	
<p>Before proceeding with this Guide, please ensure that the following steps have already been performed</p> <ul style="list-style-type: none"> The robot has been properly mounted, all required safety interlocks have been installed and tested, and power has been connected. For the <i>PrecisePlace 2300/2400 Robot, Hardware Introduction and Reference Manual</i>. If you are integrating the Guidance Controller to a new mechanical system for the first time, please see the section on "Setting Up a Controller" in the <i>PreciseFlex™ PreciseFlex Library</i> for instructions on configuring the controller. 	

Prior to reading this document, do the following:

Step	Action
1.	Perform the exercises in the <i>"Guidance Controller Setup and Operation, Quick Start Guide"</i> to familiarize yourself with the operation of the controller and to verify that the controller has been properly interfaced to the PC.
2.	Read the <i>"Guidance Programming Language, Introduction to GPL"</i> to gain a basic understanding of the functions available in GPL and their associated syntax. For the examples in this document, a thorough understanding of GPL is not necessary. However, when you wish to learn more, all of the GPL statements and classes with their methods and properties are described in the <i>"Guidance Programming Language, GPL Dictionary Pages"</i> .

3. Installation and Startup

Installing GDE/GDS on a PC

GDE is distributed as a component in the Guidance Development Suite of programs (GDS). GDS includes GDE plus other useful PC applications such as the Guidance Datalogger Viewer, which graphs the results of controller data collection sessions. To install GDE, you must install GDS. If you previously installed GDS on your computer, you should un-install the old version by performing the steps below. If you are installing GDS for the first time, you can skip the first set of instructions.

Step	Action
1.	Shut down all programs that are running including virus protection programs
2.	Bring up the Window's Control Panel by clicking Start > Settings > Control Panel .
3.	Double-click on the " Add or Remove Programs " selection.
4.	In the "Add or Remove Programs" window, scroll down and click Guidance Development Suite .
5.	Click the Remove button and click Yes to confirm the action.
6.	Close all windows.

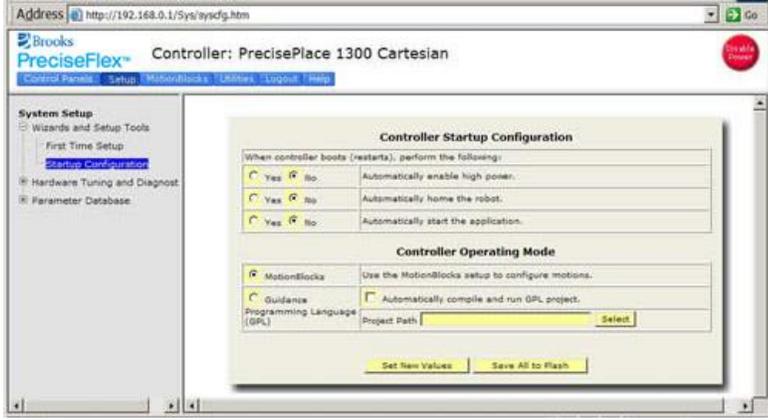
To install GDS on your computer, perform the following steps:

Step	Action
1.	Shut down all programs that are running including virus protection programs.
2.	Insert the Guidance Development Suite CD-ROM into your computer. A panel should popup that welcomes you to the GDS Setup Wizard. If the installer does not automatically start, click Start > Run , type in D:\setup.msi (where D is the CD-ROM drive), and click Ok .
3.	Follow the instructions in the Setup Wizard to install GDS.
4.	NOTE: GDS relies upon the Microsoft .NET Framework in order to operate. This is a standard module that Microsoft provides free of charge. If the Setup Wizard detects that the Microsoft .NET Framework module is not available, the software will ask if you wish to install the Framework now. Respond Yes . This will launch a browser with the download site for the required software.

At the end of the process, GDE will be installed on the PC with the other components of the GDS. Begin using GDE. Note that in order to continue to use GDE and other elements of GDS, **the product must be registered with PreciseFlex**. After starting GDE, go to the "Register product" item under the "Help" top-level menu item for information on the registration process.

Configuring the Controller for GPL Execution

To execute a GPL program, the Guidance controller must be configured for GPL execution instead of MotionBlocks execution. Once this configuration is stored in the controller's flash disk, it will be preserved even if the controller is turned off and restarted. To set this configuration, perform the following procedure. See the *Guidance System Setup and Operation, Quick Start Guide*, and follow the instructions for bringing up the web based Guidance Operator Interface for the controller.

Step	Action
1.	<p>In the web interface, open the Controller Startup Configuration page, Setup > Wizards and Setup Tools > Startup Configuration. The web page should look as follows:</p> 

Step	Action
2.	Click the red Disable Power button at the top of the page. This is required because the changes made are not permitted when power to the robot is enabled.
3.	In the <i>Controller Operating Mode</i> panel, click the Guidance Programming Language (GPL) radio button
4.	Click the Set New Values button to store this setting into memory
5.	Click the Save All to Flash button to store this setting in the flash disk. This ensures that this setting will remain in effect if the controller is restarted. The button will flash for 10-30 seconds as the data is being written. <i>NOTE: Do not turn off your controller while the button is blinking. That action may corrupt the flash disk.</i>

NOTE: When GPL execution mode was selected, the *Automatically start the operating mode* option was also set to **Yes**. With these two settings, each time the robot's power is enabled, the controller is put into a state where it is ready to accept a **Robot.Attached** command to permit a GPL program to take control of a robot.

The controller is now ready to execute a GPL program that includes motion instructions.

Activating GDS Components

To begin using the Guidance Development Environment, start the application on the PC.

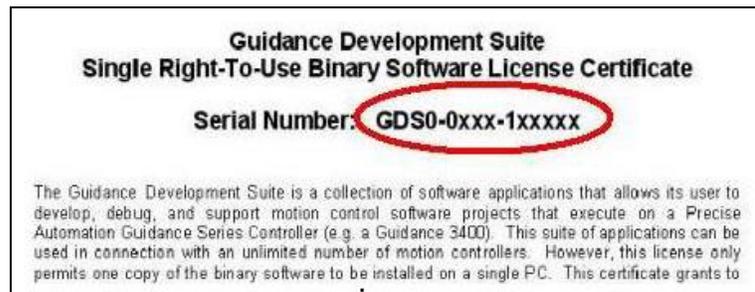
Step	Action
1.	Click Start > Programs > Precise Automation > GDS xxx > Guidance Development Environment . The first time GDE begins execution, it displays this window: 

Step	Action
2.	<p>The GDS license permits the user to use GDE to develop software for an unlimited number of controllers, but the package is only licensed to execute on a single PC. To complete the software activation process, send the PC information to Brooks. All applications within GDS are functional for 30 days without being activated. To activate all components, execute the activation process in one GDS component. This window will display each time an unactivated component of GDS is started. In any GDS component, click Help > Product Activation to display the activation window.</p>
3.	<p>To initiate the activation process, click Yes in the window above to display the following window.</p>  <p>This window displays the Computer ID you need to submit to Brooks as part of the activation process.</p>

To get a license key to activate GDS, email PreciseFlex the following information:

To: sales_preciseflex@brooks.com
 Subject: GDS License Key Request
 Customer Name: <your name here, optional>
 Customer Company: <your company name here>
 Telephone Number: <your phone number here, optional>
 GDS Serial Number: GDS0-0xxx-xxxxxx
 Computer ID Number: xxxx-xxxx-xxxx-xxxx

A user can get the GDS Serial Number from the GDS License Certificate received with an order. Below is a picture of the top portion of a typical GDS License with the serial number circled in red.



The Computer ID number should be copied from the Product Activation popup. Alternatively, if a user does not have email access, Fax this information to 408-516-8348 together with the user's fax number. Name and phone number are optional. However, we recommend providing this information in case there is a problem with the license and we need to contact the registered user. A sample email should look as follows.



In response to the request, Brooks will send a license key.

Step	Action
1.	Access the Product Activation popup by restarting GDE or clicking the GDE menu item Help > Product Activation .
2.	After receipt of the license key, enter it into the License Key box and click Activate . All of the GDS components will be activated.

Connecting to the Controller

Once the Guidance Development Environment is launched, the following screen will display connecting to the Guidance controller via Ethernet.

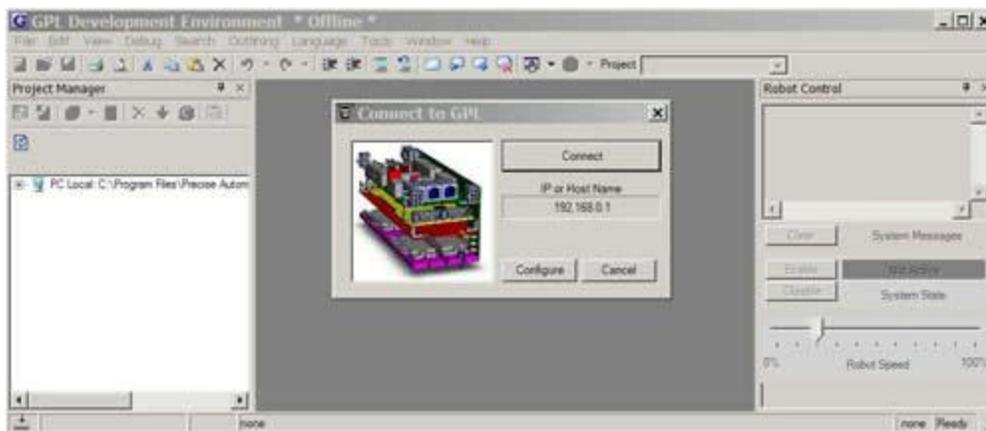


Figure 3-1: Guidance Development Environment interface

If the user only intends to develop software off-line, they could click the **Cancel** button and skip the process of connecting to the controller. However, to execute our code on the controller, the user needs to establish an Ethernet connection.

Step	Action
1.	In the box labeled IP or Host Name , verify that the IP address for the controller is correct. By default, Guidance controllers are configured to respond to 192.168.0.1. If the IP address is correct, skip the next instructions.
2.	To set a new IP address for the controller, press the Configure button.
3.	In the Preference popup window, click the Connection tab, enter the new IP address, and click Update . The user can now connect GDE to the Guidance controller.
4.	Click the Connect button on the connection popup window. The PC should connect to the controller in just a few seconds.

NOTE: To not be automatically prompted to connect each time GDE is launched, go to **File > Preferences > Connection** to disable this feature.

4. GDE Windows and Tools

GPL Projects and Libraries Overview

Before examining the specific elements of GDE, it is important to take a moment to understand GPL "Projects." In GPL, rather than executing a "program," a "Project" is the basic executable entity.

A Project consists of two or more text files that are stored within a single disk folder (i.e. directory). Each file is a standard human-readable ASCII file. The folder name and the Project name are synonymous. Project names conform to the standard GPL symbol name convention and therefore must start with an alphabetic character or "_" and cannot be a single "_." The first character can be followed by any combination of alphanumeric characters and "_." Since Project folders can be stored on the flash disk, Project names are limited to 43 characters in length. Also, since flash disk names are case sensitive, the first alphabetic character in the Project name is always upper case and all other alphabetic characters are lower case, e.g. "Test_project." Specific operations within GPL and GDE are provided for loading, compiling, and executing a Project.

The file "Project.gpr" must always be present in each project folder and is referred to as the "Project File." This file contains information on the other files within the Project. For example, the Project File stores the name of the procedure that is invoked when the Project begins execution. GDE automatically manages the contents of this file and so it is normally hidden within GDE. There can be multiple GPL source files within a Project. Each source file has a ".gpl" extension. These files each can contain one or more program modules, which in-turn can contain multiple variable declarations and procedures.

In addition, a Project can contain one, several or no files with a ".gpo" extension. This type of file stores a global module that is used to defined global **Location** and **Profile** objects. This file is convenient for saving taught robot **Locations** and general motion **Profiles** that are accessible by all procedures within the Project. Almost all of the work done within GDE involves the creation, debugging, and management of the ".gpl" and ".gpo" files for a given Project.

Since a Project consists of a collection of files within a disk folder, loading a Project into memory or copying a Project from memory or between disk units is equivalent to copying a file folder and all of its contents. So, Projects can be managed by graphically dragging and dropping their associated folder onto the desired destination device. Although only one Project can be executed at any time, multiple Projects can be concurrently loaded into memory.

GPL Project Libraries

As a convenience when developing large software projects or for sharing software modules, GDS supports GPL Project Libraries. This feature permits any Project to reference another GPL Project and utilize its public routines and data. No special operations must be performed to convert a Project into a Library. Any Project can be utilized as a Library. To reference a Project as a Library, the main Project must be modified to add the name of the Library into its Project File using the Project Manager facility within GDS. A main Project can reference multiple Libraries and Libraries can reference other Libraries.

When the main Project is compiled, all the files in its referenced Libraries are logically included into the main Project. If two different main Projects refer to the same Library, the Library files are compiled separately into each main Project. This means:

- The use of shared Library Projects does not save memory.
- Global variables defined in the Library Projects are allocated separately for each main Project, so multiple main Projects cannot share data using global variables in the Libraries.

When GDE loads a main Project from flash, it automatically loads any referenced Library Projects from flash. If the Libraries are already loaded, the loaded Libraries are used. If you use GDE to load Libraries from the PC during development, the corresponding Libraries on flash are ignored. When you unload a main Project, the referenced Libraries are not unloaded. If you use GDE to load a main Project from the PC, you must also manually load the Libraries from the PC or from flash.

GDE Screen Layout

After you have successfully connected to GDE for the first time, you will be presented with the following initial application layout.

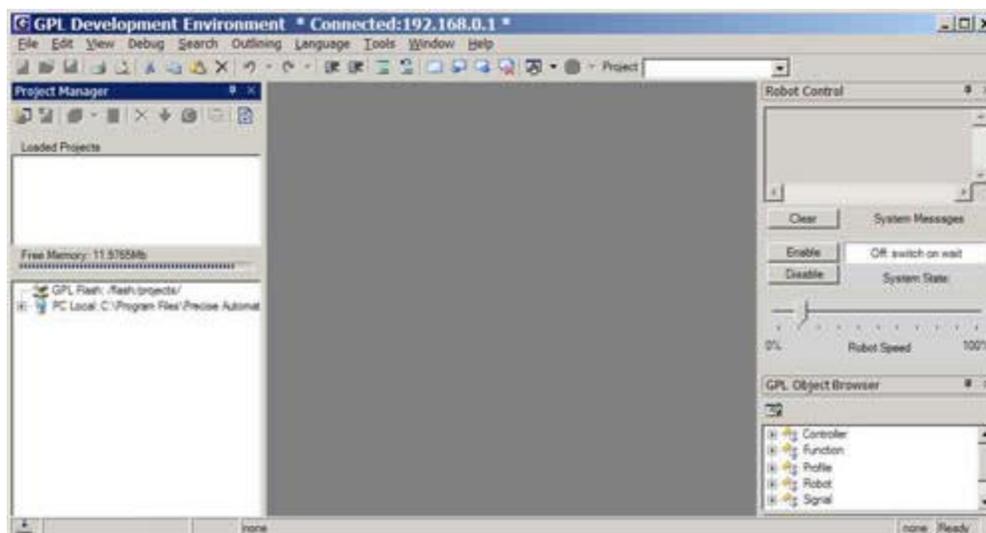


Figure 4-1: GDE Project Manager Screen

The GDE layout contains a title bar that displays the IP address of the connected controller, a top menu bar, a tool bar, the main editor/debugger area and a variety of (dockable) windows. Each of these dockable windows can be displayed or hidden, resized and repositioned into the arrangement that is most efficient for your use. Any space not occupied by a dockable window is utilized by the editor/debugger.

To reposition a window, you simply click in its title bar and drag it to its new location. If you drag a window on to another window, they can split the space or share the space using tab controls. If you click on the close icon (x) in the top right of a window's title bar, you can use the "View" top-level menu to redisplay the window. If you click on the "push pin" in the top right of a window's title bar, you will either "pin" a window and fix its location or "un-pin" a window so it can share its space with another window.

Windows can be resized by grabbing a border and stretching or shrinking it to the desired dimension.

To restore the original positions and sizes of all of the dockable windows, in the main menu, select **View > Window Layout > Load Default Layout**.

In the following sections, the functions available from each of the major components of GDE (i.e. main menu bar, tool bars, and windows) will be described. The components are presented in order of importance rather than alphabetically. The following table briefly summarizes each of the components.

Component	Description
Editor/Debugger Window	Main window for editing and debugging procedures and global motion data.
Main Menu Bar	Main menu bar that provides access to most of the functions provided by GDE.
Main Toolbar	Provides quick access to common editing and other functions.
Debug Toolbar	Provides debug functions for single stepping, pausing and stopping threads.
Project Manager Window	Displays and manages Projects that are resident in the controller's memory and in the flash disk and the PC's hard drive in the standard GPL Project areas.
Object Browser Window	Provides a list of the methods and properties of all GPL Classes. Automatically displays abbreviated help information as text is entered.
Robot Control Window	Displays the controller system messages and state. Allows robot power to be enabled and disabled and the robot speed to be reduced for testing.
Virtual Pendant Window	Permits the robot to be manually jogged and displays the current robot position.
GPL Output Window	Displays all output generated by the controller in connection with the execution of GPL Projects.
Threads Window	Displays execution status and the last procedure executed in each GPL thread.
Program Stack Window	Displays a list of procedures that are on the execution stack for a given thread.

Component	Description
File Manager Window	Displays and manages all folders and files on the controller's flash disk.
Console Window	Provides access to the controller's console. Allows GPL Console Commands to be entered and executed.
Preferences Panels	Contains application setup parameters that customize the operation and appearance of GDE

Editor and Debugger Window

The Editor and Debugger Window is the primary focus of GDE and occupies all of the space not utilized by the displayed dockable windows. This window allows you to create and modify GPL source files and global GPL motion and program variables, and to debug GPL procedures by single stepping, setting breakpoints, displaying variable values, etc.

In its normal source code editing mode, this window will look as follows.

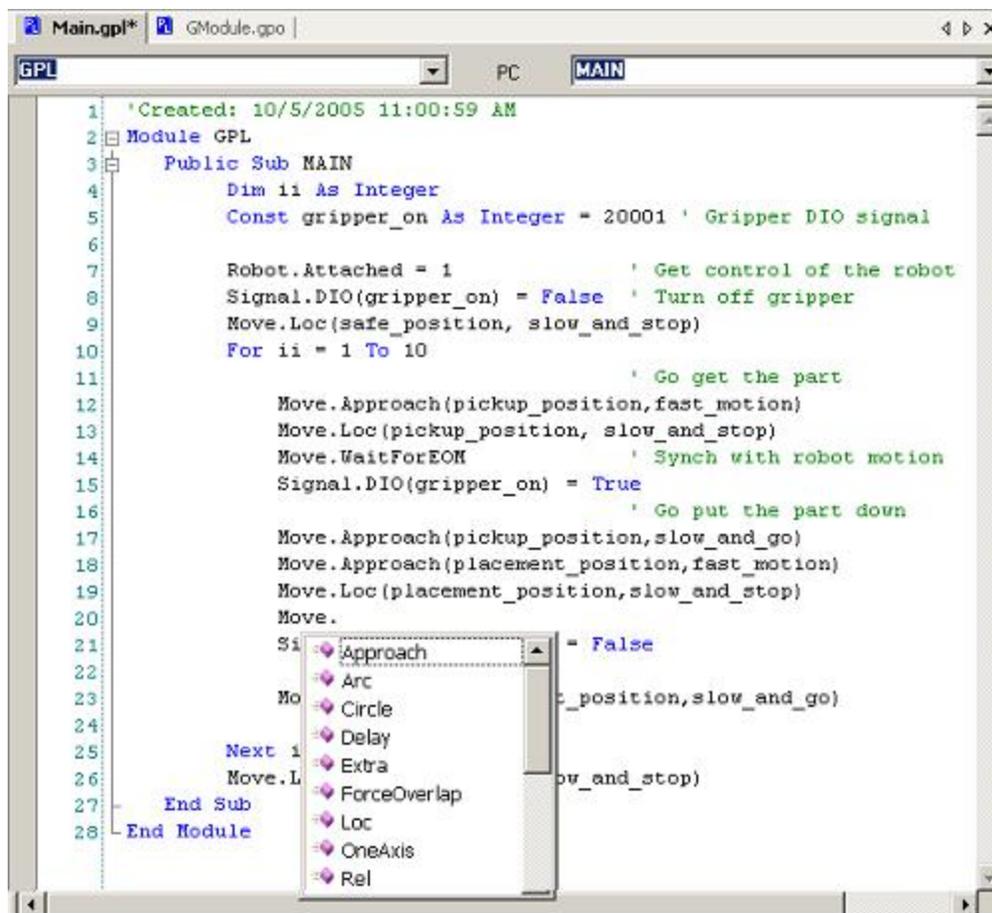


Figure 4-2: Code Editing Mode

To open a file for editing, you can either double click on the file name in the Project Manager Window or right click on the file name and choose "Edit" in the pop-up menu.

As each file is opened, at the top of the editor window, tabs with the names of opened files are dynamically created to allow you easily switch between files. You can edit files located in the controller's memory, in the controller's flash disk or in the PC's hard drive. Each tab icon is color coded to indicate where the file is stored: in memory (red), the flash (yellow) or the PC (blue). Any files that have been modified and not saved are flagged with an "*" following the name. Just below the file tabs, two pull-down menus are available for quickly positioning the cursor at a specific module and procedure within a file. For example, in the picture above, the editor is currently position within the "GPL" module and in the "Main" procedure.

The editor operates in the typical manner for inserting, deleting, cutting, and pasting text. In addition, context sensitive help is available. For example, in the screen shot above, a popup box is automatically displayed when the editor recognizes that you wish to enter a property or method for the GPL Move class. When you select a property or method, the GDE Object Browser will display the syntax for this item as well as a short description. Also, the editor automatically capitalizes keywords and built-in system classes, methods, and properties and color codes the text for greater readability. If you edit a global modules file that contains motion objects and/or global GPL variables, i.e. a *.gpo file, the editor screen will appear as follows.

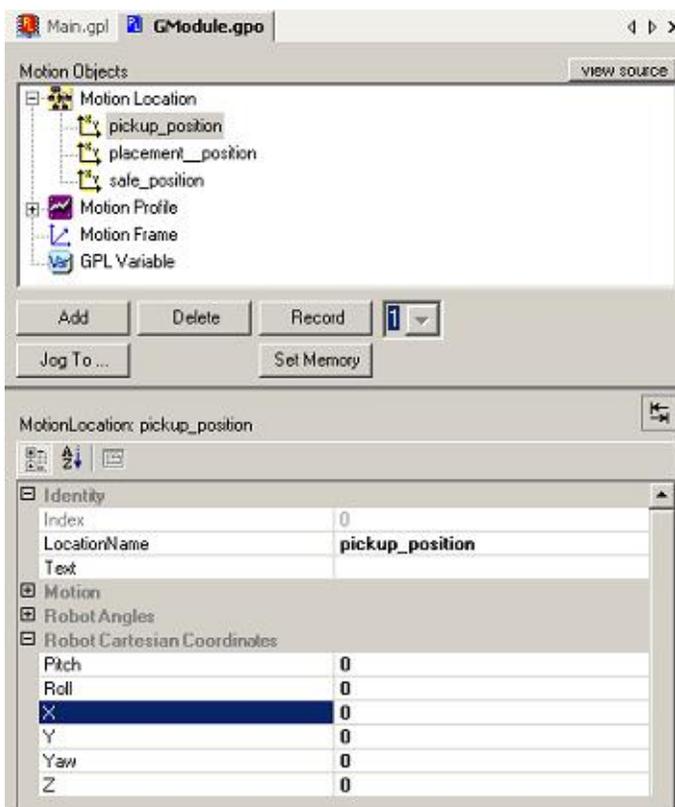


Figure 4-3: Editor Screen

This mode provides buttons that perform the following functions:

- **Add** - Creates a single item or an array of global **Location**, **Profile** and **RefFrame** motion objects as well as single or arrays of **String**, **Integer**, **Single** and **Double** GPL global variables.
- **Delete** - Deletes a selected global variable or array.
- **Record** - Sets the position of the selected global **Location** equal to the current location of a robot. The number of the robot is specified by the drop-box to the right of the **Record** button.
- **Set Memory** - Modifies the value of a variable in a GPL project while it is being executed on a controller. This function is especially convenient for tuning a robot cycle since it permits you to change the properties of a **Profile** or tweak the components of a **Location** as a project is being executed.
- **Jog To** - Manually moves the robot to the selected location (see the description below).

As a convenience in defining the value of new global's (e.g. **Locations** or **Profiles**), the value of one global can be copied and pasted to another global using the **"Copy"** and **"Paste"** selections under **"Edit"** in the Main Menu Bar or the Copy and Paste icons in the Main Toolbar or by pressing Ctrl+C and Ctrl+V or by right clicking in the Motion Objects window.

When GDE is connected to a controller, the "Jog To" function assists in modifying the position and orientation of global **Locations** by permitting you to move the robot back and forth between its current location and the value of a selected global **Location**. To activate this mode, select a **Location** value and press the **"Jog To..."** button. The following panel will then be displayed.

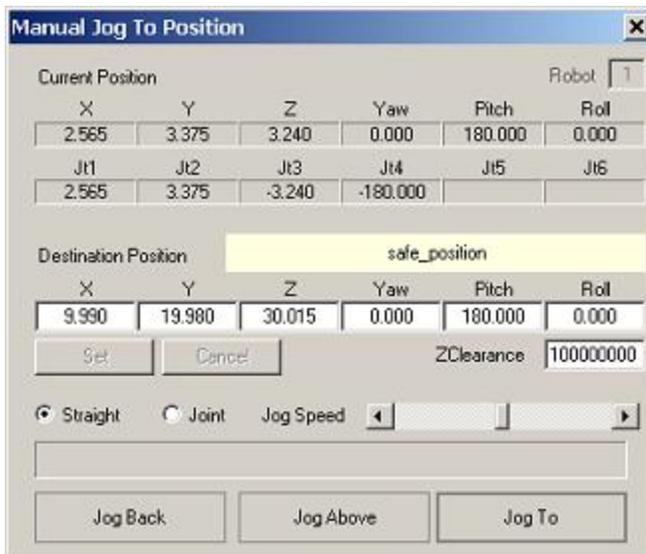


Figure 4-4: Manual Jog To Position Screen

To move the robot between its current location and the position and orientation of the selected global **Location** variable or its approach point, press and hold down the **"Jog To"**, **"Jog Back"** or **"Jog Above"** buttons. There is a Jog Speed slider bar for controlling the speed of the motion and buttons for specifying whether the motion should be joint interpolated or Cartesian controlled. You can also use this panel to edit the components of the select **Location** value as well as its ZClearance value. If any component values are changed, the **"Set"** and **"Cancel"** buttons can modify the stored Location value or cancel the change.

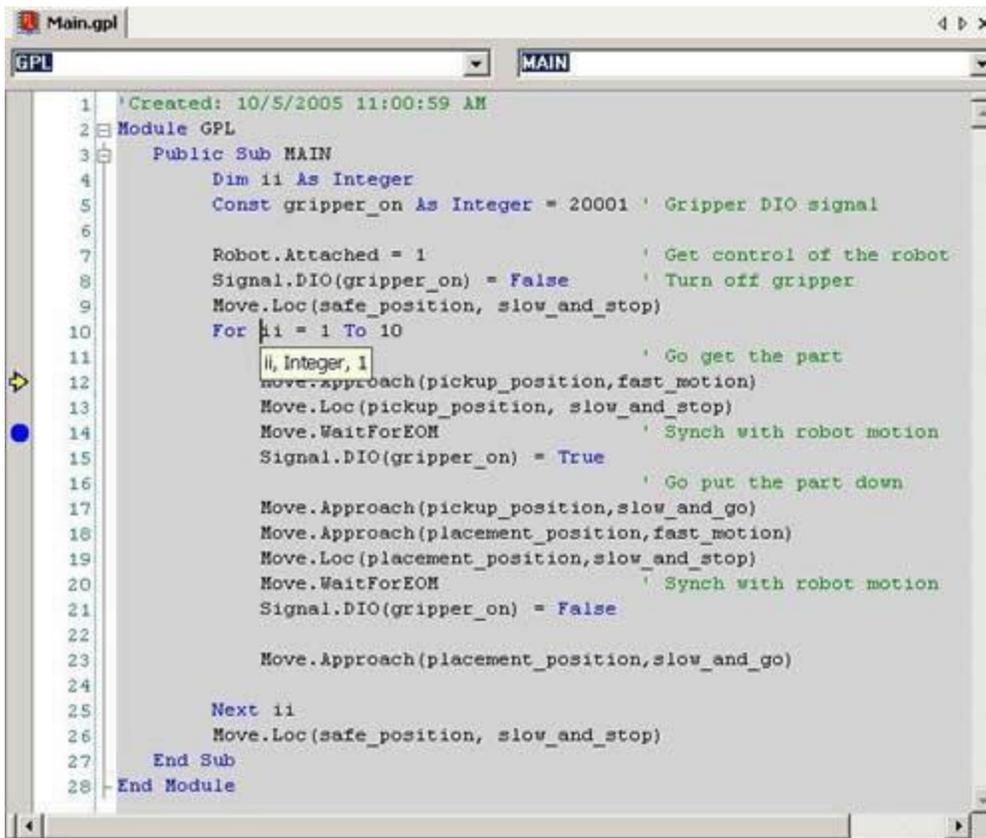
Returning to the Global Modules Files Editor, if you select an object in the top treeview panel, all of the properties of the object are displayed below in the properties window. If you select a specific property, a short description is displayed in the bottom panel.

If you would prefer to display the treeview panel and the property panel to the right and left of each other rather than vertically stacked, press the layout icon that is just above the property window and to the right.

Any objects defined within a .gpo file are globally available to all procedures within the Project. This editing window provides a very simple means of creating, managing, and teaching key global application motion control and GPL program variables.

To display the underlying text that is generated by this editor, press on the "View Source" button.

Once you start running a program, if you pause execution by hitting a breakpoint or other means, you are automatically placed into the debugger. Files being controlled by the debugger appear in their own tabs in the same manner as files being edited.



```
1 'Created: 10/5/2005 11:00:59 AM
2 Module GPL
3 Public Sub MAIN
4   Dim ii As Integer
5   Const gripper_on As Integer = 20001 ' Gripper DIO signal
6
7   Robot.Attached = 1 ' Get control of the robot
8   Signal.DIO(gripper_on) = False ' Turn off gripper
9   Move.Loc(safe_position, slow_and_stop)
10  For ii = 1 To 10
11     ii, Integer, 1 ' Go get the part
12     Move.Approach(pickup_position, fast_motion)
13     Move.Loc(pickup_position, slow_and_stop)
14     Move.WaitForEOM ' Synch with robot motion
15     Signal.DIO(gripper_on) = True
16
17     Move.Approach(pickup_position, slow_and_go)
18     Move.Approach(placement_position, fast_motion)
19     Move.Loc(placement_position, slow_and_stop)
20     Move.WaitForEOM ' Synch with robot motion
21     Signal.DIO(gripper_on) = False
22
23     Move.Approach(placement_position, slow_and_go)
24
25  Next ii
26  Move.Loc(safe_position, slow_and_stop)
27 End Sub
28 End Module
```

Figure 4-5: Debugger

As a visual cue, files that are being debugged are displayed with a gray background. Also, in the left margin, the next step to be executed is indicated by a yellow arrow. Any break points that are set

are indicated by a blue dot as in line 14 above. Break points can be set or cleared by clicking in the left margin in addition to using the selection in the top menu and toolbar.

If you hover the cursor over a variable, the variable's name, type, and current value are displayed for the current context. In the picture above, hovering is illustrated for the integer "ii." To display other variables, the right-click menu in debug mode can open the "Debug Show Variable " pop-up window. This window permits values not currently displayed to be evaluated in the current execution context.

The right-click menu also includes options to "Teach location and update editor" and "Save location back to editor." These functions allow you to single step through a program and then teach or save the position of a referenced global Location object using the Global Modules Files Editor. These operations update the Location values stored in the .gpo file for use the next time the program is loaded or executed. You can also quickly switch to the Global Modules Files Editor to view a value by highlighting a variable and selecting the "Goto Definition" entry in the right-click menu. Additionally, this menu includes a "Jog To..." selection for displaying the "Jog To..." panel that was described above. This panel permits you to easily move the robot to the selected location.

Both the standard source code editor and the debugger can have their screens split. So, multiple editors and debuggers can be simultaneously active.

For information on all of the debugging aids, please see the section on the Debug Toolbar.

Main Menu Bar

The Main Menu Bar allows you to execute the majority of the functions available within GDE. Many of the more common functions are also provided via the Toolbars or within the dockable windows.

The following tables describe the operation of each of the selections within the pull-down menus and are organized by the name of the menu.

File	Description
New Project	A new Project is created with the name specified in a subsequent popup dialog box. That popup also permits you to select whether you wish to create the Project in the controller's memory, in the flash disk, or in the PC's hard drive. The new Project will have one empty source code file and one empty global module file assigned to it. This operation is equivalent to the "Create a new project file" operation in the Project Manager.
Save	Saves the file being edited back into its Project folder. The folder can be in memory, in the flash, or on the PC hard drive. Changes to files in memory do not take effect until the file is saved back to memory.
Close All Open Editors	Closes all open text and GPO editors in the Editor and Debugger Window.
Connect	Displays the "Connect to GPL" popup window that allows you to connect GDE to a Guidance controller.
Disconnect	Terminates the connection between the controller and GDE.
Change Controller IP	Displays a pop-up that shows the IP address, serial number and name of all Guidance controllers that are active on the local network and permits their IP address to be changed. This is similar functionality to the Guidance Discovery Utility.

File	Description
Preferences	Displays the GDE Preferences window. This window allows you to change the IP address of the controller, the folder on the PC for storing GPL projects, the text font size, and other settings.
Save as HTML	Standard functions for saving the current editing window in either HTML or RTF format. Standard functions for printing or previewing the text displayed in the active editor window.
Save as RTF	
Print...	
Page Setup...	
Print	
Preview...	
Exit	

Edit	Description
Undo	Standard undo and redo functions that reverse the effect of the previous editing operations or re-instate the effect of operations that were undone.
Redo	
Cut	Standard cut, copy, paste, and delete functions that operate on the selected text within the GPL editor.
Copy	
Paste	
Delete	
Indent	For the selected GPL statements in the editor, either moves all of the text left or right by one tab (4 character positions).
Outdent	
Comment Selection	For the selected GPL statements in the editor, either inserts or deletes a comment character in the first column of each line to "comment out" or revert the code.
Uncomment Selection	
Make Uppercase	For the selected GPL statements in the editor, either converts all of the text to upper or lower case.
Make Lowercase	
Delete Horizontal Whitespace	For the selected GPL statements in the editor, deletes all leading space and tab characters.
Toggle Line Numbering	Enables or disables displaying the line numbers in the editor and debugger windows.
Mark Line Modifications	If enabled, draws a yellow bar in the left margin of all lines that have been modified but not yet saved to the Project file.
Bookmarks	Displays a submenu of functions that set, clear, and return to bookmarks in files being edited. When you set a bookmark on a line in a file, the editor remembers the line number and file name. You can then quickly scroll through all of the bookmark'ed sections of code by using the "Previous Bookmark" and "Next Bookmark" selections.
Go Back	The editor keeps a history of the lines that have been modified in all open files. These instructions move the cursor from line to line in the order that they were changed or clears the history list.
Go Forward	
Go Clear History	

View	Description
Window Layout	Provides "Load Default," "Load Layout" and "Save Layout" selections for permitting custom desktop layouts to be preserved, reloaded or set back to the normal default.
Project Manager, File Manager, ...	Contains a pull down menu selection for each of the dockable and pop-up windows. Permits windows to be re-opened if they have been closed.
Jog To...	Displays a pop-up window that permits the operator to manually move between two locations.

Debug	Description
Start with Break	Compiles and starts execution of the Project specified in the "Project" box in the main toolbar. However, execution is paused at the Project's first statement to allow you to utilize the debugging facilities
Start	Compiles and starts execution of the Project specified in the "Project" box in the main toolbar.
Compile only	Compiles the Project specified in the "Project" box in the main toolbar.
Step Into	Executes the next sequential statement in the current thread after which execution is paused once again. If the statement is a procedure call, execution is paused inside of the procedure before its first statement.
Step Over	Executes the next sequential statement in the current thread. However, if the statement is a call to a procedure, the entire procedure is executed and execution is paused at the first statement following the procedure call.
Step Out	Executes all remaining statements in the current procedure and pauses execution at the first statement following the call to the current procedure.
Stop All	Halts execution of all threads. After execution has been halted, execution can no longer be continued and must be restarted.
Toggle Break Points	Sets or clears a break point on the current line of a procedure in memory. When execution of a procedure encounters a line with a break point set, execution is paused.
Clear All Break Points	Clears all of the break points in the file currently displayed in the editor/debugger.
Open Exceptions Window	Pops up a window that permits you to determine if exceptions are to be handled in the normal fashion or if all or specific errors are to stop execution even if they occur within a Try...Catch block.
Open Debug Show Variable	Pops up a window that permits values not currently displayed to be evaluated in the current execution context.

Search	Description
Find/Replace... Incremental Search Reverse Incremental Search	Displays a standard popup window for finding or replacing text within the editing window and provides shortcuts for repeating the operation either going forward or backwards in the file.
Go To Line...	Moves the cursor to the specified line in the editor or debugger window.

Outlining	Description
Toggle Outlining Expansion	Outlining allows you to display all text lines or collapse to their first line procedures or modules within a file. This function is convenient since most files have multiple procedures and modules. Outlining allows you to display just those elements that are currently of interest to you.
Toggle All Outlining	
Stop Outlining	

Tools	Description
Web Interface	Brings up a web browser and displays a selected page in the web interface of the connected controller. The browser window is treated just like another opened file within the GDE editor and is displayed in its owned tab'ed window.

Window	Description
Split Horizontally	Divides the current editor or debugger window into 1, 2 or 4 separate panels. Each panel can be individually scrolled to view different sections of a file.
Split Vertically	
Split Four-Way	
No Splits	

Help	Description
Contents	Opens the PreciseFlex™ PreciseFlex Library in an independent window.
About GDE	Generates a popup window that displays the GDE version and ID information along with build information for key components of GDE.
Product Activation	Displays a pop-up window that is used to permanently activate GDS by applying a licensing key or deactivating GDS by deleting the key.

Main Toolbar

The Main Toolbar provides quick access to a number of commonly used functions, particularly text editing operations that are also provided in the pull-down list of the Main Menu.

The Main Toolbar, which has been split into two for easier viewing, is shown below.

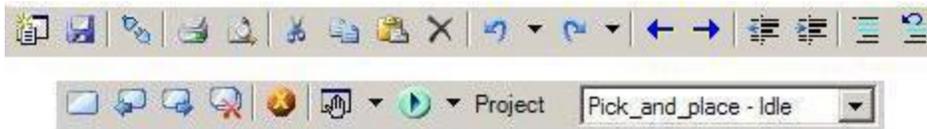


Figure 4-6: Main Toolbar

The following table describes the operations available via this toolbar.

Icon	Tool Tip Title	Description
	New Project	A new Project is created with the name specified in a subsequent popup dialog box. That popup also permits you to select whether you wish to create the Project in the controller's memory, in the flash disk, or in the PC's hard drive. The new Project will have one empty source code file and one empty global module file assigned to it. This operation is equivalent to the "Create a new project file" operation in the Project Manager.
	Save Document	Saves the file being edited back into its Project folder. The folder can be in memory, in the flash, or on the PC hard drive. Changes to files in memory do not take effect until the file is saved back to memory.
	Quick Connect / Disconnect	If GDE is not connected to a controller, displays the "Connect to GPL" popup window that allows you to connect to a Guidance controller. If GDE is connected, after verification, the connection to the controller is terminated.
	Print	Standard functions for printing or previewing the text displayed in the selected editor window.
	Print Preview	
	Cut	Standard cut, copy, paste, and delete functions that operate on the selected text within the GPL editor.
	Copy	
	Paste	
	Delete	
	Undo	Standard undo and redo functions that reverse the effect of the previous editing operations or re-instate the effect of operations that were undone.
	Redo	

Icon	Tool Tip Title	Description
	Go Back	The editor keeps a history of the lines that have been modified in all open files. These functions move the cursor from line to line in the order that they were changed.
	Go Forward	
	Outdent	For the selected GPL statements in the editor, either moves all of the text left or right by one tab (4 character positions).
	Indent	
	Comment Selection	For the selected GPL statements in the editor, either inserts or deletes a comment character in the first column of each line to "comment out" or revert the code.
	Uncomment Selection	
	Toggle Bookmark	Sets, clears, and returns to bookmarks in files being edited. When you set a bookmark on a line in a file, the editor remembers the line number and file name. You can then quickly scroll through all of the bookmark'ed sections of code by using the "Previous Bookmark" and "Next Bookmark" buttons.
	Previous Bookmark	
	Next Bookmark	
	Clear Bookmarks	
	Disable Robot Power	Immediately disables power to the robot's motors. This is equivalent to the "Disable" button in the Robot Control Window.
	Break Points	Clicking on this icon or selecting "Toggle Break Point" in the pull down list, sets or clears a break point on the current line of a procedure in memory. When execution of a procedure encounters a line with a break point set, execution is paused. A second selection in the pull down list, "Clear All Break Points," clears all of the break points in the file currently displayed in the editor/debugger.
	Compile and Run	Clicking on this icon or selecting "Start" in the pull down list compiles and starts execution of the Project specified in the "Project" box to the right. If you select "Start with Break" in the pull down list, the Project is compiled and started, but execution is paused before the first statement to allow you to utilize the debugging facilities. The "Compile Only" option is a convenience to allow you to just compile your Project to check for syntax errors. The "Stop All" selection is equivalent to the "Stop All" button in the debugger panel. This function halts execution of all active Projects. After being halted, execution cannot be continued and must be restarted.

Debug Toolbar

Whenever a Project is executing or has been paused, the Debug Toolbar is automatically displayed below the Main Toolbar. The Debug Toolbar provides easy access to the functions available for continuing, pausing, stopping and single stepping the execution of a specified thread.

The Debug Toolbar is shown below.

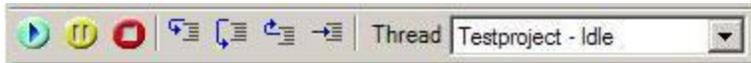


Figure 4-7: Debug Toolbar

The following table describes the operations available via this toolbar.

Icon	Tool Tip Title	Description
	Continue Execution	Continues execution of the thread specified in the "Thread" box to the right after the thread has been paused due to an error, a break point, or a Break command.
	Break	Pauses the execution of the thread specified in the "Thread" box to the right. This is equivalent to hitting a breakpoint in the specified thread.
	Stop All	Halts execution of all threads. After execution has been halted, execution can no longer be "Continued" and must be restarted.
	Step Into	Executes the next sequential statement in the specified thread after which execution is paused once again. If the statement is a procedure call, execution is paused inside of the procedure before its first statement.
	Step Over	Executes the next sequential statement in the specified thread. However, if the statement is a call to a procedure, the entire procedure is executed and execution is paused at the first statement following the procedure call.
	Step Out	Executes all remaining statements in the current procedure and pauses execution before the first statement following the call to the current procedure.
	Goto Line	Moves the next step pointer to another line within the same procedure that is currently paused. When execution is continued, it will be from the specified line.

Project Manager Window

The Project Manager Window displays Projects that are loaded in the controller's memory as well as Projects stored in the controller's flash disk and in the PC's hard drive. This window allows Projects to be created, edited, selected for execution, deleted, and transferred between the controller's memory and the disk areas. This dockable window typically looks as follows:

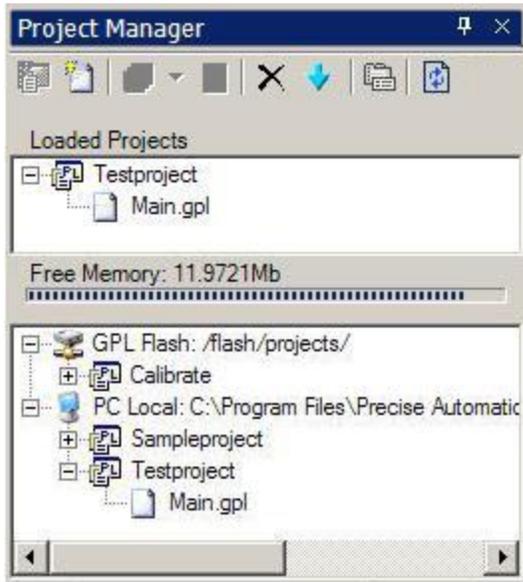


Figure 4-8: Project Manager Window

The upper window indicates all Projects loaded into the controller's memory. Once loaded into memory, these Projects are eligible to be executed. In this example, a copy of "Testproject" is loaded into memory. Just below this window, the amount of the controller's memory still available for use is displayed.

To select a Project for execution or debugging, double click on the Project in the Loaded Projects window or right click on the Project and choose **Select** in the pop-up menu. Alternatively, Project selection can be done on the top GDE menu bar. The lower window indicates the Projects stored in the controller's flash disk (/flash/projects/) directory and Projects stored in GPL Projects areas on the PC's hard drive. In this example, a Project named "Calibrate" is stored in the flash while two Projects (Sampleproject and Testproject) are stored on the hard drive.

The PC hard drive folder that contains GPL projects can be changed by accessing the [Preferences](#) pop-up panel (**File > Preferences > Project**). The Preferences panel permits multiple hard drive folders to be specified. Each specified folder will appear as a top-level node in the lower Project Manager window. This permits multiple hard drive folders to be simultaneously displayed and accessed. To edit a file within a Project, expand the contents of the Project and double click on the file of interest. You can edit Projects stored in memory, in the flash, or on the PC's hard drive.

To copy a Project between memory, the flash, or the hard drive, simply drag-and-drop the desired project. Dragging-and-dropping a Project to memory is equivalent to loading a Project in preparation for execution. Projects that have been modified while in memory must be dragged-and-dropped to flash or the hard drive if you wish to preserve the changes in the event that the controller is powered down. The following table describes the operations available via the Project Manager tool bar.

Icon	Tool Tip Title	Description
	New Project	A new Project is created with the name specified in a subsequent popup dialog box. This popup also permits you to select whether you wish to create the Project in the controller's memory, in the flash disk, or in the PC's hard drive. The new Project will have one empty source code file and one empty global module file assigned to it. This operation is equivalent to the "New Project" operation in the top GDE menu bar.
	Add Item to Project	For the selected Project, adds either an additional source code file or a global module file with the name specified in a subsequent popup dialog box.
	Save Memory To:	For a Project stored in memory, allows the operator to: "Delete from memory," "Save to flash," or "Save to PC"
	Edit File	Opens the selected component in the GDE editor. This is equivalent to double clicking on the file.
	Delete Project Or Component	After confirmation, deletes the selected Project or selected file within a Project.
	Load Project	Copies a Project stored in flash or on the PC's hard drive to the controller's memory in preparation for execution.
	Project Properties	Displays information from the Project File including the Project name and list of components. Allows the procedure, which is called when the Project is executed, to be changed. (See below)
	Refresh View	Updates the Project and component displayed for the selected device.

The following table describes the operations available by right-clicking in the Project Manager window.

Right-click	Description
Select Project	Selects a Project that is loaded into the controller's memory for execution or debugging. The Project name will be displayed in the pull-down window on the main toolbar.
Project Properties	Displays information from the Project File including the Project name and list of components. Allows the procedure, which is called when the Project is executed, to be changed. (See below)

Right-click	Description		
Set PC Project Path	These items allow the path to a PC hard drive GDE project area to be easily modified or restored. When this path is modified, the Projects in the new path are automatically displayed in the Project Manager's lower window. When GDE is restarted, the last selected path is remember and again put into effect.		
Recent Project Paths			
Import Project	Permits a Project to be copied between a folder in the PC's file system and the controller's memory, the controller's flash drive, or the hard drive GDE project area. These functions simplify sharing Projects in a common network drive and are convenient for exchanging Projects via email. In addition, Projects can be dragged from the PC's file system and dropped into the GDE project area. However, the reverse process is not currently supported.		
Export Project			
Edit	Same as the "Edit File" toolbar selection.		
New Project	Same as the "New Project" toolbar selection.		
Add New Item	Same as the "Add Item to Project" toolbar selection.		
Copy Project	Copies or duplicates the selected Project. A pop-up window is displayed that permits the destination and new name of the copy of the project to be specified.		
Delete	Same as the "Delete Project or Component" toolbar selection.		
Rename	Renames a single project file. This function can be operate on files stored in Flash, Memory and on the PC. This operation does not support renaming an entire Project.		
Protect	<p>When you wish to prevent other users from seeing or modifying your GPL source code or other data, you can encrypt ("protect") a file within a project. Protected files can be executed by anyone but cannot be edited. The protection requires a password that is embedded into the encrypted file.</p> <p>Selecting this operation will prompt the user for a password and a one line description that is shown if the file is opened in the editor.</p> <div data-bbox="326 1293 1395 1409" style="background-color: #ff9900; padding: 5px; border: 1px solid black;">  WARNING </div> <table border="1" data-bbox="326 1409 1395 1612" style="width: 100%; border-collapse: collapse;"> <tr> <td data-bbox="326 1409 1138 1612" style="padding: 5px;"> Once a file is protected, it can only be unprotected using the same password. Please be sure to keep a secure backup of the unprotected version of the file in case you forget the password. </td> <td data-bbox="1138 1409 1395 1612" style="text-align: center; padding: 5px;">  </td> </tr> </table> <p>This process can only be performed on individual files that are part of a project that is stored on the PC, not an entire project. Once a file is protected within a project, the project can then be transferred to a controller's Flash or Memory.</p> <p>Attempt to edit a protected file will invoke a read only editor with the description provided when the file was protected.</p>	Once a file is protected, it can only be unprotected using the same password. Please be sure to keep a secure backup of the unprotected version of the file in case you forget the password.	
Once a file is protected, it can only be unprotected using the same password. Please be sure to keep a secure backup of the unprotected version of the file in case you forget the password.			

Right-click	Description
Unprotect	Allows a developer to unprotect a previously protected file that is part of a project. The unprotect function can only operate on files that are stored on a PC and requires the password originally used to protect the file. If a file resides on a controller, it must first be transferred to the PC and then unprotected. This operation cannot be performed on an entire project, only individual files of a project.
Load Project to Controller Memory	Same as the "Load Project" toolbar selection.
Compile Project	Same as the "Compile Only" main toolbar selection. This is a convenience feature to allow you to just compile your Project to check for syntax errors.
Refresh	Same as the "Refresh View" toolbar selection.

When "Project Properties" is selected, the following pop-up window is displayed that contains information on the files the are contained within the selected Project.

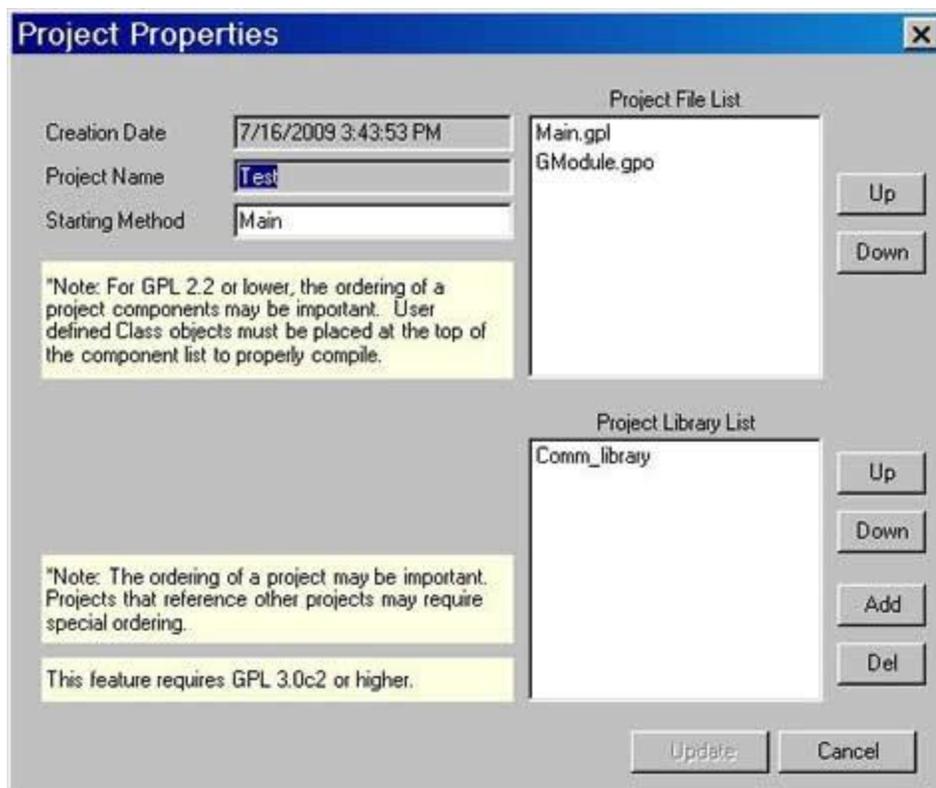


Figure 4-9: Project Properties Screen

The top right panel displays and manages the files that are contained within the Project. In GPL 3.0 and later, the ordering of files is less critical, but buttons are provided to change the ordering of the

files nonetheless. The "Starting Method" specifies the name of the GPL program that is first executed when the Project execution begins. The bottom Project Library List contains the names of other GPL Projects whose programs and data are referenced from within the specified Project. This allows you to develop libraries of GPL routines, Class definitions and Global data that can be utilized by multiple other Projects.

The files contained within each Project Library are compiled into the main Project as though they were explicitly named in the Project File List. If a GPL Project is loaded from the Flash Disk to a controller's memory, any referenced Libraries are automatically loaded. If a GPL Project is loaded from the PC, any referenced Libraries must be manually loaded into a controller's memory.

Object Browser Window

The Object Browser Window displays syntax and help information for all of the GPL statements and class methods and properties. This dockable window operates in a fashion similar to the .NET Object Browser and typically looks as follows:

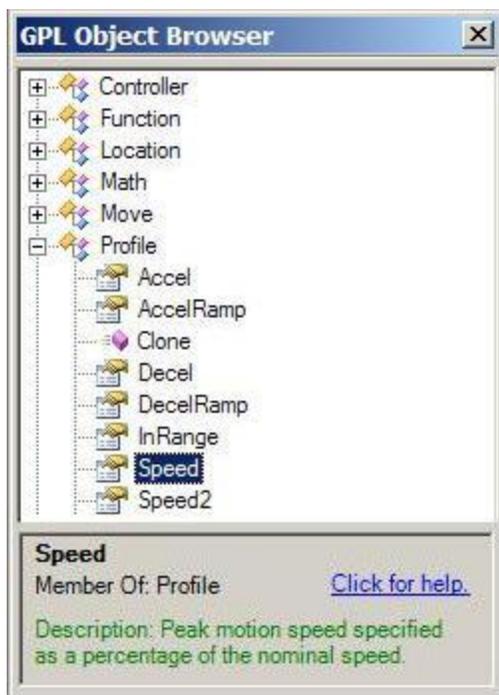


Figure 4-10: Object Browser Window

You can browse the treeview in the upper panel of the Object Browser for syntax information on specific statements, methods or properties. An icon to the left of each line provides a quick visual queue to indicate the type of the language element. When you select an item in the top panel, a short description of the language element is displayed in the lower panel. If you wish to access the complete GPL dictionary page for the item, just click on the "**Click for help**" link or double click the icon in the upper panel. This will open the PreciseFlex™ PreciseFlex Library at the dictionary page

for the selected item. As you use the GDE editor to create new program steps or modify existing steps, the Object Browser automatically updates the treeview in the top panel and the brief description in the lower panel to display the information for the language element that you are entering. The Object Browser is an information window and source of help information only, so there are no tools associated with this window.

Robot Control Window

The Robot Control Window provides the minimum functions necessary to operate and monitor the status of the robot. With this window, you can enable and disable power to the robot, home the robot, view any error and system messages that have been generated, see the current system execution state, and reduce the overall speed of the robot for testing purposes. These functions are the necessary subset of the operations available in the web based Operator Control Panel and Virtual Manual Control Pendant. A sample of the Robot Control Window is shown below.

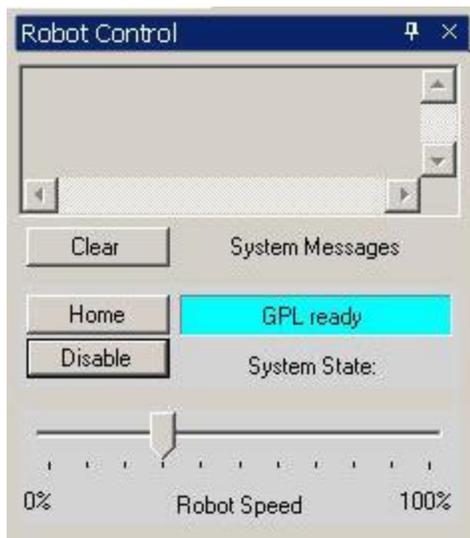


Figure 4-11: Robot Control Window

The top panel displays all recent error and system messages that have been generated by the controller. For example, when the robot is stopped due to an error condition, a message is displayed in the top panel that indicates the nature of the error. At any time, you can clear the message buffer by pressing the **Clear** button.

The **Enable/Disable** button turns the power to the robot's motors on and off. The **"Home"** button executes the robot's homing sequence to establish the zero position of each axis when the controller is restarted. The text box to the right of these buttons displays the current execution state of the system. For example, it indicates if robot power is currently enabled and whether a hardware E-Stop is being asserted. Prior to executing a GPL program that moves the robot, the robot power must be enabled, the robot must be homed, and the System State must indicate "GPL ready." If you are not familiar with the procedures that are necessary for initializing the controller and robot, please

see the “*Guidance System Setup and Operation, Quick Start Guide*”. The bottom **Robot Speed** slider is equivalent to the speed control on the web Operator Control Panel. This allows you to proportionally slow down the overall speed of the robot while you are debugging a new Project. Changes to the slider take effect immediately even during the middle of a motion.

Virtual Pendant Window

As a convenience, GDE includes a Virtual Manual Control Pendant that permits the robot to be manually jogged in the same manner as the VMCP that is built into the controller's web interface. This panel displays the current position of the robot as well as the tool transformation that is currently in effect. As an aid to setting the top speeds and accelerations for applications, this panel also includes the % Duty Cycle and instantaneous torque for each motor as a percentage of the rated motor torques. A sample of the Virtual Pendant Window is shown below.



Figure 4-12: Virtual Pendant Window

GPL Output Window

The GPL Output Window displays all output from the controller that is generated in connection with executing a GPL program. For example, when you compile a GPL program, the output of the compiler including any error messages will be displayed in this window. In addition, if your GPL program generates any text output, e.g. by executing a "Console.WriteLine" method, this output will also be displayed in the GPL Output window. If the GPL Output window is closed when you begin execution of a Project, this window will automatically be opened.

A sample of the GPL Output window is shown below after the "Testproject" has been successfully compiled and executed.

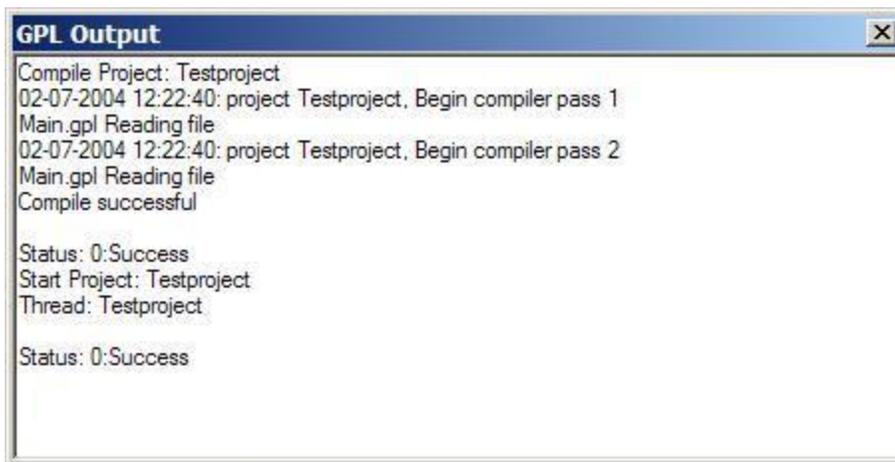


Figure 4-13: GPL Output Window

If you want to "**Clear**" the contents of this window or "**Copy**" the contents to a file, right click anywhere in the window to get a popup menu to execute these operations. For the copy function, any text that is currently selected in the window will be copied to the Windows copy/cut/paste buffer.

As a debugging convenience, if you compile a program and any compiler errors occur, the GPL Output window will split and the compiler errors will be listed in a tabular "Goto" list.

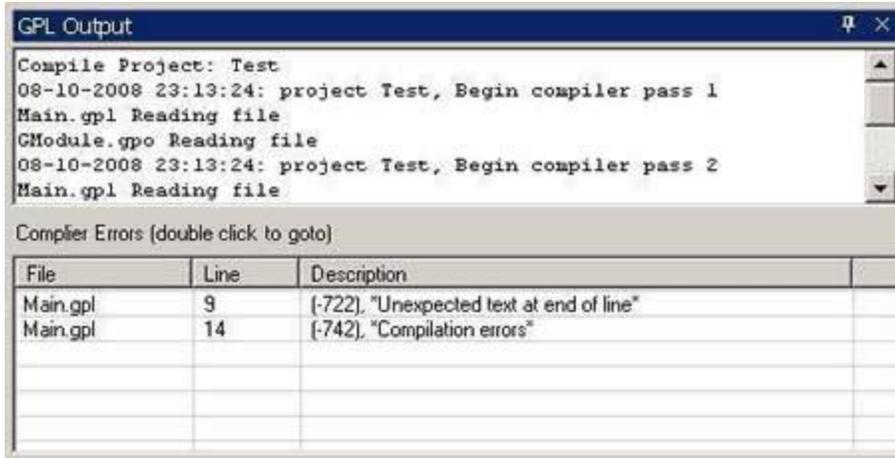


Figure 4-14: Compile Errors

You can then double click on any line in the Goto list and the cursor in the Editor Window will jump to the text line that generated the error.

Threads Window

The Threads Window displays status information for each active execution thread in the controller. The main procedures for your Project will always run in their own thread. In addition, more complex applications may initiate additional threads to allow independent execution of selected code segments. A sample of the Threads Window is shown below where each line displays the information for a different execution thread.

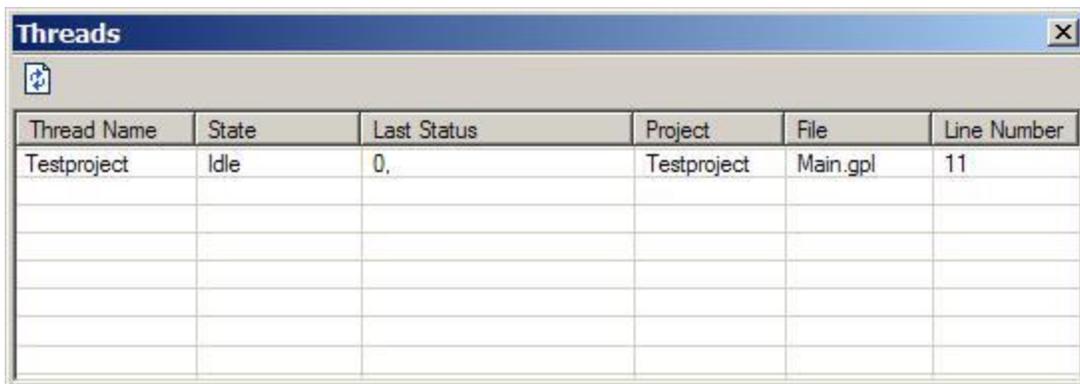


Figure 4-15: Threads Window

The first column specifies the name of the thread. The thread name is normally the same as the Project name. The "State" indicates if the thread is running or has ceased execution for some reason (e.g. paused due to a breakpoint or error). The "Last Status" displays any error message that was generated when the thread ceased execution. The "Project" displays the name of the Project

The first column specifies the frame number where 0 indicates the frame currently being executed, i.e. the top of the stack. The "Project" is the name of the Project being executed and "Process" is the name of the procedure being executed. The "Proc Line" is the number of the statement being executed relative to the start of the procedure. The "File" is the name of the source code file that contains the procedure. The "Line" column indicates the number of the statement being executed relative to the start of the file. Finally, the "Size" displays the size of the stack frame in kbytes.

For more information on the interpretation of these values, please see the documentation for the "Show Stack" Console Command.

File Manager Window

The File Manager provides access to the entire file structure of the controller. This is wider access than is available via the Project Manager, which only displays the Project area of the flash disk and the controller's memory. The entire file structure includes: all folders on the flash, such as the Parameter Database PAC files in the "/config" folder; the ROMDISK (i.e. the in-memory simulated disk) that contains the Operator Interface web pages; and the GPL memory image that displays the Projects loaded in the controller's memory.

The File Manager is provided as a convenience for displaying files not accessible using the Project Manager such as the Project.gpr files. Should you wish even greater read/write/drag-and-drop access to the controller's file system, you should utilize the File Manager that is available as part of the Backup and Restore function of the web Operator Interface. A sample of the File Manager Window is shown below. In this example, the contents of the flash's project folder and the GPL folder correspond to the contents displayed in the Project Manager Window example in this document.

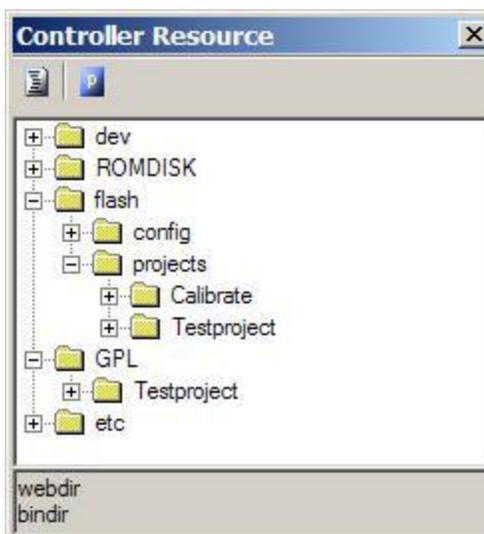


Figure 4-17: File Manager Window

The following table describes the operations available via the File Manager tool bar.

Icon	Tool Tip Title	Description
	Edit File	Opens the selected file in the GDE editor. This is equivalent to double clicking on the file.
	Preview File	Opens the selected file in a read-only text display window. This can be used for viewing any file.

Console Window

The Console Window provides access to the GPL Console Commands. This window is equivalent to connecting to the serial port of the controller. The Console Commands are simple, non-graphic text commands that perform rudimentary operations such as displaying the current memory utilization.

During normal operation and software development, you should not need to issue Console Commands since their functionality plus more is provided by the web Operator Interface and GDE. However, the Console Window is provided in GDE for completeness.

A sample of the Console Window is shown below where two typical commands have been issued.



Figure 4-18: Console Window

For more information on the GPL Console Commands, please see the Software Reference section of the PreciseFlex™ PreciseFlex Library.

Preferences Panels

The Preferences Panels contain a number of application setup parameters that customize the operation and appearance of GDE. Once updated, these preferences are automatically reloaded and put into effect each time GDE is restarted.

The Preference Panels appears as shown below.

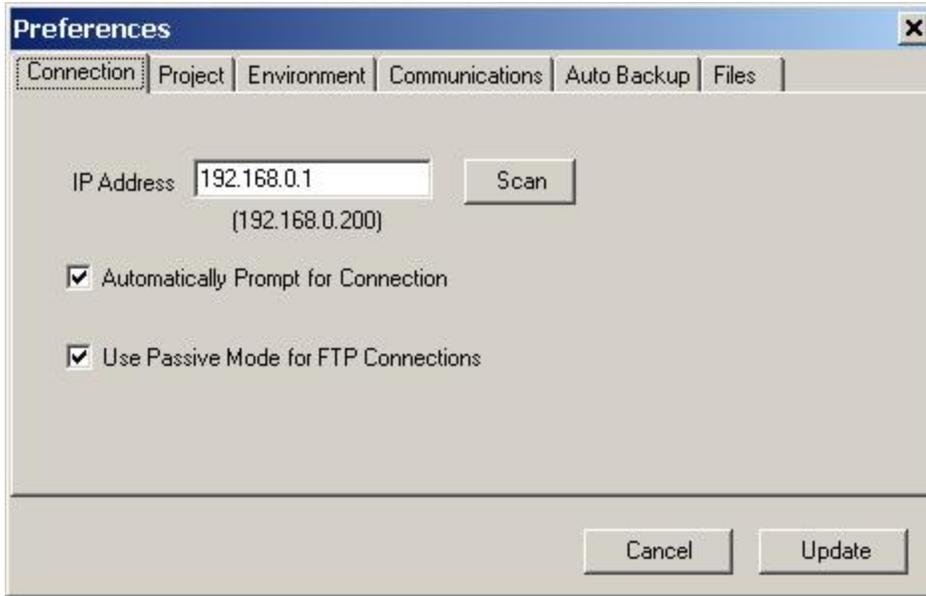


Figure 4-19: Preferences Panel

When any data in the Preferences has been modified, the **"Update"** button must be pressed before the change will take effect and be preserved.

In the following table, the configuration information presented on each tab of these panels is briefly described.

File	Description
Connection	Defines the IP address of the target motion controller on which developed GPL programs are to be executed. Includes a Scan button for locating all Guidance controllers that are connected to the local network. Also specifies if a pop-up prompt should be displayed when GDE begins execution to establish communication with the controller. <i>By default, the "Use Passive Mode for FTP Connectors" should be checked for reliable operation.</i> In certain installation where firewalls are enabled, this parameter may have to be unchecked in order for GDE to operate. However, unchecking this property can result in GDE occasionally locking up.
Project	Defines one or more folders in the PC's hard drive where GPL projects are stored. These folders are displayed in the Project Manager to permit projects to be quickly opened, edited, and loaded into the target controller.

File	Description
Environment	Specifies various options that affect how information is displayed in GDE and enables side-effects of some actions. For example, this panel defines the font size for displaying GPL programs, controls automatically displaying the values of variables using the mouse hover mode, determines if the focus of the Object Browser should automatically change to track the instruction being edited, etc.
Communications	As a diagnostic aid, enables automatic logging of all network communication between GDE and the target controller.
Auto Backup	Enables automatic back up of GPL projects when (1) a project is loaded from the PC's hard drive into the target controller's volatile memory or (2) a project in the target controller's memory is edited and the changes are written back to the controller's memory. This helps to eliminate losing important changes if the target controller is inadvertently turned off before modified projects are saved to permanent storage. The backup copies of the projects are saved in a specified folder on the PC's hard drive. If desired, the backup folder can be displayed in the Project Manager by specifying this folder using the "Project" panel described above.
Files	Includes a "Save .gpl files with international character (UTF-8 BOM) header" check box. <u><i>This box should ONLY be checked if you wish to insert UTF-8 international characters into code files using a 3rd party offline editor like Notepad.</i></u> If you only use GDE to edit files, international characters can be inserted without checking this box. In either case, GPL version 4.0F1 or later is required to support UTF-8 characters in Strings and comments. Each time a GPL code file is written, if this box is checked, GDE will insert the UTF-8 BOM 3-byte header into the code files. This header will make the code file incompatible with earlier versions of GPL and GDS.

5. GDE Programming Examples

Hello World Example

Now that you have an understanding of the features of the Guidance Development Environment, we want to put that knowledge to work by developing and executing the traditional "Hello World" computer program. In this exercise, you will learn how to create a Project, write a simple procedure that outputs the text "Hello World," load the Project into the controller, and then execute it.

At this time, your GDE environment should be connected to your controller and show look approximately like the following. Don't worry if you have re-arranged any of the windows or added any projects to the system. Also, since we will not be moving the robot, the power to the robot need not be enabled.

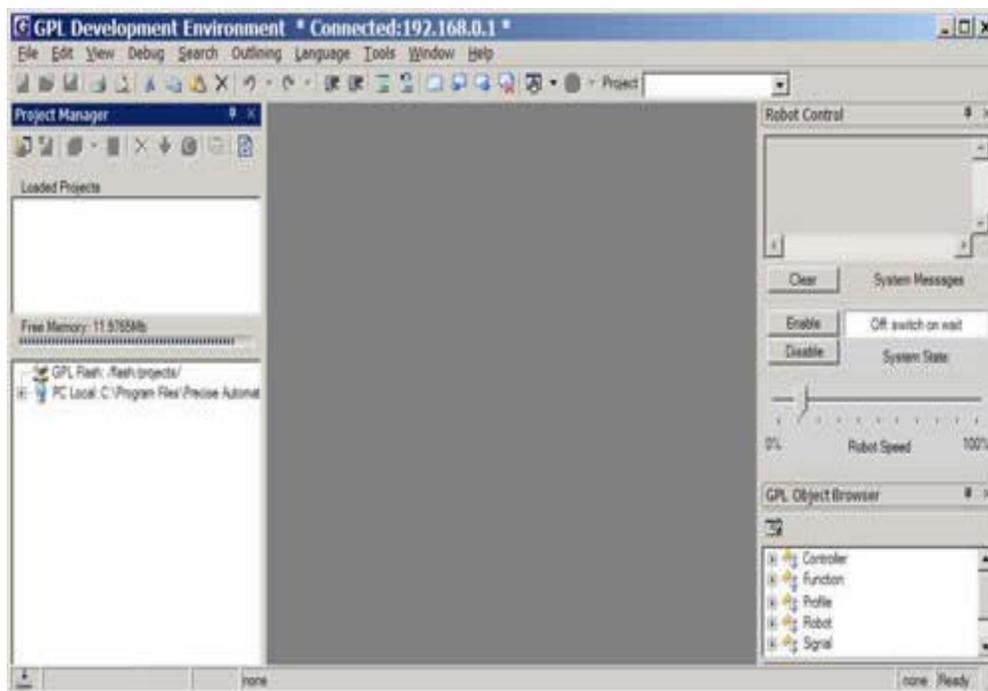
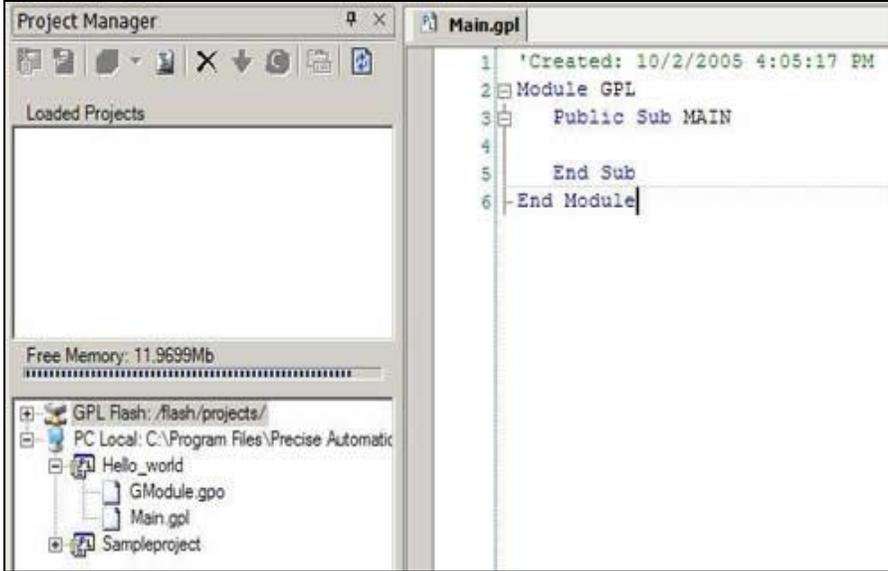


Figure 5-1: GPL Development Environment Window

The first step is to create a new Project. For this exercise, you will create this Project on your PC's hard drive although the Project could just as easily be created on the controller's flash disk.

Step	Action
1.	To create a new project, click on File > New Project .
2.	In the Add New Project popup window, enter "Hello_world" for the Project name.
3.	Ensure that PC hard drive is selected, and click the OK button.
4.	<p>To bring up the main source code file for editing, in the Project Manager, expand the new Hello_world Project folder and double-click the Main.gpl file.</p> <p>A zoomed view of the Project Manager and the GDE editor will now look like the following:</p> 

When creating the Hello_world Project, GDE automatically adds a main source code file and a global modules file to new Projects as a convenience. Also, the "Main.gpl" file already contains the definition for the "MAIN" public procedure. By default, this is the procedure that will be started when executing the project, although this can be easily changed by accessing the Project properties.

Next, we want to edit the MAIN procedure and add statements to output the desired text. Then, we want to load the Project into the controller in preparation for execution.

Step	Action
1.	<p>In the editor window, below "Public Sub Main," insert the following lines of text:</p> <pre>console.writeline("") console.writeline("Hello world!") console.writeline("")</pre>

Step	Action
2.	Click the Save Document icon on the main tool bar to save changes.
3.	In the Project Manager, drag the Hello_world Project from the PC to the top Loaded Projects panel and drop it in.

Some aids are built into GDE. First, after typing in "console," the system detected that the user was referring to the standard Console class and it displayed a popup that the user could use to automatically select "Writeline." Secondly, even though the user typed "console" in lower case, the system automatically changed the case of this word. Finally, as the user typed, the Object Browser was changing its contents to highlight the syntax for "writeline" as well as showing a brief description of this method.

At this point, the new Project is loaded into the controller and is ready to execute. An expanded view of the Project Manager and editor should now look as follows:

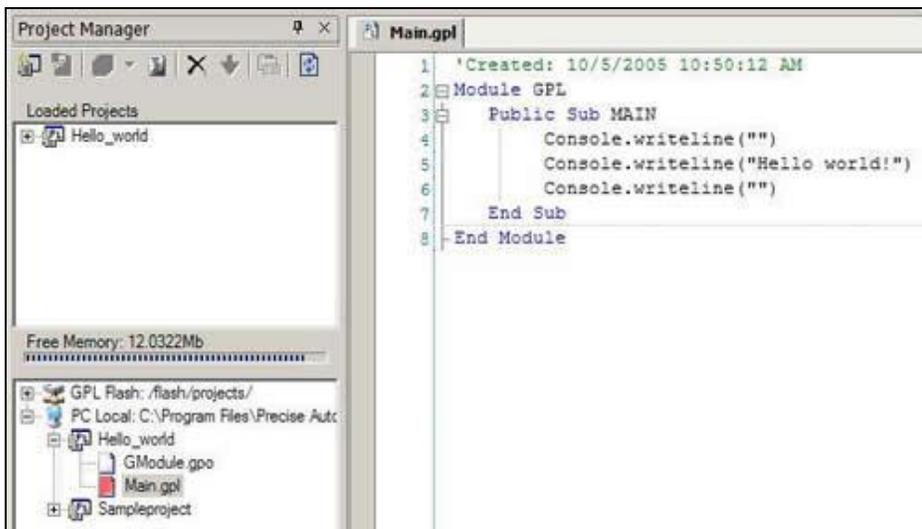
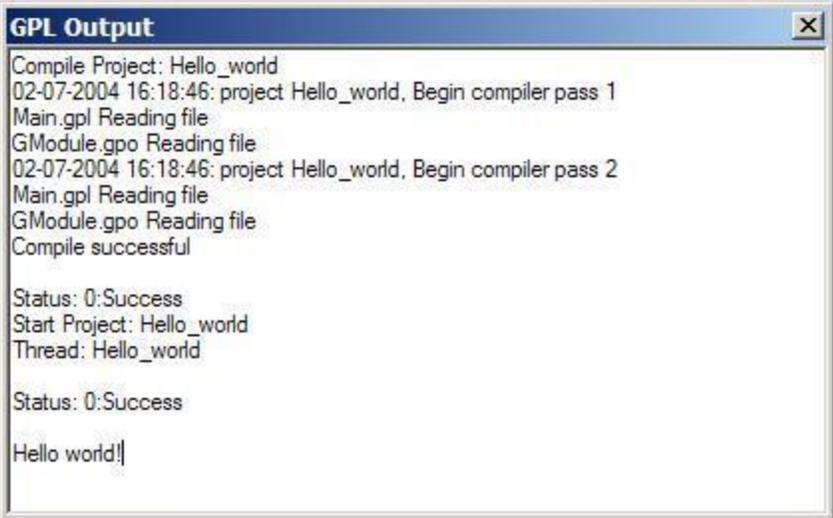


Figure 5-2: Project Manager Window

The Project is ready to be executed.

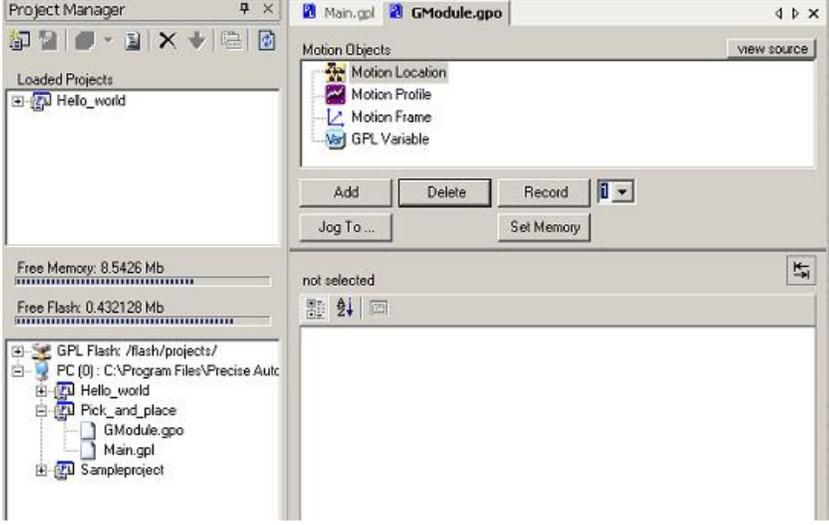
Step	Action
1.	Either double click on the Hello_world Project in the Loaded Projects panel of the Project Manager to select the project for execution or select the Project name in the top tool bar in the Project box.
2.	<p>In the top tool bar, press the Compile and Run button (the green button with the right arrow). The GPL Output window should display text similar to that illustrated below with the "Hello world!" message shown at the bottom.</p>  <pre> Compile Project: Hello_world 02-07-2004 16:18:46: project Hello_world, Begin compiler pass 1 Main.gpl Reading file GModule.gpo Reading file 02-07-2004 16:18:46: project Hello_world, Begin compiler pass 2 Main.gpl Reading file GModule.gpo Reading file Compile successful Status: 0:Success Start Project: Hello_world Thread: Hello_world Status: 0:Success Hello world! </pre>

If the user copies this Project to the flash disk, they could now utilize the web Operator Control Panel to load and execute the new Project entirely from the web interface without the use of the PC. For instructions on this procedure, please see the *Guidance System Setup and Operation, Quick Start Guide*.

Pick and Place Example

In this exercise, the user will develop a GPL Project that performs a simple simulated pick-and-place operation. A pick-and-place operation moves the robot's tool tip to a position to pick up a part and then moves to a second position to drop off the part. In order to clear any possible obstacles and to avoid dragging the part, the tool tip is retracted after picking up the part and then approaches the place location slightly over the placement position.

For this application, the robot power must be enabled and the robot must be homed. In addition, the web Guidance Operator Interface will be used to manually move the robot to teach the pick and place locations. As in the Hello World example, the starting point for this exercise is to create a new Project. However, the second step in this process will be to define the global **Location** and **Profile** data rather than writing the program.

Step	Action
1.	To create a new project, click on File > New Project .
2.	In the Add New Project popup window, enter "Pick_and_place" for the Project name, ensure PC hard drive is selected, and click OK .
3.	<p>To bring up the global module file for editing, in the Project Manager, expand the new Pick_and_place Project folder and double click on the GModule.gpo file.</p> <p>At this time, the GDE application should be displaying the panel for editing and managing global robot data. The zoomed-up view of the Project Manager and editor should look as follows:</p> 

Next, define and teach three motion **Locations**: the simulated pick-up position, the placement position, and a safe position. Start with the safe position. This should be a robot location that is above the work surface and one that can be safely reached from most positions in the work envelope.

Step	Action
1.	Using the Virtual Manual Control Pendant in the web Operator Interface, move the robot to a "safe" location.
2.	To create a new global Location object, in the Motion Objects panel, click the Motion Location icon.
3.	Click the Add button.
4.	In the pop-up window that requests the Location's name, enter safe_position and click Ok.
5.	To set the position of the new Location equal to the current position of the robot, click safe_position in the Motion Objects panel, click the Record button at the bottom of the Motion Objects panel and click Yes to confirm the operation in the pop-up window.

After these steps are completed, the main editing display should look like the following.

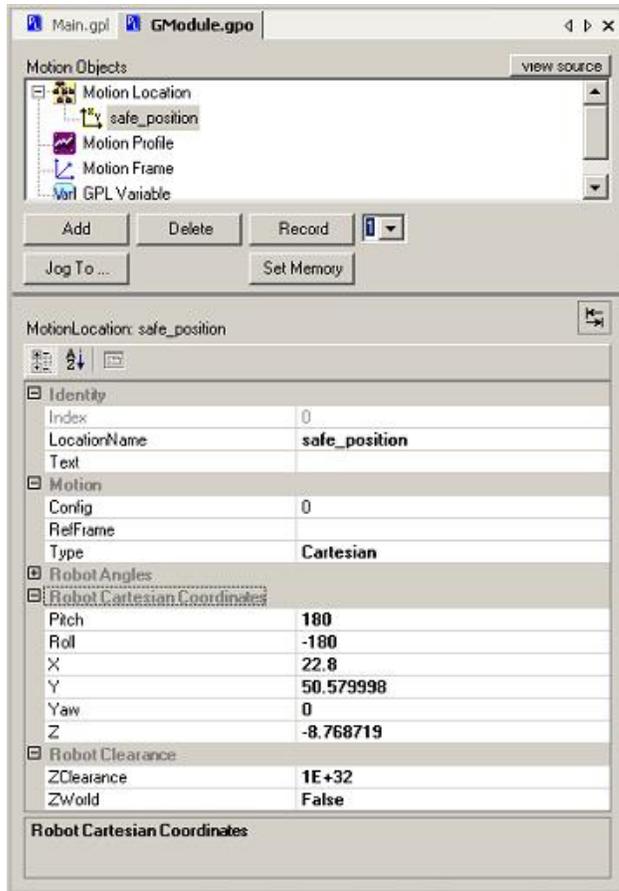


Figure 5-3: Main Editing Display

Since new **Locations** are created with a type of "Cartesian" by default, we have collapsed the "Robot Angles" properties in the picture above as these properties are only meaningful for "Angles" **Locations**. The important information to note is that a new **Location** object is displayed in the top window and the property window shows its property values along with its object name. The user's actual "Robot Cartesian Coordinates" will be different since they will reflect the user's actual safe position.

NOTE: The **ZClearance** value is huge since we have not defined a clearance distance for this position.

Repeating this procedure, define **Location** objects for the simulated pickup and the placement positions. For these positions, the user should select places in their workspace that are clear of obstacles and, just to be safe, 20-40 mm above the work surface.

Repeat the procedure above to move the robot to the pickup position, create a new global **Location** object named **pickup_position**, and record the robot location.

For this position, set the **ZClearance** value to a height above the pickup location that will allow the user to move to the pickup location without hitting any obstructions. For example, enter a small clearance value, such as 25 mm to 50 mm for the **ZClearance** property.

Repeat the procedure again and move the robot to the placement position, create a new global **Location** object named **placement_position**, and record the robot location. For this position, set the **ZClearance** value to a height above the placement location that will allow the user to move to the placement position without hitting any obstacles. For example, enter a small clearance value, such as 25 mm to 50 mm for the **ZClearance** property.

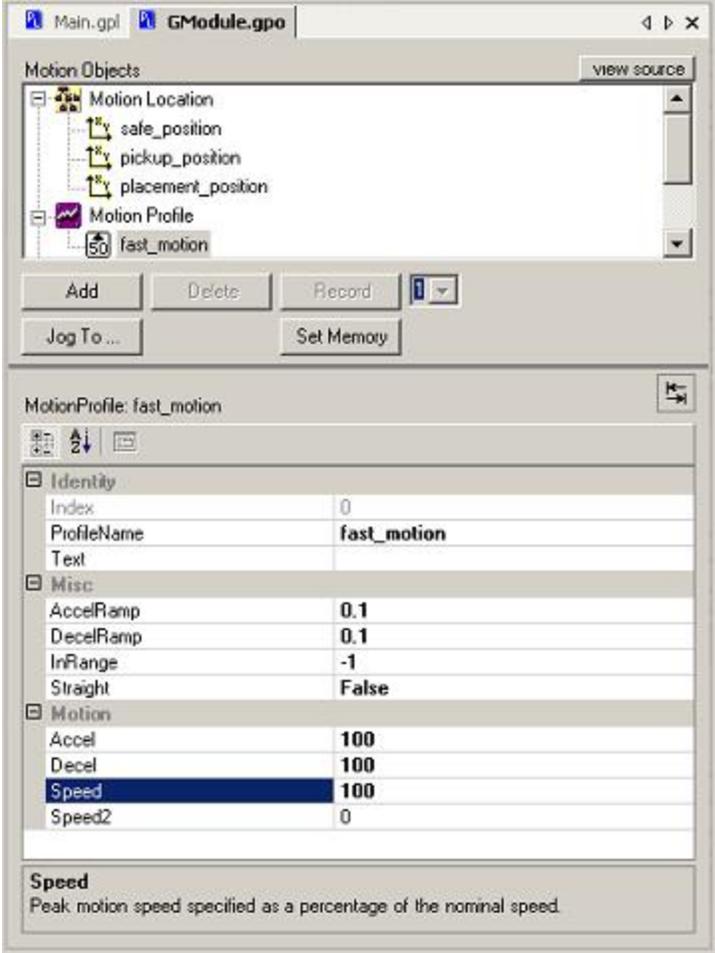
Click the **Save Document** icon in the main tool bar to store the new **Location** values.

This finishes creating and defining all of the robot position data that will be needed for this simple robot application.

Next, define some global motion **Profile** objects. These will be used to control the speed of the robot during the various types of motions and will determine when the robot stops. You will be creating three **Profiles**: one that slews the robot at high speed, a second that moves the robot at a slower speed and stops, and a final value that moves the robot at a slow speed but doesn't stop at the end of the motion.

Follow the procedure below to create a new global **Profile** object.

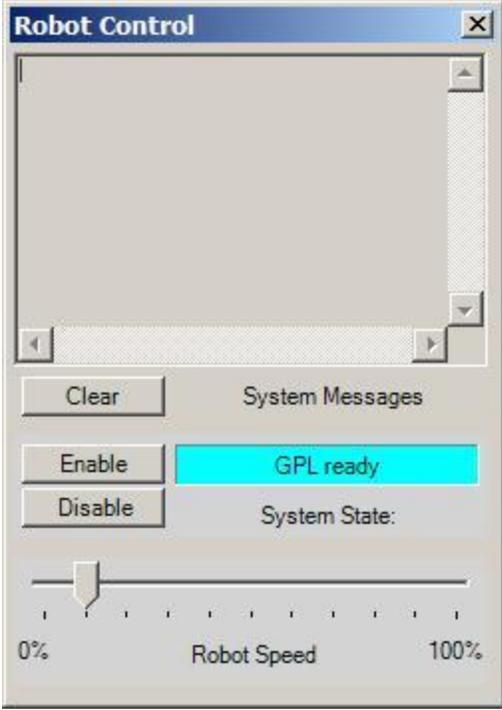
Step	Action.
1.	In the Motion Objects panel, click on the Motion Profile icon and click the Add button.
2.	In the pop-up window that requests the Profile's name, enter fast_motion and click Ok .
3.	In the properties window, set the Speed , Accel , and Decel to 100 for a high speed motion, and set Speed2 to 0 to ignore this secondary property for now.

Step	Action.
4.	<p>Set InRange to -1 to indicate that the robot does not have to stop at the end of the motion and this motion should be smoothly blended with the next motion if possible. With this data, the main editing display should look as follows:</p>  <p>The screenshot shows the GDE software interface with the following details:</p> <ul style="list-style-type: none"> Motion Objects: A tree view showing 'Motion Location' (containing 'safe_position', 'pickup_position', 'placement_position') and 'Motion Profile' (containing 'fast_motion'). MotionProfile: fast_motion: A configuration panel with the following settings: <ul style="list-style-type: none"> Identity: Index: 0, ProfileName: fast_motion, Text: (empty) Misc: AccelRamp: 0.1, DecelRamp: 0.1, InRange: -1, Straight: False Motion: Accel: 100, Decel: 100, Speed: 100, Speed2: 0 Speed: A note at the bottom states: "Peak motion speed specified as a percentage of the nominal speed."
5.	<p>Repeat the process to create a second Profile object named slow_and_stop. Set Speed, Accel and Decel to 25, InRange to 10, and Speed2 to 0. This InRange setting will force the robot to stop at the destination and delay until the final position is roughly achieved.</p>
6.	<p>Repeat the process to create a third Profile object named slow_and_go. Set Speed, Accel and Decel to 25, InRange to -1, and Speed2 to 0. Like fast_motion, the robot will not stop at the destination position.</p>
7.	<p>Click the Save Document icon in the main tool bar to store the new Profile values.</p>

This completes the generation of the global data. To write the GPL program that will make use of this information, complete the following procedure.

Step	Action
1.	In the Pick_and_place Project folder, double click on the Main.gpl file. To bring up the main source code file for editing, in the Project Manager,
2.	<p>In the editor window, below the "Public Sub Main" statement, insert the following lines of text. If you are reading this exercise in the PreciseFlex™ PreciseFlex Library (the online help file), copy and paste the text.</p> <pre> Dim ii As Integer Const gripper_on As Integer = 20001 ' Gripper DIO signal Robot.Attached = 1 ' Get control of the robot Signal.DIO(gripper_on) = False ' Turn off gripper Move.Loc(safe_position, slow_and_stop) For ii = 1 To 10 ' Go get the part Move.Approach(pickup_position, fast_motion) Move.Loc(pickup_position, slow_and_stop) Move.WaitForEOM ' Synch with robot motion Signal.DIO(gripper_on) = True ' Go put the part down Move.Approach(pickup_position, slow_and_go) Move.Approach(placement_position, fast_motion) Move.Loc(placement_position, slow_and_stop) Move.WaitForEOM ' Synch with robot motion Signal.DIO(gripper_on) = False Move.Approach(placement_position, slow_and_go) Next ii Move.Loc(safe_position, slow_and_stop) </pre>
3.	Click the Save Document icon on the main tool bar to save changes.
4.	To load the application, in the Project Manager, drag the Pick_and_place Project from the PC to the top Loaded Projects panel and drop it in.

This completes all of the software development and teaching for the Project. It's time to execute it on the controller. Before doing that, perform the following procedure to ensure that the controller is in the proper state to run the program.

Step	Action
1.	Verify that the robot power is on and that the robot has been homed.
2.	<p>Place the Virtual Manual Control Pendant back into Computer mode. This allows an automatic program to take control of the robot.</p> <p>The Robot Control Window should look as shown below. The System State should display GPL ready, indicating that power is enabled and a GPL program can take control of the robot.</p>  <p>The screenshot shows a window titled 'Robot Control'. It contains a 'System Messages' section with a 'Clear' button. Below that are 'Enable' and 'Disable' buttons, with 'GPL ready' displayed in a cyan box. Underneath is a 'System State:' label. At the bottom is a 'Robot Speed' slider ranging from 0% to 100%.</p>
3.	In the Robot Control Window, set the Robot Speed to approximately 5%. This will force the Project to move the robot at 1/20th of the normal speed for this application.
4.	Either double-click the Pick_and_place Project in the Loaded Projects panel of the Project Manager to select the project for execution, or select the Project name in the top tool bar in the Project box.
5.	<p>NOTE: This next action will move the robot automatically, so be prepared to hit the E-Stop button or disable power if any problem occurs!</p> <p>In the top tool bar, click the Compile and Run button (the green button with the right arrow).</p>

The Project should now be slowly moving the robot through its pick-and-place operation.

If the robot is moving safely in the workspace, the overall speed can be gradually increased in the Robot Control Window. To stop that Project at any time, click the yellow **Break** button in the Debug toolbar.

Also, using the web interface, the user can view the state of digital signal 20001 on the Soft Internal IO panel. The simulated gripper signal will turn on then off as the robot reaches its pickup and placement positions, respectively.

As a final step, in preparation for executing this application in a standalone controller, i.e. with the Guidance Development Environment disconnected, copy the Project to the controller's flash disk.

Independent of whether the robot is moving or stopped, to copy the application to the flash, in the Project Manager, drag the **Pick_and_place** Project from the PC to over the **GPL Flash: /flash/projects/** icon in the same panel and drop it.

With the Project in flash, the user can now execute the application using only the web interface. For detailed instructions on executing GPL Projects via the web Operator Interface, please see the *Guidance System Setup and Operation, Quick Start Guide*.

Safety Zones

For all robot types, "Safety Zones" can be defined that disable motor power and halt the motion of the robot if its tool center point (TCP) violates the requirements of a user defined 3D volume.

Types of Safety Zones

These 3D safety zones can be used to:

1. Approximately model the volume of stationary objects or personnel working areas to prevent the robot from inadvertently entering this volume and causing a collision ("keep out zones").
2. Reduce the normal working volume of the robot to prevent the robot from reaching beyond prescribed boundaries and causing a collision ("stay within zones").
3. Verify that the robot's TCP speed (when in a specified volume) is below a specified limit so that the robot can be safely decelerated and stopped before it might pin an operator's hand to a hard surface with too high a force ("speed restrict zones").

As currently implemented, the "keep out zones" and "stay within zones" are provided as general safety features, but they do not meet the stringent Category 3 safety standards that require fail safe redundant logic. However, the "speed restrict zones" do provide the requisite redundancy and are in the process of being Category 3 certified. The "keep out zones" and "stay within zones" are collectively referred to as "uncertified zones" and the "speed restrict zones" are referred to as "certified zones".

The supported zone shapes are rectangular volumes, cylinders and spheres. To define a safety zone, the type of safety zone must be specified along with its origin and dimensions.

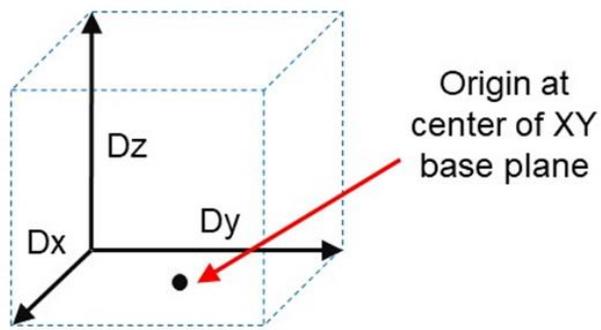


Figure 5-4: Rectangular Volume

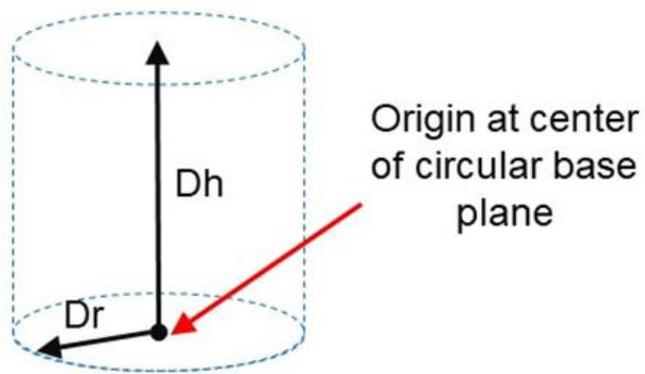


Figure 5-5: Cylinder

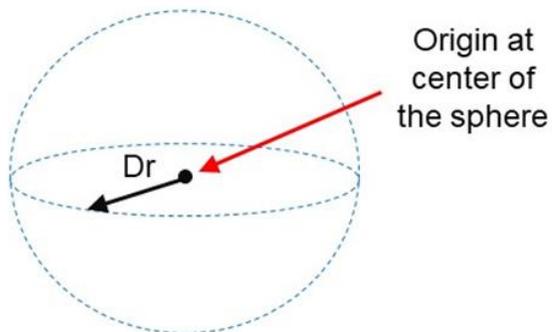


Figure 5-6: Sphere

For increased generality, uncertified zones can be arbitrarily positioned and rotated in all three dimensions. Due to implementation limitations, certified zones must be non-rotated rectangular volumes, which can be arbitrarily positioned. Up to 10 zones can be defined for each robot and any mix of certified and uncertified can be specified. Due to safety requirements, any new or modified zone specifications only go into effect after the controller is rebooted.

Safety Zone Violation Detection and Clearing

Uncertified safety zones are active in the following circumstances:

- Continuously during program-controlled motions of all types (straight line or arc Cartesian and joint interpolated).
- Continuously during manual (jog) control modes: World, Tool and Joint, but not free.
- Motion planning (final destinations only).
- Location object.KineSol method during conversions to either Cartesian or joint Locations.

Certified safety zones are only active during program-controlled motions (of all types) since this is the only circumstance where higher TCP speeds are possible.

When motor power is enabled and the robot's TCP is in violation of an uncertified safety zone, a program-controlled motion cannot be initiated. This condition can be cleared by disabling motor power and manually repositioning the robot or by manually jogging the TCP in World, Tool or Joint modes, so long as the jog motion reduces the safety zone violation distance. That is, jogging motions that increase the violation of a safety zone are not permitted.

NOTE: Safety zone testing is based on the TCP of the robot. Therefore, it is very important that the position of the tool center point relative to the robot's tool mounting flange is set correctly. Please see the Robot.Tool property for information on defining the TCP.

Certified Speed Restrict Safety Zones

While the uncertified safety zones perform conventional tests on just the position of the TCP, there are two certified safety zones and these perform special tests to detect if the speed of the TCP exceeds a limit while the TCP is within the zone. The first certified safety zone tests if the Z downward speed of the TCP exceeds a specified limit. This safety zone was implemented for the PreciseFlex 400 and PreciseFlex 3400 robots since their only intrinsically non-safe motion is a high-speed downward Z motion that could trap a person's hand between the tooltip and a fixed object or horizontal surface. The second certified safety zone tests if the horizontal, XY planar, speed of the TCP exceeds a specified limit. This test was developed for the PreciseFlex™ DD robots since robots can generate excessive speeds when moving horizontally.

For both of these tests, in order to satisfy the computational redundancy requirement of the Category 3 safety regulations, the shapes of these safety zones are limited to non-rotated rectangular volumes.

Please consult the user manuals for these PreciseFlex™ robots for when speed restrict safety zones must be defined to safely operate these mechanisms.

Configuring Safe Zones

Up to 10 safety zones can be defined per robot. Each of these zones is specified by filling in one of the Parameter Database IDs 16900 to 16909, which are labeled "Safety Zone: type, x/y/z/y/p/r, dim 1/2/3". Any combination of certified and uncertified safety zones can be specified. Each of these DataIDs consists of an array of 10 numbers and the first value defines the safety zone "type". Any DataID that has a zero "type" is ignored. [Table 5-1](#) describes the possible safety zone types:

Table 5-1: Safety Zone Types

Safety Zone Type	Description
0.	Undefined safety zone
1.	Rectangular volume, keep out zone
2.	Cylinder, keep out zone
3.	Sphere, keep out zone
4.	Rectangular volume, stay within zone
5.	Cylinder, stay within zone
6.	Sphere, stay within zone
7.	Non-rotated rectangular volume, Z downward speed restrict zone
8.	Non-rotated rectangular volume, XY speed restrict zone

[Table 5-2](#) describes the safety zone DataIDs. When any of these DataIDs are modified, the controller must be rebooted for the change to be put in effect.

Table 5-2: Data IDs

DataID	Parameter Name	Description
16900 to 16909	Safety Zone: type, x/y/z/y/p/r, dim 1/2/3	<p>Each safety zone definition consists of an array of 10 values. The first value is the safety zone “type”. The next six values define the position of the origin of the volume of interest and its orientation. This is specified as a standard Location value:</p> <p style="text-align: center;">x, y, z, yaw, pitch, roll.</p> <p>The final three values define the size of the volume of interest. For the permitted shapes, this is interpreted as:</p> <p style="text-align: center;">volume: Dx, Dy, Dz Cylinder: Dh, Dr, 0 Sphere: Dr, 0, 0</p> <p>For example, for a downward Z non-rotated rectangular volume speed restrict safety zone, a single DataID should be specified as follows:</p> <p style="text-align: center;">7, x, y, z, 0, 0, 0, Dx, Dy, Dz</p> <p>Where x, y, z are the coordinates of the center of the base of the rectangular volume and Dx, Dy, Dz are the dimensions of the volume, all in mm.</p>

In addition, the DataID in [Table 5-3](#) must be initialized to establish the maximum speed limits for the certified safety zones:

Table 5-3: Data IDs

DataID	Parameter Name	Description
2740	Certified safety zone, max Z/XY spd mm/sec	<p>These parameters define the maximum speeds that are permitted for the Certified Speed Restrict Safety Zones. The first value is the maximum downward Z speed (when within the safety zone) in mm/sec. Since this is a downward speed, it should be a negative value and defaults to -200. The second value is the maximum permitted speed in the horizontal XY plane (when within the safety zone), and defaults to 200 mm/sec.</p>

Appendices

Appendix A: Frequently Asked Questions

This section contains a compilation of frequently asked questions related to the Guidance Development Environment.

How do I transfer my GDS license to another host computer?

The Guidance Development Suite and its components (e.g. the Guidance Development Environment) can be used to develop software applications for multiple controllers but is only licensed to execute on a single PC. In general, if you need to execute GDS on additional PCs, additional licenses must be purchased.

However, if the PC on which GDS is licensed to execute is being obsoleted, you can transfer a purchased GDS license to a new PC.

NOTE: As soon as you execute the following procedure, GDS will no longer be permitted to execute on your old PC.

To transfer a purchased GDS license to another PC, perform the following procedure.

1. Follow the standard procedure for installing GDS on the **new** PC.
2. On the **new** PC, execute the procedure for licensing GDE to obtain the PC's ID information. This is needed to issue you a new license for this PC.
3. Execute GDE 2.0 or later on the **old** PC. If you are executing an older version of GDS/GDE, obtain a new version from the Precise Support website and install it on the **old** PC.
4. Select **Help > Product Activation**.
5. Click the **Uninstall** button.
6. Follow the instruction to execute the uninstall procedure.

NOTE: After you execute this procedure, you will no longer be able to execute GDS on the old PC.

7. Email the following information to sales_preciseflex@brooks.com, and request that the license be transferred.

- Your company name
- The name of the individual that the license was issued to, if known
- The original license code displayed by the uninstall procedure
- The uninstall code displayed by the uninstall procedure
- The new PC's ID displayed by the licensing procedure

8. Once you get a new license key, execute the standard activation process on the **new** PC.

This completes the license transfer process.

Appendix B: Guidance Update Wizard

Introduction

If you have multiple controllers or have purchased a new controller and wish to propagate the software from another controller, the Guidance Update Wizard can initialize any or all of the software on the new controller (or group of controllers) to duplicate your previous controller's software state. This wizard primarily copies files from the host PC to the target controller's flash disk, both to the public /flash areas and protected system areas. Software or system parameters that are currently running in the target controller(s) are generally not affected until the controller is rebooted.

Using files stored on the PC's hard drive, this wizard can perform any of the following operations on a target controller(s):

- Create a backup copy on the PC's hard drive of the controller's initial configuration files and all or selected GPL projects.
- Format the user area of the /flash disk to delete all defined files.
- Set the controller's date and time clock to the same value as the PC.
- Copy GPL projects to the /flash/projects folder.
- Copy system configuration (*.PAC) files to the /flash/config folder.
- Upgrade the GPL operating software.
- Upgrade the FPGA firmware.
- Copy any special files or folders to the /flash disk.
- Execute a series of console commands on the controller.

Once an update operation has been defined, it can be saved to the PC's hard drive. Once saved, the update operation can be easily repeated by reloading the definition file or double clicking on the definition file on the PC's hard drive.

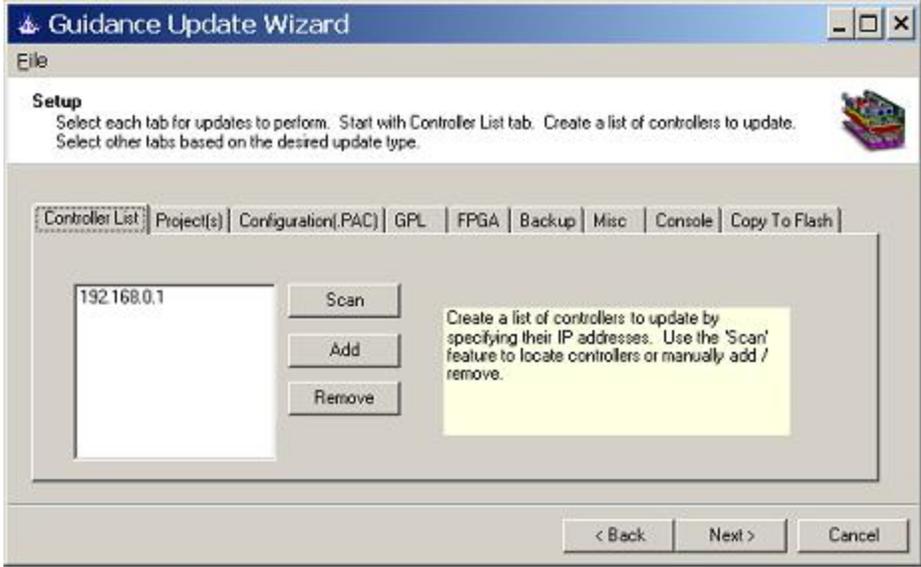
For many operations, such as updating the controller's operating software, the controller must be rebooted to put the changes into effect.

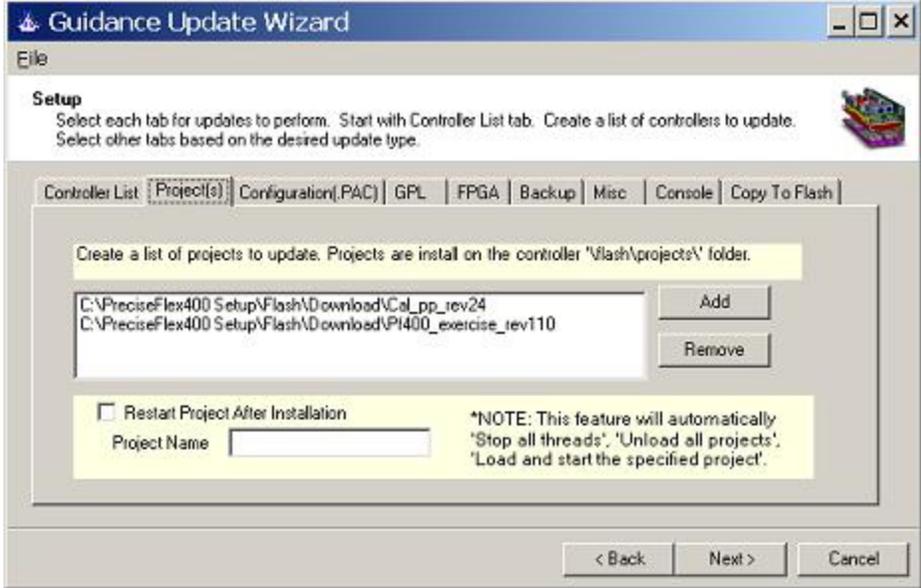
Operating Instructions

To execute the Update Wizard, the following is required:

- A copy of all of the software to be transferred to the target controller(s) must be loaded onto the PC's hard drive. The files need not be in the same folder or file path.
- A copy of the Guidance Develop Suite (GDS) version 3.1 or later must be installed on your PC. To execute this wizard, it is not necessary to purchase a licensed copy of GDS since the Update Wizard will operate without a license.
- An adequate amount of free space in the controller's flash disk to store the GPL projects and other files that are to be transferred.

To update one or more controllers, perform the following procedure:

Step	Action
1.	<p>Launch the wizard by clicking Start > Programs > Precise Automation > GDS x.x > Guidance Update Wizard.</p> 
2.	<p>Click the Next button to advance to the Setup page.</p>
3.	<p>Fill in the first tab of the Setup page, Controller List, with the IP addresses of the controllers to be updated. If more than one IP address is specified, multiple controllers will be sequentially updated. At least one IP address must always be specified.</p> 
4.	<p>Fill in the tabs for the update operations to be performed. The purpose and operation of each tab will be briefly described below.</p>

Step	Action
5.	<p>The Projects tab copies entire GPL projects to the /flash/projects folder of the controller's flash drive. In the picture below, two GPL projects are copied from the PC: "Cal_pp_rev24" and "Pf400_exercise_rev110."</p> 

The **Configuration(PAC)** tab can replace selected or all of the *.PAC files in the /flash/config folder. These files define the operating configuration of the controller. In the following picture, all of the PAC files are over-written by files for the PreciseFlex 400 Sample Handler.

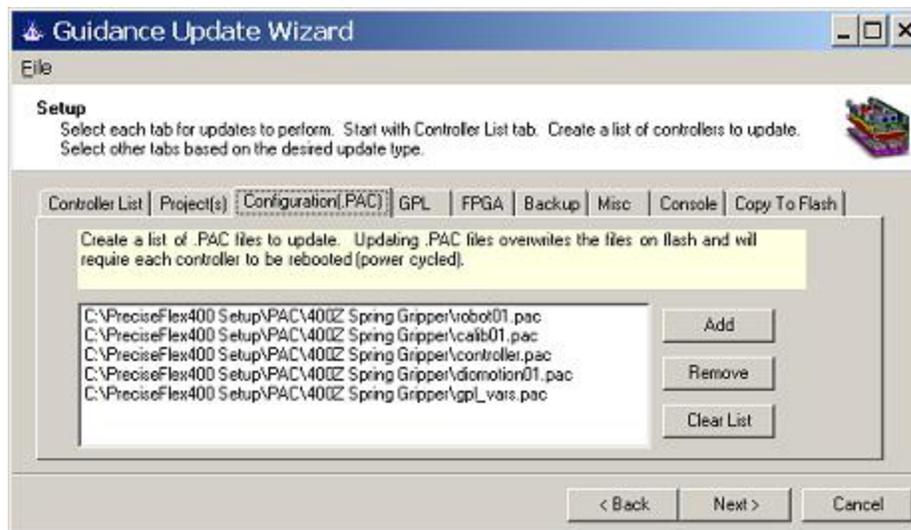


Figure 5-7: "Configuration PAC" Tab

The **GPL** tab can update the controller's GPL operating system with a different version of code. The GPL system is typically distributed in a file with a ".bin" extension.

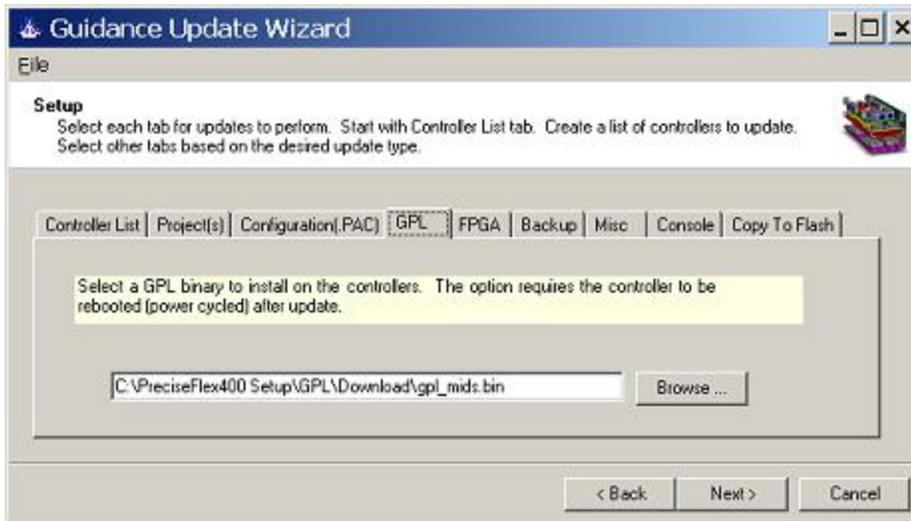


Figure 5-8: "GPL" Tab

The **FPGA** tab can update the controller's FPGA firmware. The FPGA firmware is typically distributed in a file with a ".bin" extension.

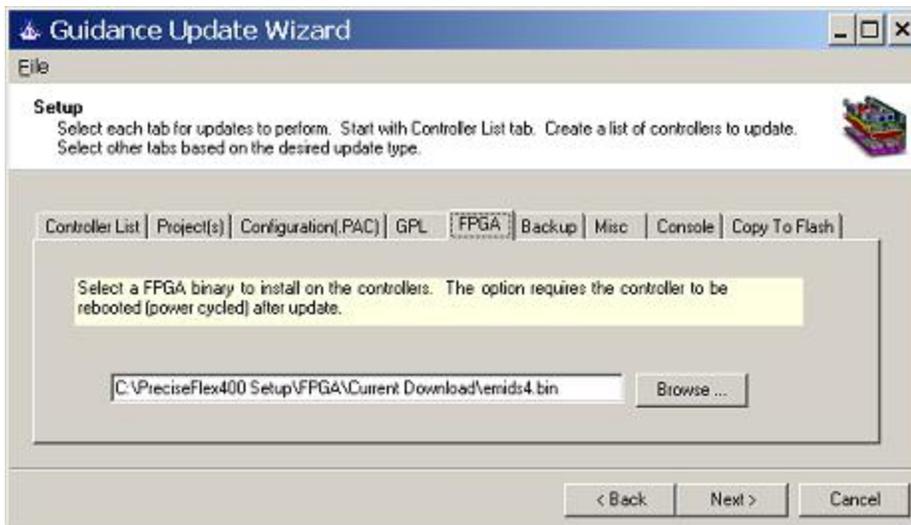


Figure 5-9: "FPGA" Tab

The Backup tab can save a backup copy on the PC's hard drive of the controller's initial configuration files and all or selected GPL projects. To define specific projects to save, click the Options button. This tab also

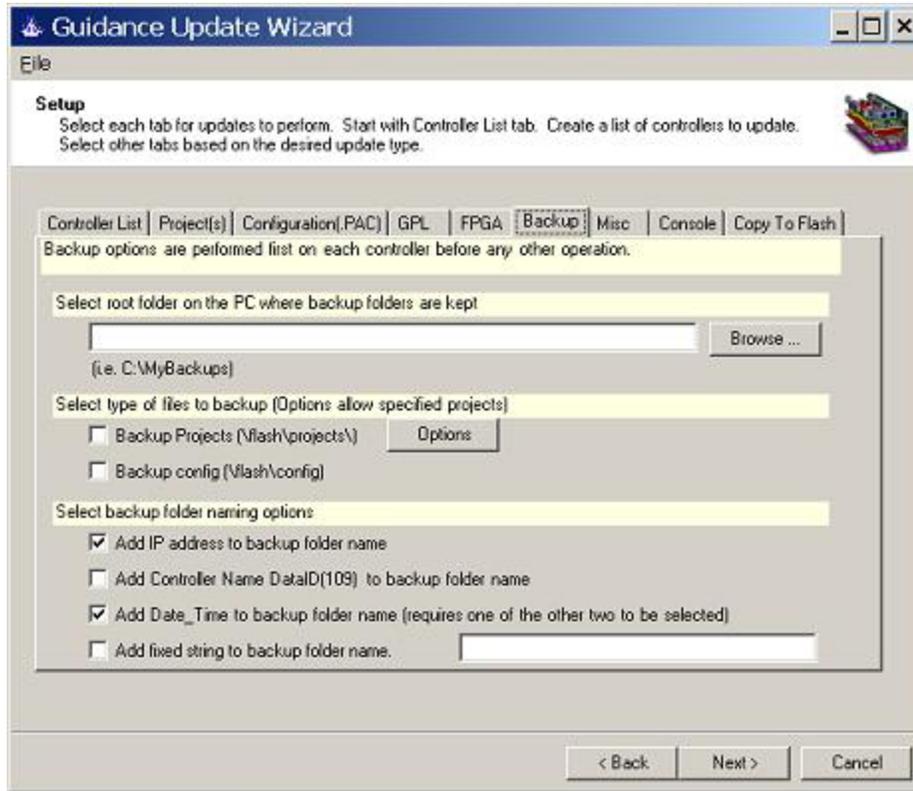


Figure 5-10: "Backup" Tab

The **Misc.** tab permits the user to perform miscellaneous operations, such as formatting the user flash area to clean it out before writing other files and to set the controller's date and time clock to the same value as that of the PC.

 WARNING	
<p>Changing the controller's date and time clock when running GPL versions before 4.0 can cause erratic motion control behavior. For these GPL systems, motor power should be disabled when the date and time clock is modified, and the controller should be rebooted after the update process is completed.</p>	

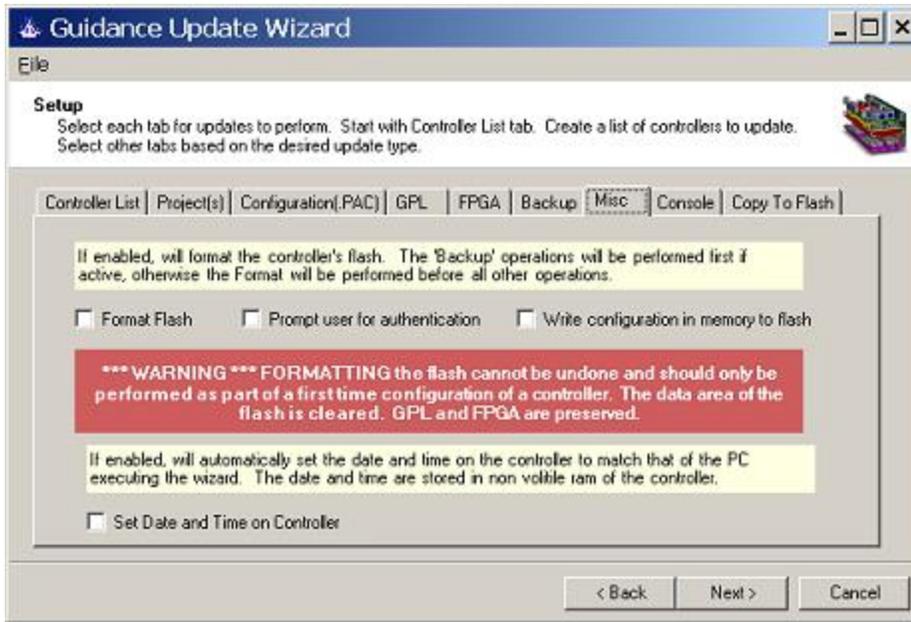


Figure 5-11: "Misc." Tab

The **Console** tab permits you to define special console commands that you wish to execute during the update process.

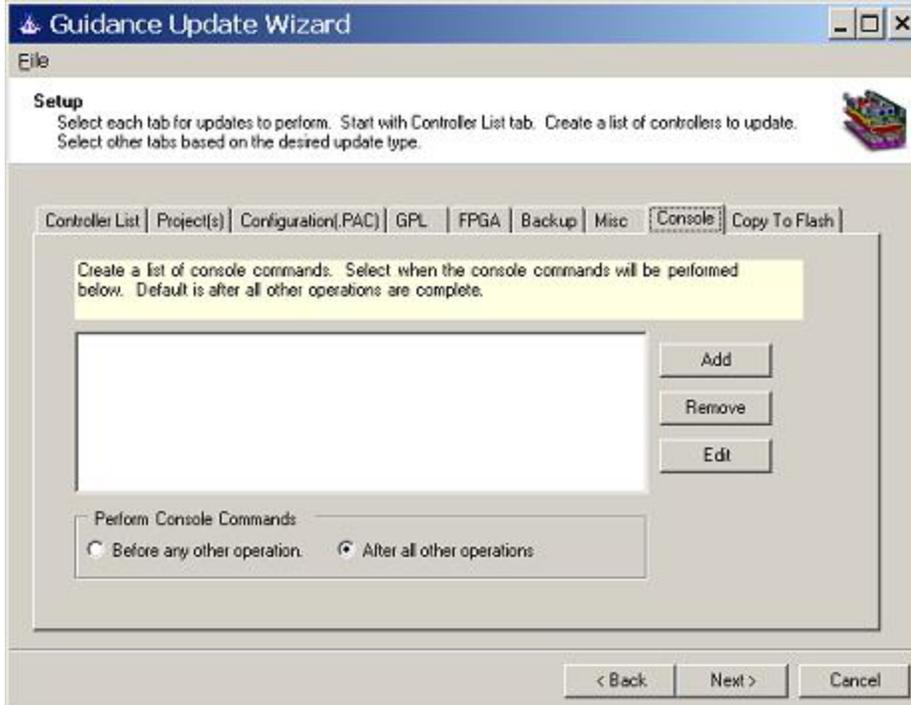


Figure 5-12: "Console" Tab

The **Copy To Flash** tab permits you to copy files into the flash disk that are not supported by the other built-in methods. For example, you can use this tab to copy a web application into the /flash/appweb folder. To specify both the source and destination for this copy operation, there is a special pop-up that is utilized for this function that is shown below.

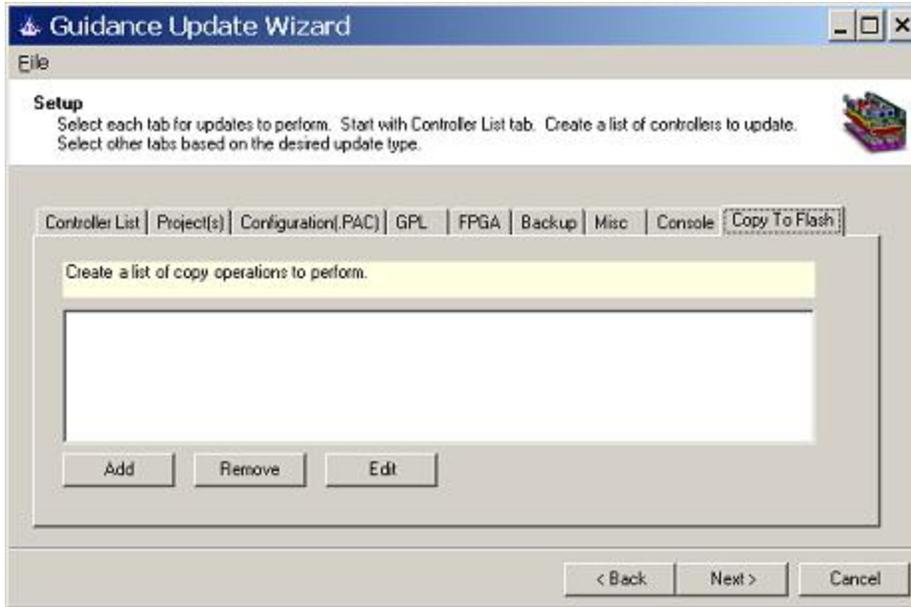


Figure 5-13: "Copy to Flash" Tab

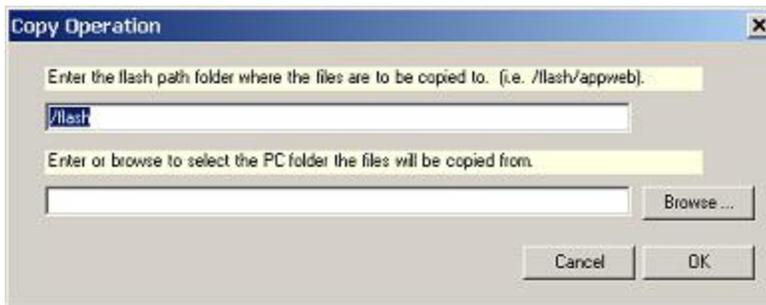
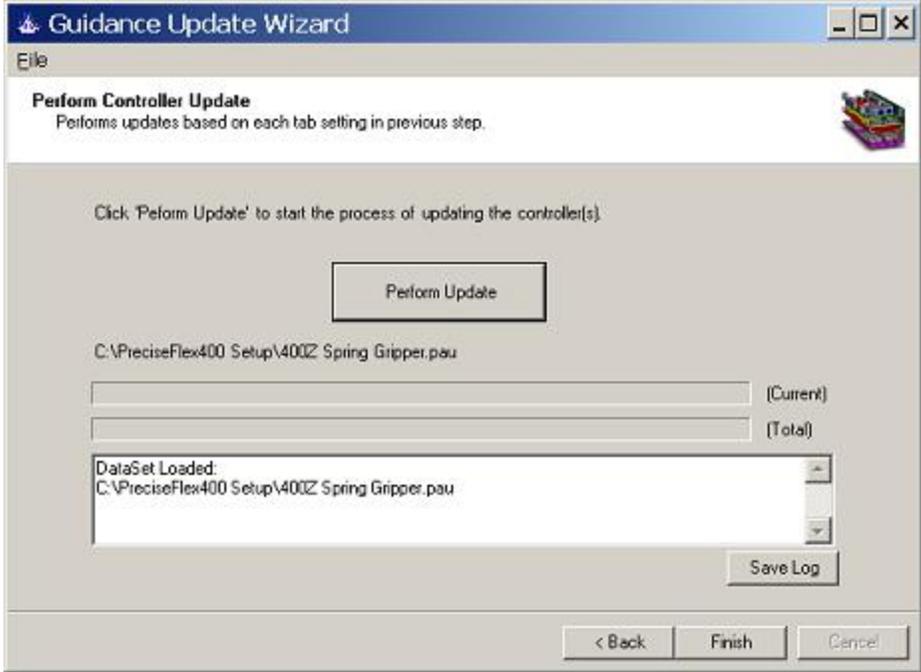


Figure 5-14: "Copy to Flash" Window

This completes defining the operations for the Update Wizard to perform. To save this to be repeated in the future, perform the following procedure:

Step	Action
1.	Select File > Save Data Set . This creates a "*.pau" file on the PC's hard drive.
2.	Click the Next button. This advances the user to the final step.

Step	Action
3.	<p>To execute the update process, click on Perform Update as shown below. The progress bars indicate how quickly the update operation is proceeding. Normally, the complete operation takes just a few minutes or less per controller.</p> 
4.	Once the wizard is completed, restart the controller(s) to have the changes put into effect.

Automatic Operation

The Guidance Update can be executed from the Windows Command Prompt, a script or a shortcut and have it automatically load a data set and run it. Perform the following procedure:

Step	Action
1.	Locate the folder that contains GuidanceUpdate.exe. The full path for this executable is typically "C:\Program Files (x86)\Precise Automation\Guidance Development Suite x.x\GU\GuidanceUpdate.exe."
2.	Issue a command that has the form "<full path>\GuidanceUpdate.exe" "<full path for dataset>" -autorun. The "" are necessary in the command specification if any of the path's contain space characters. The -autorun switch indicates that the data set is to be automatically executed. If you omit this switch, the data set will be loaded but the window for manually initiating its execution will be displayed.